Simscape™ Driveline™

Reference
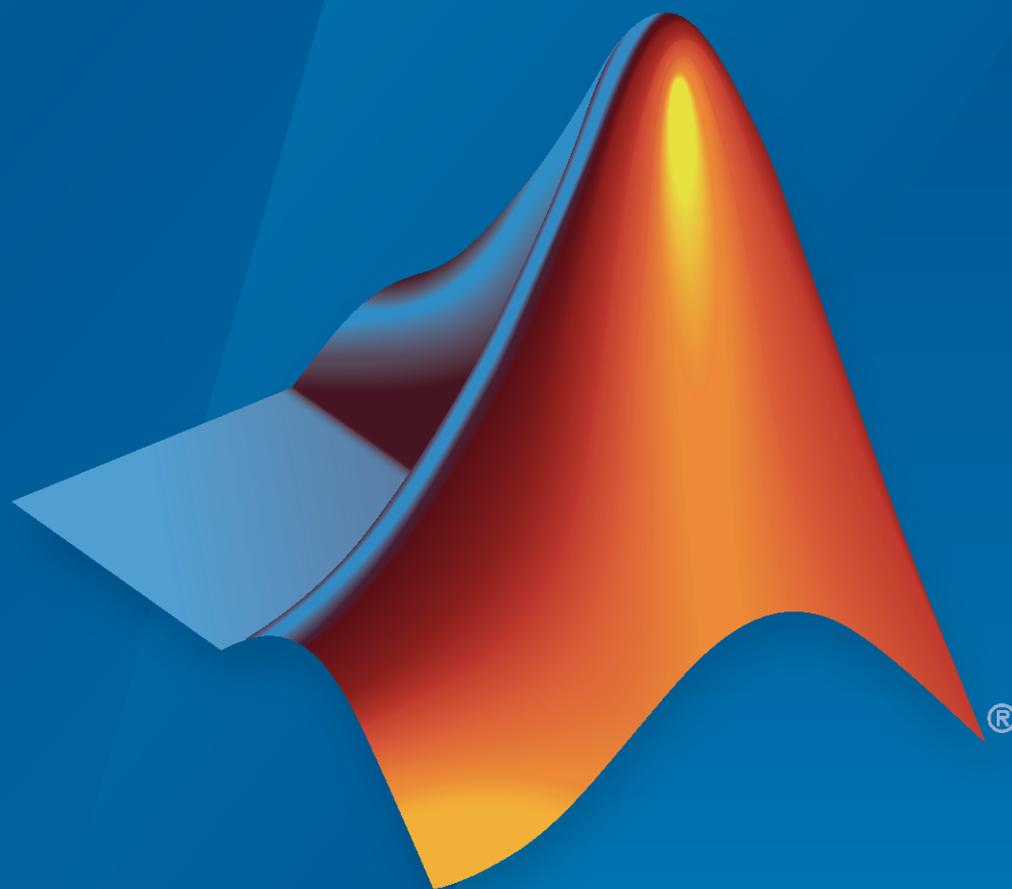
MATLAB&SIMULINK®

MathWorks®

How to Contact MathWorks

Latest news: www.mathworks.com

Sales and services: www.mathworks.com/sales_and_services

User community: www.mathworks.com/matlabcentral

Technical support: www.mathworks.com/support/contact_us

Phone: 508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Contents

Blocks

1

Technical Conventions

A

Parameter Dependencies

B

Bibliography

C

Blocks

4-Speed CR-CR

Clutch schedule for a four-speed carrier ring-carrier ring transmission
Library: Simscape / Driveline / Transmissions
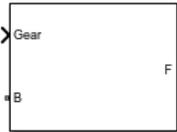
Description

The 4-Speed CR-CR block consists of two planetary gear sets and five disk friction clutches. The follower shaft connects to the planet gear carrier of the output planetary gear and the ring gear of the input planetary gear. The clutches are configured to implement four ratios, one reverse ratio, and neutral. The reverse clutch and two of the clutches determine which gears the base shaft connects to. The other two clutches act as brakes, grounding various gears of the planetary sets to the transmission housing.

This diagram shows a four-speed carrier ring-carrier ring transmission. The labels for the gear components are superimposed on the input and output gears. The table lists the gear and clutch components that are labeled in the diagram.
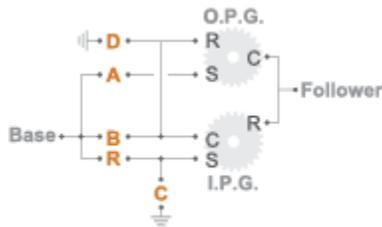
Label	Component
I.P.G.	Input planetary gear
O.P.G.	Output planetary gear
R	Ring gear
C	Planet gear carrier
S	Sun gear
R	Reverse clutch
A–B	Forward clutches that control the power flow path
C–D	Forward, braking clutches

Drive Ratios, Clutch Schedule, and Power Flow

The drive ratio between the transmission input and output shafts follows from the elementary gear ratios specified for the gear blocks. The elementary gear ratios are

$$g_1 = \frac{N_{RI}}{N_{SI}},$$

and

$$g_2 = \frac{N_{RO}}{N_{SO}},$$

where:

- N_{RI} is the number of teeth in the planetary ring gear on the input shaft side.
- N_{SI} is the number of teeth in the planetary sun gear on the input shaft side.
- N_{RO} is the number of teeth in the planetary ring gear on the output shaft side.
- N_{SO} is the number of teeth in the planetary sun gear on the output shaft side.

The table shows the clutch schedule, drive-ratio expressions, drive-ratio default values, and the power-flow diagrams for each gear of the 4-Speed CR-CR block.

The letters in the clutch schedule columns denote the brakes and clutches. A value of 1 denotes a locked state and a value of 0 an unlocked state. The clutch schedule generates these signals based on the Gear port input signal. The signals are scaled through a Gain block and used as actuation inputs in the clutch blocks.

The power-flow diagrams show the power flow paths between input and output shafts for each gear setting. Power flow is shown in orange. Connections to the transmission housing (a mechanical ground) are shown in black.

Gear	Clutch Schedule					Drive Ratio Equation	Default Ratio	Power Flow
	A	B	C	D	R			
4	0	1	1	0	0	$\dfrac{g_1}{1 + g_1}$	0.67	
3	1	1	0	0	0	1	1	

Gear	Clutch Schedule					Drive Ratio Equation	Default Ratio	Power Flow
2	**1**	0	**1**	0	0	$\dfrac{g_1 + g_2 + 1}{1 + g_1}$	1.67	
1	**1**	0	0	**1**	0	$g_2 + 1$	3	
R	0	0	0	**1**	**1**	$-g_1$	-2	
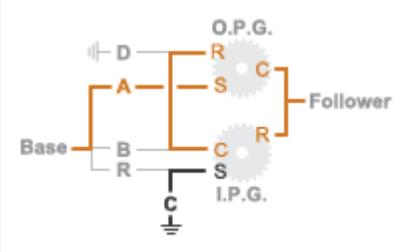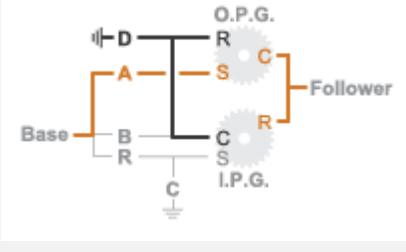
Ports

Input

Gear — Numerical input for the gear
scalar

Specify -1 for reverse, 0 for neutral, and 1–4 for the forward gears.

Data Types: single | double

Conserving

B — Base interface
mechanical rotational

Conserving port associated with the base shaft.

F — Follower interface
mechanical rotational

Conserving port associated with the follower shaft.

Parameters

`Initial gear` — Gear state at the beginning of simulation
`Neutral` (default) | `4` | `3` | `2` | `1` | `Reverse`

Sets the initial engagement or disengagement of the clutches using the clutch schedule. When the transmission is in neutral, there is no power flow from the base shaft to the follower shaft.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

4-Speed Ravigneaux | 6-Speed Lepelletier | 7-Speed Lepelletier | 8-Speed | 9-Speed | 10-Speed

Introduced in R2015a

4-Speed Ravigneaux

Clutch schedule for a four-speed Ravigneaux transmission
Library: Simscape / Driveline / Transmissions

Description

The 4-Speed Ravigneaux block consists of one Ravigneaux gear set and five clutches. The follower shaft connects to the ring gear of the Ravigneaux set. Three of the clutches determine the gears that the base shaft connects to for power transfer. The other two clutches serve as brakes, grounding various gears of the Ravigneaux set to the transmission housing.

This diagram shows a four-speed Ravigneaux gear transmission. The labels for the gear components are superimposed on the input and output gears. The table lists the gear and clutch components that are labeled in the diagram.
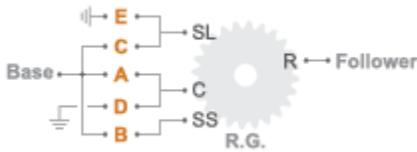
Label	Component
R.G.	Ravigneaux gear
R	Ring gear
C	Planet gear carrier
SL	Large sun gear
SS	Small sun gear
A–C	Forward clutches that control the power flow path
D–E	Forward, braking clutches

Drive Ratios, Clutch Schedule, and Power Flow

The drive ratio between the transmission input and output shafts follows from the elementary gear ratios specified for the gear blocks. The elementary gear ratios are

$$g_1 = \frac{N_R}{N_{SL}},$$

and

$$g_2 = \frac{N_R}{N_{SS}},$$

where:

- N_R is the number of teeth in the Ravigneaux ring gear.
- N_{SL} is the number of teeth in the Ravigneaux large sun gear.
- N_{SS} is the number of teeth in the Ravigneaux small sun gear.

The table shows the clutch schedule, drive-ratio expressions, drive-ratio default values, and the power-flow diagrams for each gear of the 4-Speed Ravigneaux block.

The letters in the clutch schedule columns denote the brakes and clutches. A value of 1 denotes a locked state and a value of 0 an unlocked state. The clutch schedule generates these signals based on the Gear port input signal. The signals are scaled through a Gain block and used as actuation inputs in the clutch blocks.

The power-flow diagrams show the power flow paths between input and output shafts for each gear setting. Power flow is shown in orange. Connections to the transmission housing (a mechanical ground) are shown in black.

Gear	A	B	C	D	E	Drive Ratio Equation	Default Ratio	Power Flow
4	1	0	0	0	1	$\dfrac{g_1}{1+g_1}$	0.67	
3	1	1	0	0	0	1	1	
2	0	1	0	0	1	$\dfrac{g_1+g_2}{1+g_1}$	1.67	
1	0	1	0	1	0	g_2	3	
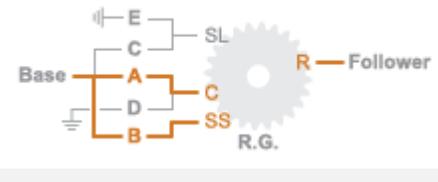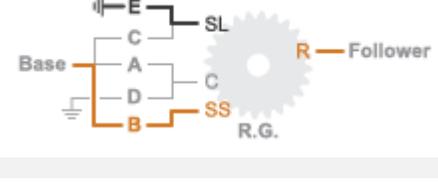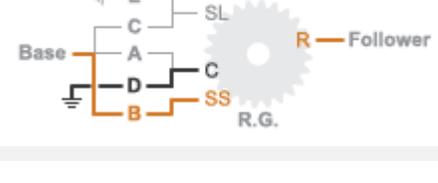
Gear	Clutch Schedule					Drive Ratio Equation	Default Ratio	Power Flow
E	0	0	**1**	**1**	0	$-g_1$	-2	

Ports

Input

`Gear` — Numerical input for the gear
scalar

Specify -1 for reverse, 0 for neutral, and 1–4 for the forward gears.

Data Types: `single` | `double`

Conserving

B — Base interface
mechanical rotational

Conserving port associated with the base shaft.

F — Follower interface
mechanical rotational

Conserving port associated with the follower shaft.

Parameters

`Initial gear` — Gear state at the beginning of simulation
`Neutral` (default) | `4` | `3` | `2` | `1` | `Reverse`

Sets the initial engagement or disengagement of the clutches using the clutch schedule. When the transmission is in neutral, there is no power flow from the base shaft to the follower shaft.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
4-Speed CR-CR | 6-Speed Lepelletier | 7-Speed Lepelletier | 8-Speed | 9-Speed | 10-Speed

Introduced in R2015a

6-Speed Lepelletier

Clutch schedule for a six-speed Lepelletier transmission
Library: Simscape / Driveline / Transmissions
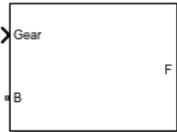
Description

The 6-Speed Lepelletier consists of one planetary gear set, one Ravigneaux gear set, and five disk friction clutches. The transmission base shaft connects to the ring gear of the planetary gear set. The follower shaft connects to the ring gear of the Ravigneaux gear set. Three of the clutches control the power flow paths between the planetary and Ravigneaux gear sets. The other two clutches serve as brakes, grounding various gears of the Ravigneaux set to the transmission housing.

This diagram shows a six-speed Lepelletier transmission. The labels for the gear components are superimposed on the input and output gears. The table lists the gear and clutch components that are labeled in the diagram.

Label	Component
P.G.	Planetary gear
R.G.	Ravigneaux gear
R	Ring gear
C	Planet gear carrier
S	Sun gear
SL	Large sun gear
SS	Small sun gear
A–C	Forward clutches that control the power flow path
D–E	Forward, braking clutches

Drive Ratios, Clutch Schedule, and Power Flow

The drive ratio between the transmission input and output shafts follows from the elementary gear ratios specified for the gear blocks. The elementary gear ratios are

$$g_1 = \frac{N_{PR}}{N_{PS}},$$

$$g_2 = \frac{N_{RR}}{N_{RSL}},$$

and

$$g_3 = \frac{N_{RR}}{N_{RSS}},$$

where:

- N_{PR} is the number of teeth in the planetary ring gear.
- N_{PS} is the number of teeth in the planetary sun gear.
- N_{RR} is the number of teeth in the Ravigneaux ring gear.
- N_{RSL} is the number of teeth in the Ravigneaux large sun gear.
- N_{RRS} is the number of teeth in the Ravigneaux small sun gear.

The table shows the clutch schedule, drive-ratio expressions, drive-ratio default values, and the power-flow diagrams for each gear of the 6-Speed Lepelletier block.

The letters in the clutch schedule columns denote the brakes and clutches. A value of 1 denotes a locked state and a value of 0 an unlocked state. The clutch schedule generates these signals based on the Gear port input signal. The signals are scaled through a Gain block and used as actuation inputs in the clutch blocks.

The power-flow diagrams show the power flow paths between input and output shafts for each gear setting. Power flow is shown in orange. Connections to the transmission housing (a mechanical ground) are shown in black.

Gear	Clutch Schedule					Drive Ratio Equation	Default Ratio	Power Flow
	A	**B**	**C**	**D**	**E**			
6	0	1	0	0	1	$\dfrac{g_2}{1 + g_2}$	0.69	
5	1	1	0	0	0	$\dfrac{g_2}{g_2 + 1/(1 + g_1)}$	0.87	
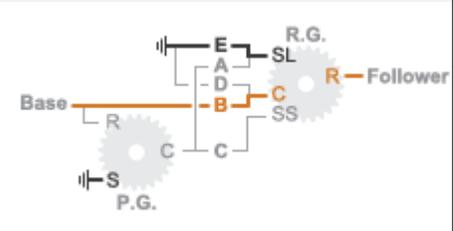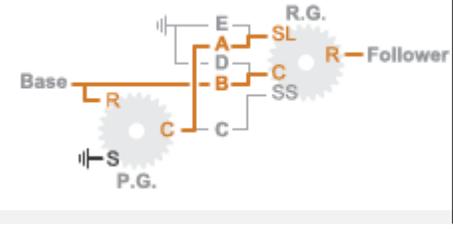
Gear	Clutch Schedule					Drive Ratio Equation	Default Ratio	Power Flow
4	0	**1**	**1**	0	0	$\dfrac{g_3(1 + g_1)}{g_3(1 + g_1) - 1}$	1.14	
3	**1**	0	**1**	0	0	$\dfrac{1 + g_1}{g_1}$	1.50	
2	0	0	**1**	0	**1**	$\dfrac{(1 + g_1)(g_3 + g_2)}{g_1(1 + g_2)}$	2.31	
1	0	0	**1**	**1**	0	$\dfrac{g_3(1 + g_1)}{g_1}$	4.12	
R	**1**	0	0	**1**	0	$-\dfrac{g_2(1 + g_1)}{g_1}$	-3.36	
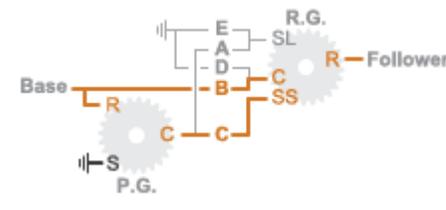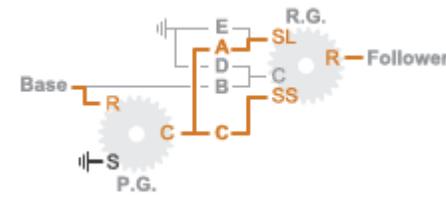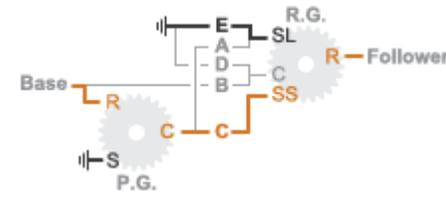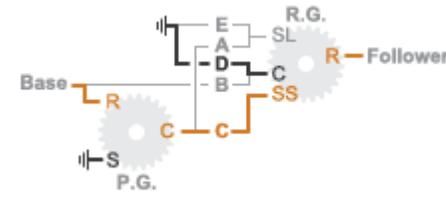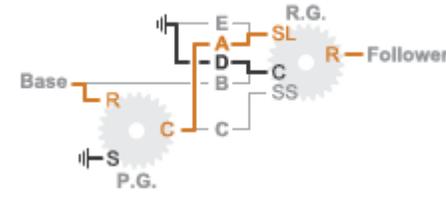
Ports

Input

Gear — Numerical input for the gear
scalar

Specify -1 for reverse, 0 for neutral, and 1–6 for the forward gears.

Data Types: `single` | `double`

Conserving

B — Base interface
mechanical rotational

Conserving port associated with the base shaft.

F — Follower interface
mechanical rotational

Conserving port associated with the follower shaft.

Parameters

`Initial gear` **— Gear state at the beginning of simulation**
`Neutral` (default) | `6` | `5` | `4` | `3` | `2` | `1` | `Reverse`

Sets the initial engagement or disengagement of the clutches using the clutch schedule. When the transmission is in neutral, there is no power flow from the base shaft to the follower shaft.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

4-Speed CR-CR | 4-Speed Ravigneaux | 7-Speed Lepelletier | 8-Speed | 9-Speed | 10-Speed

Introduced in R2015a

7-Speed Lepelletier

Clutch schedule for a seven-speed Lepelletier transmission
Library: Simscape / Driveline / Transmissions

Description

The 7–Speed Lepelletier block consists of one planetary gear set, one Ravigneaux gear set, and six disk friction clutches. The transmission base shaft connects to the ring gear of the planetary gear set. The follower shaft connects to the ring gear of the Ravigneaux gear set. Three of the clutches control the power-flow paths between the planetary and Ravigneaux gear sets. The other three clutches serve as brakes, grounding various gears of the Ravigneaux set to the transmission housing.

This diagram shows a seven-speed Lepelletier transmission. The labels for the gear components are superimposed on the input and output gears. The table lists the gear and clutch components that are labeled in the diagram.
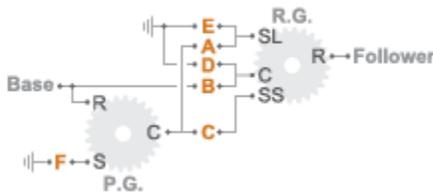
Label	Component
P.G.	Planetary gear
R.G.	Ravigneaux gear
R	Ring gear
C	Planet gear carrier
S	Sun gear
SL	Large sun gear
SS	Small sun gear
A–C	Forward clutches that control the power-flow path
D–F	Forward, braking clutches

Drive Ratios, Clutch Schedule, and Power Flow

The drive ratio between the transmission input and output shafts follows from the elementary gear ratios specified for the gear blocks. The elementary gear ratios are

$$g_1 = \frac{N_{PR}}{N_{PS}},$$

$$g_2 = \frac{N_{RR}}{N_{RSL}},$$

and

$$g_3 = \frac{N_{RR}}{N_{RSS}},$$

where:

- N_{PR} is the number of teeth in the planetary ring gear.
- N_{PS} is the number of teeth in the planetary sun gear.
- N_{RR} is the number of teeth in the Ravigneaux ring gear.
- N_{RSL} is the number of teeth in the Ravigneaux large sun gear.
- N_{RSS} is the number of teeth in the Ravigneaux small sun gear.

The table shows the clutch schedule, drive-ratio expressions, drive-ratio default values, and the power-flow diagrams for each gear of the 7-Speed Lepelletier block.

The letters in the clutch schedule columns denote the brakes and clutches. A value of 1 denotes a locked state and a value of 0 an unlocked state. The clutch schedule generates these signals based on the Gear port input signal. The signals are scaled through a Gain block and used as actuation inputs in the clutch blocks.

The power-flow diagrams show the power-flow paths between input and output shafts for each gear setting. Power flow is shown in orange. Connections to the transmission housing (a mechanical ground) are shown in black.

Gear	Clutch Schedule						Drive Ratio Equation	Default Ratio	Power Flow
	A	B	C	D	E	F			
7	0	1	0	0	1	1	$\dfrac{g_2}{1+g_2}$	0.70	
6	1	1	0	0	0	1	$\dfrac{g_2(1+g_1)}{g_2(1+g_1)+1}$	0.86	
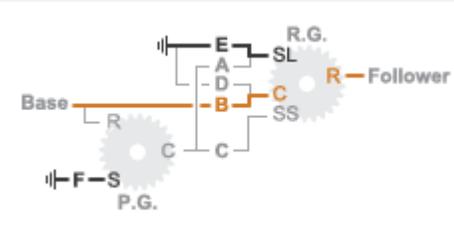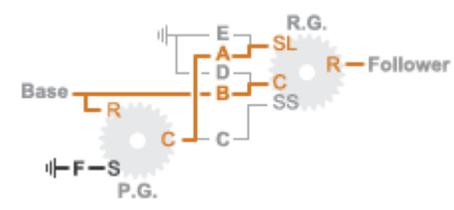
Gear	Clutch Schedule						Drive Ratio Equation	Default Ratio	Power Flow
5	1	1	1	0	0	0	1	1	
4	0	1	1	0	0	1	$\dfrac{g_3(1 + g_1)}{g_3(1 + g_1) - 1}$	1.17	
3	1	0	1	0	0	1	$\dfrac{1 + g_1}{g_1}$	1.63	
2	0	0	1	0	1	1	$\dfrac{(1 + g_1)(g_3 + g_2)}{g_1(1 + g_2)}$	2.43	
1	0	0	1	1	0	1	$\dfrac{g_3(1 + g_1)}{g_1}$	4.31	
R	1	0	0	1	0	1	$-\dfrac{g_2(1 + g_1)}{g_1}$	-3.82	

Ports

Input

Gear — Numerical input for the gear
scalar

Specify -1 for reverse, 0 for neutral, and 1–7 for the forward gears.

Data Types: `single` | `double`

Conserving

B — Base interface
mechanical rotational

Conserving port associated with the base shaft.

F — Follower interface
mechanical rotational

Conserving port associated with the follower shaft.

Parameters

Initial gear — Gear state at the beginning of simulation
`Neutral` (default) | `7` | `6` | `5` | `4` | `3` | `2` | `1` | `Reverse`

Sets the initial engagement or disengagement of the clutches using the clutch schedule. When the transmission is in neutral, there is no power flow from the base shaft to the follower shaft.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

4-Speed CR-CR | 4-Speed Ravigneaux | 6-Speed Lepelletier | 8-Speed | 9-Speed | 10-Speed

Introduced in R2015a

8-Speed

Clutch schedule for an eight-speed transmission
Library: Simscape / Driveline / Transmissions
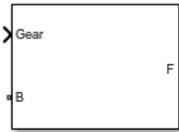
Description

The 8-Speed block consists of four planetary gear sets and five clutches. The base shaft connects to the:

- Planet gear carrier of the second planetary gear
- Other gears through three clutches that determine the power flow path.

The follower shaft connects to the planet gear carrier of the first and fourth planetary gears. The other two clutches serve as brakes, grounding various gears of planetary sets two and three to the transmission housing.

This diagram shows an eight-speed transmission. The labels for the gear components are superimposed on the input and output gears. The table lists the gear and clutch components that are labeled in the diagram.
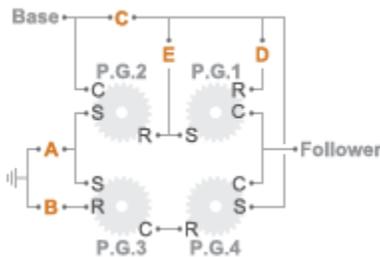
Label	Component
P.G.1–P.G.4	Planetary gears, 1–4
R	Ring gear
C	Planet gear carrier
S	Sun gear
C–E	Forward clutches that control the power flow path
A–B	Forward, braking clutches

Drive Ratios, Clutch Schedule, and Power Flow

The drive ratio between the transmission input and output shafts follows from the elementary gear ratios specified for the gear blocks. The elementary gear ratios are

$$g_x = \frac{N_{R_x}}{N_{S_x}},$$

where:

- N_{R_x} is the number of teeth in the planetary ring gear x, where x = 1, 2, 3, and 4.
- N_{S_x} is the number of teeth in the planetary sun gear x, where x = 1, 2, 3, and 4.

The table shows the clutch schedule, drive-ratio expressions, drive-ratio default values, and the power-flow diagrams for each gear of the 8-Speed block. The schedule and gear ratios are based on the manufacturer data for the GM 8L90 eight-speed automatic transmission.

The letters in the clutch schedule columns denote the brakes and clutches. A value of 1 denotes a locked state and a value of 0 an unlocked state. The clutch schedule generates these signals based on the Gear port input signal. The signals are scaled through a Gain block and used as actuation inputs in the clutch blocks.

The power-flow diagrams show the power flow paths between input and output shafts for each gear setting. Power flow is shown in orange. Connections to the transmission housing (a mechanical ground) are shown in black.

Gear	Clutch Schedule					Drive Ratio Equation	Default Ratio	Power Flow
	A	**B**	**C**	**D**	**E**			
8	**1**	0	0	**1**	**1**	$\dfrac{g_2}{1 + g_2}$	0.65	
7	**1**	0	**1**	**1**	0	$\dfrac{g_2(1 + g_1)}{1 + g_2(1 + g_1)}$	0.85	
6	0	0	**1**	**1**	**1**	1	1	

Gear	Clutch Schedule					Drive Ratio Equation	Default Ratio	Power Flow
5	0	1	1	1	0	$1 + \dfrac{g_3 g_4}{(1 + g_3 + g_4 + g_2 g_4 (1 + g_1))}$	1.27	
4	0	1	0	1	1	$\dfrac{1 + g_2 + g_3}{1 + g_2}$	1.69	
3	0	1	1	0	1	$\dfrac{(1 + g_3)(1 + g_4)}{1 + g_3 + g_4}$	2.07	
2	1	1	0	0	1	$\dfrac{g_2(g_4 + 1)}{(1 + g_2)}$	2.96	
1	1	1	1	0	0	$1 + g_4$	4.55	

Gear	Clutch Schedule					Drive Ratio Equation	Default Ratio	Power Flow
R	**1**	**1**	0	**1**	0	$-\dfrac{g_2(g_1 g_4 - 1)}{(g_2 + 1)}$	-3.91	

Ports

Input

`Gear` — Numerical input for the gear
scalar

Specify -1 for reverse, 0 for neutral, and 1–8 for the forward gears.

Data Types: `single` | `double`

Conserving

B — Base interface
mechanical rotational

Conserving port associated with the base shaft.

F — Follower interface
mechanical rotational

Conserving port associated with the follower shaft.

Parameters

`Initial gear` — Gear state at the beginning of simulation
`Neutral` (default) | `8` | `7` | `6` | `5` | `4` | `3` | `2` | `1` | `Reverse`

Sets the initial engagement or disengagement of the clutches using the clutch schedule. When the transmission is in neutral, there is no power flow from the base shaft to the follower shaft.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

4-Speed CR-CR | 4-Speed Ravigneaux | 6-Speed Lepelletier | 7-Speed Lepelletier | 9-Speed | 10-Speed

Introduced in R2015a

9-Speed

Clutch schedule for a nine-speed transmission
Library: Simscape / Driveline / Transmissions
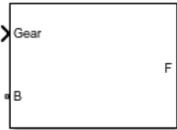
Description

The 9-Speed block consists of four planetary gear sets and six clutches. The follower shaft connects to the planet gear carrier of the fourth planetary gear. Three of the clutches determine the power flow path for the base shaft. The other three clutches serve as brakes, grounding various gears of the planetary sets to the transmission housing.

This diagram shows a nine-speed transmission. The labels for the gear components are superimposed on the input and output gears. The table lists the gear and clutch components that are labeled in the diagram.
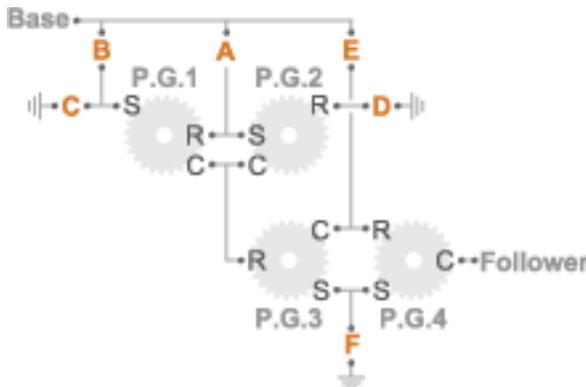
Label	Component
P.G.1–P.G.4	Planetary gears, 1–4
R	Ring gear
C	Planet gear carrier
S	Sun gear
A, B, E	Forward clutches that control the power flow path
C, D, F	Forward, braking clutches

Drive Ratios, Clutch Schedule, and Power Flow

The drive ratio between the transmission input and output shafts follows from the elementary gear ratios specified for the gear blocks. The elementary gear ratios are

$$g_x = \frac{N_{R_x}}{N_{S_x}},$$

where:

- N_{R_x} is the number of teeth in the planetary ring gear x, where x = 1, 2, 3, and 4.
- N_{S_x} is the number of teeth in the planetary sun gear x, where x = 1, 2, 3, and 4.

The table shows the clutch schedule, drive-ratio expressions, drive-ratio default values, and the power-flow diagrams for each gear of the 9-Speed block. The schedule and gear ratios are based on the manufacturer data for the ZF 9HP nine-speed automatic transmission.

The letters in the clutch schedule columns denote the brakes and clutches. A value of 1 denotes a locked state and a value of 0 an unlocked state. The clutch schedule generates these signals based on the Gear port input signal. The signals are scaled through a Gain block and used as actuation inputs in the clutch blocks.

The power-flow diagrams show the power flow paths between input and output shafts for each gear setting. Power flow is shown in orange. Connections to the transmission housing (a mechanical ground) are shown in black.

Gear	Clutch Schedule						Drive Ratio Equation	Default Ratio	Power Flow
	A	B	C	D	E	F			
9	0	1	0	1	1	0	$\dfrac{1 + g_4}{1 + g_3 + g_4 - \left(\frac{g_3}{1 - g_1 g_2}\right)}$		
8	0	0	1	1	1	0	$\dfrac{1 + g_4}{1 + g_3 + g_4}$		

Gear	Clutch Schedule						Drive Ratio Equation	Default Ratio	Power Flow
7	**1**	0	0	**1**	**1**	0	$\dfrac{1 + g_4}{1 + g_3 - \left(\frac{g_3}{1 + g_2}\right) + g_4}$	0	
6	**1**	0	**1**	0	**1**	0	$\dfrac{1 + g_4}{1 + g_3 + g_4 - \left(\frac{g_1 g_3}{1 + g_1}\right)}$	0	
5	**1**	**1**	0	0	**1**	0	1		

Gear	Clutch Schedule						Drive Ratio Equation	Default Ratio	Power Flow
4	**1**	0	0	0	**1**	**1**	$\dfrac{1 + g_4}{g_4}$	1	
3	**1**	**1**	0	0	0	**1**	$\dfrac{(1 + g_3)(1 + g_4)}{g_3 g_4}$	1	
2	**1**	0	**1**	0	0	**1**	$\dfrac{(1 + g_1)(1 + g_3)(1 + g_4)}{g_1 g_3 g_4}$	2	

Gear	Clutch Schedule						Drive Ratio Equation	Default Ratio	Power Flow
1	**1**	0	0	**1**	0	**1**	$\dfrac{(1 + g_2)(1 + g_3)(1 + g_4)}{g_3 g_4}$	4	
R	0	**1**	0	**1**	0	**1**	$\dfrac{(1 - g_1 g_2)(1 + g_3)(1 + g_4)}{g_3 g_4}$	-3	

Ports

Input

Gear — Numerical input for the gear
scalar

Specify -1 for reverse, 0 for neutral, and 1–9 for the forward gears.

Data Types: `single` | `double`

Conserving

B — Base interface
mechanical rotational

Conserving port associated with the base shaft.

F — Follower interface
mechanical rotational

Conserving port associated with the follower shaft.

Parameters

`Initial gear` — Gear state at the beginning of simulation
`Neutral` (default) | `9` | `8` | `7` | `6` | `5` | `4` | `3` | `2` | `1` | `Reverse`

Sets the initial engagement or disengagement of the clutches using the clutch schedule. When the transmission is in neutral, there is no power flow from the base shaft to the follower shaft.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

4-Speed CR-CR | 4-Speed Ravigneaux | 6-Speed Lepelletier | 7-Speed Lepelletier | 8-Speed | 10-Speed

Introduced in R2015a

10-Speed

Clutch schedule for a 10-speed transmission
Library: Simscape / Driveline / Transmissions

Description

The 10-Speed transmission block consists of four planetary gear sets and six clutches. The follower shaft connects to the planet gear carrier of the fourth planetary gear. Four of the clutches determine the power flow path for the base shaft. The other two clutches serve as brakes, grounding various gears of the planetary sets to the transmission housing.

This diagram shows a 10-speed transmission. The labels for the gear components are superimposed on the input and output gears. The table lists the gear and clutch components that are labeled in the diagram. Power elements are shown in orange. Braking elements are shown in black.
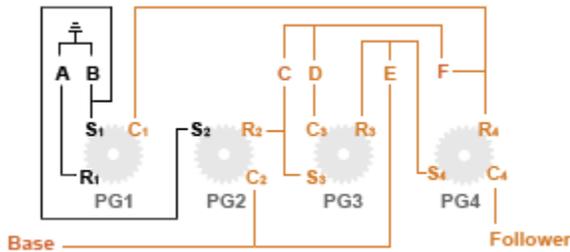
Label	Component
PG1–PG4	Planetary gears, 1–4
R_x	Ring gear of planetary gear x
C_x	Planet gear carrier of planetary gear x
S_x	Sun gear of planetary gear x
C, D, E, F	Clutches that control the power flow path
A, B	Braking clutches

Drive Ratios, Clutch Schedule, and Power Flow

The drive ratio between the transmission input and output shafts follows from the elementary gear ratios specified for the gear blocks. The elementary gear ratios are

$$g_x = \frac{N_{R_x}}{N_{S_x}},$$

where:

- N_{R_x} is the number of teeth in the planetary ring gear x, where x = 1, 2, 3, and 4.
- N_{S_x} is the number of teeth in the planetary sun gear x, where x = 1, 2, 3, and 4

The table shows the clutch schedule, drive-ratio expressions, drive-ratio default values, and the power-flow diagrams for each gear of the 10-Speed block. The schedule and gear ratios are based on the manufacturer data for a 10R80 10-speed automatic transmission.

The letters in the clutch schedule columns denote the brakes and clutches. A value of 1 denotes a locked state and a value of 0 an unlocked state. The clutch schedule generates these signals based on the Gear port input signal. The signals are scaled through a Gain block and used as actuation inputs in the clutch blocks.

The power-flow diagrams show the power flow paths between input and output shafts for each gear setting. Power flow is shown in orange. Connections to the transmission housing (a mechanical ground) are shown in black.

Gear	Clutch Schedule						Drive Ratio Equation	Default Ratio	Power Flow
	A	B	C	D	E	F			
10	0	1	1	1	0	1	$\dfrac{g_2}{1 + g_2}$		
9	0	1	1	0	1	1	$\dfrac{g_2(1 + g_4)}{(g_2(1 + g_4)) + g_4}$		
8	0	1	0	1	1	1	$\dfrac{g_2(1 + g_3)(1 + g_4)}{(g_2(1 + g_3)(1 + g4)) + g_4}$		

Gear	Clutch Schedule						Drive Ratio Equation	Default Ratio	Power Flow
7	0	0	**1**	**1**	**1**	**1**	1		
6	**1**	0	0	**1**	**1**	**1**	$\dfrac{(1 + g_4)(1 + g_1 + g_2(1 + g_3))}{(1 + g_4)(1 + g_2(1 + g_3)) + g_1}$		
5	**1**	0	**1**	0	**1**	**1**	$\dfrac{(1 + g_1 + g_2)(1 + g_4)}{(1 + g_2)(1 + g_4) + g1}$		
4	**1**	0	**1**	**1**	0	**1**	$\dfrac{1 + g_1 + g_2}{1 + g_2}$		

Gear	Clutch Schedule						Drive Ratio Equation	Default Ratio	Power Flow
3	**1**	0	**1**	**1**	**1**	0	$\dfrac{(1+g_1)(1+g_4)}{1+g_1+g_4}$		
2	**1**	**1**	**1**	**1**	0	0	$\dfrac{g_2(1+g_4)}{1+g_2}$		
1	**1**	**1**	0	**1**	**1**	0	$1+g_4$		
R	**1**	**1**	0	**1**	0	**1**	$\dfrac{-g_2g_3(1+g_4)}{(1+g_2)}$		

Ports

Input

Gear — Numerical input for the gear
scalar

Specify -1 for reverse, 0 for neutral, and 1–10 for the forward gears.

Data Types: `single` | `double`

Conserving

B — Base interface
mechanical rotational

Conserving port associated with the base shaft.

F — Follower interface
mechanical rotational

Conserving port associated with the follower shaft.

Parameters

`Initial gear` — Gear state at the beginning of simulation
`Neutral` (default) | `10` | `9` | `8` | `7` | `6` | `5` | `4` | `3` | `2` | `1` | `Reverse`

Sets the initial engagement or disengagement of the clutches using the clutch schedule. When the transmission is in neutral, there is no power flow from the base shaft to the follower shaft.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

4-Speed CR-CR | 4-Speed Ravigneaux | 6-Speed Lepelletier | 7-Speed Lepelletier | 8-Speed | 9-Speed

Introduced in R2018a

Air Spring

Sealed translational air spring
Library: Simscape / Driveline / Couplings & Drives

Description

The Air Spring block represents a generic sealed translational pneumatic spring that isolates equipment from shocks and vibrations. The compressibility of gas gives air springs desirable isolation performance. Air springs are common in automotive and industrial applications where low spring rate and low natural frequency are beneficial. Air springs can help you to:

- Reduce wear and tear on the sprung mass.
- Reduce the wear and tear that the unsprung mass deals to infrastructure.
- Lighten suspension systems to allow for greater total mass.
- Filter out specific unwanted frequencies.

Air springs consist of bellows that confine a column of compressed air or other gas. The air bears the force of the load, and the bellows hold the air. Sealed air springs maintain a constant mass of air, so the increasing load lowers the volume of air and the resulting spring rate. The reverse is also true. The natural frequency of a sealed air spring system also depends on this relationship. Due to the large amount of variability in air spring requirements and performance, manufacturers commonly provide pressure characteristic tables for each model.

Based on the data that you possess, you can set **Parameterization** to `Load as a function of height` or `Stiffness as a function of height`. The Air Spring block uses the information that you enter to create a lookup table. The lookup table allows the block to simulate the nonlinear force response of an arbitrary air spring. This is the effective force to achieve a given height, which is equivalent to the **Load vector** parameter or the element-wise product of the **Stiffness vector** parameter and the **Height vector** parameter. For more information about using lookup tables in Simscape, see `tablelookup`.

The block takes the relative translational motion between port **R** and port **C** to evaluate the air spring height and velocity and uses this information to find the appropriate lookup table entry. The block computes the nonlinear effective force response $\Sigma F(x(t),t)$, such that:

$$\sum F(x(t), t) = -c(x(t))\dot{x}(t) - k(x(t))x(t) + F_{external},$$

where

- $x(t)$ is the height of the spring relative to the undisturbed position.
- $\dot{x}(t)$ is the velocity of the spring.
- $-c(x(t))\dot{x}(t)$ is the viscous damping force. The damping coefficient c varies with spring height.

- *-k(x(t))x(t)* is the stiffness force. The spring coefficient *k* varies with spring height.
- $F_{external}$ is the force transmitted through the ports.

You can choose whether to implement hard stops in your model on the **Hard Stops** tab. When you set **Hard stop at full extension** or **Hard stop at full compression** to On, you can specify a boundary height, the stiffness and damping at that height, and the length of the transition region. The hard stops are equivalent to the Translational Hard Stop block.

Variables

Use the **Variables** tab to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

The sign of the **Force** variable is negative when the load is positive.

Ports

Conserving

C — Follower port
mechanical translational

Mechanical translational conserving port associated with the follower of the air spring. Positive physical signals travel from **R** to **C**.

R — Base port
mechanical translational

Mechanical translational conserving port associated with the base end of the air spring. Positive physical signals travels from **R** to **C**.

Parameters

Spring

Parameterization — Effective force parameterization
Load as a function of height (default) | Stiffness as a function of height

Option to parameterize by height or load. Obtain data for your air spring from manufacturer-supplied specifications or experimental results.

Height vector — Height breakpoints that correspond to load or stiffness data
[5, 7, 11, 14, 17, 20] cm (default) | nonnegative vector

Vector of spring heights. These values correspond to either the load or stiffness values, depending on the way you choose to parameterize the spring. You must specify a vector that has at least three elements.

Load vector — Load for a given height breakpoint
[3000, 2000, 1500, 1200, 800, 100] N (default) | positive vector

Vector of spring loads. This vector must be the same length as the **Height vector** parameter.

Dependencies

To enable this parameter, set **Parameterization** to Load as a function of height.

Stiffness vector — Stiffness for a given height breakpoint
[600, 300, 100, 80, 40, 5] N/cm (default) | positive vector

Vector of spring stiffness values. This vector must be the same length as the **Height vector** parameter.

Dependencies

To enable this parameter, set **Parameterization** to Stiffness as a function of height.

Damping vector — Spring damping for given height breakpoint
[100, 60, 30, 15, 10, 10] s*N/cm (default) | nonnegative vector

Vector of spring damping values. This vector must be the same length as the **Height vector** parameter.

Interpolation method — Method of interpolation between breakpoint values
Smooth (default) | Linear

Lookup table breakpoint interpolation. The block uses the `tablelookup` function to model nonlinearity by using array data to map input values to output values:

- `Linear` — Select this option for the lowest computational cost.
- `Smooth` — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information, see `tablelookup`.

Extrapolation method — Method of handling input values that fall outside range of breakpoint data set
Linear (default) | Nearest | Error

Lookup table breakpoint extrapolation. This method determines the output value when the input value is outside the range specified in the argument list. The block uses the `tablelookup` function to model nonlinearity by using array data to map input values to output values:

- `Linear` — Select this option to produce a curve with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region.
- `Nearest` — Select this option to produce an extrapolation that does not go above the highest point in the data or below the lowest point in the data.
- `Error` — Select this option to avoid extrapolating when you want your data to be within the table range. If the input signal is outside the range of the table, the simulation stops and generates an error.

For more information, see `tablelookup`.

Hard Stops

Hard stop at full extension — Simulate hard stop at maximum spring height
Off (default) | On

Option to enable a hard stop when the air spring is fully extended. The hard stop occurs as the spring reaches the value of the **Maximum height** parameter. For more information, see Translational Hard Stop.

Maximum height — Location of full-extension hard stop
25 cm (default) | positive scalar

Maximum spring height. This parameter determines the location of the full-extension hard stop.

Dependencies

To enable this parameter, set **Hard stop at full extension** to On.

Stiffness at full extension — Full-extension hard stop stiffness
1e8 N/cm (default) | positive scalar

Stiffness to apply as the spring reaches full extension.

Dependencies

To enable this parameter, set **Hard stop at full extension** to On.

Damping at full extension — Full-extension hard stop damping
1e8 s*N/cm (default) | positive scalar

Damping to apply as the spring reaches full extension.

Dependencies

To enable this parameter, set **Hard stop at full extension** to On.

Hard stop at full compression — Simulate hard stop at minimum spring height
Off (default) | On

Option to enable a hard stop when the air spring is fully extended. The hard stop occurs as the spring reaches the value of the **Minimum height** parameter. For more information, see Translational Hard Stop.

Minimum height — Location of full-compression hard stop
5 cm (default) | positive scalar

Minimum spring height. This parameter determines the location of the full-compression hard stop.

Dependencies

To enable this parameter, set **Hard stop at full compression** to On.

Stiffness at full compression — Full-compression hard stop stiffness
1e8 N/cm (default) | positive scalar

Stiffness to apply as the spring reaches full compression.

Dependencies

To enable this parameter, set **Hard stop at full compression** to On.

Damping at full compression — Full-compression hard stop damping
1e8 s*N/cm (default) | positive scalar

Damping to apply as the spring reaches full compression.

Dependencies

To enable this parameter, set **Hard stop at full compression** to On.

Hard stop model — Hard stop transition behavior
Stiffness and damping applied smoothly through transition region, damped rebound (default) | ...

Stiffness and rebound options for the hard stop model. You can choose from the following options:

- Stiffness and damping applied smoothly through transition region, damped rebound
- Full stiffness and damping applied at bounds, undamped rebound
- Full stiffness and damping applied at bounds, damped rebound

Dependencies

To enable this parameter, set **Hard stop at full extension** or **Hard stop at full compression** to On.

Transition region — Region of partial hard stop effects
0.1 mm (default) | positive scalar

Distance from full compression or full extension where the effects of stiffness and damping are partially applied. When you set **Hard stop model** to Stiffness and damping applied smoothly through transition region, damped rebound, the block will smoothly transition the onset of stiffness and damping as the spring approaches full extension or full compression.

Dependencies

To enable this parameter, set **Hard stop at full extension** or **Hard stop at full compression** to On, and set **Hard stop model** to Stiffness and damping applied smoothly through transition region, damped rebound.

See Also
Nonlinear Translational Spring | Nonlinear Translational Damper | Shock Absorber | Translational Hard Stop

Topics
tablelookup

Introduced in R2021a

Band Brake

Frictional brake with flexible band wrapped around rotating drum with triggered faults
Library: Simscape / Driveline / Brakes & Detents / Rotational
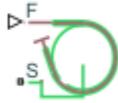
Description

The Band Brake block represents a frictional brake with a flexible band that wraps around the periphery of a rotating drum to produce a braking action. A positive actuating force causes the band to tighten around the rotating drum and it places the friction surfaces in contact. Viscous and contact friction between the surfaces of the drum and the flexible band causes the rotating drum to decelerate.

You can model the effects of heat flow and temperature change for the block by using port **H**, an optional thermal conserving port.

You can also enable faulting. When faulting occurs, the belt will exert a user-specified force. Faults can occur at a specified time or due to an external trigger at port **T**.

Band brakes provide high braking torque at the cost of reduced braking precision in applications that include winch drums, chainsaws, go-karts, and mini-bikes.

Equations

The model employs a simple parameterization with readily accessible brake geometry and friction parameters.
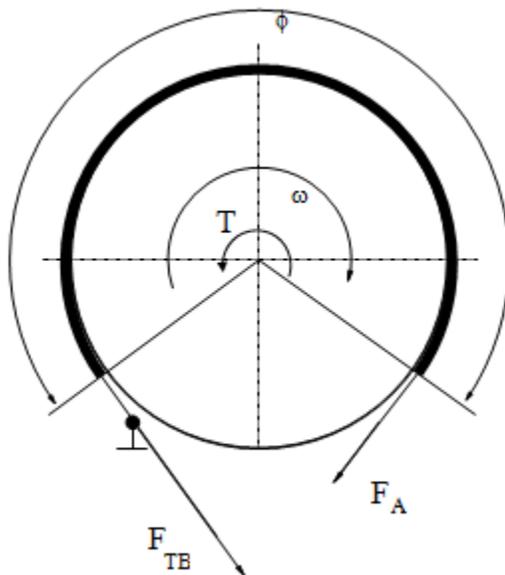
The braking torque as a function of the external brake actuation force that tightens the belt is

$$T = (F_{TB} - F_A) \cdot r_D + \mu_{visc} \cdot \omega$$

Where:

- T is the braking torque.
- F_{TB} is the force acting on the tense branch of the band.
- F_A is the external brake actuation force.
- r_D is the drum radius.
- μ_{visc} is the viscous friction coefficient.
- μ is the contact friction coefficient.
- ϕ is the wrap angle.

Forces F_{TB} and F_A satisfy the relationship

$$\frac{F_{TB}}{F_A} = e^{\mu\phi}$$

Replacing the relationship in the braking torque formula eliminates force F_{TB} such that

$$T = F_A\big(e^{\mu\phi} - 1\big) \cdot r_D + \mu_{visc} \cdot \omega$$

To avoid discontinuity at zero relative velocity, the model defines the actuation force, F_S, as a hyperbolic function

$$F_A = F_{in} \cdot \tanh\left(\frac{4\omega}{\omega_{threshold}}\right)$$

Where:

- F_{in} is the force input signal.
- $\omega_{threshold}$ is the angular velocity threshold.

Faulty Behavior

When faults are enabled, a belt force is applied in response to one or both of these triggers:

- Simulation time — Faulting occurs at a specified time.
- Simulation behavior — Faulting occurs in response to an external trigger. This exposes port **T**.

If a fault trigger occurs, the input force is replaced by the **Belt force when faulted** value for the remainder of the simulation. A value of 0 implies that no braking will occur. A relatively large value implies that the brake is stuck.

You can set the block to issue a fault report as a warning or error message in the Simulink Diagnostic Viewer with the **Reporting when fault occurs** parameter.

Thermal Model

You can model the effects of heat flow and temperature change by exposing the optional thermal port. To expose the port, in the **Friction** settings, set the **Thermal Port** parameter to Model. Exposing the thermal port also exposes these related settings:

- **Friction > Temperature**
- **Friction > Contact friction coefficient vector**
- **Thermal Port > Thermal mass**
- **Variables > Temperature**

Variables

Use the **Variables** settings to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Dependencies

Variable settings are visible only when, in the **Friction** settings, the **Thermal port** parameter is set to `Model`.

Limitations and Assumptions

The model does not account for actuator flow consumption.

Ports

Input

F — Belt tensioning force
physical signal

Physical signal inport associated with the external tensioning force that is applied to the belt.

T — External fault trigger
physical signal

Physical signal port for an external fault trigger. Triggering occurs when the value is greater than 0.5. There is no unit associated with the trigger value.

Dependencies

This port is visible when **Enable faults** is set to `On` and **Enable external fault trigger** is set to `On`.

Conserving

S — Drum shaft rotation
rotational mechanical

Rotational conserving port associated with the rotating drum shaft.

H — Heat flow, in W
thermal

Thermal conserving port associated with heat flow.

Dependencies

This port is visible only when, in the **Friction** settings, the **Thermal Port** parameter is set to `Model`.

Exposing this port makes related settings visible.

Parameters

Geometry

`Drum radius` — **Contact surface radius**
`150 mm` (default) | positive scalar

Radius of the drum contact surface. The parameter must be greater than zero.

`Wrap angle` — **Contact angle**
`270 deg` (default) | positive scalar

Contact angle between the flexible belt and the rotating drum. The parameter must be greater than zero.

Friction

`Viscous friction coefficient` — **Viscous friction**
`.01 n*m/(rad/s)` (default) | nonegative scalar

Viscous friction coefficient at the belt-drum contact surface. The parameter must be greater than or equal to zero.

`Thermal port` — **Thermal model**
`Omit` (default) | `Model`

Model for heat flow and temperature change:

- `Omit` — Neglect thermal dynamics.
- `Model` — Include thermal dynamics.

Dependencies

When this parameter is set to `Model`, the thermal port and related parameters and variables are visible.

`Temperature` — **Temperature**
`[280, 300, 320] K` (default) | increasing vector

Array of temperatures used to construct a 1-D temperature-efficiency lookup table. The array values must increase left to right.

Dependencies

This parameter is only visible when the **Thermal Port** parameter is set to `Model`.

`Contact friction coefficient` — **Coulomb friction**
`0.3` (default) | positive scalar

Coulomb friction coefficient at the belt-drum contact surface. The value must be greater than zero.

Dependencies

This parameter is only visible when the **Thermal Port** parameter is set to `Omit`.

`Contact friction coefficient vector` — **Coulomb friction**
`[.1, .05, .03]` (default) | increasing vector of positive values

Coulomb friction coefficient at the belt-drum contact surface, such that:

- The number of elements in the vector must be the same as the number of elements in the specified vector for the **Temperature** parameter
- The values increase left to right.
- Each value must be greater than zero.

Dependencies

This parameter is only visible when the **Thermal Port** parameter is set to `Model`.

Angular velocity threshold — Rotational velocity required for near steady-state contact friction
`0.01 rad/s` (default) | positive scalar

Angular velocity at which the contact friction coefficient practically reaches its steady-state value. The value must be greater than zero.

Faults

Enable faults — Fault option
`Off` (default) | `On`

Enable externally or temporally triggered faults. When faulting occurs, the brake belt force normally received at port **F** will be set to the value specified in the **Belt force when faulted** parameter.

Belt force when faulted — Set faulted belt force
`0 N` (default) | positive scalar

Set faulted belt force. When faulting occurs, the brake belt force normally received at port **F** will be set to the value specified in the **Belt force when faulted** parameter. A value of `0` implies that braking does not occur. A relatively large value implies that the brake is stuck.

Dependencies

To enable this parameter, set **Enable faults** to `On`.

Enable external fault trigger — External trigger option
`Off` (default) | `On`

Enables port **T**. A physical signal at port **T** that is greater than `0.5` triggers faulting.

Dependencies

To enable this parameter, set **Enable faults** to `On`.

Enable temporal fault trigger — Temporal trigger option
`Off` (default) | `On`

Enables fault triggering at a specified time. When the **Simulation time for fault event** is reached, the brake belt force normally received at port **F** will be set to the value specified in the **Belt force when faulted** parameter.

Dependencies

To enable this parameter, set **Enable faults** to `On`.

Simulation time for fault event — Time at which faulting is triggered
5 s (default) | positive scalar

When the **Simulation time for fault event** is reached, the brake belt force normally received at port **F** will be set to the value specified in the **Belt force when faulted** parameter.

Dependencies

To enable this parameter, set **Enable faults** to On and **Enable temporal fault trigger** to On.

Reporting when fault occurs — Fault condition report
None (default) | Warning | Error

Reporting preference for the fault condition. When reporting is set to Warning or Error, a message is displayed in the Simulink Diagnostic Viewer. When Error is selected, the simulation will stop if faulting occurs.

Dependencies

To enable this parameter, set **Enable faults** to On.

Thermal Port

Thermal mass — Resistance to temperature change
50 kJ/K (default) | scalar

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

This parameter is only visible when, in the **Friction** settings, the **Thermal Port** parameter is set to Model.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Double-Shoe Brake | Loaded-Contact Rotational Friction | Rotational Detent

Topics
"Model Thermal Losses in Driveline Components"

Introduced in R2012b

Belt Drive

Power transmission system with taut belt connecting two pulleys
Library: Simscape / Driveline / Couplings & Drives
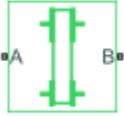
Description

The Belt Drive block represents a pair of pulleys connected with a flexible ideal, flat, or V-shaped belt. When you set **Belt type** to `Ideal - No slip`, the belt does not slip relative to the pulley surfaces.

The block accounts for friction between the flexible belt and the pulley periphery. If the friction force is not sufficient to drive the load, the block allows slip. The relationship between the tensions in the driving and driven branches conforms to the capstan equation, also known as the Euler-Eytelwein equation. The block accounts for centrifugal loading in the flexible belt, pulley inertia, and bearing friction.

The Belt Drive block is a structural component based on the Simscape Driveline Belt Pulley block and Simscape Translational Spring and Translational Damper blocks.

The Translational Spring and Translational Damper blocks simulate the compliance of the belt. For the equations governing the contact dynamics between the belt and the pulley, see the Belt Pulley block. The figure shows the functional block diagram for the Belt Drive block.
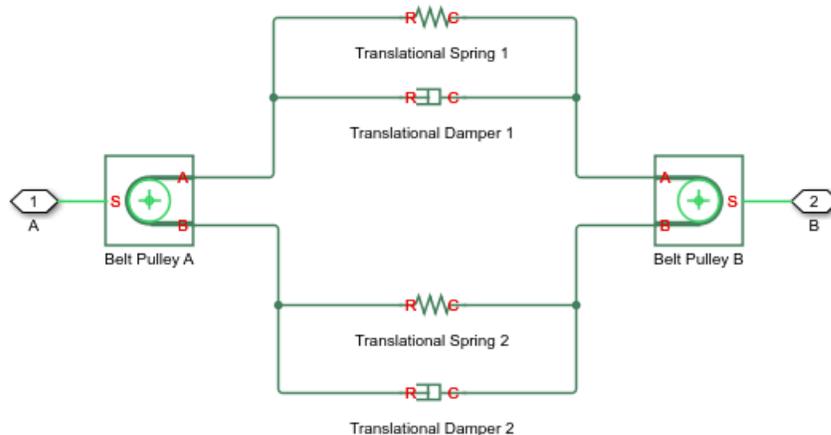
To learn more about the construction of the block, see "About Composite Components" and the reference pages for the individual component blocks.

Equations

The diagrams show the open and crossed belt drive configurations. When you set **Drive type** to `Open belt`, both pulleys tend to rotate in the same direction and the larger pulley has a larger belt wrap

angle. When you set **Drive type** to `Crossed belt`, the pulleys tend to rotate in opposite directions and have the same wrap angle.

Open Belt Configuration   Crossed Belt Configuration
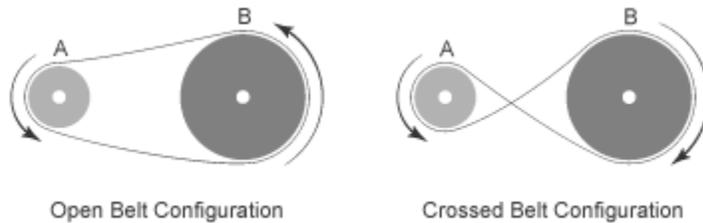
Belt Drive Diagrams

The figures and equations refer to these quantities:

- θ_A is the wrap angle of pulley A.
- θ_B is the wrap angle of pulley B.
- R_A is the effective radius of pulley A.
- R_B is the effective radius of pulley B.
- C is the distance between the centers of pulleys A and B.

Open Belt

When you set **Drive type** to `Open belt`, the block calculates the wrap angle of the belt around each pulley as:

$$\theta_A = \pi + 2 * \sin^{-1}\frac{R_A - R_B}{C},$$

$$\theta_B = \pi - 2 * \sin^{-1}\frac{R_A - R_B}{C}$$

The diagram shows the wrap angles and parameters.
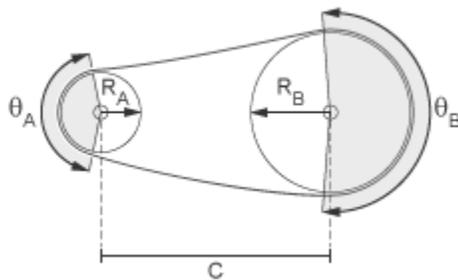
Crossed Belt

When you set **Drive type** to `Crossed belt`, the two wrap angles are equal and the wrap angle of the belt around each pulley:

$$\theta_A = \theta_B = \pi + 2 * \sin^{-1}\frac{R_A + R_B}{C}.$$

The diagram shows the wrap angles and parameters.
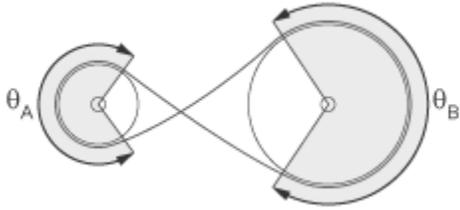
Assumptions and Limitations

- The pulleys do not translate.
- The friction coefficient and friction velocity threshold between the belt and each of the pulleys is the same. To parameterize separate friction interactions, use two Belt Pulley blocks.

Ports

Conserving

A — Pulley A shaft
mechanical rotational

Mechanical rotational conserving port associated with the shaft of pulley A.

B — Pulley B shaft
mechanical rotational

Mechanical rotational conserving port associated with the shaft of pulley B.

Parameters

Belt

Drive type — Belt drive configuration
Open belt (default) | Crossed belt

Belt drive configuration. For more information, see "Belt Drive Diagrams" on page 1-45.

Belt type — Belt parameterization
Ideal - No slip (default) | Flat belt | V-belt

Belt type selection. The belt type affects slip conditions

- Ideal - No slip — Parameterize an ideal belt, which does not slip relative to the pulley.
- Flat belt — Parameterize a belt with a rectangular cross-section.
- V-belt— Parameterize a belt with a V-shaped cross-section.

Compliance — Compliance option
No compliance - Suitable for HIL simulation (default) | Specify stiffness and damping

Option to simulate compliance.

- `No compliance - Suitable for HIL simulation` — Simulates a noncompliant belt that does not yield elastically when subjected to a force. To prioritize performance, select this option.
- `Specify stiffness and damping` — Simulates a compliant belt that does yield elastically when subjected to a force. To prioritize fidelity, select this option.

Dependencies

To enable this parameter, set **Belt type** to `Ideal - No slip`.

V-belt sheave angle — Sheave angle
30 deg (default) | nonnegative scalar

Sheave angle of the V-belt.

Dependencies

To enable this parameter, set **Belt type** to `V-belt`.

Number of V-belts — Number of belts
1 (default) | positive scalar integer

Number of V-belts.

The block rounds noninteger values to the nearest integer. Increasing the number of belts increases the friction force, effective mass per unit length, and maximum allowable tension.

Dependencies

To enable this parameter, set **Belt type** to `V-belt`.

Centrifugal force — Centrifugal force option
`Do not model centrifugal force - Suitable for HIL simulation` (default) | `Model centrifugal force`

Option to include the effects of centrifugal force. If you set this parameter to `Model centrifugal force`, centrifugal force saturates to approximately 90% of the value of the force on each belt end.

Dependencies

To enable this parameter, set **Belt type** to `Flat belt` or `V-belt`.

Belt mass per unit length — Mass per unit length
0.6 kg/m (default) | positive scalar

Centrifugal force contribution in terms of linear density.

Dependencies

To enable this parameter, set **Belt type** to `Flat belt` or `V-belt` and **Centrifugal force** to `Model centrifugal force`.

Longitudinal stiffness — Belt effective stiffness
1e4 N/m (default) | nonnegative scalar

Effective stiffness of the belt.

Dependencies

To enable this parameter, set **Belt type** to `Ideal - No slip` and **Compliance** to `Specify stiffness and damping` or set **Belt type** to `Flat belt` or `V-belt`.

Longitudinal damping — Belt effective damping
`1e3 N/(m/s)` (default) | nonnegative scalar

Effective damping of the belt.

To enable this parameter, either set

- **Belt type** to `Ideal - No slip` and **Compliance** to `Specify stiffness and damping`, or
- **Belt type** to `Flat belt` or `V-belt`

Pre-tension — Resting belt tension
`100 N` (default) | positive scalar

Tension in the belt when the belt and pulleys are at rest. The value must be positive.

To enable this parameter, either set

- **Belt type** to `Ideal - No slip` and **Compliance** to `Specify stiffness and damping`, or
- **Belt type** to `Flat belt` or `V-belt`

Maximum tension — Tension threshold option
`No maximum tension` (default) | `Specify maximum tension`

Option to specify a maximum tension. If you set this parameter to `Specify maximum tension` and the belt tension on either end of the belt meets or exceeds the value that you specify for the **Belt maximum tension** parameter, the simulation stops and generates an assertion error.

Dependencies

To enable this parameter, either set

- **Belt type** to `Ideal - No slip` and **Compliance** to `Specify stiffness and damping`, or
- **Belt type** to `Flat belt` or `V-belt`

Belt maximum tension — Maximal tension threshold
`1e5 N` (default) | positive scalar

Maximum allowable tension for each belt. When the tension on either end of the belt meets or exceeds this value, the simulation stops and generates an assertion error.

To enable this parameter, either set

- **Belt type** to `Ideal - No slip`, **Compliance** to `Specify stiffness and damping`, and **Maximum tension** to `Specify maximum tension`, or
- **Belt type** to `Flat belt` or `V-belt` and **Maximum tension** to `Specify maximum tension`

Pulley A

Pulley radius — Pulley radius
`.15 m` (default) | positive scalar

Radius of the pulley.

Bearing viscous friction coefficient — Bearing viscous friction
0 N*m/(rad/s) (default) | scalar

Viscous friction associated with the bearings that hold the axis of the pulley.

Inertia — Rotational inertia
No inertia (default) | Specify inertia and initial velocity

Option to parameterize rotational inertia with an initial velocity.

Pulley inertia — Pulley inertia
0.01 kg*m^2 (default) | positive scalar

Rotational inertia of the pulley.

Dependencies

To enable this parameter, set **Inertia** to Specify inertia and initial velocity.

Pulley initial velocity — Initial pulley rotational velocity
0 rad/s (default) | scalar

Initial rotational velocity of the pulley.

Dependencies

To enable this parameter, set **Inertia** to Specify inertia and initial velocity.

Pulley B

Pulley radius — Pulley radius
.30 m (default) | positive scalar

Radius of the pulley.

Bearing viscous friction coefficient — Bearing viscous friction
0.001 N*m/(rad/s) (default) | scalar

Viscous friction associated with the bearings that hold the axis of the pulley.

Inertia — Rotational inertia
No inertia (default) | Specify inertia and initial velocity

Option to parameterize rotational inertia with an initial velocity.

Pulley inertia — Pulley inertia
0.03 kg*m^2 (default) | positive scalar

Rotational inertia of the pulley.

Dependencies

To enable this parameter, set **Inertia** to Specify inertia and initial velocity.

Pulley initial velocity — Initial pulley rotational velocity
0 rad/s (default) | scalar

Initial rotational velocity of the pulley.

Dependencies

To enable this parameter, set **Inertia** to `Specify inertia and initial velocity`.

Contact

To enable these settings, set **Belt type** to `Flat belt` or `V-belt`.

Friction model — Friction method selection
Modal (default) | Continuous

Option to parameterize continuous or modal friction.

Pulley A initial state — Initial state of pulley A
Unlocked (default) | Locked

Option to initialize the simulation with pulley A locked or unlocked.

Dependencies

To enable this parameter, set **Friction model** to `Modal`.

Pulley B initial state — Initial state of pulley B
Unlocked (default) | Locked

Option to initialize the simulation with pulley B locked or unlocked.

Dependencies

To enable this parameter, set **Friction model** to `Modal`.

Static friction coefficient — Static friction coefficient
0.5 (default) | positive scalar

Belt friction while maintaining static contact.

Dependencies

To enable this parameter, set **Friction model** to `Modal`.

Kinetic friction coefficient — Kinetic friction coefficient
0.42 (default) | positive scalar

Belt friction while slipping.

Dependencies

To enable this parameter, set **Friction model** to `Modal`.

Contact friction coefficient — Contact friction coefficient
0.5 (default) | positive scalar

Coulomb friction coefficient between the belt and the pulley surface.

Dependencies

To enable this parameter, set **Friction model** to `Continuous`.

Wrap angle calculation — Angle calculation type
Specify pulley center separation (default)|Specify wrap angles

Option to calculate the wrap angle by the pulley center separation or to specify those values directly.

- Specify pulley center separation — The block calculates the wrap angle of the belt on the pulleys by using the pulley radii, center separation, and drive type.
- Specify wrap angles — The block uses the wrap angles that you specify.

Pulley center separation — Pulley center separation distance
0.5 m (default) | positive scalar

Distance between the centers of the pulleys.

Dependencies

To enable this parameter, set **Wrap angle calculation** to Specify pulley center separation.

Pulley A wrap angle — Pulley wrap angle
145 deg (default) | scalar

Angle of contact between the belt and pulley attached to port **A**.

Dependencies

To enable this parameter, set **Wrap angle calculation** to Specify wrap angles.

Pulley B wrap angle — Pulley wrap angle
215 deg (default) | scalar

Angle of contact between the belt and pulley attached to port **B**.

Dependencies

To enable this parameter, set **Wrap angle calculation** to Specify wrap angles.

Velocity threshold — Contact threshold
0.001 m/s (default) | positive scalar

Relative velocity required for peak kinetic friction in the contact. The friction velocity threshold improves the numerical stability of the simulation by ensuring that the force is continuous when the direction of the velocity changes.

Force threshold — Force threshold
0.01 N (default) | positive scalar

Relative force required for peak kinetic friction in the contact.

Dependencies

To enable this parameter, set **Friction model** to Continuous.

More About

Hardware-in-the-Loop Simulation

For optimal simulation performance, set the **Belt** > **Centrifugal force** parameter to `Do not model centrifugal force - Suitable for HIL simulation`.

References

[1] Johnson, Kenneth L. *Contact Mechanics*. Cambridge: Cambridge Univ. Press, 2003.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Belt Pulley | Chain Drive

Topics
"Best Practices for Modeling Pulley Networks"
"Rotational-Translational Couplings"
"About Composite Components"

Introduced in R2012a

Belt Pulley

Power transmission element with frictional belt wrapped around pulley circumference
Library: Simscape / Driveline / Couplings & Drives
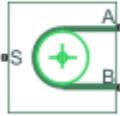
Description

The Belt Pulley block represents a pulley wrapped in a flexible ideal, flat, or V-shaped belt. When you set **Belt type** to `Ideal - No slip`, the belt does not slip relative to the pulley surfaces. You can optionally enable pulley linear translation.

The block accounts for friction between the flexible belt and the pulley surface. The tensioner slips when the load exceeds the friction force. The block accounts for centrifugal loading in the flexible belt, pulley inertia, and bearing friction.
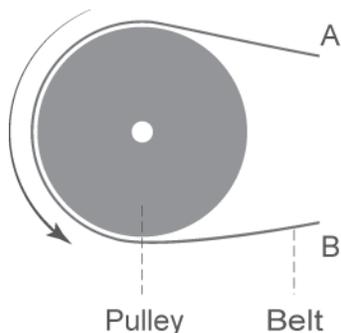
The belt ends can either move in the same direction or in the opposite direction.

The block equations relate power transmission between the belt branches or to or from the pulley. The driving and driven branches use the same calculation. Without sufficient tension, the frictional force is not enough to transmit power between the pulley and belt.

Your model is valid when both ends of the belt are in tension. You can choose to display a warning in the Simulink® Diagnostic Viewer when the leading belt end loses tension. When assembling a model, ensure that tension is maintained throughout the simulation by adding mass to at least one of the belt ends or by adding a tensioner like the Rope block. Use the Variable Viewer to ensure that any springs attached the belt are in tension. For more details on building a tensioner, see "Best Practices for Modeling Pulley Networks".

You can set **Friction model** to `Modal` to use a modal parameterization for the pulley. Choose the modal parameterization for greater numerical robustness. To enable this parameter, set **Pulley translation** to `Off`.

The Belt Pulley block uses a composite implementation of the Fundamental Friction Clutch block to produce the conditions for the modal parameterization.

Equations

For either setting of the **Friction model** parameter, the block equations refer to these quantities:

- β is the belt direction sign. When you set **Belt direction** to Ends move in same direction, β = 1. Otherwise, β = -1.
- V_{rel} is the relative velocity between the belt and pulley periphery. V_{rel} = 0 when **Belt type** is Ideal - No slip.
- V_A is the linear velocity of branch A.
- V_B is the linear velocity of branch B.
- V_C is the linear velocity of the pulley at its center. If **Pulley translation** is Off, the block constrains this to 0.
- ω_S is the angular velocity of the pulley contact surface.
- R is the radius of the pulley.
- $F_{centrifugal}$ is centrifugal force of the belt.
- F_C is the force acting through the pulley centroid. When you specify a value for **Inertia**, F_C includes force due to the pulley mass acceleration.
- ρ is the belt linear density.
- F_{fr} is the friction force between the pulley and the belt.
- F_A is the force acting along branch A.
- F_B is the force acting along branch B.
- θ is the contact wrap angle.
- τ_S is the pulley torque.

The kinematic constraints between the pulley and belt are:

$$-\beta V_A = V_C - R\omega_S - \beta V_{rel}$$

$$V_B = V_C + R\omega_S + \beta V_{rel}$$

When you set **Belt type** to either V-belt or Flat belt and set **Centrifugal force** to Model centrifugal force, the centrifugal force is:

$$F_{centrifugal} = \rho(V_B - V_C)^2.$$

When **Pulley translation** is Off, the force balancing equation is:

$$0 = F_C + (-\beta F_A + F_B) \cdot \sin\left(\frac{\theta}{2}\right).$$
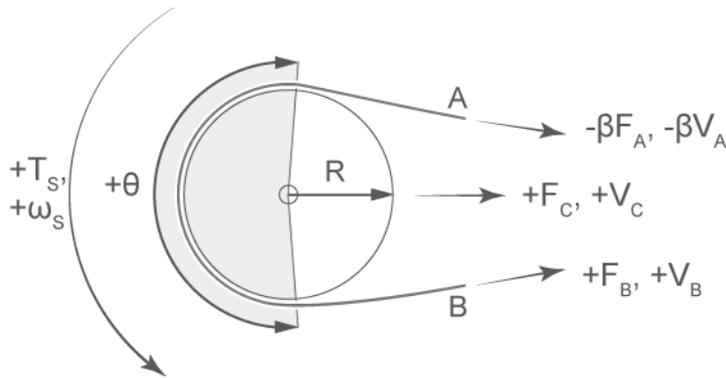
The sign convention is such that, when **Belt direction** is Ends move in opposite direction, a positive rotation in port **S** results in a negative translation for port **A** and a positive translation for port **B**.

To enable the **Friction model** parameter, set **Belt type** to Flat belt or V-belt and **Pulley translation** to Off.

Continuous Friction

When you set **Friction model** to Continuous, the block equations refer to these quantities:

- μ is the **Contact friction coefficient** parameter.
- $\mu_{smoothed}$ is the instantaneous value of the friction coefficient.
- V_{thr} is the **Velocity threshold** parameter.
- b is the viscous damping of the bearing.
- F_{thr} is the **Force threshold** parameter.

The instantaneous friction coefficient is a function of the relative velocity such that

$$\mu_{smoothed} = \mu_{sheave}\tanh\left(4\frac{V_{rel}}{V_{thr}}\right),$$

where the hyperbolic tangent function maintains numerical robustness by ensuring a smooth and continuous output for V_{rel} zero-crossings.

For a V-belt, the block derives the contact friction value using the sheave angle:

$$\mu_{sheave} = \frac{\mu}{\sin\left(\frac{\phi}{2}\right)},$$

where:

- μ_{sheave} is the effective friction coefficient.
- Φ is the sheave angle.

For a flat belt, $\mu_{sheave} = \mu$.

The friction velocity threshold controls the width of the region within which the friction coefficient changes its value from zero to a steady-state maximum. The friction velocity threshold specifies the

velocity at which the hyperbolic tangent equals `0.999`. The smaller the value, the steeper the change of μ.

The block determines the effect of friction on the force at the belt ends as:

$$-\beta F_A - F_{centrifugal} = \left(F_B - F_{centrifugal}\right)e^{-\mu_{smoothed}\theta},$$

which follows the form of the capstan equation, also known as the Euler-Eytelwein equation. The torque acting on the pulley is

$$\tau_S = (-\beta F_A - F_B)R\sigma + \omega_S b,$$

where $\sigma = 1$ when you set **Belt type** to `Ideal - No slip`. Otherwise,

$$\sigma = \tanh\left(4\frac{V_{rel}}{V_{thr}}\right)\tanh\left(\frac{F_B}{F_{thr}}\right).$$

Modal Friction

For a V-belt, the block derives the static and kinetic friction values by using the sheave angle:

$$\mu_{Static,\,sheave} = \frac{\mu_{Static}}{\sin\left(\frac{\phi}{2}\right)}$$

$$\mu_{Kinetic,\,sheave} = \frac{\mu_{Kinetic}}{\sin\left(\frac{\phi}{2}\right)}$$

where:

- μ_{Static} is the **Static friction coefficient** parameter.
- $\mu_{Static,sheave}$ is effective static friction coefficient.
- $\mu_{Kinetic}$ is the **Kinetic friction coefficient** parameter.
- $\mu_{Kinetic,sheave}$ is effective kinetic friction coefficient.
- Φ is the sheave angle.

For a flat belt, $\mu_{Static,sheave} = \mu_{Static}$, and $\mu_{Kinetic,sheave} = \mu_{Kinetic}$.

When you set **Friction model** to `Modal`, the block calculates the maximum static friction force before slipping as

$$F_{Friction,\,Static,\,Max} = \left(F_{Drive} - F_{Centrifugal}\right) \cdot \left(1 - e^{-\mu_{Static,\,sheave}\theta}\right),$$

where $F_{Friction,Static,Max}$ is the maximum force magnitude due to static friction, and F_{Drive} is the force due to tension felt at the end of the pulley with the greatest force magnitude. During slip,

$$F_{Friction,\,Slip} = \frac{\mu_{Kinetic,\,sheave}}{\mu_{Static,\,sheave}}F_{Friction,\,Static,\,Max},$$

where $\mu_{Kinetic,sheave}$ is the kinetic friction coefficient. The block resolves the torque balance using

$$\tau_S = \omega_S b + RF_{Friction},$$

and

$$R\beta F_A + RF_B = -RF_{Friction},$$

where Simscape logs $RF_{Friction}$ as `fundamental_clutch.torque`.

Assumptions and Limitations

- The block assumes noncompliance along the length of the belt.
- The block assumes both belt ends maintain adequate tension throughout the simulation.
- The block treats the translation of the pulley center as planar where the pulley travels along the bisect of the pulley wrap angle. The center velocity V_C and force F_C only account for the component along this line of motion.
- The Eytelwein equation for belt friction neglects the effect of pulley translation on friction.

Ports

Conserving

S — Pulley shaft angular velocity
mechanical rotational

Mechanical rotational conserving port associated with the angular velocity of the pulley shaft.

A — Belt end A linear velocity
mechanical translational

Mechanical translational conserving port associated with the linear velocity of belt end A.

B — Belt end B linear velocity
mechanical translational

Mechanical translational conserving port associated with the linear velocity of belt end B.

C — Pulley center linear velocity
mechanical translational

Mechanical translational conserving port associated with pulley translation. The pulley moves in the plane and along the bisect of the pulley wrap angle. When the relative velocity is positive, the pulley center moves.

Dependencies

To enable this port, set **Pulley translation** to On.

Parameters

Belt

Belt type — Belt parameterization
Ideal - No slip (default) | Flat belt | V-belt

Belt type selection. The belt type affects slip conditions.

- `Ideal - No slip` — Parameterize an ideal belt that does not slip relative to the pulley.
- `Flat belt` — Parameterize a belt with a rectangular cross-section.
- `V-belt`— Parameterize a belt with a V-shaped cross-section.

V-belt sheave angle — Sheave angle
30 deg (default) | positive scalar

Sheave angle of the V-belt.

Dependencies

To enable this parameter, set **Belt type** to `V-belt`.

Number of V-belts — Number of belts
1 (default) | positive scalar integer

Number of V-belts.

The block rounds noninteger values to the nearest integer. Increasing the number of belts increases the effective mass per unit length and maximum allowable tension.

Dependencies

To enable this parameter, set **Belt type** to `V-belt`.

Centrifugal force — Centrifugal force option
Do not model centrifugal force - Suitable for HIL simulation (default) | Model centrifugal force

Option to include the effects of centrifugal force. If you set this parameter to `Model centrifugal force`, centrifugal force saturates to approximately 90% of the value of the force on each belt end.

Dependencies

To enable this parameter, set **Belt type** to `Flat belt` or `V-belt`.

Belt mass per unit length — Mass per unit length
0.6 kg/m (default) | positive scalar

Centrifugal force contribution in terms of linear density expressed as mass per unit length.

Dependencies

To enable this parameter, set **Belt type** to `Flat belt` or `V-belt` and **Centrifugal force** to `Model centrifugal force`.

Belt direction — Initial relative motion direction of the belt end
Ends move in opposite direction (default) | Ends move in same direction

Relative direction of translational motion of one belt end with respect to the other.

Maximum tension — Tension threshold option
No maximum tension (default) | Specify maximum tension

Option to specify a maximum tension. If you select `Specify maximum tension` and the belt tension on either end of the belt meets or exceeds the value that you specify for the **Belt maximum tension** parameter, the simulation stops and generates an assertion error.

Belt maximum tension — Maximal tension threshold
`1e5 N` (default) | positive scalar

Maximum allowable tension for each belt. When the tension on either end of the belt meets or exceeds this value, the simulation stops and generates an assertion error.

Dependencies

To enable this parameter, set **Maximum tension** to `Specify maximum tension`.

Tension warning — Slack threshold reporting
`Do not check tension` (default) | `Warn if leading end loses tension`

Whether the block generates a warning when the tension at either end of the belt falls below zero.

Note When combining Belt Pulley blocks with Rope blocks to form loops, set the Belt Pulley block **Tension warning** parameter to `Do not check tension` and set the Rope block **Tension warning** parameter to `Warn if rope loses tension`.

Pulley

Pulley translation — Option to allow linear motion
`Off` (default) | `On`

Option to allow the pulley to translate. Setting this parameter to `On` enables port **C**.

Pulley radius — Pulley radius
`.15 m` (default) | positive scalar

Radius of the pulley.

Bearing viscous friction coefficient — Bearing viscous friction
`0.001 N*m/(rad/s)` (default) | scalar

Viscous friction associated with the bearings that hold the axis of the pulley.

Inertia — Rotational inertia
`No inertia` (default) | `Specify inertia and initial velocity`

Option to parameterize the rotational inertia with an initial velocity.

Pulley inertia — Pulley inertia
`0.01 kg*m^2` (default) | positive scalar

Rotational inertia of the pulley.

Dependencies

To enable this parameter, set **Inertia** to `Specify inertia and initial velocity`.

Pulley initial rotational velocity — Initial pulley rotational velocity
`0 rad/s` (default) | scalar

Initial rotational velocity of the pulley.

Dependencies

To enable this parameter, set **Inertia** to `Specify inertia and initial velocity`.

Pulley mass — Pulley mass
`0.01 kg` (default) | positive scalar

Pulley mass for the inertia calculation.

Dependencies

To enable this parameter, set

- **Pulley translation** to `On`
- **Inertia** to `Specify inertia and initial velocity`

Pulley initial translational velocity — Initial pulley rotational velocity
`0 m/s` (default) | scalar

Initial translational velocity of the pulley.

Dependencies

To enable this parameter, set

- **Pulley translation** to `On`
- **Inertia** to `Specify inertia and initial velocity`

Contact

To enable these settings, set **Belt type** to `Flat belt` or `V-belt`.

Friction model — Friction method selection
`Modal` (default) | `Continuous`

Option to parameterize continuous or modal friction.

Dependencies

To enable this parameter, set **Pulley translation** to `Off`.

Initial state — Initial state of pulley
`Unlocked` (default) | `Locked`

Option to initialize the simulation with the pulley locked or unlocked.

Dependencies

To enable this parameter, set

- **Pulley translation** to `Off`
- **Friction model** to `Modal`

Static friction coefficient — Static friction coefficient
`0.5` (default) | positive scalar

Belt friction while maintaining static contact.

Dependencies

To enable this parameter, set **Friction model** to `Modal`.

Kinetic friction coefficient — Kinetic friction coefficient
`0.42` (default) | positive scalar

Belt friction while slipping.

Dependencies

To enable this parameter, set **Friction model** to `Modal`.

Contact friction coefficient — Contact friction coefficient
`0.5` (default) | positive scalar

Coulomb friction coefficient between the belt and the pulley surface.

Dependencies

To enable this parameter, set **Friction model** to `Continuous`.

Wrap angle — Belt-to-pulley contact angle
`180 deg` (default) | positive scalar

Radial contact angle between the belt and the pulley.

Velocity threshold — Velocity threshold
`0.001 m/s` (default) | positive scalar

Relative velocity required for peak kinetic friction in the contact. The friction velocity threshold improves the numerical stability of the simulation by ensuring that the force is continuous when the direction of the velocity changes.

Force threshold — Force threshold
`0.01 N` (default) | positive scalar

Relative force required for peak kinetic friction in the contact.

Dependencies

To enable this parameter, set **Friction model** to `Continuous`.

More About

Hardware-in-the-Loop Simulation

For optimal simulation performance, set the **Belt > Centrifugal force** parameter to `Do not model centrifugal force - Suitable for HIL simulation`.

References

[1] Johnson, Kenneth L. *Contact Mechanics*. Cambridge: Cambridge Univ. Press, 2003.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Rope | Rope Drum | Belt Drive | Wheel and Axle

Topics
"Best Practices for Modeling Pulley Networks"
"Rotational-Translational Couplings"
"Mode Chart Modeling"

Introduced in R2012a

Chain Drive

Power transmission system with chain and two sprockets
Library: Simscape / Driveline / Couplings & Drives
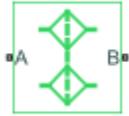
Description

The Chain Drive block represents a power transmission system with a chain and two sprockets. The chain meshes with the sprockets, transmitting rotary motion between the two. Power transmission can occur in reverse, that is, from driven to driver sprocket, due to external loads. This condition is known as back-driving.
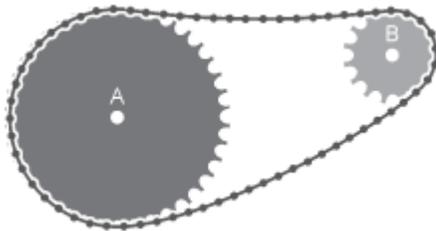
The drive chain is compliant. It can stretch under tension or slacken if loose. The compliance model consists of a linear spring-damper set in a parallel arrangement. The spring resists tensile strain in the chain. The damper resists tensile motion between chain elements.

The spring and damper forces act directly on the sprockets that the chain connects. The spring force is present when one chain branch is taut. The damper force is present continuously. To represent and report a failure condition, the simulation stops and generates an error if the net tensile force in the chain exceeds the specified maximum tension value.

The block accounts for viscous friction at the sprocket joint bearings. During motion, viscous friction causes power transmission losses, reducing chain-drive efficiency. These losses compound due to chain damping. To eliminate power transmission losses in the chain drive, in the **Dynamic** settings, set the parameters for viscous friction and chain damping to zero.

Equations

The tensile strain rate in the chain is the difference between the sprocket tangential velocities, which are each the product of the angular velocity and pitch radii. Mathematically,

$$\dot{x} = \omega_A R_A - \omega_B R_B,$$

where:

- x is the tensile strain.

- ω_A, ω_B are the sprocket angular velocities.
- R_A, R_B are the sprocket pitch radii.

The figure shows the relevant variables.
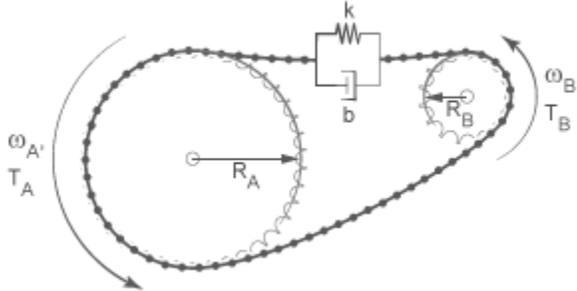
The chain tensile force is the net sum of the spring and damper forces. The spring force is the product of the tensile strain and the spring stiffness constant. This force is zero when the tensile strain is smaller than the chain slack. The damper force is the product of the tensile strain rate and the damping coefficient. Mathematically,

$$
F = \begin{cases}
-\left(x - \dfrac{S}{2}\right)k - \dot{x}b, & x > \dfrac{S}{2} \\[2mm]
-\dot{x}b, & \dfrac{S}{2} \geq x \geq -\dfrac{S}{2}, \\[2mm]
-\left(x + \dfrac{S}{2}\right)k - \dot{x}b, & x < -\dfrac{S}{2}
\end{cases}
$$

where:

- S is the chain slack.
- k is the spring stiffness constant.
- b is the damper coefficient.

The chain exerts a torque on each sprocket equal to the product of the tensile force and the sprocket pitch radius. The two torques act in opposite directions according to these equations:

$$\tau_A = -F \cdot R_A$$

$$\tau_B = -F \cdot R_B$$

Where:

- τ_A is the torque that the chain applies on sprocket A.
- τ_B is the torque that the chain applies on sprocket B.

In terms of velocity and consideration of friction, these equations apply:

$$\omega_A \cdot R_A = \omega_B \cdot R_B$$

$$(\tau_A - \mu_A \cdot \omega_A)R_B = -(\tau_B - \mu_B \cdot \omega_B)R_B$$

Where:

- ω_A is the rotational velocity for sprocket A.
- ω_B is the rotational velocity for sprocket B.
- μ_A is the coefficient of viscous friction for sprocket A.
- μ_B is the coefficient of viscous friction for sprocket B.

Variables

Use the **Variables** tab to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Assumptions and Limitations

- The sprocket tooth ratio equals the sprocket pitch radius ratio.
- Chain inertia is negligible.

Ports

Conserving

A — Sprocket A
mechanical rotational

Conserving rotational port associated with sprocket A.

B — Sprocket B
mechanical rotational

Conserving rotational port associated with sprocket B.

Parameters

Geometry

`Chain model` — Chain compliance and backlash model
`Ideal - no chain compliance or backlash` (default) | `Model chain compliance and backlash`

Compliance and backlash model for the block:

- `Ideal - no chain compliance or backlash` — Do not model chain stiffness, damping, or backlash.
- `Model chain compliance and backlash` — Model chain stiffness, damping, and backlash.

Dependencies

If this parameter is set `Model chain compliance and backlash`, related parameters and variables are enabled.

`Sprocket A pitch radius` — Sprocket *A* pitch radius
80 mm (default) | scalar

Radius of the pitch circle for sprocket A. The pitch circle is an imaginary circle passing through the contact point between a chain roller and a sprocket cog at full engagement.

Sprocket B pitch radius — Sprocket *B* pitch radius
40 mm (default) | scalar

Radius of the pitch circle for sprocket B. The pitch circle is an imaginary circle passing through the contact point between a chain roller and a sprocket cog at full engagement.

Chain slack length — Maximum slack length
50 mm (default) | scalar

Maximum distance the loose branch of the drive chain can move before it is taut. This distance equals the length difference between actual and fully taut drive chains.

If one sprocket is held in place while the top chain branch is taut, then the slack length is the tangential distance that the second sprocket must rotate before the lower chain branch becomes taut.

Dependencies

This parameter is enabled when **Chain model** is set to Model chain compliance and backlash.

Dynamics

Chain stiffness — Linear spring constant
1e5 N/m (default) | scalar

Linear spring constant in the chain compliance model. This constant describes the chain resistance to strain. The spring element accounts for elastic energy storage in the chain due to deformation.

Dependencies

This parameter is enabled when **Chain model** is set to Model chain compliance and backlash.

Chain damping — Linear damping coefficient
5 N/(m/s) (default) | scalar

Linear damping coefficient in the chain compliance model. This coefficient describes the resistance to tensile motion between adjacent elements in the chain. The damper element accounts for power losses in the chain due to deformation.

Dependencies

This parameter is enabled when **Chain model** is set to Model chain compliance and backlash.

Viscous friction coefficient of sprocket A — Sprocket *A* friction
0.001 N*m/(rad/s) (default)

Friction coefficient due to the rolling action of the joint bearing for sprocket *A* in the presence of a viscous lubricant.

Viscous friction coefficient of sprocket B — Sprocket *B* friction
0.001 N*m/(rad/s) (default)

Friction coefficient due to the rolling action of the joint bearing for sprocket *B* in the presence of a viscous lubricant.

Maximum Tension

These settings are enabled when **Chain model** is set to `Model chain compliance and backlash`.

Maximum tension — Tension limit model
`No maximum tension` (default) | `Specify maximum tension`

Select whether to constrain the maximum tensile force in the drive chain.

- `No maximum tension` — Chain tension can be arbitrarily large during simulation.
- `Specify maximum tension` — Chain tension must remain lower than a maximum value. If the tension exceeds this value, the simulation generates an error and stops.

Dependencies

When this parameter is set to `Specify maximum tension`, related parameters are enabled.

This parameter is enabled when **Chain model** is set to `Model chain compliance and backlash`.

Chain maximum tension — Upper tension limit
`1e6` N (default)

Maximum allowed value of the tensile force acting in the chain.

Dependencies

This parameter is enabled when both of these conditions are met:

- In the **Geometry** settings, the **Chain model** parameter is set to `Model chain compliance and backlash`.
- Im the **Maximum tension** settings, the **Maximum tension** parameter is set to `Specify maximum tension`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Simple Gear | Variable Ratio Transmission

Topics
"Gear Coupling Control Using Clutches"

Introduced in R2012a

Compound Planetary Gear

Planetary gear train with stepped planet gear set
Library: Simscape / Driveline / Gears
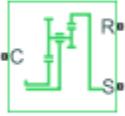
Description

The Compound Planetary Gear block represents a planetary gear train with composite planet gears. Each composite planet gear is a pair of rigidly connected and longitudinally arranged gears of different radii. One of the two gears engages the centrally located sun gear while the other engages the outer ring gear.
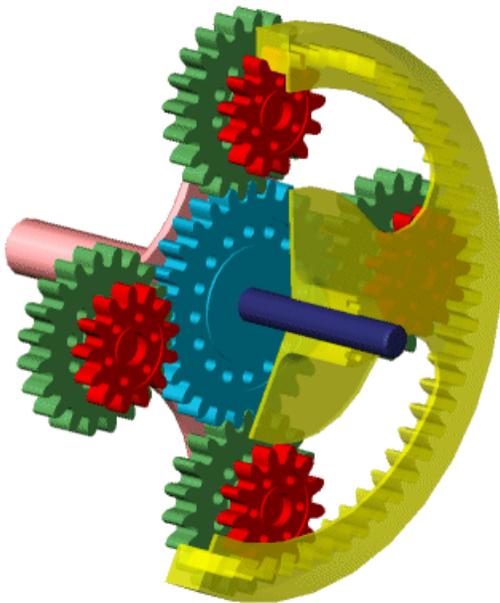
Compound Planetary Gear

The block models the compound planetary gear as a structural component based on the Simscape Driveline Sun-Planet and Ring-Planet blocks. The figure demonstrates the equivalent block diagram for the Compound planetary gear block.
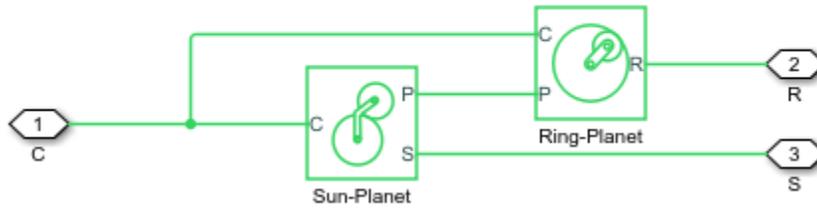
To increase the fidelity of the gear model, specify properties such as gear inertia, meshing losses, and viscous losses. By default, gear inertia and viscous losses are assumed to be negligible. The block enables you to specify the inertias of the internal planet gears. To model the inertias of the carrier, sun, and ring gears, connect Simscape Inertia blocks to ports **C**, **S**, and **R**.

Thermal Model

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Friction model** to `Temperature-dependent efficiency`.

Equations

Ideal Gear Constraints and Gear Ratios

The Compound Planetary Gear block imposes two kinematic and two geometric constraints.

$$r_C \omega_C = r_S \omega_S + r_{P1} \omega_P,$$

$$r_R \omega_R = r_C \omega_C + r_{P2} \omega_P,$$

$$r_C = r_S + r_{P1},$$

$$r_R = r_C + r_{P2},$$

where:

- r_C is the radius of the carrier gear.
- ω_C is the angular velocity of the carrier gear.
- r_S is the radius of the sun gear.
- ω_S is the angular velocity of the sun gear.
- r_{P1} is the radius of planet gear 1.
- ω_P is the angular velocity of the planet gears.
- r_{P2} is the radius of planet gear 2.
- r_R is the radius of the ring gear.

The ring-planet and planet-sun gear ratios are:

$$g_{RP} = r_R/r_{P2} = N_R/N_{P2}$$

and

$$g_{PS} = r_{P1}/r_S = N_{P1}/N_S,$$

where:

- g_{RP} is the ring-planet gear ratio.
- N_R is the number of teeth on the ring gear.
- N_{P2} is the number of teeth on planet gear 2.
- g_{PS} is the planet-sun gear ratio.
- N_{P1} is the number of teeth on planet gear 1.
- N_S is the number of teeth on the sun gear.

In terms of the gear ratios, the key kinematic constraint is:

$$(1 + g_{RP}g_{PS})\omega_C = \omega_S + g_{RP}g_{PS}\omega_R.$$

The four degrees of freedom reduce to two independent degrees of freedom. The gear pairs are (1, 2) = (*P2*, *R*) and (*S*, *P1*).

Warning The gear ratio g_{RP} must be strictly greater than one.

The torque transfers are:

$$g_{RP}\tau_{P2} + \tau_R - \tau_{loss}(P2, R) = 0$$

and

$$g_{PS}\tau_S + \tau_{P1} - \tau_{loss}(S, P1) = 0,$$

where:

- τ_{P2} is torque transfer for planet gear 2.
- τ_R is torque transfer for the ring gear.
- τ_{loss} is torque transfer loss.
- τ_S is torque transfer for the sun gear.
- τ_{P1} is torque transfer for planet gear 1.

In the ideal case where there is no torque loss, $\tau_{loss} = 0$.

Nonideal Gear Constraints and Losses

In the nonideal case, $\tau_{loss} \neq 0$. For more information, see "Model Gears with Losses".

Assumptions and Limitations

- Gear inertia is assumed to be negligible.
- Gears are treated as rigid components.
- Coulomb friction slows down simulation. For more information, see "Adjust Model Fidelity".

Ports

Conserving

C — Planet gear carrier
rotational mechanical

Rotational mechanical conserving port associated with the planet gear carrier.

R — Ring gear
rotational mechanical

Rotational mechanical conserving port associated with the ring gear.

S — Sun gear
rotational mechanical

Rotational mechanical conserving port associated with the sun gear.

H — Heat flow
thermal

Thermal conserving port associated with heat flow. Heat flow affects the power transmission efficiency by altering the gear temperatures.

Dependencies

To enable this port, set **Friction model** to `Temperature-dependent efficiency`.

Parameters

Main

Ring (R) to planet (P) teeth ratio (NR/NP) — Ring to planet gear rotation ratio
2 (default) | scalar greater than 1

Fixed ratio, g_{RP}, of the ring gear to the planet gear rotations as defined by the number of ring gear teeth divided by the number of planet gear teeth. This gear ratio must be strictly greater than 1.

Planet (P) to sun (S) teeth ratio (NP/NS) — Planet to sun gear rotation ratio
1 (default) | positive scalar

Fixed ratio, g_{PS}, of the planet gear to the sun gear rotations as defined by the number of planet gear teeth divided by the number of sun gear teeth. This gear ratio must be strictly greater than 0.

Meshing Losses

Friction model — Friction model
No meshing losses - Suitable for HIL simulation (default) | Constant efficiency | Temperature-dependent efficiency

Friction model for the block:

- `No meshing losses - Suitable for HIL simulation` — Gear meshing is ideal.

- `Constant efficiency` — Transfer of torque between the gear wheel pairs is reduced by a constant efficiency, η, such that $0 < \eta \leq 1$.
- `Temperature-dependent efficiency` — Transfer of torque between the gear wheel pairs is defined by the table lookup based on the temperature.

Sun-planet and ring-planet ordinary efficiencies — Torque transfer efficiencies
[.96 .98] (default) | vector

Vector of torque transfer efficiencies, [η_{SP} η_{RP}], for sun-planet and ring-carrier gear wheel pair meshings, respectively. The vector element must be in the interval (0,1].

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency`.

Temperature — Temperature
[280, 300, 320] K (default) | vector

Vector of temperatures used to construct a 1-D temperature-efficiency lookup table. The vector elements must increase from left to right.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Sun-planet efficiency — Sun-planet efficiency
[.95, .9, .85] (default) | vector | rot

Vector of output-to-input power ratios that describe the power flow from the sun gear to the planet gear, η_{SP}. The block uses the values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Ring-planet efficiency — Ring-planet efficiency
[.95 .9 .85] (default) | vector

Vector of component efficiencies, η_{RP}—the ratio of output power to input power, that the block uses to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Sun-carrier and planet-carrier power thresholds — Minimum efficiency power threshold for the sun-carrier and planet carrier gear couplings
[.001, .001] W (default) | vector

Vector of power thresholds above which the full efficiency factors apply. Enter the thresholds in the order sun-carrier, planet-carrier. Below these values, a hyperbolic tangent function smooths the efficiency factor.

When you set **Friction model** to `Constant efficiency`, the block lowers the efficiency losses to zero when no power is transmitted. When you set **Friction model** to `Temperature-dependent efficiency`, the block smooths the efficiency factors between zero when at rest and the values provided by the temperature-efficiency lookup tables at the power thresholds.

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency` or `Temperature-dependent efficiency`.

Viscous Losses

Sun-carrier and planet-carrier viscous friction coefficients — Gear viscous friction
`[0 0] N*m/(rad/s)` (default) | vector

Vector of viscous friction coefficients, $[\mu_S\ \mu_P]$, for the sun-carrier and planet-carrier gear motions, respectively.

Inertia

Inertia — Inertia model
`off` (default) | `on`

Inertia model for the block:

- `Off` — Model gear inertia.
- `On` — Neglect gear inertia.

Planet gear inertia — Planet gear inertia
`0.001 kg*m^2` (default) | positive scalar

Moment of inertia of the combined planet gears. This value must be positive.

Dependencies

To enable this parameter, set **Inertia** to `On`.

Thermal Port

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Thermal mass — Thermal mass
`50 J/K` (default) | scalar

Thermal energy required to change the component temperature by a single temperature unit. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Initial temperature — Initial temperature
300 K (default) | scalar

Block temperature at the start of simulation. The initial temperature sets the initial component efficiencies according to their respective efficiency vectors.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

More About

Hardware-in-the-Loop Simulation

For optimal simulation performance, set **Friction model** to the default value, `No meshing losses - Suitable for HIL simulation`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Planetary Gear | Ravigneaux Gear | Ring-Planet | Sun-Planet

Topics
"Basic Motion, Torque, and Force Modeling"
"Model Thermal Losses in Driveline Components"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

Cone Clutch

Friction clutch with conical plates that engage when normal force exceeds threshold
Library: Simscape / Driveline / Clutches
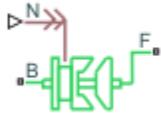
Description

The Cone Clutch block represents a friction clutch with a conical contact interface. The conical interface reduces the normal force required for clutch engagement by creating a wedging action between the clutch components, a cone and a cup. Cone clutch applications include synchromesh gearboxes, which synchronize the drive and driven shaft speeds to enable smoother engagement between transmission gears.

The cup component connects rigidly to the drive shaft, spinning with it as a unit. The cone component connects rigidly to the driven shaft, which sits in axial alignment with the drive shaft. The clutch engages when the cone slides toward the cup and presses tightly against its internal surface. Friction at the cone-cup contact interface enables the clutch to transmit rotational power between the drive and driven shafts. The friction model of this block includes both static and kinetic friction contributions, the latter of which leads to power dissipation during slip between the cone and cup components.

The Cone Clutch block is based on the Fundamental Friction Clutch block. For the complete friction clutch model, see Fundamental Friction Clutch. This section discusses the specialized model implemented in the Cone Clutch block.

When you apply a normal force, F_N, the Cone Clutch block can apply two kinds of friction, kinetic and static, to the driveline motion. The clutch applies kinetic friction torque only when one driveline axis is spinning relative to the other driveline axis. The clutch applies static friction torque when the two driveline axes lock and spin together. The block iterates through multistep testing to determine when to lock and unlock the clutch.

You can also enable faulting. When faulting occurs, the clutch will remain locked or will be unable to transmit power. Faults can occur at a specified time or due to an external trigger at port **T**.

Clutch Geometry and Variables

The figure shows the cone clutch geometry.
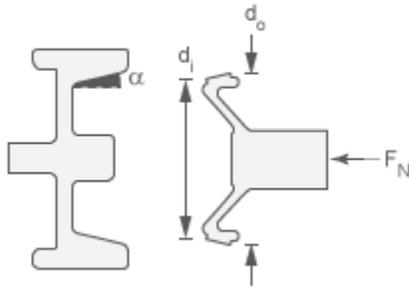
Clutch Variables

Parameter	Definition	Significance
d_o	Outer diameter of the conical contact surface	*See the preceding figure*
d_i	Inner diameter of the conical contact surface	*See the preceding figure*
α	Cone half angle	*See the preceding figure*
ω	Relative angular velocity (slip)	$\omega_F - \omega_B$
ω_{Tol}	Slip tolerance for clutch locking	*See the following model*
F_N	Normal force applied to conical surfaces	Normal force applied, if greater than threshold: $F_N > F_{th}$
α	Cone half-angle	*See the preceding figure*
r_{eff}	Effective torque radius	Effective moment arm of clutch friction force
k_K	Kinetic friction coefficient	Dimensionless coefficient of kinetic friction of conical friction surfaces. Function of ω.
k_S	Static friction coefficient	Dimensionless coefficient of static friction of conical friction surfaces.
τ_K	Kinetic friction torque	*See the following model*
τ_S	Static friction torque limit	(static friction peak factor)·(kinetic friction torque for $\omega \to 0$) (*See the following model*)

Relation to Fundamental Friction Clutch

The Cone Clutch block is based on the Fundamental Friction Clutch block. Instead of requiring the kinetic and static friction limit torques as input signals, the Cone Clutch block calculates the kinetic and static friction from the clutch parameters and the input normal force signal F_N.

Kinetic Friction

The kinetic friction torque is the product of four factors:

$$\tau_K = k_K \cdot F_N \cdot r_{eff} \cdot \text{sgn}(\omega)$$

The kinetic friction torque opposes the relative slip and is applied with an overall minus sign. It changes sign when ω changes sign.

You specify the kinetic friction coefficient, k_K, as either a constant or a tabulated discrete function of relative angular velocity, ω. The tabulated function is assumed to be symmetric for positive and negative values of the relative angular velocity. Therefore, specify k_K for positive values of ω only.

The effective torque radius, r_{eff}, is the effective radius, measured from the driveline axis, at which the kinetic friction forces are applied at the frictional surfaces. It is related to the geometry of the conical friction surface geometry by:

$$r_{\text{eff}} = \frac{1}{3\sin\alpha}\frac{d_o{}^3 - d_i{}^3}{d_o{}^2 - d_i{}^2}$$

d_o and d_i are the contact surface maximum and minimum diameters, respectively.

Static Friction

The static friction limit is related to the kinetic friction, setting ω to zero and replacing the kinetic with the static friction coefficient:

$$T_S = k_S \cdot F_N \cdot r_{eff} \geq 0$$

$k_S > k_K$, so that the torque, τ, needed across the clutch to unlock it by overcoming static friction is larger than the kinetic friction at the instant of unlocking, when $\omega = 0$.

The static friction limit defines symmetric *static friction torque limits* as:

$$\tau_S \equiv \tau_S{}^+ = -\tau_S{}^-$$

The range $[\tau_S{}^-, \tau_S{}^+]$ is used by the Fundamental Friction Clutch.

Engagement and Locking Conditions

The clutch engages (transmits torque) when the conical friction surfaces are subject to a positive normal force and generate kinetic friction: $F_N > 0$ and $\tau_K > 0$.

The clutch locks if and only if it is engaged, and the slip is less than the velocity tolerance: $|\omega| < \omega_{\text{Tol}}$.

Power Dissipated by the Clutch

The power dissipated by the clutch is $|\omega \cdot \tau_K|$. The clutch dissipates power only if it is both slipping ($\omega \neq 0$) and applying kinetic friction ($\tau_K > 0$).

Velocity-Dependent Model

You can model the effects of rotational velocity change by selecting a velocity-dependent model. To choose a velocity-dependent model, in the **Friction** settings, set the **Friction model** parameter to `Velocity-dependent kinetic friction coefficient`. For information about a friction model that depends on both velocity and temperature, see "Thermal, Velocity-Dependent Model" on page 1-78.

For the velocity-dependent model these related parameters become visible in the **Friction** settings:

- **Relative velocity vector**
- **Kinetic friction coefficient vector**
- **Friction coefficient interpolation method**
- **Friction coefficient extrapolation method**

Thermal Model

You can model the effects of heat flow and temperature change by selecting a temperature-dependent model. To choose a temperature-dependent model, in the **Friction** settings, set the **Friction model** parameter to `Temperature-dependent friction coefficients`. For information about a friction model that depends on both velocity and temperature, see "Thermal, Velocity-Dependent Model" on page 1-78.

For the temperature-dependent model, thermal port **H** and these settings are visible:

- In the **Friction** settings:

  - **Temperature vector**
  - **Static friction coefficient vector**
  - **Kinetic friction coefficient vector**
  - **Friction coefficient interpolation method**
  - **Friction coefficient extrapolation method**
- In the **Thermal Port** settings:

  - **Thermal mass**
  - **Initial Temperature**

Thermal, Velocity-Dependent Model

You can model the effects of rotational velocity change and heat flow by selecting a velocity-dependent and temperature-dependent model. To choose a model that depends on both velocity and temperature, in the **Friction** settings, set the **Friction model** parameter to `Temperature and velocity-dependent friction coefficients`.

For the velocity-dependent and temperature-dependent model, thermal port **H** and these related settings and parameters become visible:

- In the **Friction** settings:

  - **Relative velocity vector**
  - **Temperature vector**
  - **Static friction coefficient vector**
  - **Kinetic friction coefficient matrix**
  - **Friction coefficient interpolation method**
  - **Friction coefficient extrapolation method**
- In the **Thermal Port** settings:

  - **Thermal mass**
  - **Initial Temperature**

Faulty Behavior

You can enable faulty behavior in response to:

- Simulation time — Faulting occurs at a specified time.

• Simulation behavior — Faulting occurs in response to an external trigger. This exposes port **T**.

You can choose either or both of these settings for block faulting. If faulting is triggered, the clutch responds according to the **Behavior when faulted** setting for the remainder of the simulation. The fault options are:

• `Cannot transmit power`

• `Cannot unlock`

You can set the block to issue a fault report as a warning or error message in the Simulink Diagnostic Viewer with the **Reporting when fault occurs** parameter.

Ports

Input

N — Normal force
physical signal

Physical signal port associated with the normal force. This signal is positive or zero. A signal of less than zero is interpreted as zero.

Dependencies

This port is visible only if, in the **Geometry** settings, the **Shift linkage control** parameter is set to `Physical signal`. For more information, see "Shift linkage control" on page 1-0   and Geometry Parameter Dependencies.

T — External fault trigger
physical signal

Physical signal port for an external fault trigger. Triggering occurs when the value is greater than 0.5. There is no unit associated with the trigger value.

Dependencies

This port is visible when **Enable faults** is set to `On` and **Enable external fault trigger** is set to `On`.

Output

X — Shift linkage position
physical signal

Physical signal port associated with shift linkage position.

Dependencies

This port is visible only when, in the **Geometry** settings, the **Shift linkage control** parameter is set to `Conserving port`. For more information, see "Shift linkage control" on page 1-0   and Geometry Parameter Dependencies.

Conserving

B — Base
mechanical rotational

Mechanical rotational conserving port associated with the driving (base) shaft. The clutch motion is measured as the *slip* $\omega = \omega_F - \omega_B$, the angular velocity of follower relative to base.

F — Follower
mechanical rotational

Mechanical rotational conserving port associated with the driven or follower shaft

H — Heat flow
thermal

Thermal conserving port associated with heat flow.

Dependencies

This port is visible only when, in the **Friction** settings, the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see "Friction model" on page 1-0 and Friction Parameter Dependencies.

S — Shift linkage
mechanical rotational

Mechanical rotational conserving port associated with shift linkage.

Dependencies

This port is visible only when, in the **Geometry** settings, the **Shift linkage control** parameter is set to `Conserving port`. For more information, see "Shift linkage control" on page 1-0 and Geometry Parameter Dependencies.

Parameters

Geometry

The table shows how the visibility of some parameters depends on the option that you choose for other parameters. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Geometry Parameter Dependencies

Geometry	
Contact surface maximum diameter	
Contact surface minimum diameter	
Cone half angle	
Shift linkage control	
Physical signal	Conserving port
Exposes physical signal input port **N**	Exposes: • Conserving port **S** • Physical signal output port **X** • **Shift Linkage** settings

Contact surface maximum diameter — Outer diameter
150 mm (default) | positive scalar

Outer conical diameter d_o.

Contact surface minimum diameter — Inner diameter
100 mm (default) | positive scalar

Inner conical diameter d_i.

Cone half angle — Cone half angle
12 deg (default) | positive scalar

Half opening angle α of the cone geometry.

Shift linkage control — Shift linkage model
Physical signal (default) | Conserving port

Shift linkage control model:

- Physical signal — Expose physical signal ports **N**, which inputs the normal force, and **X**, which outputs for the shift linkage travel.
- Conserving port — Expose conserving port, **S**, which is associated with shift linkage control.

Dependencies

The visibility of **Shift Linkage** settings and ports **S**, **X**, and **N** depend on this setting. For more information, see Geometry Parameter Dependencies.

Shift Linkage

These settings are visible only if, in the **Geometry** settings, the **Shift linkage control** parameter is set to Conserving port. For more information, see "Shift linkage control" on page 1-0   and Geometry Parameter Dependencies.

Hard stop at back of shift linkage — Hard stop
On (default) | Off

Hard stop at back of shift linkage

Ring-hub clearance when disengaged — Ring-hub clearance
3 mm (default) | nonnegative scalar

Ring-hub clearance when disengaged

Ring stop stiffness — Ring stop stiffness
10e5 N/m (default) | nonnegative scalar

Ring stops stiffness

Ring stop damping — Ring stop damping
1e3 N/(m/s) (default) | nonnegative scalar

Ring stop damping

Shift linkage viscous friction coefficient — Shift linkage viscous friction
100 `N/(m/s)` (default) | positive scalar

Shift linkage viscous friction coefficient

Linkage travel direction — Disengagement travel direction
`Positive shift linkage displacement engages clutch` (default) | `Negative shift linkage displacement engages clutch`

Linkage travel direction that disengages the clutch

Friction

The table shows how the visibility of some ports, parameters, and settings depends on the option that you choose for other parameters. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Friction Parameter Dependencies

Friction			
Friction model			
Fixed kinetic friction coefficient	Velocity-dependent kinetic friction coefficient	Temperature-dependent friction coefficients	Temperature and velocity-dependent friction coefficients
		Exposes: • Conserving port **H** • Thermal parameters in the **Friction** settings • **Thermal Port** settings	Exposes: • Conserving port **H** • Thermal parameters in the **Friction** settings • **Thermal Port** settings
-	-	**Temperature vector**	**Temperature vector**
-	**Relative velocity vector**	-	**Relative velocity vector**
Static friction coefficient	**Static friction coefficient**	**Static friction coefficient vector**	**Static friction coefficient vector**
Kinetic friction coefficient	**Kinetic friction coefficient vector**	**Kinetic friction coefficient vector**	**Kinetic friction coefficient matrix**
-	**Friction coefficient interpolation method**	**Friction coefficient interpolation method**	**Friction coefficient interpolation method**
-	**Friction coefficient extrapolation method**	**Friction coefficient extrapolation method**	**Friction coefficient extrapolation method**
Velocity tolerance	**Velocity tolerance**	**Velocity tolerance**	**Velocity tolerance**
Threshold force	**Threshold force**	**Threshold force**	**Threshold force**
Viscous drag torque coefficient	**Viscous drag torque coefficient**	**Viscous drag torque coefficient**	**Viscous drag torque coefficient**

Friction model — Friction model
Fixed kinetic friction coefficient (default) | Velocity-dependent kinetic friction coefficient | Temperature-dependent friction coefficients | Temperature and velocity-dependent friction coefficients

Parameterization method to model the kinetic friction coefficient. The options and default values for this parameter depend on the friction model that you select for the block. The options are:

• Fixed kinetic friction coefficient — Provide a fixed value for the kinetic friction coefficient.

• Velocity-dependent kinetic friction coefficient — Define the kinetic friction coefficient by one-dimensional table lookup based on the relative angular velocity between disks.

• Temperature-dependent friction coefficients — Define the kinetic friction coefficient by table lookup based on the temperature.

- `Temperature and velocity-dependent friction coefficients` — Define the kinetic friction coefficient by table lookup based on the temperature and the relative angular velocity between disks.

Dependencies

The friction model setting affects the visibility of other parameters, settings, and ports. For more information, see Friction Parameter Dependencies.

Relative velocity vector — Relative velocity
`[0, 100, 1000]` rad/s (default) | vector

Input values for the relative velocity as a vector. The values in the vector must increase from left to right. The minimum number of values depends on the interpolation method that you select. For linear interpolation, provide at least two values per dimension. For smooth interpolation, provide at least three values per dimension.

Dependencies

This parameter is visible only when the **Friction model** parameter is set to `Velocity-dependent kinetic friction coefficient` or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Temperature vector — Temperature
`[280, 300, 320]` K (default) | increasing vector

Input values for the temperature as a vector. The minimum number of values depends on the interpolation method that you select. For linear interpolation, provide at least two values per dimension. For smooth interpolation, provide at least three values per dimension. The values in the vector must increase from left to right.

Dependencies

This parameter is visible only when the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Static friction coefficient — Static friction coefficient
`0.35` (default) | scalar

Static or peak value of the friction coefficient. The static friction coefficient must be greater than the kinetic friction coefficient.

Dependencies

this parameter is visible only when the **Friction model** parameter is set to `Fixed kinetic friction coefficient` or `Velocity-dependent kinetic friction coefficient`. For more information, see Friction Parameter Dependencies.

Static friction coefficient vector — Static friction coefficient
`[.4, .38, .36]` (default) | vector

Static, or peak, values of the friction coefficient as a vector. The vector must have the same number of elements as the temperature vector. Each value must be greater than the value of the corresponding element in the kinetic friction coefficient vector.

Dependencies

This parameter is visible only when the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

`Kinetic friction coefficient` — Kinetic friction coefficient
`0.3` (default) | positve scalar

The kinetic, or Coulomb, friction coefficient. The coefficient must be greater than zero.

Dependencies

This parameter is visible only when the **Friction model** parameter is set to `Fixed kinetic friction coefficient`. For more information, see Friction Parameter Dependencies.

`Kinetic friction coefficient vector` — Kinetic friction coefficient
`[.3, .22, .19]` (default) | `[.3, .28, .25]` | vector

Output values for kinetic friction coefficient as a vector. All values must be greater than zero.

If the **Friction model** parameter is set to

- `Velocity-dependent kinetic friction coefficient` — The vector must have same number of elements as relative velocity vector.
- `Temperature-dependent friction coefficients` — The vector must have the same number of elements as the temperature vector.

Dependencies

This parameter is visible only when the **Friction model** parameter is set to `Velocity-dependent kinetic friction coefficient` or `Temperature-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

`Kinetic friction coefficient matrix` — Kinetic friction coefficient
`[.34, .32, .3; .3, .28, .25; .25, .2, .15]` (default) | matrix

Output values for kinetic friction coefficient as a matrix. All the values must be greater than zero. The size of the matrix must equal the size of the matrix that is the result of the temperature vector × the kinetic friction coefficient relative velocity vector.

Dependencies

This parameter is visible only when the **Friction model** parameter is set to `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

`Friction coefficient interpolation method` — Interpolation method
`Linear` (default) | `Smooth`

Interpolation method for approximating the output value when the input value is between two consecutive grid points:

- `Linear` — Select this option to get the best performance.
- `Smooth` — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is visible only when the **Friction model** parameter is set to `Velocity-dependent kinetic friction coefficient`, `Temperature-dependent friction coefficients`, or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Friction coefficient extrapolation method — **Extrapolation method**
`Linear` (default) | `Nearest` | `Error`

Extrapolation method for determining the output value when the input value is outside the range specified in the argument list:

- `Linear` — Select this option to produce a curve with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region.
- `Nearest` — Select this option to produce an extrapolation that does not go above the highest point in the data or below the lowest point in the data.
- `Error` — Select this option to avoid going into the extrapolation mode when you want your data to be within the table range. If the input signal is outside the range of the table, the simulation stops and generates an error.

For more information on extrapolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is visible only when the **Friction model** parameter is set to `Velocity-dependent kinetic friction coefficient`, `Temperature-dependent friction coefficients`, or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Velocity tolerance — **Relative velocity locking threshold**
`0.001 rad/s` (default) | scalar

Relative velocity below which the two surfaces can lock. The surfaces lock if the torque across the **B** and **F** rotational ports is less than the product of the effective radius, the static friction coefficient, and the applied normal force.

Threshold force — **Normal contact force threshold**
`1 N` (default) | scalar

The normal force applied to the physical signal port **N** is applied to the contact only if the amount of force exceeds the value of the **Threshold force** parameter. Forces below the **Threshold force** are not applied so there is no transmitted frictional torque.

Viscous drag torque coefficient — **Viscous drag torque**
`0` (default) | nonnegative scalar

Viscous drag torque coefficient.

Initial Conditions

Initial state — Initial clutch state
Unlocked (default) | Locked

Clutch state at the start of simulation. The clutch can be in one of two states, locked and unlocked. A locked clutch constrains the base and follower shafts to spin at the same velocity, that is, as a single unit. An unlocked clutch allows the two shafts to spin at different velocities, resulting in slip between the clutch plates.

Faults

Enable faults — Fault option
Off (default) | On

Enable externally or temporally triggered faults. When faulting occurs, the clutch fails to unlock or cannot transmit power, according to the **Behavior when faulted** setting.

Behavior when faulted — Set fault response
Cannot transmit power (default) | Cannot unlock

Set fault response. You can select the clutch faulting as either:

- Cannot transmit power
- Cannot unlock

Dependencies

To enable this parameter, set **Enable faults** to On.

Enable external fault trigger — External trigger option
Off (default) | On

Enables port **T**. A physical signal at port **T** that is greater than 0.5 triggers faulting.

Dependencies

To enable this parameter, set **Enable faults** to On.

Enable temporal fault trigger — Temporal trigger option
Off (default) | On

Enables fault triggering at a specified time. When the **Simulation time for fault event** is reached, the clutch responds according to the **Behavior when faulted** setting.

Dependencies

To enable this parameter, set **Enable faults** to On.

Simulation time for fault event — Time at which faulting is triggered
5 s (default) | positive scalar

When the **Simulation time for fault event** is reached, the clutch responds according to the **Behavior when faulted** setting.

Dependencies

To enable this parameter, set **Enable faults** to On and **Enable temporal fault trigger** to On.

Reporting when fault occurs — Fault condition report
None (default) | Warning | Error

Reporting preference for the fault condition. When reporting is set to Warning or Error, a message is displayed in the Simulink Diagnostic Viewer. When Error is selected, the simulation will stop if faulting occurs.

Dependencies

To enable this parameter, set **Enable faults** to On.

Thermal Port

Thermal Port settings are visible only when, in the **Friction** settings, the **Friction model** parameter is set to Temperature-dependent friction coefficients or Temperature and velocity-dependent friction coefficients. For more information, see Friction Parameter Dependencies.

Thermal mass — Thermal mass
50 kJ/K (default) | scalar

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

This parameter is only visible when, in the **Friction** settings, the **Friction model** parameter is set to Temperature-dependent friction coefficients or Temperature and velocity-dependent friction coefficients. For more information, see Friction Parameter Dependencies.

Initial temperature — Initial temperature
300 K (default) | scalar

Component temperature at the start of simulation. The initial temperature alters the component efficiency according to an efficiency vector that you specify, affecting the starting meshing or friction losses.

Dependencies

This parameter is only visible when, in the **Friction** settings, the **Friction model** parameter is set to Temperature-dependent friction coefficients or Temperature and velocity-dependent friction coefficients. For more information, see Friction Parameter Dependencies.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Dog Clutch | Double-Sided Synchronizer | Logic-Controlled Clutch | Synchronizer | Fundamental Friction Clutch | Unidirectional Clutch

Topics
"Gear Coupling Control Using Clutches"

Introduced in R2011a

Cycloidal Drive

High-ratio speed reducer based on cycloidal disk motion
Library: Simscape / Driveline / Gears

Description

The Cycloidal Drive block represents a compact, high-ratio, speed-reduction mechanism that contains four key components:

- An eccentric cam
- A cycloidal disk
- Ring-gear housing
- Pin rollers

The eccentric cam, which extends from the base shaft, sits inside the cycloidal disk. This disk meshes with the ring-gear housing. The pin rollers, which extend from the follower shaft, sit in matching holes on the cycloidal disk.
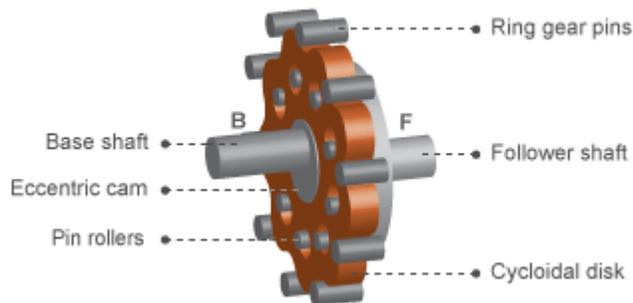
During normal operation, the base shaft drives the eccentric cam. The cam spins inside the cycloidal disk, causing it to rotate in an eccentric pattern about an offset axis. As it moves, the cycloidal disk engages the internal teeth of the ring-gear housing. The internal meshing reverses the rotational velocity direction.

Pin rollers extending from cycloidal disk holes transmit rotational motion to the follower shaft. This shaft spins counter to the base shaft at a highly reduced speed. The large reduction ratio results from the near-equal cycloidal disk and ring gear tooth numbers. The effective gear reduction ratio is

$$r = \frac{n_R - n_C}{n_C},$$

where:

- *r* is the gear reduction ratio.
- n_R is the number of teeth on the ring gear.
- n_C is the number of teeth on the cycloidal disk.

The gear reduction ratio constrains the angular velocities of the base and follower shafts according to the expression

$$\omega_F = -r\omega_B,$$

where:

- ω_F is the angular velocity of the follower shaft.
- ω_C is the angular velocity of the base shaft.

The gear reduction ratio also constrains the torques acting on the base and follower shafts, according to the expression

$$T_B = rT_F + T_f,$$

where:

- T_B is the net torque at the base shaft.
- T_F is the net torque at the follower shaft.
- T_f is the torque loss due to friction. For more information, see "Model Gears with Losses".

The figure shows the cycloidal drive in front and side views. The kinematics of the drive system cause a reversal of the base and follower shaft angular velocities so that the two shafts spin in opposite directions.
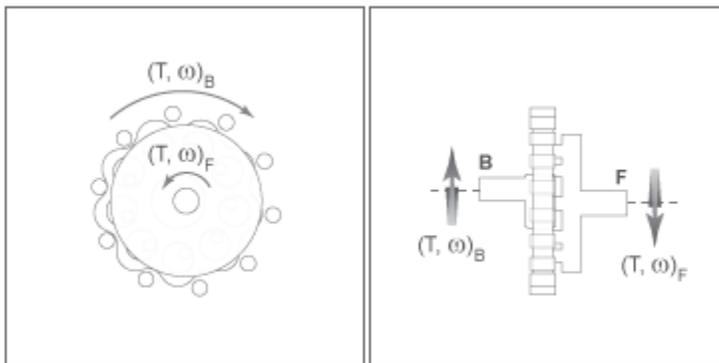
The cycloidal drive can operate in reverse mode, that is, with power flowing from the follower shaft to the base shaft. In reverse mode, torque transfer efficiencies are typically negligible. You can adjust the efficiency by changing the value of the **Efficiency from follower shaft to base shaft** parameter.

Friction Model

You can set the meshing losses friction model to:

- `No meshing losses - suitable for HIL simulation`, which ignores losses to allow for HIL-capable computation times.

- `Constant efficiency`, which is the default friction setting for block versions prior to R2020b. In this case, you specify a component efficiency that remains constant throughout the simulation
- `Temperature-dependent efficiency`, which models temperature-dependent component efficiencies by creating a 1-D lookup table based on the **Temperature** vector and the given component efficiency vector. This setting also,enables a thermal conserving port **H**. This port receives the heat flow into the block, which is translated into the block temperature according to the gear's **Thermal mass** parameter.

Thermal Model

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Friction model** to `Temperature-dependent efficiency`.

Variables

Use the **Variables** settings to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Ports

Conserving

B — Base shaft
mechanical rotational

Rotational mechanical conserving port associated with the base shaft.

F — Follower shaft
mechanical rotational

Rotational mechanical conserving port associated with the follower shaft.

H — Heat flow
thermal

Thermal conserving port associated with heat flow.

Dependencies

To enable this port, set **Friction model** to `Temperature-dependent efficiency`.

Parameters

Main

Number of teeth on cycloid disk — Number of cycloid disk teeth
20 (default) | positive scalar integer

Total number of teeth projecting outward from the cycloidal disk perimeter. This quantity should be less than the number of teeth or pins on the ring gear. The ratio of the gear tooth numbers defines the relative angular velocities of the base and follower shafts.

Number of teeth on ring gear — Number of ring gear teeth
24 (default) | positive scalar integer

Total number of teeth or pins projecting inward from the ring gear housing. This quantity should be greater than the number of teeth on the cycloidal disk. The ratio of the two gear tooth numbers defines the relative angular velocities of the base and follower shafts.

Meshing Losses

Friction model — Model for meshing losses
No meshing losses - Suitable for HIL simulation (default) | Constant efficiency | Temperature-dependent efficiency

- No meshing losses — Suitable for HIL simulation — Gear meshing is ideal.
- Constant efficiency — Transfer of torque between worm and gear is reduced by friction.
- Temperature-dependent efficiency — Torque transfer is determined from user-supplied data for meshing efficiency and temperature.

Temperature — Temperature
[280, 300, 320] K (default) | vector

Vector of temperatures used to construct 1-D temperature-efficiency lookup tables. Values of the vector elements must increase from left to right. The number of elements in the vector must match the number of elements in the vectors specified for these parameters:

- "Efficiency from base shaft to follower shaft" on page 1-0
- "Efficiency from follower shaft to base shaft" on page 1-0

Dependencies

To enable this port, set **Friction model** to Temperature-dependent efficiency.

Efficiency from base shaft to follower shaft — Base-to-follower efficiency
0.90 or [0.95, 0.85, 0.80] (default) | scalar or vector of elements in the interval (0,1)

Torque transfer efficiency with the base shaft driving the follower shaft. Efficiency values must fall in the interval (0,1). Larger efficiency values correspond to greater torque transfer between the base and follower shafts. Values approaching unity are typical.

How you specify the efficiency values depends on which **Friction model** you select:

- Constant efficiency — Specify the value as a scalar.
- Temperature-dependent efficiency— Specify the value as a vector in which each element is the ratio of output power to input power at the corresponding the temperature element in the temperature vector. The number of elements in the vector must match the number of elements in the vectors specified for the "Temperature" on page 1-0 parameters.

Dependencies

This parameter is available when you set **Friction model** to Constant efficiency or Temperature-dependent efficiency. For more information, see "Thermal Model" on page 1-92.

Efficiency from follower shaft to base shaft — Follower-to-base efficiency
0.05 or [0.10, 0.05, 0.03] (default) | scalar or vector of elements in the interval (0,1)

Torque transfer efficiency in reverse operation mode, that is, with the follower shaft driving the base shaft. Efficiency values must fall in the interval (0,1). Larger efficiency values correspond to greater torque transfer between the base and follower shafts. Values approaching zero are typical.

How you specify the efficiency values depends on which **Friction model** you select:

- `Constant efficiency` — Specify the value as a scalar.
- `Temperature-dependent efficiency`— Specify the value as a vector in which each element is the ratio of output power to input power at the corresponding the temperature element in the temperature vector. The number of elements in the vector must match the number of elements in the vectors specified for the "Temperature" on page 1-0   parameters.

Dependencies

This parameter is available when you set **Friction model** to `Constant efficiency` or `Temperature-dependent efficiency`. For more information, see "Thermal Model" on page 1-92.

Power threshold — Minimum efficiency power threshold
`0.001` W (default) | scalar

Absolute value of the cycloidal disk power above which the full efficiency factor applies. Below this value, a hyperbolic tangent function smooths the efficiency factor.

When you set **Friction model** to `Constant efficiency`, a hyperbolic tangent function smooths the efficiency factor to one, such that the efficiency losses go to zero at the resting state.

When you set **Friction model** to `Temperature-dependent efficiency`, a hyperbolic tangent function smooths the efficiency factor between zero when at rest and the value provided by the temperature-efficiency lookup table when at the specified power threshold.

The power threshold should be lower than the expected power transmitted during simulation. Higher values can cause the block to underestimate efficiency losses. However, very low values may raise the computational cost of simulation.

Thermal Port

To enable these settings, set **Friction model** to `Temperature-dependent efficiency`. For more information, see "Thermal Model" on page 1-92.

Thermal mass — Thermal mass
`50` J/K (default) | scalar

Thermal energy required to change the component temperature by a single temperature unit. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Initial temperature — Initial temperature
`300` K (default) | scalar

Component temperature at the start of simulation. The initial temperature alters the component efficiency according to an efficiency vector that you specify, affecting the starting meshing or friction losses.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Harmonic Drive | Simple Gear | Simple Gear with Variable Efficiency

Introduced in R2014a

Differential

Gear mechanism that allows driven shafts to spin at different speeds
Library: Simscape / Driveline / Gears

Description

The Differential block represents a gear mechanism that allows the driven shafts to spin at different speeds. Differentials are common in automobiles, where they enable the various wheels to spin at different speeds while cornering. Ports **D**, **S1**, and **S2** represent the longitudinal driveshaft and the sun gear shafts of the differential, respectively. Any one of the shafts can drive the other two.

The block models the differential mechanism as a structural component based on the Simple Gear and Sun-Planet Bevel Simscape Driveline blocks. The figure demonstrates the equivalent block diagram for the Differential block.
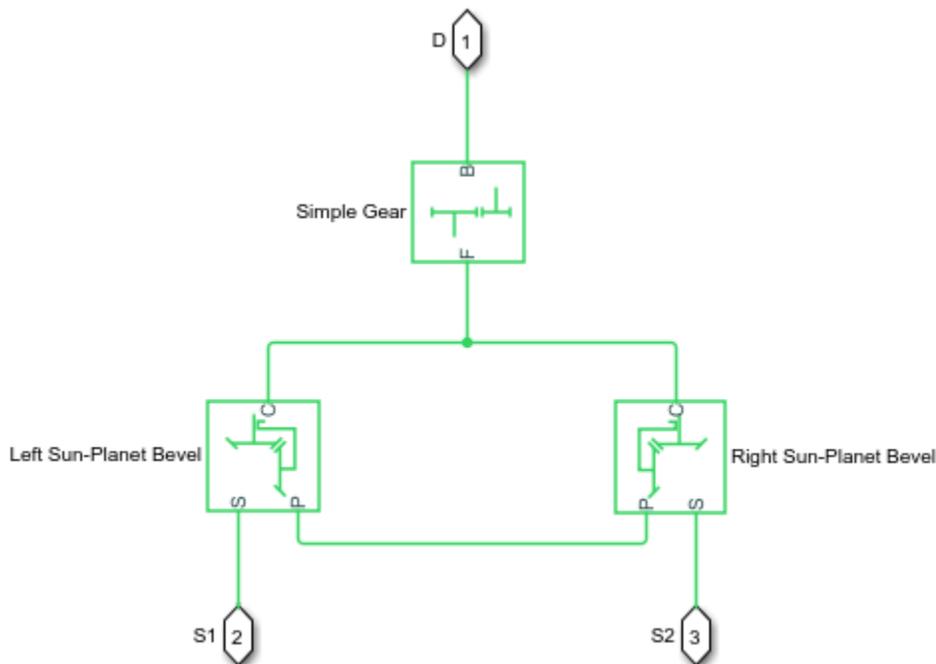
To increase the fidelity of the gear model, specify properties such as gear inertia, meshing losses, and viscous losses. By default, gear inertia and viscous losses are assumed to be negligible. The block enables you to specify the inertias of the gear carrier and internal planet gears. To model the inertias of the outer gears, connect Simscape Inertia blocks to ports **D**, **S1**, and **S2**.

Thermal Modeling

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Friction model** to `Temperature-dependent efficiency`.

Equations

Ideal Gear Constraints and Gear Ratios

The differential imposes one kinematic constraint on the three connected axes, such that

$$\omega_{S1} - \omega_{S2},$$

where:

- ω_{S1} is the velocity of sun gear shaft 1.
- ω_{S2} is the velocity of sun gear shaft 2.

Negative values imply that the differential is left of centerline. The three degrees of freedom reduce to two independent degrees of freedom. The gear pairs are $(1,2) = (S, S)$ and (C, D). C is the carrier.

The sum of the lateral motions is the transformed longitudinal motion. The difference of side motions, $\omega_{S1} - \omega_{S2}$, is independent of the longitudinal motion. The general motion of the lateral shafts is a superposition of these two independent degrees of freedom, which have this physical significance:

- The longitudinal degree of freedom is equivalent to the two lateral shafts rotating at the same angular velocity, $\omega_{S1} = \omega_{S2}$, and at a fixed ratio with respect to the longitudinal shaft.
- The differential degree of freedom is equivalent to keeping the longitudinal driving shaft locked, $\omega_D = 0$, where ω_D is the velocity of the driving shaft, while the lateral shafts rotate with respect to each other in opposite directions, $\omega_{S1} = -\omega_{S2}$.

The lateral axis torques are constrained by the longitudinal axis torque such that the net power flow sums to zero:

$$\omega_{S1}\tau_{S1} + \omega_{S2}\tau_{S2} + \omega_D\tau_D - P_{loss} = 0,$$

where:

- τ_{S1} and τ_{S2} are the torques along the lateral axes.
- τ_D is the longitudinal torque.
- P_{loss} is the power loss.

When the kinematic and power constraints are combined, the ideal case yields

$$g_D\tau_D = 2\frac{(\omega_{S1}\tau_{S1} + \omega_{S2}\tau_{S2})}{\omega_{S1} + \omega_{S2}},$$

where g_D is the gear ratio for the longitudinal driveshaft.

Ideal Fundamental Constraints

The effective Differential block constraint is composed of two sun-planet bevel gear subconstraints.

- The first subconstraint is due to the coupling of the two sun-planet bevel gears to the carrier:

$$\frac{\omega_{S1} - \omega_C}{\omega_{S2} - \omega_C} = -\frac{g_{SP2}}{g_{SP1}},$$

where g_{SP1} and g_{SP2} are the gear ratios for the sun-planet gears.

- The second subconstraint is due to the coupling of the carrier to the longitudinal driveshaft:

$$\omega_D = -g_D\omega_C.$$

The sun-planet gear ratios of the underlying sun-planet bevel gears, in terms of the radii, *r*, of the sun-planet gears are:

$$g_{SP1} = \frac{r_{S1}}{r_{P1}}$$

$$g_{SP2} = \frac{r_{S2}}{r_{P2}}$$

The Differential block is implemented with $g_{SP1} = g_{SP2} = 1$, leaving g_D free to adjust.

Nonideal Gear Constraints and Losses

In the nonideal case, $\tau_{loss} \neq 0$. For more information, see "Model Gears with Losses".

Assumptions and Limitations

- The gears are assumed to be rigid.
- Coulomb friction slows down simulation. For more information, see "Adjust Model Fidelity".

Ports

Conserving

D — Driveshaft
rotational mechanical

Rotational mechanical conserving port associated with the longitudinal driveshaft.

S1 — Sun gear 1
rotational mechanical

Rotational conserving port associated with sun gear 1.

S2 — Sun gear 2
rotational mechanical

Rotational conserving port associated with sun gear 2.

H — Heat flow
thermal

Thermal conserving port associated with heat flow. Heat flow affects the power transmission efficiency by altering the gear temperatures.

Dependencies

To enable this port, set **Friction model** to `Temperature-dependent efficiency`.

Parameters

Main

Crown gear located — Crown gear location
`To the right of centerline` (default) | `To the left of centerline`

Location of the bevel crown gear relative to the centerline of the gear assembly.

Carrier (C) to driveshaft (D) teeth ratio (NC/ND) — Carrier to driveshaft gear rotation ratio
`4` (default) | positive scalar

Fixed ratio, g_D, of the carrier gear to the longitudinal driveshaft gear rotations as defined by the number of carrier gear teeth divided by the number of driveshaft gear teeth. This gear ratio must be strictly greater than 0.

Meshing Losses

Friction model — Friction model
`No meshing losses - Suitable for HIL simulation` (default) | `Constant efficiency` | `Temperature-dependent efficiency`

Friction model for the block:

- `No meshing losses - Suitable for HIL simulation` — Gear meshing is ideal.
- `Constant efficiency` — Transfer of torque between the gear wheel pairs is reduced by a constant efficiency, η, such that $0 < \eta \leq 1$.
- `Temperature-dependent efficiency` — Transfer of torque between the gear wheel pairs is defined by the table lookup based on the temperature.

Sun-sun and carrier-driveshaft ordinary efficiencies — Torque transfer efficiencies
`[.85, .92]` (default) | vector

Vector of torque transfer efficiencies, $[\eta_{SS}, \eta_{CD}]$, from driving to driven sun gear and from the carrier to the longitudinal driveshaft, respectively. The vector elements must be in the range (0,1).

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency`.

Temperature — Temperature
`[280, 300, 320]` K (default) | vector

Vector of temperatures used to construct a 1-D temperature-efficiency lookup table. The vector elements must increase from left to right.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Sun-sun efficiency — Driving sun gear to driven sun gear torque transfer efficiency
`[.95, .9, .85]` (default) | vector

Vector of output-to-input power ratios that describe the power flow from the driving sun gear to the driven sun gear, η_{SS}. The block uses the values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Carrier-driveshaft efficiency — Carrier to driveshaft gear torque transfer efficiency
`[.95, .9, .85]` (default) | vector

Vector of output-to-input power ratios that describe the power flow from the carrier to the driveshaft, η_{CD}. The block uses the values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Sun-carrier and driveshaft-casing power thresholds — Minimum efficiency power threshold for the sun-carrier and driveshaft-casing gear couplings
`[0.001, 0.001]` W (default) | vector

Vector of power thresholds, p_{th}, for sun-carrier and longitudinal driveshaft-casing [p_S, p_D], respectively. The full efficiency loss applies above these values. Below these values, a hyperbolic tangent function smooths the efficiency factor.

When you set **Friction model** to `Constant efficiency`, the block lowers the efficiency losses to zero when no power is transmitted. When you set **Friction model** to `Temperature-dependent efficiency`, the block smooths the efficiency factors between zero when at rest and the values provided by the temperature-efficiency lookup tables at the power thresholds.

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency` or `Temperature-dependent efficiency`.

Viscous Losses

Sun-carrier and driveshaft-casing viscous friction coefficients — Viscous friction coefficients
`[0, 0]` N*m/(rad/s) (default) | vector

Vector of viscous friction coefficients, [μ_S μ_D], for the sun gear-to-carrier gears and longitudinal driveshaft-to-casing gear motions, respectively.

Inertia

`Inertia` — Inertia model
off (default) | on

Inertia model for the block:

- `Off` — Model gear inertia.
- `On` — Neglect gear inertia.

`Carrier inertia` — Carrier inertia
0.001 kg*m^2 (default) | positive scalar

Moment of inertia of the planet gear carrier. This value must be positive.

Dependencies

To enable this parameter, set **Inertia** to On.

`Planet gear inertia` — Planet gear inertia
0.001 kg*m^2 (default) | positive scalar

Moment of inertia of the combined planet gears. This value must be positive.

Dependencies

To enable this parameter, set **Inertia** to On.

Thermal Port

To enable these settings, set **Friction model** to Temperature-dependent efficiency.

`Thermal mass` — Thermal mass
50 J/K (default) | scalar

Thermal energy required to change the component temperature by a single temperature unit. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

To enable this parameter, set **Friction model** to Temperature-dependent efficiency.

`Initial temperature` — Initial temperature
300 K (default) | scalar

Block temperature at the start of simulation. The initial temperature sets the initial component efficiencies according to their respective efficiency vectors.

Dependencies

To enable this parameter, set **Friction model** to Temperature-dependent efficiency.

More About

Hardware-in-the-Loop Simulation

For optimal simulation performance, set **Friction model** to the default value, `No meshing losses - Suitable for HIL simulation`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Disk Friction Clutch | Simple Gear | Sun-Planet Bevel

Topics
"Basic Motion, Torque, and Force Modeling"
"Limited Slip Differential with Clutches"
"Torsen Differential"
"Vehicle with Four-Wheel Drive"
"Model Thermal Losses in Driveline Components"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

Disc Brake

Frictional brake with pressure-applying cylinder and pads with faulting
Library: Simscape / Driveline / Brakes & Detents / Rotational
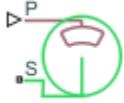
Description

The Disc Brake block represents a brake arranged as a cylinder applying pressure to one or more pads that can contact the shaft rotor. Pressure from the cylinder causes the pads to exert friction torque on the shaft. The friction torque resists shaft rotation.

You can also enable faulting. When faulting occurs, the brake will exert a user-specified pressure. Faults can occur at a specified time or due to an external trigger at port **T**.

Disc Brake Model

This figure shows the side and front views of a disc brake.
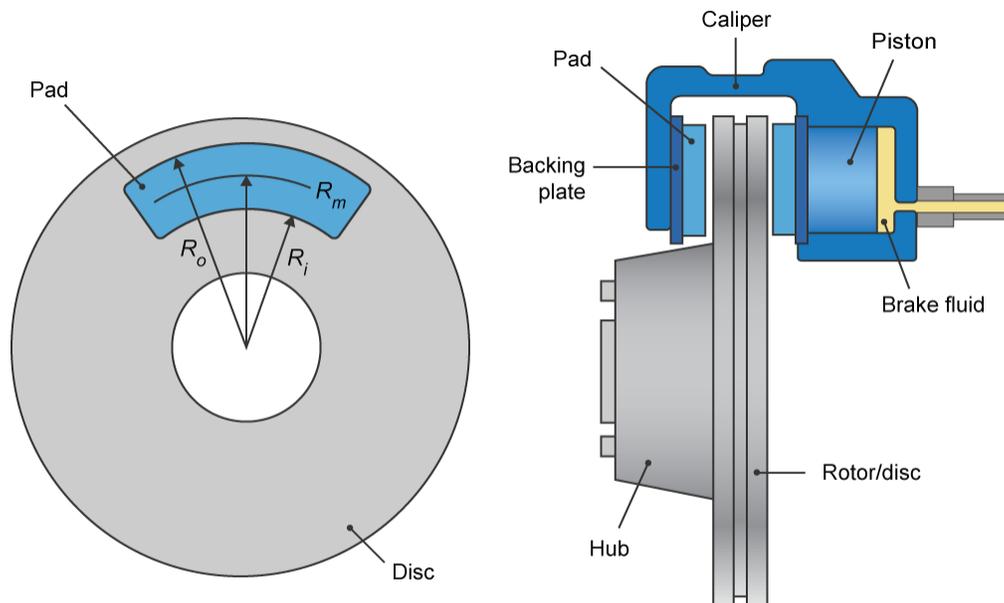
A disc brake converts brake cylinder pressure from the brake cylinder into force. The disc brake applies the force at the brake pad mean radius.

The equation that the block uses to calculate brake torque, depends on the wheel speed, Ω, such that when $\Omega \neq 0$,

$$T = \frac{\mu_k P \pi D_b^2 R_m N}{4}.$$

However when $\Omega = 0$, the torque applied by the brake is equal to the torque that is applied externally for wheel rotation. The maximum value of the torque that the brake can apply when $\Omega = 0$, is

$$T = \frac{\mu_s P \pi D_b^2 R_m N}{4}.$$

In both cases, $Rm = \dfrac{Ro + Ri}{2}$.

Where:

- T is the brake torque.
- P is the applied brake pressure.
- Ω is the wheel speed.
- N is the number of brake pads in disc brake assembly.
- μ_s is the disc pad-rotor coefficient of static friction.
- μ_k is the disc pad-rotor coefficient of kinetic friction.
- D_b is the brake actuator bore diameter.
- R_m is the mean radius of brake pad force application on brake rotor.
- R_o is the outer radius of brake pad.
- R_i is the inner radius of brake pad.

The block default models a dry brake. You can model fluid friction in a wet brake by setting the **Viscous friction coefficient**, k_v, to a nonzero value. The torque on the wheel in a wet brake system is:

$$T_{wet} = T + k_v \Omega.$$

Faulty Behavior

When faults are enabled, a brake pressure is applied in response to one or both of these triggers:

- Simulation time — Faulting occurs at a specified time.
- Simulation behavior — Faulting occurs in response to an external trigger. This exposes port **T**.

If a fault trigger occurs, the input pressure is replaced by the **Brake pressure when faulted** value for the remainder of the simulation. A value of 0 implies that no braking will occur. A relatively large value implies that the brake is stuck.

You can set the block to issue a fault report as a warning or error message in the Simulink Diagnostic Viewer with the **Reporting when fault occurs** parameter.

Thermal Model

You can model the effects of heat flow and temperature change by exposing the optional thermal port. To expose the port, in the **Friction** settings, set the **Thermal Port** parameter to `Model`. Exposing the port also exposes or changes the default value for these related settings, parameters, and variables:

- **Friction > Temperature**
- **Friction > Static friction coefficient vector**
- **Friction > Coulomb friction coefficient vector**
- **Thermal Port > Thermal mass**
- **Variables > Temperature**

Variables

Use the **Variables** settings to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Dependencies

Variable settings are visible only when, in the **Friction** settings, the **Thermal port** parameter is set to `Model`.

Ports

Input

P — Cylinder pressure, bars
physical signal

Physical signal port associated with cylinder pressure.

T — External fault trigger
physical signal

Physical signal port for an external fault trigger. Triggering occurs when the value is greater than 0.5. There is no unit associated with the trigger value.

Dependencies

This port is visible when **Enable faults** is set to `On` and **Enable external fault trigger** is set to `On`.

Conserving

S — Shaft
rotational mechanical

Rotational mechanical conserving port associated with the shaft.

H — Heat
thermal

Thermal conserving port associated with heat flow.

Dependencies

This port is visible only when, in the **Friction** settings, the **Thermal Port** parameter is set to `Model`.

Exposing this port makes related parameters and variables visible.

Parameters

Geometry

Mean pad radius — Pad radius
150 mm (default) | positive scaler

Mean radius of the friction pads.

Cylinder bore — Bore diameter
10 mm (default) | positive scaler

Diameter of the piston.

Number of brake pads — Pad quantity
2 (default) | positive scalar integer

Number of friction pads.

Friction

Thermal port — Thermal model
Omit (default) | Model

Model for heat flow and temperature change:

- `Omit` — Neglect thermal dynamics.
- `Model` — Include thermal dynamics.

Dependencies

When this parameter is set to `Model`, the thermal port and related parameter and variables are visible.

Temperature — Temperature
[280, 300, 320] K (default) | increasing vector

Array of temperatures used to construct a 1-D temperature-efficiency lookup table. The array values must increase left to right.

Dependencies

This parameter is only visible when the **Thermal Port** parameter is set to `Model`.

Static friction coefficient — Static friction
0.9 (default) | scalar

Coefficient of static friction. The value that you specify for this parameter must be greater than the value that you specify for the **Coulomb friction coefficient** parameter.

Dependencies

This parameter is only visible when the **Thermal Port** parameter is set to `Omit`.

Static friction coefficient vector — Static friction
[.9, .8, .7] | vector

Coefficient of static friction, such that:

- The number of elements in the vector must be the same as the number of elements in the specified vector for the **Temperature** parameter
- The values must be greater than the corresponding value that you specify for the **Coulomb friction coefficient vector** parameter.

Dependencies

This parameter is only visible when the **Thermal Port** parameter is set to `Model`.

Coulomb friction coefficient — Contact friction
0.7 (default) | positive scalar

Coulomb friction coefficient at the belt-drum contact surface. The value must be greater than zero.

Dependencies

This parameter is only visible when the **Thermal Port** parameter is set to `Omit`.

Coulomb friction coefficient vector — Contact friction
[.8, .7, .6] (default) | increasing vector of positive values

Coulomb friction coefficient at the belt-drum contact surface, such that:

- The number of elements in the vector must be the same as the number of elements in the specified vector for the **Temperature** parameter
- The values increase left to right.
- Each value must be greater than zero.

Dependencies

This parameter is only visible when the **Thermal Port** parameter is set to `Model`.

Breakaway friction velocity — Friction threshold
0.1 rad/s (default) | scalar

Angular speed at which friction switches from static to kinetic.

Viscous friction coefficient — Viscous friction
0 N*m/(rad/s) (default) | nonnegative scalar

For a wet brake, the viscous friction represents the energy loss to the cooling/lubricating fluid between the clutch plates. To model a wet brake, specify a nonzero value for the coefficient of viscous friction. The default value represents a dry brake.

Faults

Enable faults — Fault option
Off (default) | On

Enable externally or temporally triggered faults. When faulting occurs, the brake pressure normally received at port **P** will be set to the value specified in the **Brake pressure when faulted** parameter.

Brake pressure when faulted — Set faulted brake pressure
0 N (default) | positive scalar

Set faulted brake pressure. When faulting occurs, the brake pressure normally received at port **P** will be set to the value specified in the **Brake pressure when faulted** parameter. A value of 0 implies that braking does not occur. A relatively large value implies that the brake is stuck.

Dependencies

To enable this parameter, set **Enable faults** to On.

Enable external fault trigger — External trigger option
Off (default) | On

Enables port **T**. A physical signal at port **T** that is greater than 0.5 triggers faulting.

Dependencies

To enable this parameter, set **Enable faults** to On.

Enable temporal fault trigger — Temporal trigger option
Off (default) | On

Enables fault triggering at a specified time. When the **Simulation time for fault event** is reached, the brake pressure normally received at port **P** will be set to the value specified in the **Brake pressure when faulted** parameter.

Dependencies

To enable this parameter, set **Enable faults** to On.

Simulation time for fault event — Time at which faulting is triggered
5 s (default) | positive scalar

When the **Simulation time for fault event** is reached, the brake pressure normally received at port **P** will be set to the value specified in the **Brake pressure when faulted** parameter.

Dependencies

To enable this parameter, set **Enable faults** to On and **Enable temporal fault trigger** to On.

Reporting when fault occurs — Fault condition report
None (default) | Warning | Error

Reporting preference for the fault condition. When reporting is set to Warning or Error, a message is displayed in the Simulink Diagnostic Viewer. When Error is selected, the simulation will stop if faulting occurs.

Dependencies

To enable this parameter, set **Enable faults** to On.

Thermal Port

Thermal mass — Resistance to temperature change
50 kJ/K (default) | scalar

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

This parameter is only visible when in the **Friction** settings, the **Thermal Port** parameter is set to Model.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Band Brake | Double-Shoe Brake | Loaded-Contact Rotational Friction | Rotational Detent

Introduced in R2017b

Disk Friction Clutch

Friction clutch with disk plates that engage when plate pressure exceeds threshold
Library: Simscape / Driveline / Clutches

Description

The Disk Friction Clutch block represents a friction clutch with two flat friction plate sets that come into contact to engage. The clutch engages when the applied plate pressure exceeds an engagement threshold pressure. Once engaged, the plates experience frictional torques that enable them to transmit power between the base and follower driveshafts.

The clutch can be bidirectional or unidirectional. A bidirectional clutch can slip in the positive and negative directions. A unidirectional clutch can slip only in the positive direction. The slip direction is positive if the follower shaft spins faster than the base shaft and negative if it slips slower. The block defines the slip velocity as the difference

$$\omega = \omega_F - \omega_B,$$

where:

- ω is the relative angular velocity or slip velocity.
- ω_F is the angular velocity of the follower driveshaft.
- ω_B is the angular velocity of the base driveshaft.

The block provides a physical signal input port **P** for the applied pressure between the clutch plates. The applied pressure must be greater than or equal to zero and has units of Pascals. If the input signal falls below zero, the block treats the plate pressure as zero.

You can also enable faulting. When faulting occurs, the clutch will remain locked or will be unable to transmit power. Faults can occur at a specified time or due to an external trigger at port **T**.

Equations

The Disk Friction Clutch block is a simplified implementation of the Fundamental Friction Clutch block. The Fundamental Friction Clutch requires the kinetic and static friction limit torques as input signals. The Disk Friction Clutch does not require the input data. Instead, the block calculates the kinetic and static friction from the clutch parameters and the input pressure signal P.

When you apply a pressure signal above threshold, such that the applied pressure equals or exceeds the pressure threshold, that is, $P \geq P_{th}$, the block can apply two kinds of friction to the driveline motion, kinetic and static. The clutch applies kinetic friction torque only when one driveline axis is spinning relative to the other driveline axis. The clutch applies static friction torque when the two driveline axes lock and spin together. The block iterates through multistep testing to determine when to lock and unlock the clutch.

Kinetic Friction

The kinetic friction torque opposes the relative slip and is applied with an overall minus sign. Mathematically, the kinetic friction is the positive sum of viscous drag and surface contact friction torques:

$$\tau_k = \mu \cdot \omega + \tau_{contact}.$$

- τ_K is the kinetic friction torque.
- μ is the viscous drag coefficient.
- ω is the relative angular velocity, or slip velocity.
- $\tau_{contact}$ is the contact torque.

The contact friction is a product of six factors such that

$$\tau_{contact} = k_K \cdot D \cdot N \cdot r_{eff} \cdot P_{fric} \cdot A \geq 0,$$

where:

- k_K is the dimensionless coefficient of kinetic friction of clutch discs, which is a function of ω.
- D is the clutch de-rating factor.
- N is the number of friction surfaces.
- r_{eff} is the effective torque radius, that is, the effective moment arm of clutch friction force.
- P_{fric} is the clutch friction capacity, such that $P_{fric} = \max[(P - P_{th}), 0]$.
- A is the engagement surface area.

You specify the *kinetic friction coefficient*, k_K, as either a constant or a tabulated discrete function of relative angular velocity ω. The tabulated function is assumed to be symmetric for positive and negative values of the relative angular velocity. Therefore, specify k_K for positive values of ω only.

The clutch applies a normal force from its piston as the product of the clutch friction capacity, P_{fric}, and engagement surface area, A, on each of N friction surfaces. The pressure signal, P, should be nonnegative. If P is less than the pressure threshold. P_{th}, the clutch applies no friction at all.

The *effective torque radius*, r_{eff}, is the effective moment arm of clutch friction force, measured from the driveline axis, at which the kinetic friction forces are applied at the frictional surfaces. It is related to the geometry of the friction surface by:

$$r_{\text{eff}} = \frac{2}{3} \frac{r_0{}^3 - r_i{}^3}{r_0{}^2 - r_i{}^2},$$

where, for a friction surface, modeled as an annular disk:

- r_o is the outer disk radius.
- r_i is the inner disk radius.

The *clutch de-rating factor*, D, accounts for clutch wear. For a new clutch, D is one. For a clutch approaching a *uniform wear* state:

$$D \rightarrow \frac{3}{4} \frac{(r_o + r_i)^2}{r_o{}^2 + r_o r_i + r_i{}^2}.$$

Static Friction

The static friction limit is related to the kinetic friction, setting ω to zero and replacing the kinetic with the static friction coefficient:

$$\tau_S = k_S \cdot D \cdot N \cdot r_{eff} \cdot P_{fric} \cdot A \geq 0.$$

where:

- τ_S is the static friction torque limit, which is the product of the static friction peak factor and the kinetic friction torque as ω approaches 0.
- k_K is the dimensionless coefficient of kinetic friction of clutch discs, which is a function of ω.
- D is the clutch de-rating factor.
- N is the number of friction surfaces.
- r_{eff} is the effective torque radius, that is, the effective moment arm of clutch friction force.
- P_{fric} is the clutch friction capacity, such that $P_{fric} = \max[(P - P_{th}), 0]$.
- A is the engagement surface area.

$k_S > k_K$, so that the torque τ needed across the clutch to unlock it by overcoming static friction is larger than the kinetic friction at the instant of unlocking, when $\omega = 0$..

The *static friction torque range* or limits are then defined symmetrically as

$$\tau_S \equiv \tau_S^+ = -\tau_S^- .$$

Wait State: Locking and Unlocking

The Wait state of the Disk Friction Clutch is identical to the Wait state of the Fundamental Friction Clutch, with the replacement of the positive kinetic friction condition, $\tau_K > 0$, by the positive clutch friction capacity condition, the applied pressure equals or exceeds the pressure threshold, that is, $P \geq P_{th}$.

Power Dissipated by the Clutch

The power dissipated by the clutch is the absolute value of the product of slip velocity, ω, and the kinetic friction torque, τ_K, that is, $|\omega \cdot \tau_K|$. The clutch dissipates power only if it is both slipping, $\omega \neq 0$, and applying kinetic friction, $\tau_k > 0$.

Velocity-Dependent and Temperature-Dependent Friction Models

Velocity-Dependent Model

You can model the effects of rotational velocity change by selecting a velocity-dependent model. To choose a velocity-dependent model, in the **Friction** settings, set the **Friction model** parameter to `Velocity-dependent kinetic friction coefficient`. For information about a friction model that depends on both velocity and temperature, see "Thermal, Velocity-Dependent Model" on page 1-113.

For the velocity-dependent model these related parameters become visible in the **Friction** settings:

- **Relative velocity vector**
- **Kinetic friction coefficient vector**

- **Friction coefficient interpolation method**
- **Friction coefficient extrapolation method**

Thermal Model

You can model the effects of heat flow and temperature change by selecting a temperature-dependent model. To choose a temperature-dependent model, in the **Friction** settings, set the **Friction model** parameter to `Temperature-dependent friction coefficients`. For information about a friction model that depends on both velocity and temperature, see "Thermal, Velocity-Dependent Model" on page 1-113.

For the temperature-dependent model, thermal port **H** and these parameters are visible:

- In the **Friction** settings:

  - **Temperature vector**
  - **Static friction coefficient vector**
  - **Kinetic friction coefficient vector**
  - **Friction coefficient interpolation method**
  - **Friction coefficient extrapolation method**
- In the **Thermal Port** settings:

  - **Thermal mass**
  - **Initial Temperature**

Thermal, Velocity-Dependent Model

You can model the effects of rotational velocity change and heat flow by selecting a velocity-dependent and temperature-dependent model. To choose a model that depends on both velocity and temperature, in the **Friction** settings, set the **Friction model** parameter to `Temperature and velocity-dependent friction coefficients`.

For the velocity-dependent and temperature-dependent model, thermal port **H** and these related settings and parameters become visible:

- In the **Friction** settings:

  - **Relative velocity vector**
  - **Temperature vector**
  - **Static friction coefficient vector**
  - **Kinetic friction coefficient matrix**
  - **Friction coefficient interpolation method**
  - **Friction coefficient extrapolation method**
- In the **Thermal Port** settings:

  - **Thermal mass**
  - **Initial Temperature**

Faulty Behavior

You can enable faulty behavior in response to:

- Simulation time — Faulting occurs at a specified time.
- Simulation behavior — Faulting occurs in response to an external trigger. This exposes port **T**.

You can choose either or both of these settings for block faulting. If faulting is triggered, the clutch responds according to the **Behavior when faulted** setting for the remainder of the simulation. The fault options are:

- `Cannot transmit power`
- `Cannot unlock`

You can set the block to issue a fault report as a warning or error message in the Simulink Diagnostic Viewer with the **Reporting when fault occurs** parameter.

Ports

Input

P — Pressure
physical signal

Physical signal input port for the applied pressure between the clutch plates. This signal is positive or zero. A signal of less than zero is interpreted as zero.

T — External fault trigger
physical signal

Physical signal port for an external fault trigger. Triggering occurs when the value is greater than 0.5. There is no unit associated with the trigger value.

Dependencies

This port is visible when **Enable faults** is set to `On` and **Enable external fault trigger** is set to `On`.

Conserving

B — Base
mechanical rotational

Rotational conserving port associated with the driving, or base, shaft.

F — Follower
mechanical rotational

Rotational conserving port associated with the driven, or follower, shaft.

H — Heat flow
thermal

Thermal conserving port associated with heat flow.

Dependencies

This port is visible only when, in the **Friction** settings, the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-`

dependent friction coefficients. For more information, see Friction Parameter Dependencies and "Friction model" on page 1-0 .

Parameters

Geometry

Geometry model — Geometry parameterization method
Define effective radius (default) | Define annular region

Parameterization method for modeling the clutch friction geometry.

Dependencies

The **Effective torque radius** parameter is visible only If this parameter is set to Define effective radius:

These parameters are visible only If this parameter is set to Define annular region:

- **Friction surface outside diameter**
- **Friction surface inside diameter**

Effective torque radius — Moment arm radius
130mm (default) | positive scalar

Effective moment arm radius, r_{eff}, that determines the kinetic friction torque inside the clutch.

Dependencies

This parameter is visible only if the **Geometry model** parameter is set to Define effective radius.

Friction surface outside diameter — Friction surface outer diameter
150 mm (default) | positive scalar

Diameter, $2r_o$, across the outer edge of the friction disk annulus.

Dependencies

This parameter is visible only if the **Geometry model** parameter is set to Define annular region.

Friction surface inside diameter — Friction surface inner diameter
100 mm (default) | nonnegative scalar

Diameter, $2r_i$, across the inner edge of the friction disk annulus.

Dependencies

This parameter is visible only if the **Geometry model** parameter is set to Define annular region.

Number of friction surfaces — number of friction surfaces
4 (default) | positive scalar integer

Number, N, of friction-generating contact surfaces inside the clutch.

Engagement piston area — Pressure application area
0.001 m^2 (default) | positive scalar

Effective area, *A*, of the clutch piston when the piston is applying pressure across the clutch.

Directionality — Allowable slip directions
Bidirectional (default) | Unidirectional

Slip directions the clutch allows between its plates. A bidirectional clutch allows positive and negative slip velocities. A unidirectional clutch allows only positive slip velocities.

The unidirectional clutch is equivalent to a friction clutch connected in parallel to a one-way clutch that disengages only when the slip velocity becomes positive. To model a unidirectional clutch with slip in the negative direction, reverse the base and follower port connections.

Friction

The table shows how the visibility of some ports, parameters, and settings depends on the option that you choose for other parameters. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Friction Parameter Dependencies

Friction			
Friction model			
Fixed kinetic friction coefficient	Velocity-dependent kinetic friction coefficient	Temperature-dependent friction coefficients	Temperature and velocity-dependent friction coefficients
		Exposes: • Conserving port **H** • **Thermal Port** settings	Exposes: • Conserving port **H** • **Thermal Port** settings
-	**Relative velocity vector**	-	**Relative velocity vector**
-	-	**Temperature vector**	**Temperature vector**
Static friction coefficient	**Static friction coefficient**	**Static friction coefficient vector**	**Static friction coefficient vector**
Kinetic friction coefficient	**Kinetic friction coefficient vector**	**Kinetic friction coefficient vector**	**Kinetic friction coefficient matrix**
-	**Friction coefficient interpolation method**	**Friction coefficient interpolation method**	**Friction coefficient interpolation method**
-	**Friction coefficient extrapolation method**	**Friction coefficient extrapolation method**	**Friction coefficient extrapolation method**
De-rating factor	**De-rating factor**	**De-rating factor**	**De-rating factor**
Clutch velocity tolerance	**Clutch velocity tolerance**	**Clutch velocity tolerance**	**Clutch velocity tolerance**
Engagement threshold pressure	**Engagement threshold pressure**	**Engagement threshold pressure**	**Engagement threshold pressure**

Friction model — Friction model

Fixed kinetic friction coefficient (default) | Velocity-dependent kinetic friction coefficient | Temperature-dependent friction coefficients | Temperature and velocity-dependent friction coefficients

Parameterization method to model the kinetic friction coefficient. The options and default values for this parameter depend on the friction model that you select for the block. The options are:

- Fixed kinetic friction coefficient — Provide a fixed value for the kinetic friction coefficient.

- Velocity-dependent kinetic friction coefficient — Define the kinetic friction coefficient by one-dimensional table lookup based on the relative angular velocity between disks.

- Temperature-dependent friction coefficients — Define the kinetic friction coefficient by table lookup based on the temperature.

- `Temperature and velocity-dependent friction coefficients` — Define the kinetic friction coefficient by table lookup based on the temperature and the relative angular velocity between disks.

Dependencies

The friction model setting affects the visibility of other parameters, settings, and ports. For more information, see Friction Parameter Dependencies.

Relative velocity vector — Relative velocity
`[0, 100, 1000]` rad/s (default) | vector

Input values for the relative velocity as a vector. The values in the vector must increase from left to right. The minimum number of values depends on the interpolation method that you select. For linear interpolation, provide at least two values per dimension. For smooth interpolation, provide at least three values per dimension.

Dependencies

This parameter is visible only if the **Friction model** parameter is set to `Velocity-dependent kinetic friction coefficient` or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Temperature vector — Temperature
`[280, 300, 320]` K (default) | increasing vector

Input values for the temperature as a vector. The minimum number of values depends on the interpolation method that you select. For linear interpolation, provide at least two values per dimension. For smooth interpolation, provide at least three values per dimension. The values in the vector must increase from left to right.

Dependencies

This parameter is visible only if the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Static friction coefficient — Static friction coefficient
`0.35` (default) | positive scalar

Static or peak value of the friction coefficient. The static friction coefficient must be greater than the kinetic friction coefficient.

Dependencies

This parameter is visible only when the **Friction model** parameter is set to `Fixed kinetic friction coefficient` or `Velocity-dependent kinetic friction coefficient`. For more information, see Friction Parameter Dependencies.

Static friction coefficient vector — Static friction coefficient
`[.4, .38, .36]` (default) | vector

Static, or peak, values of the friction coefficient as a vector. The vector must have the same number of elements as the temperature vector. Each value must be greater than the value of the corresponding element in the kinetic friction coefficient vector.

Dependencies

This parameter is visible only if the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Kinetic friction coefficient — Kinetic friction coefficient
`0.3` (default) | positive scalar

The kinetic, or Coulomb, friction coefficient. The coefficient must be greater than zero.

Dependencies

This parameter is visible only if the **Friction model** parameter is set to `Fixed kinetic friction coefficient`. For more information, see Friction Parameter Dependencies.

Kinetic friction coefficient vector — Kinetic friction coefficient
`[.3, .22, .19]` (default) | `[.3, .28, .25]` | positive vector

Output values for kinetic friction coefficient as a vector. All values must be greater than zero.

If the **Friction model** parameter is set to

- `Velocity-dependent kinetic friction coefficient` — The vector must have same number of elements as relative velocity vector.
- `Temperature-dependent friction coefficients` — The vector must have the same number of elements as the temperature vector.

Dependencies

This parameter is visible only if the **Friction model** parameter is set to `Velocity-dependent kinetic friction coefficient` or `Temperature-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Kinetic friction coefficient matrix — Kinetic friction coefficient
`[.34, .32, .3; .3, .28, .25; .25, .2, .15]` (default) | positive matrix

Output values for kinetic friction coefficient as a matrix. All the values must be greater than zero. The size of the matrix must equal the size of the matrix that is the result of the temperature vector × the kinetic friction coefficient relative velocity vector.

Dependencies

This parameter is visible only if the **Friction model** parameter is set to `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Friction coefficient interpolation method — Extrapolation method
`Linear` (default) | `Smooth`

Interpolation method for approximating the output value when the input value is between two consecutive grid points:

- `Linear` — Select this option to get the best performance.
- `Smooth` — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is visible only if the **Friction model** parameter is set to `Velocity-dependent kinetic friction coefficient`, `Temperature-dependent friction coefficients`, or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Friction coefficient extrapolation method — Interpolation method
Linear (default) | Nearest | Error

Extrapolation method for determining the output value when the input value is outside the range specified in the argument list:

- `Linear` — Select this option to produce a curve with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region.

- `Nearest` — Select this option to produce an extrapolation that does not go above the highest point in the data or below the lowest point in the data.

- `Error` — Select this option to avoid going into the extrapolation mode when you want your data to be within the table range. If the input signal is outside the range of the table, the simulation stops and generates an error.

For more information on extrapolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is visible only if the **Friction model** parameter is set to `Velocity-dependent kinetic friction coefficient`, `Temperature-dependent friction coefficients`, or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

De-rating factor — De-rating factod
1 (default) | positive scalar

Dimensionless de-rating factor, D, that accounts for clutch disk wear by proportionately reducing clutch friction.

Clutch velocity tolerance — Relative velocity locking threshold
0.001 rad/s (default) | nonnegative scalar

Maximum slip velocity at which the clutch can lock. The slip velocity is the signed difference between the base and follower shaft angular velocities, that is, $w = w_F - w_B$. If the kinetic friction torque is nonzero and the transferred torque is within the static friction torque limits, then the clutch locks if the actual slip velocity falls below the velocity tolerance.

Engagement threshold pressure — Normal contact force threshold
100 Pa (default) | nonnegative scalar

Minimum pressure P_{th} at which the clutch engages. If the pressure input signal falls below this threshold, the clutch automatically disengages.

Viscous Losses

Viscous drag torque coefficient — Viscous drag torque coefficient
`0 N*m/(rad/s)` (default) | nonnegative scalar

Viscous friction coefficient μ applied to the relative slip ω between the base and follower axes.

Initial Conditions

Initial state — Initial clutch state
`Unlocked` (default) | `Locked`

Clutch state at the start of simulation. The clutch can be in one of two states, locked and unlocked. A locked clutch constrains the base and follower shafts to spin at the same velocity, that is, as a single unit. An unlocked clutch allows the two shafts to spin at different velocities, resulting in slip between the clutch plates.

Faults

Enable faults — Fault option
`Off` (default) | `On`

Enable externally or temporally triggered faults. When faulting occurs, the clutch fails to unlock or cannot transmit power, according to the **Behavior when faulted** setting.

Behavior when faulted — Set fault response
`Cannot transmit power` (default) | `Cannot unlock`

Set fault response. You can select the clutch faulting as either:

- `Cannot transmit power`
- `Cannot unlock`

Dependencies

To enable this parameter, set **Enable faults** to `On`.

Enable external fault trigger — External trigger option
`Off` (default) | `On`

Enables port **T**. A physical signal at port **T** that is greater than `0.5` triggers faulting.

Dependencies

To enable this parameter, set **Enable faults** to `On`.

Enable temporal fault trigger — Temporal trigger option
`Off` (default) | `On`

Enables fault triggering at a specified time. When the **Simulation time for fault event** is reached, the clutch responds according to the **Behavior when faulted** setting.

Dependencies

To enable this parameter, set **Enable faults** to `On`.

Simulation time for fault event — Time at which faulting is triggered
5 s (default) | positive scalar

When the **Simulation time for fault event** is reached, the clutch responds according to the **Behavior when faulted** setting.

Dependencies

To enable this parameter, set **Enable faults** to On and **Enable temporal fault trigger** to On.

Reporting when fault occurs — Fault condition report
None (default) | Warning | Error

Reporting preference for the fault condition. When reporting is set to Warning or Error, a message is displayed in the Simulink Diagnostic Viewer. When Error is selected, the simulation will stop if faulting occurs.

Dependencies

To enable this parameter, set **Enable faults** to On.

Thermal Port

Thermal Port settings are visible only when, in the **Friction** settings, the **Friction model** parameter is set to Temperature-dependent friction coefficients or Temperature and velocity-dependent friction coefficients. For more information, see Friction Parameter Dependencies.

Thermal mass — Thermal mass
50 kJ/K (default) | scalar

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

This parameter is visible only when, in the **Friction** settings, the **Friction model** parameter is set to Temperature-dependent friction coefficients or Temperature and velocity-dependent friction coefficients. For more information, see Friction Parameter Dependencies.

Initial temperature — Initial temperature
300 K (default) | scalar

Component temperature at the start of simulation. The initial temperature alters the component efficiency according to an efficiency vector that you specify, affecting the starting meshing or friction losses. The default value is 300 K.

Dependencies

This parameter is only visible when, in the **Friction** settings, the **Friction model** parameter is set to Temperature-dependent friction coefficients or Temperature and velocity-dependent friction coefficients. For more information, see Friction Parameter Dependencies.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Differential | Fundamental Friction Clutch | Variable Ratio Transmission

Topics
"Gear Coupling Control Using Clutches"
"Clutches For Accelerating and Braking"
"CR-CR Four-Speed Transmission"
"Custom Clutch"
"Hydraulically-Actuated Driveline Clutch"
"Two-Speed Transmission"
"Vehicle with Dual Clutch Transmission"
Vehicle with Four-Speed Transmission

Introduced in R2011a

Dog Clutch

Clutch with toothed plates that engage when plate teeth become enmeshed
Library: Simscape / Driveline / Clutches
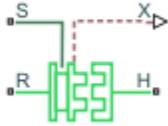
Description

This block represents a nonslip clutch, a mechanical device that relies on the positive engagement of interlocking teeth to transfer torque between driveline shafts. The clutch contains three key components:

- Ring
- Hub
- Shift linkage

The ring and the hub are toothed components. The ring spins with the output shaft, sliding along its longitudinal axis to engage or disengage the coaxial hub. The hub, which sits on a bearing encircling the same shaft, can spin independently until engaged.

Engagement occurs when the toothed components interlock. Once engaged, the ring and the hub spin together as a unit. To control engagement, the dog clutch contains a shift linkage that governs the position of the ring with respect to the hub.

Moving the ring towards the hub so that their teeth interlock changes the clutch state to engaged. Tooth overlap must exceed a minimum value for engagement. Moving the ring in reverse so that the two sets of teeth no longer interlock changes the clutch state back to disengaged.

Port **S** specifies the shift linkage position. When the clutch is fully disengaged, the shift linkage position is zero. When the clutch is fully engaged, the shift linkage position equals the sum of the tooth height and the ring-hub clearance of the fully disengaged state:

$$z = h + z_{Gap},$$

where:

- z is the shift linkage position.
- h is the tooth height.
- z_{Gap} is the ring-hub clearance when disengaged.

The figure shows side and front views of the dog clutch and some of its relevant variables.

T — Shift linkage translation

R — Mean clutch teeth radius
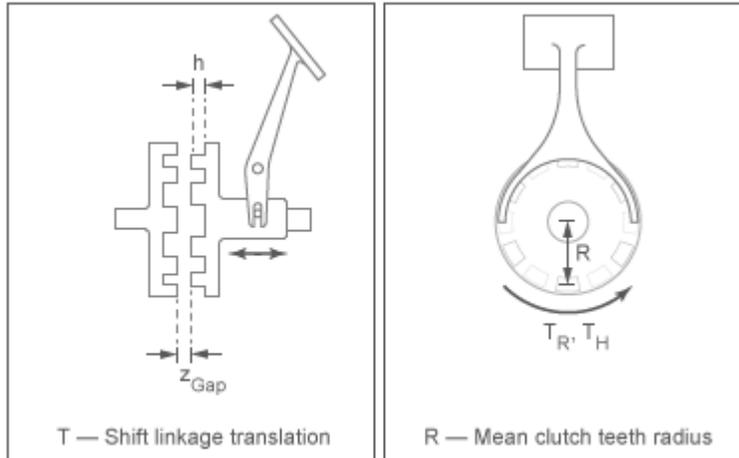
Torque Transmission Models

The Dog Clutch block provides a choice of two torque transmission models.

Friction Clutch Approximate Model

Treat clutch engagement as a friction phenomenon between the ring and the hub. This model ignores special effects such as backlash, an approximation that makes the block better suited for linearization, fixed-step simulation, and hardware-in-loop (HIL) simulation. The Fundamental Friction Clutch block provides the foundation for the model.

In the friction approximate model, the clutch has three possible configurations: disengaged, engaged, and locked. When disengaged, the contact force between the ring and the hub is zero. This force remains zero until the shift linkage reaches the minimum position for engagement.

When the ring-hub tooth overlap, h, exceeds the minimum value for engagement, the contact force between the two components begins to increase linearly with the shift linkage position, z.

At full engagement, the contact force reaches its maximum value, and the clutch state switched to locked. In this state, the ring and the hub spin as a unit without slip. To unlock the clutch, the transmitted torque must exceed the maximum allowed value that you specify.

Dynamic Model with Backlash

Capture clutch phenomena such as backlash, torsional compliance, and contact forces between ring and hub teeth. This model provides greater accuracy than the friction clutch approximation.

In the dynamic model, the clutch has two possible configurations: disengaged and engaged. When disengaged, the contact force between the ring and the hub is zero. This force remains zero until the shift linkage reaches the minimum position for engagement.

When the ring-hub tooth overlap, h, exceeds the minimum value for engagement, a contact force kicks in between the two components. This force is the sum of torsional spring and damper components. Including backlash between the ring and hub teeth:

1-125

$$T_C = \begin{cases} -k_{RH}\left(\phi - \dfrac{\delta}{2}\right) - \mu_R \cdot \omega & \phi > \dfrac{\delta}{2} \\ 0 & -\dfrac{\delta}{2} < \phi < \dfrac{\delta}{2}, \\ -k_{RH}\left(\phi + \dfrac{\delta}{2}\right) - \mu_R \omega & \phi < -\dfrac{\delta}{2} \end{cases}$$

where:

- k_{RH} is the torsional stiffness of the ring-hub coupling.
- ϕ is the relative angle, about the common rotation axis, between the ring and the hub.
- δ is the backlash between ring and hub teeth.
- ω is the relative angular velocity between the ring and the hub. This variable describes how fast the two components slip past each other.

Compliant end stops limit the translational motion of the clutch shift linkage and the ring. The compliance model treats the end stops as linear spring-damper sets. The location of the end stops depends on the relative angle and angular velocity between the ring and hub teeth:

- If the teeth align and the relative angular velocity is smaller than the maximum value for clutch engagement, the end-stop location is the sum of the ring-hub clearance when fully disengaged and the tooth height. With the end stop at this location, the clutch can engage.
- If the teeth do not align or the relative angular velocity exceeds the maximum value for clutch engagement, the end-stop location is set to prevent the ring from engaging the hub. The clutch remains disengaged.

Translational friction opposes shift linkage and ring motion. This friction is the sum of Coulomb and viscous components:

$$F_Z = -k_K \cdot F_N \cdot \tanh\left(\frac{4v}{v_{th}}\right) - \mu_T v,$$

where:

- F_Z is the net translational friction force acting on the shift linkage and ring.
- k_K is the kinetic friction coefficient between ring and hub teeth.
- F_N is the normal force between ring and hub teeth.
- v is the translational velocity of the shift linkage and the ring.
- v_{th} is the translational velocity threshold. Below this threshold, a hyperbolic tangent function smooths the Coulomb friction force to zero as the shift linkage and ring velocity tends to zero.
- μ_T is the viscous damping coefficient acting on the shift linkage and the ring.

Clutch Engagement Conditions

The clutch engages when it satisfies a set of geometrical and dynamic conditions. These conditions specify the values that certain variables can take for clutch engagement to occur:

- The minimum position at which the ring and the hub can engage is

$$z = h_0 + z_{Gap},$$

where h_0 is the minimum tooth overlap for clutch engagement. Adjust this parameter to minimize engagement instability, that is, the tendency of the clutch to switch rapidly between engaged and disengaged states

- The magnitude of the relative angular velocity between the ring and the hub is smaller than the maximum engagement velocity, that is:

$$|\omega| < |\omega_{\text{max}}|,$$

where ω_{max} is the maximum value of the relative angular velocity at which engagement can occur.

- If using the friction clutch approximate model, engagement occurs only if torque transfer between the ring and the hub remains smaller than the maximum transmitted torque that the clutch supports.

- If using the dynamic model with backlash, engagement occurs only if the relative angular position of the ring and hub teeth allows them to interlock.

Rotational Power Dissipation

When the clutch slips under an applied torque, it dissipates power. The power loss equals the product of the slip angular velocity and the contact torque between the ring and the hub:

$$P_{loss} = \omega \cdot T_C,$$

where:

- P_{loss} is the dissipated power due to slipping.
- T_C is the kinetic contact torque.

Thermal Modeling

You can model the effects of heat flow and temperature change through an optional thermal conserving port. By default, the thermal port is hidden. To expose the thermal port, in the **Clutch** settings, set the **Thermal port** parameter to `Model`. Specify the associated thermal parameters for the component.

Ports

Input

S — Shift linkage
physical signal

Physical signal input port associated with the shift linkage.

Dependencies

If, in the **Shift Linkage** settings, the **Shift linkage control** parameter is set to:

- `Physical signal` — Input shift linkage position directly through the optional physical signal port, **S**.
- `Conserving port` — Input the shift linkage position dynamically through the optional translational conserving port, **S**. When you select this option, related **Shift Linkage** parameters are exposed.

Output

X — Clutch position
physical signal

Physical signal output port associated with clutch position.

Conserving

H — Clutch hub
mechanical rotational

Mechanical rotational conserving port associated with the clutch hub

R — Clutch ring
mechanical rotational

Mechanical rotational conserving port associated with the clutch ring.

S — Shift linkage
mechanical translational

Mechanical translational conserving port associated with shift linkage.

Dependencies

If, in the **Shift Linkage** settings, the **Shift linkage control** parameter is set to:

- `Physical signal` — Input shift linkage position directly through the optional physical signal port, **S**.
- `Conserving port` — Input the shift linkage position dynamically through the optional translational conserving port, **S**. When you select this option, related **Shift Linkage** parameters are exposed.

T — Heat flow
thermal

Thermal conserving port associated with heat flow.

Dependencies

This port is visible only when, in the **Clutch** settings, the **Thermal Port** parameter is set to `Model`.

Exposing this port makes related settings visible.

Parameters

The table shows how the specified options for parameters in both the **Shift Linkage** and **Clutch** settings affect the visibility of:

- Parameters in the **Clutch**, **Shift Linkage**, **Engagement Conditions**, and **Initial Conditions** settings
- **Thermal Port** settings

To learn how to read the table, see "Parameter Dependencies" on page B-2.

Dog Clutch Parameter Dependencies

Settings	Parameters and Options					
Shift Linkage	**Shift linkage control**					
	Conserving Port			Physical signal		
Clutch	**Thermal port**			**Thermal port**		
	Omit		Model	Omit		Model
	Torque transmission model		-	**Torque transmission model**		-
	Friction clutch approximation - Suitable for HIL and linearization	Dynamic with backlash	-	Friction clutch approximation - Suitable for HIL and linearization	Dynamic with backlash	-
	-	-	**Temperature vector**	-	-	**Temperature vector**
	Maximum transmitted torque	-	**Maximum transmitted torque vector**	**Maximum transmitted torque**	-	**Maximum transmitted torque vector**
	-	-	**Interpolation method**	-	-	**Interpolation method**
	-	-	**Extrapolation method**	-	-	**Extrapolation method**
	Clutch teeth mean radius	**Clutch teeth mean radius**	**Clutch teeth mean radius**	**Clutch teeth mean radius**	-	**Clutch teeth mean radius**
	-	**Number of teeth**	-	-	**Number of teeth**	-
	-	**Rotational backlash**	-	-	**Rotational backlash**	-
	-	**Torsional stiffness**	-	-	**Torsional stiffness**	-
	-	**Torsional damping**	-	-	**Torsional damping**	-
Shift Linkage	**Tooth height**	**Tooth height**	**Tooth height**	**Tooth height**	**Tooth height**	**Tooth height**
	Ring-hub clearance when disengaged	**Ring-hub clearance when disengaged**	**Ring-hub clearance when disengaged**	**Ring-hub clearance when disengaged**	**Ring-hub clearance when disengaged**	**Ring-hub clearance when disengaged**

Settings	Parameters and Options					
	Hard stop at back of shift linkage	Hard stop at back of shift linkage	Hard stop at back of shift linkage	-	-	-
	Ring stop stiffness	Ring stop stiffness	Ring stop stiffness	-	-	-
	Ring stop damping	Ring stop damping	Ring stop damping	-	-	-
	Shift linkage viscous friction coefficient	Shift linkage viscous friction coefficient	Shift linkage viscous friction coefficient	-	-	-
	-	Tooth-tooth friction coefficient	-	-	-	-
Engagement Conditions	Linkage travel direction	Linkage travel direction	Linkage travel direction	Linkage travel direction	Linkage travel direction	Linkage travel direction
	Maximum engagement speed	Maximum engagement speed	Maximum engagement speed	-	-	-
	Tooth overlap to engage	Tooth overlap to engage	Tooth overlap to engage	Tooth overlap to engage	Tooth overlap to engage	Tooth overlap to engage
Initial Conditions	Clutch initial state	Clutch initial state	Clutch initial state	Clutch initial state	Clutch initial state	Clutch initial state
	Initial shift linkage position	Initial shift linkage position	Initial shift linkage position	-	-	-
	-	Initial ring-hub offset angle	-	-	Initial ring-hub offset angle	-
Thermal Port	-	-	Thermal mass	-	-	Thermal mass
	-	-	Initial temperature	-	-	Initial temperature

Clutch

Thermal port — Thermal model
Omit (default) | Model

Model for heat flow and temperature change:

- `Omit` — Temperature-independent model.
- `Model` — Temperature-dependent model.

Dependencies

For the temperature-independent model, that is, if this parameter is set to `Omit`, the **Torque transmission model** parameter and other related parameters in the **Clutch** settings are visible.

For the temperature-dependent model, that is, when this parameter is set to `Model`, related parameters in the **Clutch** settings, thermal port **T**, and **Thermal Port** settings are visible.

Temperature vector — Temperature
[280, 300, 320] K (default) | increasing vector

Input values for the temperature as a vector. The minimum number of values depends on the interpolation method that you select. For linear interpolation, provide at least two values per dimension. For smooth interpolation, provide at least three values per dimension. The values in the vector must increase from left to right.

Dependencies

This parameter is visible only if, in the **Clutch** settings, the **Thermal port** parameter to `Model`.

Torque transmission model — Torque transmission model
Friction clutch approximation — Suitable for HIL and linearization (default) | Dynamic with backlash

Computational framework for modeling the dynamic behavior of the dog clutch:

- `Friction clutch approximation — Suitable for HIL and linearization` — Model clutch engagement as a friction phenomenon between the ring and the hub. This model, based on the Fundamental Friction Clutch block, provides a computationally efficient approximation of the dog clutch.
- `Dynamic with backlash` — Model clutch engagement in detail, accounting for such phenomena as backlash, torsional compliance, and contact forces between ring and hub teeth. Selecting this option causes additional **Shift Linkage** and **Initial Conditions** parameters to appear.

Dependencies

This parameter is visible only if, in the **Clutch** settings, the **Thermal port** parameter is set to `Omit`.

The visibility of related parameters in the **Clutch** and **Shift Linkage** settings is affected by the option that you select for this parameter.

Temperature vector — Temperature
[280, 300, 320] K (default) | increasing vector

Input values for the temperature as a vector. The minimum number of values depends on the interpolation method that you select. For linear interpolation, provide at least two values per dimension. For smooth interpolation, provide at least three values per dimension. The values in the vector must increase from left to right.

Dependencies

This parameter is visible only if, in the **Clutch** settings, the **Thermal port** parameter to `Model`.

Maximum transmitted torque — Maximum transmitted torque
1000 N*m (default) | positive scalar

Largest torque that the clutch can transmit, corresponding to a nonslip engaged configuration. If the torque transmitted between the ring and the hub exceeds this value, the two components begin to slip with respect to each other. This torque determines the static friction limit in the friction clutch approximation

Dependencies

This parameter is visible only if, in the **Clutch** settings, the **Thermal port** parameter is set to Omit and the **Torque transmission model** parameter is set to Friction clutch approximation - Suitable for HIL and linearization.

Maximum transmitted torque vector — Maximum transmitted torque
[1000, 1050, 1000] N*m (default) | vector

Largest torque that the clutch can transmit, corresponding to a nonslip engaged configuration, specified as a vector. If the torque transmitted between the ring and the hub exceeds this value, the two components begin to slip with respect to each other. This torque determines the static friction limit in the friction clutch approximation. The vector has the same number of elements as the temperature vector.

Dependencies

This parameter is visible only if, in the **Clutch** settings, the **Thermal port** parameter to Model.

Interpolation method — Interpolation method
Linear (default) | Smooth

Interpolation method for approximating the output value when the input value is between two consecutive grid points:

- Linear — Select this option to get the best performance.
- Smooth — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is visible only if, in the **Clutch** settings, the **Thermal port** parameter to Model.

Extrapolation method — Extrapolation method
Linear (default) | Nearest | Error

Extrapolation method for determining the output value when the input value is outside the range specified in the argument list:

- Linear — Select this option to produce a curve with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region.
- Nearest — Select this option to produce an extrapolation that does not go above the highest point in the data or below the lowest point in the data.

- `Error` — Select this option to avoid going into the extrapolation mode when you want your data to be within the table range. If the input signal is outside the range of the table, the simulation stops and generates an error.

For more information on extrapolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is visible only if, in the **Clutch** settings, the **Thermal port** parameter to `Model`.

`Clutch teeth mean radius` — Clutch teeth mean radius
`50 mm` (default) | positive scalar

Distance from the ring or hub center to the corresponding tooth center. The mean tooth radius determines the normal contact forces between ring and hub teeth given the transmission torque between the two components. The value must be greater than zero.

Dependencies

This parameter is not visible if all of these conditions are met:

- In the **Clutch** settings:

  - **Thermal port** is set to `Omit`.
  - **Torque transmission model** is set to `Dynamic with backlash`.
- In the **Shift Linkage** settings, the **Shift linkage control** parameter is set to `Physical control`.

`Number of teeth` — Number of ring or hub teeth
`6` (default) | positive nonzero scalar integer

Total number of teeth in the ring or the hub. The two components have equal tooth numbers. The value must be greater than or equal to one.

Dependencies

This parameter is visible only if, in the **Clutch** settings, the **Thermal port** parameter to `Omit` and the **Torque transmission model** parameter is set to `Dynamic with backlash`.

`Rotational backlash` — Rotational backlash
`10 deg` (default) | positive scalar

Allowable angular motion, or play, between the ring and hub teeth in the engaged clutch configuration. The value must be greater than zero.

Dependencies

This parameter is visible only if, in the **Clutch** settings, the **Thermal port** parameter to `Omit` and the **Torque transmission model** parameter is set to `Dynamic with backlash`.

`Torsional stiffness` — Torsional stiffness
`10e6 N*m/rad` (default) | positive scalar

Linear torsional stiffness coefficient at the contact interface between the ring and hub teeth. This coefficient characterizes the restoring component of the contact force between the two sets of teeth.

Greater stiffness values correspond to greater contact forces. The value must be greater than zero. The default value is `10e6 N*m/rad`.

Dependencies

This parameter is visible only if, in the **Clutch** settings, the **Thermal port** parameter to `Omit` and the **Torque transmission model** parameter is set to `Dynamic with backlash`.

`Torsional damping` — **Torsional damping**
`100 N*m/(rad/s)` (default) | positive scalar

Linear torsional damping coefficient at the contact interface between the ring and hub teeth. This coefficient characterizes the dissipative component of the contact force between the two sets of teeth. Greater damping values correspond to greater energy dissipation during contact. The value must be greater than zero.

Dependencies

This parameter is visible only if, in the **Clutch** settings, the **Thermal port** parameter to `Omit` and the **Torque transmission model** parameter is set to `Dynamic with backlash`.

Shift Linkage

`Shift linkage control` — **Shift linkage control port**
`Physical signal` (default) | `Conserving port`

Port type for shift linkage control:

- `Physical signal` — Input shift linkage position directly through the optional physical signal port, **S**.
- `Conserving port` — Input the shift linkage position dynamically through the optional translational conserving port, **S**.

Dependencies

This parameter determines whether the shift linkage control port, **S**, is a conserving or physical signal port. Related **Shift Linkage** parameters are exposed if this parameter is set to `Conserving port`.

`Tooth height` — **Tooth height**
10 mm (default) | positive scalar

Distance between the base and crest of a tooth. Ring and hub teeth share the same height. The tooth height and the ring-hub clearance when fully disengaged determine the maximum travel span of the shift linkage. The value must be greater than zero.

Dependencies

This parameter is visible only if, in the **Shift Linkage** settings, the **Shift linkage control** parameter to `Conserving port`.

`Ring-hub clearance when disengaged` — **Ring-hub clearance when disengaged**
3 mm (default) | positive scalar

Maximum open gap between the ring and hub tooth crests along the shift linkage translation axis. This gap corresponds to the fully disengaged clutch state. The tooth height and the ring-hub

clearance when fully disengaged determine the maximum travel span of the shift linkage. The value must be greater than zero.

Dependencies

This parameter is visible only if, in the **Shift Linkage** settings, the **Shift linkage control** parameter to `Conserving port`.

Hard stop at back of shift linkage — Hard stop model
`On` (default) | `Off`

Hard stop that prevents the shift linkage from traveling beyond the fully disengaged position:

- `On` — Hard stop when fully disengaged.
- `Off` — No hard stop when fully disengaged.

Ring stop stiffness — Ring stop stiffness
`10e5 N/m` (default) | positive scalar

Linear stiffness coefficient of the ring end stop. This coefficient characterizes the restoring component of the contact force that resists translational motion past the end stops. Greater stiffness values correspond to greater contact forces and a smaller end stop compliance. The value must be greater than zero.

Dependencies

This parameter is visible only if, in the **Shift Linkage** settings, the **Shift linkage control** parameter to `Conserving port`.

Ring stop damping — Ring stop damping
`1e3 N/(m/s)` (default) | nonegative scalar

Linear damping coefficient of the ring end stop. This coefficient characterizes the dissipative component of the contact force that resists translational motion past the end stops. Greater damping values correspond to greater energy dissipation during contact. The value must be greater than or equal to zero

Dependencies

This parameter is visible only if, in the **Shift Linkage** settings, the **Shift linkage control** parameter to `Conserving port`.

Shift linkage viscous friction coefficient — Shift linkage viscous friction coefficient
`100 N/(m/s)` (default) | positive scalar

Linear damping coefficient acting on the shift linkage. This coefficient characterizes the dissipative force that resists shift linkage motion due to viscous damping. Greater coefficient values correspond to greater energy dissipation during shift linkage motion. The value must be greater than zero.

Dependencies

This parameter is visible only if, in the **Shift Linkage** settings, the **Shift linkage control** parameter to `Conserving port`.

Tooth-tooth friction coefficient — Tooth-tooth contact kinetic friction coefficient
`0.05` (default) | positive scalar

Kinetic friction coefficient at the contact interface between ring and hub teeth. This coefficient characterizes the dissipative force that resists shift linkage motion due to tooth-tooth contact during clutch engagement/disengagement.

Greater coefficient values correspond to greater energy dissipation during shift linkage motion. The value must be greater than zero.

Dependencies

This parameter is only visible when the **Shift linkage control** parameter is set to `Conserving port` and, in the **Clutch** settings:

- **Thermal port** is set to `Omit`
- **Torque transmission model** is set to `Dynamic with backlash`

Engagement Conditions

Linkage travel direction — Engagement condition travel direction
`Positive shift linkage displacement engages clutch` (default) | `Negative shift linkage displacement engages clutch`

Direction the shift linkage must travel in to engage the clutch. Choices include positive and negative displacements.

Maximum engagement speed — Engagement speed upper velocity threshold
`inf` (default) | positive scalar

Relative angular velocity between the ring and the hub above which the clutch cannot engage. The value is specific to the specific gearbox or transmission. Minimizing the value helps avoid high dynamic impact during engagement. The value must be greater than zero.

Dependencies

This parameter is visible only if, in the **Shift Linkage** settings, the **Shift linkage control** parameter to `Conserving port`.

Tooth overlap to engage — Tooth overlap engagement threshold
3 mm (default) | positive scalar

Overlap length between ring and hub teeth along the common longitudinal axis above which the clutch can engage. The clutch remains disengaged until the tooth overlap by at least this length. The value must be greater than zero.

Initial Conditions

Initial state — Initial clutch state
`Unlocked` (default) | `Locked`

Clutch state at the start of simulation:

- `Unlocked` — Clutch transmits zero torque between the ring and the hub.
- `Locked` — Clutch transmits torque between the ring and the hub.

Initial shift linkage position — Initial shift linkage position
0 mm (default) | scalar

Shift linkage position at simulation time zero. Values between zero and the sum of the ring-hub clearance and the tooth overlap to engage are consistent with a disengaged clutch. Larger values are consistent with an engaged clutch.

Dependencies

This parameter is only visible when both of these conditions are met:

- In the **Clutch** settings, the **Thermal port** parameter is set to `Omit`.
- In the **Shift Linkage** settings, **Shift linkage control** is set to `Conserving port`.

Initial ring-hub offset angle — Initial ring-hub offset angle
0 deg (default) | scalar

Rotation angle between the ring and the hub at simulation time zero. This angle determines whether the ring and hub teeth can interlock, and hence whether the clutch can engage. The initial offset angle must satisfy these conditions:

- If the clutch initial state is disengaged, the initial offset angle must fall in the range

$$-\frac{180°}{N} \le \phi_0 \le +\frac{180°}{N},$$

where N is the number of teeth present in the ring or the hub. The two components contain the same number of teeth.

- If the clutch initial state is engaged, the initial offset angle must fall in the range

$$-\frac{\delta}{2} \le \phi_0 \le +\frac{\delta}{2},$$

where δ is the backlash angle between the ring and hub teeth.

Dependencies

This parameter is only visible when, in the **Clutch** settings, both of these conditions are met:

- **Thermal port** is set to `Omit`.
- **Torque transmission model** is set to `Dynamic with backlash`.

Thermal Port

These settings are only visible when, in the **Clutch Settings**, the **Thermal port** parameter is set to `Model`.

Thermal mass — Resistance to temperature change
50 kJ/K (default) | scalar

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

This parameter is only visible when, in the **Clutch** settings, the **Thermal Port** parameter is set to `Model`.

Initial temperature — Initial temperature
300 K (default) | scalar

Component temperature at the start of simulation. The initial temperature alters the component efficiency according to an efficiency vector that you specify, affecting the starting meshing or friction losses.

Dependencies

This parameter is only visible when, in the **Clutch** settings, the **Thermal Port** parameter is set to `Model`.

More About

Linearization

To optimize your model for linearization, use the **Clutch** > **Torque transmission model** parameter default setting, `Friction clutch approximation - Suitable for HIL and linearization`.

Hardware-in-the-Loop Simulation

For optimal simulation performance, use the **Clutch** > **Torque transmission model** parameter default setting, `Friction clutch approximation - Suitable for HIL and linearization`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Fundamental Friction Clutch | Loaded-Contact Rotational Friction | Torsional Spring-Damper

Topics
"Gear Coupling Control Using Clutches"
"Model Thermal Losses in Driveline Components"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

Double-Pinion Planetary Gear

Planetary gear train with two meshed planet gear sets
Library: Simscape / Driveline / Gears
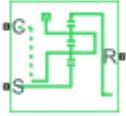
Description

The Double-Pinion Planetary Gear block represents a planetary gear train with two meshed planet gear sets between the sun gear and the ring gear. A single carrier holds the two planet gear sets at different radii from the sun gear centerline, while allowing the individual gears to rotate with respect to each other. The gear model includes power losses due to friction between meshing gear teeth and viscous damping of the spinning gear shafts.
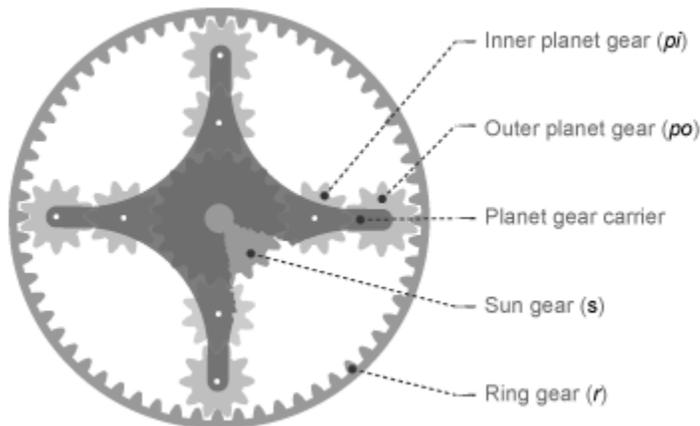
Structurally, the double-pinion planetary gear resembles a Ravigneaux gear without its second, large, sun gear. The inner planet gears mesh with the sun gear and the outer planet gears mesh with the ring gear. Because it contains two planet gear sets, the double-pinion planetary gear reverses the relative rotation directions of the ring and sun gears.

The teeth ratio of a meshed gear pair fixes the relative angular velocities of the two gears in that pair. In the Double-Pinion Planetary Gear block, you can specify the gear teeth ratios between the ring and sun gears and the outer planet and inner planet gears. A geometric constraint fixes the remaining teeth ratios—the ring gear to the outer planet gear and the inner planet gear to the sun gear. This constrains the ring gear radius to the sum of the sun gear radius and the inner and outer planet gear diameters.

$$r_r = r_s + 2 \cdot r_{pi} + 2 \cdot r_{po},$$

where:

- r_r is the ring gear radius.

- r_s is the sun gear radius.
- r_{pi} is the inner planet gear radius.
- r_{po} is the outer planet gear radius.
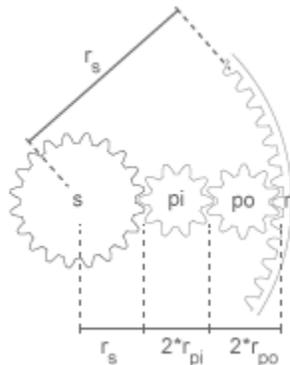
The ring to outer planet gear teeth ratio is

$$\frac{r_r}{r_{po}} = 2 \cdot \frac{\frac{r_r}{r_s}}{\left(\frac{r_r}{r_s} - 1\right)} \cdot \frac{\left(\frac{r_{po}}{r_{pi}} + 1\right)}{\frac{r_{po}}{r_{pi}}}.$$

The inner planet-sun teeth ratio is

$$\frac{r_{pi}}{r_s} = \frac{\left(\frac{r_r}{r_s} - 1\right)}{2\left(\frac{r_{po}}{r_{pi}} + 1\right)}.$$

The Double-Pinion Planetary Gear block is a composite component. It contains three underlying blocks—Ring-Planet, Planet-Planet, and Sun-Planet—connected as shown in the figure. Each block connects to a separate drive shaft through a rotational conserving port.
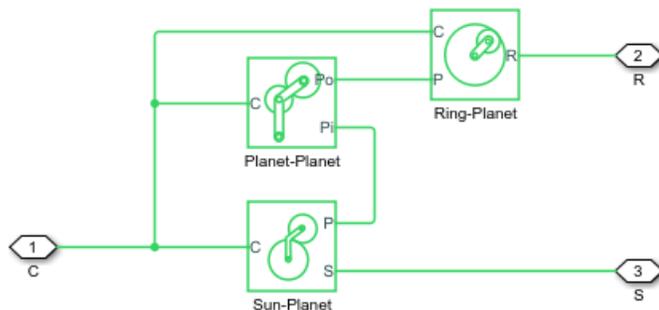
Thermal Model

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Friction model** to `Temperature-dependent efficiency`.

Assumptions and Limitations

- Gear inertia is assumed to be negligible.
- Gears are treated as rigid components.
- Coulomb friction slows down simulation. For more information, see "Adjust Model Fidelity".

Ports

Conserving

C — Planet gear carrier
rotational mechanical

Rotational mechanical conserving port associated with the planet gear carrier.

R — Ring gear
rotational mechanical

Rotational mechanical conserving port associated with the ring gear.

S — Sun gear
rotational mechanical

Rotational mechanical conserving port associated with the sun gear.

H — Heat flow
thermal

Thermal conserving port associated with heat flow. Heat flow affects the power transmission efficiency by altering the gear temperatures.

Dependencies

To enable this port, set **Friction model** to Temperature-dependent efficiency.

Parameters

Main

Ring (R) to sun (S) teeth ratio (NR/NS) — Ring to sun gear rotation ratio
2 (default) | positive scalar

Fixed ratio of the ring gear to the sun gear rotations as defined by the number of planet gear teeth divided by the number of sun gear teeth.

Outer planet (Po) to inner planet (Pi) teeth ratio (NPo/NPi) — Outer planet to inner planet gear rotation ratio
1 (default) | positive scalar

Fixed ratio of the outer planet gear to the inner planet gear rotations as defined by the number of planet gear teeth divided by the number of sun gear teeth.

Meshing Losses

Friction model — Friction model
No meshing losses - Suitable for HIL simulation (default) | Constant efficiency | Temperature-dependent efficiency

Friction model for the block:

- No meshing losses - Suitable for HIL simulation — Gear meshing is ideal.
- Constant efficiency — Transfer of torque between the gear wheel pairs is reduced by a constant efficiency, η, such that $0 < \eta \leq 1$.
- Temperature-dependent efficiency — Transfer of torque between the gear wheel pairs is defined by the table lookup based on the temperature.

Sun-planet, ring-planet and planet-planet ordinary efficiencies — Torque transfer efficiencies for the
[.98, .98, .99] (default) | vector

Vector of torque transfer efficiencies, $[\eta_{SP}, \eta_{RP}, \eta_{RPP}]$, for sun-planet, and ring-planet, and planet-planet gear wheel pair meshings, respectively. The vector elements must be in the interval (0,1].

Dependencies

To enable this parameter, set **Friction model** to Constant efficiency.

Temperature — Temperature
[280, 300, 320] K (default) | vector

Vector of temperatures used to construct a 1-D temperature-efficiency lookup table. The vector elements must increase from left to right.

Dependencies

To enable this parameter, set **Friction model** to Temperature-dependent efficiency.

Sun-planet efficiency — Torque transfer efficiency from the sun gear to the inner planet gears
[.75, .65, .6] (default) | vector

Vector of output-to-input power ratios that describe the power flow from the sun gear to the inner planet gears, η_{SP}. The block uses the values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to Temperature-dependent efficiency.

Ring-planet efficiency — Torque transfer efficiency from the ring gear to the outer planet gears
[.5, .45, .4] (default) | vector

Vector of output-to-input power ratios that describe the power flow from the ring gear to the outer planet gears, η_{RP}. The block uses the values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Planet-planet efficiency — Torque transfer efficiency from the inner planet gears to the outer planet gears
[.5, .45, .4] (default) | vector

Vector of output-to-input power ratios that describe the power flow from the inner planet gear to the outer planet gear, η_{PP}. The block uses the values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Sun-carrier, ring-carrier, and planet-carrier power thresholds — Minimum efficiency power threshold for the sun-carrier, ring-carrier, and planet-carrier gear couplings
[.001, .001, .001] W (default) | vector

Vector of power thresholds above which the full efficiency factors apply. Enter the thresholds in the order sun-carrier, ring-carrier, planet-carrier.

When you set **Friction model** to `Constant efficiency`, the block lowers the efficiency losses to zero when no power is transmitted. When you set **Friction model** to `Temperature-dependent efficiency`, the block smooths the efficiency factors between zero when at rest and the values provided by the temperature-efficiency lookup tables at the power thresholds.

The power threshold should be lower than the expected power transmitted during simulation. Higher values may cause the block to underestimate efficiency losses. Very low values may raise the computational cost.

Dependencies

To enable the parameter, set **Friction model** to `Constant efficiency` or `Temperature-dependent efficiency`.

Viscous Losses

Sun-carrier, ring-carrier, and planet-carrier viscous friction coefficients — Gear viscous friction
[0, 0, 0] N*m/(rad/s) (default) | vector

Vector of viscous friction coefficients, $[\mu_S \; \mu_R \; \mu_P]$, for the sun-carrier, ring-carrier, and planet-carrier gear motions, respectively.

Inertia

Inertia — Inertia model
off (default) | on

Inertia model for the block:

- Off — Model gear inertia.
- On — Neglect gear inertia.

Inner planet gear inertia — Inner planet gear inertia
0.001 kg*m^2 (default) | positive scalar

Moment of inertia of the inner planet gears. This value must be positive.

Dependencies

To enable this paramter, set **Inertia** to On.

Outer planet gear inertia — Outer planet gear inertia
0.001 kg*m^2 (default) | positive scalar

Moment of inertia of the outer planet gears. This value must be positive.

Dependencies

Tho enable this parameter, set **Inertia** to On.

Thermal Port

To enable these settings, set **Friction model** to Temperature-dependent efficiency.

Thermal mass — Thermal mass
50 J/K (default) | scalar

Thermal energy required to change the component temperature by a single temperature unit. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

To enable this parameter, set **Friction model** to Temperature-dependent efficiency.

Initial temperature — Initial temperature
300 K (default) | scalar

Block temperature at the start of simulation. The initial temperature sets the initial component efficiencies according to their respective efficiency vectors.

Dependencies

To enable this parameter, set **Friction model** to Temperature-dependent efficiency.

More About

Hardware-in-the-Loop Simulation

For optimal simulation performance, set **Friction model** to the default value, `No meshing losses - Suitable for HIL simulation`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Planetary Gear | Compound Planetary Gear | Ravigneaux Gear

Topics
"Model Thermal Losses in Driveline Components"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2013b

Double-Shoe Brake

Frictional brake with two pivoted shoes diametrically positioned about a rotating drum with triggered faulting

Library: Simscape / Driveline / Brakes & Detents / Rotational
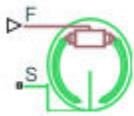
Description

The Double-Shoe Brake block represents a frictional brake with two pivoted rigid shoes that press against a rotating drum to produce a braking action. The rigid shoes sit inside or outside the rotating drum in a diametrically opposed configuration. A positive actuating force causes the rigid shoes to press against the rotating drum. Viscous and contact friction between the drum and the rigid shoe surfaces cause the rotating drum to decelerate.

Double-shoe brakes provide high braking torque with small actuator deflections in applications that include motor vehicles and some heavy machinery. The model employs a simple parameterization with readily accessible brake geometry and friction parameters.

You can also enable faulting. When faulting occurs, the belt will exert a user-specified force. Faults can occur at a specified time or due to an external trigger at port **T**.

Equations

In the schematic, a) represents an internal double-shoe brake, and b) represents an external double-shoe brake. In both configurations, a positive actuation force F brings the shoe and drum friction surfaces into contact. The result is a friction torque that causes deceleration of the rotating drum. Zero and negative forces do not bring the shoe and drum friction surfaces into contact and produce zero braking torque.

(a)   (b)
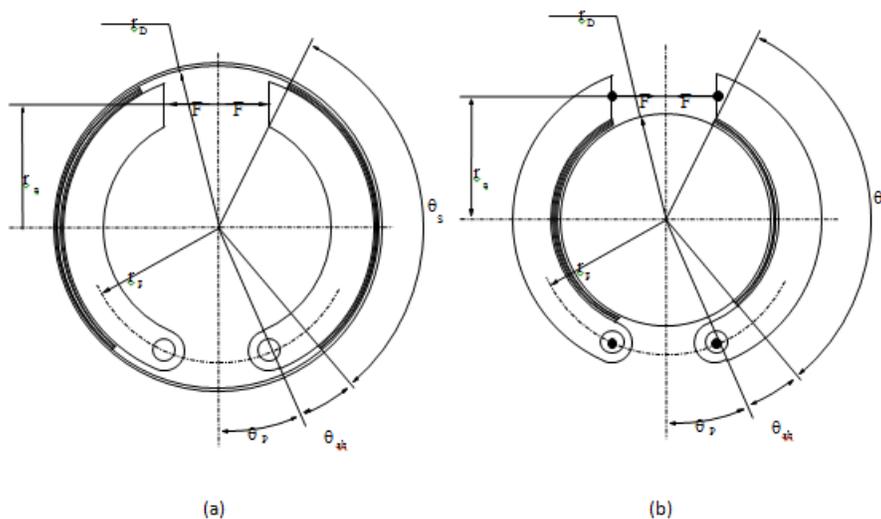
The model uses the long-shoe approximation. The equations for the friction torque that the leading and trailing shoes develop are:

$$T_{LS} = \frac{c\mu p_a r_D 2(\cos\theta_{sb} - \cos\theta_s)}{\sin\theta_a},$$

$$T_{TS} = \frac{c\mu p_b r_D 2(\cos\theta_{sb} - \cos\theta_s)}{\sin\theta_a},$$

$$c = r_a + r_p\cos\theta_p,$$

where for $0 \le \theta_s \le \pi/2$,

$$\theta_a = \theta_s,$$

and for $\theta_s \ge \pi/2$,

$$\theta_a = \frac{\pi}{2}.$$

Where:

- T_{LS} is the brake torque the leading shoe develops.
- T_{TS} is the brake torque the trailing shoe develops.
- μ is the effective contact friction coefficient.
- p_a is the maximum linear pressure in the leading shoe-drum contact.
- p_b is the maximum linear pressure in the trailing shoe-drum contact.
- r_D is the drum radius.
- θ_{sb} is the shoe beginning angle.
- θ_s is the shoe span angle.
- θ_a is the angle from hinge pin to maximum pressure point.
- c is the arm length of the cylinder force with respect to the hinge pin.
- r_p is the pin location radius.
- θ_p is the hinge pin location angle.
- r_a is the actuator location radius.

The model assumes that only Coulomb friction acts at the shoe-drum surface contact. Zero relative velocity between the drum and the shoes produces zero Coulomb friction. To avoid discontinuity at zero relative velocity, the friction coefficient formula employs the hyperbolic function

$$\mu = \mu_{Coulomb}\tanh\left(\frac{4\omega_{shaft}}{\omega_{threshold}}\right),$$

where:

- μ is the effective contact friction coefficient.
- $\mu_{Coulomb}$ is the contact friction coefficient.
- ω_{shaft} is the shaft velocity.

- $\omega_{threshold}$ is the angular velocity threshold.

Balancing the moments that act on each shoe with respect to the pin yields the pressure acting at the shoe-drum surface contact. The equations for determining the balance of moments for the leading shoe are

$$F = \frac{M_N - M_F}{c},$$

$$M_N = \frac{p_a r_p r_D}{\sin\theta_a}\left(\frac{1}{2}[\theta_s - \theta_{sb}] - \frac{1}{4}[\sin2\theta_s - \sin2\theta_{sb}]\right),$$

and

$$M_F = \frac{\mu p_a r_D}{\sin\theta_a}\left(r_D[\cos\theta_{sb} - \cos\theta_s] + \frac{r_p}{4}[\cos2\theta_s - \cos2\theta_{sb}]\right),$$

where:

- F is the actuation force.
- M_N is the moment acting on the leading shoe due to normal force.
- M_F is the moment acting on the leading shoe due to friction force.
- c is the arm length of the cylinder force with respect to the hinge pin.
- p_a is the maximum linear pressure at the shoe-drum contact surface.
- r_p is the pin location radius.
- θ_p is the hinge pin location angle.
- r_a is the actuator location radius.

The model does not simulate self-locking brakes. If brake geometry and friction parameters cause a self-locking condition, the model produces a simulation error. A brake self-locks if the friction moment exceeds the moment due to normal forces, that is, when $M_F > M_N$.

The balance of moments for the trailing shoe is

$$F = \frac{M_N + M_F}{c}.$$

The net braking torque is

$$T = T_{LS} + T_{TS} + \mu_{visc} * \omega_{shaft},$$

where μ_{visc} is the viscous friction coefficient.

Faulty Behavior

When faults are enabled, a belt force is applied in response to one or both of these triggers:

- Simulation time — Faulting occurs at a specified time.
- Simulation behavior — Faulting occurs in response to an external trigger. This exposes port **T**.

If a fault trigger occurs, the input force is replaced by the **Belt force when faulted** value for the remainder of the simulation. A value of 0 implies that no braking will occur. A relatively large value implies that the brake is stuck.

You can set the block to issue a fault report as a warning or error message in the Simulink Diagnostic Viewer with the **Reporting when fault occurs** parameter.

Thermal Model

You can model the effects of heat flow and temperature change by exposing the optional thermal port. To expose the port, in the **Friction** settings, set the **Thermal Port** parameter to `Model`. Exposing the port also exposes or changes the default value for these related settings, parameters, and variables:

- **Friction > Temperature**
- **Friction > Static friction coefficient vector**
- **Friction > Coulomb friction coefficient vector**
- **Friction > Contact friction coefficient vector**
- **Thermal Port > Thermal mass**
- **Variables > Temperature**

Variables

Use the **Variables** settings to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Dependencies

Variable settings are visible only when, in the **Friction** settings, the **Thermal port** parameter is set to `Model`.

Limitations and Assumptions

- Contact angles smaller than 45° produce less accurate results.
- The brake uses the long-shoe approximation.
- The brake geometry does not self-lock.
- The model does not account for actuator flow consumption.

Ports

Input

F — Actuating force
physical signal

Physical signal input port associated with the applied actuating force.

T — External fault trigger
physical signal

Physical signal port for an external fault trigger. Triggering occurs when the value is greater than 0.5. There is no unit associated with the trigger value.

Dependencies

This port is visible when **Enable faults** is set to `On` and **Enable external fault trigger** is set to `On`.

Conserving

S — Drum shaft rotation
rotational mechanical

Rotational conserving port associated with the rotating drum shaft.

H — Heat flow
thermal

Thermal conserving port associated with heat flow.

Dependencies

This port is visible only when, in the **Friction** settings, the **Thermal Port** parameter is set to `Model`.

Exposing this port makes related settings visible.

Parameters

Geometry

`Drum radius` — Contact surface radius
150 mm (default) | positive scalar

Radius of the drum contact surface. The value must be greater than zero.

`Actuator location radius` — Center-to-center distance between drum and force line of action
100mm (default) | positive scalar

Distance between the drum center and the force line of action. The value must be greater than zero.

`Pin location radius` — Center-to-center distance between hinge pin and drum
125mm (default) | positive scalar

Distance between the hinge pin and drum centers. The parameter must be greater than zero.

`Pin location angle` — Angular distance from hinge pin to brake symmetry axis
15deg (default) | nonnegative scalar

Angular coordinate of the hinge pin location from the brake symmetry axis. The value must be greater than or equal to zero.

`Shoe beginning angle` — Angle between hings pin and friction material
5deg (default) | positive scalar

Angle between the hinge pin and the beginning of the friction material linen of the shoe. The value of the parameter must be in the range$0 \leq \theta_{sb} \leq$ (π-pin location angle).

`Shoe span angle` — Friction material angle
120deg (default) | positive scalar

Angle between the beginning and the end of the friction material linen on the shoe. The value of the parameter must be in the range $0 < \theta_{sb} \leq$ (π -pin location angle - shoe beginning angle).

Friction

Viscous friction coefficient — Viscous friction
.01 n*m/(rad/s) (default) | nonnegatvie scalar

Viscous friction coefficient at the contact surface. The value must be greater than or equal to zero.

Thermal port — Thermal model
Omit (default) | Model

Model for heat flow and temperature change:

- Omit — Neglect thermal dynamics.
- Model — Include thermal dynamics.

Dependencies

When this parameter is set to Model, thermal port **H** and related settings are visible.

Temperature — Temperature
[280, 300, 320] K (default) | increasing vector

Array of temperatures used to construct a 1-D temperature-efficiency lookup table. The array values must increase left to right.

Dependencies

This parameter is only visible when the **Thermal Port** parameter is set to Model.

Contact friction coefficient — Coulomb friction
0.3 (default) | [0.1, 0.05, 0.03] (thermal model default) | positive scalar or vector

Coulomb friction coefficient at the belt-drum contact surface. The value must be greater than zero. For the thermal model:

- The number of elements in the vector must be the same as the number of elements in the specified vector for the **Temperature** parameter
- The values increase left to right.
- Each value must be greater than zero.

Dependencies

This parameter is specified as a:

- Scalar when the **Thermal Port** parameter is set to Omit.
- Vector when the **Thermal Port** parameter is set to Model.

Angular velocity threshold — Rotational velocity required for near steady-state contact friction
0.01 rad/s (default) | positive scalar

Angular velocity at which the contact friction coefficient practically reaches its steady-state value. The value must be greater than zero.

Faults

`Enable faults` — Fault option
`Off` (default) | `On`

Enable externally or temporally triggered faults. When faulting occurs, the brake belt force normally received at port **F** will be set to the value specified in the **Belt force when faulted** parameter.

`Belt force when faulted` — Set faulted belt force
`0 N` (default) | positive scalar

Set faulted belt force. When faulting occurs, the brake belt force normally received at port **F** will be set to the value specified in the **Belt force when faulted** parameter. A value of `0` implies that braking does not occur. A relatively large value implies that the brake is stuck.

Dependencies

To enable this parameter, set **Enable faults** to `On`.

`Enable external fault trigger` — External trigger option
`Off` (default) | `On`

Enables port **T**. A physical signal at port **T** that is greater than `0.5` triggers faulting.

Dependencies

To enable this parameter, set **Enable faults** to `On`.

`Enable temporal fault trigger` — Temporal trigger option
`Off` (default) | `On`

Enables fault triggering at a specified time. When the **Simulation time for fault event** is reached, the brake belt force normally received at port **F** will be set to the value specified in the **Belt force when faulted** parameter.

Dependencies

To enable this parameter, set **Enable faults** to `On`.

`Simulation time for fault event` — Time at which faulting is triggered
`5 s` (default) | positive scalar

When the **Simulation time for fault event** is reached, the brake belt force normally received at port **F** will be set to the value specified in the **Belt force when faulted** parameter.

Dependencies

To enable this parameter, set **Enable faults** to `On` and **Enable temporal fault trigger** to `On`.

`Reporting when fault occurs` — Fault condition report
`None` (default) | `Warning` | `Error`

Reporting preference for the fault condition. When reporting is set to `Warning` or `Error`, a message is displayed in the Simulink Diagnostic Viewer. When `Error` is selected, the simulation will stop if faulting occurs.

Dependencies

To enable this parameter, set **Enable faults** to On.

Thermal Port

Thermal mass — Resistance to temperature change
50 kJ/K (default) | scalar

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

This parameter is only visible when, in the **Friction** settings, the **Thermal Port** parameter is set to Model.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Band Brake | Loaded-Contact Rotational Friction | Rotational Detent

Topics
"Model Thermal Losses in Driveline Components"

Introduced in R2012b

Double-Sided Synchronizer

Back-to-back dog-cone clutch pairs assembled symmetrically about a translational detent to provide smooth gear engagement
Library: Simscape / Driveline / Clutches
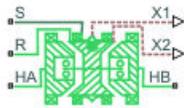
Description

The block represents a double-sided synchronizer that contains two back-to-back dog clutches, two back-to-back cone clutches, and one translational detent. Shift linkage translation along the negative direction causes the clutches to engage the ring with hub *A*. Shift linkage translation along the positive direction causes the clutches to engage the ring with hub *B*. When the magnitude of the shift linkage translation is smaller than the cone clutch ring-hub gap, the synchronizer is in neutral mode and does not transmit torque.
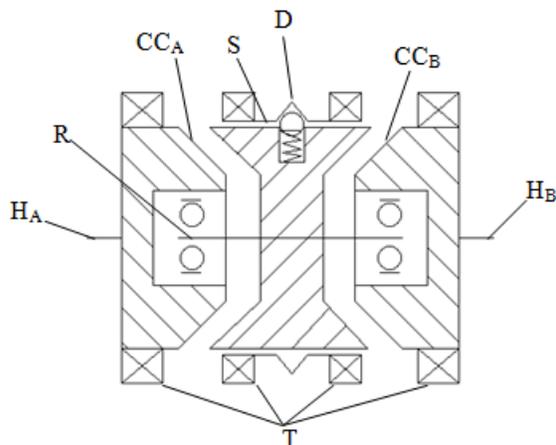
The schematic illustrates a double-sided synchronizer in the disengaged state. In this state, the ring, **R**, and hub, **H$_A$** and **H$_B$**, shafts can spin independently at different speeds. As the shift linkage, **S**, translates in the negative direction, the faces of cone clutch *A* (**CC$_A$**) come into contact. The friction in the cone clutch decreases the difference in rotational speed between the shafts. When the force on the shift linkage exceeds the peak force of detent, **D**, the dog clutch teeth, **T**, can engage. The detent peak force should be such that the cone clutch has enough time and normal force to bring the shafts to sufficiently similar speeds to allow engagement of the dog clutch. Similarly, translating the shift linkage along the positive direction allows the faces of cone clutch *B* (**CC$_B$**) to come into contact, and can allow the shaft of the ring to engage with the shaft of the hub *B* (**H$_B$**).
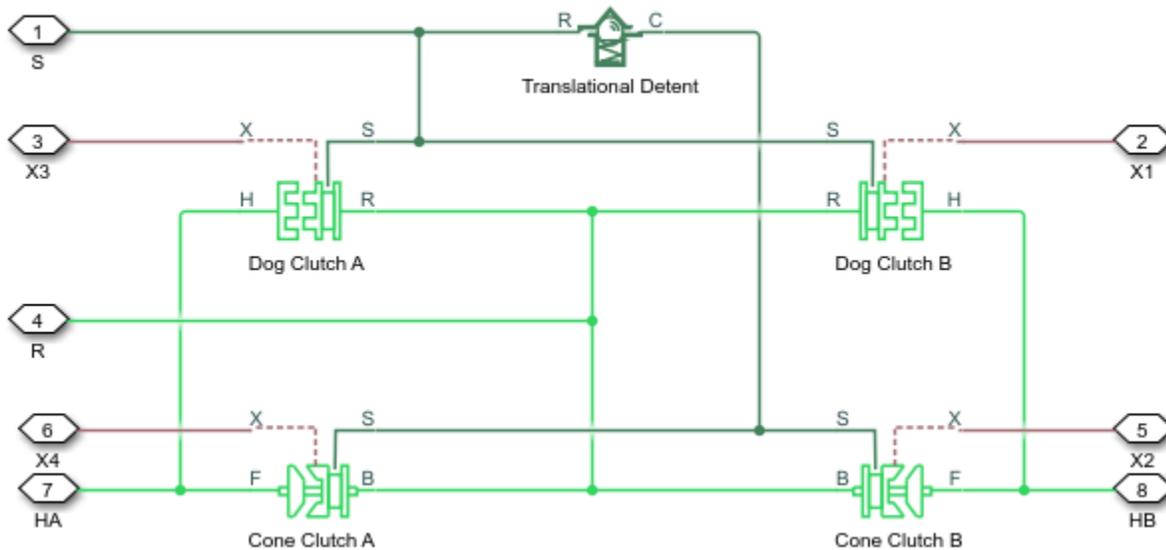
The model implements two Dog Clutch blocks, two Cone Clutch blocks, and one Translational Detent block. Refer to each block reference page for more information on the corresponding block function.

Connections **R**, **H$_A$**, and **H$_B$** are mechanical rotational conserving ports that represent the ring, **R**, hub A (**H$_A$**), and hub B (**H$_B$**), respectively. Connection **S** is a mechanical translational conserving port that represents the ring shifter handle.

Connections **X1** and **X2** are physical signal ports that output the shift linkage positions of the dog clutches and cone clutches, respectively. The tables provide the values of *X1* and *X2* in common clutch engagement cases.

Dog Clutch State	X1
Disengaged	0
Fully engaged with hub *A*	Negative sum of ring-hub gap and tooth height
Fully engaged with hub *B*	Positive sum of ring-hub gap and tooth height

Cone Clutch State	X2
Disengaged	0
Fully engaged with hub *A*	Negative value of ring-hub gap
Fully engaged with hub *B*	Positive value of ring-hub gap

The values of *X1* and *X2* are zero when the synchronizer is fully disengaged. When the dog clutch is fully engaged with hub *A*, *X1* is equal to the negative sum of its ring-hub gap and tooth height. When the dog clutch is fully engaged with hub *B*, *X1* is equal to the positive sum of its ring-hub gap and tooth height. When the cone clutch is fully engaged with hub *A*, *X2* is equal to the negative of its ring-hub gap. When the cone clutch is fully engaged with hub *B*, *X2* is equal to its ring-hub gap.

Thermal Modeling

You can model the effects of heat flow and temperature change through an optional thermal conserving port. By default, the thermal port is hidden. To expose the thermal port, in the **Clutch** settings, select a temperature-dependent setting tor the **Friction model** parameter. Specify the associated thermal parameters for the component.

Assumptions and Limitations

- The model does not account for inertia effects. You can add a Simscape Inertia block at each port to add inertia to the synchronizer model.

Ports

Output

X1 — Dog clutch translation
physical signal

Physical signal output port that measures the magnitude of the dog clutch translation.

X2 — Cone clutch translation
physical signal

Physical signal output port that measures the magnitude of the cone clutch translation.

Conserving

HA — Clutch hub *A*
mechanical rotational

Mechanical rotational conserving port associated with the clutch hub *A* shaft.

HB — Clutch hub *B*
mechanical rotational

Mechanical rotational conserving port associated with the clutch hub *B* shaft.

R — Clutch ring
mechanical rotational

Mechanical rotational conserving port associated with the clutch ring.

S — Shift linkage
mechanical rotational

Mechanical rotational conserving port associated with shift linkage.

T — Heat flow
thermal

Thermal conserving port associated with heat flow.

Dependencies

This port is visible only if, in the **Friction** settings, the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`.

Parameters

The table shows how the specified options for parameters in both the **Cone Clutch** and **Dog Clutch** settings affect the visibility of:

- Parameters in the **Cone Clutch**, **Dog Clutch**, and **Initial Conditions** settings
- **Thermal Port** settings
- Thermal port **T**

To learn how to read the table, see "Parameter Dependencies" on page B-2.

Double-Sided Synchronizer Block Parameter Dependencies

Settings	Parameters and Options			
Cone Clutch	Contact surface maximum diameter			
	Contact surface minimum diameter			
	Cone half angle			
	Friction model			
	`Fixed kinetic friction coefficient`	`Velocity-dependent kinetic friction coefficient`	`Temperature-dependent friction coefficients`	`Temperature and velocity-dependent friction coefficients`
	-	-	Exposes: • Conserving port **T** • **Thermal Port** settings	Exposes: • Conserving port **T** • **Thermal Port** settings
	-	**Relative velocity vector**		**Relative velocity vector**
	-	-	**Temperature vector**	**Temperature vector**
	Static friction coefficient	**Static friction coefficient vector**	**Static friction coefficient vector**	**Static friction coefficient matrix**
	Kinetic friction coefficient	**Kinetic friction coefficient vector**	**Kinetic friction coefficient vector**	**Kinetic friction coefficient matrix**
	-	**Friction coefficient interpolation method**	**Friction coefficient interpolation method**	**Friction coefficient interpolation method**
	-	**Friction coefficient extrapolation method**	**Friction coefficient extrapolation method**	**Friction coefficient extrapolation**
	Velocity tolerance	**Velocity tolerance**	**Velocity tolerance**	**Velocity tolerance**
	Threshold force	**Threshold force**	**Threshold force**	**Threshold force**

Settings	Parameters and Options					
Dog Clutch	**Torque transmission model**		**Torque transmission model**		-	-
	`Friction clutch approximation - Suitable for HIL and linearization`	`Dynamic with backlash`	`Friction clutch approximation - Suitable for HIL and linearization`	`Dynamic with backlash`	-	-
	-	-	-	-	**Temperature vector**	**Temperature vector**
	Maximum transmitted torque	-	**Maximum transmitted torque**	-	**Maximum transmitted torque vector**	**Maximum transmitted torque vector**
	-	-	-	-	**Interpolation method**	**Interpolation method**
	-	-	-	-	**Extrapolation method**	**Extrapolation method**
	Clutch teeth mean radius	**Clutch teeth mean radius**	**Clutch teeth mean radius**	**Clutch teeth mean radius**	**Clutch teeth mean radius**	**Clutch teeth mean radius**
	-	**Number of teeth**	-	**Number of teeth**		
		Rotational backlash		**Rotational backlash**		
	-	**Torsional stiffness**	-	**Torsional stiffness**		
	-	**Torsional damping**	-	**Torsional damping**		
	-	**Tooth-tooth friction coefficient**	-	**Tooth-tooth friction coefficient**		
Initial Conditions	**Initial state**	**Initial state**	**Initial state**	**Initial state**	**Initial state**	**Initial state**
	Dog clutch initial shift linkage position	**Dog clutch initial shift linkage position**	**Dog clutch initial shift linkage position**	**Dog clutch initial shift linkage position**	**Dog clutch initial shift linkage position**	**Dog clutch initial shift linkage position**
	Cone clutch initial shift linkage position	**Cone clutch initial shift linkage position**	**Cone clutch initial shift linkage position**	**Cone clutch initial shift linkage position**	**Cone clutch initial shift linkage position**	**Cone clutch initial shift linkage position**

Blocks

Settings	Parameters and Options					
	-	Initial dog clutch A ring-hub offset angle	-	Initial dog clutch A ring-hub offset angle	-	-
	-	Initial dog clutch B ring-hub offset angle	-	Initial dog clutch B ring-hub offset angle	-	-
Thermal Port	-	-	-	-	Thermal mass	Thermal mass
	-	-	-	-	Initial temperature	Initial temperature

Cone Clutch

Contact surface maximum diameter — Outer diameter
150 mm (default) | positive scalar

Outer conical diameter d_o.

Contact surface minimum diameter — Inner diameter
100 mm (default) | positive scalar

Inner conical diameter d_i.

Cone half angle — Cone half angle
12 deg (default) | positive scalar

Half opening angle α of the cone geometry.

Friction model — Friction model
Fixed kinetic friction coefficient (default) | Velocity-dependent kinetic friction coefficient | Temperature-dependent friction coefficients | Temperature and velocity-dependent friction coefficients

Parameterization method to model the kinetic friction coefficient. The options and default values for this parameter depend on the friction model that you select for the block. The options are:

- Fixed kinetic friction coefficient — Provide a fixed value for the kinetic friction coefficient.
- Velocity-dependent kinetic friction coefficient — Define the kinetic friction coefficient by one-dimensional table lookup based on the relative angular velocity between disks.
- Temperature-dependent friction coefficients — Define the kinetic friction coefficient by table lookup based on the temperature.
- Temperature and velocity-dependent friction coefficients — Define the kinetic friction coefficient by table lookup based on the temperature and the relative angular velocity between disks.

Dependencies

The friction model setting affects the visibility of other parameters, settings, and ports.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Relative velocity vector — Relative velocity
`[0, 100, 1000] rad/s` (default) | vector

Input values for the relative velocity as a vector. The values in the vector must increase from left to right. The minimum number of values depends on the interpolation method that you select. For linear interpolation, provide at least two values per dimension. For smooth interpolation, provide at least three values per dimension.

Dependencies

This parameter is visible only if the **Friction model** parameter is set to `Velocity-dependent kinetic friction coefficient` or `Temperature and velocity-dependent friction coefficients`.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Temperature vector — Temperature
`[280, 300, 320] K` (default) | increasing vector

Input values for the temperature as a vector. The minimum number of values depends on the interpolation method that you select. For linear interpolation, provide at least two values per dimension. For smooth interpolation, provide at least three values per dimension. The values in the vector must increase from left to right.

Dependencies

This parameter is visible only if the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Static friction coefficient — Static friction coefficient
`0.35` (default) | scalar

Static or peak value of the friction coefficient. The static friction coefficient must be greater than the kinetic friction coefficient.

Dependencies

this parameter is visible only if the **Friction model** parameter is set to `Fixed kinetic friction coefficient` or `Velocity-dependent kinetic friction coefficient`.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Static friction coefficient vector — Static friction coefficient
`[.4, .38, .36]` (default) | vector

Static, or peak, values of the friction coefficient as a vector. The vector must have the same number of elements as the temperature vector. Each value must be greater than the value of the corresponding element in the kinetic friction coefficient vector.

Dependencies

This parameter is visible only if the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Kinetic friction coefficient — Kinetic friction coefficient
`0.3` (default) | positve scalar

The kinetic, or Coulomb, friction coefficient. The coefficient must be greater than zero.

Dependencies

This parameter is visible only if the **Friction model** parameter is set to `Fixed kinetic friction coefficient`.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Kinetic friction coefficient vector — Kinetic friction coefficient
`[.3, .22, .19]` (default) | `[.3, .28, .25]` | vector

Output values for kinetic friction coefficient as a vector. All values must be greater than zero.

If the **Friction model** parameter is set to

- `Velocity-dependent kinetic friction coefficient` — The vector must have same number of elements as relative velocity vector.
- `Temperature-dependent friction coefficients` — The vector must have the same number of elements as the temperature vector.

Dependencies

This parameter is visible only if the **Friction model** parameter is set to `Velocity-dependent kinetic friction coefficient` or `Temperature-dependent friction coefficients`.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Kinetic friction coefficient matrix — Kinetic friction coefficient
`[.34, .32, .3; .3, .28, .25; .25, .2, .15]` (default) | matrix

Output values for kinetic friction coefficient as a matrix. All the values must be greater than zero. The size of the matrix must equal the size of the matrix that is the result of the temperature vector × the kinetic friction coefficient relative velocity vector.

Dependencies

This parameter is visible only if the **Friction model** parameter is set to `Temperature and velocity-dependent friction coefficients`.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Friction coefficient interpolation method — Interpolation method
`Linear` (default) | `Smooth`

Interpolation method for approximating the output value when the input value is between two consecutive grid points:

- `Linear` — Select this option to get the best performance.
- `Smooth` — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Velocity-dependent kinetic friction coefficient`, `Temperature-dependent friction coefficients`, or `Temperature and velocity-dependent friction coefficients`.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Friction coefficient extrapolation method — Extrapolation method
`Linear` (default) | `Nearest` | `Error`

Extrapolation method for determining the output value when the input value is outside the range specified in the argument list:

- `Linear` — Select this option to produce a curve with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region.
- `Nearest` — Select this option to produce an extrapolation that does not go above the highest point in the data or below the lowest point in the data.
- `Error` — Select this option to avoid going into the extrapolation mode when you want your data to be within the table range. If the input signal is outside the range of the table, the simulation stops and generates an error.

For more information on extrapolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Velocity-dependent kinetic friction coefficient`, `Temperature-dependent friction coefficients`, or `Temperature and velocity-dependent friction coefficients`.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Velocity tolerance — Relative velocity locking threshold
`0.001 rad/s` (default) | scalar

Relative velocity below which the two surfaces can lock. The surfaces lock if the torque is less than the product of the effective radius, the static friction coefficient, and the applied normal force.

Threshold force — Normal contact force threshold
`1 N` (default) | scalar

The normal force is applied only if the amount of force exceeds the value of the **Threshold force** parameter. Forces below the **Threshold force** are not applied so there is no transmitted frictional torque.

Dog Clutch

The methods that are available for parameterizing the torque transmission depend whether the friction model is temperature-dependent.

The friction model is determined, in the **Cone Clutch** settings, by the **Friction model** parameter setting:

- `Fixed kinetic friction coefficient` — Temperature independent
- `Velocity-dependent kinetic friction coefficient` — Temperature independent
- `Temperature-dependent friction coefficients` — Temperature dependent
- `Temperature and velocity-dependent friction coefficients` — Temperature dependent

For a temperature-independent model, parameterize the block using one of the options for the **Torque Transmission Model** parameter.

Torque transmission model — Torque transmission model
`Friction clutch approximation — Suitable for HIL and linearization` (default) | `Dynamic with backlash`

Computational framework for modeling the dynamic behavior of the dog clutch:

- `Friction clutch approximation — Suitable for HIL and linearization` — Model clutch engagement as a friction phenomenon between the ring and the hub. This model, based on the Fundamental Friction Clutch block, provides a computationally efficient approximation of the dog clutch.
- `Dynamic with backlash` — Model clutch engagement in detail, accounting for such phenomena as backlash, torsional compliance, and contact forces between ring and hub teeth.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Fixed kinetic friction coefficient` or `Velocity-dependent friction coefficients`.

The visibility of related parameters in the **Dog Clutch** and **Initial Conditions** settings is affected by the option that you select for this parameter.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Temperature vector — Temperature
`[280, 300, 320]` K (default) | increasing vector

Input values for the temperature as a vector. The minimum number of values depends on the interpolation method that you select. For linear interpolation, provide at least two values per dimension. For smooth interpolation, provide at least three values per dimension. The values in the vector must increase from left to right.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Maximum transmitted torque — Maximum transmitted torque
`1000 N*m` (default) | positive scalar

Largest torque that the clutch can transmit, corresponding to a nonslip engaged configuration. If the torque transmitted between the ring and the hub exceeds this value, the two components begin to slip with respect to each other. This torque determines the static friction limit in the friction clutch approximation

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Fixed kinetic friction coefficient` or `Velocity-dependent kinetic friction coefficient` and, in **Dog Clutch** settings, the **Torque transmission model** parameter is set to `Friction clutch approximation - Suitable for HIL and linearization`.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Maximum transmitted torque vector — Maximum transmitted torque
`[1000, 1050, 1000] N*m` (default) | vector

Largest torque that the clutch can transmit, corresponding to a nonslip engaged configuration, specified as a vector. If the torque transmitted between the ring and the hub exceeds this value, the two components begin to slip with respect to each other. This torque determines the static friction limit in the friction clutch approximation. The vector has the same number of elements as the temperature vector.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Temperature-dependent kinetic friction coefficient` or `Temperature and velocity-dependent kinetic friction coefficient`.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Interpolation method — Interpolation method
`Linear` (default) | `Smooth`

Interpolation method for approximating the output value when the input value is between two consecutive grid points:

- `Linear` — Select this option to get the best performance.
- `Smooth` — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Temperature-dependent kinetic friction coefficient` or `Temperature and velocity-dependent kinetic friction coefficient`.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Extrapolation method — Extrapolation method
`Linear` (default) | `Nearest` | `Error`

Extrapolation method for determining the output value when the input value is outside the range specified in the argument list:

- `Linear` — Select this option to produce a curve with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region.
- `Nearest` — Select this option to produce an extrapolation that does not go above the highest point in the data or below the lowest point in the data.
- `Error` — Select this option to avoid going into the extrapolation mode when you want your data to be within the table range. If the input signal is outside the range of the table, the simulation stops and generates an error.

For more information on extrapolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Temperature-dependent kinetic friction coefficient` or `Temperature and velocity-dependent kinetic friction coefficient`.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Clutch teeth mean radius — Clutch teeth mean radius
50 mm (default) | positive scalar

Distance from the ring or hub center to the corresponding tooth center. The mean tooth radius determines the normal contact forces between ring and hub teeth given the transmission torque between the two components. The value must be greater than zero.

Number of teeth — Number of ring or hub teeth
6 (default) | positive nonzero scalar integer

Total number of teeth in the ring or the hub. The two components have equal tooth numbers. The value must be greater than or equal to one.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Fixed kinetic friction coefficient` or `Velocity-dependent kinetic friction coefficient` and, in the **Dog Clutch** settings, the **Torque transmission model** parameter is set to `Dynamic with backlash`.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Rotational backlash — Rotational backlash
10 deg (default) | positive scalar

Allowable angular motion, or play, between the ring and hub teeth in the engaged clutch configuration. The value must be greater than zero.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to Fixed kinetic friction coefficient or Velocity-dependent kinetic friction coefficient and, in the **Dog Clutch** settings, the **Torque transmission model** parameter is set to Dynamic with backlash.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Torsional stiffness — Torsional stiffness
10e6 N*m/rad (default) | positive scalar

Linear torsional stiffness coefficient at the contact interface between the ring and hub teeth. This coefficient characterizes the restoring component of the contact force between the two sets of teeth. Greater stiffness values correspond to greater contact forces. The value must be greater than zero. The default value is 10e6 N*m/rad.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to Fixed kinetic friction coefficient or Velocity-dependent kinetic friction coefficient and, in the **Dog Clutch** settings, the **Torque transmission model** parameter is set to Dynamic with backlash.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Torsional damping — Torsional damping
100 N*m/(rad/s) (default) | positive scalar

Linear torsional damping coefficient at the contact interface between the ring and hub teeth. This coefficient characterizes the dissipative component of the contact force between the two sets of teeth. Greater damping values correspond to greater energy dissipation during contact. The value must be greater than zero.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to Fixed kinetic friction coefficient or Velocity-dependent kinetic friction coefficient and, in the **Dog Clutch** settings, the **Torque transmission model** parameter is set to Dynamic with backlash.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Tooth-tooth friction coefficient — Tooth-tooth contact kinetic friction coefficient
0.05 (default) | positive scalar

Kinetic friction coefficient at the contact interface between ring and hub teeth. This coefficient characterizes the dissipative force that resists shift linkage motion due to tooth-tooth contact during clutch engagement/disengagement.

Greater coefficient values correspond to greater energy dissipation during shift linkage motion. The value must be greater than zero.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Fixed kinetic friction coefficient` or `Velocity-dependent kinetic friction coefficient` and, in the **Dog Clutch** settings, the **Torque transmission model** parameter is set to `Dynamic with backlash`.

For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Detent

Peak force — Peak shear force
`500 N` (default) | nonnegative scalar

Peak shear force of the detent.

Notch width — Notch width
`3 mm` (default) | nonnegative scalar

Width of the region where the detent exhibits shear force.

Viscous friction coefficient — Viscous friction coefficient
`0.1 N/(m/s)` (default) | nonnegative scalar

Viscous friction coefficient at the contact surface of the detent. The value must be greater than or equal to zero.

Friction to peak force ratio — Friction to peak force ratio
`0.01` (default) | nonnegative scalar

Ratio of the kinetic friction to the peak shear force of the detent. The parameter is used to set the value of the kinetic friction. The parameter must be greater than or equal to zero.

Friction velocity threshold — Kinetic friction velocity threshold
`0.05 m/s` (default)

Velocity required for peak kinetic friction at the contact surface of the detent. The parameter ensures the force is continuous when the travel direction changes, increasing the numerical stability of the simulation. The parameter must be greater than zero. The default value is `0.05 m/s`.

Shift Linkage

Maximum engagement speed — Engagement speed upper velocity threshold
`inf` (default) | positive scalar

Relative angular velocity between the ring and the hub above which the clutch cannot engage. The value is specific to the specific gearbox or transmission. Minimizing the value helps avoid high dynamic impact during engagement. The value must be greater than zero.

Tooth overlap to engage — Tooth overlap engagement threshold
`3 mm` (default) | positive scalar

Overlap length between ring and hub teeth along the common longitudinal axis above which the clutch can engage. The clutch remains disengaged until the tooth overlap by at least this length. The value must be greater than zero.

Tooth height — Tooth height
10 mm (default) | positive scalar

Distance between the base and crest of a tooth. Ring and hub teeth share the same height. The tooth height and the ring-hub clearance when fully disengaged determine the maximum travel span of the shift linkage. The value must be greater than zero.

Ring-hub clearance when disengaged — Ring-hub clearance when disengaged
3 mm (default) | positive scalar

Maximum open gap between the ring and hub tooth crests along the shift linkage translation axis. This gap corresponds to the fully disengaged clutch state. The tooth height and the ring-hub clearance when fully disengaged determine the maximum travel span of the shift linkage. The value must be greater than zero.

Hard stop at back of shift linkage — Hard stop model
On (default) | Off

Hard stop that prevents the shift linkage from traveling beyond the fully disengaged position:

- On — Hard stop when fully disengaged.
- Off — No hard stop when fully disengaged.

Dog clutch ring stop stiffness — Dog clutch ring stop stiffness
10e5 N/m (default) | positive scalar

Stiffness of the hard stops on both sides of the dog clutch ring. The model assumes the ring and stops behave elastically. Contact deformation is proportional to the applied force and the reciprocal of the contact stiffness. The value of the stiffness must be assigned with reference to the parameter **Tooth overlap to engage**. Too low a stiffness could cause the deformation to exceed the required overlap and initiate a false engagement. The parameter must be greater than zero.

Cone clutch ring stop stiffness — Cone clutch ring stop stiffness
10e5 N/m (default) | positive scalar

Stiffness of the hard stops on both sides of the cone clutch ring. The model assumes the ring and stops behave elastically. Contact deformation is proportional to the applied force and the reciprocal of the contact stiffness.

Dog clutch ring stop damping — Dog clutch ring stop damping
1e3 N/(m/s) (default) | nonegative scalar

Translational contact damping between the dog clutch ring and the hub. The value of the damping is inversely proportional to the number of oscillations that occur after impact. The parameter must be greater than zero.

Cone clutch ring stop damping — Cone clutch ring stop damping
1e3 N/(m/s) (default) | nonegative scalar

Translational contact damping between the cone clutch ring and the hub. The value of damping is inversely proportional to the number of oscillations that occur after impact. The parameter must be greater than zero.

Shift linkage viscous friction coefficient — Shift linkage viscous friction coefficient
100 N/(m/s) (default) | positive scalar

Viscous friction coefficient for the relative translational motion between the hub and the ring. The value of the parameter depends on lubrication state and quality of contacting surfaces. The coefficient must be greater than or equal to zero.

Initial Conditions

Initial state — Initial clutch state
All clutches unlocked (default) | Cone clutch A and dog clutch A locked | Cone clutch A locked | Cone clutch B locked | Cone clutch B and dog clutch B locked

Beginning configuration of cone and dog clutches:

- Cone clutch A and dog clutch A locked — Cone clutch *A* and dog clutch *A* transmit torque between the ring and hub shafts.

- Cone clutch A locked — Cone clutch *A* transmits torque between the ring and hub shafts.

- All clutches unlocked— Cone and dog clutches transmit zero torque between the ring and hub shafts.

- Cone clutch B locked — Cone clutch *B* transmits torque between the ring and hub shafts.

- Cone clutch B and dog clutch B locked — Cone clutch *B* and dog clutch *B* transmit torque between the ring and hub shafts.

Dog clutch initial shift linkage position — Dog clutch initial shift linkage position
0 mm (default) | scalar

Initial position of the shift linkage section that attaches to the dog clutch. The value of the parameter has these restrictions:

Dog Clutch State	Parameter Restriction
Dog clutch A Initially engaged	Negative of the parameter value must be greater than the sum of parameters **Ring-hub clearance when dog clutch disengaged** and **Tooth overlap to engage**
Dog clutch A Initially disengaged	Negative of the parameter value must be smaller than the sum of parameters **Ring-hub clearance when dog clutch disengaged** and **Tooth overlap to engage**
Dog clutch B Initially engaged	Parameter value must be greater than the sum of parameters **Ring-hub clearance when dog clutch disengaged** and **Tooth overlap to engage**

Dog Clutch State	Parameter Restriction
Dog clutch B Initially disengaged	Parameter value must be smaller than the sum of parameters **Ring-hub clearance when dog clutch disengaged** and **Tooth overlap to engage**

Cone clutch initial shift linkage position — Cone clutch initial shift linkage position
0 mm (default) | scalar

Initial position of the shift linkage section that attaches to the cone clutch. The value of the parameter has these restrictions:

Cone Clutch State	Parameter Restriction
Cone clutch A initially engaged	Negative of parameter must be greater than the value of **Ring-hub clearance when cone clutch disengaged**
Cone clutch A initially disengaged	Negative of parameter must be smaller than the value of **Ring-hub clearance when cone clutch disengaged**
Cone clutch B initially engaged	Parameter must be greater than the value of **Ring-hub clearance when cone clutch disengaged**
Cone clutch B initially disengaged	Parameter must be smaller than the value of **Ring-hub clearance when cone clutch disengaged**

Initial dog clutch A ring-hub offset angle — Initial dog clutch *B* ring-hub offset angle
0 deg (default) | scalar

Rotation angle between the ring and the hub of dog clutch *B* at simulation time zero. This angle determines whether the ring and hub teeth can interlock, and hence whether the clutch can engage. The initial offset angle must satisfy these conditions:

- If the clutch initial state is disengaged, the initial offset angle must fall in the range

$$-\frac{180°}{N} \le \phi_0 \le +\frac{180°}{N},$$

  where N is the number of teeth present in the ring or the hub. The two components contain the same number of teeth.

- If the clutch initial state is engaged, the initial offset angle must fall in the range

$$-\frac{\delta}{2} \le \phi_0 \le +\frac{\delta}{2},$$

  where δ is the backlash angle between the ring and hub teeth.

Dependencies

This parameter is only visible when both of these conditions are met:

- In the **Cone Clutch** settings, the **Friction model** parameter is set to `Fixed kinetic friction coefficient` or `Velocity-dependent kinetic friction coefficient`.
- In the **Dog Clutch** settings, the **Torque transmission model** is set to `Dynamic with backlash`.

`Initial dog clutch B ring-hub offset angle` — Initial dog clutch *A* ring-hub offset angle
`0 deg` (default) | scalar

Rotation angle between the ring and the hub at simulation time zero. This angle determines whether the ring and hub teeth can interlock, and hence whether the clutch can engage. The initial offset angle must satisfy these conditions:

- If the clutch initial state is disengaged, the initial offset angle must fall in the range

$$-\frac{180°}{N} \le \phi_0 \le +\frac{180°}{N},$$

  where *N* is the number of teeth present in the ring or the hub. The two components contain the same number of teeth.

- If the clutch initial state is engaged, the initial offset angle must fall in the range

$$-\frac{\delta}{2} \le \phi_0 \le +\frac{\delta}{2},$$

  where *δ* is the backlash angle between the ring and hub teeth.

Dependencies

This parameter is only visible when both of these conditions are met:

- In the **Cone Clutch** settings, the **Friction model** parameter is set to `Fixed kinetic friction coefficient` or `Velocity-dependent kinetic friction coefficient`.
- In the **Dog Clutch** settings, the **Torque transmission model** is set to `Dynamic with backlash`.

Thermal Port

Thermal Port settings are visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

`Thermal mass` — Thermal mass
`50 kJ/K` (default) | scalar

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

This parameter is only visible when, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Initial temperature — Initial temperature
300 K (default) | scalar

Component temperature at the start of simulation. The initial temperature alters the component efficiency according to an efficiency vector that you specify, affecting the starting meshing or friction losses.

Dependencies

This parameter is only visible when, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see Double-Sided Synchronizer Block Parameter Dependencies.

Thermal Port

These thermal parameters are only visible when you select a temperature-dependent friction model.

Thermal mass

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change. The default value is `100 kJ/K`.

Initial temperature

Component temperature at the start of simulation. The initial temperature alters the component efficiency according to an efficiency vector that you specify, affecting the starting meshing or friction losses. The default value is `300 K`.

More About

Linearization

To optimize your model for linearization, use the **Dog Clutch > Torque transmission model** parameter default setting, `Friction clutch approximation - Suitable for HIL and linearization`.

Hardware-in-the-Loop Simulation

For optimal simulation performance, use the **Dog Clutch > Torque transmission model** parameter default setting, `Friction clutch approximation - Suitable for HIL and linearization`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Cone Clutch | Dog Clutch | Synchronizer | Translational Detent

Topics
"Model Thermal Losses in Driveline Components"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2012b

Flexible Shaft

Shaft with torsional and bending compliance
Library: Simscape / Driveline / Couplings & Drives

Description

The Flexible Shaft block represents a driveline shaft with torsional and bending compliance. The shaft consists of a flexible material that twists in response to an applied torque and bends in response to static mass unbalance. The twisting action delays power transmission between the shaft ends, altering the dynamic response of the driveline system.
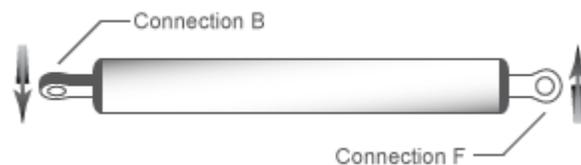
To represent a torsion-flexible shaft, the block uses a lumped mass method. This model divides the shaft into different elements that interconnect through parallel spring damper systems. The elements provide the shaft inertia while the spring damper systems provide the shaft compliance.

The block provides four parameterization methods that allow you to model compliance in either a homogeneous or an axially inhomogeneous shaft. An axially inhomogeneous shaft is one for which any of these attributes vary along the length of the shaft:

- Torsional stiffness
- Torsional inertia
- Bending rigidity
- Density
- Shear modulus
- Young's modulus
- Outer diameter
- Inner diameter

An additional parameter enables you to model the power losses in the bearings due to viscous friction at the shaft ends. For more information, see "Torsion Model" on page 1-177.

Note The viscous friction at the shaft ends is different from the internal material damping, which corresponds to losses arising in the shaft material itself.

To represent the bending-flexible shaft, the block uses either a lumped mass method or an eigenmodes method. While the lumped mass method is simpler to configure, the eigenmodes method tends to simulate faster.

Tip If simulation speed, is a high priority, first simulate using the lumped mass method, adjusting parameters as needed until the results match your mathematical models or experimental data. Next, simulate using the eigenmodes method. Again, adjust the parameters as needed until the results mathematical models or experimental data. For an example that uses both methods, see "Shaft with Torsional and Transverse Flexibility".

For the lumped mass method, the number of bending shaft elements is the same as the number of torsion shaft elements. The model divides the shaft into a series of such elements. The elements provide the shaft inertia, while the stiffness matrices provide the shaft compliance. The eigenmodes method computes effective mass-spring-damper systems that represent the bending modes of the shaft. You can specify the number of modes to include and the precision of the mode shapes. Both the lumped mass and eigenmodes methods allow you to model:

- Excitational static unbalances
- Concentrically attached rigid masses
- Up to four support locations along the shaft
- Linear damping proportional to shaft inertia
- Linear damping proportional to shaft stiffness

Note The eigenmodes method assumes that the support damping is light compared to support stiffness.

Static unbalances, which excite bending, occur when the center of mass of the shaft or an attached rigid mass is not aligned with the shaft principal axis. You can vary the locations, magnitudes, and angle offsets of static unbalances on the shaft.

You can represent concentrically attached rigid masses as disks or idealized point masses. A concentric disk adds diametric and polar moments of inertia to the shaft and mass to the translation degree of freedom of the shaft nodes. The model assumes that the disk is thin, so the shaft can still bend on either side of the axial location with the disk. The polar moment of inertia couples the two planes of bending. A concentric point mass is an idealized version of a concentric disk. A concentric point mass adds mass to the translation degrees of freedom of the shaft nodes but does not have rotational moments of inertia. You can vary the locations and inertias of concentric disks or point masses that are attached to the shaft.

You can model the supports as ideal or by using stiffness and damping matrices. For each support, you can vary the:

- Location — Any point along the shaft length.
- Type — Ideal clamp, ideal pin, free, constant bearing stiffness and damping, or speed-dependent stiffness and damping.
- Number — Two, three, or four.

For both bending methods, you can specify the shaft bending compliance by using either bending rigidity and linear mass density or Young's modulus and shaft diameter.

You can parameterize the torsional model by using either stiffness k and inertia J or the dimensions and material properties of the shaft.

Torsion Model

For the torsion model, the Flexible Shaft block approximates the distributed, continuous properties of a shaft by using a lumped mass method. The model contains a finite number, N, of lumped inertia-damped spring elements in series, plus a final inertia. The result is a series of $N + 1$ inertias connected by N rotational springs and N rotational dampers.
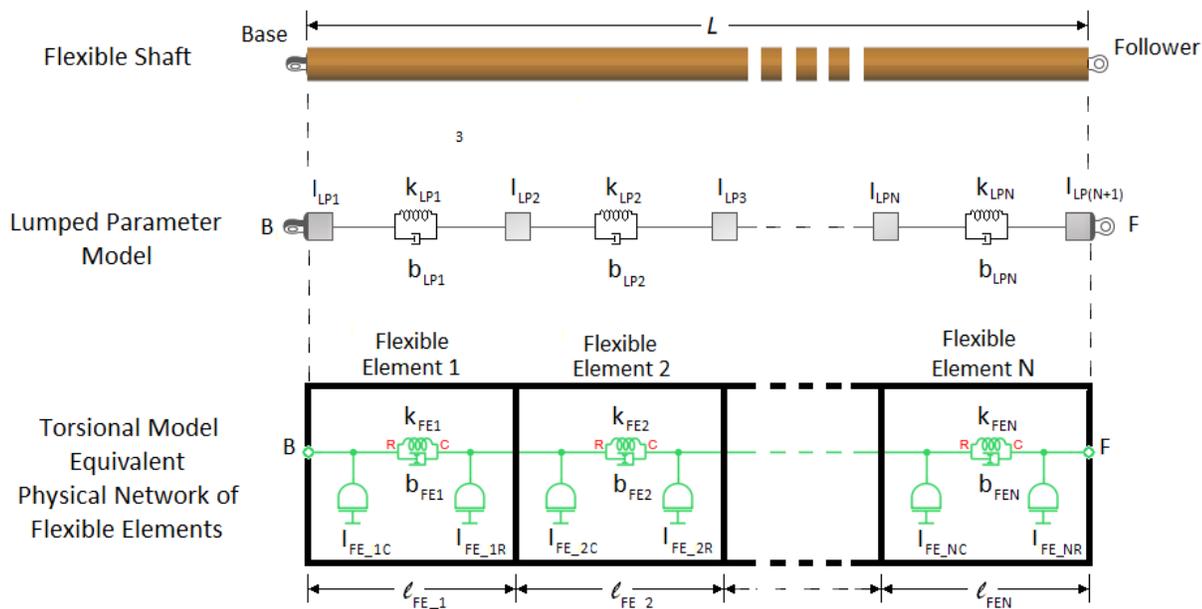
The block models the shaft as an equivalent physical network of N flexible elements. Each flexible element, FE_i, represents a short section of the driveshaft and contains:

- One spring, k_{FE_i}, for torsional compliance. The network has a total of N springs.
- One damper, b_{FE_i}, for material damping. The network has a total of N dampers.
- Two inertias, I_{FE_iC} and I_{FE_iR}, for rotational resistance. The inertias of neighboring flexible elements are consolidated together so that the network has a total of $N + 1$ inertias.

For an axially homogeneous shaft, the flexible element lengths, compliance, damping, and distributed inertias in the physical network are equal, such that:

$$l_{FE\_1} = l_{FE\_2} = \cdots = l_{FE\_N} = \frac{L}{N}$$

$$k_{FE\_1} = k_{FE\_2} = \cdots = k_{FE\_N} = k$$

$$b_{FE\_1} = b_{FE\_2} = \cdots = b_{FE\_N} = b$$

$$I_{FE\_1C} = I_{FE\_1R} = I_{FE\_2C} = I_{FE\_2R} = \cdots = I_{FE\_NC} = I_{FE\_NR} = \frac{I}{2N}$$

For an axially inhomogeneous shaft, the amount of compliance, damping, and **R**-node and **C**-node inertia can differ for individual flexible elements in the physical network model.

Node Placement Algorithm

The balance between model fidelity and simulation speed depends on N, the number of flexible elements that the block uses to represent the shaft. For information on balancing simulation speed and model fidelity, see "Improve Simulation Speed or Accuracy" on page 1-193.

The block allows you to specify a minimum number of flexible elements, N_{min}, as the value for the "Minimum number of flexible elements" on page 1-0   parameter. However, the number of flexible elements that the block actually uses depends on the complexity of the shaft that it is modeling. If the block requires more flexible elements than you specify to solve a model that contains axial inhomogeneity, intermediate supports, concentric disks or masses, or static unbalances, then $N \geq N_{min}$.

For example, suppose that, for the complex shaft in the diagram, you specify axial locations for the supports, static unbalance, larger diameter section, and concentric disk. You set the parameter for N_{min} to 7.
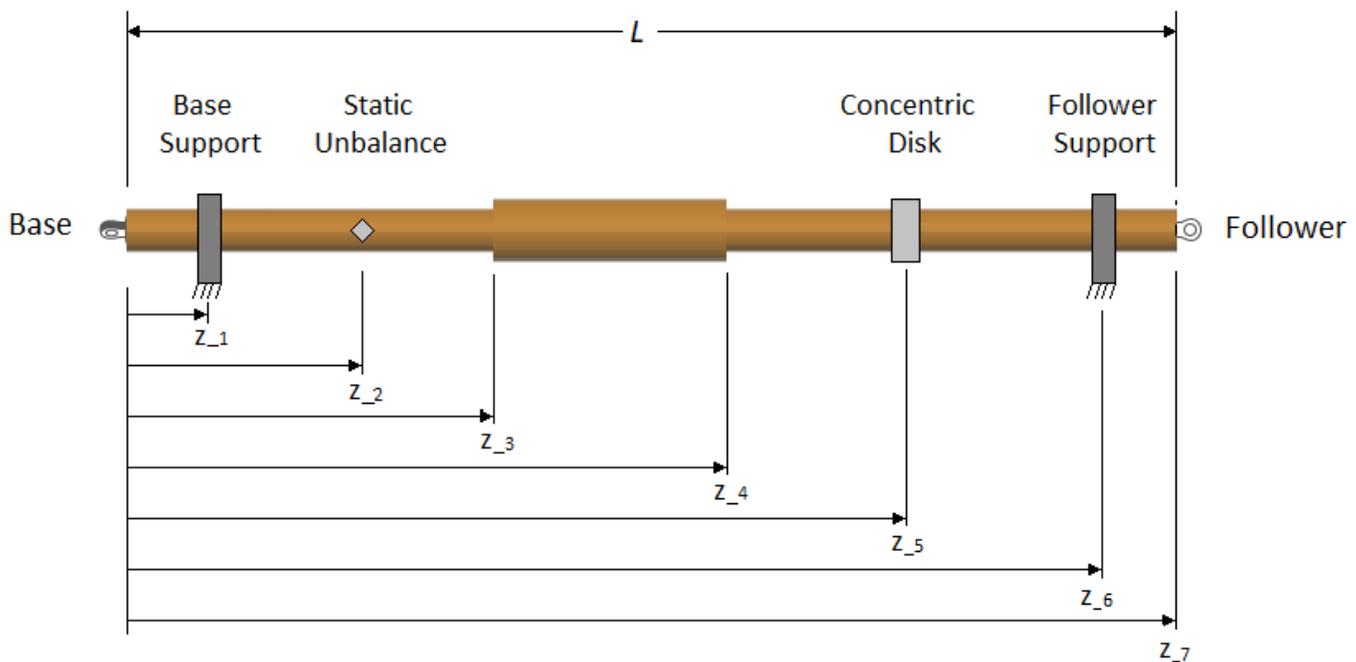
If model bending is on, the torsion model flexible element locations account for the locations of static unbalances and concentric rigid masses, so that the torsion flexible elements align with the bending flexible elements. During simulation, the torsion model is independent of any static unbalances or concentric rigid masses.

The algorithm for the block determines the number of flexible elements and the length of the individual elements that are required to solve the simulation:

1   The block places one node at the base and follower ends of the shaft. These nodes are considered *fixed* in axial location because they represent physical entities along the shaft axis. In the diagram, fixed nodes are shown in red. The block evenly distributes the other five (N_{min}-2)

internal nodes along the length of the shaft. It then places a flexible element between each consecutive pair of nodes.
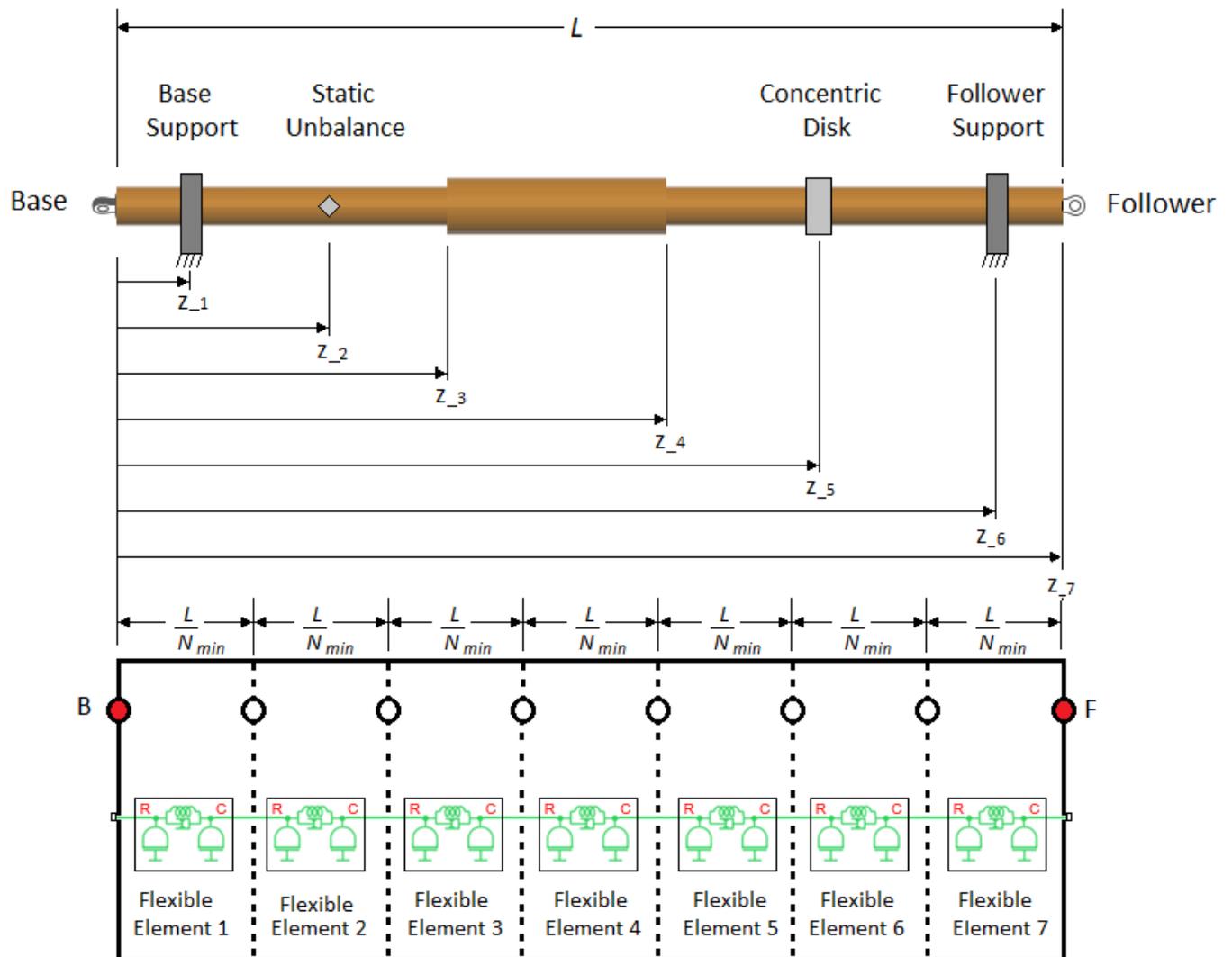
For an end-supported, axially homogenous shaft, with no static unbalances or attached concentric disks, depending on the other parameter options and values that you specify, the block might be able to solve the simulation using only N_{min} flexible elements of equivalent lengths:

$$l = \frac{L}{N_{min}}$$

In most cases, however, the block can only solve the simulation if it adds more flexible elements.

2 To add more flexible elements, the block places fixed internal nodes at these locations:

- Each shaft support location. The block allows you to specify the number and location of shaft supports. For the shaft in the diagram, there are supports at z_1 and z_6.
- Each static unbalance. For the shaft in the diagram, there is a static unbalance at z_2.

- Each rigid mass. Rigid masses are concentrically attached disks or point masses. For the shaft in the diagram, there is a rigid mass, represented as a disk, at z_5.
- Each parameterization segment boundary. *Parameterization boundaries* are locations along an axially inhomogeneous shaft where two neighboring sections of the shaft vary in stiffness, inertia, or geometry. The block allows you to define the parameterization segment boundary locations. For the shaft in the diagram, there are segment boundaries at z_3 and z_4.
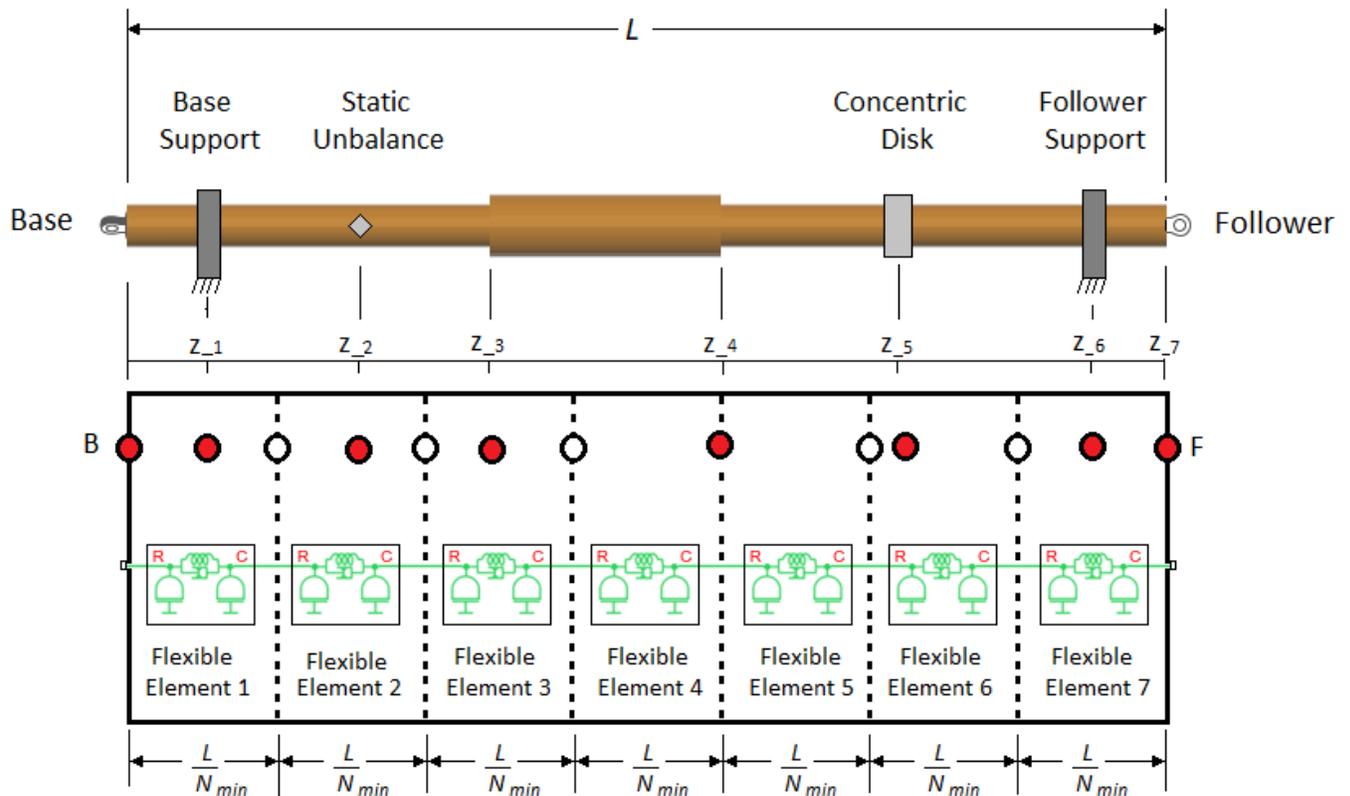
Note that the block did not add a node at z_4 because a node was already added in the previous step of the algorithm. However, the node is now fixed because it represents a physical entity along the shaft length.

3   The block adjusts the nonfixed node locations between the fixed nodes so that they are evenly distributed.
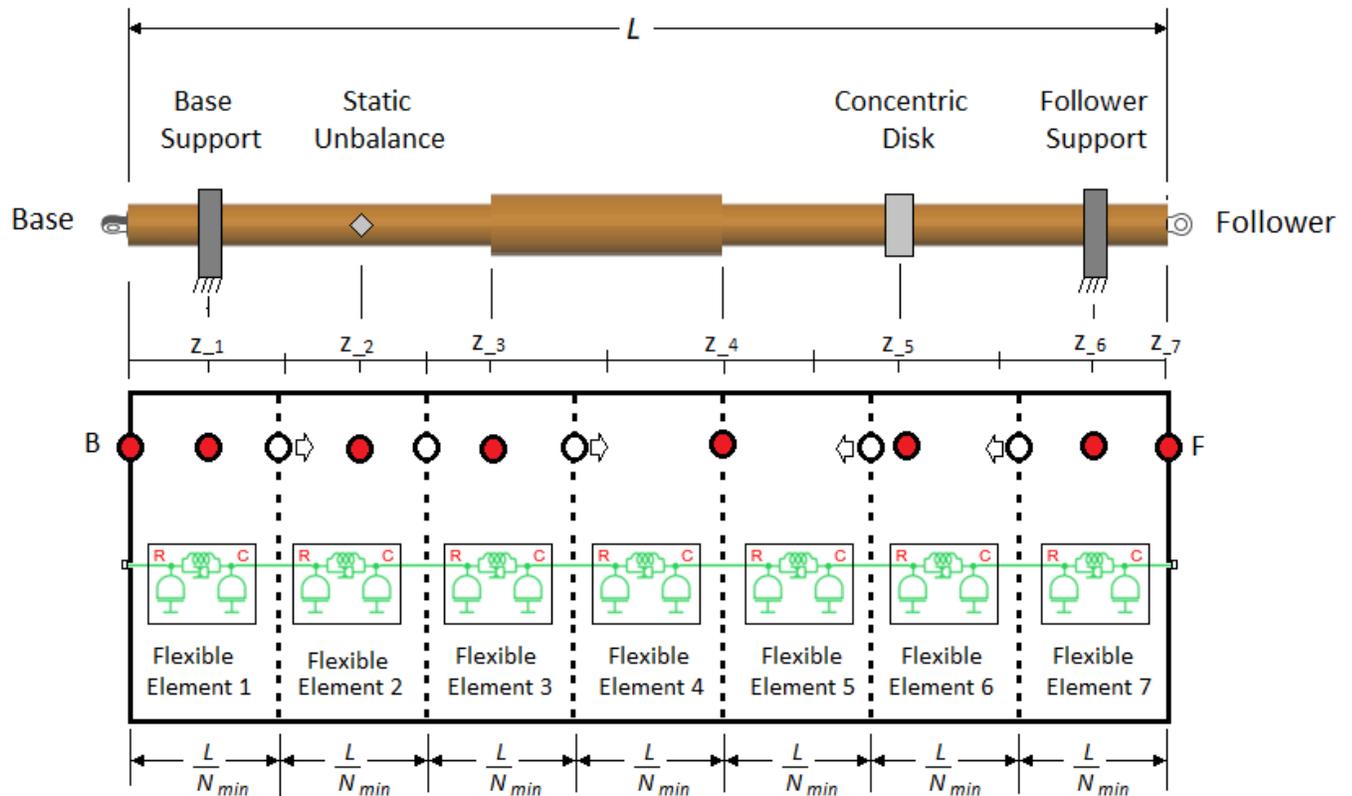
Finally, the block places flexible elements between each node. The length of each flexible element corresponds to the center-to-center distances between the neighboring nodes. The block distributes the inertia among the flexible elements based on the length of the individual element and the corresponding shaft geometry. Ultimately, this complex shaft is represented by 12 flexible elements, with $l_1 = z_1$, $l_2 = l_3 = \dfrac{(z_2 - z_1)}{2}$, $l_4 = l_5 = \dfrac{(z_3 - z_2)}{2}$, $l_6 = l_7 = \dfrac{(z_4 - z_3)}{2}$, $l_8 = l_9 = \dfrac{(z_5 - z_4)}{2}$, $l_{10} = l_{11} = \dfrac{(z_6 - z_5)}{2}$, and $l_{12} = z_7 - z_6$.
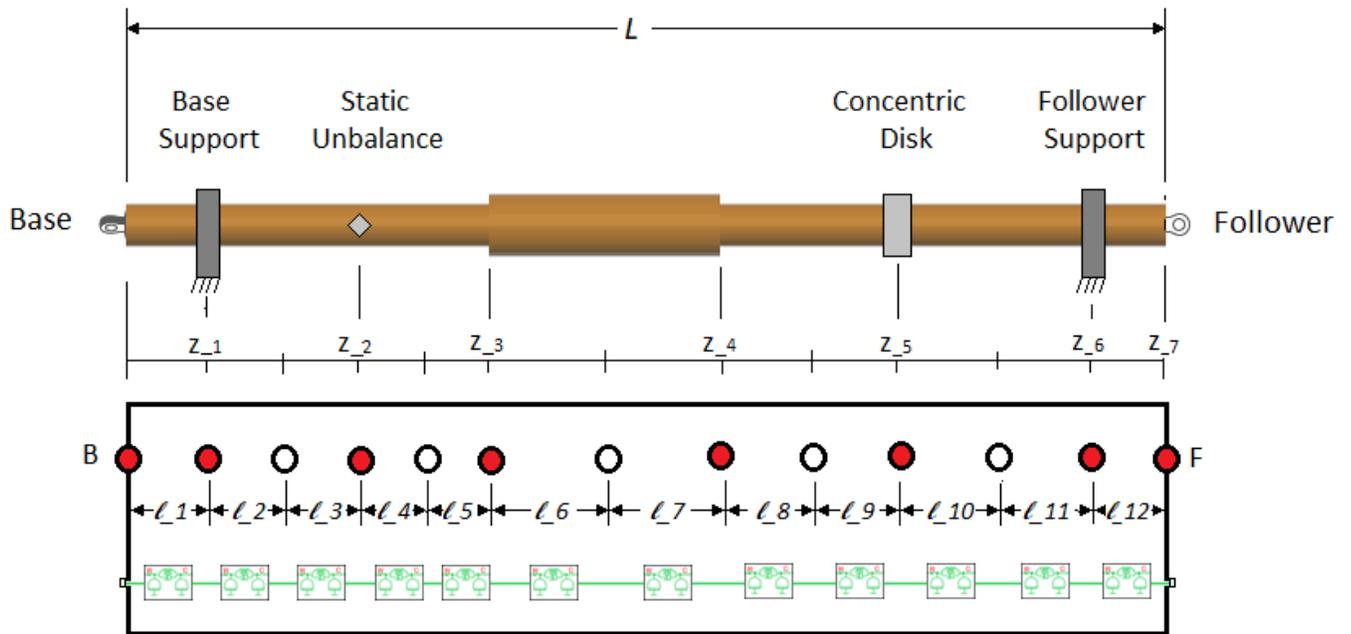
If N_{min} is large enough to yield a number of unfixed nodes that is greater than the number of fixed nodes, the block distributes more than one unfixed node between each set of neighboring fixed nodes.

Dimensions and Material Properties

You can parameterize the torsion model by using either stiffness, k, and the polar moment of inertia, J, or the dimensions and material properties of the shaft.

The stiffness and inertia for each element are computed from the shaft dimensions and material properties as:

$$J_p = \frac{\pi}{32}\left(D^4 - d^4\right)$$

$$m = \frac{\pi}{4}\left(D^2 - d^2\right)\rho l$$

$$J = \frac{m}{8}\left(D^2 + d^2\right) = \rho l \cdot Jp$$

$$k = Jp \cdot \frac{G}{l}$$

where:

- J_P is the polar moment of inertia of the shaft at the flexible element location.
- D is the outer diameter of the shaft at the flexible element location.
- d is the inner diameter of the shaft at the flexible element location. For a solid shaft, $d = 0$. For an annular shaft, $d > 0$.

- l is the flexible element length.
- m is the mass of the shaft at the flexible element location.
- J is the moment of inertia of the shaft at the flexible element location.
- ρ is the density of the shaft material.
- G is the shear modulus of elasticity of the shaft material.
- k is the rotational stiffness of the flexible element.

Internal Material Damping

For either torsional parameterization, the internal material damping is defined by the damping ratio, c, for a single-flexible element model with the equivalent torsional stiffness and inertia. The damping coefficient is then $2\dfrac{ck}{\omega_N}$, where the undamped natural frequency is $\omega_N = \sqrt{\dfrac{2k}{J}}$. The damping torque applied across an individual flexible element of a lumped mass model is equivalent to the product of the damping coefficient and the relative rotational velocity of that flexible element.

Bending Models

The "Shaft Geometry, Support Loading, and Motion" on page 1-184 figure shows how to measure:

- The static unbalance offset angle, which is the angle of a static unbalance about the shaft axis relative to the x axis
- The distances of a support, a rigid mass, and a static unbalance, relative to the base end of the shaft, **B**
- The parameterization of the segment lengths

In the figure, the shaft has three fixed supports:

1 B_1 — Base end support
2 I_1 — Intermediate support
3 F_1 — Follower end support

The shaft has translational velocity V, rotational velocity W, and exerts forces F, and moments M, on the supports. The curved arrows and sign conventions follow the right-hand rule. The signs of the physical signals that the block outputs correspond to the arrows that represent the forces, moments, and velocities of the shaft acting on the supports.

The vector signals are:

- Force, $Fr = \left[F_{xB1}, F_{yB1}, F_{xI1}, F_{yI1}, F_{xF1}, F_{yF1}\right]$
- Moment, $M = \left[M_{xB1}, M_{yB1}, M_{xI1}, M_{yI1}, M_{xF1}, M_{yF1}\right]$
- Translational velocity, $V = \left[V_{xB1}, V_{yB1}, V_{xI1}, V_{yI1}, V_{xF1}, V_{yF1}\right]$
- Rotational velocity, $M = \left[M_{xB1}, M_{yB1}, M_{xI1}, M_{yI1}, M_{xF1}, M_{yF1}\right]$

If the shaft has two supports, each vector signal has a length of four. Force, for example, is then $Fr = \left[F_{xB1}, F_{yB1}, F_{xF1}, F_{yF1}\right]$.

If the shaft has four supports, each vector signal has a length of eight. Force, for example, is then $Fr = \left[F_{xB1}, F_{yB1}, F_{xI1}, F_{yI1}, F_{xI2}, F_{yI2}, F_{xF1}, F_{yF1}\right]$.
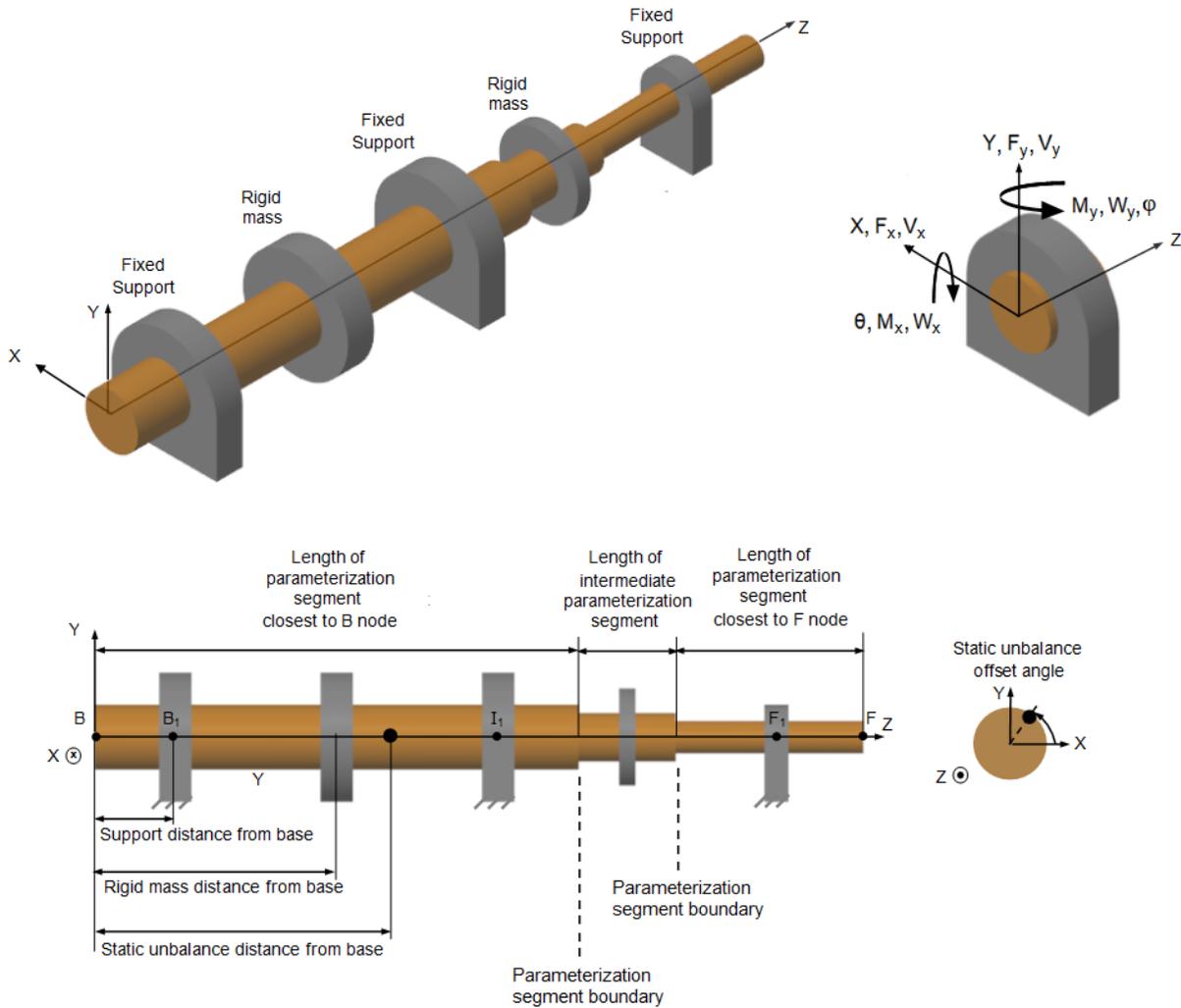
Shaft Geometry, Support Loading, and Motion

Bending Model Lumped Mass Method

Like the torsion model, the lumped mass method for the bending model discretizes the distributed, continuous properties of the shaft into a finite number, N, of flexible elements. The N flexible elements correspond to $N + 1$ lumped inertias connected in series by damping and spring elements. However, for the bending model, each mass has four degrees of freedom: translation and rotation in both the x and y directions perpendicular to the shaft axis.
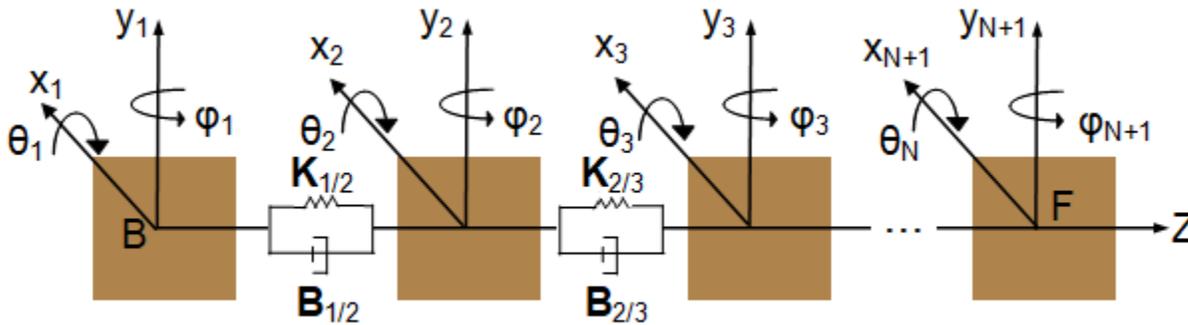
The lumped mass equation of motion [1] is

$$M\ddot{\vec{x}} + (B + G_{Disk}\Omega)\dot{\vec{x}} + \left(K + G_{Disk}\dot{\Omega}\right)\vec{x} = \vec{f}.$$

where:

- M is the $4(N + 1) \times 4(N + 1)$ matrix that represents the mass of the shaft.
- B is the $4(N + 1) \times 4(N + 1)$ matrix for the internal damping and support damping.
- G_{Disk} is the $4(N + 1) \times 4(N + 1)$ matrix that accounts for disk gyroscopics
- Ω is the shaft torsional velocity during simulation.
- K is the $4(N + 1) \times 4(N + 1)$ matrix for the spring stiffness.
- $\vec{x}$ is the $4(N + 1) \times 1$ vector that represents the degrees of freedom for all nodes.
- $\vec{f}$ is the $4(N + 1) \times 1$ vector that represents external forces due to the application of static mass unbalance.

The equation for the mass matrix [4] is

$$M = M_{1/2} + M_{2/3} + ...M_{i/i + 1} + ...M_{N/N + 1} \; + \; \sum M_{disk, \, i},$$

where:

- $M_{i/(i + 1)}$ is the mass matrix for an individual flexible element. For each flexible element, half of the mass and moment of inertia is transferred to the nodes at both ends of the flexible element. The $M_{i/(i + 1)}$ matrix has nonzero elements in the $(4i - 3):(4i + 4)$ rows and the $(4i - 3):(4i + 4)$ columns:

$$M_{i/(i+1)} = \begin{bmatrix} 0 & & & & & & & & \\ & \ddots & & & & & & & \\ & & \frac{1}{2}m & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & 0 & \frac{1}{2}m & 0 & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & I_d & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & I_d & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 & \frac{1}{2}m & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 & 0 & \frac{1}{2}m & 0 & 0 \\ & & 0 & 0 & 0 & 0 & 0 & 0 & I_d & 0 \\ & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_d \\ & & & & & & & & & \ddots \\ & & & & & & & & & & 0 \end{bmatrix},$$

where:

- l is the flexible element length along the shaft between internal nodes. To determine the length of each flexible element, the block uses the algorithm that is described in "Node Placement Algorithm" on page 1-178. Each flexible element contains two inertias. Each inertia has two translational degrees of freedom, two rotational degrees of freedom, and one stiffness matrix.

  Each flexible element in the equivalent physical model for bending in the *XZ*-plane (the beam translation in the *X*-direction and rotation about the *Y*-axis) and in the physical model for bending in the *YZ*-plane (the beam translation in the Y-direction and rotation about the X-axis) then contains two masses, two inertias, and a stiffness matrix.

To determine the internal node locations, and therefore the number and lengths of the flexible elements, the block uses the same node-placement algorithm as it uses for the torsion model. For more information, see "Node Placement Algorithm" on page 1-178.

- m is the flexible element mass. m depends on the outer, D, and inner, d, diameters, the density, ρ, of the shaft and the length of the flexible element, such that $m = \left(\frac{\pi}{4}\right)\left(D^2 - d^2\right)\rho l$.

- I_d, the half-element mass moment of inertia about an axis perpendicular to the shaft axis, depends on the mass, m, length, l, and torsion moment of inertia, J, of the flexible element, such that $I_d = \frac{J}{4} + \frac{m}{6}\left(\frac{l}{2}\right)^2$.

- $\sum M_{disk,\ i}$ is the summed mass matrices of the rigid masses concentrically attached to the shaft.

- The mass properties of each rigid mass that is concentrically attached to the shaft are added to the closest node, i, such that

$$M_{disk,\ i}([(4i-3):4i],[(4i-3):4i]) = \begin{bmatrix} M_{disk,\ i} & 0 & 0 & 0 \\ 0 & M_{disk,\ i} & 0 & 0 \\ 0 & 0 & I_{D,\,disk,\ i} & 0 \\ 0 & 0 & 0 & I_{D,\,disk,\ i} \end{bmatrix},$$

where $I_{D,disk,i}$ is the mass diametric moment of inertia about an axis perpendicular to the shaft for a rigid disk attached to the ith node. The model assumes that the disk is thin, so the shaft can still bend on either side of the axial location with the disk. A concentric point mass has $I_{D,\,disk,\ i} = 0$.

The equation for the damping matrix is

$$B = \alpha M + \beta K + B_{support},$$

where:

- α is the damping constant proportional to mass.

- β is the damping constant proportional to stiffness.

- $\boldsymbol{B}_{support}$ is the damping coefficient at each support. For a support at the ith node, the damping matrix, in terms of global coordinates, is

$$B_{support}([(4i-3):4i],[(4i-3):4i]) = \begin{bmatrix} b_{xx} & b_{xy} & 0 & 0 \\ b_{yx} & b_{yy} & 0 & 0 \\ 0 & 0 & b_{\theta\theta} & 0 \\ 0 & 0 & 0 & b_{\varphi\varphi} \end{bmatrix},$$

where:

- $\begin{bmatrix} b_{xx} & b_{xy} & b_{yx} & b_{yy} \end{bmatrix}$ is the support translational damping.

- $\begin{bmatrix} b_{\theta\theta} & b_{\varphi\varphi} \end{bmatrix}$ is the support rotational damping.

- $G_{disk,\ i}$ accounts for the gyroscopic effects of any concentrically attached disks, and is defined as

$$G_{disk,\ i}([(4i-3):4i],[(4i-3):4i]) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Omega I_{P,\,disk,\ i} \\ 0 & 0 & -\Omega I_{P,\,disk,\ i} & 0 \end{bmatrix},$$

where $I_{P,disk,i}$ is the mass polar moment of inertia about the shaft axis for the disk attached to the ith node. The mass polar moment of inertia for a concentric point mass is $I_{P,\,disk,\,i} = 0$.

The equation for the bearing stiffness matrix is

$$K = K_{1/2} + K_{2/3} + \ldots + K_{N/N+1} + \sum K_{\text{support}},$$

where:

- $K_{i/i+1}$ is the stiffness matrix for an individual shaft flexible element. The stiffness matrix for the ith shaft flexible element, between the ith and the $(i+1)^{th}$ nodes, has nonzero elements in the $(4i-3):(4i+4)$ rows and the $(4i-3):(4i+4)$ columns, such that

$$K_{i/i+1} = \frac{2EI}{l^3}
\begin{bmatrix}
0 & & & & & & & & & \\
& \ddots & & & & & & & & \\
& & 6 & 0 & 0 & 3l & -6 & 0 & 0 & 3l \\
& & 0 & 6 & -3l & 0 & 0 & -6 & -3l & 0 \\
& & 0 & -3l & 2l^2 & 0 & 0 & 3l & l^2 & 0 \\
& & 3l & 0 & 0 & 2l^2 & -3l & 0 & 0 & l^2 \\
& & -6 & 0 & 0 & -3l & 6 & 0 & 0 & -3l \\
& & 0 & -6 & 3l & 0 & 0 & 6 & 3l & 0 \\
& & 0 & -3l & l^2 & 0 & 0 & 3l & 2l^2 & 0 \\
& & 3l & 0 & 0 & l^2 & -3l & 0 & 0 & 2l^2 \\
& & & & & & & & & \ddots \\
& & & & & & & & & & 0
\end{bmatrix},$$

where:

  - l is the flexible element length.
  - EI is the shaft rigidity.

- $K_{support}$ is the stiffness at each support. For a support at the ith node, the stiffness matrix, in terms of global coordinates, is

$$K_{support}([(4i-3):4i],\,[(4i-3):4i]) =
\begin{bmatrix}
k_{xx} & k_{xy} & 0 & 0 \\
k_{yx} & k_{yy} & 0 & 0 \\
0 & 0 & k_{\theta\theta} & 0 \\
0 & 0 & 0 & k_{\varphi\varphi}
\end{bmatrix},$$

where:

- $\begin{bmatrix} k_{xx} & k_{xy} & k_{yx} & k_{yy} \end{bmatrix}$ is the support translational stiffness.
- $\begin{bmatrix} k_{\theta\theta} & k_{\varphi\varphi} \end{bmatrix}$ is the support rotational stiffness.

The support stiffness matrix, $K_{support}$, is nonzero only if you select `Bearing matrix` or `Speed-dependent bearing matrix` for the support. If you select the `Clamped` mounting type, the

kinematic conditions of zero rotation and translation are applied to the degrees of freedom that correspond to the support node (*B1*, *I1*, *I2*, or *F1*). If you select the `Pinned` mounting type, the kinematic conditions of zero translation are applied to the translational degrees of freedom that correspond to the support node (*B1*, *I1*, *I2*, or *F1*).

The table includes the boundary conditions applied to the lumped mass nodes with supports.

Support Type	Boundary Condition for the Lumped Mass Equation
`Clamped`	$x_i = 0, \quad y_i = 0, \quad \theta_i = 0, \quad \varphi_i = 0$
`Pinned`	$x_i = 0, \quad y_i = 0$
`Bearing Matrix`	$K_{support}$ is nontrivial.
`Speed-dependent bearing matrix`	$K_{support}$ is nontrivial and depends on shaft rotation speed. At each time step, $K_{Support}$ is calculated as: $$K_{Support}(\Omega) = lookup(\ \left|\Omega_{Ref}\right|,\ K_{Support,\ Ref},\ \Omega,\ interpolation = linear,\ extrapolation = nearest),$$ where: • Ω_{Ref} is the bearing speed, as specified, in the **Supports** settings, for the **Bearing speed [s1,...,sS]** parameter. • For each support, $K_{Support,Ref}$ is the bearing speed-dependent translational stiffness, which you specify in the **Supports** settings. • The lookup table uses linear interpolation and nearest extrapolation for the shaft rotation speed.

The matrix that represents the degrees of freedom for all nodes, $\vec{x}$, is calculated such that the degrees of freedom for the ith and the $(i+1)^{th}$ nodes are

$$\vec{x} = \begin{bmatrix} \vdots \\ x_i \\ y_i \\ \theta_i \\ \varphi_i \\ x_{i+1} \\ y_{i+1} \\ \theta_{i+1} \\ \varphi_{i+1} \\ \vdots \end{bmatrix}.$$

External forces due to each static mass unbalance are applied to the closest node. The forcing at the ith node is

$$
\overrightarrow{f}_{4(i-1:i-2)} = \begin{bmatrix} m\varepsilon_{j,\,offset}\left(\Omega^2{}_i\cos(\varphi_{shaft,\,i} + \varphi_{offset,\,j}) & + \frac{\partial\Omega_i}{\partial t} & \sin(\varphi_{shaft,\,i} + \varphi_{offset,\,j}) \right) \\ m\varepsilon_{j,\,offset}\left(\Omega^2{}_i\sin(\varphi_{shaft,\,i} + \varphi_{offset,\,j}) & - \frac{\partial\Omega_i}{\partial t} & \cos(\varphi_{shaft,\,i} + \varphi_{offset,\,j}) \right) \end{bmatrix},
$$

where:

- $m\varepsilon_j$ is the jth static unbalance, located at the ith node.
- Ω_i is the shaft rotational velocity during simulation for the ith node.
- $\varphi_{shaft,\,i}$ is the torsion lumped mass rotation angle for the ith node.

Bending Model Eigenmodes Method

For the eigenmodes method, the block reduces the bending dynamics from the $4(N + 1)$ degrees of freedom that the bending model lumped mass method provides, to M degrees of freedom, where M is the number of modes.

The block computes the bending mode properties of the shaft during model compilation, then solves the modal mass-spring-damper systems during model simulation.

Reducing the degrees of freedom in the model dynamics and separating the calculations into compile-time and run-time tasks improves simulation performance. The eigenmodes method assumes the mode shapes are unaffected by damping. Therefore, the method is best suited to models that include limited disk gyroscopic and support damping.

During compilation, the block computes the approximate damped eigenmodes using these steps:

1   The block computes the matrices using the same lumped mass equation of motion that it uses for the bending model lumped mass method:

$$
M\ddot{\overrightarrow{x}} + (B + G_{Disk}\Omega)\dot{\overrightarrow{x}} + \left(K + G_{Disk}\dot{\Omega}\right)\overrightarrow{x} = \overrightarrow{f} .
$$

For more information, see "Bending Model Lumped Mass Method" on page 1-184.

When determining the node axial locations for $\overrightarrow{x}$, the block uses one of two variations of the "Node Placement Algorithm" on page 1-178 that it uses for the torsion model and the bending model lumped mass method. The variation that the block uses depends on whether, in the **Advanced Bending** settings, the **Bending mode determination** parameter is set to `Simscape determined` or to `User defined`.

If the **Bending mode determination** parameter is set to `Simscape determined`, instead of using the **Minimum number of flexible elements** parameter for N_{min}, as the lumped mass methods do, the eigenmodes method calculates N_{min} as

$$
N_{Min,\,Eig} = \quad round\left(\frac{L}{dz}\right),
$$

where:

- L is the specified value, in the **Shaft** settings, for the **Shaft length** parameter.

- dz is the specified value, in the **Advanced Bending** settings, for the **Shaft length increments for mode shape computations** parameter.

To compute the m undamped eigenmodes and eigenfrequencies, the block uses the `eigs` function. The equation takes the form :

```
[H, λ] = eigs( sparse(K), sparse(M), mMax, 'smallestabs' ),
```

where:

- **H** is the $4(N + 1) \times M$ eigenvector matrix. Each column is an eigenmode in the $\vec{x}$ coordinates.
- λ are the eigenvalues, which are the square of the eigenfrequencies.
- m_{Max} is the specified value, in the **Advanced Bending** settings, for the **Limit number of modes** parameter.

The number of eigenmodes computed, m, is less than m_{Max} if:

- There are modes with eigenfrequencies that exceed the specified value, in the **Advanced Bending** settings, for the **Eigenfrequency upper limit** parameter. The block discards these modes.
- The eigenvalues fail to converge. For more information, see `eigs`.

If the **Bending mode determination** parameter is set to `User defined`, the block computes the eigenvector matrix **H** from the specified values, in the **Advanced Bending** settings, for these parameters:

- **X-direction mode shapes**
- **Y-direction mode shapes**
- **Shaft position**

To determines the node axial locations for $\vec{x}$, the block uses the elements specified for the **Shaft position** parameter as the primary nodes.

To compute the modal rotation, θ and φ, for each node, the block uses the `gradient` function. The equations take the form:

```
θ = -gradient(Y direction mode shapes)
φ = gradient(X direction mode shapes)
```

The block assembles the **X-direction mode shapes**, **Y-direction mode shapes**, and modal rotations, θ and φ, into $\vec{x}$ coordinates for each column of **H**.

2   The block computes the modal matrices, M_{Modal}, K_{Modal}, B_{Modal}, G_{Modal}, and f_{Modal}, as:

$$M_{Modal} = H^T M H$$

$$K_{Modal} = H^T K H$$

$$B_{Modal} = H^T B H$$

$$G_{Modal} = H^T G_{Disk} H$$

$$\vec{f}_{Modal} = H^T \vec{f}$$

Although the block computes undamped eigenmodes, H, in step 1, the modal damping matrix, B_{Modal}, and modal gyroscopics matrix, G_{Modal}, may model light damping. The block normalizes the matrices so that M_{Modal} is the identity matrix.

During simulation, the block simulates the eigenmode equation of motion:

$$M_{Modal}\ddot{\vec{\eta}} + (B_{Modal} + G_{Modal}\Omega)\dot{\vec{\eta}} + \left(K_{Modal} + G_{Modal}\dot{\Omega}\right)\vec{\eta} \ = \ \vec{f}_{Modal},$$

where the modal degrees of freedom, $\vec{\eta}$, relate to the node degrees of freedom by:

$$\vec{x} = H\vec{\eta}$$

Speed-Dependent Eigenmodes Method

The support stiffness and support damping vary if, in the **Supports** settings, the mounting type parameter for any of the supports is set to Speed-dependent bearing matrix. The speed-dependent eigenmodes model accounts for these effects by varying the modal properties, H, B_{Modal}, G_{Modal}, K_{Modal}, and f_{Modal} as the shaft speed changes. M_{Modal} is normalized to the identity matrix for all shaft speeds, so it does not depend on shaft speed.

If the shaft has speed-dependent bearing supports, then the block repeats the bending mode eigenmodes method steps for each element in the shaft speed vector. The shaft vector elements are the specified values, in the **Supports** settings, for the **Bearing speed [s1,...,sS]** parameter. During simulation, the modal stiffness, damping, and forcing magnitude are adjusted based on lookup tables of the properties versus the shaft speed.

That is, the block simulates the eigenmode equation of motion as:

$$M_{Modal}\ddot{\vec{\eta}} + \ (B_{Modal}(\Omega) + G_{Modal}(\Omega)\Omega)\dot{\vec{\eta}} + \ \left(K_{Modal}(\Omega) + G_{Modal}(\Omega)\dot{\Omega}\right)\vec{\eta} \ = \ \vec{f}_{Modal}(\Omega),$$

where K_{Modal}, B_{Modal}, and f_{Modal} have the form:

$$K_{Modal}(\Omega) = lookup(\ |\Omega_{Ref}|, \quad K_{Modal,\,Ref}, \quad \Omega, \quad interpolation = linear, \quad extrapolation = nearest) \ ,$$

$$B_{Modal}(\Omega) = lookup(\ |\Omega_{Ref}|, \quad B_{Modal,\,Ref}, \quad \Omega, \quad interpolation = linear, \quad extrapolation = nearest) \ ,$$

$$G_{Modal}(\Omega) = lookup(\ |\Omega_{Ref}|, \quad G_{Modal,\,Ref}, \quad \Omega, \quad interpolation = linear, \quad extrapolation = nearest) \ ,$$

$$\vec{f}_{Modal}(\Omega) = lookup(\ |\Omega_{Ref}|, \quad \vec{f}_{Modal,\,Ref}, \quad \Omega, \quad interpolation = linear, \quad extrapolation = nearest) \ ,$$

where:

- Ω_{Ref} is the specified value, in the **Supports** settings, for the **Bearing speed [s1,...,sS]** parameter.
- $K_{Modal,Ref}$ is the table of modal stiffnesses at each Ω_{Ref}.
- $B_{Modal,Ref}$ is the table of support damping at each Ω_{Ref}.
- $G_{Modal,Ref}$ is the table of disk gyroscopic damping at each Ω_{Ref}.

- $f_{Modal,Ref}$ is the table of modal forcing at each Ω_{Ref}.

The block correlates the mode shape similarity at different values of Ω_{Ref} and reorders modes, if necessary, so that each modal degree of freedom, $\vec{\eta}$, has properties that gradually change with the shaft speed.

Improve Simulation Speed or Accuracy

The balance between simulation accuracy and performance depends on N, the number of flexible elements that the block uses to represent the shaft. Simulation accuracy is a measure of how much the simulation results agree with mathematical and empirical models. Generally, as N increases, so does model fidelity and simulation accuracy. However, the computational cost of the simulation is also correlated to N, and as computational cost increases, performance decreases. Conversely, as N decreases, simulation speed increases but simulation accuracy decreases.

To increase simulation accuracy for the lumped mass method for either a torsion or bending model, increase the minimum number of flexible elements, N_{min}. The single-flexible-element torsion model exhibits a torsional eigenfrequency that is close to the first eigenfrequency of the continuous, distributed parameter model. For greater accuracy you can select 2, 4, 8, or more flexible elements. For example, the four lowest torsional eigenfrequencies are represented with an accuracy of 0.1, 1.9, 1.6, and 5.3 percent, respectively, by a 16-flexible-element model.

To increase simulation accuracy for the eigenmodes method to a bending model:

- If simulating with static eigenmode dependency on rotation speed, verify that the **Nominal shaft speed for bending modes** parameter is close to the simulation shaft speed. This parameter may affect model results if you parameterize a rigid disk attached to the shaft with a large mass moment of inertia about the shaft axis or specify any speed-dependent bearing matrix supports.

- If simulating with dynamic eigenmode dependency on rotation speed, verify that, in the **Supports** settings, the specified values for the **Bearing speed [s1,...,sS]** span the shaft speed range of the simulation or that saturation of the support stiffness and damping at shaft speeds outside the range is an acceptable approximation.

- In the **Advanced Bending** settings, decrease the value of the **Shaft length increments for mode shape computations** parameter. Reducing the value can increase the accuracy of modal frequencies and shapes.

- Decrease the support damping and disk polar moment of inertia about the shaft axis. Simscape computations of the mode shapes and frequencies before simulation do not account for this damping.

- Check the sensitivity to the **Advanced Bending** settings by using your parameters in the flexible shaft model in the "Shaft with Torsional and Transverse Flexibility" example. Adjust the parameters and use the links provided in the example to examine how the values affect the eigenmode frequencies and shapes. Adjust the parameter values in your model accordingly.

- Increase the values of the **Eigenfrequency upper limit** and **Limit number of modes** parameters. The highest modal frequency in the simulation must be significantly larger than the shaft rotation frequency.

Assumptions and Limitations

- The distributed parameter model of a continuous torsional shaft is approximated by a finite number, N, of lumped masses.

- Shaft rotation and torsion flexibility excite shaft bending, but bending does not affect shaft rotation and torsion flexibility.

- Rigid point masses or disks attached to the shaft have thin lengths parallel to the shaft axis.
- For the eigenmodes bending model, damping does not affect the eigenfrequencies.
- Shaft bending is not transmitted between Flexible Shaft blocks.
- Relative to the shaft length, the shaft outer diameter is small.
- Relative to the shaft length, the bending deflection is small.
- Static mass unbalances are the only shaft-bending external exciting loads.
- Shaft supports are stationary.
- Gyroscopic effects of the rigid disks are considered; gyroscopic effects of the shaft itself are neglected.
- Static mass unbalance forcing in the eigenmodes method uses the rotation speed at the shaft midpoint.
- If the shaft models torsion only and uses the parameterization options **By stiffness and inertia** or **By segment stiffness and inertia**, the block uses only two supports, one each at the **B** and **F** ends.

Ports

Output

Fr — Force on bearing supports
physical signal

Physical signal outport associated with the force that the shaft exerts on the bearing supports.

Dependencies

This port is visible if, in the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .

M — Moment on bearing supports
physical signal

Physical signal outport associated with the moment that the shaft exerts on the bearing supports.

Dependencies

This port is visible if, in the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .

V — Shaft translational velocity
physical signal

Physical signal outport associated with the translational velocity of the shaft at the bearing supports.

Dependencies

This port is visible if, in the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .

W — Shaft angular velocity
physical signal

Physical signal outport associated with the angular velocity of the shaft at the bearing supports.

Dependencies

This port is visible if, in the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .

Conserving

B — Base
rotational mechanical

Rotational conserving port associated with the shaft base.

F — Follower
rotational mechanical

Rotational conserving port associated with the shaft follower.

Parameters

Shaft

For certain parameters in the **Shaft** settings, the option that you choose affects the visibility of:

- Other parameters in the **Shaft** settings.
- Parameters in the **Torsion** settings.
- **Supports** settings and parameters.
- **Bending** settings
- **Advanced Bending** settings
- These output ports:

  - **Fr**
  - **M**
  - **V**
  - **W**

The table shows how the options that you choose for the **Shaft** settings affect the visibility of other parameters in the **Shaft** settings. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Shaft Parameter Dependencies

Shaft Setting Parameters and Values							
Model bending							
`Off`				`On`			
Minimum number of flexible elements				**Minimum number of flexible elements**			
Parameterization				**Parameterization**			
By stiffness and inertia	By material and geometry	By segment stiffness and inertia	By material and segment geometry	By stiffness and inertia	By material and geometry	By segment stiffness and inertia	By material and segment geometry
	Shaft Length		**Segment lengths [B,...,F]**	**Shaft Length**	**Shaft Length**	**Segment lengths [B,...,F]**	**Segment lengths [B,...,F]**
Torsional stiffness		**Segment torsional stiffness [B,...,F]**		**Torsional stiffness**		**Segment torsional stiffness [B,...,F]**	
Torsional inertia		**Segment torsional inertia [B,...,F]**		**Torsional inertia**		**Segment torsional inertia [B,...,F]**	
				Bending rigidity		**Segment bending rigidity [B,...,F]**	
	Material density		**Material density**	**Linear density**	**Material density**	**Segment linear density [B,...,F]**	**Material density**
	Shear modulus		**Shear modulus**		**Shear modulus**		**Shear modulus**
					Young's modulus		**Young's modulus**
	Shaft geometry		**Shaft geometry**		**Shaft geometry**		**Shaft geometry**
	Solid / Annular		Solid / Annular		Solid / Annular		Solid / Annular

Shaft Setting Parameters and Values											
	Shaft outer diameter	Shaft outer diameter		Segment outer diameter [B,...,F]	Segment outer diameter [B,...,F]		Shaft outer diameter	Shaft outer diameter		Segment outer diameter [B,...,F]	Segment outer diameter [B,...,F]
		Shaft inner diameter			Segment inner diameter [B,...,F]			Shaft inner diameter			Segment inner diameter [B,...,F]

Model bending — Bending model option

Off (default) | On

Option to model shaft bending.

Dependencies

These parameters, settings, and ports are affected by the **Model bending** parameter.

- For the **Shaft** settings, the visibility of dependent parameters is tabulated in Shaft Parameter Dependencies.
- In the **Torsion** settings:

  - The **Viscous friction coefficients at base (B) and follower (F)** parameter is visible if, in the **Shaft** settings, **Model bending** is set to Off and **Parameterization** is set to By stiffness and inertia or By segment stiffness and inertia.

  - The **Viscous friction coefficients at each support [B1,...,F1]** parameter is visible if, in the **Shaft** settings, one of these conditions is met:

    - **Model bending** is set to On.

    - **Model bending** is set to Off and **Parameterization** is set to By material and geometry or By material and segment geometry.

- These settings are exposed only if **Model bending** is set to On:

  - **Supports**
  - **Bending**
  - **Advanced bending**

- These ports are exposed only if **Model bending** is set to On:

- **Fr**
- **M**
- **V**
- **W**

Minimum number of flexible elements — Minimum number of flexible elements
1 (default) | positive integer

Minimum number of flexible elements, N_{min}, for the approximation.

It is possible that the flexible elements have different lengths or that the simulated number of flexible elements, N, is larger than N_{min}. For more information, see "Node Placement Algorithm" on page 1-178.

A larger number of flexible elements, N, increases the fidelity of the model, but reduces simulation performance. The single flexible element model (N=1) exhibits a torsion eigenfrequency that is close to the first eigenfrequency of the continuous, distributed parameter model.

If model fidelity is more important than performance, select 2, 4, 8, or more flexible elements. For example, the four lowest torsion eigenfrequencies are represented with an accuracy of 0.1, 1.9, 1.6, and 5.3 percent, respectively, by a model with 16 flexible elements. Generally, more flexible elements are required for accurately modeling bending dynamics than are required for accurately modeling torsion dynamics.

For more information, see "Improve Simulation Speed or Accuracy" on page 1-193.

Parameterization — Parameterization method
By stiffness and inertia (default) | By material and geometry | By segment stiffness and inertia | By material and segment geometry

Parameterization method. You can model a homogeneous shaft or one that is axially inhomogeneous for any of these attributes:

- Torsional stiffness
- Torsional inertia
- Bending rigidity
- Density
- Shear modulus
- Young's modulus
- Outer diameter
- Inner diameter

The parameterization options for a homogeneous shaft model are:

- By stiffness and inertia — Specify the torsional stiffness and inertia, and the density per unit length of the shaft. For the bending model, also specify the bending rigidity and length of the shaft.
- By material and geometry — Specify the length and axial cross-sectional geometry, in terms of inner and outer diameters, of the shaft. For the shaft material, specify the density and shear modulus. For the bending model, also specify Young's modulus for the material of the shaft.

The parameterization options for an axially inhomogeneous shaft model are:

- `By segment stiffness and inertia` — For each segment of the shaft, specify the torsional stiffness, torsional inertia, and density per unit length. For the bending model, also specify the bending rigidity and length for each segment.
- `By material and segment geometry` — For each segment of the shaft, specify the length and axial cross-sectional geometry, in terms of inner and outer diameters. For the material of the shaft, specify the density and shear modulus. For the bending model, also specify Young's modulus for the material of the shaft.

Dependencies

Each **Parameterization** option affects the visibility of:

- Dependent parameters in the **Shaft** settings. For more information, see Shaft Parameter Dependencies.
- Dependent parameters in the **Torsion** settings:

  - The **Viscous friction coefficients at base (B) and follower (F)** parameter is visible if, in the **Shaft** settings, **Model bending** is set to `Off` and **Parameterization** is set to `By stiffness and inertia` or `By segment stiffness and inertia`.
  - The **Viscous friction coefficients at each support [B1,...,F1]** parameter is visible if, in the **Shaft** settings, one of these conditions is met:

    - **Model bending** is set to `On`.
    - **Model bending** is set to `Off` and **Parameterization** is set to `By material and geometry` or `By material and segment geometry`.

- The **Supports** settings. When the **Model bending** parameter is set to `Off`, the **Supports** settings are visible when the **Parameterization** parameter is set to `By material and geometry` or `By material and segment geometry`.

Shaft length — Shaft length
1 m (default) | positive scalar

Length of the shaft.

Dependencies

This parameter is visible when one of these conditions are met:

- **Model bending** is set to `Off` and **Parameterization** is set to `By material and geometry`.
- **Model bending** is set to `On` and **Parameterization** is set to `By material and geometry` or `By stiffness and inertia`.

For more information on how other parameters affect the visibility of this parameter, see Shaft Parameter Dependencies.

Segment lengths [B,...,F] — Length of each shaft segment
[1, .5, .25] (default) | vector

Length of each shaft segment that the shaft is divided into lengthwise for modeling an axially inhomogeneous shaft. The number of elements in the vector is equal to the number of segments that you use to model the inhomogeneous shaft. The order of the elements in the vector corresponds to the segment order relative to **B**, the base end of the shaft.

Dependencies

This parameter is visible when one of these conditions are met:

- **Parameterization** is set to By material and segment geometry.
- **Model bending** is set to On and **Parameterization** is set to By segment stiffness and inertia.

For more information on how other parameters affect the visibility of this parameter, see Shaft Parameter Dependencies.

Torsional stiffness — Material stiffness
2e5 N*m/rad (default) | positive scalar

Torque per radian twist of the shaft.

Dependencies

This parameter is visible when **Parameterization** is set to By stiffness and inertia.

For more information on how other parameters affect the visibility of this parameter, see Shaft Parameter Dependencies.

Segment torsional stiffness [B,...,F] — Material stiffness for each shaft segment
[400000, 200000, 100000] N*m/rad (default) | positive vector

Torque per radian twist for each segment of the shaft. The number of elements in the vector must be the same as the number of elements specified for the **Segment lengths [B,...,F]** on page 1-0 parameter. The order of the elements in the vector corresponds to the segment order relative to **B**, the base end of the shaft.

Dependencies

This parameter is visible when **Parameterization** is set to By segment stiffness and inertia.

For more information on how other parameters affect the visibility of this parameter, see Shaft Parameter Dependencies.

Torsional Inertia — Polar moment of inertia
0.02 kg*m^2 (default) | positive scalar

Ability of the shaft to resist torsional acceleration.

Dependencies

This parameter is visible when **Parameterization** is set to By stiffness and inertia.

For more information on how other parameters affect the visibility of this parameter, see Shaft Parameter Dependencies.

Segment torsional inertia [B,...,F] — Polar moment of inertia for each shaft segment
[.025, .02, .015] kg*m^2 (default) | positive vector

Ability of the each shaft segment to resist torsional acceleration. The number of elements in the vector must be the same as the number of elements specified for the **Segment lengths [B,...,F]** on

page 1-0   parameter. The order of the elements in the vector corresponds to the segment order relative to, **B**, the base end of the shaft.

Dependencies

This parameter is visible when **Parameterization** is set to By segment stiffness and inertia.

For more information on how other parameters affect the visibility of this parameter, see Shaft Parameter Dependencies.

Bending rigidity — Bending rigidity
5e5 m^4*Pa (default) | positive scalar

Bending rigidity for the shaft material.

Dependencies

This parameter is visible when **Model bending** is set to On and **Parameterization** is set to By stiffness and inertia.

For more information on how other parameters affect the visibility of this parameter, see Shaft Parameter Dependencies.

Segment bending rigidity [B,...,F] — Bending rigidity of each segment
[600000, 500000, 400000] m^4*Pa (default) | positive vector

Bending rigidity for the material of each sequential segment of the shaft. The number of elements in the vector must be the same as the number of elements specified for the **Segment lengths [B,...,F]** on page 1-0   parameter. The order of the elements in the vector corresponds to the segment order relative to **B**, the base end of the shaft.

Dependencies

This parameter is visible when **Model bending** is set to On and **Parameterization** is set to By segment stiffness and inertia.

For more information on how other parameters affect the visibility of this parameter, see Shaft Parameter Dependencies.

Linear density — Density per unit length
19 kg/m (default) | positive scalar

Density of the shaft material per unit length of the shaft.

Dependencies

This parameter is visible when **Parameterization** is set to By stiffness and inertia.

For more information on how other parameters affect the visibility of this parameter, see Shaft Parameter Dependencies.

Segment linear density [B,...,F] — Density per unit length for each segment
[20, 19, 18] kg/m (default) | positive vector

Density of the shaft material per unit length of each segment of the shaft. The number of elements in the vector must be the same as the number of elements specified for the **Segment lengths [B,...,F]**

on page 1-0   parameter. The order of the elements in the vector corresponds to the segment order relative to **B**, the base end of the shaft.

Dependencies

This parameter is visible when **Parameterization** is set to By segment stiffness and inertia.

For more information on how other parameters affect the visibility of this parameter, see Shaft Parameter Dependencies.

Material density — Material density
7.8e3 kg/m^3 (default) | positive scalar

Density of the shaft material.

Dependencies

This parameter is visible when **Parameterization** is set to By material and geometry or By material and segment geometry.

For more information on how other parameters affect the visibility of this parameter, see Shaft Parameter Dependencies.

Shear modulus — Shear modulus
7.93e9 Pa (default) | positive scalar

Shear modulus for the shaft material.

Dependencies

This parameter is visible when **Parameterization** is set to By material and geometry or By material and segment geometry.

For more information on how other parameters affect the visibility of this parameter, see Shaft Parameter Dependencies.

Young's modulus — Young's modulus
200e9 Pa (default) | positive scalar

Young's modulus for the material.

Dependencies

This parameter is visible when **Model bending** is set to On and **Parameterization** is set to By material and geometry or By material and segment geometry.

For more information on how other parameters affect the visibility of this parameter, see Shaft Parameter Dependencies.

Shaft geometry — Cross-sectional geometry
Solid (default) | Annular

Cross-sectional geometry along the length of the shaft. If the shaft or segments of the shaft are hollow, select Annular. Otherwise, select Solid.

Dependencies

This parameter is visible when **Parameterization** is set to By material and geometry or By material and segment geometry.

For more information on how other parameters affect the visibility of this parameter, see Shaft Parameter Dependencies.

Shaft outer diameter — Outer diameter of the shaft
0.075 m (default) | positive scalar

Outer diameter of the shaft.

Dependencies

This parameter is visible when **Parameterization** is set to By material and geometry.

For more information on how other parameters affect the visibility of this parameter, see Shaft Parameter Dependencies.

Segment outer diameter [B,...,F] — Outer diameter for each shaft segment
[.085, .075, .065] m (default) | positive vector

Outer diameter of each shaft segment. The number of elements in the vector must be the same as the number of elements specified for the **Segment lengths [B,...,F]** on page 1-0   parameter. The order of the elements in the vector corresponds to the segment order relative to **B**, the base end of the shaft.

Dependencies

This parameter is visible when **Parameterization** is set to By material and segment geometry.

For more information on how other parameters affect the visibility of this parameter, see Shaft Parameter Dependencies.

Shaft inner diameter — Inner diameter of the shaft
0.05 m (default) | positve scalar

Inner diameter of the annular shaft. The value must be smaller than the value specified for the **Shaft outer diameter** parameter.

Dependencies

This parameter is visible when **Parameterization** is set to By material and geometry and **Shaft geometry** is set to Annular.

For more information on how other parameters affect the visibility of this parameter, see Shaft Parameter Dependencies.

Segment inner diameter [B,...,F] — Inner diameter for each shaft segment
[.055, .05, .045] m (default) | vector

Inner diameters of the shaft segments. The number of elements in the vector must be the same as the number of elements specified for the **Segment lengths [B,...,F]** on page 1-0   parameter. The order of the elements in the vector corresponds to the segment order relative to **B**, the base end of the shaft. Each value must be smaller than the corresponding value specified for the **Segment outer**

diameter [B,...,F] parameter. If a shaft segment is solid, specify `0` for the corresponding vector element. At least one element in the vector must be positive.

Dependencies

This parameter is visible when **Parameterization** is set to `By material and segment geometry` and **Shaft geometry** is set to `Annular`.

For more information on how other parameters affect the visibility of this parameter, see Shaft Parameter Dependencies.

Torsion

Damping ratio from internal losses — Material damping ratio
`0.01` (default) | positive scalar

Material damping ratio.

Viscous friction coefficients at base (B) and follower (F) — Viscous friction coefficients
`[0, 0]` N*m/(rad/s) (default) | nonnegative vector

Viscous friction coefficients at the base, **B**, and follower, **F**, ends of the shaft. The vector must contain two elements.

Dependencies

This parameter is visible if, in the **Shaft** settings, **Model bending** is set to `Off` and **Parameterization** is set to `By stiffness and inertia` or `By segment stiffness and inertia`.

Viscous friction coefficients at each support [B1,...,F1] — Viscous friction coefficients
`[0, 0]` N*m/(rad/s) (default) | nonnegative vector

Viscous friction coefficients at each support. The number of elements in the vector must be the same as the number specified in the **Supports** settings for the **Number of supports** on page 1-0 parameter. The order of the elements must correspond to the sequential position of each support **B**, the base end of the shaft.

Dependencies

This parameter is visible if, in the **Shaft** settings, one of these conditions is met:

• **Model bending** is set to `On`.
• **Model bending** is set to `Off` and **Parameterization** is set to `By material and geometry` or `By material and segment geometry`.

Initial shaft torsional deflection — Initial shaft torsional deflection
`0` rad (default) | nonnegative scalar

Angular deflection of the shaft at the start of simulation.

A positive initial deflection results in a positive rotation of **B**, the base end of the shaft, relative to **F**, the follower end of the shaft.

Initial shaft angular velocity — Initial angular velocity
`0 rpm` (default) | nonnegative scalar

Angular velocity of the shaft at the start of simulation.

Supports

The **Supports** settings are visible if, in the **Shaft** settings, one of these conditions is met:

- **Model bending** is set to `Off` and **Parameterization** is set to `By material and geometry` or `By material and segment geometry`.

  For this condition, only these parameters in the **Supports** settings are visible:

  - **Number of supports**
  - **Support locations relative to base (B)**
- **Model bending** is set to `On`.

  The visibility of the **Supports** settings depends on the values of several parameters in the **Supports** settings.

In the **Supports** settings, if the **Number of supports** parameter is set to 2, only mounting types and the dependent parameters for the *B1* and *F1* supports are visible. The *B1* support is the support that is closest to **B**, the base end of the shaft. The *F1* support is the support that is closest to **F**, the follower end of the shaft. If **Number of supports** is set to a value greater than 2, the mounting type and dependent parameters are exposed for the intermediate supports, I_N, where *N* is the number of intermediate supports. For example, for 3 supports, the mounting type and dependent parameters for the *B1*, *I1*, and *F1* support are exposed.

These tables show the parameter dependencies for the `Lumped Mass` and `Eigenmodes` bending vibration analysis methods for each mounting type. The parameters are visible for the *B1* and *F1* supports and for any intermediate supports, *I1* and *I2*, that you specify. The names of the dependent parameters are prefixed by the name of the corresponding support. For example, for the *B1* support, the dependent parameter **Rotational damping [xx,yy]** is named **Base (B1) rotational damping [xx,yy]**. The parameter prefixes are:

- **Base (B1)**
- **Intermediate (I1)**
- **Intermediate (I2)**
- **Follower (F1)**

- **Mounting Type Parameter Dependencies**

Supports Mounting Types and Dependent Parameters				
Clamped	Pinned	Free	Bearing matrix	Speed-dependent bearing matrix
-	Rotational damping [xx,yy]	Rotational damping [xx,yy]	Rotational damping [xx,yy]	Rotational damping [xx,yy]
			Translational damping [xx,xy,yx,yy]	Speed-dependent translational damping [xx1,xy1, yx1,yy1;...xxS,xyS,yxS,yyS]
			Rotational stiffness [xx,yy]	Rotational stiffness [xx,yy]
			Translational stiffness [xx,xy,yx,yy]	Speed-dependent translational stiffness [xx1,xy1,yx1, yy1;...xxS,xyS,yxS,yyS]
				Bearing speed [s1,...,sS]

Number of supports — Number of supports
2 (default) | 3 | 4

Number of shaft supports.

Dependencies

This parameter is visible if, in the **Shaft** settings, one of these conditions is met:

- **Model bending** is set to Off and **Parameterization** is set to By material and geometry or By material and segment geometry

- **Model bending** is set to On.

If this parameter is set to:

- 2 — Parameters for the *B1* and *F1* supports are exposed. The *B1* support is the support that is closest to **B** the base end of the shaft. The *F1* support is the support that is closest to **F**, the follower end of the shaft.

- 3 — Parameters for the *B1*, *I1*, and *F1* supports are exposed.

- 4 — Parameters for the *B1*, *I1*, *I2*, and *F1* supports are exposed.

Support locations relative to base (B) — Support locations
[0, 1] m (default) | nonnegative increasing vector

Support locations relative to **B**, the base end of the shaft. The number of elements must be the same as the number specified for the **Number of supports** parameter. The order of the elements corresponds to the sequential position of each support relative to the base end of the shaft. The largest value must be no larger than the length of the shaft. For a segmented shaft model, the shaft length is equal to the sum of the individual segment lengths.

Dependencies

This parameter is visible if, in the **Shaft** settings, one of these conditions is met:

- **Model bending** is set to `Off` and **Parameterization** is set to `By material and geometry` or `By material and segment geometry`
- **Model bending** is set to `On`.

Base (B1) mounting type — Support *B1* mounting type
`Clamped` (default) | `Pinned` | `Free` | `Bearing matrix` | `Speed-dependent bearing matrix`

Type of mounting at the base end of the shaft.

Dependencies

This parameter is visible when, in the **Shaft** settings the **Model bending** parameter is set to `On`. For more information, see Shaft Parameter Dependencies.

Setting this parameter to `Pinned`, `Free`, `Bearing matrix`, or `Speed-dependent bearing matrix` exposes related parameters. For more information, see Mounting Type Parameter Dependencies.

Base (B1) rotational damping [xx,yy] — Support *B1* rotational damping
`[1e3, 1e3]` N*m/(rad/s) (default) | vector

Rotational damping for the *B1* support. *B1* is the support that is closest to **B**, the base end of the shaft. The elements of the two-element vector are:

- *xx* — Damping about the *x*-axis
- *yy* — Damping about the *y*-axis

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings, **Base (B1) mounting type** is set to `Pinned`, `Free`, `Bearing matrix`, or `Speed-dependent bearing matrix`. For more information, see Mounting Type Parameter Dependencies.

Base (B1) translational damping [xx,xy,yx,yy] — Support *B1* translational damping
`[2e4, -4.4e4, -4.4e4, 3.5e5]` N/(m/s) (default) | vector

Translational damping for the *B1* support. The elements of the four-element vector are:

- *xx* — Damping in the *x*-axis direction
- *xy* — Damping in the *x*-axis direction coupled with motion in the *y*-axis direction

- *yx* — Damping in the *y*-axis direction coupled with motion in the *x*-axis direction
- *yy* — Damping in the *y*-axis direction

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings, **Base (B1) mounting type** is set to `Bearing matrix`. For more information, see Mounting Type Parameter Dependencies.

Base (B1) speed-dependent translational damping
[xx1,xy1,yx1,yy1;...xxS,xyS,yxS,yyS] — Support *B1* speed-dependent translational
damping
[2e4, -4.4e4, -4.4e4, 3.5e5; 1e4, -2.5e4, -2.5e4, 1.4e5; 5e3, -1.0e4, -1.0e4, 2.0e4] N/(m/s) (default) | matrix

Speed-dependent translational damping for the *B1* support. The number of rows in the matrix must equal the number of elements in the vector specified for the **Bearing speed [s1, … sS]** parameter. Each row contains four elements:

- xx_s — Damping in the *x*-axis direction at the sth speed
- xy_s — Damping in the *x*-axis direction coupled with motion in the *y*-axis direction at the sth speed
- yx_s — Damping in the *y*-axis direction coupled with motion in the *x*-axis direction at the sth speed
- yy_s — Damping in the *y*-axis direction at the sth speed

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings, **Base (B1) mounting type** is set to `Speed-dependent bearing matrix`. For more information, see Mounting Type Parameter Dependencies.

Base (B1) rotational stiffness [xx,yy] — Support *B1* rotational stiffness
[1e4, 1e4] N*m/rad (default) | nonnegative vector

Rotational stiffness for the *B1* support. The elements of the two-element vector are:

- *xx* — Stiffness about the *x*-axis
- *yy* — Stiffness about the *y*-axis

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings, **Base (B1) mounting type** is set to `Bearing matrix` or `Speed-dependent bearing matrix`. For more information, see Mounting Type Parameter Dependencies.

Base (B1) translational stiffness [xx,xy,yx,yy] — Support *B1* translational stiffness
[1e6, -2e5, -3e5, 5e] N/m (default) | vector

Translational stiffness for the *B1* support. The elements of the four-element vector are:

- *xx* — Stiffness in the *x*-axis direction
- *xy* — Stiffness in the *x*-axis direction coupled with motion in the *y*-axis direction
- *yx* — Stiffness in the *y*-axis direction coupled with motion in the *x*-axis direction
- *yy* — Stiffness in the *y*-axis direction

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to On. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings, **Base (B1) mounting type** is set to Bearing matrix. For more information, see Mounting Type Parameter Dependencies.

Base (B1) speed-dependent translational stiffness [xx1,xy1,yx1,yy1;...xxS,xyS,yxS,yyS] — Support *B1* speed-dependent translational stiffness
[1e6, -6e5, -3.5e6, 8e6; 1e6, -2e5, -3.0e6, 5e6; 1e6, -5e4, -2.5e6, 3e6] N/m (default) | matrix

Speed-dependent translational stiffness for the *B1* support. The number of rows in the matrix must equal the number of elements in the vector specified for the **Bearing speed [s1, ... sS]** parameter. Each row contains four elements:

- xx_s — Stiffness in the *x*-axis direction at the sth speed
- xy_s — Stiffness in the *x*-axis direction coupled with motion in the *y*-axis direction at the sth speed
- yx_s — Stiffness in the *y*-axis direction coupled with motion in the *x*-axis direction at the sth speed
- yy_s — Stiffness in the *y*-axis direction at the sth speed

All *xx* and *yy* stiffness values must be positive. All *xy* and *yx* values must be zero or nonzero at all speeds.

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to On. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings, **Base (B1) mounting type** is set to Speed-dependent bearing matrix. For more information, see Mounting Type Parameter Dependencies.

Intermediate (I1) mounting type — Support *I1* mounting type
Clamped (default) | Pinned | Free | Bearing matrix | Speed-dependent bearing matrix

Type of mounting at the *I1* support. The *I1* support is the closest intermediate support to the *B1* support.

Dependencies

This parameter is visible when both of these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Supports** settings, **Number of supports** is set to 3 or 4.

Setting this parameter to `Pinned`, `Free`, `Bearing matrix`, or `Speed-dependent bearing matrix` exposes related parameters. For more information, see Mounting Type Parameter Dependencies.

Intermediate (I1) rotational damping [xx,yy] — Support *I1* rotational damping
`[1e3, 1e3]` N*m/(rad/s) (default) | vector

Rotational damping for the *I1* support. The elements of the two-element vector are:

- *xx* — Damping about the *x*-axis
- *yy* — Damping about the *y*-axis

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings:

  - **Number of supports** is set to 3 or 4.
  - **Intermediate (I1) mounting type** is set to `Pinned`, `Free`, `Bearing matrix`, or `Speed-dependent bearing matrix`. For more information, see Mounting Type Parameter Dependencies.

Intermediate (I1) translational damping [xx,xy,yx,yy] — Support *I1* translational damping
`[2e4, -4.4e4, -4.4e4, 3.5e5]` N/(m/s) (default) | vector

Translational damping for the *I1* support. The elements of the four-element vector are:

- *xx* — Damping in the *x*-axis direction
- *xy* — Damping in the *x*-axis direction coupled with motion in the *y*-axis direction
- *yx* — Damping in the *y*-axis direction coupled with motion in the *x*-axis direction
- *yy* — Damping in the *y*-axis direction

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings:

  - **Number of supports** is set to 3 or 4.

- **Intermediate (I1) mounting type** is set to `Bearing matrix`. For more information, see Mounting Type Parameter Dependencies.

Intermediate (I1) speed-dependent translational damping [xx1,xy1,yx1,yy1;...xxsS,xyS,yxS,yyS] — Support *I1* speed-dependent translational damping
[2e4, -4.4e4, -4.4e4, 3.5e5; 1e4, -2.5e4, -2.5e4, 1.4e5; 5e3, -1.0e4, -1.0e4, 2.0e4] N/(m/s) (default) | matrix

Speed-dependent translational damping for the *I1* support. The number of rows in the matrix must equal the number of elements in the vector specified for the **Bearing speed [s1, … sS]** parameter. Each row contains four elements:

- xx_s — Damping in the *x*-axis direction at the sth speed
- xy_s — Damping in the *x*-axis direction coupled with motion in the *y*-axis direction at the sth speed
- yx_s — Damping in the *y*-axis direction coupled with motion in the *x*-axis direction at the sth speed
- yy_s — Damping in the *y*-axis direction at the sth speed

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings:

  - **Number of supports** is set to 3 or 4.
  - **Intermediate (I1) mounting type** is set to `Speed-dependent bearing matrix`. For more information, see Mounting Type Parameter Dependencies.

Intermediate (I1) rotational stiffness [xx,yy] — Support *I1* rotational stiffness
[1e4, 1e4] N*m/rad (default) | nonnegative vector

Rotational stiffness for the *I1* support. The elements of the two-element vector are:

- xx — Stiffness about the *x*-axis
- yy — Stiffness about the *y*-axis

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings:

  - **Number of supports** is set to 3 or 4.
  - **Intermediate (I1) mounting type** is set to `Bearing matrix` or `Speed-dependent bearing matrix`. For more information, see Mounting Type Parameter Dependencies.

Intermediate (I1) translational stiffness [xx,xy,yx,yy] — Support *I1* translational stiffness
[1e6, -2e5, -3e5, 5e] N/m (default) | vector

Translational stiffness for the *I1* support. The elements of the four-element vector are:

- *xx* — Stiffness in the *x*-axis direction
- *xy* — Stiffness in the *x*-axis direction coupled with motion in the *y*-axis direction
- *yx* — Stiffness in the *y*-axis direction coupled with motion in the *x*-axis direction
- *yy* — Stiffness in the *y*-axis direction

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings:

  - **Number of supports** is set to 3 or 4.
  - **Intermediate (I1) mounting type** is set to `Bearing matrix`. For more information, see Mounting Type Parameter Dependencies.

Intermediate (I1) speed-dependent translational stiffness [xx1,xy1,yx1,yy1;...xxS,xyS,yxS,yyS] — Support *I1* speed-dependent translational stiffness
[1e6, -6e5, -3.5e6, 8e6; 1e6, -2e5, -3.0e6, 5e6; 1e6, -5e4, -2.5e6, 3e6] N/m (default) | matrix

Speed-dependent translational stiffness for the *I1* support. The number of rows in the matrix must equal the number of elements in the vector specified for the **Bearing speed [s1, … sS]** parameter. Each row contains four elements:

- xx_s — Stiffness in the *x*-axis direction at the sth speed
- xy_s — Stiffness in the *x*-axis direction coupled with motion in the *y*-axis direction at the sth speed
- yx_s — Stiffness in the *y*-axis direction coupled with motion in the *x*-axis direction at the sth speed
- yy_s — Stiffness in the *y*-axis direction at the sth speed

All *xx* and *yy* stiffness values must be positive. All *xy* and *yx* values must be zero or nonzero at all speeds.

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings:

  - **Number of supports** is set to 3 or 4.
  - **Intermediate (I1) mounting type** is set to `Speed-dependent bearing matrix`. For more information, see Mounting Type Parameter Dependencies.

Intermediate (I2) mounting type — Support *I2* mounting type
Clamped (default) | Pinned | Free | Bearing matrix | Speed-dependent bearing matrix

Type of mounting at the *I2* support. The *I2* support is located between the *I1* and *F1* supports.

Dependencies

This parameter is visible when both of these conditions are met:

- In the **Shaft** settings, **Model bending** is set to On. For more information, see Shaft Parameter Dependencies.
- In the **Supports** settings, **Number of supports** is set to 4.

Setting this parameter to Pinned, Free, Bearing matrix, or Speed-dependent bearing matrix exposes related parameters. For more information, see Mounting Type Parameter Dependencies.

Intermediate (I2) rotational damping [xx,yy] — Support *I2* rotational damping
[1e3, 1e3] N*m/(rad/s) (default) | vector

Rotational damping for the *I2* support. The elements of the two-element vector are:

- *xx* — Damping about the *x*-axis
- *yy* — Damping about the *y*-axis

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to On. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings:

  - **Number of supports** is set to 4.
  - **Intermediate (I2) mounting type** is set to Pinned, Free, Bearing matrix, or Speed-dependent bearing matrix. For more information, see Mounting Type Parameter Dependencies.

Intermediate (I2) translational damping [xx,xy,yx,yy] — Support *I2* translational damping
[2e4, -4.4e4, -4.4e4, 3.5e5] N/(m/s) (default) | vector

Translational damping for the *I2* support. The elements of the four-element vector are:

- *xx* — Damping in the *x*-axis direction
- *xy* — Damping in the *x*-axis direction coupled with motion in the *y*-axis direction
- *yx* — Damping in the *y*-axis direction coupled with motion in the *x*-axis direction
- *yy* — Damping in the *y*-axis direction

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to On. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings:

  - **Number of supports** is set to 4.

- **Intermediate (I2) mounting type** is set to `Bearing matrix`. For more information, see Mounting Type Parameter Dependencies.

Intermediate (I2) speed-dependent translational damping [xx1,xy1,yx1,yy1;...xxS,xyS,yxS,yyS] — Support *I2* speed-dependent translational damping
[2e4, -4.4e4, -4.4e4, 3.5e5; 1e4, -2.5e4, -2.5e4, 1.4e5; 5e3, -1.0e4, -1.0e4, 2.0e4] N/(m/s) (default) | matrix

Speed-dependent translational damping for the *I2* support. The number of rows in the matrix must equal the number of elements in the vector specified for the **Bearing speed [s1, ... sS]** parameter. Each row contains four elements:

- xx_s — Damping in the *x*-axis direction at the sth speed
- xy_s — Damping in the *x*-axis direction coupled with motion in the *y*-axis direction at the sth speed
- yx_s — Damping in the *y*-axis direction coupled with motion in the *x*-axis direction at the sth speed
- yy_s — Damping in the *y*-axis direction at the sth speed

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings:

  - **Number of supports** is set to 4.
  - **Intermediate (I2) mounting type** is set to `Speed-dependent bearing matrix`. For more information, see Mounting Type Parameter Dependencies.

Intermediate (I2) rotational stiffness [xx,yy] — Support *I2* rotational stiffness
[1e4, 1e4] N*m/rad (default) | nonnegative vector

Rotational stiffness for the *I2* support. The elements of the two-element vector are:

- *xx* — Stiffness about the *x*-axis
- *yy* — Stiffness about the *y*-axis

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings:

  - **Number of supports** is set to 4.
  - **Intermediate (I2) mounting type** is set to `Bearing matrix` or `Speed-dependent bearing matrix`. For more information, see Mounting Type Parameter Dependencies.

Intermediate (I2) translational stiffness [xx,xy,yx,yy] — Support *I2* translational stiffness
[1e6, -2e5, -3e5, 5e] N/m (default) | vector

Translational stiffness for the *I2* support. The elements of the four-element vector are:

- *xx* — Stiffness in the *x*-axis direction
- *xy* — Stiffness in the *x*-axis direction coupled with motion in the *y*-axis direction
- *yx* — Stiffness in the *y*-axis direction coupled with motion in the *x*-axis direction
- *yy* — Stiffness in the *y*-axis direction

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings:

  - **Number of supports** is set to `4`.
  - **Intermediate (I2) mounting type** is set to `Bearing matrix`. For more information, see Mounting Type Parameter Dependencies.

Intermediate (I2) speed-dependent translational stiffness [xx1,xy1,yx1,yy1;...xxS,xyS,yxS,yyS] — Support *I2* speed-dependent translational stiffness
`[1e6, -6e5, -3.5e6, 8e6; 1e6, -2e5, -3.0e6, 5e6; 1e6, -5e4, -2.5e6, 3e6]` N/m (default) | vector

Speed-dependent translational stiffness for the *I2* support. The number of rows in the matrix must equal the number of elements in the vector specified for the **Bearing speed [s1, … sS]** parameter. Each row contains four elements:

- xx_s — Stiffness in the *x*-axis direction at the sth speed
- xy_s — Stiffness in the *x*-axis direction coupled with motion in the *y*-axis direction at the sth speed
- yx_s — Stiffness in the *y*-axis direction coupled with motion in the *x*-axis direction at the sth speed
- yy_s — Stiffness in the *y*-axis direction at the sth speed

All *xx* and *yy* stiffness values must be positive. All *xy* and *yx* values must be zero or nonzero at all speeds.

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings:

  - **Number of supports** is set to `4`.
  - **Intermediate (I2) mounting type** is set to `Speed-dependent bearing matrix`. For more information, see Mounting Type Parameter Dependencies.

Follower (F1) mounting type — Support *F1* mounting type
`Clamped` (default) | `Pinned` | `Free` | `Bearing matrix` | `Speed-dependent bearing matrix`

Type of mounting at the follower end of the shaft.

Dependencies

This parameter is visible when, in the **Shaft** settings, the **Model bending** parameter is set to `On`. For more information, see Shaft Parameter Dependencies.

Setting this parameter to `Pinned`, `Free`, `Bearing matrix`, or `Speed-dependent bearing matrix` exposes related parameters. For more information, see Mounting Type Parameter Dependencies and Mounting Type Parameter Dependencies.

Follower (F1) rotational damping [xx,yy] — Support *F1* rotational damping
`[1e3, 1e3] N*m/(rad/s)` (default) | vector

Rotational damping for the *F1* support, which is the support that is located closest to **F**, the follower end of the shaft. The elements of the two-element vector are:

- *xx* — Damping about the *x*-axis
- *yy* — Damping about the *y*-axis

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings, **Follower (F1) mounting type** is set to `Pinned`, `Free`, `Bearing matrix`, or `Speed-dependent bearing matrix`. For more information, see Mounting Type Parameter Dependencies.

Follower (F1) translational damping [xx,xy,yx,yy] — Support *F1* translational damping
`[2e4, -4.4e4, -4.4e4, 3.5e5] N/(m/s)` (default) | vector

Translational damping for the *F1* support, which is the support that is closest to **F**, the follower end of the shaft. The elements of the four-element vector are:

- *xx* — Damping in the *x*-axis direction
- *xy* — Damping in the *x*-axis direction coupled with motion in the *y*-axis direction
- *yx* — Damping in the *y*-axis direction coupled with motion in the *x*-axis direction
- *yy* — Damping in the *y*-axis direction

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings, **Follower (F1) mounting type** is set to `Bearing matrix`. For more information, see Mounting Type Parameter Dependencies.

Follower (F1) speed-dependent translational damping [xx1,xy1,yx1,yy1;...xxS,xyS,yxS,yyS] — Support *F1* speed-dependent translational damping
`[2e4, -4.4e4, -4.4e4, 3.5e5; 1e4, -2.5e4, -2.5e4, 1.4e5; 5e3, -1.0e4, -1.0e4, 2.0e4] N/(m/s)` (default) | matrix

Speed-dependent translational damping for the *F1* support, which is the support that is closest to **F**, the follower end of the shaft.

The number of rows in the matrix must equal the number of elements in the vector specified for the **Bearing speed [s1, … sS]** parameter. Each row contains four elements:

- xx_s — Damping in the *x*-axis direction at the sth speed
- xy_s — Damping in the *x*-axis direction coupled with motion in the *y*-axis direction at the sth speed
- yx_s — Damping in the *y*-axis direction coupled with motion in the *x*-axis direction at the sth speed
- yy_s — Damping in the *y*-axis direction at the sth speed

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to On. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings, **Follower (F1) mounting type** is set to Speed-dependent bearing matrix. For more information, see Mounting Type Parameter Dependencies.

Follower (F1) rotational stiffness [xx,yy] — Support *F1* rotational stiffness
[1e4, 1e4] N*m/rad (default) | nonnegative vector

Rotational stiffness for the *F1* support, which is the support that is closest to **F**, the follower end of the shaft. The elements of the two-element vector are:

- *xx* — Stiffness about the *x*-axis
- *yy* — Stiffness about the *y*-axis

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to On. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings, **Follower (F1) mounting type** is set to Bearing matrix or Speed-dependent bearing matrix. For more information, see Mounting Type Parameter Dependencies and Mounting Type Parameter Dependencies.

Follower (F1) translational stiffness [xx,xy,yx,yy] — Support *F1* translational stiffness
[1e6, -2e5, -3e5, 5e] N/m (default) | vector

Translational stiffness for the *F1* support, which is the support that is closest to **F**, the follower end of the shaft. The elements of the four-element vector are:

- *xx* — Stiffness in the *x*-axis direction
- *xy* — Stiffness in the *x*-axis direction coupled with motion in the *y*-axis direction
- *yx* — Stiffness in the *y*-axis direction coupled with motion in the *x*-axis direction
- *yy* — Stiffness in the *y*-axis direction

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings, **Follower (F1) mounting type** is set to `Bearing matrix`. For more information, see Mounting Type Parameter Dependencies and Mounting Type Parameter Dependencies.

Follower (F1) speed-dependent translational stiffness [xx1,xy1,yx1,yy1;...xxS,xyS,yxS,yyS] — Support *F1* speed-dependent translational stiffness
[1e6, -6e5, -3.5e6, 8e6; 1e6, -2e5, -3.0e6, 5e6; 1e6, -5e4, -2.5e6, 3e6] N/m (default) | matrix

Speed-dependent translational stiffness for the *F1* support, which is the support that is closest to **F**, the follower end of the shaft.

The number of rows in the matrix must equal the number of elements in the vector specified for the **Bearing speed [s1, ... sS]** parameter. Each row contains four elements:

- xx_s — Stiffness in the *x*-axis direction at the sth speed
- xy_s — Stiffness in the *x*-axis direction coupled with motion in the *y*-axis direction at the sth speed
- yx_s — Stiffness in the *y*-axis direction coupled with motion in the *x*-axis direction at the sth speed
- yy_s — Stiffness in the *y*-axis direction at the sth speed

All *xx* and *yy* stiffness values must be positive. All *xy* and *yx* values must be zero or nonzero at all speeds.

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings, **Follower (F1) mounting type** is set to `Speed-dependent bearing matrix`. For more information, see Mounting Type Parameter Dependencies and Mounting Type Parameter Dependencies.

Bearing speed [s1,...,sS] — Support bearing rotational speed
[200, 500, 1500] rpm (default) | nonnegative vector

Support bearing rotational speed.

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to `On`. For more information, see Shaft Parameter Dependencies.
- In the **Support** settings, at least one of these parameters is set to `Speed-dependent bearing matrix`:

- **Base (B1) mounting type**
- **Intermediate (I1) mounting type**
- **Intermediate (I2) mounting type**
- **Follower (F1) mounting type**

For more information, see Mounting Type Parameter Dependencies and Mounting Type Parameter Dependencies.

Bending

The **Bending** settings are visible when, in the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .

The table shows how, in the **Bending** settings, the specified value for the **Rigid masses concentrically attached to shaft** parameter affects the visibility of related parameters. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Rigid Masses Concentrically Attached to Shaft Parameter Dependencies

Rigid Masses Concentrically Attached to Shaft		
None	Point mass	Disk
-	**Rigid mass distances from base (B)**	**Rigid mass distances from base (B)**
	Rigid masses	**Rigid masses**
		Rigid mass diametric moments of inertia about axis perpendicular to shaft
		Rigid mass polar moments of inertia about shaft axis

Damping constant proportional to mass — Damping constant α
10 1/s (default) | positive scalar

Damping constant, α, proportional to mass.

When the eigenmodes bending model is enabled, a translation damper in each modal mass-spring-damper system has the damping coefficient aM_{Mode}, where M_{Mode} is the modal mass.

When the lumped mass bending model is enabled, a damping matrix, αM is added to the system. M is the equation of motion mass matrix.

Dependencies

This parameter is visible when, in the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .

Damping constant proportional to stiffness — Damping constant β
1e-8 s (default) | positive scalar

Damping constant, β, proportional to stiffness.

When the lumped mass bending model is enabled, a damping matrix, βK is added to the system. K is the equation of motion stiffness matrix. When the eigenmodes bending model is enabled, a damping βK_{Modal} is added to the system.

Dependencies

This parameter is visible when, in the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .

Rigid masses concentrically attached to shaft — Type of rigid masses attached to shaft
None (default) | Point mass | Disk

Type, if any, of rigid masses attached to shaft.

Dependencies

This parameter is visible when, in the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .

The specified value for this parameter affects the visibility of related parameters. For more information, see Rigid Masses Concentrically Attached to Shaft Parameter Dependencies.

Rigid mass distances from base (B) — Rigid mass locations along shaft in distance from base node
0.5 m (default) | nonnegative scalar or vector

Rigid mass locations along the shaft in distance from **B**, the base end of the shaft. For multiple masses, specify an increasing row vector. The number of elements in the vector must be equal to the number of masses that are attached to the shaft. The value of the scalar or, for multiple masses, the largest value in the vector must not exceed the length of the shaft.

Dependencies

This parameter is visible when both of these conditions are met:

*   In the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .
*   In the **Bending** settings, the **Rigid masses concentrically attached to shaft** parameter is set to Point mass or Disk. For more information, see Rigid Masses Concentrically Attached to Shaft Parameter Dependencies.

Rigid masses — Mass of rigid masses concentrically attached to shaft
2 kg (default) | positive scalar or vector

Mass of rigid masses concentrically attached to shaft. For multiple masses, specify a row vector. The number and order of the elements in the vector must correspond to the elements in the vector specified for the **Rigid mass distances from base (B)** parameter.

Dependencies

This parameter is visible when both of these conditions are met:

*   In the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .
*   In the **Bending** settings, the **Rigid masses concentrically attached to shaft** parameter is set to Point mass or Disk. For more information, see Rigid Masses Concentrically Attached to Shaft Parameter Dependencies.

Rigid mass diametric moments of inertia about axis perpendicular to shaft — Diametric moments of inertia about axis perpendicular to shaft
0.025 kg*m^2 (default) | positive scalar or vector

Rigid mass moments of inertia about the axis perpendicular to the shaft. For multiple masses, specify a row vector. The number and order of elements in the vector must correspond to the elements in the vector specified for the **Rigid mass distances from base (B)** parameter.

Dependencies

This parameter is visible when both of these conditions are met:

- In the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .
- In the **Bending** settings, the **Rigid masses concentrically attached to shaft** parameter is set to Disk. For more information, see Rigid Masses Concentrically Attached to Shaft Parameter Dependencies.

Rigid mass polar moments of inertia about shaft axis — Polar moments of inertia about shaft axis
0.050 kg*m^2 (default) | positive scalar or vector

Rigid polar mass moments of inertia about the principle shaft axis. For multiple masses, specify a row vector. The number and order of the elements in the vector must correspond to the elements in the vector specified for the **Rigid mass distances from base (B)** parameter.

Dependencies

This parameter is visible when both of these conditions are met:

- In the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .
- In the **Bending** settings, the **Rigid masses concentrically attached to shaft** parameter is set to Disk. For more information, see Rigid Masses Concentrically Attached to Shaft Parameter Dependencies.

Static unbalances that excite bending — Excitational static unbalances
[.01, .01] m*kg (default) | nonnegative vector

Static unbalances that excite bending.

Dependencies

This parameter is visible when, in the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .

Static unbalance distances from base (B) — Static unbalance distances from base
[.33, .67] m (default) | nonnegative vector

Distance of excitational static unbalances from **B**, the base end of the shaft.

Dependencies

This parameter is visible when, in the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .

Static unbalance offset angles — Static unbalance offset angles
[0, pi] rad (default) | nonnegative vector

Initial angle, about the center line of the shaft relative to the *x*-axis, of the excitational static unbalances.

Dependencies

This parameter is visible when, in the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .

Advanced Bending

The **Advanced Bending** settings are exposed when, in the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .

In the **Advanced Bending** settings, the value of the **Bending vibration analysis method** parameter affects the visibility of parameters in the **Supports** settings. For more information, see "Bending vibration analysis method" on page 1-0 .

The Advanced Bending Parameter Dependencies table shows how the visibility of certain parameters in the **Advanced Bending** settings depends on the values for a combination of parameters in one or more of these settings:

- **Shaft**
- **Supports**
- **Advanced Bending**

For example, the **Simulated eigenmode dependency on rotation speed** parameter is visible if all these conditions are met:

- In the **Shaft** settings, **Model bending** is set to On.
- In the **Supports** settings, at least one **Mounting type** parameter is set to Speed-dependent bearing matrix.
- In the **Advanced Bending** settings, **Bending vibration analysis method** is set to Eigenmodes.

Advanced Bending Parameter Dependencies

Setting > Parameter	Specified Parameter Values and Dependent Advanced Bending Parameters								
Shaft > Model bending	`Off`	`On`							
Supports > Mounting type	-	Any combination of `Clamped, Pinned, Free, or Bearing matrix`			*At least one* `Speed-dependent bearing matrix` **and any combination of** `Clamped, Pinned, Free, or Bearing matrix`				
The columns to the right show the **Advanced Bending** parameters that are visible for the parameter configuration set shown in the corresponding column in the heading rows of this table.	-	**Bending vibration analysis method**			**Bending vibration analysis method**				
		`Lumped mass`	`Eigenmodes`		`Lumped mass`	`Eigenmodes`			
			Bending mode determination			**Bending mode determination**			
			`Simscape determined`	`User Defined`		`Simscape determined`		`User Defined`	
			Limit number of modes			**Limit number of modes**			
						Simulated eigenmode dependency on rotation speed		**Simulated eigenmode dependency on rotation speed**	
						`Static`	`Dynamic`	`Static`	`Dynamic`
						Nominal shaft speed for bending modes		**Nominal shaft speed for bending modes**	
			Eigenfrequency upper limit			**Eigenfrequency upper limit**	**Eigenfrequency upper limit**		
			Shaft length increments for mode shape computations	**Modal frequencies [m]**		**Shaft length increments for mode shape computations**	**Shaft length increments for mode shape computations**	**Modal frequencies [s, m]**	**Modal frequencies [s, m]**
				Shaft position [z]				**Shaft position [z]**	**Shaft position [z]**

1-223

Setting > Parameter	Specified Parameter Values and Dependent Advanced Bending Parameters							
Shaft > Model bending	Off	On						
Supports > Mounting type	-	Any combination of Clamped, Pinned, Free, or Bearing matrix			At least one Speed-dependent bearing matrix and any combination of Clamped, Pinned, Free, or Bearing matrix			
				X-direction mode shapes [z,m]				X-direction mode shapes [z,m,s]	X-direction mode shapes [z,m,s]
				Y-direction mode shapes [z,m]				Y-direction mode shapes [z,m,s]	Y-direction mode shapes [z,m,s]

Bending vibration analysis method — Bending vibration analysis method
Lumped mass (default) | Eigenmodes

Method for analyzing the bending vibration. For more information, see "Bending Model Lumped Mass Method" on page 1-184 and "Bending Model Eigenmodes Method" on page 1-190.

Dependencies

This parameter is visible when, in the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .

The value of this parameter affects the visibility of other parameters in the **Advanced Bending** settings. For more information, see Advanced Bending Parameter Dependencies.

Bending mode determination — Bending mode determination method
Simscape determined (default) | User defined

Method for determining the eigenmode frequencies and shapes:

- Simscape determined — Simscape determines the mode based on boundary conditions that you specify.
- User defined — Specify the eigenmode frequencies and shapes directly.

Dependencies

This parameter is visible when both of these conditions are met:

- In the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .
- In the **Advanced Bending** settings, the **Bending vibration analysis method** parameter is set to Eigenmodes. For more information, see Advanced Bending Parameter Dependencies.

The value of this parameter affects the visibility of other parameters in the **Advanced Bending** settings. For more information, see Mounting Type Parameter Dependencies.

`Limit number of modes` — **Maximum number of modes**
6 (default) | positive integer scalar

Maximum number of modes that Simscape determines.

Dependencies

This parameter is visible when both of these conditions are met:

- In the **Shaft** settings, the **Model bending** parameter is set to `On`. For more information, see "Model bending" on page 1-0   .

- In the **Advanced Bending** settings, the **Bending vibration analysis method** parameter is set to `Eigenmodes` and the **Bending mode determination** parameter is set to `Simscape determined`. For more information, see Advanced Bending Parameter Dependencies.

`Simulated eigenmode dependency on rotation speed` — **Eigenmode dependency**
`Static` (default) | `Dynamic`

Simulated dependency of the eigenmode properties on shaft rotation speed:

- Static — The bending analysis holds the eigenmode properties constant during changes in the rotational speed of the shaft.

- Dynamic — The bending analysis adjusts the eigenmode properties as the rotational speed of the shaft changes. The block uses the elements in the vector specified for the **Bearing speed [s1,..., sS]** parameter as lookup table reference points. For this model, the relative magnitudes of speed-dependent translational stiffness elements may not change at each bearing speed.

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, the **Model bending** parameter is set to `On`. For more information, see "Model bending" on page 1-0   .

- In the **Supports** settings, at least one mounting type parameter is set to `Speed-dependent bearing matrix`.

- In the **Advanced Bending** settings, the **Bending vibration analysis method** parameter is set to `Eigenmodes`. For more information, see Advanced Bending Parameter Dependencies.

`Nominal shaft speed for bending modes` — **Nominal shaft speed**
5000 rpm (default) | scalar

Rated shaft speed for the bending mode analysis.

Dependencies

This parameter is visible if all these conditions are met:

- In the **Shaft** settings, the **Model bending** parameter is set to `On`.

- In the **Supports** settings, *at least one mounting type parameter* are set to `Speed-dependent bearing matrix`.

- In the **Advanced Bending** settings:

- The **Bending vibration analysis method** parameter is set to `Eigenmodes`.
- The **Simulated eigenmode dependency on rotation speed** parameter is set to `Static`.

For more information, see Advanced Bending Parameter Dependencies.

`Eigenfrequency upper limit` — Eigenfrequency upper limit
`1e8` rpm (default) | positive integer scalar

Eigenfrequency upper limit.

Dependencies

This parameter is visible when both of these conditions are met:

- In the **Shaft** settings, the **Model bending** parameter is set to `On`. For more information, see "Model bending" on page 1-0 .
- In the **Advanced Bending** settings, the **Bending vibration analysis method** parameter is set to `Eigenmodes` and the **Bending mode determination** parameter is set to `Simscape determined`. For more information, see Advanced Bending Parameter Dependencies.

`Shaft length increments for mode shape computations` — Shaft length increments
`0.01` m (default) | positive scalar

Shaft length increments used for mode mass and shape computations. For more information, see "Bending Model Eigenmodes Method" on page 1-190.

Dependencies

This parameter is visible when both of these conditions are met:

- In the **Shaft** settings, the **Model bending** parameter is set to `On`. For more information, see "Model bending" on page 1-0 .
- In the **Advanced Bending** settings, the **Bending vibration analysis method** parameter is set to `Eigenmodes` and the **Bending mode determination** parameter is set to `Simscape determined`. For more information, see Advanced Bending Parameter Dependencies.

`Modal frequencies [m]` — Bearing-speed-independent modal frequencies
`[280, 644, 615, 1285]` rad/s (default) | nonnegative vector

Modal frequencies for the bearing-speed-independent model. For more information, see "Bending Model Eigenmodes Method" on page 1-190.

Dependencies

This parameter is visible when both of these conditions are met:

- In the **Shaft** settings, the **Model bending** parameter is set to `On`. For more information, see "Model bending" on page 1-0 .
- In the **Supports** settings, no mounting types is set to `Speed-dependent bearing matrix`.
- In the **Advanced Bending** settings:

  - **Bending vibration analysis method** is set to `Eigenmodes`.
  - **Bending mode determination** is set to `User defined`.

For more information, see Advanced Bending Parameter Dependencies.

`Modal frequencies [s,m]` — Bearing-speed-dependent modal frequencies
`[257, 780, 580, 1560; 280, 644, 615, 1285; 294, 513, 636, 1020]` rad/s (default) |
nonnegative matrix

Modal frequencies for the bearing-speed-dependent model. For more information, see "Bending
Model Eigenmodes Method" on page 1-190.

Dependencies

This parameter is visible when both of these conditions are met:

- In the **Shaft** settings, the **Model bending** parameter is set to `On`. For more information, see "Model bending" on page 1-0   .
- In the **Supports** settings, at least one mounting type is set to `Speed-dependent bearing matrix`.
- In the **Advanced Bending** settings:

  - **Bending vibration analysis method** is set to `Eigenmodes`.
  - **Bending mode determination** is set to `User defined`.

For more information, see Advanced Bending Parameter Dependencies.

`Shaft position [z]` — Shaft position
`[0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]` (default) | increasing nonnegative
vector

Shaft position for mode shapes. The number of elements in the vector correspond to the number of
rows in *X*-direction and *Y*-direction mode shapes. For more information, see "Bending Model
Eigenmodes Method" on page 1-190.

Dependencies

This parameter is visible when both of these conditions are met:

- In the **Shaft** settings, the **Model bending** parameter is set to `On`. For more information, see "Model bending" on page 1-0   .
- In the **Advanced Bending** settings:

  - **Bending vibration analysis method** is set to `Eigenmodes`.
  - **Bending mode determination** is set to `User defined`.

For more information, see Advanced Bending Parameter Dependencies.

`X-direction mode shapes [z,m]` — Bearing-speed-independent *X*-direction mode shapes
`[.1648, .0086, -.2759, .0162; .1671, .0093, -.2231, .0133; .1691, .01, -.1689, .0102; .1707, .0105, -.1135, .007; .1717, .0108, -.057, .0035; .1721, .0109, 0, 0; .1717, .0108, .057, -.0035; .1707, .0105, .1135, -.007; .1691, .01, .1689, -.0102; .1671, .0093, .2231, -.0133; .1648, .0086, .2759, -.0162]` (default) |
matrix

The *X*-direction mode shapes matrix for the bearing-speed-independent model. For more information,
see "Bending Model Eigenmodes Method" on page 1-190.

The matrix must have dimensions *z*-by-*m*, where:

- *z* is the number of elements in the specified vector for the **Shaft position [z]** parameter.
- *m* is the number of columns in the specified vector for the **Modal frequencies** parameter.

The mode shape matrix has the form [U_{1x}, U_{2x}, ..., U_{mx}], where each column is the mode shape deflection in the *X* direction for the mth mode. The algorithm computes the modal properties based on the parameters in the **Shaft** and **Bending** settings.

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .
- In the **Supports** settings, no mounting type is set to Speed-dependent bearing matrix.
- In the **Advanced Bending** settings:
  - **Bending vibration analysis method** is set to Eigenmodes.
  - **Bending mode determination** is set to User defined.

For more information, see Advanced Bending Parameter Dependencies.

X-direction mode shapes [z,m,s] — Bearing-speed-dependent *X*-direction mode shapes
```
cat(3, [.1842, .0134, -.3072, .0272; .1863, .0152,
-.2482, .0226; .1882, .0169, -.1878, .0176; .1897, .0182,
-.1261, .0121; .1906, .019, -.0634, .0062; .191, .0193, 0,
0; .1906, .019, .0634, -.0062; .1897, .0182, .1261,
-.0121; .1882, .0169, .1878, -.0176; .1863, .0152, .2482,
-.0226; .1842, .0134, .3072, -.0272], [.1648, .0086,
-.2759, .0162; .1671, .0093, -.2231, .0133; .1691, .01,
-.1689, .0102; .1707, .0105, -.1135, .007; .1717, .0108,
-.057, .0035; .1721, .0109, 0, 0; .1717, .0108, .057,
-.0035; .1707, .0105, .1135, -.007; .1691, .01, .1689,
-.0102; .1671, .0093, .2231, -.0133; .1648, .0086, .2759, -.0162],
[.1291, .0046, -.2166, .0082; .131, .0048, -.1751, .0067; .1327, .005,
-.1327, .0051; .1341, .0052, -.0891, .0035; .135, .0053,
-.0448, .0017; .1353, .0053, 0, 0; .135, .0053, .0448,
-.0017; .1341, .0052, .0891, -.0035; .1327, .005, .1327,
-.0051; .131, .0048, .1751, -.0067; .1291, .0046, .2166, -.0082]) (default) |
matrix
```

The *X*-direction mode shapes matrix for the bearing-speed-dependent model. For more information, see "Bending Model Eigenmodes Method" on page 1-190.

The matrix must have dimensions *z*-by-*m*-by-*s*, where:

- *z* is the number of elements in the specified vector for the **Shaft position [z]** parameter.
- *m* is the number of coplumns in the specified vector for the **Modal frequencies [z,m]** parameter.
- *s* is the number of elements in the specified vector for the **Bearing speed [s1,…, sS]** parameter.

The mode shape matrix has the form cat(3,[U_{1x1}, U_{2x1}, ..., U_{mx1}], ..., [U_{1xs}, U_{2xs}, ..., U_{mxs}]), where each column is the mode shape deflection in the *x* direction, for the mth mode. Each page corresponds to an element in the vector specified for the **Bearing speed [s1,…, sS]** parameter. The algorithm computes the modal properties based on the parameters in the **Shaft** and **Bending** settings.

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, the **Model bending** parameter is set to `On`. For more information, see "Model bending" on page 1-0 .
- In the **Supports** settings, at least one mounting type is set to `Speed-dependent bearing matrix`.
- In the **Advanced Bending** settings:

  - **Bending vibration analysis method** is set to `Eigenmodes`.
  - **Bending mode determination** is set to `User defined`.

For more information, see Advanced Bending Parameter Dependencies.

Y-direction mode shapes [z,m] — Bearing-speed-independent *Y*-direction mode shapes
[.1193, -.1784, -.1997, -.3348; .1209, -.1933, -.1615, -.2751; .1224, -.2068, -.1223, -.2119; .1235, -.2176, -.0821, -.1445; .1243, -.2245, -.0413, -.0734; .1246, -.2269, 0, 0; .1243, -.2245, .0413, .0734; .1235, -.2176, .0821, .1445; .1224, -.2068, .1223, .2119; .1209, -.1933, .1615, .2751; .1193, -.1784, .1997, .3348] (default) | matrix

The *Y*-direction mode shapes matrix for the bearing-speed-independent model. For more information, see "Bending Model Eigenmodes Method" on page 1-190.

The matrix must have dimensions *z*-by-*m*, where:

- *z* is the number of elements in the specified vector for the **Shaft position [z]** parameter.
- *m* is the number of columns in the specified vector for the **Modal frequencies** parameter.

The mode shape matrix has the form $[U_{1y}, U_{2y}, ..., U_{my}]$, where each column is the mode shape deflection in the *Y* direction, for the mth mode. The algorithm computes the modal properties based on the parameters in the **Shaft** and **Bending** settings.

Dependencies

This parameter is visible when both of these conditions are met:

- In the **Shaft** settings, the **Model bending** parameter is set to `On`. For more information, see "Model bending" on page 1-0 .
- In the **Supports** settings, no mounting type is set to `Speed-dependent bearing matrix`.
- In the **Advanced Bending** settings:

  - **Bending vibration analysis method** is set to `Eigenmodes`.
  - **Bending mode determination** is set to `User defined`.

For more information, see Advanced Bending Parameter Dependencies.

Y-direction mode shapes [z,m,s] — Bearing-speed-dependent *y*-direction mode shapes
cat(3, [.0885, -.1625, -.1475, -.3301; .0895, -.1847, -.1192, -.2745; .0904, -.2048, -.0902, -.2141; .0911, -.2209, -.0606, -.1475; .0916, -.2313, -.0304, -.0754; .0917, -.2349, 0, 0; .0916, -.2313, .0304, .0754; .0911, -.2209, .0606, .1475; .0904, -.2048, .0902, .2141; .0895,

```
-.1847, .1192, .2745; .0885, -.1625, .1475, .3301], [.1193, -.1784, -.1997,
-.3348; .1209, -.1933, -.1615, -.2751; .1224, -.2068, -.1223, -.2119; .1235,
-.2176, -.0821, -.1445; .1243, -.2245, -.0413, -.0734; .1246, -.2269, 0,
0; .1243, -.2245, .0413, .0734; .1235, -.2176, .0821, .1445; .1224,
-.2068, .1223, .2119; .1209, -.1933, .1615, .2751; .1193,
-.1784, .1997, .3348], [.1566, -.1901, -.2628, -.3377; .1589, -.1994, -.2125,
-.2753; .161, -.2079, -.1609, -.2103; .1627, -.2146, -.1081, -.1424; .1638,
-.219, -.0544, -.0719; .1642, -.2205, 0, 0; .1638,
-.219, .0544, .0719; .1627, -.2146, .1081, .1424; .161,
-.2079, .1609, .2103; .1589, -.1994, .2125, .2753; .1566,
-.1901, .2628, .3377]) (default) | matrix
```

The *Y*-direction mode shapes matrix for the bearing-speed-dependent model. For more information, see "Bending Model Eigenmodes Method" on page 1-190.

The matrix must have dimensions *z*-by-*m*-by-*s*, where:

- *z* is the number of elements in the specified vector for the **Shaft position [z]** parameter.
- *m* is the number of columns in the specified vector for the **Modal frequencies [z,m]** parameter.
- *s* is the number of elements in the specified vector for the **Bearing speed [s1,…, sS]** parameter.

The mode shape matrix has the form cat(3,[U_{1y1}, U_{2y1}, …, U_{my1}], …, [U_{1ys}, U_{2ys}, …, U_{mys}]), where each column is the mode shape deflection in the *y* direction, for the mth mode. Each page corresponds to an element in the vector specified for the **Bearing speed [s1,…, sS]** parameter. The algorithm computes the modal properties based on the parameters in the **Shaft** and **Bending** settings.

Dependencies

This parameter is visible when all these conditions are met:

- In the **Shaft** settings, the **Model bending** parameter is set to On. For more information, see "Model bending" on page 1-0 .
- In the **Supports** settings, at least one mounting type is set to Speed-dependent bearing matrix.
- In the **Advanced Bending** settings:

  - **Bending vibration analysis method** is set to Eigenmodes.
  - **Bending mode determination** is set to User defined.

For more information, see Advanced Bending Parameter Dependencies.

References

[1] Adams, M.L. *Rotating Machinery Vibration*. CRC Press, NY: 2010.

[2] Bathe, K. J. *Finite Element Procedures*. Prentice Hall, 1996.

[3] Chudnovsky, V., D. Kennedy, A. Mukherjee, and J. Wendlandt. Modeling Flexible Bodies in SimMechanics and Simulink. *MATLAB Digest,* Volume 14, Number 3. May 2006.

[4] Miller, S., T. Soares, Y. Van Weddingen, J. Wendlandt. Modeling Flexible Bodies with Simscape Multibody. The MathWorks, 2017.

[5] Muszynska, A. *Rotordynamics*. Taylor & Francis, 2005

[6] Rao, S.S. *Vibration of Continuous Systems*. Hoboken, NJ: John Wiley & Sons, 2007.

See Also

Inertia | Rod | Rotational Damper | Rotational Spring | Torsional spring damper | `fminbnd`

Topics
"Resolve Partitioning Solver Simulation Issues for Simscape Driveline Models"

Introduced in R2018b

Force Noise Source

Model zero-mean normally (Gaussian) distributed force
Library: Simscape / Driveline / Sources

Description

The Force Noise Source block generates zero-mean normally (Gaussian) distributed force using the Random Number source in the Simscape Foundation library. The force produced by the block depends on two factors:

- The noise power spectral density input to the block
- A random number seed that you control using the **Repeatability** and, as applicable, **Seed** parameters

The defining equation for the force that the block generates is

$$F = \sqrt{PSD/2}\frac{N(0, 1)}{\sqrt{h}},$$

where:

- F is the output force.
- PSD is the single-sided noise power spectral density.
- N is a Gaussian random number with zero mean and standard deviation of one.
- h is the sampling interval.

Ports

Input

P — Single-sided noise power spectral density
physical signal

The power spectral density (PSD) of noise is the average noise power per unit of bandwidth, in N^2/Hz.

Conserving

B — Base interface
mechanical translational

Mechanical translational conserving port associated with the base interface.

F — Follower interface
mechanical translational

Mechanical translational conserving port associated with the follower interface.

Parameters

`Sample time` — Time step period and offset
`1e-3` s (default) | `[`*`step`*`,` *`offset`*`]` s

The values of the time *step* period and the initial time *offset*. If you specify a scalar value for *step*, the block assumes an *offset* value of `0`.To specify a nonzero value for the initial time offset, specify the parameter values using the vector `[`*`step`*`,` *`offset`*`]`. The *offset* value must be less than the *step* value and greater than or equal to zero.

`Repeatability` — Random number seed control
`Not repeatable` (default) | `Repeatable` | `Specify seed`

The random number seed is the number that initializes the random number generator. The seed is `0` or a positive integer. To control the random number seed, set this parameter to:

*   `Not repeatable` — The seed changes every time you simulate your model. The block resets the random seed using the MATLAB® random number generator command:

    `seed = randi(2^32-1);`
*   `Repeatable` — The seed is the same random number at the start of every simulation. The block sets the value using the same MATLAB random number generator command used by the `Not repeatable` parameter.

    When you add a Force Noise Source block to your model from the Sources library, the block generates and stores a random value for the repeated seed. When you make a copy of the Force Noise Source block from an existing block in a model, the copy generates a new random value for the repeated seed.
*   `Specify seed` — The seed is a number that you specify using the **Seed** parameter. The **Seed** parameter is only available when you choose `Specify seed` for the **Repeatability** parameter.

`Seed` — Random number generation seed value
`0` (default)

The seed must be `0` or a positive integer. This parameter is only available when you select `Specify seed` for the **Repeatability** parameter.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Random Number | Rotational Velocity Noise Source | Sinusoidal Force Source | Torque Noise Source | Translational Velocity Noise Source

Functions
`randi`

Introduced in R2016b

Fundamental Friction Clutch

Friction clutch with kinetic and static-limit friction torques as inputs
Library: Simscape / Driveline / Clutches / Fundamental Components
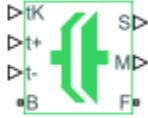
Description

The Fundamental Friction Clutch block represents a mechanism that transmits rotational power through friction. The clutch contains two friction plate sets, each rigidly connected to a driveshaft, that come into contact to engage. Once in contact, the plates experience frictional torques that enable power to flow between the driveshafts.

The clutch can be bidirectional or unidirectional. A bidirectional clutch can slip in the positive and negative directions. A unidirectional clutch can slip only in the positive direction. The slip direction is positive if the follower shaft spins faster than the base shaft and negative if it spins slower. The block defines the slip velocity as the difference

$$\omega = \omega_F - \omega_B,$$

where:

- ω is the slip velocity.
- ω_F is the angular velocity of the follower driveshaft.
- ω_B is the angular velocity of the base driveshaft.

Clutch States

The clutch can be in three states:

- Locked — Real clutch state in which the friction plates spin as a unit. A locked clutch has one rotational degree of freedom. It experiences no power losses due to friction.
- Unlocked — Real clutch state in which the friction plates slip with respect to each other. An unlocked clutch has two rotational degrees of freedom. It experiences power losses equal to the product of the slip velocity and the kinetic friction torque.
- Wait — Virtual clutch state that maintains the motion of the previous state while testing for locking and unlocking. The clutch degrees of freedom and power losses depend on the previous clutch state.

The schematic shows the conditions under which the clutch is locked and unlocked. The clutch is generally locked if the torque it transfers lies between its static friction torque limits and the magnitude of the slip velocity is smaller than its velocity tolerance. The clutch unlocked otherwise.
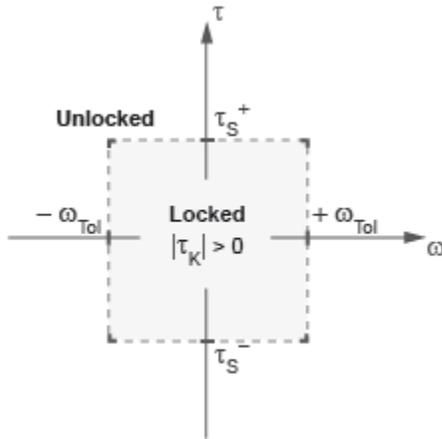
In the schematic:

- τ is the torque transferred between the clutch plates.
- τ_S^- and τ_S^+ are the static friction torque limits.
- τ_K is the kinetic friction torque between the clutch plates.
- ω_{Tol} is the slip velocity tolerance of the clutch.
- ω is the slip velocity between the clutch plates.

The block identifies the clutch state through physical signal port **M** using values -1, 0, and +1. The table summarizes the correspondence between the states and the output values.

| State	Value
Unlocked Forward or Wait Forward	+1
Unlocked Reverse or Wait Reverse	-1
Locked or Unlocked Initial State	0

State Transitions

At the start of simulation, the clutch is in one of two states — locked or initial unlocked. The initial unlocked state is unique in that it lacks a direction of motion. The clutch remains in this state until the clutch slip velocity becomes nonzero. The clutch then transitions to the appropriate state, unlocked reverse or unlocked forward, according to the schematic.

During simulation, the clutch tests various dynamic conditions to determine the appropriate state transitions, if any. The schematics show the possible transitions, their dynamic conditions, and the resulting states. If the clutch is unidirectional, the schematic reduces to the right half. The diagram shows transitions for a one-way clutch.
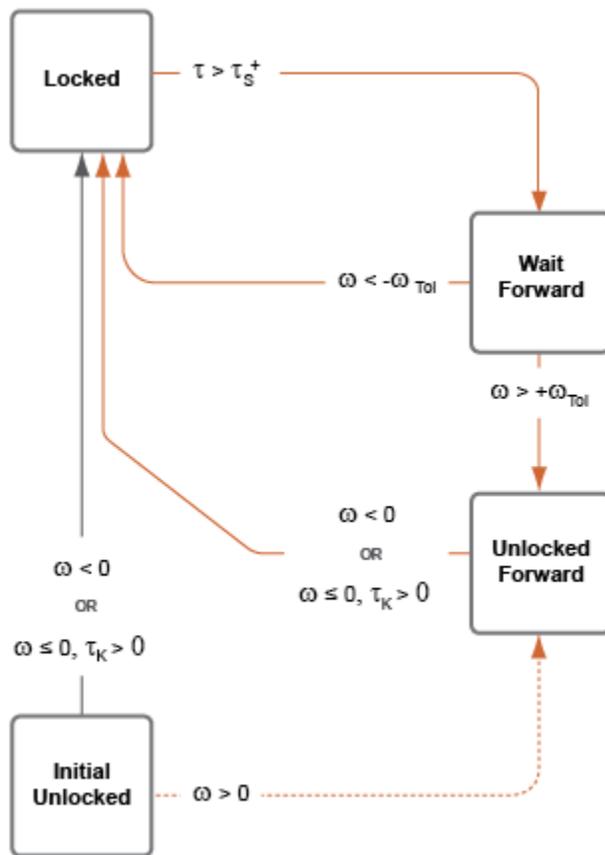
The diagram shows transitions for a two-way clutch.
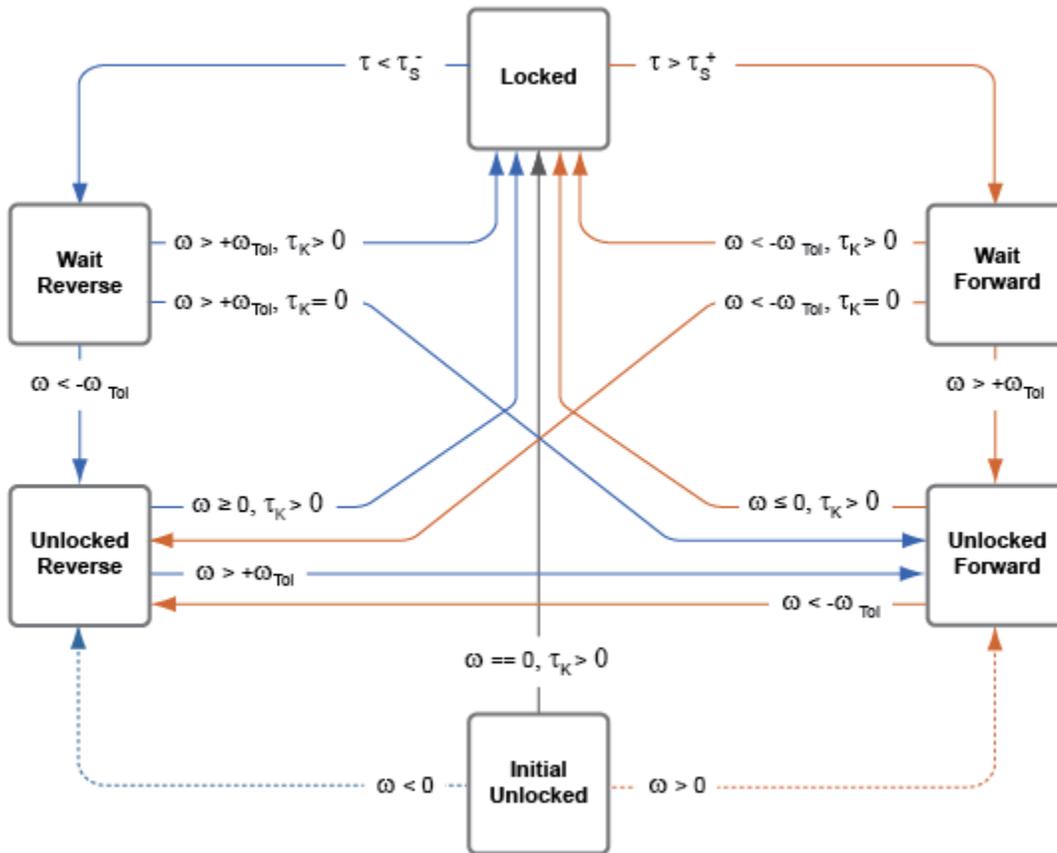
Thermal Model

The Fundamental Friction Clutch block simulates the effects of heat flow and heat generated relative to the amount of power dissipated when you enable the thermal port. To simulate heat flow, you must connect the port to another block with an enabled thermal port. Many Simscape Driveline blocks have thermal modelling options. When a parameter that supports thermal modeling is available, it appears as a vector whose length corresponds to the length of **Temperature**. For more information, see "Model Thermal Losses in Driveline Components".

To enable the thermal port, set **Thermal Port** to `Model`. Enabling the thermal port also enables these settings:

- **Parameters > Thermal Mass**
- **Variables > Temperature**

Variables

Use the **Variables** settings to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Dependencies

To enable these settings, set **Thermal port** to `Model`.

Ports

Input

tK — Kinetic friction torque, N*m
physical signal

Physical signal input port for the kinetic friction torque.

t+ — Static friction torque upper bound, N*m
physical signal

Physical signal input port for the upper static friction torque limit.

t- — Static friction torque lower bound, N*m
physical signal

Physical signal input port for the lower static friction torque limit.

Output

S — Clutch slip velocity, rad/sec
physical signal

Physical signal output port for the clutch slip velocity.

M — Clutch state
physical signal

Physical signal output port for the clutch state.

Conserving

B — Rotational velocity, rad/sec
mechanical rotational

Rotational conserving ports associated with the driving shaft.

F — Rotational speed, rad/sec
mechanical rotational

Rotational conserving ports associated with the driven shaft.

H — Heat flow
thermal

Thermal conserving port associated with heat flow. Thermal ports allow you to model heat flow between a block and the connected network. For more information, see "Model Thermal Losses in Driveline Components".

Dependencies

To enable these settings, set **Thermal Port** to Model.

Parameters

`Directionality` — **Permissible inter-plate slip directions**
`Bidirectional` (default) | `Unidirectional`

Slip directions the clutch allows between its plates. A bidirectional clutch allows positive and negative slip velocities. A unidirectional clutch allows only positive slip velocities.

The unidirectional clutch is equivalent to a friction clutch connected in parallel to a one-way clutch that disengages only when the slip velocity becomes positive. To model a unidirectional clutch with slip in the negative direction, reverse the base and follower port connections.

`Thermal port` — **Thermal model**
`Omit` (default) | `Model`

Model for heat flow and temperature change:

- `Omit` — Neglect thermal dynamics.
- `Model` — Include thermal dynamics.

`Thermal mass` — **Resistance to temperature change**
`50 kJ/K` (default) | scalar

Thermal energy required to change the component temperature by a single thermal unit. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

To enable this parameter, set **Thermal Port** to `Model`.

`Clutch velocity tolerance` — **Slip velocity threshold**
`0.001 rad/s` (default) | scalar

Slip velocity below which the clutch can lock. The clutch locks if, after falling below the clutch velocity tolerance, the kinetic friction torque is nonzero and the transferred torque is between the static friction torque limits.

`Initial state` — **Initial clutch state**
`Unlocked` (default) | `Locked`

Clutch state at the start of simulation. The clutch can be in one of two states, locked and unlocked. A locked clutch constrains the base and follower shafts to spin at the same velocity, that is, as a single unit. An unlocked clutch allows the two shafts to spin at different velocities, resulting in slip between the clutch plates.

A clutch that starts simulation unlocked lacks a direction of motion. For this reason, after checking that the unlocked state is valid, the block automatically determines the appropriate direction of motion based on the driveline dynamics. Based on the direction of motion, the clutch then transitions to an unlocked-reverse or unlocked-forward state.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Cone Clutch | Differential | Disk Friction Clutch | Dog Clutch | Double-Sided Synchronizer | Logic-Controlled Clutch | Synchronizer | Unidirectional Clutch | Variable Ratio Transmission

Topics

"Gear Coupling Control Using Clutches"

Introduced in R2011a

Generic Engine

Internal combustion engine with throttle and rotational inertia and time lag
Library: Simscape / Driveline / Engines & Motors

Description

The Generic Engine block represents a general internal combustion engine. This block is a suitable generic engine for spark-ignition and diesel. Speed-power and speed-torque parameterizations are provided. A throttle physical signal input specifies the normalized engine torque. Optional dynamic parameters include crankshaft inertia and response time lag. A physical signal port outputs the engine fuel consumption rate based on the fuel consumption model that you choose. Optional speed and redline controllers prevent engine stall and enable cruise control.
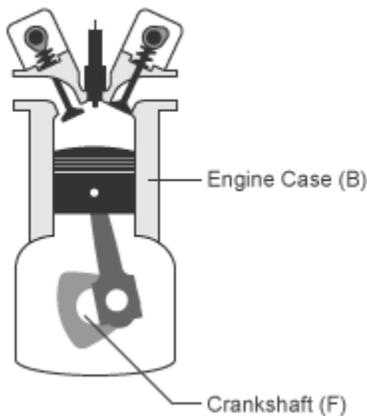
Engine Speed, Throttle, Power, and Torque

By default, the Generic Engine block uses a programmed relationship between torque and speed that is modulated by the throttle signal.

The block evaluates engine power demand as a function of engine speed, $g(\Omega)$. The function provides the maximum power available for a given engine speed, Ω. The block parameters—**Maximum power**,**Speed at maximum power**,and **Maximum speed**— normalize this function to physical maximum torque and speed values.

The normalized throttle input signal T specifies the actual engine power. The power is delivered as a fraction of the maximum power possible in a steady state at a fixed engine speed. It modulates the actual power delivered, P, from the engine: $P(\Omega,T) = T{\cdot}g(\Omega)$. The engine torque is $\tau = P/\Omega$.

Engine Power Demand

The engine power is nonzero when the speed is limited to the operating range, $\Omega_{min} \leq \Omega \leq \Omega_{max}$. The absolute maximum engine power P_{max} defines Ω_0 such that $P_{max} = g(\Omega_0)$. Define $w \equiv \Omega/\Omega_0$ and $g(\Omega) \equiv P_{max} \cdot p(w)$. Then $p(1) = 1$ and $dp(1)/dw = 0$. The torque function is:

$\tau = (P_{max}/\Omega_0) \cdot [p(w)/w]$.

You can derive forms for $p(w)$ from engine data and models. The Generic Engine block uses a third-order polynomial form

$p(w) = p_1 \cdot w + p_2 \cdot w^2 - p_3 \cdot w^3$

that satisfies

$p_1 + p_2 - p_3 = 1$, $p_1 + 2p_2 - 3p_3 = 0$.

In typical engines, the p_i are positive. This polynomial has three zeros, one at $w = 0$, and a conjugate pair. One of the pair is positive and physical; the other is negative and unphysical:

$$w_{\pm} = \frac{1}{2}\left(-p_2 \pm \sqrt{p_2^2 + 4p_1 p_3}\right).$$

For the engine power polynomial, there are restrictions on the polynomial coefficients p_i to achieve a valid power-speed curve. If you use tabulated power or torque data, the corresponding restrictions on $P(\Omega)$ apply.
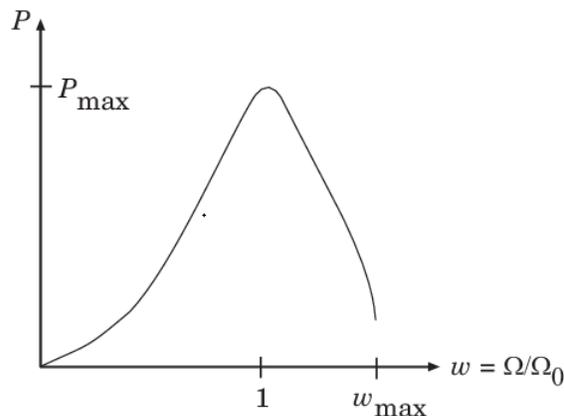
Typical Engine Power Demand Curve

Determine the speed and power, $w = \Omega/\Omega_0$ and $p = P(\Omega)/P_0$, and define the boundaries as $w_{min} = \Omega_{min}/\Omega_0$ and $w_{max} = \Omega_{max}/\Omega_0$. Then:

- The block restricts the engine speed to a positive range above the minimum speed and below the maximum speed: $0 \leq w_{min} \leq w \leq w_{max}$.
- The engine power at minimum speed must be nonnegative: $p(w_{min}) \geq 0$. If you use the polynomial form, this condition is a restriction on the p_i:

  $p(w_{min}) = p_1 \cdot w_{min} + p_2 \cdot w^2_{min} - p_3 \cdot w_{min}^3 \geq 0$.
- The engine power at maximum speed must be nonnegative: $p(w_{max}) \geq 0$. If you use the polynomial form, this condition is a restriction on w_{max}: $w_{max} \leq w_+$.

Engine Power Forms for Different Engine Types

For the default parameterization, the block provides two choices of internal combustion engine types, each with different engine power demand parameters.

Power Demand Coefficient	Engine Type	
	Spark-Ignition	Diesel
p_1	1	0.6526
p_2	1	1.6948
p_3	1	1.3474

Idle Speed Controller Model

The idle speed controller adjusts the throttle signal to increase engine rotation below a reference speed according to the following expressions:

$$\Pi = \max(\Pi_i, \Pi_c)$$

and

$$\frac{d(\Pi_c)}{dt} = \frac{0.5 \cdot \left(1 - \tanh\left(4 \cdot \frac{\omega - \omega_r}{\omega_t}\right)\right) - \Pi_c}{\tau}$$

where:

- Π is the engine throttle.
- Π_i is the input throttle (port **T**).
- Π_c is the controller throttle.
- ω is the engine speed or crankshaft angular velocity.
- ω_r is the idle speed reference.
- ω_t is the controller speed threshold.
- τ is the controller time constant.

The controlled throttle increases with a first-order lag from zero to one when engine speed falls below the reference speed. When the engine speed rises above the reference speed, the controlled throttle decreases from one to zero. When the difference between engine velocity and reference speed is smaller than the controller speed threshold, the *tanh* part of the equation smooths the time derivative of the controlled throttle. The function limits the controlled throttle to the range [0,1]. The engine uses the larger of the input and controlled throttle values. If engine time lag is included, the controller changes the input *before* it computes the lag.

Redline Controller Model

While the idle speed controller determines the minimum throttle value for maintaining engine speed, the redline controller prevents excessive speed based on a maximum throttle input. To determine the maximum throttle value, the redline controller uses the idle speed controller model equation. However, for the redline controller:

- ω_r is the redline speed reference.

- ω_t is the redline speed threshold.
- τ is the redline time constant.

Performance

To increase simulation speed, set **Fuel consumption model** to `No fuel consumption`.

If you select any other option for **Fuel consumption model**, the block must perform a nonlinear computation. The block solves the equation even if the **FC** port, which reports the fuel consumption rate, is not connected to another block.

When the parameter is set to `No fuel consumption`, the block does not calculate fuel consumption, even if the **FC** port is connected to another block.

Assumptions and Limitations

This block contains an engine time lag limitation.

Engines lag in their response to changing speed and throttle. The block optionally supports lag due to a changing throttle only. Time lag simulation increases model fidelity but reduces simulation performance.

Hardware-in-the-Loop Simulation

To improve simulation performance, set the **Dynamics > Time Constant** parameter to `No time constant - Suitable for HIL simulation`.

Ports

Input

T — Normalized engine throttle level, unitless
physical signal

Engine torque demand as a fraction of maximum possible torque.

Output

P — Instantaneous engine power, W
physical signal

Power produced by the engine, in W.

FC — Fuel consumption rate, kg/s
physical signal

Fuel consumed by the engine, in kg/s.

Conserving

B — Engine block
mechanical rotational

Mechanical rotational conserving port associated with the engine block. This is the base port. The engine block is the physical body that contains the piston cylinders.

F — Engine crankshaft
mechanical rotational

Mechanical rotational conserving port associated with the engine crankshaft. This is the follower port of engine. The crankshaft transmits the power generated from the combustion process. Typically, you attach a clutch and transmission at this port.

Parameters

Engine Torque

Model parameterization — Engine modeling parameter selection
Normalized 3rd-order polynomial matched to peak power (default) | Tabulated torque date | Tabulated power data

Engine modeling selection. Choose between these options based on the available data:

- Normalized 3rd-order polynomial matched to peak power — Parametrize the engine with a power curve lookup table defined by power and speed characteristics.
- Tabulated torque data — Engine is parametrized by a speed–torque table that you specify.
- Tabulated power data — Engine is parametrized by a speed–power table that you specify.

Engine type — Combustion process to model
Spark-ignition (default) | Diesel

Internal combustion modeling choice. Either option will cause the block to use the coefficients for that option in the third-order polynomial.

Dependencies

To enable this parameter, set **Model parameterization** to Normalized 3rd-order polynomial matched to peak power.

Maximum power — Physical limit of engine power output
150 kW (default) | positive scalar

Maximum power P_{max} that the engine can output. This value sets the peak of the speed-power curve.

Dependencies

To enable this parameter, set **Model parameterization** to Normalized 3rd-order polynomial matched to peak power.

Speed at maximum power — Crankshaft angular velocity when engine reaches maximum power output
4500 rpm (default) | positive scalar

Engine speed Ω_0 at which the engine is running at maximum power. This value sets the location along the power curve where the peak occurs.

Dependencies

To enable this parameter, set **Model parameterization** to Normalized 3rd-order polynomial matched to peak power.

`Maximum speed` — Maximum crankshaft angular velocity
6000 rpm (default) | positive scalar

Maximum speed Ω_{max} at which the engine can generate torque. This value sets the upper bound of the power curve.

During simulation, if Ω exceeds this value, the simulation stops with an error. The engine maximum speed Ω_{max} cannot exceed the engine speed at which the engine power becomes negative.

Dependencies

To enable this parameter, set **Model parameterization** to `Normalized 3rd-order polynomial matched to peak power`.

`Stall speed` — Engine cutoff threshold
500 rpm (default) | positive scalar

Minimum speed Ω_{min} at which the engine can generate torque. This value sets the lower bound of the power curve.

During simulation, if Ω falls below this value, the engine torque is blended to zero.

Dependencies

To enable this parameter, set **Model parameterization** to `Normalized 3rd-order polynomial matched to peak power`.

`Speed vector` — Tabulated engine speed data
[500, 1000, 2000, 3000, 4000, 5000, 6000, 7000] rpm (default) | vector

Vector of engine speeds that informs the model parameterization lookup table.

Dependencies

To enable this parameter, set **Model parameterization** to `Tabulated torque data` or `Tabulated power data`.

`Torque vector` — Tabulated engine torque data
[380, 380, 380, 380, 350, 280, 200, 80] N*m (default) | vector

Vector of engine torques for lookup table parameterizations. The size of this vector must match the **Speed vector** parameter.

Dependencies

To enable this parameter, set **Model parameterization** to `Tabulated torque data`.

`Power vector` — Tabulated engine power data
[20, 40, 78, 120, 145, 148, 125, 60] kW (default) | vector

Vector of values of the engine function's dependent variable, the power P. The size of this vector must match the **Speed vector** parameter.

Dependencies

To enable this parameter, set **Model parameterization** to `Tabulated power data`.

Interpolation method — Intermediate point calculation method
Linear (default) | Smooth

Method to calculate the fuel consumption at intermediate speed-torque values. Beyond the data range, fuel consumption remains constant at the final lookup table value.

Dependencies

To enable this parameter, set **Model parameterization** to Tabulated torque data or Tabulated power data.

Dynamics

Inertia — Enable inertia modeling parameters
No inertia (default) | Specify inertia and initial velocity

Option to parameterize the inertia and initial velocity. You can choose to model inertia for increased fidelity or numerical stability.

Engine inertia — Engine crankshaft rotational inertia
1 kg*m^2 (default) | positive scalar

Initial rotational inertia. This parameter initializes the engine inertia computation.

Dependencies

To enable this parameter, set **Inertia** to Specify inertia and initial velocity.

Initial velocity — Engine crankshaft angular velocity
800 rpm (default) | positive scalar

Initial angular velocity, $\Omega(0)$. This parameter initializes the engine inertia computation.

Time constant — Enable engine lag modeling parameters
No lag — Suitable for HIL simulation (default) | Specify time constant and initial value

Option to parameterize the time constant and initial normalized engine throttle.

Engine time constant — Engine lag value
0.2 s (default) | positive scalar

Engine response time delay value. This parameter specifies the lag between the throttle signal and the engine response.

Dependencies

To enable this parameter, set **Time constant** to Specify time constant and initial value.

Initial normalized throttle — Engine initial throttle value
0 (default) | nonnegative scalar in the range [0,1]

Initial normalized engine throttle, $T(0)$. This value must be in the range [0,1]. You specify this value when you want to model lag.

Dependencies

To enable this parameter, set **Time constant** to Specify time constant and initial value.

Limits

Speed threshold — Torque loss bandwidth
100 rpm (default) | positive scalar

Torque drop off range before stalling. This parameter specifies where the engine starts to experience rapid torque drop off as the crankshaft approaches the **Stall speed**. The torque is blended to 0 between this point and the point of stalling.

Speed Control

Idle speed control — Enable speed control parameters
Off (default) | On

Option to enable speed control. Speed control idles the engine at a given setpoint. The purpose of engine idle is to prevent stalling when the applied throttle is not enough to turn . When you enable speed control, the block asserts enough throttle to maintain *at least* the **Idle speed reference** parameter value. For more information, see "Idle Speed Controller Model" on page 1-244.

Idle speed reference — Steady-state idle speed
1000 rpm (default) | positive scalar

Reference crankshaft angular velocity for speed control modeling. The idle speed controller will work to maintain at least this speed.

Dependencies

To enable this parameter, set **Idle speed control** to On.

Controller time constant — Idle controller response delay
1 s (default) | positive scalar

Time to respond to deviations from the value of the **Idle speed reference** parameter.

Dependencies

To enable this parameter, set **Idle speed control** to On.

Controller threshold speed — Controller response smoothness
1 rpm (default) | positive scalar

Controller smoothing parameter. The **Controller threshold speed** smooths the controlled throttle value when the rotational speed of the engine crosses the idle speed reference. For more information, see "Idle Speed Controller Model" on page 1-244. Large values decrease controller responsiveness. Small values increase computational cost. This parameter must be positive.

Dependencies

To enable this parameter, set **Idle speed control** to On.

Redline control — Enable redline control parameters
Off (default) | On

Option to enable the engine crankshaft maximum angular velocity limiter. When you set **Redline control** to On, the block works to prevent the engine speed from exceeding a maximum setpoint.

Redline speed — Crankshaft speed control peak threshold
5000 rpm (default) | positive scalar

Redline control engine speed setpoint. Enter the value of the speed reference above which the redline control activates.

Dependencies

To enable this parameter, set **Redline control** to On.

Redline time constant — Redline controller response delay
1 s (default) | positive scalar

Time to respond to engine speeds exceeding the value of the **Redline speed** parameter. Enter the value of the time constant associated with an increase or decrease of the controlled throttle.

Dependencies

To enable this parameter, set **Redline control** to On.

Redline threshold speed — Redline controller response smoothness
1 rpm (default) | positive scalar

Redline controller smoothing parameter. Specify the width of the region around the redline speed where the controller goes from fully enabled to not enabled. The block uses this parameter for smoothing the controlled throttle value when the rotational speed of the engine crosses the redline speed reference. Large values decrease controller responsiveness. Small values increase computational cost.

Dependencies

To enable this parameter, set **Redline control** to On.

Fuel Consumption

The table shows how the specified options for the **Fuel consumption model** parameter affects the availability of dependent parameters. To learn how to read the dependencies tables, see "Parameter Dependencies" on page B-2.

Fuel Consumption Parameter Dependencies

Fuel Consumption				
Fuel consumption model				
No fuel consumption	Constant per revolution	Fuel consumption by speed and torque	Brake specific fuel consumption by speed and torque	Brake specific fuel consumption by speed and brake mean effective pressure
	Fuel consumption per revolution			**Displaced volume**
				Revolutions per cycle
		Speed vector		
		Torque vector		**Brake mean effective pressure vector**
		Fuel consumption table	**Brake specific fuel consumption table**	
		Interpolation method		

Fuel consumption model — Enable fuel consumption modeling parameters
No fuel consumption (default) | Constant per revolution | Fuel consumption by speed and torque | …

Fuel consumption method based on available data. Select a model for calculating fuel consumption. The methods enable parameters that are consistent with typical industrial data. Choose from the following options:

- No fuel consumption
- Constant per revolution
- Fuel consumption by speed and torque
- Brake specific fuel consumption by speed and torque
- Brake specific fuel consumption by speed and brake mean effective pressure

If you set **Fuel consumption model** to No fuel consumption, the block does not calculate fuel consumption even when the **FC** port is connected to another block. Selecting this option increases simulation speed.

Fuel consumption per revolution — Constant
25 mg/rev (default) | positive scalar

Constant rate of fuel consumption as a function of crankshaft revolutions. Enter the volume of fuel consumed in one crankshaft revolution.

Dependencies

To enable this parameter, set **Fuel consumption** to Constant per revolution.

Displaced volume — Fuel volume displaced by a single piston stroke
400 cm^3 (default) | positive scalar

Piston stroke fuel displacement. This is equivalent to the area of the circular piston cylinder cross-section multiplied by the piston travel distance.

Dependencies

To enable this parameter, set **Fuel consumption** to Brake specific fuel consumption by speed and brake mean effective pressure.

Revolutions per cycle — Combustion cycle revolutions
2 (default) | 1

Number of crankshaft revolutions in a combustion cycle. Enter 2 for a four-stroke engine or 1 for a two-stroke engine.

Dependencies

To enable this parameter, set **Fuel consumption** to Brake specific fuel consumption by speed and brake mean effective pressure.

Speed vector — Engine speed data
[1000, 2000, 3000, 4000, 5000, 6000] (default) | vector

Vector of engine torques that corresponds to the M rows of the fuel consumption lookup table.

Dependencies

To enable this parameter, set **Fuel consumption** to one of these settings:

- Fuel consumption by speed and torque
- Brake specific fuel consumption by speed and torque
- Brake specific fuel consumption by speed and brake mean effective pressure

Torque vector — Engine torque data
[0, 80, 160, 200, 240, 320, 360, 400] (default) | vector

Vector of engine torques that corresponds to the N columns of the fuel consumption lookup table.

Dependencies

To enable this parameter, set **Fuel consumption** to Fuel consumption by speed and torque or Brake specific fuel consumption by speed and torque.

Fuel consumption table — Engine fuel consumption data
M-by-N matrix (default) | matrix

Matrix of fuel consumption values that correspond. Enter matrix with fuel consumption rates corresponding to engine speed and torque vectors. The number of rows must equal the number of elements in the **Speed vector** parameter. The number of columns must equal the number of elements in the **Torque vector** parameter. The default is [.5, .9, 1.4, 1.6, 1.9, 2.7, 3.4, 4.4; 1, 1.7, 2.7, 3.1, 3.6, 5, 6, 7.4; 1.4, 2.7, 4, 4.8, 5.6, 7.5, 8.5, 10.5; 2, 3.6, 5.8, 6.7, 8, 10.4, 11.7, 13.3; 2.5, 4.8, 7.9, 9.4, 10.8, 14, 16.2, 18.6; 3.1, 6, 10.3, 11.9, 13.8, 18.4, 22, 26.5] g/s.

Dependencies

To enable this parameter, set **Fuel consumption** to `Fuel consumption by speed and torque`.

Brake mean effective pressure vector — Piston pressure data
[0, 250, 500, 625, 750, 1000, 1150, 1250] (default) | vector

Vector of brake mean effective pressure (BMEP) values. The BMEP satisfies the expression:

$$BMEP = T \cdot \left(\frac{2\pi \cdot n_c}{V_d} \right),$$

where:

- T is the output torque.
- n_c is the number of cycles per revolution.
- V_d is the cylinder displaced volume.

Dependencies

To enable this parameter, set **Fuel consumption** to `Brake specific fuel consumption by speed and brake mean effective pressure.`

Brake specific fuel consumption table — Brake specific fuel consumption data
M-by-*N* matrix (default) | matrix

For the `Brake specific fuel consumption by speed and torque` fuel model, enter the matrix with brake specific fuel consumption (BSFC) rates corresponding to engine speed and torque vectors. BSFC is the ratio of the fuel consumption rate to the output power.

For the `Brake specific fuel consumption by speed and brake mean effective pressure` fuel model, enter the matrix with brake specific fuel consumption (BSFC) rates corresponding to engine speed and brake mean effective pressure (BMEP) vectors. BSFC is the ratio of the fuel consumption rate to the output power.

For both fuel-consumption models, the number of rows must equal the number of elements in the **Speed vector** parameter. The number of columns must equal the number of elements in the **Torque vector** parameter or **Brake mean effective pressure vector** parameter. The default is [410, 380, 300, 280, 270, 290, 320, 380; 410, 370, 290, 270, 260, 270, 285, 320; 415, 380, 290, 275, 265, 270, 270, 300; 420, 390, 310, 290, 285, 280, 280, 285; 430, 410, 340, 320, 310, 300, 310, 320; 450, 430, 370, 340, 330, 330, 350, 380] g/hr/kW.

Dependencies

To enable this parameter, set **Fuel consumption** to `Brake specific fuel consumption by speed and torque` or `Brake specific fuel consumption by speed and brake mean effective pressure.`

Interpolation method — Intermediate step behavior
Linear (default) | Smooth

Interpolation method to calculate fuel consumption at intermediate speed and torque values. Fuel consumption remains constant beyond the range of the lookup table.

Dependencies

To enable this parameter, set **Fuel consumption** to one of these settings:

- `Fuel consumption by speed and torque`
- `Brake specific fuel consumption by speed and torque`
- `Brake specific fuel consumption by speed and brake mean effective pressure`

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Piston Engine | Piston

Topics
"Complete Vehicle Model"
"Troubleshoot Engine Issues"
"Adjust Model Fidelity"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

Harmonic Drive

High-ratio speed reducer based on elastic deformation of an elliptical gear
**Library:**   Simscape / Driveline / Gears

Description

This block represents a compact, high-ratio, speed reduction mechanism that contains three key components:

- Strain wave generator
- Elliptical gear
- Circular ring gear

The strain wave generator comprises an elliptical plug mated to a raced ball bearing. It sits inside an elastic metal gear, deforming it into a slight elliptical pattern. Rotation of the elliptical pattern in the body of the gear constitutes a strain wave.

The elliptically deformed gear engages the internal teeth of a fixed circular ring gear of only slightly larger diameter. Meshing occurs concurrently at the two elongated ends of the elliptical gear. This design doubles the teeth in mesh, boosting the torque capacity of the drive system.
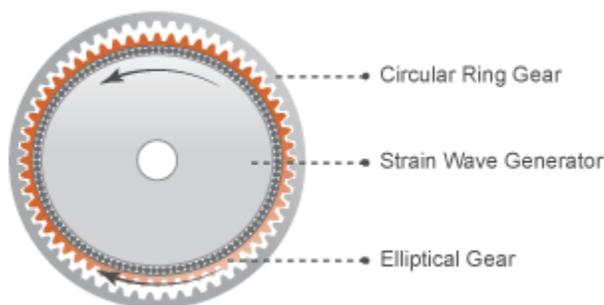
During normal operation, the base shaft drives the strain wave generator. The elliptical plug spins freely inside the elastic metal gear, propagating the strain wave about the gear rotation axis. This strain wave causes the elliptical gear teeth to engage the internal teeth of the circular ring gear progressively.

The internal meshing between the two gears causes the elliptical gear axis to spin counter to the elliptical strain wave. For every clockwise rotation that the strain wave generator completes, the elliptical gear axis rotates counterclockwise by a small amount.

Large reduction ratios arise from the near-equal gear tooth numbers. The effective gear reduction ratio is:

$$r = \frac{n_E}{n_C - n_E},$$

where:

- r is the gear reduction ratio.
- n_C is the tooth number of the circular ring gear.
- n_E is the tooth number of the deformable elliptical gear.

Optional parameters account for power losses due to gear meshing and viscous friction. The Simple Gear block provides the foundation for this block. For more information, see Simple Gear.

Thermal Model

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Friction model** to `Temperature-dependent efficiency`.

Variables

Use the **Variables** settings to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Hardware-in-the-Loop Simulation

For optimal performance of your real-time simulation, set the **Friction model** to `No meshing losses - Suitable for HIL simulation` on the **Meshing Losses** tab.

Ports

Conserving

B — Base shaft
mechanical rotational

Conserving rotational port representing the base shaft torque and angular velocity.

F — Follower shaft
mechanical rotational

Conserving rotational port representing the follower shaft torque and angular velocity.

H — Optional heat transfer port
thermal

Thermal conserving port for thermal modeling.

Dependencies

To enable this port, set **Friction model** to either:

- `Temperature-dependent efficiency`
- `Temperature and load-dependent efficiency`

.

Parameters

Main

Number of teeth on elliptical gear — Elliptical gear teeth
100 (default) | positive scalar

Total number of teeth protruding outward from the elliptical gear perimeter. This number should be slightly smaller than the number of teeth on the circular ring gear. The ratio of the two gear tooth numbers defines the relative angular velocities of the base and follower shafts.

Number of teeth on circular gear — Circular gear teeth
102 (default) | positive scalar

Number of teeth protruding inward from the circular ring gear perimeter. This number should be slightly larger than the number of teeth on the elliptical gear. The ratio of the two gear tooth numbers defines the relative angular velocities of the base and follower shafts.

Meshing Losses

Friction model — Gear friction model
No meshing losses — Suitable for HIL simulation (default) | Constant efficiency | Load-dependent efficiency | Temperature-dependent efficiency | Temperature and load-dependent efficiency

- No meshing losses — Suitable for HIL simulation — Gear meshing is ideal.
- Constant efficiency — Transfer of torque between worm and gear is reduced by friction.
- Load-dependent efficiency — Reduce torque transfer by a variable efficiency factor. This factor falls in the range $0 < \eta < 1$ and varies with the torque load.
- Temperature-dependent efficiency — Reduce torque transfer by a constant efficiency factor that is dependent on temperature but does not consider the gear load. This factor falls in the range $0 < \eta \leq 1$ and is independent from load. Torque transfer is determined from user-supplied data for gear efficiency and temperature.
- Temperature and load-dependent efficiency — Reduce torque transfer by a variable efficiency factor that is dependent on temperature and load. This factor falls in the range $0 < \eta < 1$ and varies with the torque load. Torque transfer efficiency is determined from user-supplied data for gear loading and temperature.

Efficiency — Torque transfer efficiency
0.95 (default) | positive scalar in the range of (0, 1]

Torque transfer efficiency (η) between base and follower shafts. This parameter is inversely proportional to the meshing power losses.

Dependencies

To enable this parameter, set **Friction model** to Constant efficiency.

Follower power threshold — Value above which the full efficiency factor is applied
0.001 W (default) | positive scalar

Absolute value of the follower shaft power above which the full efficiency factor is in effect. A hyperbolic tangent function smooths the efficiency factor from zero when at rest to the full efficiency value at the power threshold.

As a guideline, the power threshold should be lower than the expected power transmitted during simulation. Higher values might cause the block to underestimate efficiency losses. However, very low values may raise the computational cost of simulation.

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency`.

Input shaft torque at no load — Net torque on idle shaft
`0.1` N*m (default) | positive scalar

Net torque (τ_{idle}) acting on the input shaft in idle mode, e.g., when torque transfer to the output shaft equals zero. For nonzero values, the power input in idle mode completely dissipates due to meshing losses.

Dependencies

To enable this parameter, set **Friction model** to `Load-dependent efficiency`.

Nominal output torque — Value of torque at which efficiency is normalized
`5` N*m (default) | positive scalar

Output torque (τ_F) at which to normalize the load-dependent efficiency.

Dependencies

To enable this parameter, set **Friction model** to `Load-dependent efficiency`.

Efficiency at nominal output torque — Transfer efficiency
`0.95` (default) | positive scalar in the range of (0,1]

Torque transfer efficiency (η) at the nominal output torque. Larger efficiency values correspond to greater torque transfer between the input and output shafts.

Dependencies

To enable this parameter, set **Friction model** to `Load-dependent efficiency`.

Follower angular velocity threshold — Value above which the full efficiency factor is applied
`0.01` rad/s (default) | positive scalar

Absolute value of the follower shaft angular velocity above which the full efficiency factor is in effect (ω_F). Below this value, a hyperbolic tangent function smooths the efficiency factor to one, lowering the efficiency losses to zero when at rest.

As a guideline, the angular velocity threshold should be lower than the expected angular velocity during simulation. Higher values might cause the block to underestimate efficiency losses. However, very low values may raise the computational cost of simulation.

Dependencies

To enable this parameter, set **Friction model** to `Load-dependent efficiency`.

Temperature — Array for tabular parameterization of efficiency
`[280 300 320]` K (default) | vector

Array of temperatures used to construct an efficiency lookup table. The array values must increase from left to right. The temperature array must be the same size as the efficiency array in temperature-dependent models. The array must be the same size as the **Efficiency** array.

Dependencies

To enable this parameter, set **Friction model** to either:

- `Temperature-dependent efficiency`
- `Temperature and load-dependent efficiency`

Efficiency — Array of gear efficiencies
[0.95 0.9 0.85] (default) | vector

Array of efficiencies used to construct a 1-D temperature-efficiency lookup table for temperature-dependent efficiency models. The array elements are the efficiencies at the temperatures in the **Temperature** array. The two arrays must be the same size.

Dependencies

To enable this parameter, set **Friction model** to either:

- `Temperature-dependent efficiency`
- `Temperature and load-dependent efficiency`

Follower power threshold — Power below which numerical smoothing is applied
0.001 W (default) | positive scalar

Absolute value of the follower shaft power above which the full efficiency factor is in effect. A hyperbolic tangent function smooths the efficiency factor between zero when at rest and the value provided by the temperature-efficiency lookup table when at the power threshold.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Load at elliptical gear — Elliptical-gear loads for tabular parameterization of efficiency
[1 5 10] N*m (default) | numerical array

Array of elliptical-gear loads used to construct a 2-D temperature-load-efficiency lookup table for temperature-and-load-dependent efficiency models. The array values must increase from left to right. The load array must be the same size as a single column of the efficiency matrix.

Dependencies

To enable this parameter, set **Friction model** to `Temperature and load-dependent efficiency`.

Efficiency matrix — Efficiency array for tabular parameterization of efficiency
[0.85 0.8 0.75; 0.95 0.9 0.85; 0.85 0.8 0.7] (default) | numerical array

Matrix of component efficiencies used to construct a 2-D temperature-load-efficiency lookup table. The matrix elements are the efficiencies at the temperatures in the **Temperature** array and at the loads in the **Load at elliptical gear** array.

The number of rows must be the same as the number of elements in the **Temperature** array. The number of columns must be the same as the number of elements in the **Load at elliptical gear** array.

Dependencies

To enable this parameter, set **Friction model** to `Temperature and load-dependent efficiency`.

Follower angular velocity threshold — Angular velocity below which numerical smoothing is applied
`0.01` rad/s (default) | positive scalar

Absolute value of the follower shaft angular velocity above which the full efficiency factor is in effect. Below this value, a hyperbolic tangent function smooths the efficiency factor to one, lowering the efficiency losses to zero when at rest.

Dependencies

To enable this parameter, set **Friction model** to `Temperature and load-dependent efficiency`.

Viscous Losses

Viscous friction coefficients at base (B) and follower (F) — Fluid dynamic friction coefficients
`[0 0] N*m/(rad/s)` (default) | positive two-element vector

Two-element array with the viscous friction coefficients in effect at the base and follower shafts. The default array corresponds to zero viscous losses.

Thermal Port

Thermal mass — Thermal mass
`50` J/K (default) | positive scalar

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change.

Initial temperature — Initial temperature
`300` K (default) | positive scalar

Component temperature at the start of simulation. The initial temperature alters the component efficiency according to an efficiency vector that you specify, affecting the starting meshing or friction losses.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Cycloidal Drive | Simple Gear | Simple Gear with Variable Efficiency

Topics
"Model Thermal Losses in Driveline Components"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2014a

Leadscrew

Leadscrew gear set of threaded rotating screw and translating nut, with adjustable thread and friction losses

Library: Simscape / Driveline / Gears / Rotational- Translational

Description

The Leadscrew block represents a threaded rotational-translational gear that constrains the two connected driveline axes, screw (S) and nut (N), to, respectively, rotate and translate together in a fixed ratio that you specify. You can choose whether the nut axis translates in a positive or negative direction, as the screw rotates in a positive right-hand direction. If the screw helix is right-hand, ω_S and v_N have the same sign. If the screw helix is left-hand, ω_S and v_N have opposite signs.

Ideal Gear Constraint and Gear Ratio

Leadscrew imposes one kinematic constraint on the two connected axes:

$$\omega_S L = 2\pi v_N . \tag{1-1}$$

The transmission ratio is $R_{NS} = 2\pi/L$. L is the screw lead, the translational displacement of the nut for one turn of the screw. In terms of this ratio, the kinematic constraint is:

$$\omega_S = R_{NS} v_N . \tag{1-2}$$

The two degrees of freedom are reduced to one independent degree of freedom. The forward-transfer gear pair convention is (1,2) = (S,N).

The torque-force transfer is:

$$R_{NS}\tau_S + F_N - F_{loss} = 0 , \tag{1-3}$$

with $F_{loss} = 0$ in the ideal case.

Nonideal Gear Constraint and Losses

In the nonideal case, $F_{loss} \neq 0$. For general considerations on nonideal gear modeling, see "Model Gears with Losses".

Geometric Surface Contact Friction

In the contact friction case, η_{SN} and η_{NS} are determined by:

- The screw-nut threading geometry, specified by lead angle λ and acme thread half-angle α.
- The surface contact friction coefficient k.

$$\eta_{SN} = (\cos\alpha - k\cdot\tan\alpha)/(\cos\alpha + k/\tan\lambda) , \tag{1-4}$$

$$\eta_{NS} = (\cos\alpha - k/\tan\lambda)/(\cos\alpha + k\cdot\tan\alpha)\,. \tag{1-5}$$

Constant Efficiencies

In the constant efficiency case, you specify η_{SN} and η_{NS}, independently of geometric details.

Self-Locking and Negative Efficiency

η_{NS} has two distinct regimes, depending on lead angle λ, separated by the *self-locking point* at which $\eta_{NS} = 0$ and $\cos\alpha = k/\tan\lambda$.

- In the *overhauling regime*, $\eta_{NS} > 0$. The force acting on the nut can rotate the screw.
- In the *self-locking regime*, $\eta_{NS} < 0$. An external torque must be applied to the screw to release an otherwise locked mechanism. The more negative is η_{NS}, the larger the torque must be to release the mechanism.

η_{SN} is conventionally positive.

Meshing Efficiency

The efficiencies η of meshing between screw and nut are fully active only if the transmitted power is greater than the power threshold.

If the power is less than the threshold, the actual efficiency is automatically regularized to unity at zero velocity.

Viscous Friction Force

The viscous friction coefficient μ controls the viscous friction torque experienced by the screw from lubricated, nonideal gear threads. The viscous friction torque on a screw driveline axis is $-\mu_S\omega_S$. ω_S is the angular velocity of the screw with respect to its mounting.

Thermal Modeling

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Friction model** to `Temperature-dependent efficiency`.

Hardware-in-the-Loop Simulation

For optimal performance of your real-time simulation, set the **Friction model** to `No meshing losses - Suitable for HIL simulation` on the **Meshing Losses** tab.

Variables

Use the **Variables** settings to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Limitations

- Gear inertia is assumed to be negligible.
- Gears are treated as rigid components.
- Coulomb friction slows down simulation. For more information, see "Adjust Model Fidelity".

Ports

Port	Description
S	Rotational conserving port representing the screw
N	Translational conserving port representing the nut
H	Thermal conserving port for thermal modeling

Parameters

Main

Screw lead — Nut displacement per revolution
0.015 m (default) | positive scalar

Translational displacement *L* of the nut per revolution of the screw.

Screw helix type — Rotational directional sense
Right-hand (default) | Left-hand

Choose the directional sense of screw rotation corresponding to positive nut translation. For the Right-hand orientation, the screw angular velocity and the nut velocity have the same sign.

Meshing Losses

Friction model — Screw friction model
No meshing losses — Suitable for HIL simulation (default) | Constant efficiency | Temperature-dependent efficiency

- No meshing losses — Suitable for HIL simulation — Screw meshing is ideal.
- Constant efficiency — Transfer of torque between screw and nut is reduced by friction.
- Temperature-dependent efficiency — Torque transfer is determined from user-supplied data for screw-nut efficiency, nut-screw efficiency, and temperature.

Friction parametrization — Friction losses from nonideal meshing of gear threads
Friction coefficient and geometrical parameters (default) | Efficiencies

- Friction coefficient and geometrical parameters — Friction is determined by contact friction between surfaces.
- Efficiencies — Friction is determined by constant efficiencies $0 < \eta < 1$.

Dependencies

To enable this parameter, set **Friction model** to Constant efficiency.

Lead angle — Thread helix angle
15 deg (default) | positive scalar

Thread helix angle $\lambda = \arctan[L/(\pi d)]$, where:

- *L* is the worm lead.
- *d* is the worm pitch diameter.

The value must be greater than zero.

Dependencies

To enable this parameter, set **Friction model** to Constant efficiency and **Friction parameterization** to Friction coefficient and geometrical parameters.

Acme thread half angle — Thread half-angle
14.5 deg (default) | positive scalar

Half-angle of the acme thread α in the normal plane. In the case of a square thread, $\alpha = 0$. The value must be greater than zero.

Dependencies

To enable this parameter, set **Friction model** to Constant efficiency and **Friction parameterization** to Friction coefficient and geometrical parameters.

Friction coefficient — Thread friction coefficient
0.08 (default) | positive scalar

Dimensionless coefficient of normal friction in the thread. The value must be greater than zero.

Dependencies

To enable this parameter, set **Friction model** to Constant efficiency and **Friction parameterization** to Friction coefficient and geometrical parameters.

Screw-nut efficiency — Efficiency from screw to nut
0.75 (default) | positive scalar in the range of (0,1]

Efficiency η_{SN} of the power transfer from screw to nut.

Dependencies

To enable this parameter, set **Friction model** to Constant efficiency and **Friction parameterization** to Efficiencies.

Nut-screw efficiency — Efficiency from nut to screw
0.45 (default) | positive scalar in the range of (0,1]

Efficiency η_{NS} of the power transfer from gear to worm.

Dependencies

To enable this parameter, set **Friction model** to Constant efficiency and **Friction parameterization** to Efficiencies.

Temperature — Array for tabular parameterization of efficiency
[280 300 320] K (default) | vector

Array of temperatures used to construct a 1-D temperature-efficiency lookup table. The array values must increase left to right. The temperature array must be the same size as the **Screw-nut efficiency** and **Nut-screw efficiency** arrays.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

`Screw-nut efficiency` — **Array of screw-to-nut efficiencies**
`[0.75 0.7 0.65]` (default) | vector

Array of component efficiencies with the screw as the driver— that is, with power from the screw to the nut. The array values are the efficiencies at the temperatures in the **Temperature** array. The two arrays must be the same size.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

`Nut-screw efficiency` — **Array of nut-to-screw efficiencies**
`[0.5 0.45 0.4]` (default) | vector

Array of component efficiencies with the nut as the driver— that is, with power flowing from the nut to the screw. The array values are the efficiencies at the temperatures in the **Temperature** array. The two arrays must be the same size.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

`Power threshold` — **Power below which numerical smoothing is applied**
`0.001` W (default) | positive scalar

Power threshold above which full efficiency factor is in effect. A hyperbolic tangent function smooths the efficiency factor between zero at rest and the current efficiency set point.

Viscous Losses

`Viscous friction coefficient` — **Fluid dynamic friction coefficient**
`0`N*m/(rad/s) (default) | positive scalar

Viscous friction coefficient μ_S for the screw.

Thermal Port

`Thermal mass` — **Thermal mass**
`50` J/K (default) | positive scalar

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change.

`Initial temperature` — **Initial temperature**
`300` K (default) | positive scalar

Component temperature at the start of simulation. The initial temperature alters the component efficiency according to an efficiency vector that you specify, affecting the starting meshing or friction losses.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Rack & Pinion | Worm Gear

Topics
"Basic Motion, Torque, and Force Modeling"
"Ratchet Leadscrew Mechanism"
"Model Thermal Losses in Driveline Components"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

Limited-Slip Differential

Reduce velocity difference between two connected shafts
Library: Simscape / Driveline / Gears

Description

The Limited-Slip Differential block represents a limited-slip differential (LSD), which is a gear assembly that limits the velocity difference between two connected shafts. The block models the LSD mechanism as a structural component that combines a differential and a clutch.

The differential component in the LSD block is an open differential. An *open differential* is a gear mechanism that allows two driven shafts to spin at different speeds. In an automobile, a differential allows the inner wheels to spin more slowly than the outer wheels when the vehicle is cornering. A vehicle that has wheel shafts connected by an open differential can get stuck when one of the wheels slips and spins freely due to traction loss. This vehicle stops moving because the driveshaft supplies less power to the wheel with traction than it supplies to the spinning wheel.

In the same scenario, a vehicle that has an LSD is less likely to get stuck because it contains a clutch assembly that can transmit power to the wheel that retains traction. The clutch component in the LSD block is a friction clutch that has two sets of flat friction plates. The clutch engages when applied pressure exceeds the engagement threshold pressure. In an LSD, a spring preload that separates the sun gears presses the plates in both sets together. When the shafts experience a traction differential, the planet pinion gears exert an additional force in the direction of the high-traction shaft. If the additional pressure exceeds the engagement threshold, the clutch assembly engages. The engagement allows the driveshaft to transmit more power to the slower-spinning high-traction wheel. The additional power reduces the difference in velocity of the two shafts. Because the high-traction wheel continues to rotate, the vehicle continues to move.

The figure shows the orientation of the major components in an LSD mechanism. The driveshaft pinion gear is not visible is this representation.
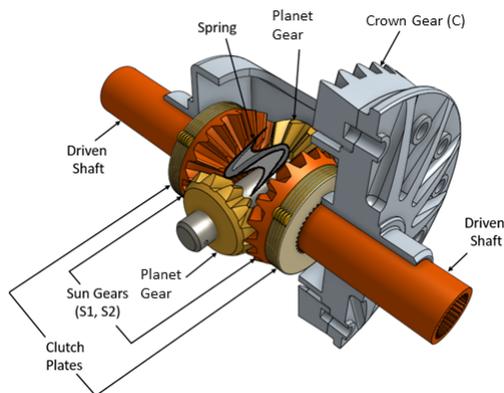
The Limited-Slip Differential block models the LSD mechanism as a structural component based on the Simscape Driveline Differential and Disk Friction Clutch blocks. The differential mechanism modeled by the Differential block is a structural component based on two other Simscape Driveline blocks, the Simple Gear and the Sun-Planet Bevel. The block diagram shows the structural components of the LSD.
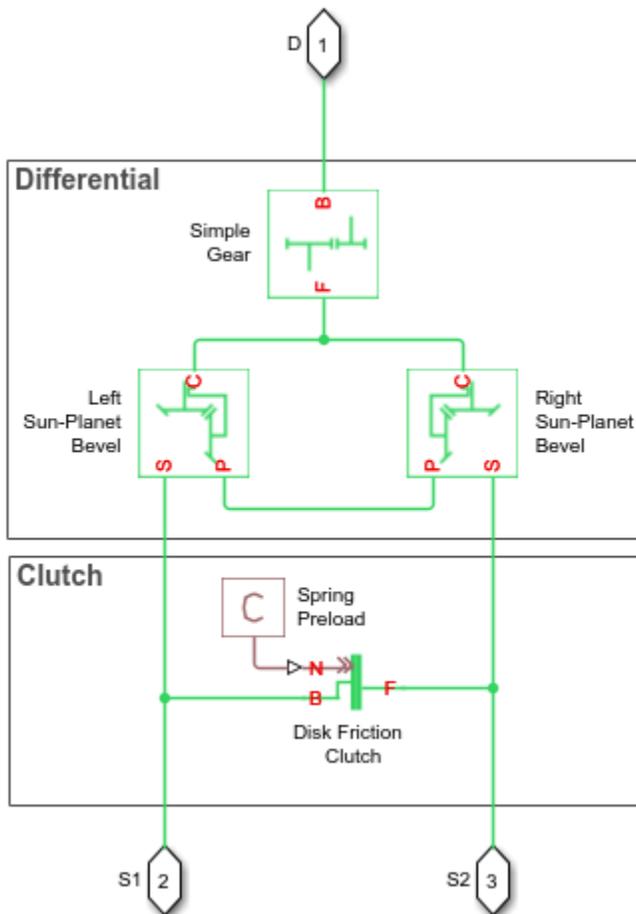
The ports of the Limited-Slip Differential block are associated with the driveshaft (port **D**) and the two driven shafts (ports **S1** and **S2**), which connect the sun gears to the wheels.

The Limited-Slip Differential block enables you to specify inertias only for the gear carrier and internal planet gears. By default, the inertias of the outer gears are assumed negligible. To model the inertias of the outer gears, connect Simscape Inertia blocks to the **D**, **S1**, and **S2** ports.

The table shows the rotation direction of the driven shaft ports for different block parameterizations and input conditions.

Rotation Direction of the Driven Shaft Ports (S1 and S2)	Crown Gear Location Relative to the Centerline	Rotation Direction of Driveshaft Port D	Relative Slippage Across the Differential
Positive	Right	Positive	0
• Positive for the nonslipping port • Negative the slipping port	Right	Positive	> 0
Negative	Right	Negative	0
• Negative for the nonslipping port • Positive the slipping port	Right	Negative	> 0
Negative	Left	Positive	0
• Negative for the nonslipping port • Positive the slipping port	Left	Positive	> 0
Positive	Left	Negative	0
• Positive for the nonslipping port • Negative the slipping port	Left	Negative	> 0

Model

To examine the mathematical models for the structural components of the Limited-Slip Differential block, see:

- Disk Friction Clutch
- Differential
- Simple Gear
- Sun-Planet Bevel

Thermal Model

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Friction model** to Temperature-dependent efficiency.

Ports

Conserving

D — Driving shaft
rotational mechanical

Rotational mechanical conserving port associated with the driveshaft.

S1 — Sun gear 1
rotational mechanical

Rotational mechanical conserving port associated with the sun gear 1 shaft.

S2 — Sun gear 2
rotational mechanical

Rotational mechanical conserving port associated with the sun gear 2 shaft.

H — Heat flow
thermal

Thermal conserving port associated with heat flow.

Dependencies

To enable this port, in the **Differential** tab, set **Friction model** to `Temperature-dependent efficiency`.

Parameters

Differential

`Crown gear located — Crown gear location`
`To the right of centerline` (default) | `To the left of centerline`

Location of the bevel crown gear relative to the centerline of the gear assembly.

`Carrier (C) to driveshaft (D) teeth ratio (NC/ND) —` **Crown gear to driveshaft pinion gear teeth ratio**
`4` (default) | positive scalar

Fixed ratio, g_D, of the number of crown gear teeth NC to the number of driveshaft pinion gear teeth ND. This gear ratio must be strictly greater than 0. The crown gear is rigidly mounted to the carrier.

`Friction model — Friction model`
`No meshing losses - Suitable for HIL simulation` (default) | `Constant efficiency` | `Temperature-dependent efficiency`

Friction model for the block:

- `No meshing losses - Suitable for HIL simulation`— Gear meshing is ideal.
- `Constant efficiency`— Transfer of torque between the gear wheel pairs is reduced by a constant efficiency, η, such that $0 < \eta \leq 1$.
- `Temperature-dependent efficiency`— Transfer of torque between the gear wheel pairs is defined by the temperature lookup table

`Sun-sun and carrier-driveshaft ordinary efficiencies —` **Differential torque constant transfer efficiencies**
`[.85, .92]` (default) | vector

Vector of torque transfer efficiencies, $[\eta_{SS}, \eta_{CD}]$. Here,

- η_{SS} is the output-to-input power ratio that describes the power flow from the driving sun gear to the driven sun gear.
- η_{CD} is the output-to-input power ratio that describes the power flow from the crown gear to the driveshaft pinion gear.

The carrier is rigidly mounted to the crown gear. The vector elements must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency`.

Temperature — Temperature
[280, 300, 320] K (default) | vector

Vector of temperatures used to construct a 1-D temperature-efficiency lookup table. The vector elements must increase from left to right.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Sun-sun efficiency — Torque transfer efficiency from the driving sun gear to the driven sun gear
[.95, .9, .85] (default) | vector

Vector of output-to-input power ratios that describe the power flow from the driving sun gear to the driven sun gear, η_{SS}. The block uses the values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Carrier-driveshaft efficiency — Torque transfer efficiency from the crown gear to the driveshaft pinon gear
[.95, .9, .85] (default) | vector

Vector of output-to-input power ratios that describe the power flow from the crown gear to the driveshaft pinion gear, η_{CD}. The block uses the values to construct a 1-D temperature-efficiency lookup table. The carrier gear is rigidly mounted to the crown gear.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Sun-carrier and driveshaft-casing power thresholds — Minimum efficiency power thresholds for the sun gear to the crown gear and driveshaft-casing gear couplings
[.001, .001] W (default) | vector

Vector of power thresholds, p_{th}, for sun-carrier and longitudinal driveshaft-casing [p_S, p_D], respectively. The full efficiency loss applies above these values. Below these values, a hyperbolic tangent function smooths the efficiency factor.

When you set **Friction model** to `Constant efficiency`, the block lowers the efficiency losses to zero when no power is transmitted. When you set **Friction model** to `Temperature-dependent efficiency`, the block smooths the efficiency factors between zero when at rest and the values provided by the temperature-efficiency lookup tables at the power thresholds.

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency` or `Temperature-dependent efficiency`.

`Sun-carrier and driveshaft-casing viscous friction coefficients` — Viscous friction coefficients
[0, 0] N*m/(rad/s) (default) | vector

Vector of viscous friction coefficients $[\mu_S, \mu_D]$ for the sun-carrier and longitudinal driveshaft-casing gear motions, respectively.

`Inertia` — Inertia model
off (default) | on

Inertia model for the block:

- `Off` — Model gear inertia.
- `On` — Neglect gear inertia.

`Carrier inertia` — Planet gear carrier moment of inertia
0.001 kg*m^2 (default) | positive scalar

Moment of inertia of the planet gear carrier assembly including the crown gear. This value must be positive.

Dependencies

To enable this parameter, set **Inertia** to `On`.

`Planet gear inertia` — Planet gear moment of inertia
0.001 kg*m^2 (default) | positive scalar

Moment of inertia of the combined planet gears. This value must be positive.

Dependencies

To enable this parameter, set **Inertia** to `On`.

Clutch

`Number of friction surfaces` — Number of contact surfaces
4 (default) | positive scalar

Number, N, of friction-generating contact surfaces inside the clutch.

`Effective torque radius` — Friction force application radius
130 mm (default) | positive scalar

Effective moment arm radius, r_{eff}, that determines the kinetic friction torque inside the clutch.

`Preload force` — Clutch plate engagement pressure
300 N (default) | nonnegative scalar

Force that the spring preload exerts on the clutch plate assemblies. Must be greater than or equal to zero.

Static friction coefficient — Static friction coefficient
0.35 (default) | positive scalar

Static or peak value of the friction coefficient. The static friction coefficient must be greater than the kinetic friction coefficient.

Dependencies

To enable this parameter, set **Friction model** to No meshing losses - Suitable for HIL simulation or Fixed kinetic friction coefficient.

Static friction coefficient vector — Static friction coefficient
[.4, .38, .36] (default) | vector

Vector of static or peak values of the friction coefficient for a given temperature. The vector must be the same length as the **Temperature**. Each element must be greater than the maximum value of the corresponding row in the **Kinetic friction coefficient matrix**.

Dependencies

To enable this parameter, set **Friction model** to Temperature-dependent efficiency.

Kinetic friction coefficient relative velocity vector — Absolute surface velocity
[0, 100, 1000] rad/s (default) | vector

Vector of input values for the relative velocity. The values in the vector must increase from left to right. The minimum number of values depends on the interpolation method that you select. For linear interpolation, provide at least two values per dimension. For smooth interpolation, provide at least three values per dimension.

Kinetic friction coefficient vector — Coulomb coefficient of friction, k_K
[.3, .22, .19] (default) | vector

Vector of output values for the kinetic friction coefficient. All values must be greater than zero.

Dependencies

To enable this parameter, in the **Differential** tab, set **Friction model** to No meshing losses - Suitable for HIL simulation or Constant efficiency.

Kinetic friction coefficient matrix — Coulomb coefficient of friction, k_K
[.34, .32, .3; .3, .28, .25; .25, .2, .15] (default) | matrix

Matrix of output values for the kinetic friction coefficient. All values must be greater than zero.

Dependencies

To enable this parameter, in the **Differential** tab, set **Friction model** to Temperature-dependent efficiency.

Friction coefficient interpolation method — Output value filling method
Linear (default) | Smooth

Interpolation methods for approximating the output value when the input value is between two consecutive grid points. To optimize performance, select Linear. To produce a continuous curve with continuous first-order derivatives, select Smooth.

For more information on interpolation algorithms, see PS Lookup Table (1D).

Friction coefficient extrapolation method — Output value extending method
Linear (default) | Nearest

Extrapolation methods for approximating the output value when the input value is outside the range specified in the argument list. To produce a curve with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region, select Linear. To produce an extrapolation that does not go above the highest point in the data or below the lowest point in the data, select Nearest.

For more information on interpolation algorithms, see PS Lookup Table (1D).

Velocity tolerance — Slip velocity locking boundary
0.001 rad/s (default)

Maximum slip velocity at which the clutch can lock. The slip velocity is the signed difference between the base and follower shaft angular velocities, $w = w_F - w_B$. When the kinetic friction torque is nonzero and the transferred torque is within the static friction torque limits, the clutch locks if the actual slip velocity falls below the velocity tolerance.

Initial state — Initial clutch state
Unlocked (default) | Locked

Clutch state at the start of simulation. The clutch can be in one of two states, locked and unlocked. A locked clutch constrains the base and follower shafts to spin at the same velocity, that is, as a single unit. An unlocked clutch allows the two shafts to spin at different velocities, resulting in slip between the clutch plates.

Thermal Port

To enable these settings, set **Friction model** to Temperature-dependent efficiency.

Thermal mass — Thermal mass
50 J/K (default) | scalar

Thermal energy required to change the component temperature by a single temperature unit. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

To enable this parameter, set**Friction model** to Temperature-dependent efficiency.

Initial temperature — Initial temperature
300 K (default) | scalar

Block temperature at the start of simulation. The initial temperature sets the initial component efficiencies according to their respective efficiency vectors.

Dependencies

To enable this parameter, set **Friction model** to Temperature-dependent efficiency.

More About

Hardware-in-the-Loop Simulation

For optimal simulation performance, set **Friction model** to the default value, `No meshing losses - Suitable for HIL simulation`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Differential | Disk Friction Clutch | Simple Gear | Sun-Planet Bevel

Topics
"Model Thermal Losses in Driveline Components"
"Basic Motion, Torque, and Force Modeling"

Introduced in R2017a

Loaded-Contact Rotational Friction

Loaded-contact friction between two rotating surfaces
Library: Simscape / Driveline / Brakes & Detents / Rotational

Description

The Loaded-Contact Rotational Friction block simulates friction between two rotating surfaces loaded with a normal force.

The block is implemented as a structural component based on the Fundamental Friction Clutch block. From the locked state, the two surfaces unlock if the transmitted torque exceeds the static friction, as defined by the static coefficient of friction and current normal force. For details on how the locking and unlocking are modeled, see the Fundamental Friction Clutch block reference page.

Torque is transmitted for normal forces larger than the **Threshold force** parameter.

You can also enable faulting. When faulting occurs, the surfaces will remain locked or will be unable to transmit power. Faults can occur at a specified time or due to an external trigger at port **T**.

Equations

The block simulates friction between two rotating surfaces loaded with a normal force. When the two rotating surfaces are not locked, the transmitted torque is determined with the following equations:

$$\tau \;=\; N \cdot \; \mu \cdot \; r_{\text{eff}} \; \cdot \text{sign}(\omega) \;+\; \tau_{\text{visc}} \;,$$

$$r_{\text{eff}} = \; \frac{2}{3} \cdot \frac{r_o^3 - r_i^3}{r_o^2 - r_i^2} \;,$$

$$\tau_{visc} \;=\; \mu_{\text{visc}} \cdot \; \omega \;,$$

where:

- τ is the transmitted torque.
- N is the normal force.
- μ is the friction coefficient.
- r_{eff} is the effective radius.
- r_o is the surface outside radius.
- r_i is the surface inside radius.
- ω is the relative angular velocity.
- τ_{visc} is the viscous drag torque.
- μ_{visc} is the viscous drag torque coefficient.

Velocity-Dependent Model

You can model the effects of rotational velocity change by selecting a velocity-dependent model. To choose a velocity-dependent model, in the **Friction** settings, set the **Friction model** parameter to `Velocity-dependent kinetic friction coefficient`. For information about a friction model that depends on both velocity and temperature, see "Thermal, Velocity-Dependent Model" on page 1-278.

For the velocity-dependent model these related parameters become visible in the **Friction** settings:

* **Relative velocity vector**
* **Kinetic friction coefficient vector**
* **Friction coefficient interpolation method**
* **Friction coefficient extrapolation method**

Thermal Model

You can model the effects of heat flow and temperature change by selecting a temperature-dependent model. To choose a temperature-dependent model, in the **Friction** settings, set the **Friction model** parameter to `Temperature-dependent friction coefficients`. For information about a friction model that depends on both velocity and temperature, see "Thermal, Velocity-Dependent Model" on page 1-278.

For the temperature-dependent model, thermal port **H** and these settings are visible:

* In the **Friction** settings:

  * **Temperature vector**
  * **Static friction coefficient vector**
  * **Kinetic friction coefficient vector**
  * **Friction coefficient interpolation method**
  * **Friction coefficient extrapolation method**
* In the **Thermal Port** settings:

  * **Thermal mass**
  * **Initial Temperature**

Thermal, Velocity-Dependent Model

You can model the effects of rotational velocity change and heat flow by selecting a velocity-dependent and temperature-dependent model. To choose a model that depends on both velocity and temperature, in the **Friction** settings, set the **Friction model** parameter to `Temperature and velocity-dependent friction coefficients`.

For the velocity-dependent and temperature-dependent model, thermal port **H** and these related settings and parameters become visible:

* In the **Friction** settings:

  * **Relative velocity vector**
  * **Temperature vector**

- **Static friction coefficient vector**
- **Kinetic friction coefficient matrix**
- **Friction coefficient interpolation method**
- **Friction coefficient extrapolation method**
- In the **Thermal Port** settings:
  - **Thermal mass**
  - **Initial Temperature**

Faulty Behavior

You can enable faulty behavior in response to:

- Simulation time — Faulting occurs at a specified time.
- Simulation behavior — Faulting occurs in response to an external trigger. This exposes port **T**.

You can choose either or both of these settings for block faulting. If faulting is triggered, the clutch responds according to the **Behavior when faulted** setting for the remainder of the simulation. The fault options are:

- `Cannot transmit power`
- `Cannot unlock`

You can set the block to issue a fault report as a warning or error message in the Simulink Diagnostic Viewer with the **Reporting when fault occurs** parameter.

Limitations and Assumptions

- The model does not account for inertia. Add inertia terms externally to the **B** and **F** ports as required.
- The model computes the torque assuming a uniform distribution of the normal force.

Ports

Input

N — Normal force
physical signal

Physical signal port associated with the normal force. This signal is positive or zero. A signal of less than zero is interpreted as zero.

T — External fault trigger
physical signal

Physical signal port for an external fault trigger. Triggering occurs when the value is greater than 0.5. There is no unit associated with the trigger value.

Dependencies

This port is visible when **Enable faults** is set to `On` and **Enable external fault trigger** is set to `On`.

Conserving

B — Base
mechanical rotational

Rotational conserving port associated with the driving (base) surface.

F — Follower
mechanical rotational

Rotational conserving port associated with the driven (follower) surface.

H — Heat flow
thermal

Thermal conserving port associated with heat flow.

Dependencies

This port is visible only when, in the **Friction** settings, the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see "Friction model" on page 1-0 and Friction Parameter Dependencies.

Parameters

Geometry

Geometry model — Parameterization method
`Define effective radius` (default) | `Define annular region`

Parameterization method for the loaded-contact friction model:

- `Define effective radius` — Provide a value for the friction effective radius.
- `Define annular region` — Define the friction effective radius in terms of the inside and outside diameters of the friction surface. If you select this option, the panel changes from its default.

Dependencies

The value of this parameter affects the visibility of other parameters in the **Geometry** settings.

Effective torque radius — Effective friction surface radius
130 mm (default) | positive scalar

Effective radius r_{eff}. The value must be greater than zero.

Dependencies

This parameter is visible only if the **Geometry model** parameter is set to `Define effective radius`.

Friction surface outside diameter — Friction surface outer diameter
150 mm (default) | positive scalar

Outside diameter, $2 * r_o$, of the friction surfaces. Must be greater than zero.

Dependencies

This parameter is visible only if the **Geometry model** parameter is set to `Define annular region`.

`Friction surface inside diameter` — Friction surface inner diameter
100 mm (default) | positive scalar less than outer diameter

Inside diameter, $2 * r_i$, of the friction surfaces. Must be greater than zero, but less than the friction surface outside diameter.

Dependencies

This parameter is visible only if the **Geometry model** parameter is set to `Define annular region`.

Friction

The table shows how the visibility of some ports, parameters, and settings depends on the option that you choose for other parameters. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Friction Parameter Dependencies

Friction			
Friction model			
Fixed kinetic friction coefficient	Velocity-dependent kinetic friction coefficient	Temperature-dependent friction coefficients	Temperature and velocity-dependent friction coefficients
		Exposes: • Conserving port **H** • Thermal parameters in the **Friction** settings • **Thermal Port** settings	Exposes: • Conserving port **H** • Thermal parameters in the **Friction** settings • **Thermal Port** settings
-	-	**Temperature vector**	**Temperature vector**
-	**Relative velocity vector**	-	**Relative velocity vector**
Static friction coefficient	**Static friction coefficient**	**Static friction coefficient vector**	**Static friction coefficient vector**
Kinetic friction coefficient	**Kinetic friction coefficient vector**	**Kinetic friction coefficient vector**	**Kinetic friction coefficient matrix**
-	**Friction coefficient interpolation method**	**Friction coefficient interpolation method**	**Friction coefficient interpolation method**
-	**Friction coefficient extrapolation method**	**Friction coefficient extrapolation method**	**Friction coefficient extrapolation method**
Velocity tolerance	**Velocity tolerance**	**Velocity tolerance**	**Velocity tolerance**
Threshold force	**Threshold force**	**Threshold force**	**Threshold force**
Viscous drag torque coefficient	**Viscous drag torque coefficient**	**Viscous drag torque coefficient**	**Viscous drag torque coefficient**

Friction model — Friction model

Fixed kinetic friction coefficient (default) | Velocity-dependent kinetic friction coefficient | Temperature-dependent friction coefficients | Temperature and velocity-dependent friction coefficients

Parameterization method to model the kinetic friction coefficient. The options and default values for this parameter depend on the friction model that you select for the block. The options are:

• Fixed kinetic friction coefficient — Provide a fixed value for the kinetic friction coefficient.

• Velocity-dependent kinetic friction coefficient — Define the kinetic friction coefficient by one-dimensional table lookup based on the relative angular velocity between disks.

• Temperature-dependent friction coefficients — Define the kinetic friction coefficient by table lookup based on the temperature.

- `Temperature and velocity-dependent friction coefficients` — Define the kinetic friction coefficient by table lookup based on the temperature and the relative angular velocity between disks.

Dependencies

The friction model setting affects the visibility of other parameters, settings, and ports. For more information, see Friction Parameter Dependencies.

Relative velocity vector — Relative velocity
`[0, 100, 1000]` rad/s (default) | vector

Input values for the relative velocity as a vector. The values in the vector must increase from left to right. The minimum number of values depends on the interpolation method that you select. For linear interpolation, provide at least two values per dimension. For smooth interpolation, provide at least three values per dimension.

Dependencies

This parameter is only visible when you set the **Friction model** parameter to `Velocity-dependent kinetic friction coefficient` or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Temperature vector — Temperature
`[280, 300, 320]` K (default) | increasing vector

Input values for the temperature as a vector. The minimum number of values depends on the interpolation method that you select. For linear interpolation, provide at least two values per dimension. For smooth interpolation, provide at least three values per dimension. The values in the vector must increase from left to right.

Dependencies

This parameter is only visible when you set the **Friction model** parameter to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Static friction coefficient — Static friction coefficient
`0.35` (default) | scalar

Static or peak value of the friction coefficient. The static friction coefficient must be greater than the kinetic friction coefficient.

Dependencies

this parameter is visible only when the **Friction model** parameter is set to `Fixed kinetic friction coefficient` or `Velocity-dependent kinetic friction coefficient`. For more information, see Friction Parameter Dependencies.

Static friction coefficient vector — Static friction coefficient
`[.4, .38, .36]` (default) | vector

Static, or peak, values of the friction coefficient as a vector. The vector must have the same number of elements as the temperature vector. Each value must be greater than the value of the corresponding element in the kinetic friction coefficient vector.

Dependencies

This parameter is only visible when you set the **Friction model** parameter to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Kinetic friction coefficient — Kinetic friction coefficient
`0.3` (default) | positve scalar

The kinetic, or Coulomb, friction coefficient. The coefficient must be greater than zero.

Dependencies

This parameter is only visible when you set the **Friction model** parameter to `Fixed kinetic friction coefficient`. For more information, see Friction Parameter Dependencies.

Kinetic friction coefficient vector — Kinetic friction coefficient
`[.3, .22, .19]` (default) | `[.3, .28, .25]` | vector

Output values for kinetic friction coefficient as a vector. All values must be greater than zero.

If the **Friction model** parameter is set to

- `Velocity-dependent kinetic friction coefficient` — The vector must have same number of elements as relative velocity vector.
- `Temperature-dependent friction coefficients` — The vector must have the same number of elements as the temperature vector.

Dependencies

This parameter is only visible when you set the **Friction model** parameter to `Velocity-dependent kinetic friction coefficient` or `Temperature-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Kinetic friction coefficient matrix — Kinetic friction coefficient
`[.34, .32, .3; .3, .28, .25; .25, .2, .15]` (default) | matrix

Output values for kinetic friction coefficient as a matrix. All the values must be greater than zero. The size of the matrix must equal the size of the matrix that is the result of the temperature vector × the kinetic friction coefficient relative velocity vector.

Dependencies

This parameter is only visible when you set the **Friction model** parameter to `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Friction coefficient interpolation method — Extrapolation method
`Linear` (default) | `Smooth`

Interpolation method for approximating the output value when the input value is between two consecutive grid points:

- `Linear` — Select this option to get the best performance.
- `Smooth` — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is only visible when you set the **Friction model** parameter to `Velocity-dependent kinetic friction coefficient`, `Temperature-dependent friction coefficients`, or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Friction coefficient extrapolation method — **Interpolation method**
`Linear` (default) | `Nearest` | `Error`

Extrapolation method for determining the output value when the input value is outside the range specified in the argument list:

- `Linear` — Select this option to produce a curve with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region.

- `Nearest` — Select this option to produce an extrapolation that does not go above the highest point in the data or below the lowest point in the data.

- `Error` — Select this option to avoid going into the extrapolation mode when you want your data to be within the table range. If the input signal is outside the range of the table, the simulation stops and generates an error.

For more information on extrapolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is only visible when you set the **Friction model** parameter to `Velocity-dependent kinetic friction coefficient`, `Temperature-dependent friction coefficients`, or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Velocity tolerance — **Relative velocity locking threshold**
`0.001 rad/s` (default) | scalar

Relative velocity below which the two surfaces can lock. The surfaces lock if the torque across the **B** and **F** rotational ports is less than the product of the effective radius, the static friction coefficient, and the applied normal force.

Threshold force — **Normal contact force threshold**
`1 N` (default) | scalar

The normal force applied to the physical signal port **N** is applied to the contact only if the amount of force exceeds the value of the **Threshold force** parameter. Forces below the **Threshold force** are not applied, so there is no transmitted frictional torque.

Viscous Losses

Viscous drag torque coefficient — **Viscous drag torque**
`0 N*m/(rad/s)` (default) | nonnegative scalar

Viscous drag coefficient, μ_{visc}, for computing the drag torque. The coefficient depends on the type of operating fluid, fluid temperature, and the maximum distance between the surfaces.

Initial Conditions

Initial state — Initial clutch state
Unlocked (default) | Locked

State of the internal Fundamental Friction Clutch block state at the start of simulation.

- Locked — Rotational ports **B** and **F** are initially locked together.
- Unlocked — Rotational ports **B** and **F** are initially sliding relative to each other.

Faults

Enable faults — Fault option
Off (default) | On

Enable externally or temporally triggered faults. When faulting occurs, the surfaces fail to unlock or cannot transmit power, according to the **Behavior when faulted** setting.

Behavior when faulted — Set fault response
Cannot transmit power (default) | Cannot unlock

Set fault response. You can select the faulting between the surfaces as either:

- Cannot transmit power
- Cannot unlock

Dependencies

To enable this parameter, set **Enable faults** to On.

Enable external fault trigger — External trigger option
Off (default) | On

Enables port **T**. A physical signal at port **T** that is greater than 0.5 triggers faulting.

Dependencies

To enable this parameter, set **Enable faults** to On.

Enable temporal fault trigger — Temporal trigger option
Off (default) | On

Enables fault triggering at a specified time. When the **Simulation time for fault event** is reached, the surfaces respond according to the **Behavior when faulted** setting.

Dependencies

To enable this parameter, set **Enable faults** to On.

Simulation time for fault event — Time at which faulting is triggered
5 s (default) | positive scalar

When the **Simulation time for fault event** is reached, the surfaces respond according to the **Behavior when faulted** setting.

Dependencies

To enable this parameter, set **Enable faults** to On and **Enable temporal fault trigger** to On.

`Reporting when fault occurs` — Fault condition report
None (default) | Warning | Error

Reporting preference for the fault condition. When reporting is set to Warning or Error, a message is displayed in the Simulink Diagnostic Viewer. When Error is selected, the simulation will stop if faulting occurs.

Dependencies

To enable this parameter, set **Enable faults** to On.

Thermal Port

Thermal Port settings are visible only when, in the **Friction** settings, the **Friction model** parameter is set to Temperature-dependent friction coefficients or Temperature and velocity-dependent friction coefficients. For more information, see Friction Parameter Dependencies.

`Thermal mass` — Thermal mass
50 kJ/K (default) | scalar

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

This parameter is only visible when, in the **Friction** settings, the **Friction model** parameter is set to Temperature-dependent friction coefficients or Temperature and velocity-dependent friction coefficients. For more information, see Friction Parameter Dependencies.

`Initial temperature` — Initial temperature
300 K (default) | scalar

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

This parameter is only visible when, in the **Friction** settings, the **Friction model** parameter is set to Temperature-dependent friction coefficients or Temperature and velocity-dependent friction coefficients. For more information, see Friction Parameter Dependencies.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Fundamental Friction Clutch | Loaded-Contact Translational Friction

Topics
"Gear Coupling Control Using Clutches"

Introduced in R2011a

Loaded-Contact Translational Friction

Loaded-contact friction between two sliding surfaces
Library: Simscape / Driveline / Brakes & Detents / Translational

Description

The Loaded-Contact Translational Friction block simulates friction between two sliding surfaces loaded with a normal force.

The block is implemented as a structural component based on the Fundamental Friction Clutch block. From the locked state, the two surfaces unlock if the exerted force exceeds the static friction, as defined by the static coefficient of friction and current normal force. For details on how the locking and unlocking are modeled, see the Fundamental Friction Clutch block reference page.

Force is transmitted for normal forces larger than the **Threshold force** parameter.

You can also enable faulting. When faulting occurs, the surfaces will remain locked or will be unable to transmit power. Faults can occur at a specified time or due to an external trigger at port **T**.

Equations

The block simulates friction between two sliding surfaces loaded with a normal force. When the two sliding surfaces are not locked, the friction force is determined with these equations:

$$F = N \cdot \mu \cdot \text{sign}(v_{rel}) + F_{\text{visc}} \quad,$$

$$F_{visc} = \mu_{\text{visc}} \cdot v_{rel} \quad,$$

where:

- F is the transmitted force.
- N is the normal force.
- μ is the friction coefficient.
- v_{rel} is the relative velocity.
- F_{visc} is the viscous drag force.
- μ_{visc} is the viscous drag force coefficient.

Velocity-Dependent Model

You can model the effects of rotational velocity change by selecting a velocity-dependent model. To choose a velocity-dependent model, in the **Friction** settings, set the **Friction model** parameter to `Velocity-dependent kinetic friction coefficient`. For information about a friction model that depends on both velocity and temperature, see "Thermal, Velocity-Dependent Model" on page 1-78.

For the velocity-dependent model these related parameters become visible in the **Friction** settings:

- **Relative velocity vector**
- **Kinetic friction coefficient vector**
- **Friction coefficient interpolation method**
- **Friction coefficient extrapolation method**

Thermal Model

You can model the effects of heat flow and temperature change by selecting a temperature-dependent model. To choose a temperature-dependent model, in the **Friction** settings, set the **Friction model** parameter to `Temperature-dependent friction coefficients`. For information about a friction model that depends on both velocity and temperature, see "Thermal, Velocity-Dependent Model" on page 1-78.

For the temperature-dependent model, thermal port **H** and these settings are visible:

- In the **Friction** settings:

  - **Temperature vector**
  - **Static friction coefficient vector**
  - **Kinetic friction coefficient vector**
  - **Friction coefficient interpolation method**
  - **Friction coefficient extrapolation method**
- In the **Thermal Port** settings:

  - **Thermal mass**
  - **Initial Temperature**

Thermal, Velocity-Dependent Model

You can model the effects of rotational velocity change and heat flow by selecting a velocity-dependent and temperature-dependent model. To choose a model that depends on both velocity and temperature, in the **Friction** settings, set the **Friction model** parameter to `Temperature and velocity-dependent friction coefficients`.

For the velocity-dependent and temperature-dependent model, thermal port **H** and these related settings and parameters become visible:

- In the **Friction** settings:

  - **Relative velocity vector**
  - **Temperature vector**
  - **Static friction coefficient vector**
  - **Kinetic friction coefficient matrix**
  - **Friction coefficient interpolation method**
  - **Friction coefficient extrapolation method**
- In the **Thermal Port** settings:

  - **Thermal mass**
  - **Initial Temperature**

Faulty Behavior

You can enable faulty behavior in response to:

- Simulation time — Faulting occurs at a specified time.
- Simulation behavior — Faulting occurs in response to an external trigger. This exposes port **T**.

You can choose either or both of these settings for block faulting. If faulting is triggered, the clutch responds according to the **Behavior when faulted** setting for the remainder of the simulation. The fault options are:

- `Cannot transmit power`
- `Cannot unlock`

You can set the block to issue a fault report as a warning or error message in the Simulink Diagnostic Viewer with the **Reporting when fault occurs** parameter.

Limitations and Assumptions

- The model does not account for body mass. Add mass terms externally to the **B** and **F** ports as required.
- The model computes the transmitted force assuming a uniform distribution of the normal force.

Ports

Input

N — Normal force
physical signal

Physical signal port associated with the normal force. This signal is positive or zero. A signal of less than zero is interpreted as zero.

T — External fault trigger
physical signal

Physical signal port for an external fault trigger. Triggering occurs when the value is greater than 0.5. There is no unit associated with the trigger value.

Dependencies

This port is visible when **Enable faults** is set to `On` and **Enable external fault trigger** is set to `On`.

Conserving

B — Base
mechanical rotational

Rotational conserving port associated with the driving (base) surface.

F — Follower
mechanical rotational

Rotational conserving port associated with the driven (follower) surface.

H — Heat flow
thermal

Thermal conserving port associated with heat flow.

Dependencies

This port is visible only when, in the **Friction** settings, the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see "Friction model" on page 1-0 and Friction Parameter Dependencies.

Parameters

Friction

The table shows how the visibility of some ports, parameters, and settings depends on the option that you choose for other parameters. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Friction Parameter Dependencies

Friction			
Friction model			
Fixed kinetic friction coefficient	Velocity-dependent kinetic friction coefficient	Temperature-dependent friction coefficients	Temperature and velocity-dependent friction coefficients
		Exposes: • Conserving port **H** • Thermal parameters in the **Friction** settings • **Thermal Port** settings	Exposes: • Conserving port **H** • Thermal parameters in the **Friction** settings • **Thermal Port** settings
-	-	**Temperature vector**	**Temperature vector**
-	**Relative velocity vector**	-	**Relative velocity vector**
Static friction coefficient	**Static friction coefficient**	**Static friction coefficient vector**	**Static friction coefficient vector**
Kinetic friction coefficient	**Kinetic friction coefficient vector**	**Kinetic friction coefficient vector**	**Kinetic friction coefficient matrix**
-	**Friction coefficient interpolation method**	**Friction coefficient interpolation method**	**Friction coefficient interpolation method**
-	**Friction coefficient extrapolation method**	**Friction coefficient extrapolation method**	**Friction coefficient extrapolation method**
Velocity tolerance	**Velocity tolerance**	**Velocity tolerance**	**Velocity tolerance**
Threshold force	**Threshold force**	**Threshold force**	**Threshold force**
Viscous drag torque coefficient	**Viscous drag torque coefficient**	**Viscous drag torque coefficient**	**Viscous drag torque coefficient**

Friction model — Friction model
Fixed kinetic friction coefficient (default) | Velocity-dependent kinetic friction coefficient | Temperature-dependent friction coefficients | Temperature and velocity-dependent friction coefficients

Parameterization method to model the kinetic friction coefficient. The options and default values for this parameter depend on the friction model that you select for the block. The options are:

• `Fixed kinetic friction coefficient` — Provide a fixed value for the kinetic friction coefficient.

• `Velocity-dependent kinetic friction coefficient` — Define the kinetic friction coefficient by one-dimensional table lookup based on the relative angular velocity between disks.

• `Temperature-dependent friction coefficients` — Define the kinetic friction coefficient by table lookup based on the temperature.

- `Temperature and velocity-dependent friction coefficients` — Define the kinetic friction coefficient by table lookup based on the temperature and the relative angular velocity between disks.

Dependencies

The friction model setting affects the visibility of other parameters, settings, and ports. For more information, see Friction Parameter Dependencies.

`Relative velocity vector` — Relative velocity
`[0, .1, 1]` m/s (default) | vector

Input values for the relative velocity as a vector. The values in the vector must increase from left to right. The minimum number of values depends on the interpolation method that you select. For linear interpolation, provide at least two values per dimension. For smooth interpolation, provide at least three values per dimension.

Dependencies

This parameter is only visible when you set the **Friction model** parameter to `Velocity-dependent kinetic friction coefficient` or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

`Temperature vector` — Temperature
`[280, 300, 320]` K (default) | increasing vector

Input values for the temperature as a vector. The minimum number of values depends on the interpolation method that you select. For linear interpolation, provide at least two values per dimension. For smooth interpolation, provide at least three values per dimension. The values in the vector must increase from left to right.

Dependencies

This parameter is only visible when you set the **Friction model** parameter to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

`Static friction coefficient` — Static friction coefficient
`0.35` (default) | scalar

Static or peak value of the friction coefficient. The static friction coefficient must be greater than the kinetic friction coefficient.

Dependencies

this parameter is visible only when the **Friction model** parameter is set to `Fixed kinetic friction coefficient` or `Velocity-dependent kinetic friction coefficient`. For more information, see Friction Parameter Dependencies.

`Static friction coefficient vector` — Static friction coefficient
`[.4, .38, .36]` (default) | vector

Static, or peak, values of the friction coefficient as a vector. The vector must have the same number of elements as the temperature vector. Each value must be greater than the value of the corresponding element in the kinetic friction coefficient vector.

Dependencies

This parameter is only visible when you set the **Friction model** parameter to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

`Kinetic friction coefficient` — Kinetic friction coefficient
`0.3` (default) | positve scalar

The kinetic, or Coulomb, friction coefficient. The coefficient must be greater than zero.

Dependencies

This parameter is only visible when you set the **Friction model** parameter to `Fixed kinetic friction coefficient`. For more information, see Friction Parameter Dependencies.

`Kinetic friction coefficient vector` — Kinetic friction coefficient
`[.3, .22, .19]` (default) | `[.3, .28, .25]` | vector

Output values for kinetic friction coefficient as a vector. All values must be greater than zero.

If the **Friction model** parameter is set to

- `Velocity-dependent kinetic friction coefficient` — The vector must have same number of elements as relative velocity vector.
- `Temperature-dependent friction coefficients` — The vector must have the same number of elements as the temperature vector.

Dependencies

This parameter is only visible when you set the **Friction model** parameter to `Velocity-dependent kinetic friction coefficient` or `Temperature-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

`Kinetic friction coefficient matrix` — Kinetic friction coefficient
`[.34, .32, .3; .3, .28, .25; .25, .2, .15]` (default) | matrix

Output values for kinetic friction coefficient as a matrix. All the values must be greater than zero. The size of the matrix must equal the size of the matrix that is the result of the temperature vector × the kinetic friction coefficient relative velocity vector.

Dependencies

This parameter is only visible when you set the **Friction model** parameter to `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

`Friction coefficient interpolation method` — Extrapolation method
`Linear` (default) | `Smooth`

Interpolation method for approximating the output value when the input value is between two consecutive grid points:

- `Linear` — Select this option to get the best performance.
- `Smooth` — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is only visible when you set the **Friction model** parameter to `Velocity-dependent kinetic friction coefficient`, `Temperature-dependent friction coefficients`, or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

`Friction coefficient extrapolation method` — **Interpolation method**
`Linear` (default) | `Nearest` | `Error`

Extrapolation method for determining the output value when the input value is outside the range specified in the argument list:

- `Linear` — Select this option to produce a curve with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region.
- `Nearest` — Select this option to produce an extrapolation that does not go above the highest point in the data or below the lowest point in the data.
- `Error` — Select this option to avoid going into the extrapolation mode when you want your data to be within the table range. If the input signal is outside the range of the table, the simulation stops and generates an error.

For more information on extrapolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is only visible when you set the **Friction model** parameter to `Velocity-dependent kinetic friction coefficient`, `Temperature-dependent friction coefficients`, or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

`Velocity tolerance` — **Relative velocity locking threshold**
`0.001` m/s (default) | scalar

Relative velocity below which the two surfaces can lock. The surfaces lock if the torque across the **B** and **F** rotational ports is less than the product of the effective radius, the static friction coefficient, and the applied normal force.

`Threshold force` — **Normal contact force threshold**
`1` N (default) | scalar

The normal force applied to the physical signal port **N** is applied to the contact only if the amount of force exceeds the value of the **Threshold force** parameter. Forces below the **Threshold force** are not applied, so there is no transmitted frictional torque.

Viscous Losses

`Viscous drag torque coefficient` — **Viscous drag torque**
`0` N/(m/s) (default) | nonnegative scalar

Viscous drag coefficient, μ_{visc}, for computing the drag torque. The coefficient depends on the type of operating fluid, fluid temperature, and the maximum distance between the surfaces.

Initial Conditions

Initial state — Initial clutch state
Unlocked (default) | Locked

State of the internal Fundamental Friction Clutch block state at the start of simulation.

- Locked — Rotational ports **B** and **F** are initially locked together.
- Unlocked — Rotational ports **B** and **F** are initially sliding relative to each other.

Faults

Enable faults — Fault option
Off (default) | On

Enable externally or temporally triggered faults. When faulting occurs, the surfaces fail to unlock or cannot transmit power, according to the **Behavior when faulted** setting.

Behavior when faulted — Set fault response
Cannot transmit power (default) | Cannot unlock

Set fault response. You can select the faulting between the surfaces as either:

- Cannot transmit power
- Cannot unlock

Dependencies

To enable this parameter, set **Enable faults** to On.

Enable external fault trigger — External trigger option
Off (default) | On

Enables port **T**. A physical signal at port **T** that is greater than 0.5 triggers faulting.

Dependencies

To enable this parameter, set **Enable faults** to On.

Enable temporal fault trigger — Temporal trigger option
Off (default) | On

Enables fault triggering at a specified time. When the **Simulation time for fault event** is reached, the surfaces respond according to the **Behavior when faulted** setting.

Dependencies

To enable this parameter, set **Enable faults** to On.

Simulation time for fault event — Time at which faulting is triggered
5 s (default) | positive scalar

When the **Simulation time for fault event** is reached, the surfaces respond according to the **Behavior when faulted** setting.

Dependencies

To enable this parameter, set **Enable faults** to `On` and **Enable temporal fault trigger** to `On`.

Reporting when fault occurs — Fault condition report
None (default) | Warning | Error

Reporting preference for the fault condition. When reporting is set to `Warning` or `Error`, a message is displayed in the Simulink Diagnostic Viewer. When `Error` is selected, the simulation will stop if faulting occurs.

Dependencies

To enable this parameter, set **Enable faults** to `On`.

Thermal Port

Thermal Port settings are visible only when, in the **Friction** settings, the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Thermal mass — Thermal mass
50 kJ/K (default) | scalar

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

This parameter is only visible when, in the **Friction** settings, the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Initial temperature — Initial temperature
300 K (default) | scalar

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

This parameter is only visible when, in the **Friction** settings, the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see Friction Parameter Dependencies.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Fundamental Friction Clutch | Loaded-Contact Rotational Friction

Topics

"Gear Coupling Control Using Clutches"

Introduced in R2011a

Logic-Controlled Clutch

Friction clutch with input control
Library: Simscape / Driveline / Clutches

Description

The Logic-Controlled Clutch represents a logic-controlled mechanism that transmits rotational power through friction. The block is an advanced version of the Fundamental Friction Clutch block. You control the Logic-Controlled Clutch block with physical signal at port **E**.

If the clutch is engaging and in motion, the block applies the kinetic torque. If the clutch is engaging and not in motion, the block applies the static torque.

Ports

Input

E — Control clutch engagement
physical signal

Physical signal port associated with the clutch logic.

Conserving

B — Base interface
mechanical rotational

Mechanical rotational conserving port associated with the base interface.

F — Follower interface
mechanical rotational

Mechanical rotational conserving port associated with the follower interface.

Parameters

Transition model — Clutch engagement parameterization
Instantaneous (default) | First-order lag | Scaled by input

Clutch engagement transition control. When you select:

- Instantaneous — The clutch engages when the input signal at port **E** rounds to 1, and the clutch does not engage when the input signal rounds to 0.
- First-order lag — The block introduces a first-order delay to the clutch engagement according to the value that you specify for the **Transition time constant** parameter.

- `Scaled by input` — The block smooths the signal input at port **E**. The block applies full static and kinetic friction when the input signal is greater than or equal to 1 and no static or kinetic friction torque when the signal is nonpositive. The block linearly scales the friction torques when the input signal is between 0 and 1.

Transition time constant — First-order lag time constant
`0.1 s` (default) | positive scalar

Time constant for first-order lag clutch engagement.

Dependencies

To enable this parameter, set **Transition model** to `First-order lag`.

Kinetic friction torque — Kinetic friction torque threshold
`10 N*m` (default) | nonnegative scalar

Minimum torque threshold to maintain relative motion between the clutch plates.

Static friction torque — Static friction torque threshold
`20 N*m` (default) | positive scalar

Minimum torque threshold to cause relative motion between the clutch plates.

Clutch velocity tolerance — Motion boundary condition
`0.001 rad/s` (default) | nonnegative scalar

Clutch minimum angular velocity.

Initial State — Clutch condition at simulation start
`Unlocked` (default) | `Locked`

Option to start the simulation with the clutch unlocked or locked.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Cone Clutch | Disk Friction Clutch | Dog Clutch | Fundamental Friction Clutch

Introduced in R2016b

Longitudinal Vehicle

Longitudinal motion abstract vehicle with ideal brakes and tires
Library: Simscape / Driveline / Tires & Vehicles
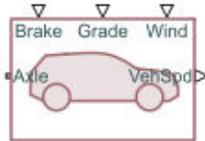
Description

The Longitudinal Vehicle block represents an abstract vehicle confined to longitudinal motion. You can parameterize an arbitrary vehicle or choose from predefined parameterizations. The block includes optional non-slipping tires and ideal brakes.

You can use this block as a simpler alternative to the Vehicle Body block. The Longitudinal Vehicle block requires less information to parameterize and is more suitable for a nascent design.

The block equations use these variables:

- m is the **Vehicle mass** parameter.
- v_x is the longitudinal velocity output at the **VehSpd** port.
- F_x is the sum of longitudinal forces.
- r_{tire} is the radius of the tire and is equivalent to the **Tire rolling radius** parameter.
- F_{drive}, F_{brake}, and F_{resist} are the driving force of the vehicle, the braking force, and the vehicle resistance force, respectively. F_{brake} and F_{resist} oppose the driving motion of the vehicle in the force balance equation.
- τ_{axle} is the torque at the axle.
- F_B is the brake force input from the **Brake** port.
- ω_{axle} is the rotational speed of the axle.
- ω_1 is an arbitrary constant that adjusts the speed range in which the block suppresses F_{brake}.
- F_{tire} is the tire rolling resistance.
- F_{air} is the drag force due to air resistance.
- θ is the road incline angle input from either the **Angle** or **Grade** port.
- v_1 is an arbitrary constant that adjusts the speed range in which the block suppresses F_{air} and F_{tire}.

Parameterizations

For all settings of the **Parameterization type** parameter, the block calculates the equations of longitudinal vehicle speed as

$$m\frac{dv_x}{dt} = \sum F_x,$$

where

$$v_x = r_{tire}\omega_{axle},$$

and

$$\sum F_x = F_{drive} - F_{brake} - F_{resist}.$$

The block treats the drive direction as positive in the force balance equation. It solves this equation using these substitutions:

$$\sum F_x = \underbrace{\frac{\tau_{axle}}{r_{tire}}}_{F_{drive}} - \underbrace{F_B \tanh\left(\frac{\omega_{axle}}{\omega_1}\right)}_{F_{brake}} - \underbrace{\left\{(F_{tire}\cos\theta + F_{air})\tanh\left(\frac{v_x}{v_1}\right) + mg\sin\theta\right\}}_{F_{resist}}.$$

You can set **Parameterization type** to `Road-load` or `Regular parameter set` to simulate an arbitrary vehicle. You can also choose from three predefined parameterizations, `Typical small car`, `Typical medium-sized car`, and `Typical large sports utility vehicle`.

Regular parameter set

When you set **Parameterization type** to `Regular parameter set`, you can specify the **Tire rolling coefficient**, **Air drag coefficient**, and **Vehicle frontal area** parameters of the vehicle, which the block converts to road-load coefficients.

The block computes the tire rolling resistance as

$$F_{tire} = C_R mg,$$

where C_R is the **Tire rolling coefficient** parameter.

The block computes the air resistance as

$$F_{air} = \frac{1}{2}C_D A_f \rho_a (v_x + v_w)^2,$$

where C_D is the **Air drag coefficient** parameter, A_f is the **Vehicle frontal area** parameter in m^2, and ρ_A is the density of dry air. The block assumes $\rho_A = 1.184\ kg/m^3$ at 1 atmosphere.

Road-load

When you set **Parameterization type** to `Road-load`, the block applies the standard road-load formulation and computes the core physics as:

$$F_{RL} = \underbrace{A_{RL} + B_{RL}v_x}_{F_{tire}} + \underbrace{C_{RL}v_x^2}_{F_{air}},$$

where A_{RL}, B_{RL}, and C_{RL} are the road-load coefficients.

Note Each road-load coefficient is specific to its associated units of v_x. There are no standard units for these coefficients; however, some texts may assume certain units. Select the correct units when you parameterize these coefficients.

The block uses these coefficients to solve for the values in the core physics as

$$F_{tire} = A_{RL} + B_{RL}v_x,$$

and

$$F_{air} = C_{RL}(v_x + v_w)^2 .$$

Here, v_w is the wind speed from the **Wind** port.

Predefined Parameterizations

You can set **Parameterization type** to three predefined parameterizations: `Typical small car`, `Typical medium-sized car`, and `Typical large sports utility vehicle`. The values represent industry averages rather than specific vehicles. You can start with one of these predefined parameterizations and later adjust these values by setting **Parameterization type** to `Road-load` or `Regular parameter set` and using the pertinent information from the tables.

Regular parameter set

Parameterization type	Vehicle mass	Tire rolling radius	Tire rolling coefficient	Drag coefficient	Frontal Area
Typical small car	1100 kg	0.3 m	0.013	0.3	0.9*1.65*1.45 = 2.153 m^2
Typical medium-sized car	1800 kg	0.3 m	0.0136	0.31	0.9*1.75*1.5 = 2.36 m^2
Typical large sports utility vehicle	2600 kg	0.4 m	0.014	0.36	0.9*1.88*1.85 = 3.13 m^2

Note that each predefined parameterization assumes $B = 0$ for the road-load method.

Road-load

Parameterization type	A	B	C
Typical small car	140.3 N	0 N/(m/s)	0.3824 N*s^2/m^2
Typical medium-sized car	240.1 N	0 N/(m/s)	0.4336 N*s^2/m^2
Typical large sports utility vehicle	357.1 N	0 N/(m/s)	0.6671 N*s^2/m^2

Where the **Vehicle frontal area** equation corresponds to *0.9·vehicle width·vehicle height*.

Variables

Use the **Variables** tab to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Ports

Input

Brake — Longitudinal braking force, N
physical signal

Physical signal input port associated with the brakes. The brake port ignores negative values.

Dependencies

To enable this port, set **Show brake force inport** to On.

Angle — Road incline, rad
physical signal

Physical signal input port associated with the road incline. A road grade of 1 is equivalent to a π/4 radian angle.

Dependencies

To enable this port, set

- **Show road slope inport** to On
- **Slope input type** to Incline angle

Grade — Normalized road grade, unitless
physical signal

Physical signal input port associated with the road grade. A road grade of 1 is equivalent to a π/4 radian angle.

Dependencies

To enable this port, set

- **Show road slope inport** to On
- **Slope input type** to Grade percent

Wind — Longitudinal wind speed, m/s
physical signal

Physical signal input port associated with the longitudinal wind speed. This value is the longitudinal component of the wind speed, where positive values represent a headwind and negative values represent a tailwind.

Dependencies

To enable this port, set **Show wind speed inport** to On.

Output

VehSpd — Vehicle speed, m/s
physical signal

Physical signal output port associated with the longitudinal velocity of the vehicle.

G — Longitudinal G force, unitless
physical signal

Physical signal port associated with the ratio of the longitudinal vehicle acceleration, a_x, to the gravitational acceleration, g.

Dependencies

To enable this port, set **Show G outport** to On.

Conserving

Axle — Wheel rotation
mechanical rotational

Mechanical rotational conserving port associated with the axle connection to the wheel hub. Connect this port to a gear system such as a reduction gear.

Parameters

Vehicle

Parameterization type — Vehicle parameterization selection
Regular parameter set (default) | Road-load | Typical small car | Typical medium-sized car | Typical large sports utility vehicle

Option to parameterize the longitudinal vehicle. You can parameterize an arbitrary vehicle by choosing Regular parameter set or Road-load, or select one of three typical vehicle body specifications. Refer to "Parameterizations" on page 1-302 for more information.

Vehicle mass — Inertial mass
1200 kg (default) | positive scalar

Effective mass of the longitudinal vehicle.

Dependencies

To enable this parameter, set **Parameterization type** to Regular parameter set or Road-load.

Tire rolling radius — Rolling radius of tire
0.3 m (default) | positive scalar

Distance from the wheel center to the outside surface of the tire while resisting the force of the vehicle due to gravity.

Dependencies

To enable this parameter, set **Parameterization type** to Regular parameter set or Road-load.

Tire rolling coefficient — Rolling coefficient of tire
0.0136 (default) | nonnegative scalar

Dimensionless coefficient of the tire rolling resistance. The block uses this value to calculate F_{tire} as $F_{tire} = C_R mg$. Here, the variable C_R represents this parameter.

Dependencies

To enable this parameter, set **Parameterization type** to Regular parameter set.

Air drag coefficient — Vehicular drag coefficient
0.31 (default) | nonnegative scalar

Dimensionless coefficient of vehicular drag. The block uses this value to compute F_{air} as $F_{air} = 1/2 \cdot C_D A_f \rho (v_x + v_w)^2$, where C_D represents this parameter.

Dependencies

To enable this parameter, set **Parameterization type** to `Regular parameter set`.

Vehicle frontal area — Forward-facing surface area
2.3 m^2 (default) | positive scalar

Effective area of the vehicle that succumbs to drag. You can use a low-fidelity heuristic formula to find this measurement during early design: $A_f = 0.9 \cdot height \cdot width$.

Dependencies

To enable this parameter, set **Parameterization type** to `Regular parameter set`.

Road-load coefficient A — A_{RL} coefficient
120 N (default) | positive scalar

Road-load coefficient A_{RL}.

Dependencies

To enable this parameter, set **Parameterization type** to `Road-load`.

Road-load coefficient B — B_{RL} coefficient
1.8 N/(m/s) (default) | positive scalar

Road-load coefficient B_{RL}.

Dependencies

To enable this parameter, set **Parameterization type** to `Road-load`.

Road-load coefficient C — C_{RL} coefficient
0.389 N*s^2/m^2 (default) | positive scalar

Road-load coefficient C_{RL}.

Dependencies

To enable this parameter, set **Parameterization type** to `Road-load`.

Gravitational acceleration — Acceleration due to gravity
9.81 m/s^2 (default) | positive scalar

Gravitational constant that acts on the mass of the vehicle as a function of the axle rotation. Here, vehicle weight = vehicle mass·gravitational acceleration. The block applies this weight to the tire to calculate the rolling resistance. The block also uses this value to determine the G-force multiple when you set **Show G outport** to `On`.

Simulation

Vehicle speed threshold — Tolerance for vehicle motion
0.3 m/s (default) | scalar

Minimum speed at which the block simulates vehicle motion. Below the absolute value of this threshold, the block smoothly transitions to zero.

Axle speed threshold — Tolerance for axle rotation
0.01 rad/s (default) | scalar

Minimum rotational speed at which the block simulates axle rotation. Below the absolute value of this threshold, the block smoothly transitions to zero.

Show brake force inport — Ideal brakes port
On (default) | Off

Option to display **Brake** port.

Show road slope inport — Road slope port
On (default) | Off

Option to enable a port to input data related to the road grade or slope. Setting this parameter to On enables the **Slope input type** parameter.

Slope input type — Type of slope measurement
Grade percent (default) | Incline angle

Option to display either the **Grade** or **Angle** port.

Dependencies

To enable this parameter, set **Show road slope inport** to On.

Show wind speed inport — Wind port
On (default) | Off

Option to display the **Wind** port.

Show G outport — G-force outport
Off (default) | On

Option to display the **G** port. record G-force as a multiple of the **Gravitational acceleration** parameter.

References

[1] Eriksson, Lars, and Lars Nielsen. *Modeling and Control of Engines and Drivelines: Eriksson/ Modeling and Control of Engines and Drivelines*. Chichester, UK: John Wiley & Sons, Ltd, 2014. https://doi.org/10.1002/9781118536186.

[2] Heywood, John B. *Internal Combustion Engine Fundamentals*. Second edition. New York: McGraw-Hill Education, 2018.

[3] Society of Automotive Engineers: Light Duty Vehicle Performance and Economy Measure Committee. "Chassis Dynamometer Simulation of Road Load Using Coastdown Techniques. SAE J2264" SAE International. 2014. https://doi.org/10.4271/J2264_201401.

[4] Society of Automotive Engineers: Light Duty Vehicle Performance and Economy Measure Committee. "Road Load Measurement and Dynamometer Simulation Using Coastdown Techniques. SAE J1263" SAE International. 2010. https://doi.org/10.4271/J1263_201003.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Vehicle Body

Introduced in R2021b

Motor & Drive

Generic motor and drive with closed-loop torque control
Library: Simscape / Driveline / Engines & Motors
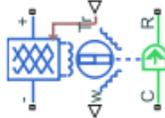
Description

The Motor & Drive block represents a generic brushless motor and drive with closed-loop torque control. It is a simplified version of the Motor & Drive (System Level) block. The Motor & Drive block is useful if you need a generic or low-fidelity motor implementation in your system. It is also suited for cases when you do not know all of your motor specifications or you want to use the block to find an appropriate motor for your system.

To enable faster simulation, the block abstracts the motor, drive electronics, and control. The block generates a torque-speed envelope that saturates the input torque, and it permits only the range of torques and speeds that the envelope defines.

Modeling Electrical Losses

The Motor & Drive block models first-order losses based on the overall efficiency for a given speed and torque, which you specify as **Motor and driver overall efficiency (percent)**, **Speed at which efficiency is measured**, and **Torque at which efficiency is measured**, respectively. The block uses the speed and torque to generate a torque-speed envelope. The envelope saturates the input torque, which yields the torque that the motor responds to, τ_{elec}. This is also the torque that the block uses to compute the electrical losses.

The block only considers torque-dependent resistive losses such that

$$P_{losses} = k\tau_{elec}^2,$$

where

$$k = \frac{\omega_\eta(1 - \eta/100)}{\tau_\eta \cdot \eta/100}.$$

Resistive losses are also known as Ohmic losses and occur due to the tendency of the armature windings to resist the flow of electrons. The electrical power includes these losses such that

$$P_{elec} = P_{losses} + \omega\tau_{elec}.$$

The rate of conversion from electrical energy to heat energy is defined by Joule's law:

$$I = \frac{P_{elec}}{V},$$

where:

- P_{elec} is the electrical power that the block calculates and uses in the governing equation.

- P_{losses} is the electrical power lost during operation. When you model the effects of heat flow and temperature change, this value represents the rate of heat flow that gets distributed into the thermal mass or out port **H**.

- ω is the angular velocity of the rotor. This is equivalent to the **W** output port value.

- τ_{elec} is the saturated torque demand.

- k is the proportionality constant for resistance losses, which has the units *(energy*time)⁻¹*.

- η is the efficiency of the motor and driver for a given speed and torque. This value is equivalent to the **Motor and driver overall efficiency (percent)** parameter.

- ω_η is the angular velocity that corresponds to the overall efficiency. This value is equivalent to the **Speed at which efficiency is measured** parameter.

- τ_η is the torque that corresponds to the overall efficiency. This value is equivalent to the **Torque at which efficiency is measured** parameter.

- V is the voltage across the terminals.

- I is the current through the terminals.

When you enable thermal modeling, P_{losses} represents the contribution from the block to the heat flow.

To include series resistance, fixed losses, and iron losses, you can add blocks to your model or use the Motor & Drive (System Level) block.

Tip You can add damping and inertia with the Rotational Damper block and Inertia block, respectively.

Thermal Modeling

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Thermal port** to Model.

When you model the effects of heat flow and temperature change, the electrical losses from the motor contribute to these effects.

Variables

Use the **Variables** tab to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Dependencies

To enable this setting, set **Thermal port** to Model.

Assumptions and Limitations

- The motor driver tracks a torque demand with the time constant T_c.
- Motor speed fluctuations due to mechanical load do not affect the motor torque tracking.

Ports

Input

Tr — Reference torque demand, N*m
physical signal

Physical signal input port associated with the reference torque demand.

Output

W — Mechanical rotational speed, rad/s
physical signal

Physical signal output port associated with the mechanical rotational speed.

Conserving

+ — Positive electrical DC supply
electrical

Electrical conserving port associated with the positive electrical DC supply.

- — Negative electrical DC supply
electrical

Electrical conserving port associated with the negative electrical DC supply.

C — Motor case
mechanical rotational

Mechanical rotational conserving port associated with the motor case.

R — Motor rotor
mechanical rotational

Mechanical rotational conserving port associated with the motor rotor.

H — Heat flow
thermal

Thermal conserving port associated with heat flow. Electrical losses from the motor contribute to the heat flow through this port.

Dependencies

To enable this port, set **Thermal port** to Model.

Parameters

Maximum torque — Torque to define torque-speed envelope boundary
400 N*m (default) | positive scalar

Maximum permissible torque value. The block uses this value and the **Maximum power** parameter to define the torque-speed envelope.

Maximum power — Power to define torque-speed envelope boundary
200 kW (default) | positive scalar

Maximum permissible power value. The block uses this value and the **Maximum torque** parameter to define the torque-speed envelope.

Torque control time constant, Tc — Output torque time step
0.02 s (default) | positive scalar

Torque controller output time interval. Use this parameter to tell the block how long to wait between issuing torque value output information.

Motor and driver overall efficiency (percent) — Energy efficiency for given speed and torque
100 (default) | scalar in the range (0,100]

Efficiency to convert from electrical to mechanical rotational energy.

Speed at which efficiency is measured — Given speed for efficiency measurement
3750 rpm (default) | positive scalar

Speed that the block uses to calculate torque-dependent electrical losses.

Torque at which efficiency is measured — Given torque for efficiency measurement
100 N*m (default) | positive scalar

Torque that the block uses to calculate torque-dependent electrical losses.

Thermal port — Simulate thermal losses in the mass of the electrical wiring
Omit (default) | Model

Option to enable the thermal port and include thermal losses in your simulation.

Thermal mass — Tendency of the wiring to retain heat
100 J/K (default) | positive scalar

Thermal mass of the electrical winding, defined as the energy required to raise the temperature by one temperature unit.

Dependencies

To enable this parameter, set **Thermal port** to Model.

See Also
Motor & Drive (System Level) (Simscape Electrical)

Introduced in R2021a

Nonlinear Rotational Damper

Rotational damper based on polynomial or table lookup parameterizations

Library

Simscape / Driveline / Couplings & Drives / Springs & Dampers

Description

The block represents a nonlinear rotational damper. Polynomial and table lookup parameterizations define the nonlinear relationship between damping torque and relative angular velocity. The damping torque can be symmetric or asymmetric about the zero velocity point. The block applies equal and opposite damping torques on the two rotational conserving ports.

The symmetric polynomial parameterization defines the damping torque for both positive and negative relative velocities according to the expression:

$$T = b_1\omega + sign(\omega) \cdot b_2\omega^2 + b_3\omega^3 + sign(\omega) \cdot b_4\omega^4 + b_5\omega^5,$$

where:

- T — Damping torque
- b_1, b_2, ..., b_5 — Damping coefficients
- ω — Relative angular velocity between ports R and C, $\omega = \omega_R - \omega_C$
- ω_R — Absolute angular velocity associated with port R
- ω_C — Absolute angular velocity associated with port C

To avoid zero-crossings that slow simulation, eliminate the sign function from the polynomial expression by specifying an odd polynomial ($b_2, b_4 = 0$).

The two-sided polynomial parameterization defines the damping torque for both positive and negative relative velocities according to the expression:

$$T = \begin{cases} b_{1p}\omega + b_{2p}\omega^2 + b_{3p}\omega^3 + b_{4p}\omega^4 + b_{5p}\omega^5, \ \omega \geq 0 \\ b_{1n}\omega + b_{2n}\omega^2 + b_{3n}\omega^3 + b_{4n}\omega^4 + b_{5n}\omega^5, \ \omega < 0 \end{cases},$$

where:

- b_{1p}, b_{2p}, ..., b_{5p} — Damping coefficients for positive relative velocities
- b_{1n}, b_{2n}, ..., b_{5n} — Damping coefficients for negative relative velocities

Both polynomial parameterizations use a fifth-order polynomial expression. To use a lower-order polynomial, set the unneeded higher-order coefficients to zero. To use a higher-order polynomial, fit to a lower-order polynomial or use the table lookup parameterization.

The table lookup parameterization defines damping torque based on a set of torque and angular velocity vectors. If not specified, the block automatically adds a data point at the origin (zero angular velocity and zero torque).

Assumptions and Limitations

The block assumes viscous damping. The damping force depends only on velocity.

Ports

C

   Rotational conserving port.

R

   Rotational conserving port.

Parameters

Parameterization

   Select damping parameterization. Options are `By polynomial` and `By table lookup`.

   By polynomial

   Symmetry

   Choose between symmetric and two-sided polynomial parameterizations.

   - `Symmetric` — Specify a single set of polynomial coefficients governing damping for both positive and negative relative velocities.

     Vector of damping coefficients

     Enter five-element vector with polynomial damping coefficients. Physical units are for the first coefficient.

     The default vector is `[1e-4 0 1e-5 0 1e-6]`. The default unit is `N*m/(rad/s)`.

   - `Two-sided` — Specify two sets of polynomial coefficients governing damping, one for positive relative velocities, the other for negative relative velocities.

     Vector of positive rotation damping coefficients

     Enter five-element vector with polynomial damping coefficients for positive relative velocities. Physical units are for the first coefficient.

     The default vector is `[1e-4 0 1e-5 0 1e-6]`. The default unit is `N*m/(rad/s)`.

     Vector of negative rotation damping coefficients

     Enter five-element vector with polynomial damping coefficients for negative relative velocities. Physical units are for the first coefficient.

     The default vector is `[0.001 -1e-5 1e-4 0 0]`.

1-315

By table lookup

Angular velocity vector

Enter vector with relative velocity values. The vector requires a minimum number of elements, based on the selected interpolation method — two for `Linear`, and three for `Smooth`. The number of elements must match the torque vector.

The default vector is `[-1 -0.5 -0.3 −0.1 0.1 0.3 0.5 1]`. The default unit is `rad/s`.

Torque vector

Enter vector with damping torque values corresponding to velocity vector. The vector requires a minimum number of elements, based on the selected interpolation method — two for `Linear`, three for `Smooth`. The number of elements must match the velocity vector.

The default vector is `[-0.001 -4e-4 -2e-4 -5e-5 5e-5 2e-4 4e-4 0.001]`. The unit is `N*m`.

Interpolation Method

Select method used to find velocity–torque values between specified data points.

- `Linear` — Select this option to get the best performance.
- `Smooth` — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

Extrapolation Method

Select method used to calculate values outside the table lookup data range.

- `Linear` — Select this option to produce a curve with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region.
- `Nearest` — Select this option to produce an extrapolation that does not go above the highest point in the data or below the lowest point in the data.

For more information on extrapolation algorithms, see the PS Lookup Table (1D) block reference page.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Nonlinear Rotational Damper | Nonlinear Translational Damper | Rotational Damper | Translational Damper | Variable Translational Damper

Introduced in R2013a

Nonlinear Rotational Spring

Torsional spring based on polynomial or table lookup parameterizations

Library

Simscape / Driveline / Couplings & Drives / Springs & Dampers

Description

The block represents a torsional spring with nonlinear torque-displacement curve. The spring torque magnitude is a general function of displacement. It need not satisfy Hooke's law. Polynomial and table lookup parameterizations provide two ways to specify the torque-displacement relationship. The spring torque can be symmetric or asymmetric with respect to zero deformation.

The symmetric polynomial parameterization defines spring torque according to the expression:

$$T = k_1\theta + sign(\theta) \cdot k_2\theta^2 + k_3\theta^3 + sign(\theta) \cdot k_4\theta^4 + k_5\theta^5,$$

where:

- T — Spring force
- k_1, k_2, ...,k_5 — Spring coefficients
- θ — Relative displacement between ports R and C, $\theta = \theta_{init} + \theta_R - \theta_C$
- θ_{init} — Initial spring deformation
- θ_R — Absolute angular position of port R
- θ_C — Absolute angular position of port C

At simulation start (t=0), θ_R and θ_C are zero, making θ equal to θ_{init}.

To avoid zero-crossings that slow simulation, eliminate the sign function from the polynomial expression by specifying an odd polynomial ($b_2, b_4 = 0$).

The two-sided polynomial parameterization defines spring torque according to the expression:

$$T = \begin{cases} k_{1t}\theta + k_{2t}\theta^2 + k_{3t}\theta^3 + k_{4t}\theta^4 + k_{5t}\theta^5, & \theta \geq 0 \\ k_{1c}\theta - k_{2c}\theta^2 + k_{3c}\theta^3 - k_{4c}\theta^4 + k_{5c}\theta^5, & \theta < 0 \end{cases},$$

where:

- k_{1t}, k_{2t}, ..., k_{5t} — Spring tension coefficients
- k_{1c}, k_{2c}, ..., k_{5c} — Spring compression coefficients

Both polynomial parameterizations use a fifth-order polynomial expression. To use a lower-order polynomial, set the unneeded higher-order coefficients to zero. To use a higher-order polynomial, fit to a lower-order polynomial or use the table lookup parameterization.

The table lookup parameterization defines spring torque based on a set of torque and angular velocity vectors. If not specified, the block automatically adds a data point at the origin (zero angular velocity and zero torque).

Variables

Use the **Variables** tab to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Ports

C

    Rotational conserving port

R

    Rotational conserving port

Parameters

Parameterization

    Select spring parameterization. Options are `By polynomial` and `By table lookup`.

    By Polynomial

    Specify nonlinear spring function in terms of polynomial coefficients.

    Symmetry

        Choose between symmetric and two-sided polynomial parameterizations.

        • `Symmetric` — Specify one set of polynomial coefficients governing spring torque in both tension and compression.

        Vector of spring coefficients

            Enter five-element vector with polynomial spring coefficients. The highest non-zero order must be positive. Physical units are for the first coefficient.

            The default vector is `[1 0 0.1 0 0.01]`. The default unit is `N*m/rad`.

        • `Two-sided` — Specify two sets of polynomial coefficients governing spring torque, one for positive deformation, the other for negative deformation.

        Vector of spring tension coefficients

            Enter a five-element vector containing the coefficients of the polynomial spring *tension* function. The highest order non-zero coefficient must be positive. The specified physical unit is the unit of the first polynomial coefficient.

        Vector of spring compression coefficients

            Enter a five-element vector containing the coefficients of the polynomial spring *compression* function. The highest order non-zero coefficient must be positive. The specified physical unit is the unit of the first polynomial coefficient.

By table lookup

Deformation vector

Enter a vector containing the deformation values used in the table lookup. The vector must contain a minimum number of elements based on the interpolation method: two for `Linear`, and three for `Smooth`. The vector must also contain the same number of elements as the torque vector.

Torque vector

Enter a vector containing the torque values that correspond to the deformation vector values. The vector must contain a minimum number of elements based on the interpolation method: two for `Linear`, and three for `Smooth`. The vector must also contain the same number of elements as the deformation vector. If not included in the vectors, the block automatically adds a point at the (0,0) deformation–torque coordinate.

Interpolation method

Select method used to calculate deformation-torque values between table lookup data points.

- `Linear` — Select this option to get the best performance.
- `Smooth` — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

Extrapolation method

Select method used to calculate deformation-torque values outside the table lookup data range.

- `Linear` — Select this option to produce a curve with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region.
- `Nearest` — Select this option to produce an extrapolation that does not go above the highest point in the data or below the lowest point in the data.

For more information on extrapolation algorithms, see the PS Lookup Table (1D) block reference page.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Nonlinear Translational Spring | Variable Rotational Spring | Variable Translational Spring

Introduced in R2013a

Nonlinear Translational Damper

Translational damper based on polynomial or table lookup parameterizations

Library

Simscape / Driveline / Couplings & Drives / Springs & Dampers

Description

The block represents a nonlinear translational damper. Polynomial and table lookup parameterizations define the nonlinear relationship between damping force and relative linear velocity. The damping force can be symmetric or asymmetric about the zero velocity point. The block applies equal and opposite damping forces on the two translational conserving ports.

The symmetric polynomial parameterization defines the damping force for both positive and negative relative velocities according to the expression:

$$F = b_1 v + sign(v) \cdot b_2 v^2 + b_3 v^3 + sign(v) \cdot b_4 v^4 + b_5 v^5,$$

where:

- F — Damping force
- b_1, b_2, ..., b_5 — Damping coefficients
- v — Relative linear velocity between ports R and C, $v = v_R - v_C$
- v_R — Absolute linear velocity associated with port R
- v_C — Absolute linear velocity associated with port C

To avoid zero-crossings that slow simulation, eliminate the sign function from the polynomial expression by specifying an odd polynomial ($b_2, b_4 = 0$).

The two-sided polynomial parameterization defines the damping force for both positive and negative relative velocities according to the expression:

$$F = \begin{cases} b_{1e}v + b_{2e}v^2 + b_{3e}v^3 + b_{4e}v^4 + b_{5e}v^5, & v \geq 0 \\ b_{1c}v - b_{2c}v^2 + b_{3c}v^3 - b_{4c}v^4 + b_{5c}v^5, & v < 0 \end{cases},$$

where:

- b_{1e}, b_{2e}, ..., b_{5e} — Damping coefficients for positive relative velocities
- b_{1c}, b_{2c}, ..., b_{5c} — Damping coefficients for negative relative velocities

Positive relative velocities correspond to damper extension (ports R and C moving away from each other). Negative relative velocities correspond to damper contraction (ports R and C moving towards each other).

Both polynomial parameterizations use a fifth-order polynomial expression. To use a lower-order polynomial, set the unneeded higher-order coefficients to zero. For polynomials of order greater than five, fit to a polynomial of order smaller than or equal to five, or use the table lookup parameterization.

The table lookup parameterization defines damping force based on a set of velocity and force vectors. If not included in the vectors, the block automatically adds a data point at the origin (zero velocity and zero force).

Assumptions and Limitations

The block assumes viscous damping. The damping force depends only on velocity.

Ports

C

    Rotational conserving port

R

    Rotational conserving port

Parameters

Parameterization

    Select damping parameterization. Options are `By polynomial` and `By table lookup`.

    By polynomial

    Specify coefficients of polynomial damping function.

    Symmetry

        Choose between symmetric and two-sided polynomial parameterizations.

        • `Symmetric` — Specify a single set of polynomial coefficients governing damping for both positive and negative relative velocities.

        Vector of damping coefficients

            Enter five-element vector with polynomial damping coefficients. Physical units are for the first coefficient.

            The default vector is `[10 0 1 0 0.1]`. The default unit is `N/(m/s)`.

        • `Two-sided` — Specify two sets of polynomial coefficients governing damping, one for positive relative velocities, the other for negative relative velocities.

        Vector of extension damping coefficients

            Enter five-element vector with polynomial damping coefficients for positive relative velocities. Physical units are for the first coefficient.

            The default vector is `[10 0 1 0 0.1]`. The default unit is `N/(m/s)`.

Vector of contraction damping coefficients

Enter five-element vector with polynomial damping coefficients for negative relative velocities. Physical units are for the first coefficient.

The default vector is `[10 0 1 0 0.1]`. The default unit is `N/(m/s)`.

By table lookup

Velocity vector

Enter vector with relative velocity values. The vector requires a minimum number of elements, based on the selected interpolation method — two for `Linear`, and three for `Smooth`. The number of elements must match the force vector.

The default vector is `[-1 -0.5 -0.3 -0.1 0.1 0.3 0.5 1]`. The default unit is `m/s`.

Force vector

Enter vector with damping force values corresponding to velocity vector. The vector requires a minimum number of elements, based on the selected interpolation method — two for `Linear`, three for `Smooth`. The number of elements must match the velocity vector.

The default vector is `[-100 -40 -20 -5 5 20 40 100]`. The default unit is `N`.

Interpolation Method

Select method used to find intermediate velocity–force values between table lookup data points.

- `Linear` — Select this option to get the best performance.
- `Smooth` — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

Extrapolation Method

Select method used to calculate values outside the table lookup data range.

- `Linear` — Select this option to produce a curve with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region.
- `Nearest` — Select this option to produce an extrapolation that does not go above the highest point in the data or below the lowest point in the data.

For more information on extrapolation algorithms, see the PS Lookup Table (1D) block reference page.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Nonlinear Rotational Damper | Rotational Damper | Translational Damper | Variable Rotational Damper | Variable Translational Damper

Introduced in R2013a

Nonlinear Translational Spring

Translational spring based on polynomial or table lookup parameterizations
Library: Simscape / Driveline / Couplings & Drives / Springs & Dampers

Description

The Nonlinear Translational Spring block represents a translational spring with a nonlinear force-displacement curve. The spring force magnitude is a general function of displacement and does not necessarily need to satisfy Hooke's law. You can parameterize the force-displacement relationship using a polynomial or a lookup table. The spring force can be symmetric or asymmetric with respect to zero deformation.

The symmetric polynomial parameterization defines spring force according to the expression

$$F = k_1 x + sign(x) \cdot k_2 x^2 + k_3 x^3 + sign(x) \cdot k_4 x^4 + k_5 x^5,$$

where:

- F is the spring force.
- $k_1, k_2, ..., k_5$ are the spring coefficients.
- x is the relative displacement between ports **R** and **C**.
- x_{init} is the initial relative displacement.

At simulation start x is equal to x_{init}.

To avoid zero-crossings that slow simulation, eliminate the sign function from the polynomial expression by specifying an odd polynomial ($b_2, b_4 = 0$).

The asymmetric polynomial parameterization defines spring force according to the expression:

$$F = \begin{cases} k_{1t}x + k_{2t}x^2 + k_{3t}x^3 + k_{4t}x^4 + k_{5t}x^5, & x \geq 0 \\ k_{1c}x + k_{2c}x^2 + k_{3c}x^3 + k_{4c}x^4 + k_{5c}x^5, & x < 0 \end{cases},$$

where:

- $k_1, k_2, ..., k_5$ are the spring tension coefficients.
- $k_1, k_2, ..., k_5$ are the spring compression coefficients.

Both polynomial parameterizations use a fifth-order polynomial expression. To use a lower-order polynomial, set the unneeded higher-order coefficients to zero. To use a higher-order polynomial, fit to a lower-order polynomial or use the table lookup parameterization.

Setting **Parameterization** to By table lookup defines the damping torque based on a set of torque and angular velocity vectors. If you do not enter a datapoint for the origin, the block automatically adds one (zero angular velocity and zero torque).

Variables

Use the **Variables** tab to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Ports

Conserving

C — Follower port
mechanical rotational

Mechanical rotational conserving port associated with the follower end.

R — Base port
mechanical rotational

Mechanical rotational conserving port associated with the base end.

Parameters

Parameterization — Method to characterize spring behavior
By polynomial (default) | By table lookup

Spring behavior selection. You can choose to parameterize by polynomial or lookup table.

Symmetry — Spring behavior versus zero deformation position
Symmetric (default) | Two-sided

Spring behavior symmetry about the point of zero deformation. When you select Symmetric, the block applies one polynomial symmetrically about the point of zero deformation. When you select Two-sided, the block applies unique polynomials for tension and compression.

Dependencies

To enable this parameter, set **Parameterization** to By polynomial.

Vector of spring coefficients — Symmetric deformation polynomial coefficients
[100, 0, 10, 0, 1] N/m (default) | vector

Polynomial spring rate coefficients. The highest non zero-order element must be positive. Physical units represent the first coefficient, and the block selects the appropriate higher-order units for you. This vector must have a length of 5.

Dependencies

To enable this parameter, set **Parameterization** to By polynomial and **Symmetry** to Symmetric.

Vector of spring tension coefficients — Positive two-sided deformation polynomial coefficients
[100, 0, 10, 0, 1] N/m (default)

Polynomial coefficients for the spring tension. Physical units represent the first coefficient, and the block selects the appropriate higher-order units for you. This vector must have a length of 5.

Dependencies

To enable this parameter, set **Parameterization** to By polynomial and **Symmetry** to Two-sided.

Vector of spring compression coefficients — Negative two-sided deformation polynomial coefficients
[1000, -10, 100, 0, 0] N/m (default)

Polynomial coefficients for the spring compression. Physical units represent the first coefficient, and the block selects the appropriate higher-order units for you. This vector must have a length of 5.

Deformation vector — Deformation due to force table lookup entry
[-1, -.5, -.3, -.1, .1, .3, .5, 1] (default)

Relative displacement values with respect to the zero-deformation position for the corresponding elements in the **Force vector** parameter. This parameter must be the same length as the **Force vector** parameter. The minimum vector length depends on the interpolation method: two for Linear and three for Smooth. If not included in the vector, the block automatically adds a point at the origin deformation–force coordinate.

Force vector — Applied force table lookup entry
[-1000, -400, -200, -50, 50, 200, 400, 1000] (default)

Force values for the corresponding element in the **Deformation vector** parameter. This parameter must be the same length as the **Deformation vector** parameter. The minimum vector length is: two for Linear and three for Smooth. The block automatically adds a point at the origin deformation-force coordinate if you do not specify one.

Interpolation method — Method of interpolation between breakpoint values
Smooth (default) | Linear

Lookup table breakpoint interpolation. The block uses the tablelookup function to model nonlinearity by using vector data to map input values to output values:

- Linear — Select this option for the lowest computational cost.
- Smooth — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information, see tablelookup.

Extrapolation method — Method of handling input values that fall outside range of breakpoint data set
Linear (default) | Nearest | Error

Lookup table breakpoint extrapolation. This method determines the output value when the input value is outside the range specified in the argument list. The block uses the tablelookup function to model nonlinearity by using vector data to map input values to output values:

- Linear — Select this option to produce a curve with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region.
- Nearest — Select this option to produce an extrapolation that does not go above the highest point in the data or below the lowest point in the data.

- `Error` — Select this option to avoid extrapolating when you want your data to be within the table range. If the input signal is outside the range of the table, the simulation stops and generates an error.

For more information, see `tablelookup`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Nonlinear Rotational Spring | Variable Rotational Spring | Variable Translational Spring

Introduced in R2013a

Piston

Piston mechanism of reciprocating combustion engine
Library: Simscape / Driveline / Engines & Motors

Description

The Piston block represents the piston mechanism of a reciprocating combustion engine. The Piston block accounts for the instantaneous torque transmitted to the engine crankshaft, which enables you to simulate vibrations in the drivetrain due to piston revolution. To model a multi-piston engine, use the Piston Engine block.

Port **B** represents the translating piston and port **F** represents the rotating crankshaft. The piston force follows from the cylinder pressure and cross-sectional area. The block obtains the combustion pressure from a lookup table parameterized in terms of the crank angle and, optionally, the crank angular velocity and engine throttle level.

The crank torque follows from the piston force and crank angle, the crank and connecting rod lengths. In terms of these inputs, the ratio of the piston force and crank torque is

$$\frac{T_F}{F_B} = -c\left(sin(\theta) + \frac{sin(2\theta)}{2\sqrt{\left(\frac{r}{c}\right)^2 - sin^2(\theta)}}\right),$$

where:

- F_B is the instantaneous piston force associated with the base port.
- T_F is the instantaneous crank torque associated with the follower port.
- c is the crank length.
- θ is the instantaneous crank angle.
- r is the connecting rod length.
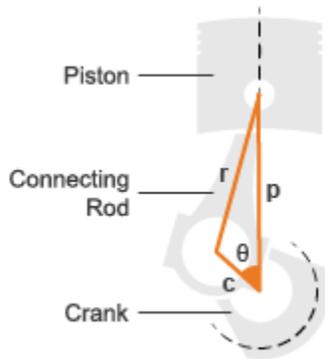
Piston Dimensions

Port **T** lets you specify the engine throttle level as a fraction between 0 and 1. This fraction corresponds to the percentage of full power generated. The block uses the data at point **T** whenever the pressure lookup table in the block dialog box is parameterized only in terms of the crank angle.

Ports

Input

T — Normalized engine throttle level, unitless
physical signal

Physical signal input port that specifies engine throttle setting.

Output

FC — Fuel consumption, kg/s
physical signal

Physical signal output port that returns the fuel consumption rate.

Dependencies

To enable this port, set **Fuel consumption model** to one of the following.

- `Constant per revolution`
- `Fuel consumption by speed and torque`
- `Brake specific fuel consumption by speed and torque`
- `Brake specific fuel consumption by speed and brake mean effective pressure`

Conserving

B — Base port associated with piston force
mechanical translational

Translational mechanical conserving port that connects to the engine block.

F — Follower port associated with crank torque
mechanical rotational

Rotational mechanical conserving port that connects to the engine crankshaft.

Parameters

Main

Cylinder bore — Inside diameter of piston cylinder wall
.10 m (default) | positive scalar

Inside diameter of the piston cylinder wall. The Piston block uses this measurement to compute the torque table. You must specify a value greater than zero.

Piston stroke — Length of translational piston travel
.06 m (default) | positive scalar

Distance from the fully retracted position to the fully extended position of the piston. The Piston block uses this measurement to convert pressure on the piston into torque values. You must specify a value greater than zero.

Piston rod length — Length of piston-to-crankshaft connecting rod
.1 m (default) | positive scalar

Distance from the center of the piston pin hole to the center of the crankshaft hole in the piston rod. The block uses this measurement to convert pressure on the piston into torque. You must specify a value greater than zero.

Number of strokes per cycle — Number of piston strokes to complete one combustion cycle
4 (default) | positive even scalar

Number of piston phases required to intake, compress, combust, and exhaust the combustion gases. Typically, engines are two-stroke or four-stroke. You must use a multiple of two.

Pressure parameterization — Method used to determine pressure applied to piston
By crank angle (default) | By crank angle and throttle | By crank angle, throttle, and crank velocity

Parameter group that you want the block to use to parameterize the pressure data.

Crank angle vector — Different piston crank angles
[-360, -90, -30, 10, 30, 90, 160, 360] deg (default) | vector

Angle of the piston crank starting at or above the minimum angle and ending at or below the maximum angle. You can calculate the minimum or maximum angle by multiplying the number of strokes by -90 degrees or 90 degrees, respectively. You must specify at least two values.

Each element in **Crank angle vector** corresponds to an element in **Pressure vector (gauge)** or the *M*-row of **Pressure matrix (gauge)** or each **Pressure 3D matrix (gauge)** matrix.

Throttle vector — Different throttle positions
[0, .3, .8, 1] (default) | vector

Different throttle positions that correspond to the pressures in the **Pressure matrix (gauge)** or **Pressure 3D matrix (gauge)** parameters. The throttle position must remain in the range [0,1] with 0 representing no throttle and 1 representing full throttle.

Each element in **Throttle vector** corresponds to the *N*-column of **Pressure matrix (gauge)** or each **Pressure 3D matrix (gauge)** matrix.

Dependencies

To enable this parameter, set **Pressure parameterization** to By crank angle and throttle or By crank angle, throttle, and crank velocity.

Crank velocity vector — Different angular velocities
[0, 1000, 6000] rpm (default) | vector

Different crank velocities that correspond to pressures in **Pressure 3D matrix (gauge)**.

Each element in **Crank velocity vector** corresponds to one of the *O*-matrices of **Pressure 3D matrix (gauge)**.

Dependencies

To enable this parameter, set **Pressure parameterization** to By crank angle, throttle, and crank velocity.

Pressure vector (gauge) — Crank angle parameterized pressures
[0, 3, 20, 50, 20, 10, 8, 0] bar (default) | vector

Vector of pressure values that correspond to different crank angle positions.

Dependencies

To enable this parameter, set **Pressure parameterization** to By crank angle.

Pressure matrix (gauge) — Crank angle and throttle parametrized pressures
M-by-*N* matrix (default) | matrix

Matrix of pressure values that correspond to different crank angle and throttle combinations. The default value is [0, 0, 0, 0; 0, .9, 2.4, 3; 0, 6, 16, 20; 0, 15, 40, 50; 0, 6, 16, 20; 0, 3, 8, 10; 0, 2.4, 6.4, 8; 0, 0, 0, 0] bar.

The elements of **Crank angle vector** correspond to the *M*-rows in **Pressure matrix (gauge)**. The elements of **Throttle vector** correspond to the *N*-columns in **Pressure matrix (gauge)**.

Dependencies

To enable this parameter, set **Pressure parameterization** to By crank angle and throttle.

Pressure 3D matrix (gauge) — Crank angle, throttle, and crank velocity parameterized pressures
M-by-*N*-by-*O* matrix (default) | 3-D matrix

Concatenated matrix of pressure values that correspond to the various crank angle, throttle, and crank velocity combinations. The default value is cat(3, [0, 0, 0, 0; 0, .9, 2.4, 3; 0, 6, 16, 20; 0, 15, 40, 50; 0, 6, 16, 20; 0, 3, 8, 10; 0, 2.4, 6.4, 8; 0, 0, 0, 0], [0, 0, 0, 0; 0, .9, 2.4, 3; 0, 6, 16, 20; 0, 15, 40, 50; 0, 6, 16, 20; 0, 3, 8, 10; 0, 2.4, 6.4, 8; 0, 0, 0, 0], [0, 0, 0, 0; 0, .9, 2.4, 3; 0, 6, 16, 20; 0, 15, 40, 50; 0, 6, 16, 20; 0, 3, 8, 10; 0, 2.4, 6.4, 8; 0, 0, 0, 0]) bar.

The elements of **Crank angle vector** correspond to the *M*-rows. The elements of **Throttle vector** correspond to the *N*-columns. The elements of **Crank velocity vector** correspond to the matrices you concatenate along dimension *O*.

By default, *M* = 8, *N* = 4, and *O* = 3.

Dependencies

To enable this parameter, set **Pressure parameterization** to By crank angle, throttle, and crank velocity.

Bearing viscous friction coefficient — Tendency of the main and rod bearings to dissipate power
.001 N*m/(rad/s) (default) | positive scalar

Aggregate viscous friction coefficient of the main and rod bearings. The block uses this parameter to compute the viscous power losses between the base and follower shafts.

Offset angle — Angular deviation between middle of engine cycle and top dead center
0 deg (default) | scalar

Point in the engine cycle when the piston reaches top dead center. The engine cycle spans in angle from -*S**180 to +*S**180 degrees, where *S* is the value of the **Number of strokes per cycle** parameter.

Initial crank angle — Crank angle at simulation start
90 deg (default) | scalar

Crank angle at time zero relative to a top dead center position.

Fuel Consumption

Fuel consumption model — Parameterization of fuel consumption
No fuel consumption (default) | Constant per revolution | Fuel consumption by speed and torque | Brake specific fuel consumption by speed and torque | Brake specific fuel consumption by speed and brake mean effective pressure

Model for calculating fuel consumption. Model parameterizations are compatible with standard industrial data.

Selecting any option other than No fuel consumption enables the **FC** port and related parameters. When you set **Fuel consumption model** to No fuel consumption, the block does not calculate fuel consumption. Selecting this option increases simulation speed.

Fuel consumption per revolution — Fuel consumed during a single crank revolution
25 mg/rev (default) | scalar

Volume of fuel consumed in one crankshaft revolution.

Dependencies

To enable this parameter, set **Fuel consumption model** to Constant per revolution.

Speed vector — Crankshaft angular velocity
[1000, 2000, 3000, 4000, 5000, 6000] rpm (default) | vector

Vector of engine speeds used in lookup table parameterizations. The vector length must match the **Torque vector**.

Dependencies

To enable this parameter, set **Fuel consumption model** to one of these settings:

- `Fuel consumption by speed and torque`
- `Brake specific fuel consumption by speed and torque`
- `Brake specific fuel consumption by speed and brake mean effective pressure`

Torque vector — Torque applied for given speed
`[0, 80, 160, 200, 240, 320, 360, 400]` N*m (default) | vector

Vector of engine torques used in the lookup table parameterizations. The vector length must match the **Speed vector** parameter.

Dependencies

To enable this parameter, set **Fuel consumption model** to `Fuel consumption by speed and torque` or `Brake specific fuel consumption by speed and torque`.

Fuel consumption table — Matrix of fuel consumption rates for given combinations of speed and torque
P-by-*Q* matrix (default) | matrix

Matrix of fuel consumption rates corresponding to engine speed and torque vectors. The number of rows, *P*, must equal the number of elements in the **Speed vector** parameter. The number of columns, *Q*, must equal the number of elements in the **Torque vector** parameter. The default value is `[.5, .9, 1.4, 1.6, 1.9, 2.7, 3.4, 4.4; 1, 1.7, 2.7, 3.1, 3.6, 5, 6, 7.4; 1.4, 2.7, 4, 4.8, 5.6, 7.5, 8.5, 10.5; 2, 3.6, 5.8, 6.7, 8, 10.4, 11.7, 13.3; 2.5, 4.8, 7.9, 9.4, 10.8, 14, 16.2, 18.6; 3.1, 6, 10.3, 11.9, 13.8, 18.4, 22, 26.5]` g/s.

Dependencies

To enable this parameter, set **Fuel consumption model** to `Fuel consumption by speed and torque` or `Brake specific fuel consumption by speed and torque`.

Brake mean effective pressure vector — Average cylinder pressure required to produce a given torque
`[0, 250, 500, 625, 750, 1000, 1150, 1250]` kPa (default)

Vector of brake mean effective pressure (BMEP) values. The BMEP satisfies the expression:

$$BMEP = T \cdot \left(\frac{2\pi \cdot n_c}{V_d} \right),$$

where:

- *T* is the output torque.
- n_c is the number of cycles per revolution.
- V_d is the cylinder displaced volume.

Dependencies

To enable this parameter, set **Fuel consumption model** to Fuel consumption by speed and torque or Brake specific fuel consumption by speed and brake mean effective pressure.

Brake specific fuel consumption table — Fuel efficiency for brake specific fuel consumption rates
P-by-*Q* matrix (default) | matrix

Matrix of brake specific fuel consumption (BSFC) rates. BSFC is the ratio of the fuel consumption rate to the output power. The number of rows, *P,* must equal the number of elements in the **Speed vector** parameter. The number of columns, *Q,* must equal the number of elements in the **Torque vector** parameter. The default value is [410, 380, 300, 280, 270, 290, 320, 380; 410, 370, 290, 270, 260, 270, 285, 320; 415, 380, 290, 275, 265, 270, 270, 300; 420, 390, 310, 290, 285, 280, 280, 285; 430, 410, 340, 320, 310, 300, 310, 320; 450, 430, 370, 340, 330, 330, 350, 380] g/hr/kW.

When you set **Fuel consumption model** to Brake specific fuel consumption by speed and torque, enter a matrix with BSFC rates corresponding to engine speed and torque vectors.

When you set **Fuel consumption model** to Brake specific fuel consumption by speed and brake mean effective pressure, enter a matrix with BSFC rates corresponding to engine speed and brake mean effective pressure (BMEP) vectors.the number of rows must equal the number of elements in the **Speed vector** parameter. The number of columns must equal the number of elements in the **Brake mean effective pressure vector** parameter.

Dependencies

To enable this parameter, set **Fuel consumption model** to Brake specific fuel consumption by speed and torque or Brake specific fuel consumption by speed and brake mean effective pressure.

Interpolation method — Speed and torque behavior between data points
Linear (default) | Smooth

Interpolation method to calculate fuel consumption at intermediate speed and torque values. Outside the data range, fuel consumption remains constant at the last value given in the lookup table for both methods.

Dependencies

To enable this parameter, set **Fuel consumption model** to one of these settings:

- Fuel consumption by speed and torque
- Brake specific fuel consumption by speed and torque
- Brake specific fuel consumption by speed and brake mean effective pressure

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

## See Also
Generic Engine | Piston Engine

Topics
"Troubleshoot Engine Issues"

Introduced in R2016a

Piston Engine

Reciprocating combustion engine with variable number of pistons
Library: Simscape / Driveline / Engines & Motors

Description

The Piston Engine block represents a reciprocating combustion engine with multiple cylinders. The piston model accounts for the instantaneous torque transmitted to the engine drive shaft. The instantaneous torque enables you to model vibrations in the drivetrain due to piston revolution. To model just the piston mechanism of a combustion engine, use the Piston block.

Port **B** represents the translating piston and port **F** the rotating crankshaft. The piston force follows from the cylinder pressure and cross-sectional area. The block obtains the combustion pressure from a lookup table parameterized in terms of the crank angle and, optionally, the crank angular velocity and engine throttle level.

The crank torque follows from the piston force and crank angle as well as the crank and connecting rod lengths. In terms of these inputs, the ratio of the piston force and crank torque is

$$\frac{T_F}{F_B} = -c\left(sin(\theta) + \frac{sin(2\theta)}{2\sqrt{\left(\frac{r}{c}\right)^2 - sin^2(\theta)}} \right),$$

where:

- F_B is the instantaneous piston force associated with the base port.
- T_F is the instantaneous crank torque associated with the follower port.
- c is the crank length.
- θ is the instantaneous crank angle.
- r is the connecting rod length.
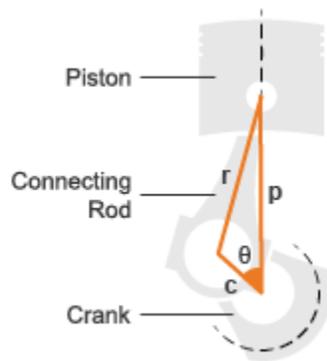
Piston Dimensions

Physical signal port T lets you specify the engine throttle level as a fraction between 0 and 1. This fraction corresponds to the percentage of full power generated. The block uses the physical signal input whenever the pressure lookup table in the block dialog box is parameterized only in terms of the crank angle.

Ports

Input

T — Normalized engine throttle level, unitless
nonnegative scalar

Engine torque demand as a fraction of maximum possible torque.

Output

FC — Fuel consumption rate, kg/s
physical signal

Fuel consumed by engine.

Conserving

B — Base port associated with piston
mechanical translational

Translational mechanical conserving port that connects to the engine block.

F — Engine crankshaft
mechanical rotational

Follower port of engine. The crankshaft transmits the power generated from the combustion process. Typically, this is where you would attach a clutch and transmission.

Parameters

Pistons

Number of pistons — Total displacement calculation
4 (default) | positive scalar

Number of pistons in the combustion engine.

Offset angle vector — Top dead center offset for each piston
[0, 180, 360, -180] (default) | vector

Vector of piston offset angles. The offset angle specifies the point in the engine cycle when the piston reaches top dead center. The engine cycle spans in angle from -S*180 to +S*180 degrees, where S is the number of strokes per cycle.

The vector size must be the same as the number of pistons. The default vector corresponds to a four-stroke, four-piston engine.

Cylinder bore — Inside diameter of piston cylinder wall
.10 m (default) | positive scalar

Inside diameter of the piston cylinder wall. The block uses this measurement to compute the torque table. You must specify a value greater than zero.

Piston stroke — Length of translational piston travel
.06 m (default) | positive scalar

Distance from the fully retracted position to the fully extended position of the piston. The block uses this measurement to convert pressure on the piston into torque values. You must specify a value greater than zero.

Piston rod length — Length of piston-to-crankshaft connecting rod
.1 m (default) | positive scalar

Distance from the center of the piston pin hole to the center of the crankshaft hole in the piston rod. The block uses this measurement to convert pressure on the piston into torque. You must specify a value greater than zero.

Number of strokes per cycle — Number of piston strokes to complete one combustion cycle
4 (default) | positive even scalar

Number of piston phases required to intake, compress, combust, and exhaust the combustion gases. Typically, engines are two-stroke or four-stroke. You must use a multiple of two.

Pressure parameterization — Determine pressure applied to piston
By crank angle (default) | By crank angle and throttle | By crank angle, throttle, and crank velocity

Parameter group that you want the block to use to parameterize the pressure data.

Crank angle vector — Different piston crank angles
[-360, -90, -30, 10, 30, 90, 160, 360] deg (default) | vector

Angle of the piston crank starting at or above the minimum angle and ending at or below the maximum angle. You can calculate the minimum or maximum angle by multiplying the number of strokes by -90 degrees or 90 degrees, respectively. You must specify at least two values.

Each element in **Crank angle vector** corresponds to an element in **Pressure vector (gauge)** or the *M*-row of **Pressure matrix (gauge)** or each **Pressure 3D matrix (gauge)** matrix.

Throttle vector — Different throttle positions
[0, .3, .8, 1] (default) | vector

Different throttle positions that correspond to the pressures in the **Pressure matrix (gauge)** or **Pressure 3D matrix (gauge)** parameters. The throttle position must remain in the range [0,1] with 0 representing no throttle and 1 representing full throttle.

Each element in **Throttle vector** corresponds to the *N*-column of **Pressure matrix (gauge)** or each **Pressure 3D matrix (gauge)** matrix.

Dependencies

To enable this parameter, set **Pressure parameterization** to By crank angle and throttle or By crank angle, throttle, and crank velocity.

Crank velocity vector — Different angular velocities
[0, 1000, 6000] rpm (default) | vector

Different crank velocities that correspond to pressures in **Pressure 3D matrix (gauge)**.

Each element in **Crank velocity vector** corresponds to one of the *O*-matrices of **Pressure 3D matrix (gauge)**.

Dependencies

To enable this parameter, set **Pressure parameterization** to By crank angle, throttle, and crank velocity.

Pressure vector (gauge) — Crank angle parameterized pressures
[0, 3, 20, 50, 20, 10, 8, 0] bar (default) | vector

Vector of pressure values that correspond to different crank angle positions.

Dependencies

To enable this parameter, set **Pressure parameterization** to By crank angle.

Pressure matrix (gauge) — Crank angle and throttle parametrized pressures
M-by-*N* matrix (default) | matrix

Matrix of pressure values that correspond to different crank angle and throttle combinations. The default value is [0, 0, 0, 0; 0, .9, 2.4, 3; 0, 6, 16, 20; 0, 15, 40, 50; 0, 6, 16, 20; 0, 3, 8, 10; 0, 2.4, 6.4, 8; 0, 0, 0, 0] bar.

The elements of **Crank angle vector** correspond to the *M*-rows in **Pressure matrix (gauge)**. The elements of **Throttle vector** correspond to the *N*-columns in **Pressure matrix (gauge)**.

Dependencies

To enable this parameter, set **Pressure parameterization** to By crank angle and throttle.

Pressure 3D matrix (gauge) — Crank angle, throttle, and crank velocity parameterized pressures
M-by-*N*-by-*O* matrix (default) | 3-D matrix

Concatenated matrix of pressure values that correspond to the various crank angle, throttle, and crank velocity combinations. The default value is `cat(3, [0, 0, 0, 0; 0, .9, 2.4, 3; 0, 6, 16, 20; 0, 15, 40, 50; 0, 6, 16, 20; 0, 3, 8, 10; 0, 2.4, 6.4, 8; 0, 0, 0, 0], [0, 0, 0, 0; 0, .9, 2.4, 3; 0, 6, 16, 20; 0, 15, 40, 50; 0, 6, 16, 20; 0, 3, 8, 10; 0, 2.4, 6.4, 8; 0, 0, 0, 0], [0, 0, 0, 0; 0, .9, 2.4, 3; 0, 6, 16, 20; 0, 15, 40, 50; 0, 6, 16, 20; 0, 3, 8, 10; 0, 2.4, 6.4, 8; 0, 0, 0, 0])` bar.

The elements of **Crank angle vector** correspond to the *M*-rows. The elements of **Throttle vector** correspond to the *N*-columns. The elements of **Crank velocity vector** correspond to the matrices you concatenate along dimension *O*.

By default, *M* = 8, *N* = 4, and *O* = 3.

Dependencies

To enable this parameter, set **Pressure parameterization** to By `crank angle, throttle, and crank velocity`.

Crankshaft

Shaft dynamics — Enable shaft dynamics modeling parameters
`No shaft dynamics - Suitable for HIL simulation` (default) | `Specify shaft stiffness, damping, and inertia`

Option to parameterize the shaft dynamics.

Base and follower bearing viscous friction coefficients — Characterize system friction
`[0, 0]` deg (default) | vector

Viscous friction coefficients for the base bearing and follower bearing, in that order.

Initial crank angle — Starting crank position
`90` deg (default) | scalar

Crank angle at time zero relative to a top dead center position. The engine cycle spans in angle from -*S**180 to +*S**180 degrees, where *S* is the number of strokes per cycle.

Stiffness — Resistance to deformation
`1e6` N*m/rad (default) | positive scalar

Translational spring stiffness of engine crankshaft. The spring stiffness accounts for elastic energy storage in the crankshaft due to material compliance.

Stiffness coefficient of the engine crankshaft. This parameter accounts for resistance to shaft deformation.

Dependencies

To enable this parameter, set **Shaft dynamics** to `Specify shaft stiffness, damping, and inertia`.

Damping — Tendency to drain energy from the system
1000 N*m/(rad/s) (default) | positive scalar

Translational damping of engine crankshaft. The damping accounts for energy dissipation in the crankshaft due to material compliance.

Dependencies

To enable this parameter, set **Shaft dynamics** to Specify shaft stiffness, damping, and inertia.

Inertia — Tendency to resist change in motion
.02 kg*m^2 (default) | positive scalar

Moment of inertia of crankshaft about its rotational axis. This parameter accounts for resistance to sudden changes in motion.

Dependencies

To enable this parameter, set **Shaft dynamics** to Specify shaft stiffness, damping, and inertia.

Initial angular deflection — Angular deflection initializing parameter
0 deg (default)

Deflection angle between the base and follower ends of the crankshaft at time zero. The deflection angle measures the angular deformation of the crankshaft due to torsion.

Dependencies

To enable this parameter, set **Shaft dynamics** to Specify shaft stiffness, damping, and inertia.

Initial angular velocity — Angular velocity initializing parameter
0 (default) | scalar

Angular velocity of the crankshaft at time zero.

Dependencies

To enable this parameter, set **Shaft dynamics** to Specify shaft stiffness, damping, and inertia.

Fuel Consumption

The table shows how the specified options for the **Fuel consumption model** parameter affects the availability of dependent parameters. To learn how to read the dependencies tables, see "Parameter Dependencies" on page B-2.

Fuel Consumption Parameter Dependencies

Fuel Consumption				
Fuel consumption model — Choose No fuel consumption, Constant per revolution, Fuel consumption by speed and torque, Brake specific fuel consumption by speed and torque, or Brake specific fuel consumption by speed and brake mean effective pressure				
No fuel consumption	Constant per revolution	Fuel consumption by speed and torque	Brake specific fuel consumption by speed and torque	Brake specific fuel consumption by speed and brake mean effective pressure
	Fuel consumption per revolution			
		Speed vector		
		Torque vector		Brake mean effective pressure vector
		Fuel consumption table	Brake specific fuel consumption table	
		Interpolation method — Choose Linear or Smooth		

Fuel consumption model — Enable fuel consumption modeling parameters
No fuel consumption (default) | Constant per revolution | Fuel consumption by speed and torque | …

Fuel consumption model based on available data. Select a model for calculating engine-fuel consumption. Model parameterizations are compatible with typical industrial data. Choose from the following options:

- No fuel consumption — The default option
- Constant per revolution
- Fuel consumption by speed and torque
- Brake specific fuel consumption by speed and torque
- Brake specific fuel consumption by speed and brake mean effective pressure

If you leave **Fuel consumption model** set to No fuel consumption, the block does not calculate fuel consumption even when the **FC** port is connected to another block. Selecting this option increases simulation speed.

Fuel consumption per piston revolution — Constant
25 mg/rev (default) | positive scalar

Constant rate of fuel consumption as a function of crankshaft revolutions. Enter the volume of fuel consumed in one crankshaft revolution.

Dependencies

To enable this parameter, set **Fuel consumption** to Constant per revolution.

Speed vector — Engine speed data
[1000, 2000, 3000, 4000, 5000, 6000] (default) | vector

Vector of engine torques that corresponds to the *M* rows of the fuel consumption lookup table.

Dependencies

To enable this parameter, set **Fuel consumption** to Fuel consumption by speed and torque, Brake specific fuel consumption by speed and torque, or Brake specific fuel consumption by speed and brake mean effective pressure.

Torque vector — Engine torque data
[0, 80, 160, 200, 240, 320, 360, 400] (default) | vector

Vector of engine torques that corresponds to the *N* columns of the fuel consumption lookup table.

Dependencies

To enable this parameter, set **Fuel consumption** to Fuel consumption by speed and torque or Brake specific fuel consumption by speed and torque.

Fuel consumption table — Engine fuel consumption data
M-by-*N* matrix (default) | matrix

Matrix of fuel consumption values that correspond Enter matrix with fuel consumption rates corresponding to engine speed and torque vectors. The number of rows must equal the number of elements in the **Speed vector**. The number of columns must equal the number of elements in the **Torque vector**. The default is [.5, .9, 1.4, 1.6, 1.9, 2.7, 3.4, 4.4; 1, 1.7, 2.7, 3.1, 3.6, 5, 6, 7.4; 1.4, 2.7, 4, 4.8, 5.6, 7.5, 8.5, 10.5; 2, 3.6, 5.8, 6.7, 8, 10.4, 11.7, 13.3; 2.5, 4.8, 7.9, 9.4, 10.8, 14, 16.2, 18.6; 3.1, 6, 10.3, 11.9, 13.8, 18.4, 22, 26.5] g/s.

Dependencies

To enable this parameter, set **Fuel consumption** to Fuel consumption by speed and torque.

Brake mean effective pressure vector — Piston pressure data
[0, 250, 500, 625, 750, 1000, 1150, 1250] (default) | vector

Vector of brake mean effective pressure (BMEP) values. The BMEP satisfies the expression:

$$BMEP = T \cdot \left(\frac{2\pi \cdot n_c}{V_d} \right),$$

where:

- *T* is the output torque.
- n_c is the number of cycles per revolution.
- V_d is the cylinder displaced volume.

Dependencies

To enable this parameter, set **Fuel consumption** to Brake specific fuel consumption by speed and brake mean effective pressure.

Brake specific fuel consumption table — Brake specific fuel consumption data
M-by-*N* matrix (default) | matrix

For the Brake specific fuel consumption by speed and torque fuel model, enter the matrix with brake specific fuel consumption (BSFC) rates corresponding to engine speed and torque

vectors. BSFC is the ratio of the fuel consumption rate to the output power. The number of rows must equal the number of elements in the **Speed vector**. The number of columns must equal the number of elements in the **Torque vector**.

For the `Brake specific fuel consumption by speed and brake mean effective pressure` fuel model, enter the matrix with brake specific fuel consumption (BSFC) rates corresponding to engine speed and brake mean effective pressure (BMEP) vectors. BSFC is the ratio of the fuel consumption rate to the output power. The number of rows must equal the number of elements in the **Speed vector**. The number of columns must equal the number of elements in the **Brake mean effective pressure vector**.

For both fuel-consumption models, the default is [410, 380, 300, 280, 270, 290, 320, 380; 410, 370, 290, 270, 260, 270, 285, 320; 415, 380, 290, 275, 265, 270, 270, 300; 420, 390, 310, 290, 285, 280, 280, 285; 430, 410, 340, 320, 310, 300, 310, 320; 450, 430, 370, 340, 330, 330, 350, 380] g/hr/kW.

Dependencies

To enable this parameter, set **Fuel consumption** to `Brake specific fuel consumption by speed and torque` or `Brake specific fuel consumption by speed and brake mean effective pressure`.

Interpolation method — Intermediate step behavior
`Linear` (default) | `Smooth`

Interpolation method to calculate fuel consumption at intermediate speed and torque values. Outside the data range, fuel consumption remains constant at the last value given in the lookup table for both methods.

Dependencies

To enable this parameter, set **Fuel consumption** to `Fuel consumption by speed and torque`, `Brake specific fuel consumption by speed and torque`, or `Brake specific fuel consumption by speed and brake mean effective pressure`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Generic Engine | Piston

Topics
"Troubleshoot Engine Issues"

**Introduced in R2016a**

Planetary Gear

Gear train with sun, planet, and ring gears
**Library:**   Simscape / Driveline / Gears
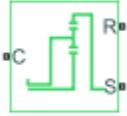
Description

The Planetary Gear block models a gear train with sun, planet, and ring gears. Planetary gears are common in transmission systems, where they provide high gear ratios in compact geometries. A carrier connected to a drive shaft holds the planet gears. Ports **C**, **R**, and **S** represent the shafts connected to the planet gear carrier, ring gear, and sun gear.
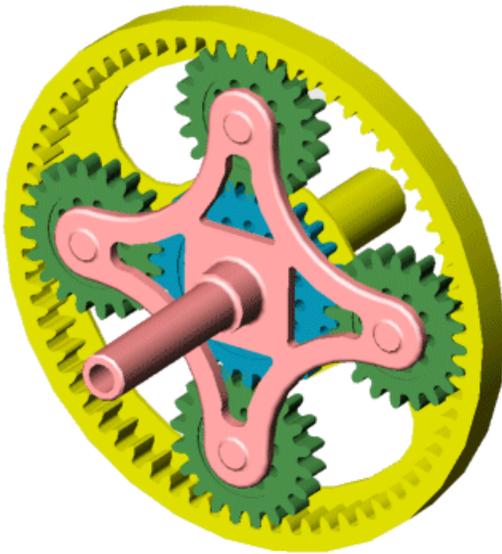
The block models the planetary gear as a structural component based on Sun-Planet and Ring-Planet Simscape Driveline blocks. The figure shows the block diagram of this structural component.

To increase the fidelity of the gear model, you can specify properties such as gear inertia, meshing losses, and viscous losses. By default, gear inertia and viscous losses are assumed to be negligible. The block enables you to specify the inertias of the internal planet gears. To model the inertias of the carrier, sun, and ring gears, connect Simscape Inertia blocks to ports **C**, **S**, and **R**.

Thermal Model

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Friction model** to Temperature-dependent efficiency.

Equations

Ideal Gear Constraints and Gear Ratios

The Planetary Gear block imposes two kinematic and two geometric constraints,

$$r_C \omega_C = r_S \omega_S + r_P \omega_P$$

$$r_R \omega_R = r_C \omega_C + r_P \omega_P$$

$$r_C = r_S + r_P$$

$$r_R = r_C + r_P$$

where:

- r_C is the radius of the carrier gear.
- ω_C is the angular velocity of the carrier gear.
- r_S is the radius of the sun gear.
- ω_S is the angular velocity of the sun gear.
- r_P is the radius of planet gear.
- ω_p is the angular velocity of the planet gears.
- r_R is the radius of the ring gear.

The ring-sun gear ratio is

$$g_{RS} = r_R / r_S = N_R / N_S,$$

where N is the number of teeth on each gear.

In terms of this ratio, the key kinematic constraint is:

$$(1 + g_{RS})\omega_C = \omega_S + g_{RS}\omega_R.$$

The four degrees of freedom reduce to two independent degrees of freedom. The gear pairs are (1, 2) = (S, P) and (P, R).

Warning The gear ratio g_{RS} must be strictly greater than one.

The torque transfer is

$$g_{RS}\tau_S + \tau_R - \tau_{loss} = 0,$$

where:

- τ_S is torque transfer for the sun gear.
- τ_R is torque transfer for the ring gear.
- τ_{loss} is torque transfer loss.

In the ideal case where there is no torque loss, $\tau_{loss} = 0$.

Nonideal Gear Constraints and Losses

In the nonideal case, $\tau_{loss} \neq 0$. For more information, see "Model Gears with Losses".

Assumptions and Limitations

- Gear inertia is assumed to be negligible.
- Gears are treated as rigid components.
- Coulomb friction slows down simulation. For more information, see "Adjust Model Fidelity".

Ports

Conserving

C — Planet gear carrier
rotational mechanical

Rotational mechanical conserving port associated with the planet gear carrier.

R — Ring gear
rotational mechanical

Rotational mechanical conserving port associated with the ring gear.

S — Sun gear
rotational mechanical

Rotational mechanical conserving port associated with the sun gear.

H — Heat flow
thermal

Thermal conserving port associated with heat flow. Heat flow affects the power transmission efficiency by altering the gear temperatures.

Dependencies

To enable this port, set **Friction model** to `Temperature-dependent efficiency`.

Parameters

Main

Ring (R) to sun (S) teeth ratio (NR/NS) — Ring gear to sun gear rotation ratio
2 (default) | scalar integer

Fixed ratio, g_{RS}, of the ring gear to the sun gear rotations as defined by the number of ring gear teeth divided by the number of sun gear teeth. The gear ratio must be strictly greater than 1.

Meshing Losses

`Friction model` — Friction model
`No meshing losses - Suitable for HIL simulation` (default) | `Constant efficiency` | `Temperature-dependent efficiency`

Friction model for the block:

- `No meshing losses - Suitable for HIL simulation` — Gear meshing is ideal.
- `Constant efficiency` — Transfer of torque between the gear wheel pairs is reduced by a constant efficiency, η, such that $0 < \eta \leq 1$.
- `Temperature-dependent efficiency` — Transfer of torque between the gear wheel pairs is defined by the table lookup based on the temperature.

`Sun-planet and ring-planet ordinary efficiencies` — Torque transfer efficiencies from the sun gear to the planet gear and from the ring gear to the planet gear
`[.96, .98]` (default) | vector

Vector of torque transfer efficiencies, $[\eta_{SP}\ \eta_{RP}]$, for sun-planet and ring-carrier gear wheel pair meshings, respectively.

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency`.

`Temperature` — Temperature
`[280, 300, 320]` K (default) | vector

Vector of temperatures used to construct a 1-D temperature-efficiency lookup table. The vector elements must increase from left to right.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

`Sun-planet efficiency` — Sun gear to planet gear torque transfer efficiency
`[.95, .9, .85]` (default) | vector

Vector of output-to-input power ratios that describe the power flow from the sun gear to the planet gear, η_{SP}. The block uses the values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1).

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

`Ring-planet efficiency` — Torque transfer efficiency from the ring gear to the planet gear
`[.95, .9, .85]` (default) | vector

Vector of output-to-input power ratios that describe the power flow from the ring gear to the planet gear, η_{RP}. The block uses the values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

`Sun-carrier and planet-carrier power thresholds` — Minimum efficiency power threshold for the sun-carrier and planet-carrier gear couplings
`[.001, .001]` W (default) | vector

Vector of power thresholds above which full efficiency factors apply. Enter the thresholds in the order sun-carrier, planet-carrier. Below these values, a hyperbolic tangent function smooths the efficiency factor.

When you set **Friction model** to `Constant efficiency`, the block lowers the efficiency losses to zero when no power is transmitted. When you set **Friction model** to `Temperature-dependent efficiency`, the block smooths the efficiency factors between zero when at rest and the values provided by the temperature-efficiency lookup tables at the power thresholds.

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency` or `Temperature-dependent efficiency`.

Viscous Losses

`Sun-carrier and planet-carrier viscous friction coefficients` — Gear viscous friction
`[0, 0]` N*m/(rad/s) (default) | vector

Vector of viscous friction coefficients, $[\mu_S, \mu_P]$, for the sun-carrier and planet-carrier gear motions, respectively.

Inertia

`Inertia` — Inertia model
`off` (default) | `on`

Inertia model for the block:

- `Off` — Model gear inertia.
- `On` — Neglect gear inertia.

`Planet gear inertia` — Planet gear inertia
`0.001` kg*m^2 (default) | positive scalar

Moment of inertia of the combined planet gears. This value must be positive.

Dependencies

To enable this parameter, set **Inertia** to `On`.

Thermal Port

To enable these settings, set **Friction model** to `Temperature-dependent efficiency`.

`Thermal mass` — Thermal mass
50 J/K (default) | scalar

Thermal energy required to change the component temperature by a single temperature unit. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

`Initial temperature` — Initial temperature
300 K (default) | scalar

Block temperature at the start of simulation. The initial temperature sets the initial component efficiencies according to their respective efficiency vectors.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

More About

Hardware-in-the-Loop Simulation

For optimal simulation performance, set **Friction model** to the default value, `No meshing losses - Suitable for HIL simulation`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Double-Pinion Planetary Gear | Compound Planetary Gear | Ravigneaux Gear

Topics
"Basic Motion, Torque, and Force Modeling"
"Model Thermal Losses in Driveline Components"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

Planet-Planet

Planetary gear set of carrier, inner planet, and outer planet wheels with adjustable gear ratio and friction losses

Library: Simscape / Driveline / Gears / Planetary Subcomponents

Description

The Planet-Planet gear block represents a carrier and two inner-outer planet gear couples. Both planet gears are connected to and rotate with respect to the carrier. The planet gears corotate with a fixed gear ratio that you specify. For model details, see "Equations" on page 1-351.
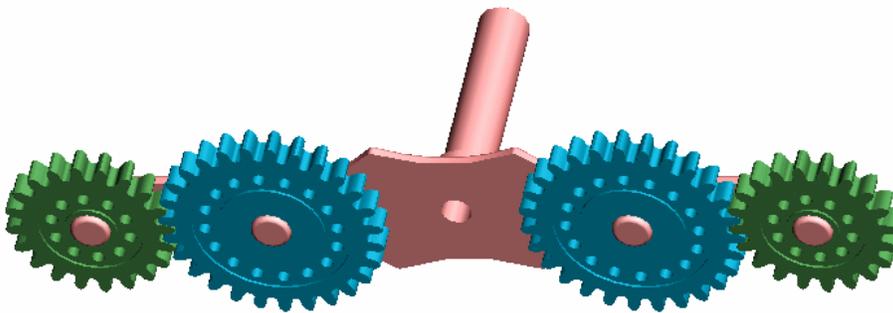
Thermal Model

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Friction model** to Temperature-dependent efficiency.

Equations

Ideal Gear Constraints and Gear Ratios

The Planet-Planet block imposes one kinematic and one geometric constraint on the three connected axes:

$$r_C \omega_C = r_{Po} \omega_{Po} + r_{Pi} \omega_{Pi}$$

$$r_C = r_{Po} + r_{Pi}$$

The outer planet-to-inner planet gear ratio is

$$g_{oi} = r_{Po}/r_{Pi} = N_{Po}/N_{Pi},$$

where N is the number of teeth on each gear. In terms of this ratio, the key kinematic constraint is

$$(1 + g_{oi})\omega_C = \omega_{Pi} + g_{oi}\omega_{Po}.$$

The three degrees of freedom reduce to two independent degrees of freedom. The gear pair is (1, 2) = (*Pi,Po*).

The torque transfer is

$$g_{oi}\tau_{Pi} + \tau_{Po} - \tau_{loss} = 0.$$

In the ideal case where there is no torque loss, $\tau_{loss} = 0$.

Nonideal Gear Constraints and Losses

In the nonideal case, $\tau_{loss} \neq 0$. For more information, see "Model Gears with Losses".

Variables

Use the **Variables** settings to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Assumptions and Limitations

- Gear inertia is assumed to be negligible.
- Gears are treated as rigid components.
- Coulomb friction slows down simulation. For more information, see "Adjust Model Fidelity".

Ports

Conserving

C — Planet gear carrier
rotational mechanical

Rotational mechanical conserving port associated with the planet gear carrier.

Po — Outer planet gear
rotational mechanical

Rotational mechanical conserving port associated with the outer planet gear.

Pi — Inner planet gear
rotational mechanical

Rotational mechanical conserving port associated with the inner planet gear.

H — Heat flow
thermal

Thermal conserving port associated with heat flow. Heat flow affects the power transmission efficiency by altering the gear temperatures.

Dependencies

To enable this port, set **Friction model** to `Temperature-dependent efficiency`.

Parameters

Main

Outer planet (Po) to inner planet (Pi) teeth ratio (NPo/NPi) — Outer to inner planet gear rotation ratio
2 (default) | positive scalar

Ratio, g_{oi}, of the outer planet gear to inner planet gear rotations as defined by the number of outer planet gear teeth divided by the number of inner planet gear teeth. This gear ratio must be strictly positive.

Meshing Losses

Friction model — Friction model
No meshing losses - Suitable for HIL simulation (default) | Constant efficiency | Temperature-dependent efficiency

Friction model for the block:

* No meshing losses - Suitable for HIL simulation — Gear meshing is ideal.
* Constant efficiency — Transfer of torque between the gear wheel pairs is reduced by a constant efficiency, η, such that $0 < \eta \leq 1$.
* Temperature-dependent efficiency — Transfer of torque between the gear wheel pairs is defined by the table lookup based on the temperature.

Ordinary efficiency — Torque transfer efficiency
0.98 (default) | scalar

Torque transfer efficiency, η_{PP}, for the outer and inner planet gear wheel pair meshing. This value must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to Constant efficiency.

Temperature — Temperature
[280, 300, 320] K (default) | vector

Vector of temperatures used to construct a 1-D temperature-efficiency lookup table. The vector elements must increase from left to right.

Dependencies

To enable this parameter, set **Friction model** to Temperature-dependent efficiency.

Efficiency — Gear efficiency
[.95, .9, .85] (default) | vector

Vector of output-to-input power ratios that describe the power flow from the outer planet gear to the inner planet gear, η_{PP}. The block uses the values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Inner planet-carrier power threshold — Minimum efficiency power threshold for the inner planet-carrier gear coupling
`0.001 W` (default) | scalar

Power threshold, p_{th}, above which full efficiency is in effect. Below this value, a hyperbolic tangent function smooths the efficiency factor.

When you set **Friction model** to `Constant efficiency`, the block lowers the efficiency losses to zero when no power is transmitted. When you set **Friction model** to `Temperature-dependent efficiency`, the block smooths the efficiency factors between zero when at rest and the values provided by the temperature-efficiency lookup tables at the power thresholds.

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency` or `Temperature-dependent efficiency`.

Viscous Losses

Inner planet-carrier viscous friction coefficient — Gear viscous friction
`0 N*m/(rad/s)` (default) | scalar

Viscous friction coefficient μ_{P_i} for the inner planet-carrier gear motion.

Thermal Port

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Thermal mass — Thermal mass
`50 J/K` (default) | scalar

Thermal energy required to change the component temperature by a single temperature unit. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

More About

Hardware-in-the-Loop Simulation

For optimal simulation performance, set **Friction model** to the default value, `No meshing losses - Suitable for HIL simulation`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Planetary Gear | Ravigneaux Gear | Ring-Planet | Sun-Planet | Sun-Planet Bevel

Topics
"Basic Motion, Torque, and Force Modeling"
"Model Thermal Losses in Driveline Components"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

Propeller

Marine propeller that converts torque into thrust
Library: Simscape / Driveline / Engines & Motors
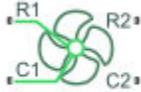
Description

The Propeller block represents a propeller with fixed or controllable blades that converts rotational mechanical energy into translational mechanical energy. You can parameterize the propeller by using constants, polynomials, or tabulated data to characterize the thrust and torque coefficients. Propellers that allow negative pitch or are designed to operate in reverse may include thrust and torque coefficient curves specific to the astern direction, which you can use in the parameterization.

You can include the wake effects of the vessel hull in the block. You can specify a constant wake fraction or enable a physical signal port, and the block will calculate the wake effects automatically.

You can use a physical signal to control the blade pitch.

This terminology is helpful for understanding the block:

- *Wake fraction* is the difference between the vessel velocity and the advance velocity expressed as a ratio of the vessel velocity.
- *Advance velocity* is the speed of the flow through the propeller, V_a.
- *Advance ratio* is the speed of the flow through the propeller with respect to the propeller tip angular speed expressed as a ratio.
- *Quadrant* is the relative two-dimensional location of the propeller operating condition where the vertical axis is V_a and the horizontal axis is ω.

  - *First quadrant*: $+V_a$, $+\omega$
  - *Second quadrant*: $+V_a$, $-\omega$
  - *Third quadrant*: $-V_a$, $-\omega$
  - *Fourth quadrant*: $-V_a$, $+\omega$
- *Pitch* is the ideal translational propeller advance distance for a single revolution.
- *Open water* is when the effects of the hull are not present.

Equations

The block equations refer to these quantities:

- *T(t)* is the smoothed propeller thrust.
- *Q(t)* is the smoothed propeller torque.
- ρ, $\rho(t)$ is the fluid density, which can function with time. You can specify the fluid density with the **Density** parameter or the **Rho** port.

- P is the pitch.
- D is the propeller diameter. This value is equivalent to the **Propeller diameter** parameter.
- $\omega(t)$ is the propeller angular speed input at port **R**. For more information about using angular units in Simscape, see "Angular Units".
- $n(t)$ is the propeller angular speed in revolutions per second, which serves to consistently nondimensionalize the torque and thrust. Here, $\omega = 2\pi n(t)$.
- n_{Thr} is the **Rotational speed threshold** parameter.
- k_T is the thrust coefficient. This value is equivalent to the **kT** parameter.
- k_Q is the torque coefficient. This value is equivalent to the **kQ** parameter.
- p_{kT} is the **kT polynomial fit coefficients** parameter.
- p_{kQ} is the **kQ polynomial fit coefficients** parameter.
- k_{Thr} is the nondimensional coefficient threshold. This value is equivalent to the **Saturation threshold for nondimensional coefficients** parameter.
- J is the advance ratio.
- V_a is the advance velocity. You can specify the advance velocity using the **Va** port.
- η is the smoothed efficiency.

The block smooths the propeller thrust and torque with respect to the rotational speed such that:

$$T = k_T \rho D^4 n \sqrt{n^2 + n_{thr}^2}$$

$$Q = k_Q \rho D^5 n \sqrt{n^2 + n_{thr}^2}$$

The block uses coefficients of thrust and torque to parameterize the performance of the propeller. You can provide static coefficients, or you can specify the coefficients as a polynomial that acts as a function of the advance ratio. The block defines the advance ratio as:

$$J = \frac{V_a n}{D(n^2 + n_{Thr}^2)},$$

where the propeller rotational speed n is linearized with the angular speed threshold n_{Thr} for smoothing.

When you set **Parameterization** to `Polynomial fit`, the block calculates the thrust and torque coefficients as:

$$k_T = \sum_{j=1:N}^{N} p_{kT,j} J^j$$

$$k_Q = \sum_{j=1:N}^{N} p_{kQ,j} J^j$$

respectively, where p_{kT} and p_{kQ} represent the polynomial coefficients.

When you set **Efficiency sensor** to `On`, the block outputs the smoothed efficiency:

$$\eta = \frac{|J|}{2\pi} \frac{k_T}{\sqrt{k_Q^2 + (0.1 k_{Thr})^2}} \cdot$$

Propeller Parameterizations

You can choose different options to parameterize the propeller k_T and k_Q based on the fidelity you desire or the type of information that is available to you. If you want to parameterize the propeller performance as a function of J, you can set **Parameterization** to Polynomial fit or Tabulated coefficients. If you want to use an asymmetrical parameterization for negative values of J, you must set **Parameterization** to Tabulated coefficients.

- Constant coefficients — This simple parameterization does not function with J. You specify k_T and k_Q as constants.
- Polynomial fit — You can specify a vector of polynomial coefficients in descending degree. For example, if you enter [.063, -.19, -.25, .37] for the **kT polynomial fit coefficients (pN...p0)** parameter, the block interprets this vector as $k_T = .063J^3 - .19J^2 - .25J + .37$. The block saturates J to be between 0 and the first positive root of the polynomial and restricts k_T and k_Q to always be positive.
- Tabulated coefficients — You can specify tabulated values for k_T and k_Q for given values of J and P/D. You must select this option if you want to use negative coefficients.

Environment Interaction

When you set **Translational connections** to Conserving, the block uses a constant wake fraction to relate the vessel velocity to the advance velocity. You input the thrust and velocity of the vessel using the **R2** and **C2** ports. The block computes the advance velocity as:

$$V_A = V(1 - w),$$

where:

- V is the vessel velocity. You can specify the vessel velocity relative to the reference using the **R2** and **C2** ports, where $V = V_{R2} - V_{C2}$..
- w is the wake fraction. This is equivalent to the **Wake fraction** parameter.

When you set **Translational connections** to Physical connections, you can use the **Va** port to supply the advance velocity as a physical signal. The block outputs the propeller thrust as a physical signal from the **Th** port.

Controlled Pitch

When you set **Blade pitch type** to Controlled, you can parameterize the propeller over a range of pitch-diameter ratios, P/D. You must specify the P/D range as a vector in the **Pitch-diameter ratio vector, P/D** parameter, where each element corresponds to a row in the k_T and k_Q matrices.

Assumptions and Limitations

- When you set **Parameterization** to Polynomial fit, the block assumes that the propeller torque and thrust coefficients are symmetric with the first quadrant.
- When you set **Parameterization** to Tabulated coefficients, the block assumes identical torque and thrust coefficients in the first quadrant and third quadrant as well as identical torque and thrust coefficients in the second quadrant and fourth quadrant.

Variables

Use the **Variables** tab to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Ports

Inputs

Va — Advance velocity, m/s
physical signal

Physical signal input port associated with the speed of the flow through the propeller.

Dependencies

To enable this port, set **Translational connections** to Physical signals.

PR — Blade pitch diameter ratio, unitless
physical signal

Physical signal input port associated with the blade pitch for a given propeller diameter.

Dependencies

To enable this port, set **Blade pitch type** to Controlled.

Rho — Variable density, kg/m^3
physical signal

Physical signal input port associated with the fluid density.

Dependencies

To enable this port, set **Fluid density specification** to Variable.

Outputs

Th — Propeller thrust, N
physical signal

Physical signal output port associated with the thrust generated by the propeller.

Dependencies

To enable this port, set **Translational connections** to Physical signals.

E — Efficiency, unitless
physical signal

Physical signal output port associated with the efficiency of the propeller. The efficiency signal is a function of the absolute value of the advance ratio.

Dependencies

To enable this port, set **Efficiency sensor** to On.

Conserving

R1 — Rod interface
mechanical rotational

Mechanical rotational conserving port associated with the rod interface.

C1 — Case interface
mechanical rotational

Mechanical rotational conserving port associated with the case interface.

R2 — Vessel velocity and thrust
mechanical translational

Mechanical translational conserving port associated with the vessel velocity and thrust.

Dependencies

To enable this port, set **Translational connections** to Conserving.

C2 — Reference current
mechanical translational

Mechanical translational conserving port associated with the motion of the current that the propeller pushes against.

Dependencies

To enable this port, set **Translational connections** to Conserving.

Parameters

Propeller

Parameterization — Parameterization selection
Constant coefficients (default) | Polynomial fit | Tabulated coefficients

Option to parameterize the propeller by constant, polynomial, or tabulated thrust and torque coefficients.

Propeller diameter — Propeller diameter
0.2 m (default) | positive scalar

Diameter of the propeller.

kT — Coefficient of thrust
0.25 (default) | scalar

Nondimensional constant thrust coefficient.

Dependencies

To enable this parameter, set **Parameterization** to Constant coefficients.

kQ — Coefficient of torque
0.05 (default) | scalar

Nondimensional constant torque coefficient.

Dependencies

To enable this parameter, set **Parameterization** to Constant coefficients.

`Blade pitch type` — Constant or controllable blade pitch
`Constant` (default) | `Controlled`

Type of blade to model. Select `Constant` for a blade with constant pitch or `Controlled` for a blade with variable pitch that you specify using the **PR** port.

Dependencies

To enable this parameter, set **Parameterization** to `Polynomial fit` or `Tabulated coefficients`.

`kT polynomial fit coefficients (pN...p0)` — Coefficient of thrust polynomial coefficients
`[.063, -.19, -.25, .37]` (default) | vector

Vector of nondimensional polynomial thrust coefficients. Specify the elements in descending order. The block uses these coefficients to generate a lookup table. For more information, see "Using Lookup Tables in Equations".

Dependencies

To enable this parameter, set:

- **Parameterization** to `Polynomial fit`
- **Blade pitch type** to `Constant`

`kQ polynomial fit coefficients (pN...p0)` — Coefficient of torque polynomial coefficients
`[.0043, -.021, -.03, .05]` (default) | vector

Vector of nondimensional polynomial thrust coefficients. Specify the elements in descending order. The block uses these coefficients to generate a lookup table. For more information, see "Using Lookup Tables in Equations".

Dependencies

To enable this parameter, set:

- **Parameterization** to `Polynomial fit`
- **Blade pitch type** to `Constant`

`Pitch-diameter ratio vector, P/D` — Pitch-diameter ratio values for thrust and torque coefficients
`[-.5, .5, .7, .9, 1.1, 1.4]` (default) | vector

Ratios of blade pitch to diameter. Each element has a corresponding row in the **kT table of polynomial coefficients (P/D, pN...p0)** and **kQ table of polynomial coefficients (P/D, pN...p0)** parameters.

Dependencies

To enable this parameter, set:

- **Parameterization** to `Polynomial fit` or `Tabulated coefficients`.
- **Blade pitch type** to `Controlled`.

kT table of polynomial coefficients (P/D, pN...p0) — Polynomial thrust coefficient table for controllable pitch propellers
[.04, -.1, -.24, .12; .118, -.24, -.24, .19; .082, -.21, -.25, .28; .063, -.19, -.25, .37; .055, -.17, -.25, .45; .064, -.21, -.19, .55] (default) | matrix

Table of polynomial thrust coefficient vectors for the given *P/D* ratios.

Dependencies

To enable this parameter, set:

- **Parameterization** to Tabulated coefficients
- **Blade pitch type** to Controlled

kQ table of polynomial coefficients (P/D, pN...p0) — Polynomial torque coefficient table for controllable pitch propellers
[.02, -.06, -.02, .02; .04, -.034, -.016, .018; .0076, -.02, -.023, .031; .0043, -.021, -.03, .05; .0047, -.024, -.036, .073; .0045, -.025, -.044, .11] (default) | matrix

Table of polynomial torque coefficient vectors for the given *P/D* ratios.

Dependencies

To enable this parameter, set:

- **Parameterization** to Tabulated coefficients
- **Blade pitch type** to Controlled

Advance ratio vector, J — Range of advance ratios for tabulated thrust and torque coefficients
[0, .3, .5, .7, .9, 1.2, 1.5] (default) | vector

Tabulated advance ratios. Each element has a corresponding element in the **kQ table, kQ(J)** and **kQ table, kQ(J)** parameters.

Dependencies

To enable this parameter, set **Parameterization** to Tabulated coefficients.

kT table, kT(J) — Tabulated thrust coefficient values for given advance ratio
[.37, .28, .21, .12, .033, -.098, -.23] (default) | vector

Tabulated thrust coefficient values as a function of the advance ratio.

Dependencies

To enable this parameter, set:

- **Parameterization** to Tabulated coefficients
- **Blade pitch type** to Constant

kQ table, kQ(J) — Tabulated torque coefficient values for given advance ratio
[.049, .039, .03, .02, .0096, -.0093, -.028] (default) | vector

Tabulated torque coefficient values as a function of the advance ratio.

Dependencies

To enable this parameter, set:

- **Parameterization** to Tabulated coefficients
- **Blade pitch type** to Constant

kT table, kQ(P/D, J) — Tabulated thrust coefficient values for given pitch-diameter ratio and advance ratio
[.12, .04, -.02, -.083, -.15, -.24, -.33; .19, .1, .029, -.047, -.12, -.23, -.35; .28, .19, .12, .032, -.054, -.18, -.31; .37, .28, .21, .12, .033, -.098, -.23; .45, .36, .29, .21, .13, -.0057, -.16; .54, .47, .41, .33, .25, .13, .0029] (default) | matrix

Tabulated thrust coefficient values as a function of the pitch-diameter ratio and the advance ratio. The columns correspond to the elements in the **Advance ratio vector, J** parameter and the rows correspond to the elements in the **Pitch-diameter ratio vector, P/D** parameter.

Dependencies

To enable this parameter, set:

- **Parameterization** to Tabulated coefficients
- **Blade pitch type** to Controlled

kQ table, kQ(P/D, J) — Tabulated torque coefficient values for given pitch-diameter ratio and advance ratio
[.02, .0091, -.0025, -.017, -.032, -.056, -.078; .017, .011, .006, .0009, -.0041, -.012, -.019; .03, .022, .015, .0074, -.0008, -.013, -.026; .049, .039, .03, .02, .0096, -.0093, -.028; .072, .06, .05, .038, .025, .0042, -.015; .11, .097, .085, .071, .057, .033, .0065] (default) | matrix

Tabulated torque coefficient values as a function of the pitch-diameter ratio and the advance ratio. The columns correspond to the elements in the **Advance ratio vector, J** parameter and the rows correspond to the elements in the **Pitch-diameter ratio vector, P/D** parameter.

Dependencies

To enable this parameter, set:

- **Parameterization** to Tabulated coefficients
- **Blade pitch type** to Controlled

Interpolation method — Method of interpolation between breakpoint values
Linear (default) | Smooth

Method to use for lookup table breakpoint interpolation. The block uses the `tablelookup` function to model nonlinearity by using array data to map input values to output values:

- Linear — Select this option for the lowest computational cost.
- Smooth — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information, see `tablelookup`.

Dependencies

To enable this parameter, set **Parameterization** to `Tabulated coefficients`.

Extrapolation method — Method of handling input values that fall outside range of breakpoint data set
`Linear` (default) | `Nearest` | `Error`

Method to use for lookup table breakpoint extrapolation. This method determines the output value when the input value is outside the range specified in the argument list. The block uses the `tablelookup` function to model nonlinearity by using array data to map input values to output values:

- `Linear` — Select this option to produce a curve with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region.
- `Nearest` — Select this option to produce an extrapolation that does not go above the highest point in the data or below the lowest point in the data.
- `Error` — Select this option to avoid extrapolating when you want your data to be within the table range. If the input signal is outside the range of the table, the simulation stops and generates an error.

Dependencies

To enable this parameter, set **Parameterization** to `Tabulated coefficients`.

Environment

Fluid density specification — Constant or variable fluid density selection
`Constant` (default) | `Variable`

Option to specify a constant or variable fluid density.

Fluid density — Constant fluid density value
`1000 kg/m^3` (default) | positive scalar

Constant fluid density value.

Dependencies

To enable this parameter, set **Fluid density specification** to `Constant`.

Translational connections — Translational motion input
`Conserving` (default) | `Physical signals`

Option to simulate advance velocity and propeller thrust as physical signal inputs and outputs, respectively, or as translational connections. This setting determines the color of the block icon, which indicates whether the block is in the translational conserving or physical signal domain. When you select `Conserving`, the constant wake fraction reduces the advance velocity relative to the vessel velocity.

Wake fraction — Wake adjusted percentage of vessel velocity
`0.2` (default) | nonnegative scalar

Reduction to vessel velocity with respect to the advance velocity. A wake fraction of `0` is equivalent to the open water condition.

Dependencies

To enable this parameter, set **Translational connections** to `Conserving`.

Advanced

`Efficiency sensor` — Operating efficiency output option
`Off` (default) | `On`

Option to enable the efficiency port **E**, which outputs a positive-valued efficiency signal.

`Rotational speed threshold` — Rotational speed smoothing point
`0.001 Hz` (default) | positive scalar

Saturation threshold value, n_{Thr}, beyond which the block applies smoothing up to the point of saturation.

`Saturation threshold for nondimensional coefficients` — Nondimensional coefficient smoothing point
`0.001` (default) | positive scalar

Saturation threshold value, k_{Thr}, where the block applies smoothing to nondimensional coefficients.

Dependencies

To enable this parameter, set **Parameterization** to `Polynomial fit` or `Tabulated coefficients`.

`Check if operating beyond normal propeller operation` — Operating bounds tolerance
`Warning` (default) | `None` | `Error`

Option to generate a warning or error when the propeller exceeds the operating parameters. The block checks whether the propeller is operating in the first quadrant. If either V_a or $n(t)$ are not positive, the propeller generates thrust and torque coefficients with the appropriate signs and also generates a warning or error, depending on the parameter setting. As you raise the value for the **Rotational speed threshold** parameter, the trigger becomes less sensitive.

When you set **Parameterization** to `Polynomial fit`, the block generates a warning or error when the propeller exceeds the first positive root of the k_T parameter.

The block approximates the hydrodynamics using symmetric or asymmetric behavior with respect to the first quadrant.

Dependencies

To enable this parameter, set **Parameterization** to `Polynomial fit` or `Tabulated coefficients`.

References

[1] Bernitsas, Michael M., D. Ray, P. Kinley. "Kt, Kq and Efficiency Curves for the Wageningen B-Series Propellers." *Report* 237. Department of Naval Architecture and Marine Engineering. College of Engineering. University of Michigan, 1981.

[2] Carlton, J. S. *Marine Propellers and Propulsion*. Second edition. Oxford: Elsevier, 2007.

See Also

Introduced in R2021b

Rack & Pinion

Rack and pinion gear coupling translational and rotational motion, with adjustable pinion radius and friction losses

Library: Simscape / Driveline / Gears / Rotational- Translational
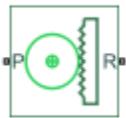
Description

The Rack & Pinion block represents rack and pinion gear that converts between translational and rotational motion. The rotational-translational gear constrains the pinion (P) and rack (R) to, respectively, rotate and translate together in a fixed ratio that you specify. You can choose whether the rack axis translates in a positive or negative direction, as the pinion rotates in a positive direction, by using the **Rack direction** parameter.

Model Variables

R_{RP}	Rack-pinion gear ratio
ω_P	Angular velocity of the pinion shaft
v_R	Translational velocity of the rack
r_P	Effective radius of the pinion
N_P	Number of teeth on the pinion
x_R	Rack tooth spacing
τ_P	Pinion shaft torque
F_R	Rack force
F_{loss}	Total loss force
F_{Coul}	Friction force
η	Torque transfer efficiency
p_{th}	Power threshold
μ_P	Viscous friction coefficient for the pinion shaft
μ_R	Viscous friction coefficient for the rack motion

Ideal Gear Constraint and Gear Ratio

Rack & Pinion imposes one kinematic constraint on the two connected axes:

$$\omega_P = R_{RP} v_R .\tag{1-6}$$

The transmission ratio is:

$$R_{RP} = 1 / r_P = \omega_P / v_N = \pm 2\pi / N_P v_R .\tag{1-7}$$

The two degrees of freedom are reduced to one independent degree of freedom. The forward-transfer gear pair convention is (1,2) = (P,R).

The torque-force transfer is:

$$R_{RP}\tau_P + F_R - F_{loss} = 0 \, , \tag{1-8}$$

with $F_{loss} = 0$ in the ideal case.

Nonideal Gear Constraint

In the nonideal case, $F_{loss} \neq 0$. For general considerations on nonideal gear modeling, see "Model Gears with Losses".

In a nonideal pinion-rack pair (P,R), the angular velocity and geometric constraints are unchanged. But the transferred torque, force, and power are reduced by:

- Coulomb friction between teeth surfaces on P and R, characterized by constant efficiency η
- Viscous coupling of driveshafts with bearings, parametrized by viscous friction coefficients μ

Meshing Efficiency

The efficiency η of meshing between pinion and rack is fully active only if the transmitted power is greater than the power threshold.

If the power is less than the threshold, the actual efficiency is automatically regularized to unity at zero velocity.

Efficiency is assumed equal for both the forward and reverse power flow.

Viscous Friction Force

The viscous friction coefficients μ_P and μ_R control the viscous friction torque and force experienced by the rack and pinion from lubricated, nonideal bearings. The viscous friction torque on the pinion axis is $-\mu_P\omega_P$. The viscous friction force on the rack motion is $-\mu_R v_R$.

Thermal Model

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Friction model** to `Temperature-dependent efficiency`.

Hardware-in-the-Loop Simulation

For optimal performance of your real-time simulation, set the **Friction model** to `No meshing losses - Suitable for HIL simulation` on the **Meshing Losses** tab.

Variables

Use the **Variables** settings to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Limitations

- Gear inertia is assumed to be negligible.

- Gears are treated as rigid components.
- Coulomb friction slows down simulation. For more information, see "Adjust Model Fidelity".

Ports

Port	Description
P	Rotational conserving port representing the pinion
R	Translational conserving port representing the rack
H	Thermal conserving port for modeling heat transfer

P is a rotational conserving port. R is a translational conserving port. They represent the pinion and the rack, respectively.

Parameters

Main

Parameterize by — Rack and gear parameterization method
Pinion radius (default) | Tooth parameters

Parameterization method of the rack and pinion gear.

- Pinion radius — Gear ratio is defined by the effective radius of the pinion.
- Tooth parameters — Gear ratio is defined by the number of teeth on the pinion gear and the rack tooth spacing.

Pinion radius — Pinion effective radius
100 mm (default) | positive scalar

Effective radius of the pinion r_P. The value must be greater than zero.

Dependencies

To enable this parameter, set **Parameterize by** to Pinion radius.

Number of pinion teeth — Number of pinion teeth
20 (default) | positive scalar

Number of teeth on the pinion N_P. The value must be greater than zero.

Dependencies

To enable this parameter, set **Parameterize by** to Tooth parameters.

Rack tooth spacing — Space between rack and teeth
5 mm (default) | positive scalar

Spacing between the teeth on the rack x_R. The value must be greater than zero.

Dependencies

To enable this parameter, set **Parameterize by** to `Tooth parameters`.

Rack direction — Rack rotational orientation
`Positive for positive pinion rotation` (default) | `Negative for positive pinion rotation`

Choose whether the rack axis translates in a positive or negative direction when the pinion rotates in a positive direction.

Meshing Losses

Friction model — Friction losses from nonideal meshing of gear threads
`No meshing losses — Suitable for HIL simulation` (default) | `Constant efficiency` | `Temperature-dependent efficiency`

- `No meshing losses — Suitable for HIL simulation` — Gear meshing is ideal.
- `Constant efficiency` — Transfer of torque between rack and pinion is reduced by friction.
- `Temperature-dependent efficiency` — Torque transfer is determined from user-supplied data for efficiency and temperature.

Efficiency — Torque transfer efficiency
`0.8` (default) | positive scalar in the range of (0,1]

Torque transfer efficiency η for the rack-pinion gear meshing, which is the same for the forward and reverse power flows. Must be greater than zero but less than, or equal to, one.

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency`.

Temperature — Array for tabular parameterization of efficiency
`[280 300 320]` K (default) | vector

Array of temperatures used to construct a 1-D temperature-efficiency lookup table. The array values must increase left to right. The temperature array must be the same size as the **Efficiency** array.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Efficiency — Array of efficiencies
`[0.95 0.9 0.85]` (default) | vector

Array of component efficiencies used to construct a 1-D temperature-efficiency lookup table. The array values are the efficiencies at the temperatures in the **Temperature** array. The two arrays must be the same size.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Rack power threshold — Power below which numerical smoothing is applied
`0.001` W (default) | positive scalar

Power threshold above which the full efficiency factor is in effect. Below this value, a hyperbolic tangent function smooths the efficiency factor, lowering the efficiency losses to zero when no power is transmitted.

Viscous Losses

`Pinion rotational viscous friction coefficient` — Pinion fluid dynamic friction coefficient
0 N*m/(rad/s) (default) | positive scalar

Viscous friction coefficient μ_P for the pinion shaft.

`Rack translational viscous friction coefficient` — Numerical constant
0 N/(m/s) (default) | positive scalar

Viscous friction coefficient μ_R for the rack motion.

Thermal Port

`Thermal mass` — Thermal mass
50 J/K (default) | positive scalar

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change.

`Initial temperature` — Initial temperature
300 K (default) | positive scalar

Component temperature at the start of simulation. The initial temperature alters the component efficiency according to an efficiency vector that you specify, affecting the starting meshing or friction losses.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Leadscrew | Simple Gear

Topics
"Basic Motion, Torque, and Force Modeling"
"Model Thermal Losses in Driveline Components"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

Ravigneaux Gear

Planetary gear with two sun gears and two planet gear sets
**Library:** Simscape / Driveline / Gears
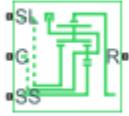
Description

The Ravigneaux Gear block represents a planetary gear train with dual sun and planet gear sets. The two sun gears are centrally located and separated longitudinally along a common rotation axis. The smaller of these gears engages an inner planet gear set, which in turn engages an outer planet gear set. The outer planet gear set, whose length spans the distance between the two sun gears, engages both the larger sun gear and the ring gear.

A carrier holds the planet gear sets in place at different radii. The carrier, which rigidly connects to a drive shaft, can spin as a unit with respect to the sun and ring gears. Revolute joints, each located between a planet gear and the carrier, enable the gears to spin about their individual longitudinal axes.

The relative angular velocities of the sun, planet, and ring gears follow from the kinematic constraints between them. For more information, see "Equations" on page 1-373.
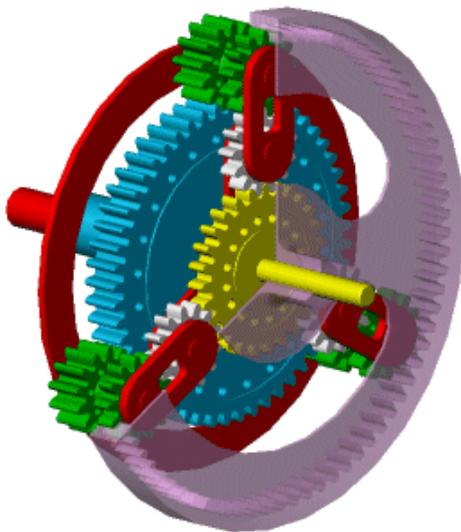
The block models the Ravigneaux gear as a structural component based on the Sun-Planet, Planet-Planet, and Ring-Planet Simscape Driveline blocks. The figure shows the equivalent block diagram of this structural component.
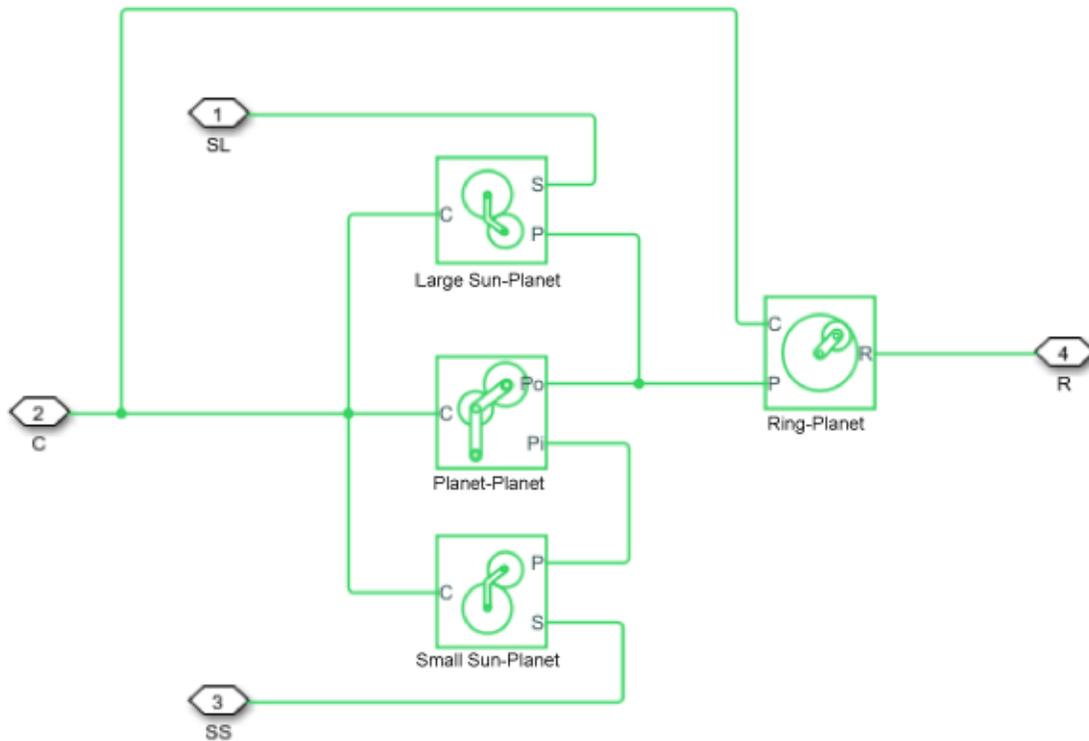
To increase the fidelity of the gear model, you can specify properties such as gear inertia, meshing losses, and viscous losses. By default, gear inertia and viscous losses are assumed to be negligible. The block enables you to specify the inertias of the internal planet gears. To model the inertias of the carrier, large sun, small sun, and ring gears, connect Simscape Inertia blocks to ports **C**, **SL**, **SS**, and **R**.

Thermal Model

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Friction model** to `Temperature-dependent efficiency`.

Equations

Ideal Gear Constraints and Gear Ratios

The Ravigneaux Gear block imposes four kinematic and four geometric constraints on the four connected axes and the two internal wheels (the inner and outer planet gears):

$$r_{Ci}\omega_C = r_{SS}\omega_{SS} + r_{Pi}\omega_{Pi}$$

$$r_{Ci} = r_{SS} + r_{Pi}$$

$$r_{Co}\omega_C = r_{SL}\omega_{SL} + r_{Po}\omega_{Po}$$

$$r_{Co} = r_{SL} + r_{Po}$$

$$(r_{Co} - r_{Ci})\omega_C = r_{Pi}\omega_{Pi} + r_{Po}\omega_{Po}$$

$$r_{Co} - r_{Ci} = r_{Po} + r_{Pi}$$

$$r_R\omega_R = r_{Co}\omega_C + r_{Po}\omega_{Po}$$

$$r_R = r_{Co} + r_{Po}$$

where:

- r_{Ci} is the radius of the inner carrier gear.
- ω_C is the angular velocity of the carrier gears.
- r_{SS} is the radius of the small sun gear.
- ω_{SS} is the angular velocity of the small sun gear.
- r_{Pi} is the radius of the inner planet gear.
- ω_{Pi} is the angular velocity of the inner planet gear.
- r_{Co} is the radius of the outer carrier gear.
- r_{SL} is the radius of the large sun gear.
- ω_{SL} is the angular velocity of the large sun gear.
- r_{Po} is the radius of the outer planet gear.
- ω_{Po} is the angular velocity of the outer planet gear.
- ω_R is the angular velocity of the ring gear.

The ring-to-sun ratios are:

$$g_{RSS} = r_R/r_{SS} = N_R/N_{SS}$$

$$g_{RSL} = r_R/r_{SL} = N_R/N_{SL}$$

where:

- g_{RSS} is the ring-to-small sun gear ratio.
- N_R is the number of teeth in the ring gear.
- N_{SS} is the number of teeth in the small sun gear.
- g_{RSS} is the ring-to-large sun gear ratio.
- N_{SL} is the number of teeth in the large sun gear.

In terms of these gear ratios, the key kinematic constraints are:

$$(g_{RSS} - 1)\omega_C = g_{RSS}\omega_R - \omega_{SS}$$

$$(g_{RSL} - 1)\omega_C = g_{RSL}\omega_R - \omega_{SL}$$

The six degrees of freedom reduce to two independent degrees of freedom. The gear pairs are (1,2) = (*LS*,*P*), (*SS*,*P*), (*P*,*R*), and (*P*,*P*).

Warning The gear ratio g_{RSS} must be strictly greater than the gear ratio g_{RSL}. The gear ratio g_{RSL} must be strictly greater than one.

The torque transfers are:

$$g_{RSS}\tau_{SS} + \tau_R - \tau_{loss}(SS, R) = 0$$

$$g_{RSL}\tau_{SL} + \tau_R - \tau_{loss}(SL, R) = 0$$

where:

- τ_{SS} is the torque transfer for the small sun gear.
- τ_R is the torque transfer for the ring gear.
- $\tau_{loss}(SS,R)$ is the torque transfer loss between the small sun gear and the ring gear.
- τ_{SL} is the torque transfer for the large sun gear.
- $\tau_{loss}(SL,R)$ is the torque transfer loss between the large sun gear and the ring gear.

In the ideal case where there is no torque loss, $\tau_{loss} = 0$.

Nonideal Gear Constraints and Losses

In the nonideal case, $\tau_{loss} \neq 0$. For more information, see "Model Gears with Losses".

Assumptions and Limitations

- The gears are assumed to be rigid.
- Coulomb friction slows down simulation. For more information, see "Adjust Model Fidelity".

Ports

Conserving

C — Planet gear carrier
rotational mechanical

Rotational mechanical conserving port associated with the planet gear carrier.

R — Ring gear
rotational mechanical

Rotational mechanical conserving port associated with the ring gear.

SL — Large sun gear
rotational mechanical

Rotational mechanical conserving port associated with the large sun gear.

SS — Small sun gear
rotational mechanical

Rotational mechanical conserving port associated with the small sun gear.

H — Heat flow
thermal

Thermal conserving port associated with heat flow. Heat flow affects the power transmission efficiency by altering the gear temperatures.

Dependencies

To enable this port, set **Friction model** to `Temperature-dependent efficiency`.

Parameters

Main

Ring (R) to large sun (SL) teeth ratio (NR/NSL) — Ring to large sun gear rotation ratio
2 (default) | positive scalar

Ratio g_{RLS} of the ring gear to sun gear rotations as defined by the number of ring gear teeth divided by the number of large sun gear teeth.

Ring (R) to small sun (SS) teeth ratio (NR/NSS) — Ring to small sun gear rotation ratio
3 (default) | positive scalar greater than (NR/NSL)

Ratio, g_{RSS}, of the ring gear wheel to small sun gear rotations as defined by the number of ring gear teeth divided by the number of small sun gear teeth. This gear ratio must be strictly greater than **Ring (R) to large sun (SL) teeth ratio (NR/NSL)**.

Meshing Losses

Friction model — Friction model
`No meshing losses - Suitable for HIL simulation` (default) | `Constant efficiency` | `Temperature-dependent efficiency`

Friction model for the block:

- `No meshing losses - Suitable for HIL simulation` — Gear meshing is ideal.
- `Constant efficiency` — Transfer of torque between the gear wheel pairs is reduced by a constant efficiency, η, such that $0 < \eta \leq 1$.
- `Temperature-dependent efficiency` — Transfer of torque between the gear wheel pairs is defined by the table lookup based on the temperature.

Large sun-planet, small sun-planet, ring-planet and planet-planet ordinary efficiencies — Torque transfer efficiencies
`[.98, .98, .98, .99]` (default) | vector

Vector of torque transfer efficiencies, $[\eta_{LS}, \eta_{SS}, \eta_{RP}, \eta_{PP}]$, for large sun-planet, small sun-planet, ring-planet, and planet-planet gear wheel pair meshings, respectively. The vector elements must be in the range (0,1).

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency`.

Temperature — Temperature
`[280, 300, 320]` K (default) | vector

Vector of temperatures used to construct a 1-D temperature-efficiency lookup table. The vector elements must increase from left to right.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

`Large sun-planet efficiency` — Torque transfer efficiency from the large sun gears to the planet gears
`[.95, .9, .85]` (default) | vector

Vector of output-to-input power ratios that describe the power flow from the large sun gear to the planet gears, η_{LSP}. The block uses the values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

`Small sun-planet efficiency` — Torque transfer efficiency from the small sun gears to the planet gears
`[.95, .9, .85]` (default) | vector

Vector of output-to-input power ratios that describe the power flow from the small sun gears to the planet gears, η_{SSP}. The block uses the values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

`Ring-planet efficiency` — Torque transfer efficiency from the ring gear to the planet gears
`[.95, .9, .85]` (default) | vector

Vector of output-to-input power ratios that describe the power flow from the ring gear to the planet gears, η_{RP}. The block uses the values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

`Planet-planet efficiency` — Planet-planet efficiency
`[.95, .9, .85]` (default) | vector

Vector of output-to-input power ratios that describe the power flow from the small sun planet gears to the large sun planet gears, η_{PP}. The block uses the values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Large sun-carrier, small sun-carrier, large sun planet-carrier and small sun planet-carrier power thresholds — Minimum efficiency power threshold for the large sun-carrier, small sun-carrier, large sun planet-carrier, and small sun planet-carrier gear couplings
[.001, .001, .001, .001] W (default) | vector

Vector of power thresholds above which the full efficiency factors apply. Enter the thresholds in the order large sun gear, small sun gear, large sun planet gears, and small sun planet gears, all relative to the gear carrier. Below these values, a hyperbolic tangent function smooths the efficiency factor.

When you set **Friction model** to `Constant efficiency`, the block lowers the efficiency losses to zero when no power is transmitted. When you set **Friction model** to `Temperature-dependent efficiency`, the block smooths the efficiency factors between zero when at rest and the values provided by the temperature-efficiency lookup tables at the power thresholds.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Viscous Losses

Large sun-carrier, small sun-carrier, large sun planet-carrier and small sun planet-carrier viscous friction coefficients — Gear viscous friction
[0, 0, 0, 0] N*m/(rad/s) (default) | vector

Vector of viscous friction coefficients, [μ_{LS}, μ_{SS}, μ_{LSP}, μ_{SSP}], for the large sun-carrier, small sun-carrier, large sun planet-carrier, and small sun planet-carrier gear motions, respectively.

Inertia

Inertia — Inertia model
off (default) | on

Inertia model for the block:

- `Off` — Model gear inertia.
- `On` — Neglect gear inertia.

Inner planet gear inertia — Inner planet gear inertia
0.001 kg*m^2 (default) | positive scalar

Moment of inertia of the inner planet gear. This value must be positive.

Dependencies

To enable this parameter, set **Inertia** to `On`.

Outer planet gear inertia — Outer planet gear inertia
0.001 kg*m^2 (default) | positive scalar

Moment of inertia of the outer planet gear. This value must be positive.

Dependencies

To enable this parameter, set **Inertia** to On.

Thermal Port

To enable these settings, set **Friction model** to Temperature-dependent efficiency.

Thermal mass — Thermal mass
50 J/K (default) | scalar

Thermal energy required to change the component temperature by a single temperature unit. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

To enable this parameter, set **Friction model** to Temperature-dependent efficiency.

Initial temperature — Initial temperature
300 K (default) | scalar

Block temperature at the start of simulation. The initial temperature sets the initial component efficiencies according to their respective efficiency vectors.

Dependencies

To enable this parameter, set **Friction model** to Temperature-dependent efficiency.

More About

Hardware-in-the-Loop Simulation

For optimal simulation performance, set **Friction model** to the default value, No meshing losses - Suitable for HIL simulation.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Compound Planetary Gear | Planet-Planet | Planetary Gear | Ring-Planet | Sun-Planet

Topics
"Basic Motion, Torque, and Force Modeling"
"Ravigneaux Four-Speed Transmission"
"Model Thermal Losses in Driveline Components"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

Ring-Planet

Planetary gear set of carrier, planet, and ring wheels with adjustable gear ratio and friction losses
Library: Simscape / Driveline / Gears / Planetary Subcomponents
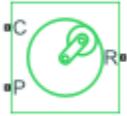
Description

The Ring-Planet block represents a carrier, a ring gear, and a set of planet gears. The planet gears are connected to and rotate with respect to the carrier. The planet and ring gears corotate with a fixed gear ratio that you specify. A ring-planet and a sun-planet gear are basic elements of a planetary gear set. For model details, see "Equations" on page 1-381.

Thermal Model

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Friction model** to `Temperature-dependent efficiency`.

Equations

Ideal Gear Constraints and Gear Ratios

The Ring-Planet block imposes one kinematic and one geometric constraint on the three connected axes:

$$r_R \omega_R = r_C \omega_C + r_P \omega_P$$

$$r_R = r_C + r_P$$

The ring-planet gear ratio is

$$g_{RP} = r_R/r_P = N_R/N_P,$$

where N is the number of teeth on each gear. In terms of this ratio, the key kinematic constraint is

$$g_{RP}\omega_R = \omega_P + (g_{RP} - 1)\omega_C.$$

The three degrees of freedom reduce to two independent degrees of freedom. The gear pair is (1, 2) = (P, R).

Warning The ring-planet gear ratio g_{RP} must be strictly greater than one.

The torque transfer is:

$$g_{RP}\tau_P + \tau_R - \tau_{loss} = 0.$$

In the ideal case where there is no torque loss, $\tau_{loss} = 0$.

Nonideal Gear Constraints and Losses

In the nonideal case, $\tau_{loss} \neq 0$. For more information, see "Model Gears with Losses".

Variables

Use the **Variables** settings to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Assumptions and Limitations

- Gear inertia is assumed to be negligible.
- Gears are treated as rigid components.
- Coulomb friction slows down simulation. For more information, see "Adjust Model Fidelity".

Ports

Conserving

C — Planet gear carrier
rotational mechanical

Rotational mechanical conserving port associated with the planet gear carrier.

R — Ring gear
rotational mechanical

Rotational mechanical conserving port associated with the ring gear.

P — Planet gear
rotational mechanical

Rotational mechanical conserving port associated with the planet gear.

H — Heat flow
thermal

Thermal conserving port associated with heat flow. Heat flow affects the power transmission efficiency by altering the gear temperatures.

Dependencies

To enable this port, set **Friction model** to `Temperature-dependent efficiency`.

Parameters

Main

Ring (R) to planet (P) teeth ratio (NR/NP) — Ring to planet gear rotation ratio
2 (default) | scalar greater than 1

Ratio, g_{RP}, of the ring gear to planet gear rotations as defined by the number of ring gear teeth divided by the number of planet gear teeth. This gear ratio must be strictly greater than 1.

Meshing Losses

Friction model — Friction model
No meshing losses - Suitable for HIL simulation (default) | Constant efficiency | Temperature-dependent efficiency

Friction model for the block:

- `No meshing losses - Suitable for HIL simulation` — Gear meshing is ideal.
- `Constant efficiency` — Transfer of torque between the gear wheel pairs is reduced by a constant efficiency, η, such that $0 < \eta \le 1$.
- `Temperature-dependent efficiency` — Transfer of torque between the gear wheel pairs is defined by the table lookup based on the temperature.

Ordinary efficiency — Torque transfer efficiency
0.98 (default) | scalar

Torque transfer efficiency, η_{RP}, for the outer and inner planet gear wheel pair meshing. This value must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency`.

Temperature — Temperature
[280, 300, 320] K (default) | vector

Vector of temperatures used to construct a 1-D temperature-efficiency lookup table. The vector elements must increase from left to right.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Efficiency — Gear efficiency
[.95, .9, .85] (default) | vector

Vector of output-to-input power ratios that describe the power flow from the ring gear to the planet gear, η_{RP}. The block uses the values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Planet-carrier power threshold — Minimum efficiency power threshold for the planet-carrier gear coupling
`0.001` W (default) | scalar

Power threshold, p_{th}, above which full efficiency is in effect. Below this value, a hyperbolic tangent function smooths the efficiency factor.

When you set **Friction model** to `Constant efficiency`, the block lowers the efficiency losses to zero when no power is transmitted. When you set **Friction model** to `Temperature-dependent efficiency`, the block smooths the efficiency factors between zero when at rest and the values provided by the temperature-efficiency lookup tables at the power thresholds.

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency` or `Temperature-dependent efficiency`.

Viscous Losses

Planet-carrier viscous friction coefficient — Gear viscous friction
`0 N*m/(rad/s)` (default) | scalar

Viscous friction coefficient, μ_P, for the planet-carrier gear motion.

Thermal Port

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Thermal mass — Thermal mass
`50 J/K` (default) | scalar

Thermal energy required to change the component temperature by a single temperature unit. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

More About

Hardware-in-the-Loop Simulation

For optimal simulation performance, set **Friction model** to the default value, `No meshing losses - Suitable for HIL simulation`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Compound Planetary Gear | Planet-Planet | Planetary Gear | Sun-Planet | Sun-Planet Bevel

Topics
"Basic Motion, Torque, and Force Modeling"
"Adjust Model Fidelity"
"Epicyclic Gear Efficiency Measurement"
"Model Thermal Losses in Driveline Components"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

Road Profile

Road profile with varying elevation and friction properties
Library: Simscape / Driveline / Tires & Vehicles

Description

The Road Profile block represents a road profile with varying elevation and friction properties. As a vehicle travels, the axle parameters and the position of the center of gravity (CG) determine the position of the front and rear axles. The block uses the axle positions to compute vehicle angle and, optionally, tire friction coefficients. The block models friction using either Magic Formula or kinetic and static friction coefficients.

Variables

Use the **Variables** tab to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Ports

Conserving

H — Horizontal motion
mechanical translational

Conserving port associated with the horizontal motion. Connect to the **H** port of the Vehicle Body block.

Output

MF — Front axle friction coefficients
vector

Physical signal output port for the friction coefficients of the front axle.

The vector contains four elements for the Magic Formula friction model or two elements for the friction coefficients friction model. Connect to the **M** port of a front axle Tire (Friction Parameterized) or Tire (Magic Formula) block.

This port is visible only when the **Friction output** parameter is set to one of these values:

- **Physical signal Magic Formula coefficients**
- **Physical signal friction coefficients**

MR — Rear axle friction coefficients
vector

Physical signal output port for the friction coefficients of the rear axle.

The vector contains four elements for the Magic Formula friction model or two elements for the friction coefficients friction model. Connect to the **M** port of a rear axle Tire (Friction Parameterized) or Tire (Magic Formula) block.

Dependencies

This port is visible only when the **Friction** parameter **Friction output** is set to one of these values:

- **Physical signal Magic Formula coefficients**
- **Physical signal friction coefficients**

beta — Vehicle angle
scalar

Physical signal output port for the vehicle angle.

Parameters

Main

Horizontal distance from CG to front axle — Distance between CG and front axle
`1.4 m` (default) | positive scalar

Horizontal distance from the center of gravity (CG) to the front axle. Specify the same value as is specified for the **Horizontal distance from CG to front axle** parameter of the connected Vehicle Body block.

Horizontal distance from CG to rear axle — Distance between CG and rear axle
`1.6 m` (default) | positive scalar

Horizontal distance from the center of gravity (CG) to the rear axle. Specify the same value as is specified for the **Horizontal distance from CG to rear axle** parameter of the connected Vehicle Body block.

Horizontal distance for vertical profile — Vertical profile horizontal distance
`[-10, 0, 10] m` (default) | vector

Table lookup corresponding to the distance that the CG of the vehicle moves relative to the reference frame *x*-axis. For example the figure shows a horizontal profile of [-10, 0, 10]. Table lookup extrapolation is nearest.
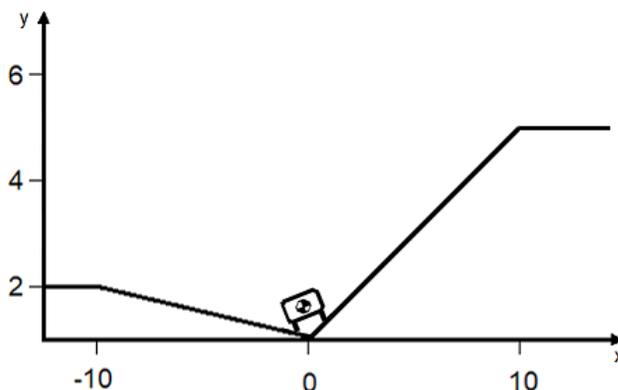
Vertical profile — Vertical profile
`[0, 0, .25]` m (default) | vector

Table lookup corresponding to the distance that the CG of the vehicle moves relative to the reference frame *y*-axis. For example the figure shows a vertical profile of [2, 0, 5]. Table lookup extrapolation is nearest.
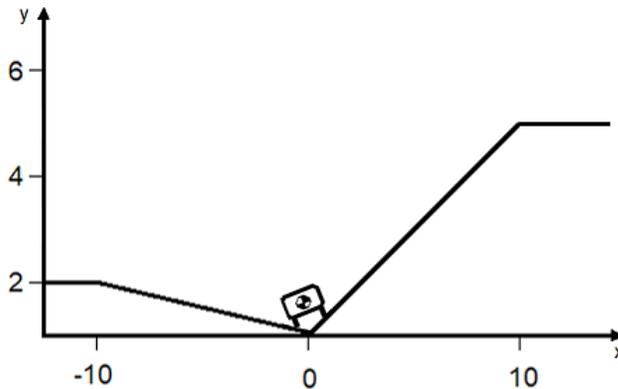
Friction

Friction output — Friction coefficient output type
None (default) | Physical signal Magic Formula coefficientsPhysical signal friction coefficients

Type of friction coefficient that the block outputs. Select `Physical signal Magic Formula coefficients` to vary friction for Tire (Magic Formula) blocks. Select `Physical signal friction coefficients` to vary friction for Tire (Friction Parameterized) blocks.

Dependencies

Setting this parameter to **Physical signal Magic Formula coefficients** or **Physical signal friction coefficients** makes the **MF** and **MR** ports and associated **Friction** parameters visible.

Horizontal distance for friction profile — Horizontal distance for friction profile
`[-10, 2, 2.1, 6, 6.1, 10]` m (default) | vector

Friction profile horizontal distance.

Dependencies

This parameter is visible only when the **Friction** parameter **Friction output** is set to one of these values:

- **Physical signal Magic Formula coefficients**
- **Physical signal friction coefficients**

Magic Formula coefficients for front axle — Front axle Magic Formula coefficients
`[10, 2, 1, 1; 10, 2, 1, 1; 8, 3, 1, 1; 8, 3, 1, 1; 9, 4, 1, 1; 9, 4, 1, 1]` (default) | matrix

Magic Formula coefficients specified as an *n*-by-*4* matrix, where *n* is the number of points in the vector specified by the **Horizontal distance for friction profile** parameter.

Dependencies

This parameter is visible only when the **Friction** parameter **Friction output** is set to **Physical signal Magic Formula coefficients**.

Magic Formula coefficients for rear axle — Rear axle Magic Formula coefficients
[10, 2, 1, 1; 10, 2, 1, 1; 8, 3, 1, 1; 8, 3, 1, 1; 9, 4, 1, 1; 9, 4, 1, 1] (default) | matrix

Magic Formula coefficients specified as an n-by-4 matrix, where n is the number of points in the vector specified by the **Horizontal distance for friction profile** parameter.

Dependencies

This parameter is visible only when the **Friction** parameter **Friction output** is set to **Physical signal Magic Formula coefficients**.

Static and kinetic friction coefficients for front axle — Front axle friction coefficients
[.9, .7; .9, .7; .7, .5; .7, .5; .8, .6; .8, .6] (default) | matrix

Front axle static and kinetic friction coefficients specified as an n-by-2 matrix, where n is the number of points in the vector specified by the **Horizontal distance for friction profile** parameter.

Dependencies

This parameter is visible only when the **Friction** parameter **Friction output** is set to **Physical signal friction coefficients**.

Static and kinetic friction coefficients for rear axle — Rear axle friction coefficients
[.9, .7; .9, .7; .7, .5; .7, .5; .8, .6; .8, .6] (default) | matrix

Rear axle static and kinetic friction coefficients specified as an n-by-2 matrix, where n is the number of points in the vector specified by the **Horizontal distance for friction profile** parameter.

Dependencies

This parameter is visible only when the **Friction** parameter **Friction output** is set to **Physical signal friction coefficients**.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Tire (Friction Parameterized) | Tire-Road Interaction (Magic Formula) | Tire (Magic Formula) | Vehicle Body

Topics
"Model a Road Profile with Varying Elevation and Friction"

Introduced in R2019a

Rod

Axially flexible bar or cable
Library: Simscape / Driveline / Couplings & Drives

Description

The Rod block represents an axially flexible bar or cable in tension or compression.

Tension

Compression
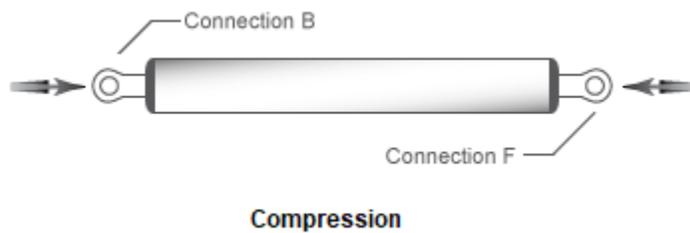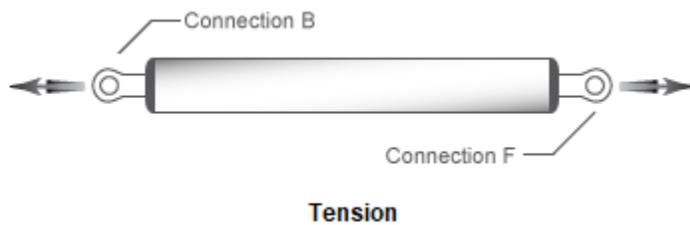
Model

To represent the bar or cable, the block uses a lumped parameter model. The model is composed of N +1 lumped masses that are connected in series by N sets of parallel-connected spring and damper circuits. The spring represents elasticity. The damper represents material damping.
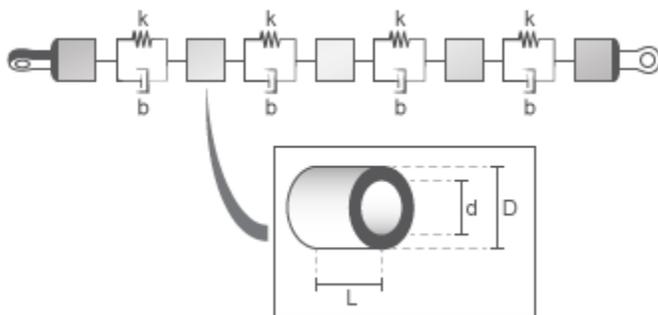
A single flexible element model exhibits an eigenfrequency that is close to the first eigenfrequency of the distributed parameter model. For more accurate analysis, select 2, 4, 8, or more flexible elements.

The equivalent physical network contains *N* spring and damper circuits and *N+1* mass blocks. The total mass of the rod is distributed evenly over the mass blocks. The stiffness of the spring in each spring and damper circuit is equal to the *N* times the stiffness of the rod material.
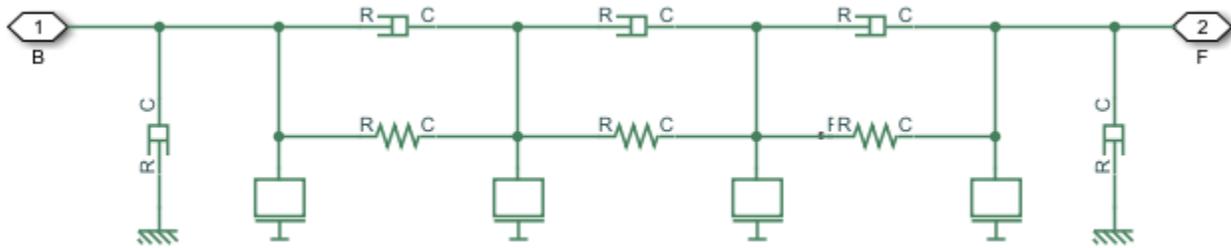
Equations

The defining equations for the model are:

$$\eta = 2\zeta\sqrt{\frac{K \cdot m}{2}}$$

$$A = \frac{\pi(D^2 - d^2)}{4}$$

$$K = \frac{E \cdot A}{L}$$

$$m = A \cdot L \cdot \rho$$

Where:

- η is the damping of the rod.
- ς is the damping ratio of the rod material.
- K is the stiffness of the rod.
- A is the cross-sectional area of the rod.
- D is the outer diameter of the rod.
- d is the inner diameter of the rod, where

  - $d = 0$ for a solid rod.
  - $d > 0$ for an annular, that is, hollow, rod.
- E is Young's modulus, that is, the modulus of elasticity, of the rod material.
- L is the length of the rod.
- m is the mass of the rod.

- ρ is the density of the rod material.

Assumptions and Limitations

- The rod does not buckle, in the case of a bar, or go slack, in the case of a cable, in tension.
- The rod has a constant cross-section along its length.
- The distributed parameter model is approximated as a finite number of flexible elements, *N*.

Ports

Conserving

B — Base
translational mechanical

Translational conserving port associated with rod base or input.

F — Follower
translational mechanical

Translational conserving port associated with rod follower or output.

Parameters

Rod

The table shows how the visibility of some **Rod** parameters depends on choices that you make for other **Rod** parameters. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Rod Parameter Dependencies Table

Rod		
Parameterization		
By stiffness and inertia	By material and geometry	
Stiffness	**Geometry**	
Mass	Solid	Annular
	Length	
	Outer diameter	
	-	**Inner diameter**
	Density	
	Young's modulus	
Damping ratio		
Number of flexible elements		

Parameterization — Parameterization method
By stiffness and inertia (default) | By material and geometry

Parameterization method.

Each **Parameterization** option exposes other parameters. For more information, see Rod Parameter Dependencies Table.

Stiffness — Material stiffness
5e8 N/m (default) | non-negative scalar

Material stiffness.

Setting the **Parameterization** parameter to By stiffness and inertia exposes this parameter. For more information, see Rod Parameter Dependencies Table.

Mass — Rod mass
20 kg (default) | positive scalar

Rod mass.

Dependencies

Setting the **Parameterization** parameter to By stiffness and inertia exposes this parameter. For more information, see Rod Parameter Dependencies Table.

Geometry — Rod cross-section
Solid (default) | Annular

Rod cross-sectional geometry.

Dependencies

Setting the **Parameterization** parameter to By material and geometry exposes this parameter. Each **Geometry** option exposes other parameters. For more information, see Rod Parameter Dependencies Table.

Length — Rod length
1 m (default) | positive scalar

Rod length.

Dependencies

Setting the **Parameterization** parameter to By material and geometry exposes this parameter. For more information, see Rod Parameter Dependencies Table.

Outer diameter — Rod outer diameter
0.075 m (default) | positive scalar

Rod outer diameter.

Dependencies

Setting the **Parameterization** parameter to By material and geometry exposes this parameter. For more information, see Rod Parameter Dependencies Table.

Inner diameter — Rod inner diameter
0.050 m (default) | positive scalar

Rod inner diameter. If the rod is solid, specify 0.

Dependencies

Setting the **Parameterization** parameter to By material and geometry and the **Geometry** parameter to Annular exposes this parameter. For more information, see Rod Parameter Dependencies Table.

Density — Material density
7.8e3 kg/m^3 (default) | positive scalar

Material density.

Dependencies

Setting the **Parameterization** parameter to By material and geometry exposes this parameter. For more information, see Rod Parameter Dependencies Table.

Young's modulus — Young's modulus
200e9 Pa (default) | positive scalar

Young's modulus for the rod material.

Dependencies

Setting the **Parameterization** parameter to By material and geometry exposes this parameter. For more information, see Rod Parameter Dependencies Table.

Damping ratio — Material damping ratio
0.01 (default) | non-negative scalar

Material damping ratio.

Number of flexible elements — *N*
1 (default) | positive integer scalar

Number of flexible elements, N, for the approximation.

A larger number of flexible elements, N, increases the accuracy of the model but reduces simulation performance, that is, simulation speed. The single-element model (N=1) exhibits an eigenfrequency that is close to the first eigenfrequency of the continuous, distributed parameter model.

If accuracy is more important than performance, select 2, 4, 8, or more flexible elements. For example, the four lowest eigenfrequencies are represented with an accuracy of 0.1, 1.9, 1.6, and 5.3 percent, respectively, by a 16-element model.

Viscous Losses

Viscous friction coefficients at base (B) and follower (F) — Viscous friction coefficients
[0, 0] N/(m/s) (default) | non-negative vector

Viscous friction coefficients at base port, **B**, and the follower port, **F**.

Initial Conditions

Initial deflection — Initial deflection
0 m (default) | non-negative scalar

Deflection of the rod at the start of simulation.

A positive initial deflection results in a positive translation of the base, **B**, end of the rod relative to the follower, **F**, end of the rod.

Initial velocity — Initial velocity
0 m/s (default) | non-negative scalar

Longitudinal velocity of the base, **B**, end of the rod relative to the follower, **F**, end of the rod at the start of simulation.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Flexible Shaft | Mass | Translational Damper | Translational Spring

Introduced in R2018b

Rolling Resistance

Model rolling resistance
Library: Simscape / Driveline / Tires & Vehicles / Tire Subcomponents

Description

The **Rolling Resistance** block models the resistance force that acts on the wheel hub due to the rolling resistance at the road-wheel contact surface. The block can use a constant resistance coefficient or the pressure and velocity dependence of the SAE J2452 standard. The resistance force is zero when the normal force acting at the wheel-road surface is less than or equal to zero.

Constant Resistance Coefficient Model

When you set **Resistance model** to `Constant coefficient`, the rolling resistance is directly proportional to the resistance coefficient $F = N\mu$, where parameters represent the following quantities:

- F is the rolling resistance force
- N is the normal Force
- μ is the rolling resistance coefficient

The rolling resistance coefficient has a hyperbolic form that eliminates discontinuity at $v_{hub} = 0$:

$$\mu = \mu_0 \cdot \tanh\left(\frac{4 \cdot v_{hub}}{v_{threshold}}\right),$$

where parameters represent the following quantities:

- μ_0 is the asymptotic rolling resistance coefficient
- v_{hub} is the hub velocity
- $v_{threshold}$ is the threshold velocity

Pressure and Velocity Dependent Model

When you set **Resistance model** to `Pressure and velocity dependent`, the block uses the formula:

$$F = \left(\frac{P}{P_0}\right)^{\alpha}\left(\frac{N}{N_0}\right)^{\beta} N_0 \cdot (A + B|v_{hub}| + Cv_{hub}2),$$

where parameters represent the following quantities:

- P is the tire pressure

- v_{hub} is the hub velocity
- α, β, A, B, C are the approximating coefficients
- P_0 is defined as 1 Pascal (Pa)
- N_0 is defined as 1 Newton (N)

In this equation, the parameters P_0 and N_0 remove the physical units from each exponential expression base.

Ports

Input

N — Normal force, N
physical signal

Physical signal input port that represents the normal force. The positive normal force is down.

Conserving

H — Wheel hub, m/s
mechanical translational

Mechanical translational conserving port that represents the wheel hub.

Parameters

Resistance model — Rolling resistance computational modeling method
`Constant coefficient` (default) | `Pressure and velocity dependent`

Method to compute the rolling resistance on a wheel hub. The parameter has two options:

- `Constant coefficient`
- `Pressure and velocity dependent`

Constant coefficient — Resistance coefficient for rolling tire
`0.015` (default) | positive scalar

Constant coefficient value to compute rolling resistance.

Dependencies

To enable this parameter, set **Resistance model** to `Constant coefficient`.

Tire pressure — Inflation pressure of rolling tire
`250e3 Pa` (default) | positive scalar

Constant inflation pressure of the rolling tire.

Dependencies

To enable this parameter, set **Resistance model** to `Pressure and velocity dependent`.

Alpha — Approximating coefficient α
`-0.003` (default) | scalar

SAE J2452 coefficient for pressure and velocity parameterization. You determine this value using empirical data in accordance with SAE J2452.

Dependencies

To enable this parameter, set **Resistance model** to `Pressure and velocity dependent`.

Beta — Approximating coefficient β
`0.97` (default) | scalar

SAE J2452 coefficient for pressure and velocity parameterization. You determine this value using empirical data in accordance with SAE J2452.

Dependencies

To enable this parameter, set **Resistance model** to `Pressure and velocity dependent`.

Coefficient A — Approximating coefficient A
`84e-4` (default) | positive scalar

SAE J2452 coefficient for pressure and velocity parameterization. You determine this value using empirical data in accordance with SAE J2452.

Dependencies

To enable this parameter, set **Resistance model** to `Pressure and velocity dependent`.

Coefficient B — Approximating coefficient B
`6.2e-4 s/m` (default) | nonnegative scalar

SAE J2452 coefficient for pressure and velocity parameterization. You determine this value using empirical data in accordance with SAE J2452.

Dependencies

To enable this parameter, set **Resistance model** to `Pressure and velocity dependent`.

Coefficient C — Approximating coefficient C
`1.6e-4 s^2/m^2` (default) | nonnegative scalar

SAE J2452 coefficient for pressure and velocity parameterization. You determine this value using empirical data in accordance with SAE J2452.

Dependencies

To enable this parameter, set **Resistance model** to `Pressure and velocity dependent`.

Velocity threshold — Velocity when full rolling resistance force is transmitted to rolling hub
`0.001 m/s` (default) | positive scalar

Minimum velocity that attains the maximum amount of rolling resistance.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Tire-Road Interaction (Magic Formula) | Tire (Magic Formula) | Tire (Friction Parameterized)

Introduced in R2012a

Rope

Physical connection that enables tension transfer
Library: Simscape / Driveline / Couplings & Drives

Description

The Rope block represents an element that transfers tension between two contact points. You can treat it as an arbitrary tension transfer device to simulate a variety of situations, including pulley networks, cable drives, and serpentine belts. You can choose whether the block exhibits slack upon loss of tension. The block has a base port, **B**, and a follower port, **F**. The rope stretches in tension when the base port leads the follower port.

The block considers only the longitudinal translational motion and force of the rope ends. For the ideal case, the block is equivalent to the spring-damper or spring-mass-damper systems illustrated in the figures, depending on whether you choose to model the mass. These figures do not necessarily apply when you simulate slack.
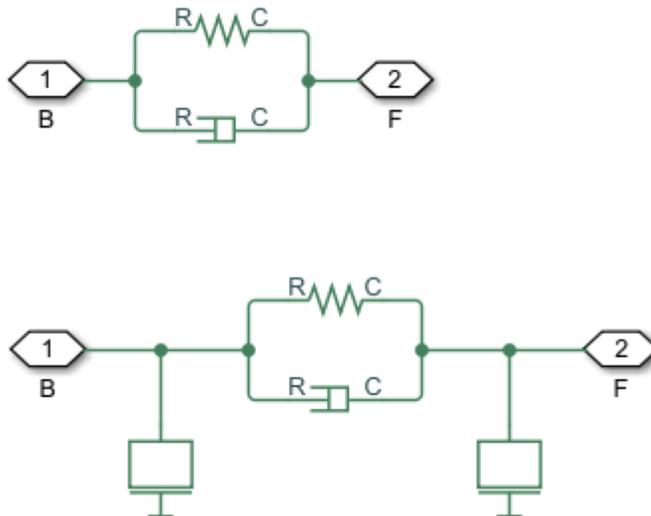
Tip Confirm that your model performs correctly with **Rope model** set to `Ideal - No tension`. While in this mode, verify that the **B** and **F** ports are oriented properly by viewing the tension plots of each rope in the **Simscape Results Viewer** and looking for unexpected negative values.

You simulate a rope with arbitrary physical properties by adjusting the block parameters. The Rope block can act as a belt, cable, or any other device that has tensile strength and transmits tension between two contact points. A contact point can be a pulley, a drum, or an ideal source.

The block is useful for modeling stationary or travelling pulley networks. You can drive the Rope block with sources like the Ideal Translational Velocity Source block, or a Rope Drum block attached to an Ideal Force Source block.

For greater fidelity or numerical stability, you can choose to model the effect of mass on the system. When you set **Model mass** to `On`, the block distributes half of the total mass to either end of the rope. Adding even a small amount of mass can improve the initialization of your model. You can also specify the initial conditions of the rope. If you choose to simulate slack, the block will stop applying stiffness and damping when the rope is no longer in tension.

Equations

The Rope block characterizes the external load at the base port as

$$F_B(x_s(t), t) = \frac{m}{2}\ddot{x}_B(t) + D(x_s(t))\dot{x}_s(t) + K(x_s(t))x_s(t),$$

and at the follower port,

$$F_F(x_s(t), t) = \frac{m}{2}\ddot{x}_F(t) - D(x_s(t))\dot{x}_s(t) - K(x_s(t))x_s(t),$$

where $x_s(t) = x_B(t) - x_F(t)$. The forces vary as functions of time and the stretch deformation, x_s. Here, K and B are functions of x_s, such that $K(x_s) = K$ when there is no slack or you are not simulating slack. Otherwise, $K(x_s) = 0$ when slack occurs. That is, when $x_s < 0$. The same logic applies for B. Ignore the mass terms unless you are modeling mass.

The variables have the following meanings:

- F is the tension force, where $F = F_B = -F_F$.
- K is the spring stiffness coefficient. $K(x_s) = 0$ when you set **Rope model** to `Model rope slack` and $x_s < 0$.
- D is the damping coefficient.
- m is the total mass. This is equivalent to the **Mass** parameter. The block distributes half of the total mass to each end.
- x_B is the position of the base node.
- x_F is the position of the follower node.
- x_s is the deformation due to stretching. This is equivalent to the **Stretch** variable.
- $\dot{x}_s$ is the stretch velocity.

Variables

Use the **Variables** tab to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Ports

Conserving

B — Base end of rope
mechanical translational

Mechanical translational conserving port associated with the leading end of the rope.

F — Follower end of rope
mechanical translational

Mechanical translational conserving port associated with the following end of the rope.

Parameters

`Rope model` — Slack behavior
`Ideal - no slack` (default) | `Model rope slack`

Option to model slack. By default, the rope will transfer force while in tension or compression. When you select `Model rope slack`, the block stops transferring spring force when it is not in tension. The Rope block in slack is equivalent to a Translational Hard Stop block with **Upper bound** set to `0` and **Lower bound** set to `-inf`. For more information, see Translational Hard Stop.

`Model mass` — Mass parameterization option
`Off` (default) | `On`

Option to parameterize mass. Setting this to `On` will cause the block to consider the mass of the rope segment that it represents. Simulating mass can help initialize some models but can also be more computationally costly.

`Mass` — Rope segment mass
`1 kg` (default) | positive scalar

Mass of the rope segment that the block represents. The block distributes half of the total mass to each end of the rope.

Dependencies

To enable this parameter, set **Model mass** to `On` or set **Rope model** to `Model slack`.

`Stiffness` — Resistance to deformation
`1000 N/m` (default) | positive scalar

Effective translational spring stiffness of the rope segment.

`Damping` — Tendency to dissipate energy in the system
`100 N/(m/s)` (default) | positive scalar

Effective translational damping of the rope segment. When you set **Rope model** to `Model slack` and $x_s < 0$, the block ignores damping.

`Slack model` — Slack transition behavior
`Stiffness and damping applied smoothly through transition region, damped rebound` (default) | ...

Stiffness and rebound options for the slack model. The slack model is equivalent to a Translational Hard Stop block. You can choose from the following options:

- `Stiffness and damping applied smoothly through transition region, damped rebound`
- `Full stiffness and damping applied at bounds, undamped rebound`
- `Full stiffness and damping applied at bounds, damped rebound`

Dependencies

To enable this parameter, set **Rope model** to `Model rope slack`.

Transition region — Region of partial slack effects
`0.1` mm (default) | positive scalar

Distance from full compression or full extension where the effects of stiffness and damping are partially applied. When you set **Slack model** to `Stiffness and damping applied smoothly through transition region, damped rebound`, the block will smoothly transition the onset of stiffness and damping as the spring approaches full extension or full compression.

Dependencies

To enable this parameter, set **Rope model** to `Model rope slack`.

Maximum tension — Tension limit model
`No maximum tension` (default) | `Specify maximum tension`

Option to constrain the maximum tensile force in the rope.

- `No maximum tension` — Tension can be arbitrarily large during simulation.
- `Specify maximum tension` — Tension must remain lower than a maximum value. If the tension exceeds this value, the simulation generates an error and stops.

Rope maximum tension — Maximum allowable tension force in rope segment
`1e5` N (default) | positive scalar

Tension value error target. When the tension in the rope segment exceeds this parameter, the block will generate an error.

Dependencies

To enable this parameter, set **Maximum tension** to `Specify maximum tension`.

Tension warning — Slack threshold notification
`Do not check tension` (default) | `Warn if rope loses tension`

Generate a warning when the rope segment is no longer in tension. This setting can help you discover if your system experiences instances of slack.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Belt Pulley | Rope Drum | Ideal Force Source | Ideal Translational Velocity Source | Translational Hard Stop

Introduced in R2021a

Rope Drum

Power transmission system with tightly wound rope around cylindrical drum
**Library:** Simscape / Driveline / Couplings & Drives
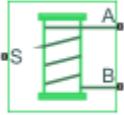
Description

The Rope Drum block represents a power transmission system where a rope is tightly wound around a cylindrical drum at a sufficient tension level to prevent slipping. You can configure the rope so that the ends point in the same or opposite directions. You can set the direction that the ends move with the **Rope windup type** parameter:

- If the rope ends point in the same direction, they translate in opposite directions.
- If the rope ends point in opposite directions, they translate in the same direction.

The equations refer to rope ends A and B, as illustrated by the figure.
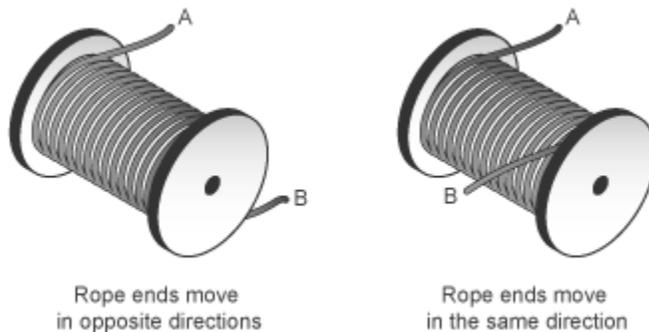
During normal operation, a driving torque causes the cylindrical drum to rotate about its axis of symmetry. The rotating drum transmits tensile force to the rope ends, which translate with respect to the drum centerline. The relative direction of translation between the two rope ends depends on the setting of the **Rope windup type** parameter. However, a positive drum rotation always corresponds to a positive translation at port **A**.

The block assumes that each rope end remains taut during simulation because slack rope ends do not transmit force. You can set the block to warn you if the rope loses tension.

You can optionally set parameters for:

- Drum inertia
- Viscous friction at the drum bearing

The net torque acting on the cylinder satisfies the force balance equation

$$T = F_B \cdot R \cdot \delta - F_A \cdot R + \mu \cdot \omega,$$

where:

- T is the net torque acting on the drum.
- F_A and F_B are the external forces pulling on rope ends A and B.
- R is the drum radius.
- μ is the viscous friction coefficient at the drum bearings.
- ω is the drum angular velocity.
- δ is the rope windup type according to the parameter settings:

    - When you set **Rope windup type** to `Ends move in the same direction`, $\delta = -1$.
    - When you set **Rope windup type** to `Ends move in the opposite directions`, $\delta = +1$.

The figure shows the equation variables.
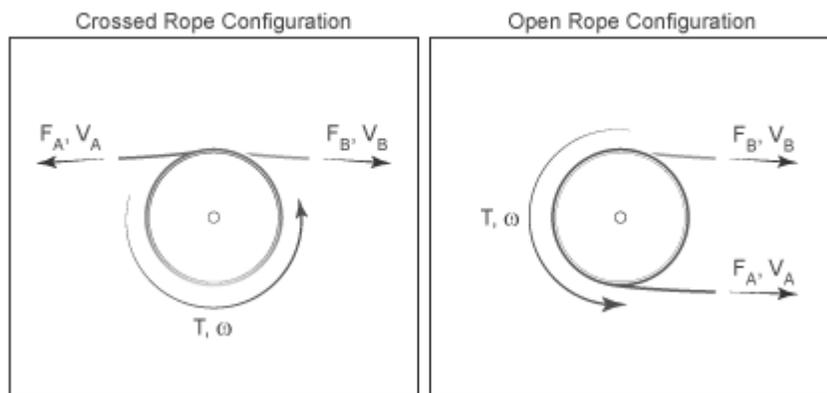
The translational velocities of the two rope ends are functions of the drum radius and angular velocity. Each translational velocity is equal to the tangential velocity of a point at the drum periphery according to the expressions

$$V_A = -\omega \cdot R$$
$$V_B = +\omega \cdot R$$

where:

- V_A is the translational velocity of rope end A.
- V_B is the translational velocity of rope end B.

Assumptions and Limitations

- The block ignores slip at the rope-drum contact surface.
- The block ignores rope compliance.

Ports

Conserving

S — Drum shaft
mechanical rotational

Mechanical rotational conserving port associated with the drum shaft.

A — Rope end A
mechanical translational

Mechanical translational conserving port associated with rope end A.

B — Rope end B
mechanical translational

Mechanical translational conserving port associated with rope end B.

Parameters

Rope Winding

Drum radius — Size of drum
`0.10` m (default) | positive scalar

Distance between the drum center and its periphery. The drum periphery coincides with the contact surface between the drum and the rope.

Rope windup type — Direction of B rope end with respect to A rope end
`Ends move in same direction` (default) | `Ends move in opposite direction`

Relative direction of the translation motion of the two rope ends, A and B, where B is the following end.

Tension warning — Optional lost tension warning
`Do not check tension` (default) | `Warn if either end loses tension`

Option to generate a warning when a rope end becomes slack. A rope end becomes slack when the net tensile forces responsible for keeping it taut fall below zero. Because slack cables do not transmit force, the simulation results may not be accurate when the rope loses tension.

Drum Dynamics

Bearing viscous friction coefficient — Viscous friction for drum rotation
`0.001` N*m/(rad/s) (default) | nonnegative scalar

Linear damping coefficient in effect at the drum bearing. This coefficient determines the power losses due to viscous friction at a given drum angular velocity.

Inertia — Resistance to angular acceleration
`No inertia` (default) | `Specify inertia and initial velocity`

Option to account for drum inertia.

- `No inertia` — Select this setting if drum inertia has a negligible impact on driveline dynamics. This setting sets drum inertia to zero.
- `Specify inertia and initial velocity` — Select this setting if drum inertia has a significant impact on driveline dynamics.

Drum inertia — Drum tendency to remain in motion
`0.01` kg*m^2 (default) | positive scalar

Moment of inertia of the drum about its rotation axis. In the simple case of a solid cylindrical drum, the moment of inertia is

$$I = \frac{1}{2}MR^2,$$

where M is the drum mass and R is the drum radius.

Dependencies

To enable this parameter, set **Inertia** to `Specify inertia and initial velocity`.

Drum initial velocity — Initial drum state
`0 rad/s` (default) | scalar

Angular velocity of the drum at the start of the simulation.

Dependencies

To enable this parameter, set **Inertia** to `Specify inertia and initial velocity`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Rope | Wheel and Axle

Topics
"Basic Motion, Torque, and Force Modeling"

Introduced in R2012a

Rotational Damper

Faultable linear rotational damper

Library: Simscape / Driveline / Couplings & Drives / Springs & Dampers

Description

The Rotational Damper block models a faultable linear rotational viscous damper. From the start of the simulation, the block uses the nominal damping coefficient. If you enable faults for the block, the damping coefficient changes in response to one or both of these triggers:

- Simulation time — A fault occurs at a specified time.
- Simulation behavior — If the magnitude of the rotational acceleration exceeds a specified maximum value, the block increments the number of shocks by one. A fault occurs if the number of shocks exceeds a specified maximum number of shocks.

If a fault trigger event occurs, the block uses the faulted damping coefficient for the remainder of the simulation. You can program the block to issue a fault report as a warning or error message.

The defining equations are

$$T = D\omega$$

and

$$\omega = \omega_R - \omega_C,$$

where

- T is the torque transmitted through the damper.
- D is the damping (viscous friction) coefficient.
- ω is the relative velocity.
- ω_R and ω_C are the absolute velocities of terminals R and C, respectively.

The positive block direction is from port R to port C. Therefore, the torque is positive if it acts in the direction from R to C.

Thermal Modelling

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Thermal port** to Model.

Enabling thermal modelling exposes the **Thermal mass** parameter.

Variables

Use the **Variables** tab to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Ports

Conserving

R — Rod interface
mechanical rotational

Mechanical rotational conserving port associated with the rod interface.

C — Case interface
mechanical rotational

Mechanical rotational conserving port associated with the case interface.

H — Heat flow
thermal

Thermal conserving port associated with heat flow. Heat flow affects damper temperature, and therefore, power transmission efficiency.

Dependencies

To enable this port, set **Thermal port** to Model.

Parameters

Main

Damping coefficient — Damping coefficient for the no-fault condition
1e-3 N*m/(rad/s) (default)

The oscillation-reducing torque due to viscous friction that the block applies if no trigger event occurs. Specify a value that is greater than or equal to zero.

Faults

Enable faults — Fault mode capability
Off (default) | On

To enable the block to use a different damping coefficient based on the simulation time or when the magnitude of the rotational acceleration exceeds the specified number of shocks, select On.

Dependencies

Selecting On makes these parameters visible:

- **Faulted damping coefficient**
- **Enable temporal fault trigger**
- **Enable behavioral fault trigger**

- **Reporting when fault occurs**

Faulted damping coefficient — Damping coefficient for the fault condition
100 N*m/(rad/s) (default)

The oscillation-reducing torque due to viscous friction that the block applies if a trigger event occurs. Specify a value that is greater than or equal to zero.

Dependencies

This parameter is only visible if you select On for the **Enable faults** parameter.

Enable temporal fault trigger — Fault trigger capability
Off (default) | On

To enable the block to use the faulted damping coefficient starting at a specified time in the simulation, select On.

Dependencies

This parameter is only visible if you select On for the **Enable faults** parameter.

Selecting On makes the **Simulation time for fault event** parameter visible.

Simulation time for fault event — Temporal fault trigger
5 s (default)

Specify the time during the simulation that you want the block to start using the faulted damping coefficient.

Dependencies

This parameter is only visible if you select On for the **Enable faults** parameter and On for the **Enable temporal fault trigger** parameter.

Enable behavioral fault trigger — Fault trigger capability
Off (default) | On

To use the faulted damping coefficient when the number of sustained shocks exceeds a specified maximum number of shocks, select On.

Dependencies

This parameter is only visible if you select On for the **Enable faults** parameter.

Selecting On makes these parameters visible:

- **Maximum permissible acceleration**
- **Maximum number of shocks**

Maximum permissible acceleration — Behavioral fault trigger
100 rad/s^2 (default)

The maximum value that the magnitude of the rotational acceleration can reach without causing the block to increment the number of sustained shocks.

Dependencies

This parameter is only visible if you select On for the **Enable faults** parameter and On for the **Enable behavioral fault trigger** parameter.

Maximum number of shocks — Behavioral fault trigger
1 (default)

The maximum number of shocks that can be sustained without triggering the block to use the faulted damping coefficient.

Dependencies

This parameter is only visible if you select On for the **Enable faults** parameter and On for the **Enable behavioral fault trigger** parameter.

Reporting when fault occurs — Fault reporting
None (default) | Warning | Error

Select your reporting preference for the fault condition.

Dependencies

This parameter is only visible if you select On for the **Enable faults** parameter.

Thermal port

Thermal port — Thermal modelling
Omit (default) | Model

Option to model effects of heat flow and temperature change.

Thermal mass — Thermal mass
50 kJ/K (default) | postive scalar

Thermal energy required to change the component temperature by a unit increase. The thermal mass represents the ability of the damper to absorb and retain heat energy.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Nonlinear Rotational Damper | Nonlinear Translational Damper | Translational Damper | Variable Rotational Damper | Variable Translational Damper

Introduced in R2016b

Rotational Detent

Double-sided, spring-loaded rotational detent

Library

Simscape / Driveline / Brakes & Detents / Rotational

Description

The Rotational Detent block represents a double-sided, spring-loaded rotational detent.

The detent rod rotates over the detent case. The case has notches or magnets that create a net relative torque with the rod as the rod rotates. With the block, you can model rotational detents with peak torque and notch width, or with a table lookup. The model also includes viscous damping and kinetic friction between the rod and case. You can set the friction to zero.

Rotational Detent Model

As the rod rotates over the case, the relative rotation $\phi = \phi_R - \phi_C$ causes a torque τ to develop. Depending on your choice of parameterization, the torque model is defined by the peak torque and notch width, or by a table lookup specifying relative rotation versus torque.

Regardless of the parameterization, if you specify multiple detents, all detents have the same torque-relative rotation function.

Peak Torque and Notch Width

If you choose the peak torque and notch width parameterization, the block ensures that the torque-relative rotation curve provides a continuous torque and torque derivative over the detent region. The peak torques are halfway between the detent center and detent edge, as shown in the following figure.
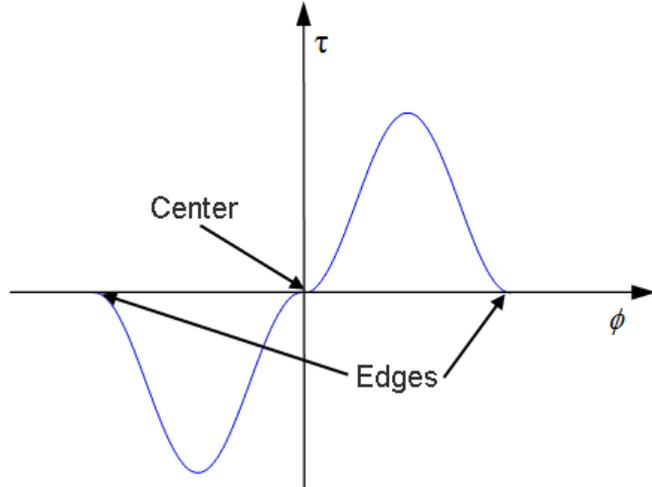
By Table Lookup

With the table lookup parameterization, you can create an arbitrary function relating torque to relative rotation. If you create such a function, consider the following best practices.

Energy Conservation

If you want to ensure that the detent conserves energy, the total integral of the torque-relative rotation curve (area under the curve) must be zero.

Simulation Performance

To stabilize simulation of the detent, avoid discontinuities in the torque-relative rotation function.

Limitations

- The model does not account for inertia. Add inertia terms externally to the R and C ports as required.
- If you use the peak torque-notch width or the table lookup parameterization, the kinetic friction is independent of the detent normal force.

Ports

R and **C** are rotational conserving ports associated with the rod and case, respectively.

Parameters

Detent Characteristics

Parameterization

    Select how to specify the detent characteristics. The default is `By peak torque and notch width`.

    - `By peak torque and notch width` — Specify detent characteristics by the peak torque and the notch width.

Peak Torque and Notch Width

Peak torque

The peak torque developed inside the detent. The default is `0.25`.

From the drop-down list, choose units. The default is newton-meters (`N*m`).

Notch width

The width of the region where the torque develops. The default is `5`.

From the drop-down list, choose units. The default is degrees (`deg`).

- By `table lookup` — Define the detent characteristics by one-dimensional table lookup based on the relative displacement between the slider and case. If you select this option, the panel changes from its default option.

Table Lookup

Relative rotation vector

Specify the vector of input relative rotations that correspond to the values in **Torque vector**. These two vectors must be of the same size. The angles must be specified in strictly increasing order. The default values are `[-5.0 -4.0 -3.0 -1.0 1.0 3.0 4.0 5.0]`.

From the drop-down list, choose units. The default is degrees (`deg`).

Torque vector

Specify the vector of input values for the developed torque that correspond to the values in **Relative rotation vector**. The default values are `[0 0 -0.25 -0.10 0.10 0.25 0 0]`.

From the drop-down list, choose units. The default is newton-meters (`N*m`).

Initial relative rotation

The relative angle of the rod and case when simulation starts. The default is `0`.

From the drop-down list, choose units. The default is degrees (`deg`).

Interpolation method

Select one of these interpolation methods for approximating the output value when the input value is between two consecutive grid points:

- `Linear` — Select this, default, option to get the best performance.
- `Smooth` — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

Spacing

Detent spacing

Select how to specify the detent spacing. The default is `Regularly spaced`.

- `Regularly spaced` — Specify detent spacing by regular intervals.

Regular spacing

Number of detents

Specify the number of identical, regularly spaced detents. Must be greater than or equal to 1. The default is 1.

Creates a number of detents evenly spaced from –180 to +180 degrees, with one detent center at 0 degrees. Non-integer values are rounded to the nearest integer.

- `By angle vector` — Specify detent spacing by custom intervals. If you select this option, the panel changes from its default option.

Custom intervals

Vector of detent centers

Specify the locations of the centers of identical detents as a vector of angles. These values must lie between –180 and +180 degrees. The minimum spacing between the detents must be greater than the **Notch width** parameter. The default values are `[-40 0 30 85]`.

From the drop-down list, choose units. The default is degrees (`deg`).

Friction

Viscous friction coefficient

Specify the viscous friction coefficient for the rod-case contact. Must be greater than or equal to 0. The default is `0.1`.

From the drop-down list, choose units. The default is newton-meters/(radians/second) (`N*m/(rad/s)`).

Friction to peak torque ratio

Specify the kinetic friction coefficient for the rod-case contact. Must be greater than or equal to 0. The default is `0.01`.

The kinetic friction is this ratio multiplied by the peak torque.

Friction velocity threshold

Specify the relative angular velocity required for peak kinetic friction in the detent. Must be greater than 0. The default is `10`.

From the drop-down list, choose units. The default is revolutions/minute (`rpm`).

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Loaded-Contact Rotational Friction | Translational Detent | Tire (Friction Parameterized)

Topics
"Clutches, Clutch-Like Elements, and Coulomb Friction"

Introduced in R2011b

Rotational Power Sensor

Measure average or instantaneous rotational power
Library: Simscape / Driveline / Sensors
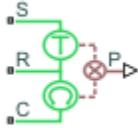
Description

The Rotational Power Sensor block measures the rotational power change between ports **R** and **C**. The physical signal output can be an instantaneous or averaged value depending on what you select for the **Measurement type** parameter.

When you set:

*   **Measurement type** to `Instantaneous power`, the instantaneous power is the product of the current angular velocity and torque. Use this setting to analyze peak power transmission and identify heavily loaded components.

*   **Measurement type** to `Average power over a specified period`, the time-averaged power is the integral of the instantaneous power over one sample period, divided by that period. For simulation times smaller than the sample period, the average power is zero. The power output is constant for each sample period.

*   **Measurement type** to `Average power over a vibration`, the average vibration power is the integral of the power over one vibration period, divided by that period. The vibration period is the time between two consecutive crossings from negative to positive velocity values. If the vibration has a variable frequency, the averaging period can change. For simulation times smaller than the vibration period, the average power is zero. The power output is constant for each vibration period.

*   **Measurement type** to `Average power over a revolution`, the average revolution power is the integral of the instantaneous power over one revolution period, divided by that period. The revolution period is defined as the time between two consecutive crossings of zero angular position. If the revolution has a variable frequency, the averaging period can change. For simulation times smaller than the revolution period, the average power is zero. The power output is constant for each revolution period.

Ports

Physical Signal

P — Change in rotational power, W
physical signal

Physical signal port that outputs the change in rotational power.

Conserving

C — Follower port
mechanical rotational

Mechanical rotational conserving port associated with the case.

R — Base port
mechanical rotational

Mechanical rotational conserving port associated with the load.

S — Source port
mechanical rotational

Mechanical rotational conserving port associated with driving source.

Parameters

Measurement type — Power measuring method
Instantaneous power (default) | Average power over a specified period | Average power over a vibration | Average power over a revolution

Option to measure the instantaneous power or an average power measurement.

- Instantaneous power — Output the rotational power measured at the current simulation time step.
- Average power over a specified period — Output the rotational power averaged over a fixed time period.
- Average power over a vibration — Output the rotational power averaged over one vibration period.
- Average power over a revolution — Output the rotational power averaged over one revolution period.

Averaging period — Timespan for power averaged over a specified period
1 s (default) | positive scalar

Time period that the block uses to compute the average power. For example, you can set this value to the cycle length of a rotating device.

Dependencies

To enable this parameter, set **Measurement type** to Average power over a specified period.

Maximum vibration frequency — Highest frequency to use when averaging power over a vibration
1000 Hz (default) | positive scalar

Highest vibration frequency to use to average the rotational power. Very high values can reduce simulation speed. Very low values can result in lower power measurements.

Dependencies

To enable this parameter, set **Measurement type** to Average power over a vibration.

Maximum rotational speed — Highest angular velocity to use when averaging power over revolution
1000 rpm (default) | positive scalar

Highest speed to use to average the rotational power. Very high values can reduce simulation speed. Very low values can result in lower power measurements.

Dependencies

To enable this parameter, set **Measurement type** to Average power over a revolution.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Translational Power Sensor

Introduced in R2013a

Rotational Velocity Noise Source

Produce zero-mean normally (Gaussian) distributed rotational velocity
Library: Simscape / Driveline / Sources

Description

The Rotational Velocity Source block generates zero-mean normally (Gaussian) distributed rotational velocity using the Random Number source in the Simscape Foundation library. The rotational velocity produced by the block depends on two factors:

- The noise power spectral density input to the block
- A random number seed that you control using the **Repeatability** and, as applicable, **Seed** parameters

The defining equation for the rotational velocity that the block generates is

$$v = \sqrt{PSD/2}\frac{N(0,1)}{\sqrt{h}},$$

where:

- v is the output rotational velocity.
- PSD is the single-sided noise power spectral density.
- N is a Gaussian random number with zero mean and standard deviation of one.
- h is the sampling interval.

Ports

Input

P — Single-sided noise power spectral density
physical signal

The power spectral density (PSD) of noise is the average noise power per unit of bandwidth, in $(\text{rad/s})^2/\text{Hz}$.

Conserving

B — Base interface
mechanical rotational

Mechanical rotational conserving port associated with the base interface.

F — Follower interface
mechanical rotational

Mechanical rotational conserving port associated with the follower interface.

Parameters

Sample time — Time step period and offset
1e-3 s (default) | [*step*, *offset*] s

The values of the time *step* period and the initial time *offset*. If you specify a scalar value for *step*, the block assumes an *offset* value of 0.To specify a nonzero value for the initial time offset, specify the parameter values using the vector [*step*, *offset*]. The *offset* value must be less than the *step* value and greater than or equal to zero.

Repeatability — Random number seed control
Not repeatable (default) | Repeatable | Specify seed

The random number seed is the number that initializes the random number generator. The seed is 0 or a positive integer. To control the random number seed, set this parameter to:

- Not repeatable — The seed changes every time you simulate your model. The block resets the random seed using the MATLAB random number generator command:

  seed = randi(2^32-1);
- Repeatable — The seed is the same random number at the start of every simulation. The block sets the value using the same MATLAB random number generator command used by the Not repeatable parameter.

  When you add a Rotational Velocity Noise Source block to your model from the Sources library, the block generates and stores a random value for the repeated seed. When you make a copy of the Rotational Velocity Noise Source block from an existing block in a model, the copy generates a new random value for the repeated seed.

- Specify seed — The seed is a number that you specify using the **Seed** parameter. The **Seed** parameter is only available when you choose Specify seed for the **Repeatability** parameter.

Seed — Random number generation seed value
0 (default)

The seed must be 0 or a positive integer. This parameter is only available when you select Specify seed for the **Repeatability** parameter.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Force Noise Source | Random Number | Sinusoidal Rotational Velocity Source | Torque Noise Source | Translational Velocity Noise Source

Functions
randi

Introduced in R2016b

Shock Absorber

Mechanism for damping translational vibrations

Library

Simscape / Driveline / Couplings & Drives

Description

The Shock Absorber block represents a spring-damper system commonly used to dampen vibration in mechanical systems. An internal force acts between ports R and C. This force is the sum of spring stiffness, damping, Coulomb friction, and hard-stop contributions. All force contributions are optional.
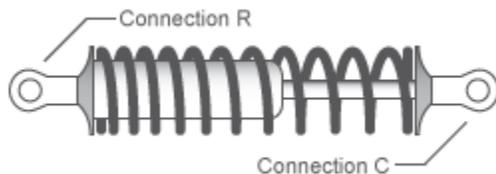
The Shock Absorber block uses the models of these blocks:

Block	Contribution	Library
Loaded-Contact Translational Friction	Coulomb friction	**Simscape / Driveline / Brakes & Detents / Translational**
Rotational Damper	Damping	**Simscape / Foundation Library / Mechanical / Translational Elements**
Rotational Spring	Spring	
Rotational Hard Stop	Hard stop	

Assumptions and Limitations

- Including hard-stop and Coulomb friction enhances model fidelity, but reduces simulation speed. For more information, see "Driveline Simulation Performance".

Ports

Conserving

C

Mechanical translational port associated with the slider that travels between stops installed on the case.

R

Mechanical translational port associated with the rod.

Parameters

Spring-Damper

Spring stiffness

Enter the value of the viscous spring stiffness constant, k. The default value is `1e4 N/m`. The value must be greater than or equal to zero.

Viscous friction coefficient

Enter the value of the viscous damping constant, b. The default value is `1e2 m/s`. The value must be greater than or equal to zero.

Coulomb friction force

Enter the value of the Coulomb friction force. Setting the value to `0` eliminates the Coulomb friction force contribution. This enhances simulation speed, making the model more suitable for HIL testing. The default value is `0 N`. The value must be greater than or equal to zero.

Ratio of static to kinetic friction

Enter the value of the static/kinetic friction ratio, that is F_s/F_k. The value must be greater than one. The default value is `1.1`. The value must be greater than or equal to one.

Velocity tolerance

Enter the value of the relative velocity below which ports R and C lock and translate together. The default value is `0.001 m/s`. The value must be greater than zero.

Hard Stops

Hard stop

Include or exclude hard-stop force by selecting one of these options:

- `No hard stops — Suitable for HIL simulation` — To enhance simulation speed by excluding the hard-stop force contribution, select this default option.
- `Compliant hard stops` — To enhance model fidelity by including the hard-stop force contribution, select this option. Selecting this option enables other parameters.

Upper bound

Upper hard-stop position, *UB*. Positive displacement beyond the upper bound activates the hard-stop contact force. The default value is `0.1 m`. The value must be greater than the lower bound.

Selecting `Compliant hard stops` for the **Hard stop** parameter enables this parameter.

Lower bound

Lower hard-stop position, *LB*. Negative displacement beyond the lower bound activates the hard-stop contact force. The default value is `-0.1 m`. The value must be smaller than the upper bound.

Selecting `Compliant hard stops` for the **Hard stop** parameter enables this parameter.

Contact stiffness

Enter the value of the hard-stop stiffness constant, k_{HS}. The default value is `1e6 N/m`. The value must be greater than or equal to zero.

Selecting `Compliant hard stops` for the **Hard stop** parameter enables this parameter.

Contact damping

Enter the value of the hard-stop damping constant, b_{HS}. This parameter specifies dissipating property of colliding bodies. The greater the value of the parameter, the more energy dissipates during an interaction. The default value is `1e4 N/(m/s)`. The value must be greater than zero.

Selecting `Compliant hard stops` for the **Hard stop** parameter enables this parameter.

Hard stop model

Select the hard-stop model:

- `Stiffness and damping applied smoothly through transition region, damped rebound` — Specify a transition region, in which the force is scaled from zero. At the end of the transition region, the full stiffness and damping are applied. This model has damping applied on the rebound, but it is limited to the value of the stiffness force. In this sense, damping can reduce or eliminate the force provided by the stiffness, but never exceed it. All equations are smooth and produce no zero crossings.

  Selecting this option enables the `Transition region` parameter.

- `Full stiffness and damping applied at bounds, undamped rebound` — This model has full stiffness and damping applied with impact at upper and lower bounds, with no damping on the rebound. Equations produce no zero crossings when velocity changes sign, but there is a position-based zero crossing at the bounds. Having no damping on rebound helps to push the slider past this position quickly. This model has nonlinear equations.

- `Full stiffness and damping applied at bounds, damped rebound` — This model has full stiffness and damping applied with impact at upper and lower bounds, with damping applied on the rebound as well. Equations are switched linear, but produce position-based zero crossings. Use this hard-stop model if `simscape.findNonlinearBlocks` indicates that this is the block that prevents the whole network from being switched linear.

Selecting `Compliant hard stops` for the **Hard stop** parameter enables this parameter.

Transition region

Region where the force is ramped up from zero to the full value. At the end of the transition region, the full stiffness and damping are applied. The default value is `0.1` mm.

Selecting `Stiffness and damping applied smoothly through transition region, damped rebound` for the **Hard stop model** enables this parameter.

Initial Conditions

Initial deformation

Enter the initial value of the spring deformation. The default value is `0` m.

Real-Time Simulation

Hardware-in-the-Loop Simulation

For optimal simulation performance, use the **Hard Stops > Hard stop** parameter default setting, `No hard stops - Suitable for HIL simulation`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Nonlinear Translational Spring | Rotational Hard Stop | Variable Translational Spring | Nonlinear Translational Damper | Variable Translational Damper

Topics
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2013a

Simple Gear

Simple gear of base and follower wheels with adjustable gear ratio, friction losses, and triggered faults

Library: Simscape / Driveline / Gears
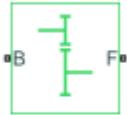
Description

The Simple Gear block represents a gearbox that constrains the connected driveline axes of the base gear, *B*, and the follower gear, *F*, to corotate with a fixed ratio that you specify. You choose whether the follower axis rotates in the same or opposite direction as the base axis. If they rotate in the same direction, the angular velocity of the follower, ω_F, and the angular velocity of the base, ω_B, have the same sign. If they rotate in opposite directions, ω_F and ω_B have opposite signs. You can easily add and remove backlash, faults, and thermal effects.

Ideal Gear Constraint and Gear Ratio

The kinematic constraint that the Simple Gear block imposes on the two connected axes is

$$r_F \omega_F = r_B \omega_B$$

where:

- r_F is the radius of the follower gear.
- ω_F is the angular velocity of the follower gear.
- r_B is the radius of the base gear.
- ω_B is the angular velocity of the base gear.

The follower-base gear ratio is

$$g_{FB} = \frac{r_F}{r_B} = \frac{N_F}{N_B}$$

where:

- N_B is the number of teeth in the base gear.
- N_{BF} is the number of teeth in the follower gear.

Reducing the two degrees of freedom to one independent degree of freedom yields the torque transfer equation

$$g_{FB}\tau_B + \tau_F - \tau_{loss} = 0$$

where:

- τ_B is the input torque.
- τ_F is the output torque.
- τ_{loss} is the torque loss due to friction.

For the ideal case, $\tau_{loss} = 0$.

Nonideal Gear Constraint and Losses

In the nonideal case, $\tau_{loss} \neq 0$. For general considerations on nonideal gear modeling, see "Model Gears with Losses".

In a nonideal gear pair (B,F), the angular velocity, gear radii, and gear teeth constraints are unchanged. But the transferred torque and power are reduced by:

- Coulomb friction between teeth surfaces on gears B and F, characterized by efficiency, η
- Viscous coupling of driveshafts with bearings, parametrized by viscous friction coefficients, μ

Constant Efficiency

In the constant efficiency case, η is constant, independent of load or power transferred.

Load-Dependent Efficiency

In the load-dependent efficiency case, η depends on the load or power transferred across the gears. For either power flow,

$$\tau_{Coul} = g_{FB}\tau_{idle} + k\tau_F$$

where:

- τ_{Coul} is the Coulomb friction dependent torque.
- k is a proportionality constant.
- τ_{idle} is the net torque acting on the input shaft in idle mode.

Efficiency, η, is related to τ_{Coul} in the standard, preceding form but becomes dependent on load:

$$\eta = \frac{\tau_F}{g_{FB}\tau_{idle} + (k + 1)\tau_F}$$

Backlash

You can incorporate the effects of backlash in your model. Backlash is the excess space between a gear tooth and the mating gear teeth. Increasing the backlash compensates for lowering manufacturing tolerances and allows the free motion of lubricants in the gears to prevent jamming. However, excess backlash can cause premature wear on your system components and can affect measurements that rely on gear position. The block applies backlash for start-ups and reversals.

You can set **Backlash model** to:

- `No backlash` — The block ignores the effects of backlash.
- `Spring-damper` — The block uses a Translational Hard Stop implementation to simulate backlash similar to a spring and damper in parallel.

- `Instantaneous impulse` — The block locks the gears in contact while the relative backlash velocity remains below the **Static Contact Threshold** parameter. The block unlocks the gears when the contact force changes direction.

    This option is suitable for HIL simulation.

When you set **Backlash model** to `Spring-damper` or `Instantaneous impulse`, the block relates gear rotation to linear backlash as:

$$v_{Tooth} = r_B\omega_B - \beta r_F\omega_F,$$

where:

- v_{Tooth} is the relative linear velocity of the gear tooth.
- r_B is the **Base (B) gear radius** parameter associated with the base gear.
- r_F is the follower gear radius, where $r_F = N_F/N_B \cdot r_B$, and the **Follower (F) to base (B) teeth ratio (NF/NB)** parameter represents N_F/N_B.
- ω_B and ω_F are the angular velocities of the base and follower gears, respectively.
- β is the gear direction sign. When you set:

    - **Output shaft rotates** to `In same direction as input shaft`, $\beta = 1$.
    - **Output shaft rotates** to `In opposite direction as input shaft`, $\beta = -1$.

The block treats the meshing gear tooth as a position, x_{Tooth}, with respect to the linear backlash, *Backlash*, where *-1/2·Backlash < x_{Tooth} < 1/2·Backlash*. *Backlash* is equivalent to the **Linear backlash** parameter. The **Initial offset** parameter is equivalent to the initial position of x_{Tooth}.

When you set **Backlash model** to `Instantaneous impulse`, the hard stop can incorporate a nonzero value for the **Coefficient of restitution** parameter, *e*, into the momentum balance equation. During a collision,

$$e = \frac{v_{Backlash,\, t-}}{v_{Backlash,\, t+}},$$

where *t-* and *t+* are the instants before and after the collision, respectively. The block asserts *0 < e < 1*. For more information, see "State Reset Modeling". Simscape logs the mode state of the gear as the intermediate **M**.

State	Value
M = 0	Disengaged
M = 1	Forwards engaged with x_{tooth} = *1/2·Backlash*
M = -1	Backwards engaged with x_{tooth} = *-1/2·Backlash*

The hard stop simulates static contact at the bounds. The gears lock when a collision occurs and |v_{Tooth}| < v_{tol}, such that v_{Tooth} = *0*. Here, v_{tol} is equivalent to the **Static Contact Threshold** parameter.

Faults

If you enable faults for the block, the efficiency changes in response to one or both of these triggers:

- Simulation time — A fault occurs at a specified time.

- Simulation behavior — A fault occurs in response to an external trigger. Enabling an external fault trigger exposes port **T**.

If a fault trigger occurs, for the remainder of the simulation, the block uses the faulted efficiency in one of these ways:

- Throughout rotation
- When the rotation angle is within a faulted range that you specify

You can program the block to issue a fault report as a warning or error message.

Thermal Model

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Friction model** to `Temperature-dependent efficiency`.

Additionally, you can choose to model efficiency that varies with loading and temperature by setting **Friction model** to `Temperature and load-dependent efficiency`. Selecting a thermal variant:

- Exposes port **H**, a conserving port in the thermal domain.
- Enables the **Thermal mass** parameter, which allows you to specify the ability of the component to resist changes in temperature.
- Enables the **Initial Temperature** parameter, which allows you to set the initial temperature.

Variables

Use the **Variables** settings to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Assumptions

- Gear inertia is assumed to be negligible.
- Gears are treated as rigid components.
- Coulomb friction slows down simulation. For more information, see "Adjust Model Fidelity".

Ports

Input

T — External fault trigger
physical signal

Physical signal input port for an external fault trigger.

Dependencies

To enable the **T** port:

1   On the **Meshing Losses** tab, set **Friction model** to `Constant efficiency`, `Load-dependent efficiency`, `Temperature-dependent efficiency`, or `Temperature and load-dependent efficiency`.

2 On the **Faults** tab:

- Set **Enable faults** to On.
- Set **Enable external fault trigger** to On.

3 Click **OK** or **Apply**.

For information on related dependencies, see "Parameter Dependencies Table" on page 1-432.

Conserving

B — Base
rotational mechanical

Rotational mechanical conserving port associated with the base, or input, shaft.

F — Follower
rotational mechanical

Rotational mechanical conserving port associated with the follower, or output, shaft.

H — Heat flow
thermal

Thermal conserving port associated with heat flow. Heat flow affects gear temperature, and therefore, power transmission efficiency.

Dependencies

To enable this port, on the **Meshing Losses** tab set **Friction model** to either:

- Temperature-dependent efficiency
- Temperature and load-dependent efficiency

Parameters

Parameter Dependencies Table

The table shows how the visibility of some **Meshing Losses** parameters and **Faults** parameters depend on the thermal model and the option that you choose for other parameters. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Default Model — For nonthermal models, thermal port H is not visible.			Thermal Model — For thermal models, thermal port H is visible.		
Meshing Losses			Meshing Losses		
Friction model — Choose No meshing losses - Suitable for HIL simulation, Constant efficiency, or Load-dependent efficiency			Friction model — Choose Temperature-dependent efficiency or Temperature and load-dependent efficiency		
No meshing losse	Constant efficiency	Load-dependent efficiency	Temperature-dependent efficiency	Temperature and load-dependent efficiency	

Default Model — For nonthermal models, thermal port H is not visible.			Thermal Model — For thermal models, thermal port H is visible.	
Meshing Losses			Meshing Losses	
Friction model — Choose `No meshing losses - Suitable for HIL simulation`, `Constant efficiency`, or `Load-dependent efficiency`			Friction model — Choose `Temperature-dependent efficiency` or `Temperature and load-dependent efficiency`	
s - Suita ble for HIL simul ation	Efficiency	Input shaft torque at no load	Temperature	Temperature
	Follower power threshold	Nominal output torque	Efficiency	Load at base gear
		Efficiency at nominal output torque	Follower power threshold	Efficiency matrix
		Follower angular velocity threshold		Follower angular velocity threshold
Fault s	Faults			
	Enable faults — Choose `Off` or `On`			
	Off	On		
		Faulted efficiency		
		Enable external fault trigger — Choose `Off` or `On`. Selecting `On` makes thermal port T visible.		
		Enable temporal fault trigger — Choose `Off` or `On`		
		Off	On	
			Simulation time for fault event	
		Faulted angle range		
		Reporting when fault occurs — Choose `None`, or `Warning`, or `Error`		

Main

Follower (F) to base (B) teeth ratio (NF/NB) — Gear ratio
2 (default) | positive scalar

Fixed ratio g_{FB} of the follower axis to the base axis. The gear ratio must be strictly positive.

Output shaft rotates — Motion direction
`In opposite direction to input shaft` (default) | `In same direction as input shaft`

Direction of motion of the follower (output) driveshaft relative to the motion of the base (input) driveshaft.

Meshing Losses

Meshing losses parameters depend on the thermal model. For more information, see "Parameter Dependencies Table" on page 1-432.

Friction model — Default friction model
No meshing losses - Suitable for HIL simulation (default) | Constant efficiency |
Load-dependent efficiency | Temperature-dependent efficiency | Temperature and
load-dependent efficiency

Friction models at various precision levels for estimating power losses due to meshing.

- No meshing losses - Suitable for HIL simulation — Neglect friction between gear cogs. Meshing is ideal.
- Constant efficiency — Reduce torque transfer by a constant efficiency factor. This factor falls in the range $0 < \eta \le 1$ and is independent from load.
- Load-dependent efficiency — Reduce torque transfer by a variable efficiency factor. This factor falls in the range $0 < \eta < 1$ and varies with the torque load.
- Temperature-dependent efficiency — Reduce torque transfer by a constant efficiency factor that is dependent on temperature but does not consider the gear load. This factor falls in the range $0 < \eta \le 1$ and is independent from load. Torque transfer is determined from user-supplied data for gear efficiency and temperature.
- Temperature and load-dependent efficiency — Reduce torque transfer by a variable efficiency factor that is dependent on temperature and load. This factor falls in the range $0 < \eta < 1$ and varies with the torque load. Torque transfer efficiency is determined from user-supplied data for gear loading and temperature.

Efficiency — Torque transfer efficiency, η
0.8 (default) | $0 \le \eta < 1$

Torque transfer efficiency, η, between base and follower shafts. Efficiency is inversely proportional to the meshing power losses.

Dependencies

To enable this parameter, set **Friction model** to Constant efficiency.

For more information, see "Parameter Dependencies Table" on page 1-432.

Follower power threshold — Efficiency factor power basis
0.001 W (default) | positive scalar

Absolute value of the follower shaft power above which the full efficiency factor is in effect. Below this value, a hyperbolic tangent function smooths the efficiency factor to 1, lowering the efficiency losses to 0 when no power is transmitted.

As a guideline, the power threshold should be lower than the expected power transmitted during simulation. Higher values might cause the block to underestimate efficiency losses. Very low values tend to raise the computational cost of simulation.

Dependencies

To enable this parameter, set **Friction model** to Constant efficiency.

For more information, see "Parameter Dependencies Table" on page 1-432.

Input shaft torque at no load — No-load torque
0.1 N*m (default) | positive scalar

Net torque,τ_{idle}, acting on the input shaft in idle mode, that is, when torque transfer to the output shaft equals zero. For nonzero values, the power input in idle mode completely dissipates due to meshing losses.

Dependencies

To enable this parameter, set **Friction model** to `Load-dependent efficiency`.

For more information, see "Parameter Dependencies Table" on page 1-432.

Nominal output torque — Nominal torque
`5 N*m` (default) | positive scalar

Output torque, τ_F, at which to normalize the load-dependent efficiency.

Dependencies

To enable this parameter, set **Friction model** to `Load-dependent efficiency`.

For more information, see "Parameter Dependencies Table" on page 1-432.

Efficiency at nominal output torque — Nominal efficiency, *η*
`0.8` (default) | $0 \leq \eta < 1$ | scalar

Torque transfer efficiency, η, at the nominal output torque. Larger efficiency values correspond to greater torque transfer between the input and output shafts.

Dependencies

To enable this parameter, set **Friction model** to `Load-dependent efficiency`.

For more information, see "Parameter Dependencies Table" on page 1-432.

Follower angular velocity threshold — Efficiency factor angular velocity basis
`0.01 rad/s` (default) | positive scalar

Absolute value of the follower shaft angular velocity above which the full efficiency factor is in effect, ω_F. Below this value, a hyperbolic tangent function smooths the efficiency factor to one, lowering the efficiency losses to zero when at rest.

As a guideline, the angular velocity threshold should be lower than the expected angular velocity during simulation. Higher values might cause the block to underestimate efficiency losses. Very low values tend to raise the computational cost of simulation.

Dependencies

To enable this parameter, set **Friction model** to `Load-dependent efficiency`.

For more information, see "Parameter Dependencies Table" on page 1-432.

Temperature — Temperature
`[280, 300, 320]` K (default) | positive array

Array of temperatures used to construct an efficiency lookup table. The array values must increase from left to right. The temperature array must be the same size as the efficiency array in temperature-dependent models. The array must be the same size as a single row of the efficiency matrix in temperature and load dependent models.

Dependencies

To enable this parameter, set **Friction model** to either:

- `Temperature-dependent efficiency`
- `Temperature and load-dependent efficiency`

For more information, see "Parameter Dependencies Table" on page 1-432.

Efficiency — Efficiency, η
`[.95, .9, .85]` (default) | $0 \leq \eta < 1$ | array

Array of efficiencies used to construct a 1-D temperature-efficiency lookup table for temperature-dependent efficiency models. The array values are the efficiencies at the temperatures in the **Temperature** array. The number of elements must be the same as the number of elements in the **Temperature** array.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

For more information, see "Parameter Dependencies Table" on page 1-432.

Follower power threshold — Efficiency factor power basis
`0.001` W (default) | positive array

Absolute value of the follower shaft power above which the full efficiency factor is in effect, p_F. Below this value, a hyperbolic tangent function smooths the efficiency factor to 1, lowering the efficiency losses to 0 when no power is transmitted.

As a guideline, the power threshold should be lower than the expected power transmitted during simulation. Higher values might cause the block to underestimate efficiency losses. Very low values tend to raise the computational cost of simulation.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

For more information, see "Parameter Dependencies Table" on page 1-432.

Load at base gear — Base gear load
`[1, 5, 10]` N*m (default) | positive array

Array of base-gear loads used to construct a 2-D temperature load efficiency lookup table for temperature and load dependent efficiency models. The array values must increase left to right. The load array must be the same size as a single column of the efficiency matrix.

Dependencies

To enable this parameter, set **Friction model** to `Temperature and load-dependent efficiency`.

For more information, see "Parameter Dependencies Table" on page 1-432.

Efficiency matrix — Efficiency
`[.85, .8, .75; .95, .9, .85; .85, .8, .7]` (default) | numerical array with positive elements in the range of [0,1)

Matrix of component efficiencies used to construct a 2-D temperature load efficiency lookup table. The matrix elements are the efficiencies at the temperatures given by the **Temperature** array and at the loads given by the **Load at base gear** array.

The number of rows must be the same as the number of elements in the **Temperature** array. The number of columns must be the same as the number of elements in the **Load at base gear** array.

Dependencies

To enable this parameter, set **Friction model** to `Temperature and load-dependent efficiency`.

For more information, see "Parameter Dependencies Table" on page 1-432.

Follower angular velocity threshold — Efficiency factor angular velocity basis
`0.01 rad/s` (default) | positive scalar

Absolute value of the follower shaft angular velocity above which the full efficiency factor is in effect, ω_F. Below this value, a hyperbolic tangent function smooths the efficiency factor to one, lowering the efficiency losses to zero when at rest.

As a guideline, the angular velocity threshold should be lower than the expected angular velocity during simulation. Higher values might cause the block to underestimate efficiency losses. Very low values tend to raise the computational cost of simulation.

Dependencies

To enable this parameter, set **Friction model** to `Temperature and load-dependent efficiency`.

For more information, see "Parameter Dependencies Table" on page 1-432.

Backlash

Backlash model — Option to include backlash
`No backlash` (default) | `Spring-damper` | `Instantaneous Impulse`

Backlash option. You can select:

- `No backlash` — The block ignores the effects of backlash.
- `Spring-damper` — The block uses a hard stop implementation to simulate backlash.
- `Instantaneous impulse` — The block locks the gears in contact while the relative backlash velocity remains below the **Static Contact Threshold** parameter. The block unlocks the gears when the contact force changes direction.

  This option is suitable for HIL simulation.

Linear backlash — Translational backlash distance
`1e-3 mm` (default) | positive scalar

Translational distance that a gear tooth can travel between meshing teeth.

Dependencies

To enable this parameter, set **Backlash** to `Spring-damper` or `Instantaneous impulse`.

1-437

Initial offset — Relative initial gear tooth position with respect to mating teeth
0 mm (default) | scalar

Initial translational position of the gear tooth with respect to the amount of backlash you specify in the **Linear backlash** parameter. Specify a position, x_{Tooth}, that meets the condition $-1/2 \cdot Backlash < x_{Tooth} < 1/2 \cdot Backlash$, where $Backlash$ is equivalent to the value that you specify for the **Linear backlash** parameter.

Dependencies

To enable this parameter, set **Backlash** to Spring-damper or Instantaneous impulse.

Base radius — Base gear pitch radius
0.1 m (default) | positive scalar

Base gear distance from the center to the meshing point on the gear tooth.

Dependencies

To enable this parameter, set **Backlash** to Spring-damper or Instantaneous impulse.

Hard stop model — Hard stop transition behavior
Stiffness and damping applied smoothly through transition region, damped rebound (default) | ...

Stiffness and rebound options for the hard stop model. You can choose from these options:

- Stiffness and damping applied smoothly through transition region, damped rebound
- Full stiffness and damping applied at bounds, undamped rebound
- Full stiffness and damping applied at bounds, damped rebound

For more information, see Translational Hard Stop.

Dependencies

To enable this parameter, set **Backlash** to Spring-damper.

Transition region — Region of gradual hard stop effects
1e-4 mm (default) | positive scalar

Distance where the block gradually applies effects of stiffness and damping. When you set **Hard stop model** to Stiffness and damping applied smoothly through transition region, damped rebound, the block smoothly transitions the onset of stiffness and damping as the hard stop approaches full stiffness.

Dependencies

To enable this parameter, set:

- **Backlash** to Spring-damper.
- **Hard stop model** to Stiffness and damping applied smoothly through transition region, damped rebound.

Linear stiffness — Resistance to translational deformation
1e7 N/m (default) | positive scalar

Effective translational spring stiffness of the gear collision.

Dependencies

To enable this parameter, set **Backlash** to `Spring-damper`.

Linear damping — Tendency to dissipate translational energy
`1e6 N/(m/s)` (default) | positive scalar

Effective translational damping of the gear collision.

Dependencies

To enable this parameter, set **Backlash** to `Spring-damper`.

Coefficient of restitution — Collision energy loss coefficient
`0` (default) | positive scalar

Loss of translational kinetic energy during collisions. A value of `0` represents an inelastic collision, and a value of `1` represents a perfectly elastic collision where the gear retains all kinetic energy. The default of `0` is equivalent to simulating gear torque when the gears are in contact, and removing torque when the gear changes direction and the tooth travels the backlash distance.

Dependencies

To enable this parameter, set **Backlash** to `Instantaneous impulse`.

Static contact threshold — Collision velocity tolerance
`0.001` m/s (default) | positive scalar

Velocity, v_{tol}, below which the gear tooth becomes locked with the meshing teeth. The block sets $v_{Tooth} = 0$ when $|v_{Tooth}| < v_{tol}$.

Dependencies

To enable this parameter, set **Backlash** to `Instantaneous impulse`.

Viscous Losses

Viscous friction coefficients at base (B) and follower (F) — Coefficients of viscous friction
`[0, 0] N*m/(rad/s)` (default) | positive array

Two-element array with the viscous friction coefficients in effect at the base and follower shafts. To neglect viscous losses, use the default setting, `[0, 0]`.

Faults

The **Faults** tab is not visible when you set **Friction model** to `No meshing losses - Suitable for HIL simulation`.

Enable faults — Fault option
`on` (default) | `off`

Enable externally or temporally triggered faults.

Dependencies

This parameter is not visible when you set **Friction model** to `No meshing losses - Suitable for HIL simulation` on the **Meshing Losses** tab. This parameter affects the visibility of other **Faults** parameters.

For more information, see "Parameter Dependencies Table" on page 1-432.

Faulted efficiency — Faulted efficiency
`0.05` (default) | $0 \leq \eta < 1$ | scalar

Efficiency when a fault is triggered.

Dependencies

This parameter is not visible when either:

- On the **Meshing Losses** tab, **Friction model** is set to `No meshing losses - Suitable for HIL simulation`.
- **Enable faults** is set to `Off`.

For more information, see "Parameter Dependencies Table" on page 1-432.

Enable external fault trigger — External fault trigger option
`On` (default) | `Off`

Option to enable an externally triggered fault.

Dependencies

This parameter is not visible when either:

- On the **Meshing Losses** tab, **Friction model** is set to `No meshing losses - Suitable for HIL simulation`.
- **Enable faults** is set to `Off`. When you select `On` for this parameter, the **T** port is exposed.

For more information, see "Parameter Dependencies Table" on page 1-432.

Enable temporal fault trigger — Temporal fault trigger option
`On` (default) | `Off`

Option to enable a temporally triggered fault.

Dependencies

This parameter is not visible when either:

- On the **Meshing Losses** tab, **Friction model** is set to `No meshing losses - Suitable for HIL simulation`.
- **Enable faults** is set to `Off`.

For more information, see "Parameter Dependencies Table" on page 1-432.

Simulation time for fault event — Fault time trigger
`5 s` (default) | positive scalar

Simulation time that triggers a temporal fault.

Dependencies

This parameter is not visible when either:

- On the **Meshing Losses** tab, **Friction model** is set to `No meshing losses - Suitable for HIL simulation`.
- **Enable faults** is set to `On` and **Enable temporal fault trigger** is set to `Off`.

For more information, see "Parameter Dependencies Table" on page 1-432.

Faulted angle range — Fault angle range
`pi / 12 rad` (default)

Rotational angle range for the faulted efficiency. For a value or multiples of 2π rad, the faulted efficiency is applicable throughout rotation.

Dependencies

This parameter is not visible when either:

- On the **Meshing Losses** tab, **Friction model** is set to `No meshing losses - Suitable for HIL simulation`.
- **Enable faults** is set to `Off`.

For more information, see "Parameter Dependencies Table" on page 1-432.

Reporting when fault occurs — Fault report
`None` (default) | `Warning` | `Error`

Reporting preference for the fault condition.

Dependencies

This parameter is not visible when either:

- On the **Meshing Losses** tab, **Friction model** is set to `No meshing losses - Suitable for HIL simulation`.
- **Enable faults** is set to `Off`.

For more information, see "Parameter Dependencies Table" on page 1-432.

Thermal Port

Thermal mass — Thermal mass
`50 J/K` (default) | positive scalar

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

To enable this parameter, set **Friction model** to either:

- `Temperature-dependent efficiency`
- `Temperature and load-dependent efficiency`

`Initial temperature` — Temperature
300 K (default) | positive scalar

Temperature at simulation start.

Dependencies

To enable this parameter, set **Friction model** to either:

- `Temperature-dependent efficiency`
- `Temperature and load-dependent efficiency`

More About

Hardware-in-the-Loop Simulation

For optimal performance of your real-time simulation, set the **Friction model** to `No meshing losses - Suitable for HIL simulation` on the **Meshing Losses** tab.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Variable Ratio Transmission

Topics
"Basic Motion, Torque, and Force Modeling"
"Gearbox Efficiency Measurement"
"Simple Gear"
"Gear with Backlash"
"Model Thermal Losses in Driveline Components"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

Simple Gear with Variable Efficiency

Gear set with parallel-axis rotation and variable meshing efficiency
Library: Simscape / Driveline / Gears
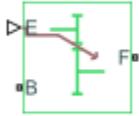
Description

The block represents a simple gear train with variable meshing efficiency. The gear train transmits torque at a specified ratio between base and follower shafts arranged in a parallel configuration. Shaft rotation can occur in equal or opposite directions. Gear losses are optional. They include meshing and viscous bearing losses. To specify the variable meshing efficiency, the block contains a physical signal port that you can use to input a general time-varying signal. Inertia and compliance effects are ignored.

Simple Gear Model

Ideal Gear Constraint and Gear Ratio

Simple Gear imposes one kinematic constraint on the two connected axes:

$$r_F\omega_F = r_B\omega_B \ . \tag{1-9}$$

The follower-base gear ratio $g_{FB} = r_F/r_B = N_F/N_B$. N is the number of teeth on each gear. The two degrees of freedom reduce to one independent degree of freedom.

The torque transfer is:

$$g_{FB}\tau_B + \tau_F - \tau_{loss} = 0 \ , \tag{1-10}$$

with $\tau_{loss} = 0$ in the ideal case.

Nonideal Gear Constraint and Losses

In the nonideal case, $\tau_{loss} \neq 0$. For general considerations on nonideal gear modeling, see "Model Gears with Losses".

Thermal Model

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Friction model** to `Temperature-dependent efficiency`.

Variables

Use the **Variables** settings to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Limitations

- Gear inertia is assumed to be negligible.
- Gears are treated as rigid components.
- Coulomb friction slows down simulation. For more information, see "Adjust Model Fidelity".

Ports

Port	Description
B	Rotational conserving port representing the base shaft
F	Rotational conserving port representing the follower shaft
H	Thermal conserving port for thermal modeling

Parameters

Main

Follower (F) to base (B) teeth ratio (NF/NB) — Gear ratio
2 (default) | positive scalar

Gear ratio. This is the fraction of follower over base gear teeth numbers, NF/NB. The ratio must be positive.

Output shaft rotates — Shaft rotation orientation
In opposite direction to input shaft (default) | In same direction as input shaft

Relative rotation between shafts. This is the rotation direction of the output shaft with respect to the input shaft. You can choose either equal or opposite rotational directions.

Meshing Losses

Minimum efficiency — Minimum efficiency
0.01 (default) | positive scalar in the range (0,1]

Smallest efficiency value allowed for the gear. The efficiency is the power ratio between output and input shafts. The physical signal input is maintained at values below the minimum efficiency or above a value of 1. The value must be positive.

Follower power threshold — Power below which numerical smoothing is applied
0.001 W (default) | positive scalar

Follower shaft power above which full efficiency factor is in effect. A hyperbolic tangent function smooths the efficiency factor between zero, when at rest, and the efficiency set point.

Viscous Losses

Viscous friction coefficients at base (B) and follower (F) — Fluid dynamic friction coefficients
[0 0] N*m/(rad/s) (default) | positive two-element vector

Two-element array with the viscous friction coefficients in effect at the base and follower shafts. The default array corresponds to zero viscous losses.

Thermal Port

Thermal port — Whether to model thermal effects
Omit (default) | Model

Whether to model thermal effects and enable port **H** for temperature-dependent performance.

Thermal mass — Thermal mass
50 J/K (default) | positive scalar

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

To enable this parameter, set **Thermal port** to Model.

Initial temperature — Initial temperature
300 K (default) | positive scalar

Component temperature at the start of simulation. The initial temperature alters the component efficiency according to an efficiency vector that you specify, affecting the starting meshing or friction losses.

Dependencies

To enable this parameter, set **Thermal port** to Model.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Simple Gear | Variable Ratio Transmission

Topics
"Basic Motion, Torque, and Force Modeling"

Introduced in R2013a

Sinusoidal Force Source

Produce sinusoidal force
Library: Simscape / Driveline / Sources

Description

The Sinusoidal Force Source block produces sinusoidal force with a constant offset. A positive force value generates force acting from port B to port F.

The defining equation for the sinusoidal force that the block generates is

$$F = F_o + A * \sin(2\pi f t + \varphi),$$

where:

- F is the output force.
- F_o is the offset force of the output.
- A is the peak amplitude of the sinusoid.
- f is frequency of the sinusoid in Hz.
- t is time.
- φ is phase shift of the sinusoid.

Ports

Conserving

B — Base interface
mechanical translational

Mechanical translational conserving port associated with the base interface.

F — Follower interface
mechanical translational

Mechanical translational conserving port associated with the follower interface.

Parameters

Offset — Output average value
0 N (default)

Average value that the output sinusoidal force oscillates about.

Sinusoid peak amplitude — Output peak amplitude
1 N (default)

Peak amplitude of the output sinusoidal force.

Phase shift — Output phase shift
0 deg (default)

Phase shift of the output sinusoidal force.

Frequency — Output frequency
1 Hz (default)

Frequency of the output sinusoidal force. The frequency must be greater than or equal to 0.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Force Noise Source | Sinusoidal Rotational Velocity Source | Sinusoidal Torque Source | Sinusoidal Translational Velocity Source

Introduced in R2016b

Sinusoidal Rotational Velocity Source

Produce sinusoidal rotational velocity
Library: Simscape / Driveline / Sources

Description

The Sinusoidal Rotational Velocity Source block produces sinusoidal rotational velocity with a constant offset.

The relative rotational velocity between the base and follower interfaces is

$$v_{FB} = v_F - v_B,$$

where:

- v_{FB} is the relative rotational velocity.
- v_F is the absolute rotational velocity at the follower interface.
- v_B is the absolute rotational velocity at the base interface.

The defining equation for the sinusoidal rotational velocity that the block generates is

$$v = v_o + A * \sin(2\pi f t + \varphi),$$

where:

- v is the output rotational velocity.
- v_o is the offset rotational velocity of the output.
- A is the peak amplitude of the sinusoid.
- f is frequency of the sinusoid in Hz.
- t is time.
- φ is phase shift of the sinusoid.

Ports

Conserving

B — Base interface
mechanical rotational

Mechanical rotational conserving port associated with the base interface.

F — Follower interface
mechanical rotational

Mechanical rotational conserving port associated with the follower interface.

Parameters

`Offset` — Output average value
`0 rad/s` (default)

Average value that the output sinusoidal rotational velocity oscillates about.

`Sinusoid peak amplitude` — Output peak amplitude
`1 rad/s` (default)

Peak amplitude of the output sinusoidal rotational velocity.

`Phase shift` — Output phase shift
`0 deg` (default)

Phase shift of the output sinusoidal rotational velocity.

`Frequency` — Output frequency
`1 Hz` (default)

Frequency of the output sinusoidal rotational velocity. The frequency must be greater than or equal to 0.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Rotational Velocity Noise Source | Sinusoidal Force Source | Sinusoidal Torque Source | Sinusoidal Translational Velocity Source

Introduced in R2016b

Sinusoidal Torque Source

Produce sinusoidal torque
Library: Simscape / Driveline / Sources

Description

The Sinusoidal Torque Source block produces sinusoidal torque with a constant offset. A positive torque value generates torque acting from port B to port F.

The defining equation for the sinusoidal torque that the block generates is

$$T = T_o + A * \sin(2\pi f t + \varphi),$$

where:

- T is the output torque.
- T_o is the offset torque of the output.
- A is the peak amplitude of the sinusoid.
- f is frequency of the sinusoid in Hz.
- t is time.
- φ is phase shift of the sinusoid.

Ports

Conserving

B — Base interface
mechanical rotational

Mechanical rotational conserving port associated with the base interface.

F — Follower interface
mechanical rotational

Mechanical rotational conserving port associated with the follower interface.

Parameters

Offset — Output average value
0 N*m (default)

Average value that the sinusoidal output torque oscillates about.

Sinusoid peak amplitude — Output peak amplitude
1 N*m (default)

Peak amplitude of the sinusoidal output torque.

`Phase shift` — Output phase shift
`0 deg` (default)

Peak amplitude of the sinusoidal output torque.

`Frequency` — Output frequency
`1 Hz` (default)

Frequency of the sinusoidal output torque. The frequency must be greater than or equal to 0.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Sinusoidal Force Source | Sinusoidal Rotational Velocity Source | Sinusoidal Translational Velocity Source | Torque Noise Source

Introduced in R2016b

Sinusoidal Translational Velocity Source

Produce sinusoidal translational velocity
Library: Simscape / Driveline / Sources

Description

The Sinusoidal Translational Velocity Source block produces sinusoidal translational velocity with a constant offset.

The relative translational velocity between the base and follower interfaces is

$$v_{FB} = v_F - v_B,$$

where:

- v_{FB} is the relative translational velocity.
- v_F is the absolute translational velocity at the follower interface.
- v_B is the absolute translational velocity at the base interface.

The defining equation for the sinusoidal translational velocity that the block generates is

$$v = v_o + A * \sin(2\pi f t + \varphi),$$

where:

- v is the output translational velocity.
- v_o is the offset translational velocity of the output.
- A is the peak amplitude of the sinusoid.
- f is frequency of the sinusoid in Hz.
- t is time.
- φ is phase shift of the sinusoid.

Ports

Conserving

B — Base interface
mechanical translational

Mechanical translational conserving port associated with the base interface.

F — Follower interface
mechanical translational

Mechanical translational conserving port associated with the follower interface.

Parameters

`Offset` — Output average value
`0 m/s` (default)

Average value that the output sinusoidal translational velocity oscillates about.

`Sinusoid peak amplitude` — Output peak amplitude
`1 m/s` (default)

Peak amplitude of the output sinusoidal translational velocity.

`Phase shift` — Output phase shift
`0 deg` (default)

Phase shift of the output sinusoidal translational velocity

`Frequency` — Output frequency
`1 Hz` (default)

Frequency of the output sinusoidal translational velocity. The frequency must be greater than or equal to 0.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Sinusoidal Force Source | Sinusoidal Rotational Velocity Source | Sinusoidal Torque Source | Translational Velocity Noise Source

Introduced in R2016b

Sun-Planet

Planetary gear set of carrier, planet, and sun wheels with adjustable gear ratio and friction losses
Library: Simscape / Driveline / Gears / Planetary Subcomponents
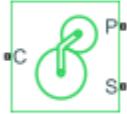
Description

The Sun-Planet gear block represents a set of carrier, planet, and sun gear wheels. The planet is connected to and rotates with respect to the carrier. The planet and sun corotate with a fixed gear ratio that you specify and in the same direction with respect to the carrier. A sun-planet and a ring-planet gear are basic elements of a planetary gear set. For model details, see "Equations" on page 1-454.

Thermal Model

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Friction model** to Temperature-dependent efficiency.

Equations

Ideal Gear Constraints and Gear Ratios

Sun-Planet imposes one kinematic and one geometric constraint on the three connected axes:

$$r_C\omega_C = r_S\omega_S + r_P\omega_P$$

The planet-sun gear ratio is

$$g_{PS} = r_P/r_S = N_P/N_S$$

Where N is the number of teeth on each gear. In terms of this ratio, the key kinematic constraint is:

$$\omega_S = -g_{PS}\omega_P + (1 + g_{PS})\omega_C$$

The three degrees of freedom reduce to two independent degrees of freedom. The gear pair is (1, 2) = (S, P).

Warning The planet-sun gear ratio g_{PS} must be strictly greater than one.

The torque transfer is:

$$g_{PS}\tau_S + \tau_P - \tau_{loss} = 0$$

In the ideal case, there is no torque loss, that is $\tau_{loss} = 0$.

Nonideal Gear Constraints and Losses

In the nonideal case, $\tau_{loss} \neq 0$. For more information, see "Model Gears with Losses".

Variables

Use the **Variables** settings to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Limitations and Assumptions

- Gear inertia is assumed to be negligible.
- Gears are treated as rigid components.
- Coulomb friction slows down simulation. For more information, see "Adjust Model Fidelity".

Ports

Conserving

C — Planet gear carrier
rotational mechanical

Rotational conserving port associated with the planet gear carrier.

P — Planet gear
rotational mechanical

Rotational conserving port associated with the panet gear.

S — Sun gear
rotational mechanical

Rotational conserving port associated with the sun gear.

H — Heat flow
thermal

Thermal conserving port associated with heat flow. Heat flow affects the power transmission efficiency by altering the gear temperatures.

Dependencies

To enable this port, set **Friction model** to Temperature-dependent efficiency.

Parameters

Main

Planet (P) to sun (S) teeth ratio (NP/NS) — Planet-to-sun gear ratio
2 (default) | scalar greater than 1

Ratio g_{PS} of the planet gear wheel radius to the sun gear wheel radius. This gear ratio must be strictly greater than 1.

Meshing Losses

Friction model — Friction model
No meshing losses - Suitable for HIL simulation (default) | Constant efficiency | Temperature-dependent efficiency

Friction model for the block:

- No meshing losses - Suitable for HIL simulation — Gear meshing is ideal.
- Constant efficiency — Transfer of torque between the gear wheel pairs is reduced by a constant efficiency, η, such that $0 < \eta \leq 1$.
- Temperature-dependent efficiency — Transfer of torque between the gear wheel pairs is defined by the table lookup based on the temperature.

Ordinary efficiency — Efficiency
0.98 (default) | scalar | $0 < \eta_{SP} \leq 1$

Torque transfer efficiency, η_{SP}, for sun-planet gear wheel pair meshings. The value must be greater than 0 and less than or equal to 1.

Dependencies

This parameter is exposed when the **Friction model** parameter is set to Constant efficiency.

Temperature — Temperature
[280, 300, 320] K (default) | vector

Vector of temperatures used to construct a 1-D temperature-efficiency lookup table. The vector elements must increase from left to right.

Dependencies

To enable this parameter, set **Friction model** to Temperature-dependent efficiency.

Efficiency — Gear efficiency
[.95, .9, .85] (default) | array | $0 < \eta_{SP} \leq 1$

Array of mechanical efficiencies, ratios of output power to input power, for the power flow from the ring gear to the planet gear, η_{RP}. The block uses the values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

This parameter is exposed when the **Friction model** parameter is set to Temperature-dependent efficiency.

Sun-carrier power threshold — Power threshold
0.001 W (default) | scalar

Power threshold, p_{th}, above which full efficiency is in effect. Below this values, a hyperbolic tangent function smooths the efficiency factor. For a model without thermal losses, the function lowers the efficiency losses to zero when no power is transmitted. For a model that considers thermal losses, the

function smooths the efficiency factors between zero at rest and the values provided by the temperature-efficiency lookup tables at the power thresholds.

Dependencies

This parameter is exposed when the **Friction model** parameter is set to `Constant efficiency` or `Temperature-dependent efficiency`.

Viscous Losses

Sun-carrier viscous friction coefficient — Viscous friction
`0 N*m/(rad/s)` (default)

Viscous friction coefficient μ_S for the sun-carrier gear motion.

Thermal Port

These settings are visible when, in the **Meshing Losses** settings, the **Friction model** parameter is set to `Temperature-dependent efficiency`.

Thermal mass — Thermal mass
`50 J/K` (default) | scalar

Thermal energy required to change the component temperature by a single temperature unit. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

More About

Hardware-in-the-Loop Simulation

For optimal simulation performance, set **Friction model** to the default value, `No meshing losses - Suitable for HIL simulation`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Compound Planetary Gear | Planet-Planet | Planetary Gear | Ring-Planet | Sun-Planet Bevel

Topics
"Basic Motion, Torque, and Force Modeling"
"Model Thermal Losses in Driveline Components"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

Sun-Planet Bevel

Planetary gear set of carrier and beveled planet and sun wheels with adjustable gear ratio, assembly orientation, and friction losses
Library: Simscape / Driveline / Gears / Planetary Subcomponents
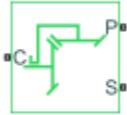
Description

The Sun-Planet Bevel gear block represents a carrier and beveled planet and sun wheels. The planet is connected to and rotates with respect to the carrier. The planet and sun gears corotate with a fixed gear ratio. You control the direction of rotation by setting the assembly orientation left or right. For model details, see "Equations" on page 1-458.

Thermal Model

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Friction model** to `Temperature-dependent efficiency`.

Equations

Ideal Gear Constraints and Gear Ratios

The Sun-Planet Bevel block imposes one kinematic and one geometric constraint on the three connected axes:

$$r_C \omega_C = r_S \omega_S \pm r_P \omega_P$$

$$r_C = r_S \pm r_P$$

where:

- r_C is the radius of the carrier gear.
- ω_C is the angular velocity of the carrier gear.
- r_S is the radius of the sun gear.
- ω_S is the angular velocity of the sun gear.
- r_P is the radius of the planet gear.
- ω_P is the angular velocity of the planet gear.

The planet-sun gear ratio is defined as

$$g_{PS} = \frac{r_P}{r_S} = \frac{N_P}{N_S},$$

where:

- g_{PS} is the planet-sun gear ratio. As $r_P > r_S$, $g_{PS} > 1$.

- N_P is the number of teeth in the planet gear.
- N_S is the angular velocity of the sun gear.

In terms of this ratio, the key kinematic constraint is:

- $\omega_S = g_{PS}\omega_P - \omega_C$ for a left-oriented bevel assembly
- $\omega_S = g_{PS}\omega_P + \omega_C$ for a right-oriented bevel assembly

The three degrees of freedom reduce to two independent degrees of freedom. The gear pair is $(1,2) = (S,P)$.

Warning The planet-sun gear ratio, g_{PS}, must be strictly greater than one.

The torque transfer is defined as

$$\tau_P = \tau_{loss} - g_{PS}\tau_S,$$

where:

- τ_{loss} is the torque loss.
- τ_s is the torque for the sun gear.
- τ_p is the torque for the planet gear.

In the ideal case where there is no torque loss, $\tau_{loss} = 0$. The resulting torque transfer equation is $\tau_P = g_{PS}\tau_S$.

Nonideal Gear Constraints and Losses

In the nonideal case, $\tau_{loss} \neq 0$. For more information, see "Model Gears with Losses".

Variables

Use the **Variables** settings to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Assumptions and Limitations

- Gear inertia is assumed to be negligible.
- Gears are treated as rigid components.
- Coulomb friction slows down simulation. For more information, see "Adjust Model Fidelity".

Ports

Conserving

C — Planet gear carrier
rotational mechanical

Rotational mechanical conserving port associated with the planet gear carrier.

P — Planet gear
rotational mechanical

Rotational mechanical conserving port associated with the planet gear.

H — Heat flow
thermal

Thermal conserving port associated with heat flow. Heat flow affects the power transmission efficiency by altering the gear temperatures.

Dependencies

To enable this port, set **Friction model** to Temperature-dependent efficiency.

Parameters

Main

Planet (P) to sun (S) teeth ratio (NP/NS) — Planet to sun gear rotation ratio
2 (default) | scalar

Ratio, g_{PS}, of the planet gear to sun gear rotations as defined by the number of planet gear teeth divided by the number of sun gear teeth. This gear ratio must be strictly greater than 1.

Assembly orientation — Gear relative rotational direction
Left - Sun and planet gears rotate in same direction (default) | Right - Sun and planet gears rotate in opposite directions

Relative orientation of the sun and planet gears and their corotation direction. Left or right orientation implies that the gears corotate in the same or opposite direction, respectively.

Meshing Losses

Friction model — Friction model
No meshing losses - Suitable for HIL simulation (default) | Constant efficiency | Temperature-dependent efficiency

Friction model for the block:

- No meshing losses - Suitable for HIL simulation — Gear meshing is ideal.
- Constant efficiency — Transfer of torque between the gear wheel pairs is reduced by a constant efficiency, η, such that $0 < \eta \leq 1$.
- Temperature-dependent efficiency — Transfer of torque between the gear wheel pairs is defined by the table lookup based on the temperature.

Ordinary efficiency — Torque transfer efficiency
0.9 (default) | scalar

Torque transfer efficiency, η_{PS}, for the planet gear to the sun gear pair meshing. This value must be greater than 0 and less than or equal to 1.

Dependencies

To enable this parameter, set **Friction model** to Constant efficiency.

Temperature — Temperature
[280, 300, 320] K (default) | vector

Vector of temperatures used to construct a 1-D temperature-efficiency lookup table. The vector elements must increase from left to right.

Dependencies

To enable this parameter, set **Friction model** to Temperature-dependent efficiency.

Efficiency — Gear efficiency
[.95, .9, .85] (default) | vector

Vector of output-to-input power ratios that describe the power flow from the outer planet gear to the inner planet gear, η_{PS}. The block uses the values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set **Friction model** to Temperature-dependent efficiency.

Sun-carrier power threshold — Minimum efficiency power threshold for the sun-carrier gear coupling
0.001 W (default) | scalar

Power threshold, p_{th}, above which full efficiency is in effect. Below this value, a hyperbolic tangent function smooths the efficiency factor.

When you set **Friction model** to Constant efficiency, the block lowers the efficiency losses to zero when no power is transmitted. When you set **Friction model** to Temperature-dependent efficiency, the block smooths the efficiency factors between zero when at rest and the values provided by the temperature-efficiency lookup tables at the power thresholds.

Dependencies

To enable this parameter, set **Friction model** to Constant efficiency or Temperature-dependent efficiency.

Viscous Losses

Sun-carrier viscous friction coefficient — Gear viscous friction
0 N*m/(rad/s) (default) | scalar

Viscous friction coefficient, μ_S, for the sun-carrier gear motion.

Thermal Port

To enable these settings, set **Friction model** to Temperature-dependent efficiency.

Thermal mass — Thermal mass
50 J/K (default) | scalar

Thermal energy required to change the component temperature by a single temperature unit. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

More About

Hardware-in-the-Loop Simulation

For optimal simulation performance, set **Friction model** to the default value, `No meshing losses - Suitable for HIL simulation`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Differential | Planet-Planet | Planetary Gear | Ring-Planet | Sun-Planet | Sun-Planet Worm Gear

Topics
"Basic Motion, Torque, and Force Modeling"
"Limited Slip Differential with Clutches"
"Custom Planetary Gear"
"Model Thermal Losses in Driveline Components"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

# Sun-Planet Worm Gear

Planetary gear set of carrier, worm planet, and sun wheels with adjustable gear ratio, worm thread type, and friction losses
Library: Simscape / Driveline / Gears / Planetary Subcomponents
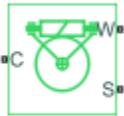
Description

The Sun-Planet Worm Gear block represents a two-degree-of-freedom planetary gear built from a carrier and sun and planet gears. By type, the sun and planet gears are crossed helical spur gears arranged as a worm drive, in which the planet gear is the worm. Such transmissions are used in the Torsen® T-1 differential. When transmitting power, the sun gear can be independently rotated by the worm (planet) gear, by the carrier, or by both.
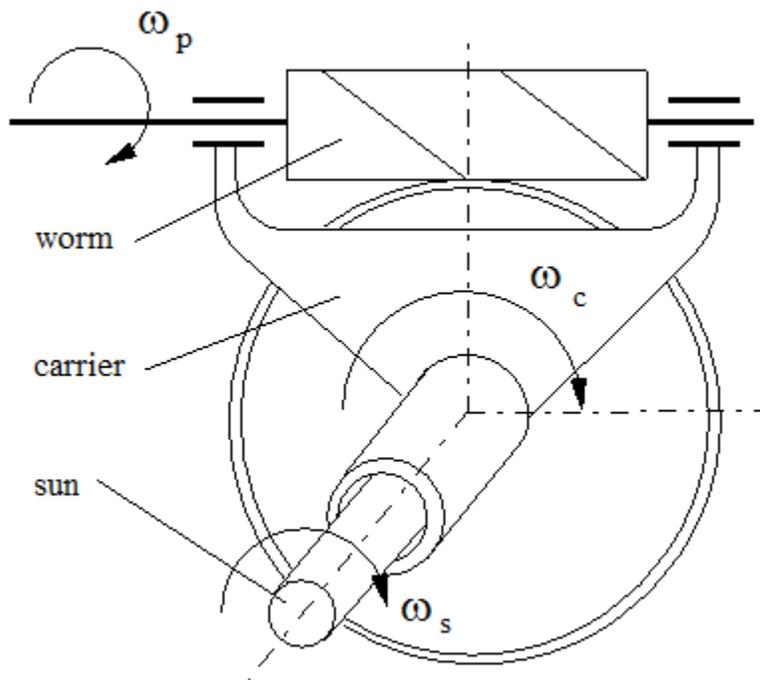
You specify a fixed gear ratio, which signifies the worm angular velocity divided by the sun gear angular velocity. You control the direction by setting the worm thread type to left-hand or right-hand. Rotation of the right-hand worm in the positive direction causes the sun gear to rotate in the positive direction. The positive directions of the sun gear and the carrier are the same.

Thermal Model

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Friction model** to `Temperature-dependent efficiency`.

Equations

Variables

Equation variables are:

R_{WG}	Gear ratio that signifies the worm angular velocity divided by the sun gear angular velocity: The ratio is positive for the right-hand worm and negative for the left-hand worm
ω_S	Angular velocity of the sun gear
ω_P	Angular velocity of the worm gear
ω_C	Angular velocity of the carrier
ω_{SC}	Angular velocity of the sun gear with respect to the carrier
α	Normal pressure angle
λ	Worm lead angle
L	Worm lead
d	Worm pitch diameter
τ_S	Torque applied to sun shaft
τ_P	Torque applied to planet shaft
τ_C	Torque applied to carrier shaft
τ	Torque due to meshing friction: The loss depends on the device efficiency and the power flow direction. To avoid an abrupt change of the friction torque at $\omega_S = 0$, the friction torque is introduced via the hyperbolic function.
τ_{instfr}	Instantaneous value of the friction torque used to simulate friction losses
τ_{fr}	Torque due to friction in steady-state
k	Friction coefficient
η_{WG}	Efficiency of worm to gear power transfer
η_{GW}	Efficiency of gear to worm power transfer
p_{th}	Power threshold
μ_{SC}	Viscous friction coefficient for the sun-carrier interface
μ_{WC}	Viscous friction coefficient for the worm-carrier interface

Ideal Gear Constraints and Gear Ratio

The Sun-Planet Worm Gear block imposes one kinematic constraint on the three connected axes:

$$\omega_S = \frac{\omega_P}{R_{WG}} + \omega_C .$$

The gear has two independent degrees of freedom. The gear pair is (1,2) = (S,P).

The torque transfer is:

$$R_{\mathrm{WG}}\tau_{\mathrm{P}} + \tau_{\mathrm{S}} - \tau_{\mathrm{loss}} = 0$$

$$\tau_{\mathrm{C}} = -\tau_{\mathrm{S}}$$

In the ideal case where there is no torque loss, $\tau_{loss} = 0$.

Nonideal Gear Constraints

In the nonideal case, $\tau_{loss} \neq 0$. For more information, see "Model Gears with Losses".

In a nonideal gear, the angular velocity and geometric constraints are unchanged, but the transferred torque and power are reduced by:

- Coulomb friction due to worm-sun gear meshing, which is characterized by the friction coefficient k or constant efficiencies [η_{WG}, η_{GW}]
- Viscous couplings of driveshafts with bearings, which are parametrized by viscous friction coefficients μ_{SC} and μ_{WC}

Because the transmission incorporates a worm gear, the efficiencies are different for the direct and reverse power transfer. The table shows the value of the efficiency for all combinations of the power transfer.

Driving Shaft	Driven Shaft		
	Planet	Sun	Carrier
Planet	N/A	η_{WG}	η_{WG}
Sun	η_{GW}	N/A	No loss
Carrier	η_{GW}	No loss	N/A

Geometric Surface Contact Friction

When you set **Friction model** to `Constant efficiency` and leave **Friction parameterization** set to `Friction coefficient and geometrical parameters`, the model considers geometric surface contact friction. In this case, η_{WG} and η_{GW} rely on:

- The worm-gear threading geometry, specified by lead angle λ and normal pressure angle α.
- The surface contact friction coefficient k.

$$\eta_{\mathrm{WG}} = \frac{(\cos\alpha - k \cdot \tan\lambda)}{\left(\cos\alpha + \frac{k}{\tan\lambda}\right)}$$

$$\eta_{\mathrm{GW}} = \frac{\left(\cos\alpha - \frac{k}{\tan\lambda}\right)}{(\cos\alpha + k \cdot \tan\alpha)}$$

Constant Efficiencies

When you set **Friction model** to `Constant efficiency` and set **Friction parameterization** to `Efficiencies`, or when you set **Friction model** to `Temperature-dependent efficiency`, the model treats the efficiencies as constant. In this case, you specify η_{WG} and η_{GW} independently of geometric details.

Self-Locking and Negative Efficiency

You can enable self-locking behavior by making the efficiency negative. Power cannot be transmitted from the sun gear to the worm or from the carrier to the worm unless some torque is applied to the

worm to release the train. In this case, the absolute value of the efficiency specifies the ratio at which the train is released. The smaller the train lead angle, the smaller the reverse efficiency.

Meshing Efficiency

The efficiencies, η, of meshing between the worm gear and planet gear are fully active only if the transmitted power is greater than the power threshold.

If the power is less than the threshold, the actual efficiency is automatically regularized to unity at zero velocity.

Viscous Friction Force

The viscous friction coefficients of the worm-carrier and sun-carrier bearings control the viscous friction torque experienced by the carrier from lubricated, nonideal gear threads. For details, see "Nonideal Gear Constraints" on page 1-465.

Variables

Use the **Variables** settings to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Assumptions and Limitations

- Gear inertia is assumed to be negligible.
- Gears are treated as rigid components.
- Coulomb friction slows down simulation. For more information, see "Adjust Model Fidelity".

Ports

Conserving

C — Planet gear carrier
rotational mechanical

Rotational mechanical conserving port associated with the planet gear carrier.

W — Worm gear
rotational mechanical

Rotational mechanical conserving port associated with the worm gear.

S — Sun gear
rotational mechanical

Rotational mechanical conserving port associated with the sun gear.

H — Heat flow
thermal

Thermal conserving port associated with heat flow. Heat flow affects the power transmission efficiency by altering the gear temperatures.

Dependencies

To enable this port, set **Friction model** to Temperature-dependent efficiency.

Parameters

Main

`Gear ratio` — Gear ratio
25 (default) | positive scalar

Gear or transmission ratio, R_{WG}, which signifies the worm gear angular velocity divided by the sun gear angular velocity. This gear ratio must be strictly positive.

`Worm thread type` — Thread rotation direction
`Right-hand` (default) | `Left-hand`

Direction of positive worm rotation. If you select `Left-hand`, positive worm rotation results in negative gear rotation.

Meshing Losses

The table shows how the options that you choose for **Friction model** affect the visibility of other parameters in the **Meshing Losses** tab. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Meshing Losses Parameter Dependencies

Meshing Losses Setting Parameters and Values				
Friction Model				
No meshing losses - Suitable for HIL simulation	Constant efficiency			Temperature-dependent efficiency
	Friction parameterization			**Temperature**
	Friction coefficient and geometrical parameters	Efficiencies		
	Normal pressure angle	**Worm-gear efficiency**		**Worm-gear efficiency**
	Lead angle	**Gear-worm efficiency**		**Gear-worm efficiency**
	Friction coefficient			
	Power threshold	**Power threshold**		**Power threshold**

`Friction model` — Friction model
`No meshing losses - Suitable for HIL simulation` (default) | `Constant efficiency` | `Temperature-dependent efficiency`

Friction model for the block:

- `No meshing losses - Suitable for HIL simulation` — Gear meshing is ideal.
- `Constant efficiency` — Transfer of torque between the gear wheel pairs is reduced by a constant efficiency, η, such that $0 < \eta \leq 1$.

- `Temperature-dependent efficiency` — Transfer of torque between the gear wheel pairs is defined by the table lookup based on the temperature.

Friction parameterization — Friction parameterization method
`Friction coefficient and geometrical parameters` (default) | `Efficiencies`

Characterization of the friction between gear threads:

- `Friction coefficient and geometrical parameters` — Friction is determined by geometric surface contact friction.
- `Efficiencies` — Friction is determined by constant efficiencies, where $0 < \eta < 1$.

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency`.

Normal pressure angle — Thread pressure angle
`17.5 deg` (default) | scalar

Thread pressure angle, α, in the normal plane. The value must be greater than zero and less than 90 degrees.

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency` and **Friction parameterization** to `Friction coefficient and geometrical parameters`.

Lead angle — Thread helix angle
`20 deg` (default) | positive scalar

Thread helix angle, where $\lambda = \arctan[L/(\pi d)]$. L is the worm lead and d is the worm pitch diameter. This value must be greater than zero.

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency` and **Friction parameterization** to `Friction coefficient and geometrical parameters`.

Friction coefficient — Thread friction
`0.08` (default) | positive scalar

Dimensionless coefficient of normal friction in the thread. Must be greater than zero.

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency` and **Friction parameterization** to `Friction coefficient and geometrical parameters`.

Temperature — Temperature
`[280, 300, 320]` K (default) | vector

Vector of temperatures used to construct a 1-D temperature-efficiency lookup table. The vector elements must increase from left to right.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Worm-gear efficiency — Torque transfer efficiency from the worm gear to the sun gear
`[.75, .65, .6]` or `0.74` (default) | vector or scalar

Vector of output-to-input power ratios that describe the power flow from the worm gear to the sun gear, η_{WG}. When you set **Friction model** to `Constant efficiency`, specify the value as a scalar. When you set **Friction model** to `Temperature-dependent efficiency`, specify the value as a vector. The block uses the vector values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set either:

- **Friction model** to `Constant efficiency` and **Friction parameterization** to `Efficiencies` — In this case, specify the value as a scalar.

- **Friction model** to `Temperature-dependent efficiency` — In this case, specify the value as a vector.

Gear-worm efficiency — Torque transfer efficiency from the sun gear to the worm gear
`[.5, .45, .4]` or `0.65` (default) | vector or scalar

Vector of output-to-input power ratios that describe the power flow from the sun gear to the worm gear, η_{GW}. When you set **Friction model** to `Constant efficiency`, specify the value as a scalar. When you set **Friction model** to `Temperature-dependent efficiency`, specify the value as a vector. The block uses the vector values to construct a 1-D temperature-efficiency lookup table.

Each element is an efficiency that relates to a temperature in the **Temperature** vector. The length of the vector must be equal to the length of the **Temperature** vector. Each element in the vector must be in the range (0,1].

Dependencies

To enable this parameter, set either:

- **Friction model** to `Constant efficiency` and **Friction parameterization** to `Efficiencies` — In this case, specify the value as a scalar.

- **Friction model** to `Temperature-dependent efficiency` — In this case, specify the value as a vector.

Power threshold — Minimum efficiency power threshold
`0.001` W (default) | scalar

Power threshold, p_{th}, above which full efficiency is in effect. Below this value, a hyperbolic tangent function smooths the efficiency factor.

When you set **Friction model** to `Constant efficiency`, the block lowers the efficiency losses to zero when no power is transmitted. When you set **Friction model** to `Temperature-dependent efficiency`, the function smooths the efficiency factors between zero when at rest and the values provided by the temperature-efficiency lookup tables at the power thresholds.

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency` or `Temperature-dependent efficiency`.

Viscous Losses

Worm-carrier and sun-carrier viscous friction coefficients — Gear viscous friction
`[0, 0] N*m/(rad/s)` (default) | vector

Vector of viscous friction coefficients [μ_{WC} μ_{SC}], for the worm-carrier and sun-carrier shafts, respectively.

Thermal Port

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

Thermal mass — Thermal mass
50 J/K (default) | scalar

Thermal energy required to change the component temperature by a single temperature unit. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

More About

Hardware-in-the-Loop Simulation

For optimal simulation performance, set **Friction model** to the default value, `No meshing losses - Suitable for HIL simulation`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Leadscrew | Sun-Planet | Sun-Planet Bevel | Worm Gear

Topics
"Basic Motion, Torque, and Force Modeling"
"Torsen Differential"
"Model Thermal Losses in Driveline Components"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

Synchronizer

Cone clutch, dog clutch, and translational detent assembled to provide smooth gear engagement
**Library:**   Simscape / Driveline / Clutches
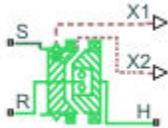
Description

The block represents a synchronizer that contains a dog clutch, a cone clutch, and a translational detent. The shift linkage first translates to engage the cone clutch. Frictional torque causes the shift linkage and cone clutch shaft to rotate at equal speed. When the force acting on the shift linkage exceeds the detent force, the dog clutch can engage.
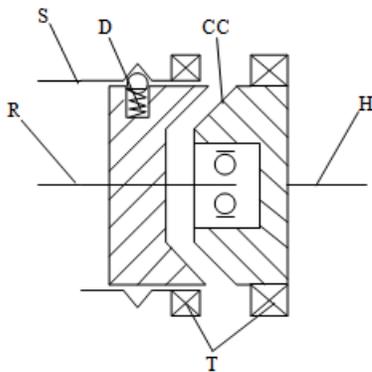
The schematic illustrates a synchronizer in the disengaged state. In this state, the ring (R) and hub (H) shafts can spin independently at different speeds. To synchronize ring and hub shaft speeds, the shift linkage (S) translates toward the hub shaft to engage the cone clutch. The friction surfaces of the cone clutch produce a frictional torque that equalizes the rotational speeds of the ring and hub shafts. The dog clutch teeth (T) can engage when the translational force acting on the shift linkage exceeds the peak detent force. The peak detent force should allow sufficient time and normal force to equalize ring and hub shaft speeds so that the dog clutch can engage.
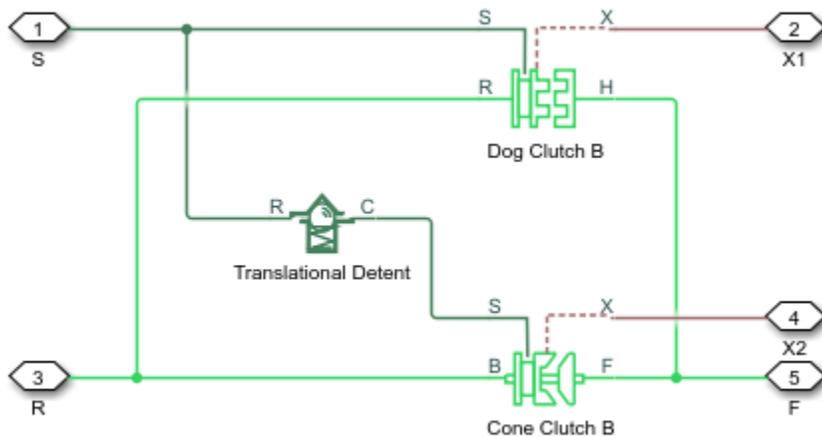
The model implements the Dog Clutch, Cone Clutch, and Translational Detent blocks. Refer to each block reference page for more information on the corresponding block function. You can use a similar approach to model customized versions of the synchronizer. One example is the Transmission (Detailed) subsystem in the model sdl_vehicle_manual_transmission.

Connections R and H are mechanical rotational conserving ports that represent the ring (R) and hub (H), respectively. Connection S is a mechanical translational conserving port that represents the ring shifter handle.

Connections X1 and X2 are physical signal outputs that report the shift linkage positions of the dog clutch and cone clutch, respectively. The shift linkage positions are zero when the clutch is fully disengaged. When the dog clutch is fully engaged, the dog clutch shift linkage position has a magnitude equal to the sum of the dog clutch ring-hub gap and the tooth height. When cone clutch is fully engaged, the cone clutch shift linkage position has a magnitude equal to the cone clutch ring-hub gap.

Thermal Modeling

You can model the effects of heat flow and temperature change through an optional thermal conserving port. By default, the thermal port is hidden. To expose the thermal port, in the **Clutch** settings, select a temperature-dependent setting tor the **Friction model** parameter. Specify the associated thermal parameters for the component.

Assumptions and Limitations

- The model does not account for inertia effects. You can add a Simscape Inertia block at each port to add inertia to the synchronizer model.

Ports

Output

X1 — Dog clutch translation
physical signal

Physical signal output port that measures the magnitude of the dog clutch translation.

X2 — Cone clutch translation
physical signal

Physical signal output port that measures the magnitude of the cone clutch translation.

Conserving

H — Clutch hub
mechanical rotational

Mechanical rotational conserving port associated with the clutch hub

R — Clutch ring
mechanical rotational

Mechanical rotational conserving port associated with the clutch ring.

S — Shift linkage
mechanical rotational

Mechanical rotational conserving port associated with shift linkage.

T — Heat flow
thermal

Thermal conserving port associated with heat flow.

Dependencies

This port is visible only when, in the **Friction** settings, the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`.

Parameters

The table shows how the specified options for parameters in both the **Cone Clutch** and **Dog Clutch** settings affect the visibility of:

- Parameters in the **Cone Clutch**, **Dog Clutch**, and **Initial Conditions** settings
- **Thermal Port** settings
- Thermal port **T**

To learn how to read the table, see "Parameter Dependencies" on page B-2.

Synchronizer Block Parameter Dependencies

Settings	Parameters and Options			
Cone Clutch	Contact surface maximum diameter			
	Contact surface minimum diameter			
	Cone half angle			
	Friction model			
	`Fixed kinetic friction coefficient`	`Velocity-dependent kinetic friction coefficient`	`Temperature-dependent friction coefficients`	`Temperature and velocity-dependent friction coefficients`
	-	-	Exposes: • Conserving port **T** • **Thermal Port** settings	Exposes: • Conserving port **T** • **Thermal Port** settings
	-	**Relative velocity vector**		**Relative velocity vector**
	-	-	**Temperature vector**	**Temperature vector**
	Static friction coefficient	**Static friction coefficient vector**	**Static friction coefficient vector**	**Static friction coefficient matrix**
	Kinetic friction coefficient	**Kinetic friction coefficient vector**	**Kinetic friction coefficient vector**	**Kinetic friction coefficient matrix**
	-	**Friction coefficient interpolation method**	**Friction coefficient interpolation method**	**Friction coefficient interpolation method**
	-	**Friction coefficient extrapolation method**	**Friction coefficient extrapolation method**	**Friction coefficient extrapolation**
	Velocity tolerance	**Velocity tolerance**	**Velocity tolerance**	**Velocity tolerance**
	Threshold force	**Threshold force**	**Threshold force**	**Threshold force**

Settings	Parameters and Options					
Dog Clutch	**Torque transmission model**		**Torque transmission model**		-	-
	Friction clutch approximat ion - Suitable for HIL and linearizat ion	Dynamic with backlash	Friction clutch approximat ion - Suitable for HIL and linearizat ion	Dynamic with backlash	-	-
	-	-	-	-	**Temperatur e vector**	**Temperatur e vector**
	Maximum transmitted torque	-	**Maximum transmitted torque**	-	**Maximum transmitted torque vector**	**Maximum transmitted torque vector**
	-	-	-	-	**Interpolatio n method**	**Interpolatio n method**
	-	-	-	-	**Extrapolati on method**	**Extrapolati on method**
	Clutch teeth mean radius	**Clutch teeth mean radius**	**Clutch teeth mean radius**	**Clutch teeth mean radius**	**Clutch teeth mean radius**	**Clutch teeth mean radius**
	-	**Number of teeth**	-	**Number of teeth**		
		Rotational backlash		**Rotational backlash**		
	-	**Torsional stiffness**	-	**Torsional stiffness**		
	-	**Torsional damping**	-	**Torsional damping**		
	-	**Tooth-tooth friction coefficient**	-	**Tooth-tooth friction coefficient**		
Initial Conditions	**Initial state**	**Initial state**	**Initial state**	**Initial state**	**Initial state**	**Initial state**
	Dog clutch initial shift linkage position	**Dog clutch initial shift linkage position**	**Dog clutch initial shift linkage position**	**Dog clutch initial shift linkage position**	**Dog clutch initial shift linkage position**	**Dog clutch initial shift linkage position**
	Cone clutch initial shift linkage position	**Cone clutch initial shift linkage position**	**Cone clutch initial shift linkage position**	**Cone clutch initial shift linkage position**	**Cone clutch initial shift linkage position**	**Cone clutch initial shift linkage position**

Settings	Parameters and Options					
	-	Initial dog clutch ring-hub offset angle	-	Initial dog clutch ring-hub offset angle	-	-
Thermal Port	-	-	-	-	Thermal mass	Thermal mass
	-	-	-	-	Initial temperature	Initial temperature

Cone Clutch

Contact surface maximum diameter — Outer diameter
150 mm (default) | positive scalar

Outer conical diameter d_o.

Contact surface minimum diameter — Inner diameter
100 mm (default) | positive scalar

Inner conical diameter d_i.

Cone half angle — Cone half angle
12 deg (default) | positive scalar

Half opening angle α of the cone geometry.

Friction model — Friction model
Fixed kinetic friction coefficient (default) | Velocity-dependent kinetic friction coefficient | Temperature-dependent friction coefficients | Temperature and velocity-dependent friction coefficients

Parameterization method to model the kinetic friction coefficient. The options and default values for this parameter depend on the friction model that you select for the block. The options are:

- Fixed kinetic friction coefficient — Provide a fixed value for the kinetic friction coefficient.
- Velocity-dependent kinetic friction coefficient — Define the kinetic friction coefficient by one-dimensional table lookup based on the relative angular velocity between disks.
- Temperature-dependent friction coefficients — Define the kinetic friction coefficient by table lookup based on the temperature.
- Temperature and velocity-dependent friction coefficients — Define the kinetic friction coefficient by table lookup based on the temperature and the relative angular velocity between disks.

Dependencies

The friction model setting affects the visibility of other parameters, settings, and ports.

For more information, see Synchronizer Block Parameter Dependencies.

Relative velocity vector — Relative velocity
[0, 100, 1000] rad/s (default) | vector

Input values for the relative velocity as a vector. The values in the vector must increase from left to right. The minimum number of values depends on the interpolation method that you select. For linear interpolation, provide at least two values per dimension. For smooth interpolation, provide at least three values per dimension.

Dependencies

This parameter is visible only when the **Friction model** parameter is set to Velocity-dependent kinetic friction coefficient or Temperature and velocity-dependent friction coefficients.

For more information, see Synchronizer Block Parameter Dependencies.

Temperature vector — Temperature
[280, 300, 320] K (default) | increasing vector

Input values for the temperature as a vector. The minimum number of values depends on the interpolation method that you select. For linear interpolation, provide at least two values per dimension. For smooth interpolation, provide at least three values per dimension. The values in the vector must increase from left to right.

Dependencies

This parameter is visible only when the **Friction model** parameter is set to Temperature-dependent friction coefficients or Temperature and velocity-dependent friction coefficients.

For more information, see Synchronizer Block Parameter Dependencies.

Static friction coefficient — Static friction coefficient
0.35 (default) | scalar

Static or peak value of the friction coefficient. The static friction coefficient must be greater than the kinetic friction coefficient.

Dependencies

this parameter is visible only when the **Friction model** parameter is set to Fixed kinetic friction coefficient or Velocity-dependent kinetic friction coefficient.

For more information, see Synchronizer Block Parameter Dependencies.

Static friction coefficient vector — Static friction coefficient
[.4, .38, .36] (default) | vector

Static, or peak, values of the friction coefficient as a vector. The vector must have the same number of elements as the temperature vector. Each value must be greater than the value of the corresponding element in the kinetic friction coefficient vector.

Dependencies

This parameter is visible only when the **Friction model** parameter is set to Temperature-dependent friction coefficients or Temperature and velocity-dependent friction coefficients.

For more information, see Synchronizer Block Parameter Dependencies.

Kinetic friction coefficient — Kinetic friction coefficient
`0.3` (default) | positve scalar

The kinetic, or Coulomb, friction coefficient. The coefficient must be greater than zero.

Dependencies

This parameter is visible only when the **Friction model** parameter is set to `Fixed kinetic friction coefficient`.

For more information, see Synchronizer Block Parameter Dependencies.

Kinetic friction coefficient vector — Kinetic friction coefficient
`[.3, .22, .19]` (default) | `[.3, .28, .25]` | vector

Output values for kinetic friction coefficient as a vector. All values must be greater than zero.

If the **Friction model** parameter is set to

- `Velocity-dependent kinetic friction coefficient` — The vector must have same number of elements as relative velocity vector.
- `Temperature-dependent friction coefficients` — The vector must have the same number of elements as the temperature vector.

Dependencies

This parameter is visible only when the **Friction model** parameter is set to `Velocity-dependent kinetic friction coefficient` or `Temperature-dependent friction coefficients`.

For more information, see Synchronizer Block Parameter Dependencies.

Kinetic friction coefficient matrix — Kinetic friction coefficient
`[.34, .32, .3; .3, .28, .25; .25, .2, .15]` (default) | matrix

Output values for kinetic friction coefficient as a matrix. All the values must be greater than zero. The size of the matrix must equal the size of the matrix that is the result of the temperature vector × the kinetic friction coefficient relative velocity vector.

Dependencies

This parameter is visible only when the **Friction model** parameter is set to `Temperature and velocity-dependent friction coefficients`.

For more information, see Synchronizer Block Parameter Dependencies.

Friction coefficient interpolation method — Interpolation method
`Linear` (default) | `Smooth`

Interpolation method for approximating the output value when the input value is between two consecutive grid points:

- `Linear` — Select this option to get the best performance.
- `Smooth` — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is visible only when, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Velocity-dependent kinetic friction coefficient`, `Temperature-dependent friction coefficients`, or `Temperature and velocity-dependent friction coefficients`.

For more information, see Synchronizer Block Parameter Dependencies.

`Friction coefficient extrapolation method` — **Extrapolation method**
`Linear` (default) | `Nearest` | `Error`

Extrapolation method for determining the output value when the input value is outside the range specified in the argument list:

- `Linear` — Select this option to produce a curve with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region.
- `Nearest` — Select this option to produce an extrapolation that does not go above the highest point in the data or below the lowest point in the data.
- `Error` — Select this option to avoid going into the extrapolation mode when you want your data to be within the table range. If the input signal is outside the range of the table, the simulation stops and generates an error.

For more information on extrapolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is visible only when, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Velocity-dependent kinetic friction coefficient`, `Temperature-dependent friction coefficients`, or `Temperature and velocity-dependent friction coefficients`.

For more information, see Synchronizer Block Parameter Dependencies.

`Velocity tolerance` — **Relative velocity locking threshold**
`0.001 rad/s` (default) | scalar

Relative velocity below which the two surfaces can lock. The surfaces lock if the torque is less than the product of the effective radius, the static friction coefficient, and the applied normal force.

`Threshold force` — **Normal contact force threshold**
`1 N` (default) | scalar

The normal force is applied only if the amount of force exceeds the value of the **Threshold force** parameter. Forces below the **Threshold force** are not applied so there is no transmitted frictional torque.

Dog Clutch

The methods that are available for parameterizing the torque transmission depend whether the friction model is temperature-dependent.

The friction model is determined, in the **Cone Clutch** settings, by the **Friction model** parameter setting:

- `Fixed kinetic friction coefficient` — Temperature independent
- `Velocity-dependent kinetic friction coefficient` — Temperature independent
- `Temperature-dependent friction coefficients` — Temperature dependent
- `Temperature and velocity-dependent friction coefficients` — Temperature dependent

For a temperature-independent model, parameterize the block using one of the options for the **Torque Transmission Model** parameter.

Torque transmission model — Torque transmission model
`Friction clutch approximation — Suitable for HIL and linearization` (default) | `Dynamic with backlash`

Computational framework for modeling the dynamic behavior of the dog clutch:

- `Friction clutch approximation — Suitable for HIL and linearization` — Model clutch engagement as a friction phenomenon between the ring and the hub. This model, based on the Fundamental Friction Clutch block, provides a computationally efficient approximation of the dog clutch.
- `Dynamic with backlash` — Model clutch engagement in detail, accounting for such phenomena as backlash, torsional compliance, and contact forces between ring and hub teeth.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Fixed kinetic friction coefficient` or `Velocity-dependent friction coefficients`.

The visibility of related parameters in the **Dog Clutch** and **Initial Conditions** settings is affected by the option that you select for this parameter.

For more information, see Synchronizer Block Parameter Dependencies.

Temperature vector — Temperature
`[280, 300, 320]` K (default) | increasing vector

Input values for the temperature as a vector. The minimum number of values depends on the interpolation method that you select. For linear interpolation, provide at least two values per dimension. For smooth interpolation, provide at least three values per dimension. The values in the vector must increase from left to right.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`.

For more information, see Synchronizer Block Parameter Dependencies.

Maximum transmitted torque — Maximum transmitted torque
`1000` N*m (default) | positive scalar

Largest torque that the clutch can transmit, corresponding to a nonslip engaged configuration. If the torque transmitted between the ring and the hub exceeds this value, the two components begin to slip with respect to each other. This torque determines the static friction limit in the friction clutch approximation

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Fixed kinetic friction coefficient` or `Velocity-dependent kinetic friction coefficient` and, in **Dog Clutch** settings, the **Torque transmission model** parameter is set to `Friction clutch approximation - Suitable for HIL and linearization`.

For more information, see Synchronizer Block Parameter Dependencies.

Maximum transmitted torque vector — Maximum transmitted torque
`[1000, 1050, 1000]` N*m (default) | vector

Largest torque that the clutch can transmit, corresponding to a nonslip engaged configuration, specified as a vector. If the torque transmitted between the ring and the hub exceeds this value, the two components begin to slip with respect to each other. This torque determines the static friction limit in the friction clutch approximation. The vector has the same number of elements as the temperature vector.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Temperature-dependent kinetic friction coefficient` or `Temperature and velocity-dependent kinetic friction coefficient`.

For more information, see Synchronizer Block Parameter Dependencies.

Interpolation method — Interpolation method
`Linear` (default) | `Smooth`

Interpolation method for approximating the output value when the input value is between two consecutive grid points:

- `Linear` — Select this option to get the best performance.
- `Smooth` — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Temperature-dependent kinetic friction coefficient` or `Temperature and velocity-dependent kinetic friction coefficient`.

For more information, see Synchronizer Block Parameter Dependencies.

Extrapolation method — Extrapolation method
`Linear` (default) | `Nearest` | `Error`

Extrapolation method for determining the output value when the input value is outside the range specified in the argument list:

- `Linear` — Select this option to produce a curve with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region.
- `Nearest` — Select this option to produce an extrapolation that does not go above the highest point in the data or below the lowest point in the data.
- `Error` — Select this option to avoid going into the extrapolation mode when you want your data to be within the table range. If the input signal is outside the range of the table, the simulation stops and generates an error.

For more information on extrapolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Temperature-dependent kinetic friction coefficient` or `Temperature and velocity-dependent kinetic friction coefficient`.

For more information, see Synchronizer Block Parameter Dependencies.

Clutch teeth mean radius — Clutch teeth mean radius
50 mm (default) | positive scalar

Distance from the ring or hub center to the corresponding tooth center. The mean tooth radius determines the normal contact forces between ring and hub teeth given the transmission torque between the two components. The value must be greater than zero.

Number of teeth — Number of ring or hub teeth
6 (default) | positive nonzero scalar integer

Total number of teeth in the ring or the hub. The two components have equal tooth numbers. The value must be greater than or equal to one.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Fixed kinetic friction coefficient` or `Velocity-dependent kinetic friction coefficient` and, in the **Dog Clutch** settings, the **Torque transmission model** parameter is set to `Dynamic with backlash`.

For more information, see Synchronizer Block Parameter Dependencies.

Rotational backlash — Rotational backlash
10 deg (default) | positive scalar

Allowable angular motion, or play, between the ring and hub teeth in the engaged clutch configuration. The value must be greater than zero.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Fixed kinetic friction coefficient` or `Velocity-dependent kinetic friction coefficient` and, in the **Dog Clutch** settings, the **Torque transmission model** parameter is set to `Dynamic with backlash`.

For more information, see Synchronizer Block Parameter Dependencies.

Torsional stiffness — Torsional stiffness
10e6 N*m/rad (default) | positive scalar

Linear torsional stiffness coefficient at the contact interface between the ring and hub teeth. This coefficient characterizes the restoring component of the contact force between the two sets of teeth. Greater stiffness values correspond to greater contact forces. The value must be greater than zero. The default value is 10e6 N*m/rad.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to Fixed kinetic friction coefficient or Velocity-dependent kinetic friction coefficient and, in the **Dog Clutch** settings, the **Torque transmission model** parameter is set to Dynamic with backlash.

For more information, see Synchronizer Block Parameter Dependencies.

Torsional damping — Torsional damping
100 N*m/(rad/s) (default) | positive scalar

Linear torsional damping coefficient at the contact interface between the ring and hub teeth. This coefficient characterizes the dissipative component of the contact force between the two sets of teeth. Greater damping values correspond to greater energy dissipation during contact. The value must be greater than zero.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to Fixed kinetic friction coefficient or Velocity-dependent kinetic friction coefficient and, in the **Dog Clutch** settings, the **Torque transmission model** parameter is set to Dynamic with backlash.

For more information, see Synchronizer Block Parameter Dependencies.

Tooth-tooth friction coefficient — Tooth-tooth contact kinetic friction coefficient
0.05 (default) | positive scalar

Kinetic friction coefficient at the contact interface between ring and hub teeth. This coefficient characterizes the dissipative force that resists shift linkage motion due to tooth-tooth contact during clutch engagement/disengagement.

Greater coefficient values correspond to greater energy dissipation during shift linkage motion. The value must be greater than zero.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to Fixed kinetic friction coefficient or Velocity-dependent kinetic friction coefficient and, in the **Dog Clutch** settings, the **Torque transmission model** parameter is set to Dynamic with backlash.

For more information, see Synchronizer Block Parameter Dependencies.

Detent

`Peak force` — Peak shear force
`500 N` (default) | nonnegative scalar

Peak shear force of the detent.

`Notch width` — Notch width
`3 mm` (default) | nonnegative scalar

Width of the region where the detent exhibits shear force.

`Viscous friction coefficient` — Viscous friction coefficient
`0.1 N/(m/s)` (default) | nonnegative scalar

Viscous friction coefficient at the contact surface of the detent. The value must be greater than or equal to zero.

`Friction to peak force ratio` — Friction to peak force ratio
`0.01` (default) | nonnegative scalar

Ratio of the kinetic friction to the peak shear force of the detent. The parameter is used to set the value of the kinetic friction. The parameter must be greater than or equal to zero.

`Friction velocity threshold` — Kinetic friction velocity threshold
`0.05 m/s` (default)

Velocity required for peak kinetic friction at the contact surface of the detent. The parameter ensures the force is continuous when the travel direction changes, increasing the numerical stability of the simulation. The parameter must be greater than zero. The default value is `0.05 m/s`.

Shift Linkage

`Shift linkage travel direction` — Engagement condition travel direction
`Positive shift linkage displacement engages clutch` (default) | `Negative shift linkage displacement engages clutch`

Direction the shift linkage must travel in to engage the clutch. Choices include positive and negative displacements.

`Maximum engagement speed` — Engagement speed upper velocity threshold
`inf` (default) | positive scalar

Relative angular velocity between the ring and the hub above which the clutch cannot engage. The value is specific to the specific gearbox or transmission. Minimizing the value helps avoid high dynamic impact during engagement. The value must be greater than zero.

`Tooth overlap to engage` — Tooth overlap engagement threshold
`3 mm` (default) | positive scalar

Overlap length between ring and hub teeth along the common longitudinal axis above which the clutch can engage. The clutch remains disengaged until the tooth overlap by at least this length. The value must be greater than zero.

`Tooth height` — Tooth height
`10 mm` (default) | positive scalar

Distance between the base and crest of a tooth. Ring and hub teeth share the same height. The tooth height and the ring-hub clearance when fully disengaged determine the maximum travel span of the shift linkage. The value must be greater than zero.

Ring-hub clearance when disengaged — Ring-hub clearance when disengaged
3 mm (default) | positive scalar

Maximum open gap between the ring and hub tooth crests along the shift linkage translation axis. This gap corresponds to the fully disengaged clutch state. The tooth height and the ring-hub clearance when fully disengaged determine the maximum travel span of the shift linkage. The value must be greater than zero.

Hard stop at back of shift linkage — Hard stop model
On (default) | Off

Hard stop that prevents the shift linkage from traveling beyond the fully disengaged position:

- On — Hard stop when fully disengaged.
- Off — No hard stop when fully disengaged.

Dog clutch ring stop stiffness — Dog clutch ring stop stiffness
10e5 N/m (default) | positive scalar

Stiffness of the hard stops on both sides of the dog clutch ring. The model assumes the ring and stops behave elastically. Contact deformation is proportional to the applied force and the reciprocal of the contact stiffness. The value of the stiffness must be assigned with reference to the parameter **Tooth overlap to engage**. Too low a stiffness could cause the deformation to exceed the required overlap and initiate a false engagement. The parameter must be greater than zero.

Cone clutch ring stop stiffness — Cone clutch ring stop stiffness
10e5 N/m (default) | positive scalar

Stiffness of the hard stops on both sides of the cone clutch ring. The model assumes the ring and stops behave elastically. Contact deformation is proportional to the applied force and the reciprocal of the contact stiffness.

Dog clutch ring stop damping — Dog clutch ring stop damping
1e3 N/(m/s) (default) | nonegative scalar

Translational contact damping between the dog clutch ring and the hub. The value of the damping is inversely proportional to the number of oscillations that occur after impact. The parameter must be greater than zero.

Cone clutch ring stop damping — Cone clutch ring stop damping
1e3 N/(m/s) (default) | nonegative scalar

Translational contact damping between the cone clutch ring and the hub. The value of damping is inversely proportional to the number of oscillations that occur after impact. The parameter must be greater than zero.

Shift linkage viscous friction coefficient — Shift linkage viscous friction coefficient
100 N/(m/s) (default) | positive scalar

Viscous friction coefficient for the relative translational motion between the hub and the ring. The value of the parameter depends on lubrication state and quality of contacting surfaces. The coefficient must be greater than or equal to zero.

Initial Conditions

Initial state — Initial clutch state
All clutches unlocked (default) | Cone clutch locked | All clutches locked

Beginning configuration of cone and dog clutches:

- All clutches unlocked — Cone and dog clutches transmit zero torque between the ring and hub shafts.
- Cone clutch locked — Cone clutch transmits torque between the ring and hub shafts.
- All clutches locked — Cone and dog clutches transmit torque between the ring and hub shafts.

Dog clutch initial shift linkage position — Dog clutch initial shift linkage position
0 mm (default) | scalar

Initial position of the shift linkage section that attaches to the dog clutch. The value of the parameter has these restrictions:

Linkage Travel Direction	Dog Clutch State	Parameter Restriction
Positive shift linkage displacement engages clutch	Initially engaged	Parameter must be greater than the sum of parameters **Ring-hub clearance when dog clutch disengaged** and **Tooth overlap to engage**
	Initially disengaged	Parameter must be smaller than the sum of parameters **Ring-hub clearance when dog clutch disengaged** and **Tooth overlap to engage**
Negative shift linkage displacement engages clutch	Initially engaged	Negative of the parameter must be greater than the sum of parameters **Ring-hub clearance when dog clutch disengaged** and **Tooth overlap to engage**
	Initially disengaged	Negative of the parameter must be smaller than the sum of parameters **Ring-hub clearance when dog clutch disengaged** and **Tooth overlap to engage**

Cone clutch initial shift linkage position — Cone clutch initial shift linkage position
0 mm (default) | scalar

Initial position of the shift linkage section that attaches to the cone clutch. The value of the parameter has these restrictions:

Linkage Travel Direction	Dog Clutch State	Parameter Restriction
Positive shift linkage displacement engages clutch	Initially engaged	Parameter must be greater than the value of **Ring-hub clearance when cone clutch disengaged**
	Initially disengaged	Parameter must be smaller than the value of **Ring-hub clearance when cone clutch disengaged**
Negative shift linkage displacement engages clutch	Initially engaged	Negative of the parameter must be greater than the value of **Ring-hub clearance when dog cone disengaged**
	Initially disengaged	Negative of the parameter must be smaller than the value of **Ring-hub clearance when dog cone disengaged**

Initial dog clutch ring-hub offset angle — Initial dog clutch ring-hub offset angle
0 deg (default) | scalar

Rotation angle between the ring and the hub at simulation time zero. This angle determines whether the ring and hub teeth can interlock, and hence whether the clutch can engage. The initial offset angle must satisfy these conditions:

- If the clutch initial state is disengaged, the initial offset angle must fall in the range

$$-\frac{180°}{N} \leq \phi_0 \leq +\frac{180°}{N},$$

  where N is the number of teeth present in the ring or the hub. The two components contain the same number of teeth.

- If the clutch initial state is engaged, the initial offset angle must fall in the range

$$-\frac{\delta}{2} \leq \phi_0 \leq +\frac{\delta}{2},$$

  where δ is the backlash angle between the ring and hub teeth.

Dependencies

This parameter is visible only if both of these conditions are met:

- In the **Cone Clutch** settings, the **Friction model** parameter is set to Fixed kinetic friction coefficient or Velocity-dependent kinetic friction coefficient.
- In the **Dog Clutch** settings, the **Torque transmission model** is set to Dynamic with backlash.

Thermal Port

Thermal Port settings are visible only when, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see Synchronizer Block Parameter Dependencies.

Thermal mass — Thermal mass
50 kJ/K (default) | scalar

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see Synchronizer Block Parameter Dependencies.

Initial temperature — Initial temperature
300 K (default) | scalar

Component temperature at the start of simulation. The initial temperature alters the component efficiency according to an efficiency vector that you specify, affecting the starting meshing or friction losses.

Dependencies

This parameter is visible only if, in the **Cone Clutch** settings, the **Friction model** parameter is set to `Temperature-dependent friction coefficients` or `Temperature and velocity-dependent friction coefficients`. For more information, see Synchronizer Block Parameter Dependencies.

More About

Linearization

To optimize your model for linearization, use the **Dog Clutch > Torque transmission model** parameter default setting, `Friction clutch approximation - Suitable for HIL and linearization`.

Hardware-in-the-Loop Simulation

For optimal simulation performance, use the **Dog Clutch > Torque transmission model** parameter default setting, `Friction clutch approximation - Suitable for HIL and linearization`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Cone Clutch | Dog Clutch | Double-Sided Synchronizer | Translational Detent

Topics
"Vehicle with Manual Transmission"
"Model Thermal Losses in Driveline Components"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2012b

Tire (Friction Parameterized)

Tire with friction parameterized in terms of static and kinetic coefficients
Library: Simscape / Driveline / Tires & Vehicles

Description

The Tire (Friction Parameterized) block models a tire with friction parameterized in terms of static and kinetic coefficients. The static friction coefficient determines the applied torque at which the tire loses traction and begins to slip. The kinetic friction coefficient determines the amount of torque that the tire transmits to the pavement once it begins to slip. The tire regains traction once its relative velocity over the pavement falls below the traction velocity tolerance specified.

To increase the fidelity of the tire model, the block enables you to specify properties such as tire compliance, inertia, and rolling resistance. However, these properties increase the complexity of the tire model and can slow down simulation. Consider ignoring tire compliance and inertia if simulating the model in real time or if preparing the model for hardware-in-the-loop (HIL) simulation.

The traction state model of this block is based on the traction state model of the Fundamental Friction Clutch block. For more information on the state model, see "Clutch States" on page 1-235.

Ports

Input

N — Normal force
physical signal

Physical signal input port associated with the normal force acting on the tire. The normal force is positive if it acts downward on the tire, pressing it against the pavement.

M — Friction coefficients
physical signal | vector | $[\mu_s, \mu_k]$

Physical signal input port associated with the static (μ_s) and kinetic (μ_k) friction coefficients. Provide the friction coefficients as a two-element vector, specified in the order $[\mu_s, \mu_k]$.

Dependencies

Port **M** is exposed only if the **Main > Friction model** parameter is set to `Physical signal friction coefficients`. For more information, see Main Parameter Dependencies.

Output

S — Slip
physical signal

Physical signal output port associated with the relative slip between the tire and road.

Conserving

A — Axle
mechanical rotational

Mechanical rotational port associated with the axle that the tire sits on.

H — Hub
mechanical translational

Mechanical translational port associated with the wheel hub that transmits the thrust generated by the tire to the remainder of the vehicle.

Parameters

Main

Specify characteristics of the tire, such as the rolling radius or static friction coefficient. The table shows how the visibility of some **Main** parameters depends on the options that you choose for other parameters. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Main Parameter Dependencies

Main		
Rolling radius		
Friction model — Choose `Fixed kinetic friction coefficient`, `Table lookup kinetic friction coefficient`, or `Physical signal friction coefficients`		
Fixed kinetic friction coefficient	**Table lookup kinetic friction coefficient**	**Physical signal friction coefficients** — Exposes physical signal input port **M** for providing the static (μ_s) and kinetic (μ_k) friction coefficients to the block as an array of two elements in the order [μ_s, μ_k].
Static friction coefficient		
Kinetic friction coefficient	Tire slip vector	
	Kinetic friction coefficient vector	
	Interpolation method	
	Extrapolation method	

Rolling radius — Unloaded tire-wheel radius
`0.3 m` (default) | positive scalar

Distance between the pavement and the center of the tire.

Friction model — Friction model
`Fixed kinetic friction coefficient` (default) | `Table lookup kinetic friction coefficient` | `Physical signal friction coefficients`

The block provides three friction models. The default model, `Fixed kinetic friction coefficient`, uses constant static and kinetic friction coefficients that you specify.

To specify friction using table lookup, set the **Friction model** parameter to `Table lookup kinetic friction coefficient`. While this model treats the static coefficient as a constant, it

treats the kinetic coefficient as a constant or function of tire slip. Use this model to model tire dynamics under constant pavement conditions.

To model tire dynamics under variable pavement conditions, set the **Friction model** parameter to `Physical signal friction coefficients`. Selecting this model exposes physical signal inport **M**. Use the **M** port to provide the static (μ_s) and kinetic (μ_k) friction coefficients to the block as an array of two elements in the order [μ_s, μ_k].

Dependencies

Each friction model option exposes related parameters and hides unrelated parameters. Selecting `Physical signal friction coefficients` exposes physical signal input port **M**. For more information, see Main Parameter Dependencies.

Static friction coefficient — Static friction coefficient
`0.90` (default) | positive scalar

Ratio of the allowable longitudinal force to the normal force allowed before the tire begins to slip (μ_s). The parameter must be greater than either the kinetic friction coefficient or the largest value in the kinetic friction coefficient vector.

Dependencies

This parameter is visible when the **Friction model** parameter is set to `Fixed kinetic friction coefficient` or `Table lookup kinetic friction coefficient`. For more information, see Main Parameter Dependencies.

Kinetic friction coefficient — Kinetic friction coefficient
`0.70` (default) | positive scalar

Ratio of the transmitted longitudinal force to the normal force allowed during tire slip (μ_k). The ratio must be greater than zero.

Dependencies

This parameter is visible when the **Friction model** parameter is set to `Fixed kinetic friction coefficient`. For more information, see Main Parameter Dependencies.

Tire slip vector — Tire slip vector
`[0, .02, .06, .15, .6, 1]` rad/s (default) | vector

Vector of tire slip values that correspond to the kinetic friction coefficients in the **Kinetic friction coefficient vector** parameter. The vectors must be the same size. If the **Tire slip vector** parameter contains only nonnegative values, the slip versus friction function is assumed to be symmetric about the slip axis.

Dependencies

This parameter is visible when the **Friction model** parameter is set to `Table lookup kinetic friction coefficient`. For more information, see Main Parameter Dependencies.

Kinetic friction coefficient vector — Kinetic friction coefficient
`[.89, .88, .8, .75, .7, .7]` (default) | vector

Vector of kinetic friction coefficients that correspond to the tire slip values specified in the **Tire slip vector** parameter. The vectors must be the same size.

Dependencies

This parameter is visible when the **Friction model** parameter is set to `Table lookup kinetic friction coefficient`. For more information, see Main Parameter Dependencies.

`Interpolation method` — Interpolation method
`Linear` (default) | `Smooth`

Interpolation method for the lookup table to use for processing the tire slip-kinetic friction coefficient characteristic. To prioritize performance, select `Linear`. To produce a continuous curve with continuous first-order derivatives, select `Smooth`.

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is visible when the **Friction model** parameter is set to `Table lookup kinetic friction coefficient`. For more information, see Main Parameter Dependencies.

`Extrapolation method` — Extrapolation method
`Linear` (default) | `Nearest` | `Error`

Extrapolation method for the lookup table to use for processing the tire slip-kinetic friction coefficient characteristic. To produce:

- A curve with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region, select `Linear`.
- An extrapolation that does not go above the highest point in the data or below the lowest point in the data, select `Nearest`.
- An error if the input signal is outside the range of the table, select `Error`. This option ensures that simulation occurs only if your data is within the table range.

For more information on extrapolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is visible when the **Friction model** parameter is set to `Table lookup kinetic friction coefficient`. For more information, see Main Parameter Dependencies.

Dynamics

The table shows how the visibility of some parameters depends on the options that you choose for other parameters. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Dynamics Parameter Dependencies Table

Dynamics	
Compliance — Choose `No compliance - Suitable for HIL simulation` or `Specify stiffness and damping`	
No compliance - Suitable for HIL simulation	**Specify stiffness and damping**
	Longitudinal stiffness
	Longitudinal damping
Inertia — Choose `No Inertia` or `Specify inertia and initial velocity`	
No Inertia	**Specify inertia and initial velocity**
	Tire inertia
	Initial velocity

Compliance — Dynamical compliance model
`No compliance - Suitable for HIL simulation` (default) | `Specify stiffness and damping`

Model for the dynamical compliance of the tire.

- `No compliance - Suitable for HIL simulation` — Tire is modeled with no dynamical compliance.
- `Specify stiffness and damping` — Tire is modeled as a stiff, dampened spring and deforms under load.

Dependencies

Selecting the `Specify stiffness and damping` parameterization method, exposes stiffness and damping parameters. For more information, see Dynamics Parameter Dependencies Table.

Longitudinal stiffness — Longitudinal stiffness
`1e6 N/m` (default) | positive scalar

Tire longitudinal stiffness C_{Fx}.

Dependencies

Selecting `Specify stiffness and damping` for the **Compliance** parameter, exposes this parameter. For more information, see Dynamics Parameter Dependencies Table.

Longitudinal damping — Longitudinal damping
`1000 N/(m/s)` (default) | positive scalar

Tire longitudinal damping b_{Fx}.

Dependencies

Selecting `Specify stiffness and damping` for the **Compliance** parameter, exposes this parameter. For more information, see Dynamics Parameter Dependencies Table.

Inertia — Inertia model
`No inertia` (default) | `Specify inertia and initial velocity`

Model for the rotational inertia of the tire.

- `No inertia` — Tire is modeled with no dynamical compliance.
- `Specify inertia and initial velocity` — Tire is modeled as a stiff, dampened spring and deforms under load.

Dependencies

Selecting the `Specify inertia and initial velocity` parameterization method, exposes inertia and velocity parameters. For more information, see Dynamics Parameter Dependencies Table.

Tire inertia — Rotational inertia
1 kg*m^2 (default) | positive scalar

Rotational inertia I_w of the wheel-tire assembly.

Dependencies

Selecting `Specify inertia and initial velocity` for the **Inertia** parameter, exposes this parameter. For more information, see Dynamics Parameter Dependencies Table.

Initial velocity — Initial rotational velocity
0 rad/s (default) | scalar

Initial angular velocity, $\Omega(0)$, of the tire.

Dependencies

Selecting `Specify inertia and initial velocity` for the **Inertia** parameter, exposes this parameter. For more information, see Dynamics Parameter Dependencies Table.

Rolling Resistance

The table shows how the visibility of some **Rolling Resistance** parameters depends on the options that you choose for other parameters. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Rolling Resistance Parameter Dependencies Table

Rolling Resistance		
Rolling resistance — Choose `Off` or `On`		
Off	**On**	
	Resistance model — Choose `Constant coefficient` or `Pressure and velocity dependent`	
	Constant coefficient	**Pressure and velocity dependent**
	Constant coefficient	Tire pressure
		Alpha
		Beta
		Coefficient A
		Coefficient B
		Coefficient C
	Velocity threshold	

Rolling resistance — Rolling resistance option
`Off` (default) | `On`

Options for modeling rolling resistance are:

- `Off` — Neglect rolling resistance.
- `On` — Include rolling resistance.

Dependencies

Selecting `On` exposes rolling resistance parameters. For more information, see Rolling Resistance Parameter Dependencies Table.

Resistance model — Rolling resistance model
`Constant coefficient` (default) | `Pressure and velocity dependent`

Model for the rolling resistance of the tire.

- `Constant coefficient` — Neglect rolling resistance.
- `Pressure and velocity dependent` — Include rolling resistance.

Dependencies

Each **Resistance model** option exposes related parameters. For more information, see Rolling Resistance Parameter Dependencies Table.

Constant coefficient — Proportionality constant
`0.015` (default) | positive scalar

Coefficient that sets the proportionality between the normal force and the rolling resistance force. The parameter must be greater than zero.

Dependencies

Selecting `On` for the **Rolling resistance** parameter and `Constant coefficient` for the **Resistance model** parameter exposes this parameter. For more information, see Rolling Resistance Parameter Dependencies Table.

`Tire pressure` — Tire pressure
`250e3 Pa` (default) | positive scalar

Inflation pressure of the tire. The parameter must be greater than zero.

Dependencies

Selecting `On` for the **Rolling resistance** parameter and `Pressure and velocity dependent` for the **Resistance model** parameter exposes this parameter. For more information, see Rolling Resistance Parameter Dependencies Table.

`Alpha` — Tire pressure equation exponent
`-0.003` (default) | scalar

Exponent of the tire pressure in the model equation. See Rolling Resistance Parameter Dependencies Table.

Dependencies

Selecting `On` for the **Rolling resistance** parameter and `Pressure and velocity dependent` for the **Resistance model** parameter exposes this parameter. For more information, see Rolling Resistance Parameter Dependencies Table.

`Beta` — Normal force equation exponent
`0.97` (default) | scalar

Exponent of the normal force model equation. See Rolling Resistance Parameter Dependencies Table.

Dependencies

Selecting `On` for the **Rolling resistance** parameter and `Pressure and velocity dependent` for the **Resistance model** parameter exposes this parameter. For more information, see Rolling Resistance Parameter Dependencies Table.

`Coefficient A` — Velocity-independent force component *A*
`84e-4` (default)

Velocity-independent force component in the model equation. The parameter must be greater than zero.

Dependencies

Selecting `On` for the **Rolling resistance** parameter and `Pressure and velocity dependent` for the **Resistance model** parameter exposes this parameter. For more information, see Rolling Resistance Parameter Dependencies Table.

`Coefficient B` — Velocity-dependent force component *B*
`6.2e-4 s/m` (default) | positive scalar

Velocity-dependent force component in the model equation. The parameter must be greater than zero.

Dependencies

Selecting On for the **Rolling resistance** parameter and `Pressure and velocity dependent` for the **Resistance model** parameter exposes this parameter. For more information, see Rolling Resistance Parameter Dependencies Table.

Coefficient C — Velocity-dependent force component *C*
1.6e-4 s^2/m^2 (default) | positive scalar

Force component that depends on the square of the velocity term in the model equation. The parameter must be greater than zero.

Dependencies

Selecting On for the **Rolling resistance** parameter and `Pressure and velocity dependent` for the **Resistance model** parameter exposes this parameter. For more information, see Rolling Resistance Parameter Dependencies Table.

Velocity threshold — Wheel hub velocity threshold for mathematical slip model
0.001 m/s (default) | positive scalar

Velocity at which the full rolling resistance force is transmitted to the rolling hub. The parameter ensures that the force remains continuous during velocity direction changes, which increases the numerical stability of the simulation. The parameter must be greater than zero.

Dependencies

Selecting On for the **Rolling resistance** parameter exposes this parameter. For more information, see Rolling Resistance Parameter Dependencies.

Advanced

Use the **Advanced** tab to specify parameters that the state machine uses to determine the model of the tire.

Traction velocity tolerance — Traction velocity tolerance
0.01 m/s (default) | positive scalar

Magnitude of the relative velocity between the tire and ground at which the tire regains traction. Setting this value too low prevents the tire from entering a state where it has traction. Setting it too high can cause the tire velocity to change suddenly when the tire gains traction, and can result in an unstable simulation. The parameter must be greater than zero.

Engagement threshold force — Engagement threshold force
10 N (default) | positive scalar

Normal force values below the **Engagement threshold force** are not applied to the tire. Setting this value too low can cause the tire to gain and lose traction rapidly. Setting this value too high can give unrealistically low static and dynamic friction forces. The parameter must be greater than zero.

Initial traction state — Initial traction state
Tire is initially slipping (default) | Tire is initially in traction

Option to have the tire in traction or slipping at the start of simulation.

More About

Real-Time and Hardware-in-the-Loop Simulation

For optimal simulation performance, set the **Dynamics** > **Compliance** parameter to `No compliance - Suitable for HIL simulation`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Fundamental Friction Clutch | Tire (Simple) | Tire (Magic Formula) | Tire-Road Interaction (Magic Formula)

Topics
"Complete Vehicle Model"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"
"Clutch States" on page 1-235

Introduced in R2012a

Tire (Magic Formula)

Tire with longitudinal behavior given by Magic Formula coefficients
Library: Simscape / Driveline / Tires & Vehicles
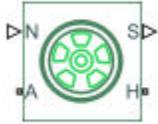
Description

The Tire (Magic Formula) block models a tire with longitudinal behavior given by the Magic Formula [1], an empirical equation based on four fitting coefficients. The block can model tire dynamics under constant or variable pavement conditions.

The longitudinal direction of the tire is the same as its direction of motion as it rolls on pavement. This block is a structural component based on the Tire-Road Interaction (Magic Formula) block.

To increase the fidelity of the tire model, you can specify properties such as tire compliance, inertia, and rolling resistance. However, these properties increase the complexity of the tire model and can slow down simulation. Consider ignoring tire compliance and inertia if simulating the model in real time or if preparing the model for hardware-in-the-loop (HIL) simulation.

Tire Model

The Tire (Magic Formula) block models the tire as a rigid wheel-tire combination in contact with the road and subject to slip. When torque is applied to the wheel axle, the tire pushes on the ground (while subject to contact friction) and transfers the resulting reaction as a force back on the wheel. This action pushes the wheel forward or backward. If you include the optional tire compliance, the tire also flexibly deforms under load.

The figure shows the forces acting on the tire. The table defines the tire model variables.

Tire Model Variables

Symbol	Description and Unit
r_w	Wheel radius
V_x	Wheel hub longitudinal velocity
u	Tire longitudinal deformation
Ω	Wheel angular velocity
Ω'	Contact point angular velocity. If there is no tire longitudinal deformation, that is, if $u = 0$, $\Omega' = \Omega$.
$r_w\Omega'$	Tire tread longitudinal velocity
$V_{sx} = r_w\Omega - V_x$	Wheel slip velocity
$V'_{sx} = r_w\Omega' - V_x$	Contact slip velocity. If there is no tire longitudinal deformation, that is, if $u = 0$, $V'_{sx} = V_{sx}$.
$k = \dfrac{V_{sx}}{|V_x|}$	Wheel slip
$k' = \dfrac{V'_{sx}}{|V_x|}$	Contact slip. If there is no tire longitudinal deformation, that is, if $u = 0$, $k' = k$.
V_{th}	Wheel hub threshold velocity
F_z	Vertical load on tire
F_x	Longitudinal force exerted on the tire at the contact point
$C_{Fx} = \left(\dfrac{\partial F_x}{\partial u}\right)_0$	Tire longitudinal stiffness under deformation
$b_{Fx} = \left(\dfrac{\partial F_x}{\partial \dot{u}}\right)_0$	Tire longitudinal damping under deformation
I_w	Wheel-tire inertia, such that the effective mass is equal to $\dfrac{I_w}{r_w^2}$
τ_{drive}	Torque applied by the axle to the wheel

Tire Kinematics and Response

Roll and Slip

A nonslipping tire would roll and translate as $V_x = r_w\Omega$. However, as tires do slip, they respond by developing a longitudinal force, F_x.

The wheel slip velocity is $V_{sx} = r_w\Omega - V_x$. The *wheel slip* is $k = \dfrac{V_{sx}}{|V_x|}$. For a locked, sliding wheel, $k = -1$. For perfect rolling, $k = 0$.

Slip at Low Speed

For low speeds, as defined by $|V_x| \le |V_{th}|$, the wheel slip becomes:

$$k = \frac{2V_{sx}}{\left(V_{th} + \dfrac{V_x^2}{V_{th}}\right)}$$

This modification allows for a nonsingular, nonzero slip at zero wheel velocity. For example, for perfect slipping, that is, in the case of a nontranslating spinning tire, $V_x = 0$ while $k = \dfrac{2r_w\Omega}{V_{th}}$ is finite.

Deformation

If the tire is modeled with compliance, it is also flexible. In this case, because the tire deforms, the tire-road contact point turns at a slightly different angular velocity, Ω', from the wheel, Ω, and requires, instead of the wheel slip, the *contact point* or *contact patch slip* κ'. The block models the deforming tire as a translational spring-damper of stiffness, C_{F_x}, and damping, b_{F_x}.

If you model a tire without compliance, that is, if $u = 0$, then there is no tire longitudinal deformation at any time in the simulation and:

* $k' = k$
* $V'_{sx} = V_{sx}$
* $\Omega' = \Omega$

Tire and Wheel Dynamics
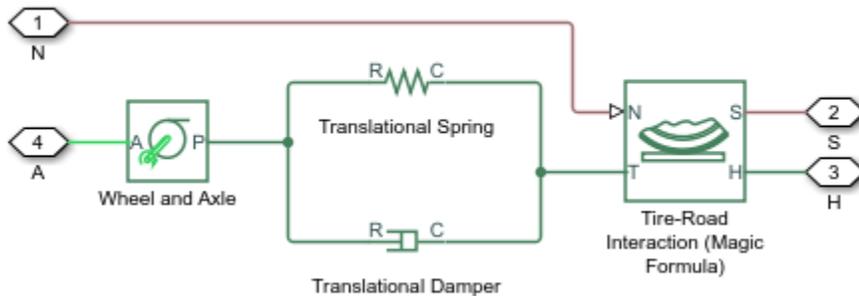
The full tire model is equivalent to this Simscape/Simscape Driveline component diagram. It simulates both transient and steady-state behavior and correctly represents starting from, and coming to, a stop. The Translational Spring and Translational Damper are equivalent to the tire stiffness C_{F_x} and damping b_{F_x}. The Tire-Road Interaction (Magic Formula) block models the longitudinal force F_x on the tire as a function of F_z and k' using the Magic Formula, with k' as the independent slip variable.

The Wheel and Axle radius is the wheel radius r_w. The Mass value is the effective mass, $\dfrac{I_w}{r_w^2}$. The tire characteristic function $f(k', F_z)$ determines the longitudinal force F_x. Together with the driveshaft torque applied to the wheel axis, F_x determines the wheel angular motion and longitudinal motion.

Without tire compliance, the Translational Spring and Translational Damper are omitted, and contact variables revert to wheel variables. In this case, the tire effectively has infinite stiffness, and port **P** of Wheel and Axle connects directly to port **T** of Tire-Road Interaction (Magic Formula).

Without tire inertia, the Mass is omitted.

Assumptions and Limitations

- The Tire (Magic Formula) block assumes longitudinal motion only and includes no camber, turning, or lateral motion.
- Tire compliance implies a time lag in the tire response to the forces on it. Time lag simulation increases model fidelity but reduces simulation performance. See "Adjust Model Fidelity".

Ports

Input

N — Normal force
physical signal

Physical signal input port associated with the normal force acting on the tire. The normal force is positive if it acts downward on the tire, pressing it against the pavement.

M — Magic formula coefficients
physical signal | vector | [*B*, *C*, *D*, *E*]

Physical signal input port associated with the Magic Formula coefficients. Provide the Magic Formula coefficients as a four-element vector, specified in the order [*B*, *C*, *D*, *E*].

Dependencies

Port **M** is exposed only if the **Main > Parameterize by** parameter is set to `Physical signal Magic Formula coefficients`. For more information, see Main Parameter Dependencies.

Output

S — Slip
physical signal

Physical signal output port associated with the relative slip between the tire and road.

Conserving

A — Axle
mechanical rotational

Mechanical rotational port associated with the axle that the tire sits on.

H — Hub
mechanical translational

Mechanical translational port associated with the wheel hub that transmits the thrust generated by the tire to the remainder of the vehicle.

Parameters

Main

The table shows how the visibility of some **Main** parameters depends on the options that you choose for other parameters. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Main Parameter Dependencies

Main			
Parameterize by — Choose `Peak longitudinal force and corresponding slip`, `Constant Magic Formula coefficients`, `Load-dependent Magic Formula coefficients`, or `Physical signal Magic Formula coefficients`			
Peak longitudinal force and corresponding slip	**Constant Magic Formula coefficients**	**Load-dependent Magic Formula coefficients**	**Physical signal Magic Formula coefficients** — Exposes physical signal input port **M** for providing the Magic Formula coefficients to the block as an array of elements in this order [B, C, D, E].
Rated vertical load	Magic Formula B coefficient	Magic Formula C-coefficient parameter, p_Cx1	
Peak longitudinal force at rated load	Magic Formula C coefficient	Magic Formula D-coefficient parameters, [p_Dx1 p_Dx2]	
Slip at peak force at rated load (percent)	Magic Formula D coefficient	Magic Formula E-coefficient parameters, [p_Ex1 p_Ex2 p_Ex3 p_Ex4]	
	Magic Formula E coefficient	Magic Formula BCD-coefficient parameters, [p_Kx1 p_Kx2 p_Kx3]	
		Magic Formula H-coefficient parameters, [p_Hx1 p_Hx2]	
		Magic Formula V-coefficient parameters, [p_Vx1 p_Vx2]	

Parameterize by — Parameterization method
`Peak longitudinal force and corresponding slip` (default) | `Constant Magic Formula coefficients` | `Load-dependent Magic Formula coefficients` | `Physical signal Magic Formula coefficients`

Magic Formula tire-road interaction model.

To model tire dynamics under constant pavement conditions, select one of these models:

- `Peak longitudinal force and corresponding slip` — Parameterize the Magic Formula with physical characteristics of the tire.
- `Constant Magic Formula coefficients` — Specify the parameters that define the constant B, C, D, and E coefficients as scalars, with these default values.

Coefficient	Default Value
B	10
C	1.9
D	1
E	0.97

- `Load-dependent Magic Formula coefficients` — Specify the parameters that define the load-dependent *C*, *D*, *E*, *K*, *H*, and *V* coefficients as vectors, one for each coefficient, with these default values.

Coefficient	Parameters	Default Values
C	**p_Cx1**	1.685
D	**[p_Dx1 p_Dx2]**	[1.21 –0.037]
E	**[p_Ex1 p_Ex2 p_Ex3 p_Ex4]**	[0.344 0.095 –0.02 0]
K	**[p_Kx1 p_Kx2 p_Kx3]**	[21.51 –0.163 0.245]
H	**[p_Hx1 p_Hx2]**	[–0.002 0.002]
V	**[p_Vx1 p_Vx2]**	[0 0]

To model tire dynamics under variable pavement conditions, select `Physical signal Magic Formula coefficients`. Selecting this model exposes a physical signal port **M**. Use the port to input the Magic Formula coefficients as a four-element vector, specified in the order [*B*, *C*, *D*, *E*].

Dependencies

Each parameterization method option exposes related parameters and hides unrelated parameters. Selecting `Physical signal Magic Formula coefficients` exposes a physical signal input port. For more information, see Main Parameter Dependencies.

Rated vertical load — Rated load force
3000 N (default) | positive scalar

Rated vertical load force F_{z0}.

Dependencies

This parameter is exposed only when you select the `Peak longitudinal force and corresponding slip` parameterization method. For more information, see Main Parameter Dependencies.

Peak longitudinal force at rated load — Maximum longitudinal force at rated load
3500 N (default) | positive scalar

Maximum longitudinal force F_{x0} that the tire exerts on the wheel when the vertical load equals its rated value F_{z0}.

Dependencies

This parameter is exposed only when you select the `Peak longitudinal force and corresponding slip` parameterization method. For more information, see Main Parameter Dependencies.

Slip at peak force at rated load (percent) — Percent slip at maximum longitudinal force and rated load
10 (default) | positive scalar

Contact slip κ'_0, expressed as a percentage (%), when the longitudinal force equals its maximum value F_{x0} and the vertical load equals its rated value F_{z0}.

Dependencies

This parameter is exposed only when you select the `Peak longitudinal force and corresponding slip` parameterization method. For more information, see Main Parameter Dependencies.

Magic Formula B coefficient — Constant Magic Formula *B* coefficient
10 (default) | positive scalar

Load-independent Magic Formula *B* coefficient.

Dependencies

This parameter is exposed only when you select the `Constant Magic Formula coefficients` parameterization method. For more information, see Main Parameter Dependencies.

Magic Formula C coefficient — Constant Magic Formula *C* coefficient
1.9 (default) | positive scalar

Load-independent Magic Formula *C* coefficient.

Dependencies

This parameter is exposed only when you select the `Constant Magic Formula coefficients` parameterization method. For more information, see Main Parameter Dependencies.

Magic Formula D coefficient — Constant Magic Formula *D* coefficient
1 (default) | positive scalar

Load-independent Magic Formula *D* coefficient.

Dependencies

This parameter is exposed only when you select the `Constant Magic Formula coefficients` parameterization method. For more information, see Main Parameter Dependencies.

Magic Formula E coefficient — Constant Magic Formula *E* coefficient
0.97 (default) | positive scalar

Load-independent Magic Formula *E* coefficient.

Dependencies

This parameter is exposed only when you select the `Constant Magic Formula coefficients` parameterization method. For more information, see Main Parameter Dependencies.

Tire nominal vertical load — Normal force
4000 N (default) | positive scalar

Nominal normal force F_{z0} on tire.

Dependencies

This parameter is exposed only when you select the `Load-dependent Magic Formula coefficients` parameterization method. For more information, see Main Parameter Dependencies.

Magic Formula C-coefficient parameter, p_Cx1 — Variable Magic Formula *C* coefficient
1.685 (default) | scalar

Load-dependent Magic Formula *C* coefficient.

Dependencies

This parameter is exposed only when you select the `Load-dependent Magic Formula coefficients` parameterization method. For more information, see Main Parameter Dependencies.

Magic Formula D-coefficient parameters, [p_Dx1 p_Dx2] — Variable Magic Formula *D* coefficient
[1.21, -.037] (default) | vector

Load-dependent Magic Formula *D* coefficient.

Dependencies

This parameter is exposed only when you select the `Load-dependent Magic Formula coefficients` parameterization method. For more information, see Main Parameter Dependencies.

Magic Formula E-coefficient parameters, [p_Ex1 p_Ex2 p_Ex3 p_Ex4] — Variable Magic Formula *E* coefficient
[.344, .095, -.02, 0] (default) | vector

Load-dependent Magic Formula *E* coefficient.

Dependencies

This parameter is exposed only when you select the `Load-dependent Magic Formula coefficients` parameterization method. For more information, see Main Parameter Dependencies.

Magic Formula BCD-coefficient parameters, [p_Kx1 p_Kx2 p_Kx3] — Variable Magic Formula *K* coefficient
[21.51, -.163, .245] (default) | vector

Load-dependent Magic Formula *K* coefficient.

Dependencies

This parameter is exposed only when you select the `Load-dependent Magic Formula coefficients` parameterization method. For more information, see Main Parameter Dependencies.

Magic Formula H-coefficient parameters, [p_Hx1 p_Hx2] — Variable Magic Formula *H* coefficient
[-.002, .002] (default) | vector

Load-dependent Magic Formula *H* coefficient.

Dependencies

This parameter is exposed only when you select the `Load-dependent Magic Formula coefficients` parameterization method. For more information, see Main Parameter Dependencies.

Magic Formula V-coefficient parameters, [p_Vx1 p_Vx2] — Variable Magic Formula *V* coefficient
[0, 0] (default) | vector

Load-dependent Magic Formula *V* coefficient.

Dependencies

This parameter is exposed only when you select the Load-dependent Magic Formula coefficients parameterization method. For more information, see Main Parameter Dependencies.

Geometry

Rolling radius — Unloaded tire-wheel radius
0.3 m (default) | positive scalar

Unloaded tire-wheel radius, r_w.

Dynamics

The table shows how the visibility of some **Dynamics** parameters depends on the options that you choose for other parameters. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Dynamics Parameter Dependencies

Dynamics	
Compliance — Choose No compliance - Suitable for HIL simulation or Specify stiffness and damping	
No compliance - Suitable for HIL simulation	**Specify stiffness and damping**
	Longitudinal stiffness
	Longitudinal damping
Inertia — Choose No Inertia or Specify inertia and initial velocity	
No Inertia	**Specify inertia and initial velocity**
	Tire inertia
	Initial velocity

Compliance — Dynamical compliance model
No compliance - Suitable for HIL simulation (default) | Specify stiffness and damping

Model for the dynamical compliance of the tire.

- No compliance - Suitable for HIL simulation — Tire is modeled with no dynamical compliance.
- Specify stiffness and damping — Tire is modeled as a stiff, dampened spring and deforms under load.

Dependencies

Selecting the Specify stiffness and damping parameterization method, exposes stiffness and damping parameters. For more information, see Dynamics Parameter Dependencies.

Longitudinal stiffness — Longitudinal stiffness
1e6 N/m (default) | positive scalar

Tire longitudinal stiffness C_{Fx}.

Dependencies

Selecting `Specify stiffness and damping` for the **Compliance** parameter, exposes this parameter. For more information, see Dynamics Parameter Dependencies.

Longitudinal damping — Longitudinal damping
1000 N/(m/s) (default) | positive scalar

Tire longitudinal damping b_{Fx}.

Dependencies

Selecting `Specify stiffness and damping` for the **Compliance** parameter, exposes this parameter. For more information, see Dynamics Parameter Dependencies.

Inertia — Inertia model
No inertia (default) | Specify inertia and initial velocity

Model for the rotational inertia of the tire.

- `No inertia` — Tire is modeled with no dynamical compliance.
- `Specify inertia and initial velocity` — Tire is modeled as a stiff, dampened spring and deforms under load.

Dependencies

Selecting the `Specify inertia and initial velocity` parameterization method, exposes inertia and velocity parameters. For more information, see Dynamics Parameter Dependencies.

Tire inertia — Rotational inertia
1 kg*m^2 (default) | positive scalar

Rotational inertia I_w of the wheel-tire assembly.

Dependencies

Selecting `Specify inertia and initial velocity` for the **Inertia** parameter, exposes this parameter. For more information, see Dynamics Parameter Dependencies.

Initial velocity — Initial rotational velocity
0 rad/s (default) | scalar

Initial angular velocity, $\Omega(0)$, of the tire.

Dependencies

Selecting `Specify inertia and initial velocity` for the **Inertia** parameter, exposes this parameter. For more information, see Dynamics Parameter Dependencies.

Rolling Resistance

The table shows how the visibility of some parameters depends on the options that you choose for other parameters. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Rolling Resistance Parameter Dependencies

Rolling Resistance			
Rolling resistance — Choose `Off` or `On`			
Off	**On**		
	Resistance model — Choose `Constant coefficient` or `Pressure and velocity dependent`		
	Constant coefficient	**Pressure and velocity dependent**	
	Constant coefficient	Tire pressure	
		Alpha	
		Beta	
		Coefficient A	
		Coefficient B	
		Coefficient C	
	Velocity threshold		

Rolling resistance — Rolling resistance option
Off (default) | On

Options for modeling rolling resistance are:

- Off — Neglect rolling resistance.
- On — Include rolling resistance.

Dependencies

Selecting On exposes rolling resistance parameters. For more information, see Rolling Resistance Parameter Dependencies.

Resistance model — Rolling resistance model
Constant coefficient (default) | Pressure and velocity dependent

Model for the rolling resistance of the tire.

- Constant coefficient — Neglect rolling resistance.
- Pressure and velocity dependent — Include rolling resistance.

Dependencies

Each **Resistance model** option exposes related parameters. For more information, see Rolling Resistance Parameter Dependencies.

Constant coefficient — Proportionality constant
0.015 (default) | positive scalar

Coefficient that sets the proportionality between the normal force and the rolling resistance force. The parameter must be greater than zero.

Dependencies

Selecting On for the **Rolling resistance** parameter and `Constant coefficient` for the **Resistance model** parameter exposes this parameter. For more information, see Rolling Resistance Parameter Dependencies.

Tire pressure — Tire pressure
250e3 Pa (default) | positive scalar

Inflation pressure of the tire. The parameter must be greater than zero.

Dependencies

Selecting On for the **Rolling resistance** parameter and `Pressure and velocity dependent` for the **Resistance model** parameter exposes this parameter. For more information, see Rolling Resistance Parameter Dependencies.

Alpha — Tire pressure equation exponent
-0.003 (default) | scalar

Exponent of the tire pressure in the model equation.

Dependencies

Selecting On for the **Rolling resistance** parameter and `Pressure and velocity dependent` for the **Resistance model** parameter exposes this parameter. For more information, see Rolling Resistance Parameter Dependencies.

Beta — Normal force equation exponent
0.97 (default) | scalar

Exponent of the normal force model equation.

Dependencies

Selecting On for the **Rolling resistance** parameter and `Pressure and velocity dependent` for the **Resistance model** parameter exposes this parameter. For more information, see Rolling Resistance Parameter Dependencies.

Coefficient A — Velocity-independent force component *A*
84e-4 (default)

Velocity-independent force component in the model equation. The parameter must be greater than zero.

Dependencies

Selecting On for the **Rolling resistance** parameter and `Pressure and velocity dependent` for the **Resistance model** parameter exposes this parameter. For more information, see Rolling Resistance Parameter Dependencies.

Coefficient B — Velocity-dependent force component *B*
6.2e-4 s/m (default) | positive scalar

Velocity-dependent force component in the model equation. The parameter must be greater than zero.

Dependencies

Selecting On for the **Rolling resistance** parameter and Pressure and velocity dependent for the **Resistance model** parameter exposes this parameter. For more information, see Rolling Resistance Parameter Dependencies.

Coefficient C — Velocity-dependent force component *C*
1.6e-4 s^2/m^2 (default) | positive scalar

Force component that depends on the square of the velocity term in the model equation. The parameter must be greater than zero.

Dependencies

Selecting On for the **Rolling resistance** parameter and Pressure and velocity dependent for the **Resistance model** parameter exposes this parameter. For more information, see Rolling Resistance Parameter Dependencies.

Velocity threshold — Wheel hub velocity threshold for mathematical slip model
0.001 m/s (default) | positive scalar

Velocity at which the full rolling resistance force is transmitted to the rolling hub. The parameter ensures that the force remains continuous during velocity direction changes, which increases the numerical stability of the simulation. The parameter must be greater than zero.

Dependencies

Selecting On for the **Rolling resistance** parameter exposes this parameter. For more information, see Rolling Resistance Parameter Dependencies.

Advanced

Velocity threshold — Wheel hub velocity threshold for mathematical slip model
0.1 m/s (default) | positive scalar

Wheel hub velocity V_{th} below which the slip calculation is modified to avoid singular evolution at zero velocity. Must be positive.

More About

Real-Time and Hardware-in-the-Loop Simulation

For optimal simulation performance, set the **Dynamics > Compliance** parameter to No compliance - Suitable for HIL simulation.

References

[1] Pacejka, H. B. *Tire and Vehicle Dynamics.* Elsevier Science, 2005.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Tire (Friction Parameterized) | Tire (Simple) | Tire-Road Interaction (Magic Formula)

Topics
"Complete Vehicle Model"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

Tire-Road Interaction (Magic Formula)

Tire-road dynamics given by magic formula coefficients
Library: Simscape / Driveline / Tires & Vehicles / Tire Subcomponents
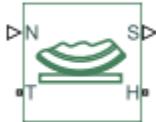
Description

The Tire-Road Interaction (Magic Formula) block models the interaction between the tire tread and road pavement. The longitudinal force arising from this interaction is given by the magic formula, an empirical equation based on four fitting coefficients. Tire properties such as compliance and inertia are ignored.

Tire-Road Interaction Model

The Tire-Road Interaction (Magic Formula) block models the longitudinal forces at the tire-road contact patch using the Magic Formula of Pacejka [1].

The figure displays the forces on the tire. The table defines the tire model variables.

Tire-Road Contact Variables

Symbol	Description and Unit
Ω	Wheel angular velocity
r_w	Wheel radius
V_x	Wheel hub longitudinal velocity
$r_w\Omega$	Tire tread longitudinal velocity
$V_{sx} = r_w\Omega - V_x$	Wheel slip velocity, defined as the difference between the longitudinal velocities of the wheel hub and the tire tread
$k = \dfrac{V_{sx}}{\lvert V_x \rvert}$	Wheel slip
F_z	Vertical load on tire
F_{z0}	Nominal vertical load on tire
$F_x = f(\kappa, F_z)$	Longitudinal force exerted on the tire at the contact point. Also a characteristic function f of the tire.

Tire Response

Forces and Characteristic Function

A tire model provides a steady-state tire characteristic function, $F_x = f(\kappa, F_z)$, the longitudinal force F_x on the tire, based on:

- Vertical load F_z
- Wheel slip κ

Magic Formula with Constant Coefficients

The Magic Formula is a specific form for the tire characteristic function, characterized by four dimensionless coefficients, B, C, D, and E, or stiffness, shape, peak, and curvature:

$$F_x = f(\kappa, F_z) = F_z \cdot D \cdot \sin(C \cdot \arctan\{B\kappa - E[B\kappa - \arctan(B\kappa)]\})$$

The slope of f at $k = 0$ is $BCD \cdot F_z$.

Magic Formula with Load-Dependent Coefficients

A more general Magic Formula uses dimensionless coefficients that are functions of the tire load. A more complex set of parameters `p_i`, entered in the dialog box, specifies these functions:

$$F_{x0} = D_x\sin(C_x\arctan\{B_x\kappa_x - E_x[B_x\kappa_x - \arctan(B_x\kappa_x)]\}) + S_{Vx}$$

Where:

$$df_z = \frac{(F_z - F_{z0})}{F_{z0}}$$

$$\kappa_x = \kappa + S_{Hx}$$

$$C_x = \text{p\_Cx1}$$

$$D_x = \mu_x \cdot F_z$$

$$\mu_x = \text{p\_Dx1} + \text{p\_Dx2} \cdot df_z$$

$$E_x = (\text{p\_Ex1} + \text{p\_Ex2} \cdot df_z + \text{p\_Ex3} \cdot df_z^2)[1 - \text{p\_Ex4} \cdot \text{sgn}(\kappa_x)]$$

$$K_{x\kappa} = F_z \cdot (\text{p\_Kx1} + \text{p\_Kx2} \cdot df_z) \cdot e^{(\text{p\_Kx3} \cdot df_z)}$$

$$B_x = \frac{K_{x\kappa}}{C_x D_x + \varepsilon_x}$$

$$S_{Hx} = \text{p\_Hx1} + \text{p\_Hx2} \cdot df_z$$

$$S_{Vx} = F_z \cdot (\text{p\_Vx1} + \text{p\_Vx2} \cdot df_z)$$

S_{Hx} and S_{Vx} represent offsets to the slip and longitudinal force in the force-slip function, or horizontal and vertical offsets if the function is plotted as a curve. μ_x is the longitudinal load-dependent friction coefficient. ε_x is a small number inserted to prevent division by zero as F_z approaches zero.

Peak Longitudinal Force and Corresponding Slip

The block uses a representative set of Magic Formula coefficients. The block scales the coefficients to yield the peak longitudinal force F_{x0} at the corresponding slip κ_0 that you specify, for rated vertical load F_{z0}.

Magic Formula Coefficients for Typical Road Conditions

Numerical values are based on empirical tire data. These values are typical sets of constant Magic Formula coefficients for common road conditions.

Surface	B	C	D	E
Dry tarmac	10	1.9	1	0.97
Wet tarmac	12	2.3	0.82	1
Snow	5	2	0.3	1
Ice	4	2	0.1	1

Assumptions and Limitations

- The Tire-Road Interaction (Magic Formula) block assumes longitudinal motion only and includes no camber, turning, or lateral motion.

Ports

Input

N — Normal force
physical signal

Physical signal input port associated with the normal force acting on the tire. The normal force is positive if it acts downward on the tire, pressing it against the pavement.

M — Magic formula coefficients
physical signal | vector | [*B*, *C*, *D*, *E*]

Physical signal input port associated with the Magic Formula coefficients. Provide the Magic Formula coefficients as a four-element vector, specified in the order [*B*, *C*, *D*, *E*].

Dependencies

Port **M** is exposed only if the **Parameterize by** parameter is set to `Physical signal Magic Formula coefficients`. For more information, see Parameter Dependencies.

Output

S — Slip
physical signal

Physical signal output port associated with the relative slip between the tire and road.

Conserving

A — Axle
mechanical rotational

Mechanical rotational port associated with the axle that the tire sits on.

H — Hub
mechanical translational

Mechanical translational port associated with the wheel hub that transmits the thrust generated by the tire to the remainder of the vehicle.

Parameters

The table shows how the visibility of some **Main** parameters depends on the options that you choose for other parameters. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Parameter Dependencies

Parameters			
Parameterize by — Choose `Peak longitudinal force and corresponding slip`, `Constant Magic Formula coefficients`, `Load-dependent Magic Formula coefficients`, or `Physical signal Magic Formula coefficients`			
Peak longitudinal force and corresponding slip	**Constant Magic Formula coefficients**	**Load-dependent Magic Formula coefficients**	**Physical signal Magic Formula coefficients** — Exposes physical signal input port **M** for providing the Magic Formula coefficients to the block as an array of elements in this order [*B*, *C*, *D*, *E*].
Rated vertical load	Magic Formula B coefficient	Magic Formula C-coefficient parameter, p_Cx1	
Peak longitudinal force at rated load	Magic Formula C coefficient	Magic Formula D-coefficient parameters, [p_Dx1 p_Dx2]	
Slip at peak force at rated load (percent)	Magic Formula D coefficient	Magic Formula E-coefficient parameters, [p_Ex1 p_Ex2 p_Ex3 p_Ex4]	
	Magic Formula E coefficient	Magic Formula BCD-coefficient parameters, [p_Kx1 p_Kx2 p_Kx3]	
		Magic Formula H-coefficient parameters, [p_Hx1 p_Hx2]	
		Magic Formula V-coefficient parameters, [p_Vx1 p_Vx2]	
Velocity threshold			

`Parameterize by` — Parameterization method
`Peak longitudinal force and corresponding slip` (default) | `Constant Magic Formula coefficients` | `Load-dependent Magic Formula coefficients` | `Physical signal Magic Formula coefficients`

Magic Formula tire-road interaction model.

To model tire dynamics under constant pavement conditions, select one of these models:

*   `Peak longitudinal force and corresponding slip` — Parametrize the Magic Formula with physical characteristics of the tire.
*   `Constant Magic Formula coefficients` — Specify the parameters that define the constant *B*, *C*, *D*, and *E* coefficients as scalars, with these default values.

Coefficient	Default Value
B	10
C	1.9
D	1

Coefficient	Default Value
E	0.97

- `Load-dependent Magic Formula coefficients` — Specify the parameters that define the load-dependent *C*, *D*, *E*, *K*, *H*, and *V* coefficients as vectors, one for each coefficient, with these default values.

Coefficient	Parameters	Default Values
C	**p_Cx1**	1.685
D	**[p_Dx1 p_Dx2]**	[1.21 –0.037]
E	**[p_Ex1 p_Ex2 p_Ex3 p_Ex4]**	[0.344 0.095 –0.02 0]
K	**[p_Kx1 p_Kx2 p_Kx3]**	[21.51 –0.163 0.245]
H	**[p_Hx1 p_Hx2]**	[–0.002 0.002]
V	**[p_Vx1 p_Vx2]**	[0 0]

To model tire dynamics under variable pavement conditions, select `Physical signal Magic Formula coefficients`. Selecting this model exposes a physical signal port **M**. Use the port to input the Magic Formula coefficients as a four-element vector, specified in the order [*B*, *C*, *D*,*E*].

Dependencies

Each parameterization method option exposes related parameters and hides unrelated parameters. Selecting `Physical signal Magic Formula coefficients` exposes a physical signal input port. For more information, see Parameter Dependencies.

Rated vertical load — Rated load force
3000 N (default) | positive scalar

Rated vertical load force F_{z0}.

Dependencies

This parameter is exposed only when you select the `Peak longitudinal force and corresponding slip` parameterization method. For more information, see Parameter Dependencies.

Peak longitudinal force at rated load — Maximum longitudinal force at rated load
3500 N (default) | positive scalar

Maximum longitudinal force F_{x0} that the tire exerts on the wheel when the vertical load equals its rated value F_{z0}.

Dependencies

This parameter is exposed only when you select the `Peak longitudinal force and corresponding slip` parameterization method. For more information, see Parameter Dependencies.

Slip at peak force at rated load (percent) — Percent slip at maximum longitudinal force and rated load
10 (default) | positive scalar

Contact slip κ'_0, expressed as a percentage (%), when the longitudinal force equals its maximum value F_{x0} and the vertical load equals its rated value F_{z0}.

Dependencies

This parameter is exposed only when you select the `Peak longitudinal force and corresponding slip` parameterization method. For more information, see Parameter Dependencies.

Magic Formula B coefficient — Constant Magic Formula *B* coefficient
10 (default) | positive scalar

Load-independent Magic Formula *B* coefficient.

Dependencies

This parameter is exposed only when you select the `Constant Magic Formula coefficients` parameterization method. For more information, see Parameter Dependencies.

Magic Formula C coefficient — Constant Magic Formula *C* coefficient
1.9 (default) | positive scalar

Load-independent Magic Formula *C* coefficient.

Dependencies

This parameter is exposed only when you select the `Constant Magic Formula coefficients` parameterization method. For more information, see Parameter Dependencies.

Magic Formula D coefficient — Constant Magic Formula *D* coefficient
1 (default) | positive scalar

Load-independent Magic Formula *D* coefficient.

Dependencies

This parameter is exposed only when you select the `Constant Magic Formula coefficients` parameterization method. For more information, see Parameter Dependencies.

Magic Formula E coefficient — Constant Magic Formula *E* coefficient
0.97 (default) | positive scalar

Load-independent Magic Formula *E* coefficient.

Dependencies

This parameter is exposed only when you select the `Constant Magic Formula coefficients` parameterization method. For more information, see Parameter Dependencies.

Tire nominal vertical load — Normal force
4000 N (default) | positive scalar

Nominal normal force F_{z0} on tire.

Dependencies

This parameter is exposed only when you select the `Load-dependent Magic Formula coefficients` parameterization method. For more information, see Parameter Dependencies.

Magic Formula C-coefficient parameter, p_Cx1 — Variable Magic Formula *C* coefficient
1.685 (default) | scalar

Load-dependent Magic Formula *C* coefficient.

Dependencies

This parameter is exposed only when you select the `Load-dependent Magic Formula coefficients` parameterization method. For more information, see Parameter Dependencies.

Magic Formula D-coefficient parameters, [p_Dx1 p_Dx2] — Variable Magic Formula *D* coefficient
[1.21, -.037] (default) | vector

Load-dependent Magic Formula *D* coefficient.

Dependencies

This parameter is exposed only when you select the `Load-dependent Magic Formula coefficients` parameterization method. For more information, see Parameter Dependencies.

Magic Formula E-coefficient parameters, [p_Ex1 p_Ex2 p_Ex3 p_Ex4] — Variable Magic Formula *E* coefficient
[.344, .095, -.02, 0] (default) | vector

Load-dependent Magic Formula *E* coefficient.

Dependencies

This parameter is exposed only when you select the `Load-dependent Magic Formula coefficients` parameterization method. For more information, see Parameter Dependencies.

Magic Formula BCD-coefficient parameters, [p_Kx1 p_Kx2 p_Kx3] — Variable Magic Formula *K* coefficient
[21.51, -.163, .245] (default) | vector

Load-dependent Magic Formula *K* coefficient.

Dependencies

This parameter is exposed only when you select the `Load-dependent Magic Formula coefficients` parameterization method. For more information, see Parameter Dependencies.

Magic Formula H-coefficient parameters, [p_Hx1 p_Hx2] — Variable Magic Formula *H* coefficient
[-.002, .002] (default) | vector

Load-dependent Magic Formula *H* coefficient.

Dependencies

This parameter is exposed only when you select the `Load-dependent Magic Formula coefficients` parameterization method. For more information, see Parameter Dependencies.

Magic Formula V-coefficient parameters, [p_Vx1 p_Vx2] — Variable Magic Formula *V* coefficient
[0, 0] (default) | vector

Load-dependent Magic Formula *V* coefficient.

Dependencies

This parameter is exposed only when you select the `Load-dependent Magic Formula coefficients` parameterization method. For more information, see Parameter Dependencies.

Velocity threshold — Wheel hub velocity threshold for mathematical slip model
`0.1 m/s` (default) | positive scalar

Wheel hub velocity V_{th} below which the slip calculation is modified to avoid singular evolution at zero velocity. Must be positive.

References

[1] Pacejka, H. B. *Tire and Vehicle Dynamics.* Elsevier Science, 2005.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Tire (Magic Formula) | Tire (Simple) | Tire (Friction Parameterized)

Topics
"Complete Vehicle Model"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

Tire (Simple)

No-slip tire model with minimal parameters
Library: Simscape / Driveline / Tires & Vehicles

Description

The Tire (Simple) block represents a simple, no-slip model of a tire that you parameterize by tire radius.

You can optionally add inertia, stiffness, or damping. These dynamics require additional computation that tends to make models less suitable for real-time simulation.

Assumptions and Limitations

- The tire does not slip.
- Torque-transfer capability is infinite.

Ports

Conserving

A — Axle
mechanical rotational

Mechanical rotational port associated with the axle that the tire sits on.

H — Hub
mechanical translational

Mechanical translational port associated with the wheel hub that transmits the thrust generated by the tire to the remainder of the vehicle.

Parameters

The table shows how the visibility of some Tire (Simple) parameters depends on the option that you choose for other parameters. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Parameter Dependencies

Parameter	
Rolling Radius	
Compliance — Choose `No compliance - Suitable for HIL simulation` or `Specify stiffness and damping`	
No compliance - Suitable for HIL simulation	**Specify stiffness and damping**
	Longitudinal stiffness
	Longitudinal damping
Inertia — Choose `No Inertia` or `Specify inertia and initial velocity`	
No Inertia	**Specify inertia and initial velocity**
	Tire inertia
	Initial velocity

Rolling radius — Radius
`0.3 m` (default) | positive scalar

Rolling radius of the tire. The rolling radius must be greater than zero.

Compliance — Compliance model
`No compliance - Suitable for HIL simulation` (default) | `Specify stiffness and damping`

Include or neglect longitudinal stiffness and damping.

Dependencies

Selecting `Specify stiffness and damping` exposes other parameters. For more information, see Parameter Dependencies.

Longitudinal stiffness — Longitudinal stiffness
`1e6 N/m` (default) | positive scalar

Longitudinal stiffness of the tire. The parameter must be greater than zero.

Dependencies

This parameter appears only if the **Compliance** parameter is set to `Specify stiffness and damping`. For more information, see Parameter Dependencies.

Longitudinal damping — Longitudinal damping
`1000 N/(m/s)` (default) | positive scalar

Longitudinal damping of the tire. The parameter must be greater than zero.

Dependencies

This parameter appears only if the **Compliance** parameter is set to `Specify stiffness and damping`. For more information, see Parameter Dependencies.

Inertia — Inertia model
`No inertia` (default) | `Specify inertia and initial velocity`

Include or neglect tire inertia.

Dependencies

Selecting `Specify inertia and initial velocity` exposes other parameters. For more information, see Parameter Dependencies.

`Tire inertia` — **Tire inertia**
`1 kg*/m^2` (default) | positive scalar

Rotational inertia of the tire.

Dependencies

This parameter appears only if the **Inertia** parameter is set to `Specify inertia and initial velocity`. For more information, see Parameter Dependencies.

`Initial velocity` — **Initial velocity**
`0 rad/s` (default)

Initial rotational velocity of the tire.

Dependencies

This parameter appears only if the **Inertia** parameter is set to `Specify inertia and initial velocity`. For more information, see Parameter Dependencies.

More About

Real-Time and Hardware-in-the-Loop Simulation

For optimal simulation performance, set the **Parameters > Compliance** parameter to `No compliance - Suitable for HIL simulation`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Tire (Friction Parameterized) | Tire (Magic Formula) | Tire-Road Interaction (Magic Formula)

Topics
"Complete Vehicle Model"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2012a

Torque Converter

Viscous fluid coupling between rotating driveline shafts
Library: Simscape / Driveline / Couplings & Drives
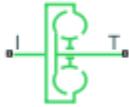
Description

The Torque Converter block models a torque converter. The Torque Converter block has two mechanical rotational conserving ports that are associated with the impeller and turbine, respectively. The block transfers torque and angular velocity between the impeller port **I** and turbine port **T** by acting as a lookup table. The block can simulate drive (power flows from **I** port to **T** port) and coast (power flows from **T** port to **I** port) modes.
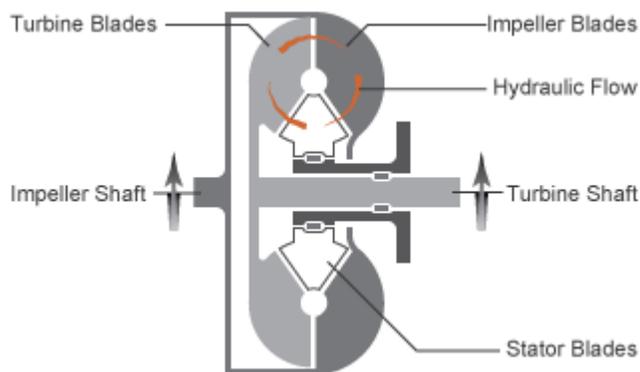
Limitations

When **Coast mode modeling** is set to `Continuous`:

*   The impeller shaft must always rotate in a positive direction. Simulation is not valid for $\omega_I < 0$.
*   If you drive the Torque Converter block by using a torque source, such as the Generic Engine block, you must include an inertia in the source to represent the engine, shaft inertia, or other source components. To ensure that the impeller starts by rotating in a positive direction, set the initial speed for this inertia to a positive value.

Ports

Conserving

I — Impeller
mechanical rotational

Mechanical rotational conserving port associated with the impeller.

T — Turbine
mechanical rotational

Mechanical rotational conserving port associated with turbine.

Parameters

Torque Characteristics

Coast mode modeling — Modeling type of torque converter
Two-mode (default) | Continuous

Modeling type of the torque converter specified as either Two-mode or Continuous. The Continuous modeling type supports both drive and coast modes but has reduced accuracy and robustness when modeling near the transition between coasting and driving modes. Therefore, if the simulation involves a coast mode, use the Two-mode modeling type due to its better robustness and accuracy when modeling the coast mode.

Drive mode speed ratio vector — Turbine speed to impeller speed ratios
[0, .5, .6, .7, .8, .87, .92, .94, .96, .97, 1] (default) | vector

Speed ratios, R_ω, of the drive mode. The vector elements must be in ascending order starting at 0 and ending at 1.

$$R_\omega = \omega_T/\omega_I$$

Dependencies

To enable this parameter, set **Coast mode modeling** to Two-mode.

Drive mode torque ratio vector — Turbine torque to impeller torque ratios
[2.232, 1.5462, 1.4058, 1.2746, 1.1528, 1.0732, 1.0192, 1, 1, 1, 1] (default) | vector

Torque ratios, R_τ, of the drive mode. Each element of the vector must be greater than or equal to 1, and the last element must be 1.

$$R_\tau = \tau_T/\tau_I$$

Dependencies

To enable this parameter, set **Coast mode modeling** to Two-mode.

Drive mode capacity factor vector — Capacity factors of drive mode
1e-4 * [.72558, .66322, .63463, .59042, .51331, .4144, .29287, .22444, .12186, .04386, 0] m*N/rpm^2 (default) | vector

Capacity factors,$K*$, of the drive mode. Each element of the vector must be nonnegative, and the last element must be 0.

$$K* = \tau_I/\omega_I^2$$

Dependencies

To enable this parameter, set **Coast mode modeling** to Two-mode.

Coast mode speed ratio vector — Impeller speed to turbine speed ratios
[0, .5, .6, .7, .8, .87, .92, .94, .96, .97, 1] (default) | vector

Speed ratios, $\widehat{R}_\omega$, of the coast mode. The vector elements must be in ascending order starting at 0 and ending at 1.

$$\widehat{R}_\omega = \omega_I/\omega_T$$

Dependencies

To enable this parameter, set **Coast mode modeling** to Two-mode.

Coast mode capacity factor vector — Capacity factors of coast mode
1e-4 * [.5079, .4643, .4442, .4133, .3593, .2901, .205, .1571, .0853, .0307, 0] m*N/rpm^2 (default) | vector

Capacity factors, $\widehat{K}*$, of the coast mode. Each element of the vector must be nonnegative, and the last element must be 0.

$$\widehat{K}* = \tau_T/\omega_T^2$$

Dependencies

To enable this parameter, set **Coast mode modeling** to Two-mode.

Interpolation method — Interpolation method of lookup function
Linear (default) | Smooth

Interpolation method of the lookup function, specified as either Linear or Smooth. The method interpolates torque ratio and capacity factor functions between the discrete relative velocity values within the definition range. For more information about Linear and Smooth, see tablelookup.

Extrapolation method — Extrapolation method of lookup function
Linear (default) | Nearest | Error

Extrapolation method of the lookup function, specified as Linear, Smooth, or Error. The method extrapolates torque ratio and capacity factor functions. For more information about Linear, Smooth, and Error, see tablelookup.

Initial mode — Initial mode of simulation
Drive mode (default) | Coast mode

Initial mode of the simulation, specified as either Drive mode or Coast mode.

Mode transition threshold — Threshold of mode transition
1 rpm (default) | positive scalar

Mode transition threshold of the simulation. Setting a threshold for the mode transition can increase the simulation robustness by avoiding the high frequency mode switching.

Dependencies

To enable this parameter, set **Coast mode modeling** to Two-mode.

Speed ratio vector — Turbine speed to impeller speed ratios
[0, .5, .6, .7, .8, .87, .92, .94, .96, .97] (default) | vector

Speed ratios, R_ω, of the torque converter. Each element of the vector must be in ascending order and in the range [0,1].

$$R_\omega = \omega_T/\omega_I$$

Dependencies

To enable this parameter, set **Coast mode modeling** to Continuous.

Torque ratio vector — Turbine torque to impeller torque ratios
[2.232, 1.5462, 1.4058, 1.2746, 1.1528, 1.0732, 1.0192, .9983, .9983, .9983, .9983] (default) | vector

Torque ratios, R_τ, of the torque converter. Each element of the vector must be positive.

$$R_\tau = \tau_T/\tau_I$$

Dependencies

To enable this parameter, set **Coast mode modeling** to Continuous.

Capacity factor parameterization — Definition of capacity factor
Ratio of speed to square root of impeller torque (default) | Ratio of impeller torque to square of speed

Definition of the capacity factor of the converter, defined as either `Ratio of speed to square root of impeller torque` or `Ratio of impeller torque to square of speed`. The setting of this parameter affects the **Capacity factor vector**.

- For `Ratio of speed to square root of impeller torque` parameter:

  $$K = \omega/\sqrt{\tau_I}$$

- For `Ratio of impeller torque to square of speed` parameter:

  $$K^* = \tau_I/\omega^2$$

Dependencies

To enable this parameter, set **Coast mode modeling** to Continuous.

Capacity factor reference speed — Choice of speed for capacity factor definition
Always impeller speed (default) | Turbine speed for speed ratios greater than one

Choice of speed for the capacity factor definition, specified as either `Always impeller speed` or `Turbine speed for speed ratios greater than one`.

- `Always impeller speed`: Use impeller speed ω_I for all values of R_ω.

- `Turbine speed for speed ratios greater than one`: Use impeller speed ω_I for all values of $R_\omega < 1$, and use turbine speed ω_T when $R_\omega > 1$.

Dependencies

To enable this parameter, set **Coast mode modeling** to Continuous.

Capacity factor vector — Capacity factors
[12.2938, 12.8588, 13.1452, 13.6285, 14.6163, 16.2675, 19.3503, 22.1046, 29.9986, 50] `(rad/s)/` `(N*m)^0.5` (default) | vector

Capacity factors of the converter. You can define the capacity factor as:

Capacity factor

$K = \omega/\sqrt{\tau_I}$	Set **Capacity factor parameterization** parameter to `Ratio of speed to square root of impeller torque.`
$K^* = \tau_I/\omega^2$	Set **Capacity factor parameterization** parameter to `Ratio of impeller torque to the square of the speed`. The default value is 1e-3 * [6.616, 6.048, 5.787, 5.384, 4.681, 3.779, 2.671, 2.047, 1.111, .4] `N*m/(rad/s)^2`.

Note If you do not specify capacity factor data for a speed ratio of 1, the block uses a capacity factor value of 10*K_{Max}, where K_{Max} is the maximum value in the specified capacity factor vector. The corresponding torque ratio is assumed to be 0. For all other speed ratio values not explicitly specified in the lookup table data, the block uses the interpolation or extrapolation method selected in the block dialog box.

Dependencies

To enable this parameter, set **Coast mode modeling** to `Continuous`.

Dynamics

To enable the **Dynamics**, set the **Coast mode modeling** parameter to `Continuous`.

Model transmission lag — Transmission lag setting
`No lag — Suitable for HIL simulation` (default) | `Specify time constant and initial value`

Transmission lag setting, specified as either `No lag — Suitable for HIL simulation` or `Specify time constant and initial value`.

- `No lag — Suitable for HIL simulation`: Torque transfer is instantaneous.

  When there is no time lag, the input impeller torque, τ_I, and output turbine torque, τ_T, are:

  $$\tau_I = \text{sgn}(1 - \omega_T/\omega_I)(\omega_I/K)^2$$

  $$\tau_T = \tau_T R_\tau$$

- `Specify time constant and initial value`: Torque is transferred with a time lag. If you select this option, you can specify the **Torque transmission time constant** and **Initial turbine-to-impeller torque ratio** parameters.

Note For optimal simulation performance, select `No lag - Suitable for HIL simulation`.

Torque transmission time constant — Time constant of torque transmission
.02 s (default) | positive scalar

Torque transmission time. The time lag increases model fidelity but reduces simulation performance. See "Adjust Model Fidelity" for more information.

Dependencies

To enable this parameter, set **Model transmission lag** to Specify time constant and initial value.

Initial turbine-to-impeller torque ratio — Initial torque ratio of turbine to impeller
0 (default) | positive scalar

Initial torque ratio of the turbine to the impeller.

You can optionally include the effect of torque transmission time lag that is caused by internal fluid flow and compressibility. Instead of τ_T and τ_I being instantaneously constrained to one another, a first-order time lag introduces a delayed response in the impeller torque:

$$t_c(d\tau_I/dt) + \tau_I = \tau_I(steadystate)$$

The preceding instantaneous function of the capacity factor K determines the steady-state value of τ_I.

Dependencies

To enable this parameter, set **Model transmission lag** to Specify time constant and initial value.

References

[1] Society of Automotive Engineers, *Hydrodynamic Drive Test Code (Surface Vehicle Recommended Practice),* SAE J643, Dec 2018.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Disk Friction Clutch | Fundamental Friction Clutch | Generic Engine

Topics
"Complete Vehicle Model"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

Torque Noise Source

Model zero-mean normally (Gaussian) distributed torque
Library: Simscape / Driveline / Sources

Description

The Torque Noise Source block generates zero-mean normally (Gaussian) distributed torque using the Random Number source in the Simscape Foundation library. The torque produced by the block depends on two factors:

- The noise power spectral density input to the block
- A random number seed that you control using the **Repeatability** and, as applicable, **Seed** parameters

The defining equation for the torque that the block generates is

$$T = \sqrt{PSD/2}\frac{N(0, 1)}{\sqrt{h}},$$

where:

- T is the output torque.
- PSD is the single-sided noise power spectral density.
- N is a Gaussian random number with zero mean and standard deviation of one.
- h is the sampling interval.

Ports

Input

P — Single-sided noise power spectral density
physical signal

The power spectral density (PSD) of noise is the average noise power per unit of bandwidth, in $(N*m)^2/Hz$.

Conserving

B — Base interface
mechanical rotational

Mechanical rotational conserving port associated with the base interface.

F — Follower interface
mechanical rotational

Mechanical rotational conserving port associated with the follower interface.

Parameters

Sample time — Time step period and offset
1e-3 s (default) | [*step*, *offset*] s

The values of the time *step* period and the initial time *offset*. If you specify a scalar value for *step*, the block assumes an *offset* value of 0.To specify a nonzero value for the initial time offset, specify the parameter values using the vector [*step*, *offset*]. The *offset* value must be less than the *step* value and greater than or equal to zero.

Repeatability — Random number seed control
Not repeatable (default) | Repeatable | Specify seed

The random number seed is the number that initializes the random number generator. The seed is 0 or a positive integer. To control the random number seed, set this parameter to:

- Not repeatable — The seed changes every time you simulate your model. The block resets the random seed using the MATLAB random number generator command:

  seed = randi(2^32-1);

- Repeatable — The seed is the same random number at the start of every simulation. The block sets the value using the same MATLAB random number generator command used by the Not repeatable parameter.

  When you add a Torque Noise Source block to your model from the Sources library, the block generates and stores a random value for the repeated seed. When you make a copy of the Torque Noise Source block from an existing block in a model, the copy generates a new random value for the repeated seed.

- Specify seed — The seed is a number that you specify using the **Seed** parameter. The **Seed** parameter is only available when you choose Specify seed for the **Repeatability** parameter.

Seed — Random number generation seed value
0 (default)

The seed must be 0 or a positive integer. This parameter is only available when you select Specify seed for the **Repeatability** parameter.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Force Noise Source | Random Number | Rotational Velocity Noise Source | Sinusoidal Torque Source | Translational Velocity Noise Source

Functions
randi

Introduced in R2016b

Torsional Spring-Damper

Rotational spring and damper coupling with Coulomb friction, locking, and hard stops
Library: Simscape / Driveline / Couplings & Drives

Description

The Torsional Spring-Damper block represents a rotational spring and damper that imposes a combination of internally generated torques between the two connected driveshaft axes, the rod and the case. The block defines the torque using the following behaviors:

- Linear damped spring
- Coulomb friction (including locking static friction)
- Hard-stop compliance

This block is a composite component that consists of these blocks:

Block	Contribution	Library
Loaded-Contact Rotational Friction	Coulomb friction	**Simscape** / **Driveline** / **Brakes & Detents** / **Rotational**
Rotational Damper, Rotational Spring	Damping	**Simscape** / **Foundation Library** / **Mechanical** / **Rotational Elements**
Rotational Spring	Spring	
Rotational Hard Stop	Hard stop	

For more information, see "Composite Components".

Assumptions and Limitations

- Including hard-stop and Coulomb friction enhances model fidelity, but reduces simulation speed. For more information, see "Driveline Simulation Performance".

Ports

Conserving

C — Follower port
mechanical rotational

Mechanical rotational port associated with the slider that travels between the stops installed on the case.

R — Base port
mechanical rotational

Mechanical rotational port associated with the rod.

Parameters

Spring-Damper

`Spring stiffness` — **Torsional spring stiffness**
`1000 N*m/rad` (default) | nonnegative scalar

Torsional spring stiffness that acts on connected driveshafts.

`Viscous friction coefficient` — **Torsional viscous friction**
`10 N*m/(rad/s)` (default) | nonnegative scalar

Torsional friction that acts on the connected driveshafts.

`Coulomb friction torque` — **Kinetic friction torque**
`0 N*m` (default) | nonnegative scalar

Constant kinetic friction torque that acts on the connected driveshafts. This constant torque opposes rotation at any velocity.

`Ratio of static to kinetic friction` — **Static friction with respect to kinetic friction**
`1.1` (default) | positive scalar

Constant ratio of static Coulomb friction torque to kinetic Coulomb friction torque that acts on connected driveshafts. The value must be greater than or equal to one.

`Velocity tolerance` — **Mating driveshaft differential speed tolerance**
`0.001 rad/s` (default) | positive scalar

Minimum relative angular speed below which the two connected driveshafts can lock and rotate together.

Hard Stops

`Hard stop` — **Simulate boundary hard stops**
`No hard stops - Suitable for HIL simulation` (default) | `Compliant hard stops`

Include or exclude hard-stop torque by selecting one of these options:

- `No hard stops - Suitable for HIL simulation` — Select this option to enhance simulation speed by excluding the hard-stop torque contribution.
- `Compliant hard stops` — Select this option to enhance model fidelity by including the hard stop torque contribution.

`Upper bound` — **Location of upper hard stop**
`10 deg` (default) | scalar

Upper hard-stop angular displacement from the zero-torque reference angle. The value must be greater than the **Lower bound** parameter.

Dependencies

To enable this parameter, set **Hard stop** to `Compliant hard stops`.

`Lower bound` — **Location of lower hard stop**
`-10 deg` (default) | scalar

Lower hard-stop angular displacement from the zero spring force reference angle. The value must be less than the **Upper bound** parameter.

Dependencies

To enable this parameter, set **Hard stop** to `Compliant hard stops`.

Contact stiffness — Hard stop proportional spring constant
`1e6 N*m/(rad/s)` (default) | nonnegative scalar

Hard-stop stiffness. This value only applies if the relative angle moves into the hard-stop region.

Dependencies

To enable this parameter, set **Hard stop** to `Compliant hard stops`.

Contact damping — Hard stop proportional damping
`10 (N*m)/(rad/s)` (default) | positive scalar

Hard-stop damping. This value only applies if the relative angle moves into the hard-stop region. The value must be greater than zero.

Dependencies

To enable this parameter, set **Hard stop** to `Compliant hard stops`.

Hard stop model — Hard stop transition behavior
`Stiffness and damping applied smoothly through transition region, damped rebound` (default) | `Full stiffness and damping applied at bounds, undamped rebound` `Full stiffness and damping applied at bounds, damped rebound`

Stiffness and rebound options for the hard stop model.

Dependencies

To enable this parameter, set **Hard stop** to `Compliant hard stops`.

Transition region — Region of partial hard stop effects
`0.001 rad` (default) | positive scalar

Distance from full compression or full extension where the effects of stiffness and damping are partially applied. When you set **Hard stop model** to `Stiffness and damping applied smoothly through transition region, damped rebound`, the block smoothly transitions the onset of stiffness and damping as the spring approaches full extension or full compression.

Dependencies

To enable this parameter, set **Hard stop** to `Compliant hard stops` and **Hard stop model** to `Stiffness and damping applied smoothly through transition region, damped rebound`.

Initial Conditions

Initial deformation — Starting deformation value
`0 deg` (default) | scalar

Initial deformation of the torsional spring relative to the zero-torque reference angle.

1-537

More About

Hardware-in-the-Loop Simulation

For the best simulation performance, use the **Hard Stops > Hard stop** parameter default setting, `No hard stops - Suitable for HIL simulation`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

Translational Damper

Faultable linear translational damper
Library: Simscape / Driveline / Couplings & Drives / Springs &
                    Dampers

Description

The Translational Damper block models a faultable linear translational viscous damper. From the start of the simulation, the block uses the nominal damping coefficient. If you enable faults for the block, the damping coefficient changes in response to one or both of these triggers:

- Simulation time — A fault occurs at a specified time.
- Simulation behavior — If the magnitude of the translational acceleration exceeds a specified maximum value, the block increments the number of shocks by one. A fault occurs if the number of shocks exceeds a specified maximum number of shocks.

If a fault trigger event occurs, the block uses the faulted damping coefficient for the remainder of the simulation. You can program the block to issue a fault report as a warning or error message.

The defining equations are

$$F = Dv$$

and

$$v = v_R - v_C,$$

where

- F is the force transmitted through the damper.
- D is the damping (viscous friction) coefficient.
- v is the relative velocity.
- v_R and v_C are the absolute velocities of terminals R and C, respectively.

The positive block direction is from port R to port C. Therefore, the torque is positive if it acts in the direction from R to C.

Thermal Modelling

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Thermal port** to Model.

Enabling thermal modelling exposes the **Thermal mass** parameter.

Ports

Conserving

R — Rod interface
mechanical translational

Mechanical translational conserving port associated with the rod interface.

C — Case interface
mechanical translational

Mechanical translational conserving port associated with the case interface.

H — Heat flow
thermal

Thermal conserving port associated with heat flow. Heat flow affects damper temperature, and therefore, power transmission efficiency.

Dependencies

To enable this port, set **Thermal port** to Model.

Parameters

Main

Damping coefficient — Damping coefficient for the no-fault condition
100 N/(m/s) (default)

The oscillation-reducing force due to viscous friction that the block applies if no trigger events occur. Specify a value that is greater than or equal to zero.

Faults

Enable faults — Fault mode capability
Off (default) | On

To enable the block to use a different damping coefficient based on the simulation time or when the magnitude of the translational acceleration exceeds the specified number of shocks, select On.

Dependencies

Selecting On makes these parameters visible:

- **Faulted damping coefficient**
- **Enable temporal fault trigger**
- **Enable behavioral fault trigger**
- **Reporting when fault occurs**

Faulted damping coefficient — Damping coefficient for the fault condition
1e6 N/(m/s) (default)

The oscillation-reducing force due to viscous friction that the block applies if a trigger event occurs. Specify a value that is greater than or equal to zero.

Dependencies

This parameter is only visible if you select On for the **Enable faults** parameter.

Enable temporal fault trigger — Fault trigger capability
Off (default) | On

To use the faulted damping coefficient starting at a specified time in the simulation, select On.

Dependencies

This parameter is only visible if you select On for the **Enable faults** parameter.

Selecting On makes the **Simulation time for fault event** parameter visible.

Simulation time for fault event — Temporal fault trigger
5 s (default)

Specify the time during the simulation that you want the block to start using the faulted damping coefficient.

Dependencies

This parameter is only visible if you select On for the **Enable faults** parameter and On for the **Enable temporal fault trigger** parameter.

Enable behavioral fault trigger — Fault trigger capability
Off (default) | On

To use the faulted damping coefficient when the number of sustained shocks exceeds a specified maximum number of shocks, select On.

Dependencies

This parameter is only visible if you select On for the **Enable faults** parameter.

Selecting On makes these parameters visible:

- **Maximum permissible acceleration**
- **Maximum number of shocks**

Maximum permissible acceleration — Behavioral fault trigger
10 m/s^2 (default)

The maximum value that the magnitude of the translational acceleration can reach without causing the block to increment the number of sustained shocks.

Dependencies

This parameter is only visible if you select On for the **Enable faults** parameter and On for the **Enable behavioral fault trigger** parameter.

Maximum number of shocks — Behavioral fault trigger
1 (default)

The maximum number of shocks that can be sustained without triggering the block to use the faulted damping coefficient.

Dependencies

This parameter is only visible if you select On for the **Enable faults** parameter and On for the **Enable behavioral fault trigger** parameter.

Reporting when fault occurs — Fault reporting
None (default) | Warning | Error

Select your reporting preference for the fault condition.

Dependencies

This parameter is only visible if you select On for the **Enable faults** parameter.

Thermal port

Thermal port — Thermal modelling
Omit (default) | Model

Option to model effects of heat flow and temperature change.

Thermal mass — Thermal mass
50 kJ/K (default) | postive scalar

Thermal energy required to change the component temperature by a unit increase. The thermal mass represents the ability of the damper to absorb and retain heat energy.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Nonlinear Rotational Damper | Nonlinear Translational Damper | Rotational Damper | Variable Rotational Damper | Variable Translational Damper

Introduced in R2016b

Translational Detent

Double-sided, spring-loaded translational detent
Library: Simscape / Driveline / Brakes & Detents / Translational

Description

The Translational Detent block represents a double-sided, spring-loaded translational detent with a ball and conical notch. The detent slider slides horizontally over the detent case. This interaction creates horizontal shear force. Inside of the slider is the conical notch, and the detent-case contains a vertical spring. The detent ball lies between the spring and the conical notch. The model also includes viscous damping and kinetic friction between the slider and case.

The figure demonstrates the geometry of the ball notch detent.
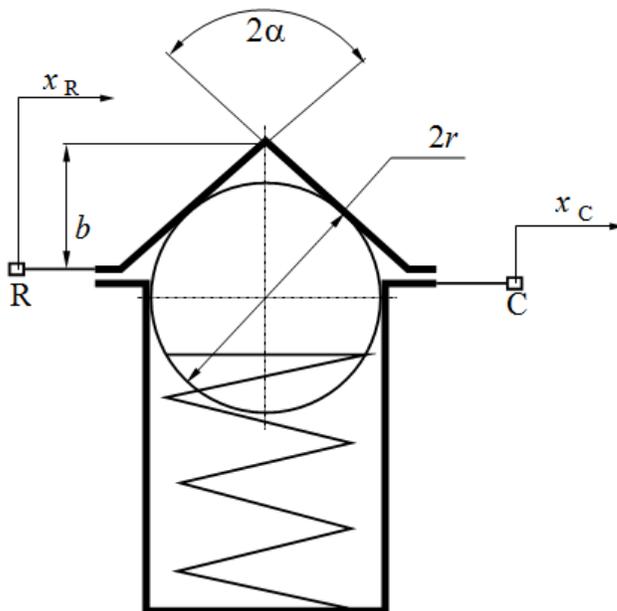
Here,

* *b* is the notch depth.
* *r* is the ball radius.
* *α* is the notch half angle.

Depending on how you parameterize the block, the shear force model is defined by geometric and spring parameters, by the peak force and notch width, or by a table lookup that specifies relative displacement versus force.

Translational Detent Parameterization

When you set **Parameterization** to `By peak force and notch width`, the block ensures that the force-relative displacement curve provides a continuous force and force derivative over the detent region. The peak forces are halfway between the detent center and detent edge, as the figure shows.
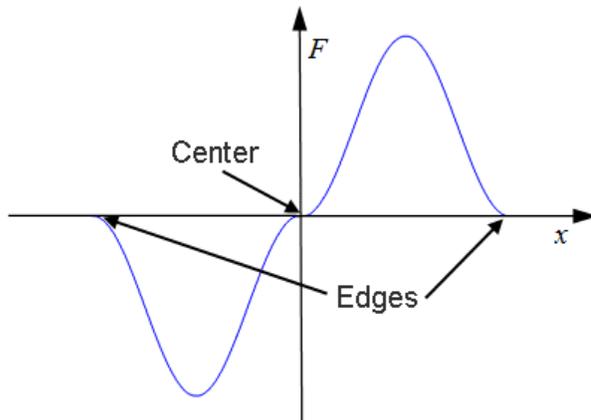
When you set **Parameterization** to `By geometry`, the block uses four regions to define the shear force. The figure displays these regions.
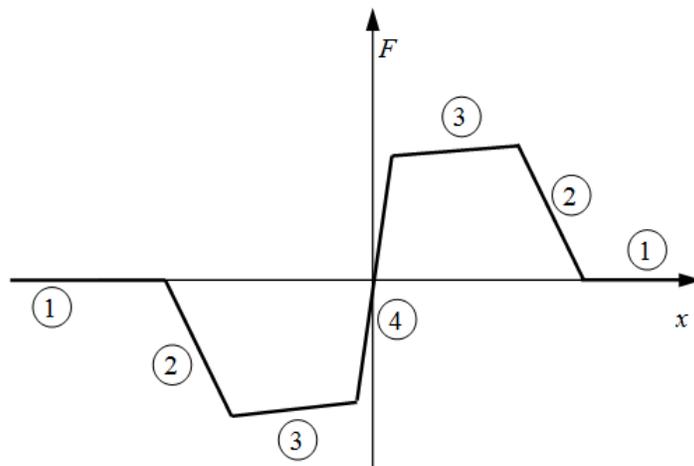
Because the ball is spherical and the notch is symmetric, the horizontal force versus displacement function is symmetric about the origin.

The table describes the ball position, contact angle, and shear force contributions for each region.

Region	Ball Position and Contact Angle	Shear Force
1	The ball is outside of the conical notch. The contact angle is vertical.	Viscous damping and kinetic friction only. No spring contribution.
2	The ball is entering notch and is in contact with the notch corner. The contact angle rotates from vertical.	Spring, viscous damping, and kinetic friction contributions. Peaks at transition from region 2 to region 3.

Reg ion	Ball Position and Contact Angle	Shear Force
3	The ball slides along the face of the notch. The contact angle is constant.	Force decreases as spring extends until reaching the spring preload force.
4	The ball moves from one face to the other. The contact angle switches direction.	Force switches direction. The shear reversal region width controls how smooth this switch is.

When you set **Parameterization** to `By table lookup`, you can create an arbitrary function that relates shear force to relative displacement. If you create such a function, consider these best practices.

Energy Conservation

To ensure that the detent conserves energy, the integral of the force-relative displacement curve must be zero.

Simulation Performance

Avoid discontinuities in the force-relative displacement function. The most important requirement is to have a shear reversal region of nonzero width, analogous to region 4 in the geometric parameterization.

Assumptions and Limitations

- The block does not account for inertia. Add mass externally to the **R** and **C** ports.
- When you set **Parameterization** to `By peak force and notch width` or `By table lookup`, the kinetic friction is independent of the detent normal force.

Ports

Conserving

C — Case
mechanical translational

Mechanical translational conserving port associated with the case.

R — Rod
mechanical translational

Mechanical translational conserving port associated with the slider or rod.

Parameters

Detent Characteristics

Parameterization — Detent characterization parameters
`By peak force and notch width` (default) | `By table lookup` | `By geometry`

Detent characteristic specification.

- `By peak force and notch width` — Specify detent characteristics by the peak shear force and the notch width.

- By `table lookup` — Define the detent characteristics by one-dimensional table lookup based on the relative displacement between the slider and case.
- By `geometry` — Define the detent characteristics by ball-notch geometry and dynamics.

Peak force — Maximum detent shear force
10 N (default) | positive scalar

Peak shear force developed inside the detent.

Dependencies

To enable this parameter, set **Parameterization** to By `peak force and notch width`.

Notch width — Shear force region
30 mm (default) | positive scalar

Width of the region where the shear force develops.

Dependencies

To enable this parameter, set **Parameterization** to By `peak force and notch width`.

Relative displacement vector — Displacement lookup vector
[-15 -12.3 -7.6 -1.1 1.1 7.6 12.3 15] mm | vector

Vector of input relative displacements that corresponds to the values in the **Force vector** parameter. This vector must be same the size as the **Force vector** parameter. You must specify the displacements in increasing order.

Dependencies

To enable this parameter, set **Parameterization** to By `table lookup`.

Force vector — Force lookup vector
[0, 0, -8.365, -5.87, 5.87, 8.365, 0, 0] N (default) | vector

Vector of input values for the shear force that corresponds to the **Relative displacement vector** parameter.

Dependencies

To enable this parameter, set **Parameterization** to By `table lookup`.

Interpolation method — Method of interpolation between breakpoint values for table lookup
Smooth (default) | Linear

Lookup table breakpoint interpolation. The block uses the `tablelookup` function to model nonlinearity by using array data to map input values to output values.

- `Linear` — Select this option for the lowest computational cost.
- `Smooth` — Select this option to produce a continuous curve with continuous first-order derivatives.

For more information, see `tablelookup`.

Dependencies

To enable this parameter, set **Parameterization** to By table lookup.

Ball diameter — Ball diameter measurement for ball-notch geometry
12 mm (default) | positive scalar

Cross sectional diameter of the ball..

Dependencies

To enable this parameter, set **Parameterization** to By geometry.

Notch angle — Notch angle measurement for ball-notch geometry
120 deg (default) | positive scalar

Full angle of the notch.

Dependencies

To enable this parameter, set **Parameterization** to By geometry.

Notch depth — Notch depth measurement for ball-notch geometry
4.5 mm (default) | nonnegative scalar

Depth of the notch in the slider.

Dependencies

To enable this parameter, set **Parameterization** to By geometry.

Preload force — Spring preload force measurement for ball-notch geometry
15 N (default) | nonnegative scalar

Force to preload the detent spring.

Dependencies

To enable this parameter, set **Parameterization** to By geometry.

Spring stiffness — Spring stiffness measurement for ball-notch geometry
3000 N/m (default) | nonnegative scalar

Stiffness of the detent spring.

Dependencies

To enable this parameter, set **Parameterization** to By geometry.

Shear reversal region — Shear reversal region length measurement for ball-notch geometry
0.1 mm (default) | nonnegative scalar

Width of the region between notch faces.

Dependencies

To enable this parameter, set **Parameterization** to By geometry.

Initial relative position — Starting location
`0 mm` (default) | scalar

Relative displacement of the slider and case when simulation starts.

Friction

Viscous friction coefficient — Proportional viscous friction
`0.1 N/(m/s)` (default) | nonnegative scalar

Viscous friction coefficient for the ball-notch contact.

Friction to peak force ratio — Kinetic friction ratio
`0.01` (default) | positive scalar

Ratio by which to compute kinetic friction. The kinetic friction is this ratio multiplied by the peak shear force.

Dependencies

To enable this parameter, set **Parameterization** to `By peak force and notch width` or `By table lookup`.

Kinetic friction coefficient — Proportional kinetic friction
`0.1` (default) | positive scalar

Kinetic friction coefficient for the ball-notch contact.

Dependencies

To enable this parameter, set **Parameterization** to `By geometry`.

Friction velocity threshold — Velocity that relates to peak kinetic friction
`0.05 m/s` (default) | positive scalar

Relative velocity required for peak kinetic friction in the detent.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Loaded-Contact Translational Friction | Rotational Detent

Introduced in R2011b

Translational Power Sensor

Mechanical sensor used to measure average or instantaneous translational power
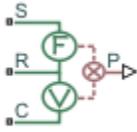
Library

Sensors

Description

The block represents a sensor that measures the translational power change due to one or more driveline components connected between ports R and C. The power output can be an instantaneous or average value. Two averaging options are available: period and vibration averaging.

The instantaneous power is the product of the current velocity and force. Use the instantaneous power to analyze peak power transmission and identify heavily loaded components.

The time-averaged power is the integral of the instantaneous power over one sample period, divided by that period. For simulation times smaller than the sample period, the average power is zero. The power is constant for each sample period.

The vibration-averaged power is the integral of the power over one vibration period, divided by that period. The vibration period is the time between two consecutive crossings from negative to positive velocity values. Use this averaging method only for oscillations with velocity passing through zero. If the vibration has a variable frequency, the averaging period can change. For simulation times smaller than the vibration period, the average power is zero. The power output is constant for each vibration period.

Parameters

Measurement Type

Select either instantaneous or average power measurement. The default type is `Instantaneous Power`.

- `Instantaneous power` — Output the power measured for the current time step.
- `Average power over a specified period` — Output the power averaged over a fixed time period.

    Averaging period

    Enter the time period used to compute the average power. Typical values include the cycle length of a reciprocating device or the actuation force.

    The default value is `1`. The default unit is `s`.

- `Average power over a vibration`— Output the power averaged over one vibration period.

    Maximum vibration frequency

    Enter the maximum vibration frequency over which to average the power. Very high values can reduce simulation speed. Very low values can result in lower power measurements.

    The default value is `1000`. The default unit is `Hz`.

Ports

Port	Description
P	Physical signal output port for translational power
S	Mechanical translational conserving port associated with driving source
R	Mechanical translational conserving port associated with loaded driveline components
C	Mechanical translational conserving port associated with case holding loaded driveline components

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Rotational Power Sensor

Introduced in R2013a

Translational Velocity Noise Source

Produce zero-mean normally (Gaussian) distributed translational velocity
Library: Simscape / Driveline / Sources

Description

The Translational Velocity Noise Source block generates zero-mean normally (Gaussian) distributed translational velocity using the Random Number source in the Simscape Foundation library. The translational velocity produced by the block depends on two factors:

- The noise power spectral density input to the block
- A random number seed that you control using the **Repeatability** and, as applicable, **Seed** parameters

The defining equation for the translational velocity that the block generates is

$$v = \sqrt{PSD/2}\frac{N(0, 1)}{\sqrt{h}},$$

where:

- v is the output translational velocity.
- PSD is the single-sided noise power spectral density.
- N is a Gaussian random number with zero mean and standard deviation of one.
- h is the sampling interval.

Ports

Input

P — Single-sided noise power spectral density
physical signal

The power spectral density (PSD) of noise is the average noise power per unit of bandwidth, in $(m/s)^2$/Hz.

Conserving

B — Base interface
mechanical translational

Mechanical translational conserving port associated with the base interface.

F — Follower interface
mechanical translational

Mechanical translational conserving port associated with the follower interface.

Parameters

Sample time — Time step period and offset
1e-3 s (default) | [*step*, *offset*] s

The values of the time *step* period and the initial time *offset*. If you specify a scalar value for *step*, the block assumes an *offset* value of 0.To specify a nonzero value for the initial time offset, specify the parameter values using the vector [*step*, *offset*]. The *offset* value must be less than the *step* value and greater than or equal to zero.

Repeatability — Random number seed control
Not repeatable (default) | Repeatable | Specify seed

The random number seed is the number that initializes the random number generator. The seed is 0 or a positive integer. To control the random number seed, set this parameter to:

- Not repeatable — The seed changes every time you simulate your model. The block resets the random seed using the MATLAB random number generator command:

  seed = randi(2^32-1);

- Repeatable — The seed is the same random number at the start of every simulation. The block sets the value using the same MATLAB random number generator command used by the Not repeatable parameter.

  When you add a Translational Velocity Noise Source block to your model from the Sources library, the block generates and stores a random value for the repeated seed. When you make a copy of the Translational Velocity Noise Source block from an existing block in a model, the copy generates a new random value for the repeated seed.

- Specify seed — The seed is a number that you specify using the **Seed** parameter. The **Seed** parameter is only available when you choose Specify seed for the **Repeatability** parameter.

Seed — Random number generation seed value
0 (default)

The seed must be 0 or a positive integer. This parameter is only available when you select Specify seed for the **Repeatability** parameter.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Force Noise Source | Random Number | Rotational Velocity Noise Source | Sinusoidal Translational Velocity Source | Torque Noise Source

Functions
randi

Introduced in R2016b

Unbalanced Load

Load with angle-dependent rotational inertia

Library

Simscape / Driveline / Inertias & Loads

Description

This block models the effects of a load whose rotational inertia varies instantaneously with the rotation angle. Examples include the wobbling of a spinning axle and shaking of an off-center rotating machine. You can specify the load inertia in terms of its rotation path or as an angle-inertia lookup table.

The block provides two rotation path parameterizations. One assumes circular motion of the unbalanced load. The other assumes elliptical motion. In each case, the distance between the unbalanced load and the main rotation axis varies with the rotation angle. The varying distance causes the moment of inertia, given by the parallel-axis theorem, to vary during rotation:

$$I(\theta) = I_0 + Mr^2(\theta),$$

where:

- I is the moment of inertia about the main rotation axis.
- I_0 is the moment of inertia about the intrinsic rotation axis of the unbalanced load.
- M is the total mass of the unbalanced load.
- r is the distance between the unbalanced load and the main rotation axis.

The angle dependence of the moment of inertia leads to an angle dependence of the inertial torque acting on the unbalanced load:

$$T_I(\theta) = I(\theta)\dot{\omega},$$

where:

- T_I is the inertial torque.
- ω is the angular velocity of the load.

If the load moves in a circular path, its distance to the main rotation axis is given by

$$r(\theta) = \sqrt{(r_C cos(\theta) + r_A)^2 + (r_C sin(\theta))^2},$$

where:

- r_A is the distance between the main rotation axis and the offset rotation axis.
- r_C is the radius of the circular path.
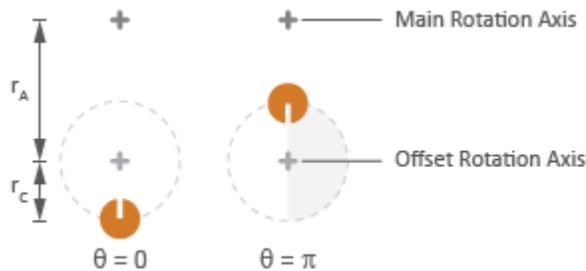
Circular Rotation Path

If the load moves in an elliptical path, its distance to the main rotation axis is given by

$$r(\theta) = \sqrt{(r_{\parallel}cos(\theta) + r_A)^2 + (r_{\perp}sin(\theta))^2},$$

where:

- $r_{\parallel}$ is the radius of the elliptical path in line with the distance from the main rotation axis.
- $r_{\perp}$ is radius of the elliptical path perpendicular to the distance from the main rotation axis.
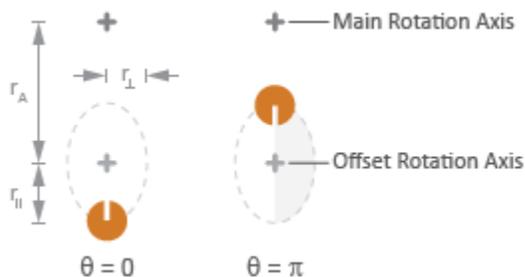
Elliptical Rotation Path

Ports

The block has one conserving rotational port. This port identifies the location of the unbalanced load in a driveline model.

Parameters

Inertia

Parameterization

> Choice of parameterization for modeling the moment of inertia of the unbalanced load. Options include `Circular rotation`, `Elliptical rotation`, and `By table lookup`. The default setting is `Circular rotation`.

Select `Circular rotation` or `Elliptical rotation` to parameterize the load inertia in terms of the load's rotation path. Select `By table lookup` to provide the load inertia directly across its rotation path.

Moment of inertia about own axis

Moment of inertia about the intrinsic rotation axis of the unbalanced load. This moment of inertia is assumed constant throughout simulation. This parameter appears only when **Parameterization** is set to `Circular rotation`. The default value is 1 kg*m2.

Mass

Mass of the unbalanced load. This parameter appears only when **Parameterization** is set to `Circular rotation`. The default value is 1 kg.

Distance from axis to center of rotation

Distance between the main rotation axis and the center of the rotation path. This parameter appears only when **Parameterization** is set to `Circular rotation`. The default value is 0.01 m.

Radius of rotation

Radius of the circular rotation path traversed by the unbalanced load. This parameter appears only when **Parameterization** is set to `Circular rotation`. The default value is 0.01 m.

Radius of rotation inline with distance from axis

Length of the semimajor axis of the elliptical rotation path traversed by the unbalanced load. The semimajor axis is in line with the line segment joining the main rotation axis and the elliptical rotation center. This parameter appears only when **Parameterization** is set to `Elliptical rotation`. The default value is 0.01 m.

Radius of rotation perpendicular to distance from axis

Length of the semiminor axis of the elliptical rotation path traversed by the unbalanced load. The semiminor axis is perpendicular to the line segment joining the main rotation axis and the elliptical rotation center. This parameter appears only when **Parameterization** is set to `Elliptical rotation`.The default value is 0.005 m.

Angle vector

Array of angular positions at which to specify the load inertia. The block calculates the load inertia between the angles specified using linear interpolation. This parameter appears only when **Parameterization** is set to `By table lookup`. The default vector is [0,pi,3*pi/2] rad.

Inertia vector

Array of load inertias about the intrinsic rotation axis of the load. Each inertia corresponds to an angle specified in the **Angle vector** parameter. This parameter appears only when **Parameterization** is set to `By table lookup`. The default vector is [1,10,1] kg*m^2.

Initial Conditions

Initialization

Choice of parameter through which to specify the initial state of the unbalanced load. Options include `Specify angular momentum` and `Specify angular velocity`.

Initial angular momentum

Angular momentum of the unbalanced load at the start of simulation. This parameter appears only when the **Initialization** parameter is set to `Specify angular momentum`. The default value is 0 m^2*kg/s.

Initial angular velocity

Angular velocity of the unbalanced load at the start of simulation. This parameter appears only when the **Initialization** parameter is set to `Specify angular velocity`. The default value is `0` rad/s.

Initial angle

Angular position of the unbalanced load at the start of simulation. The initial angle determines the initial inertia of the unbalanced load. The default value is `0` rad.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Variable Inertia | Variable Mass

Introduced in R2015b

Unidirectional Clutch

Clutch that transmits power in a single direction
Library: Simscape / Driveline / Clutches

Description

The Unidirectional Clutch block models an ideal one-way clutch based on a coupled ring-hub. It transmits torque when the relative velocity of hub and ring becomes positive or negative, depending on how you set the transmission direction. For model details, see "Equations" on page 1-558.

Equations

The figure displays the geometry of the unidirectional clutch. The table summarizes the unidirectional clutch variables.
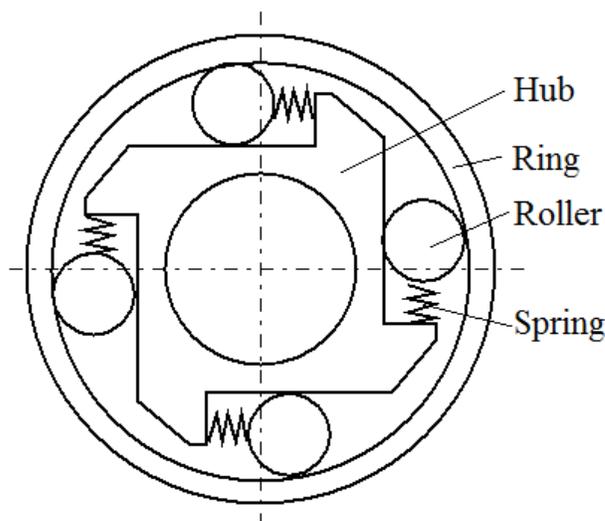
If locked, the clutch transmits torque from the hub to the ring. If unlocked, the clutch transmits no torque. The clutch does not have a state of being unlocked but engaged (transmitting torque). The relative angular velocity (slip), ω, is the angular velocity of hub, ω_H, relative to the velocity of the ring, ω_R, that is, $\omega = \omega_H - \omega_R$.

The Unidirectional Clutch block implements an ideal model of a one-way clutch and does not include any internal dynamics, such as the rollers and springs.

Locking Condition

The transmission direction determines the condition for locking.

• If you choose transmission for the positive direction, the clutch locks if $\omega = \omega_H - \omega_R \geq 0$ and becomes zero from positive values. The slip ω then remains zero, and the clutch transfers torque.

If the driveline dynamics then causes ω < 0 later in the simulation, the clutch abruptly unlocks and stops transferring torque.

- If you choose transmission for the negative direction, the clutch locks if $\omega = \omega_H - \omega_R \leq 0$ and becomes zero from negative values. The slip ω then remains zero, and the clutch transfers torque.

  If the driveline dynamics then causes ω > 0 later in the simulation, the clutch abruptly unlocks and stops transferring torque.

For example, locking and torque transfer in the positive direction as follows.

- When the hub rotates faster than the ring (ω > 0), the small gap prearranged between the hub and the ring is cleared. The spring-loaded rollers wedge between the hub and the ring, causing the two to contact, lock, and rotate together; ω resets to zero. (The rotation convention in the figure is clockwise.)
- If the ring rotates faster than the hub (ω < 0), the rollers are pushed out of contact and the clutch unlocks.

For a clutch schematic, see "Equations" on page 1-558.

Power Dissipated by the Clutch

The unidirectional clutch is ideal and does not dissipate power. Either the clutch is unlocked and applies no kinetic friction, or it is locked (no slip) and applies only static friction.

Ports

Conserving

H — Hub velocity
mechanical rotational

Velocity of the hub driveshaft axis.

R — Ring velocity
mechanical rotational

Velocity of the ring driveshaft axis.

Parameters

Transmission direction — Relative rotation direction
From hub to ring for positive rotation (default) | From hub to ring for negative rotation

Relative rotation direction for which the clutch locks and transfers torque.

Initial state — Clutch initial state
Unlocked (default) | Locked

Clutch state at the start of simulation. The clutch can be in one of two states, locked and unlocked. A locked clutch constrains the base and follower shafts to spin at the same velocity, that is, as a single unit. An unlocked clutch allows the two shafts to spin at different velocities, resulting in slip between the clutch plates.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Dog Clutch | Fundamental Friction Clutch

Topics
"Ratchet Leadscrew Mechanism"
"Gear Coupling Control Using Clutches"

Introduced in R2011a

# Universal Joint

Rotational coupling between two driveline shafts
Library: Simscape / Driveline / Couplings & Drives
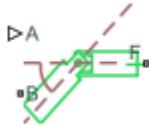
Description

The Universal Joint block represents a rotational coupling between two driveline shafts. The coupling transfers torque between the shafts so they spin as a unit under an applied load. Two rotational degrees of freedom, internal to the coupling, allow the shafts to connect at an angle. This intersection angle varies according to the physical signal input from port **A**. Optional compliance, modeled as a parallel spring-damper set, allows the coupling to deform under load.

You can use the Universal Joint block as a connection between two rotational driveline components— for example, between the driving and driven shafts in an automobile drive train.
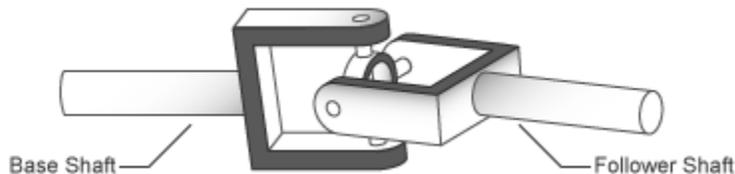
The ratio of the shaft angular velocities depends on two parameters: the intersection angle between the two shafts and the rotation angle of the base shaft. A physical signal input provides the intersection angle while a dialog box parameter provides the initial base shaft angle. These two angles fix the ratio of the two shaft angular velocities according to the nonlinear equation:

$$\omega_F = \frac{\cos(A)}{1 - \sin^2(A) \cdot \cos^2(\theta_B)} \omega_B,$$

where:

- ω_F is the angular velocity of the follower shaft about its length axis.
- ω_B is the angular velocity of the base shaft about its length axis.
- θ_B is the rotation angle of the base shaft about its length axis.
- A is the intersection angle between base and follower shafts about the base shaft pin.

The two schematics in the figure illustrate the equation parameters. In each schematic, the left shaft represents the base shaft, while the right shaft represents the follower shaft. The right schematic shows the coupling seen in the left schematic after the shafts spin 90° about their length axes (dashed line segments).
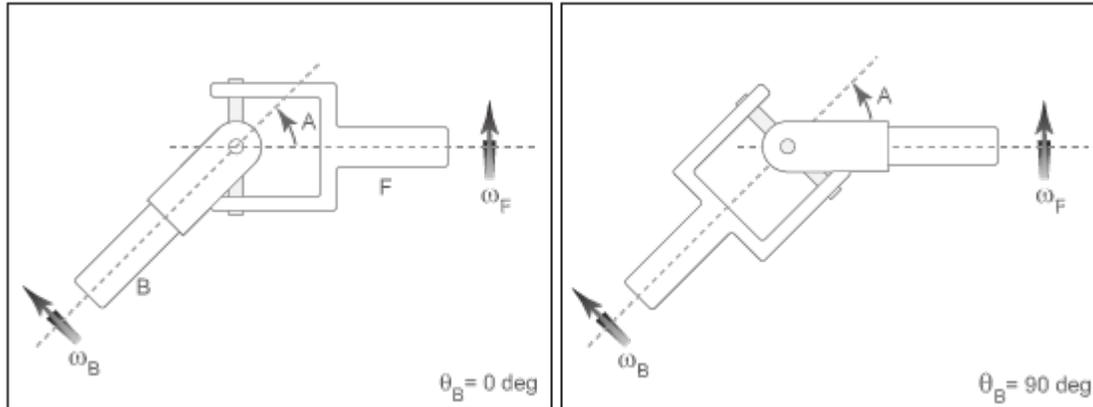
In the figure, the intersection angle is the angle between the two shafts about the pin of the base shaft. The absolute value of this angle must fall in the range `0 ≤ A < Maximum intersection angle`. The base shaft angle is the angle of the base shaft about its length axis. The base shaft angle is also the time-integral of the base shaft angular velocity, ω_B.

Ports

Conserving

B — Base shaft
mechanical rotational

Conserving rotational port associated with the base shaft.

F — Follower shaft
mechanical rotational

Conserving rotational port associated with the folower shaft.

Input

A — Intersection angle
physical signal

Physical signal input port for the intersection angle.

Parameters

Maximum joint angle — Maximum joint intersection angle, *A*
pi/4 rad (default) | 0 ≤ A < pi/2 | scalar

Maximum intersection angle the joint allows. This angle measures the rotation between base and follower shafts about the base shaft pin. The value of this angle must fall in the range $0 ≤ A < pi/2$ rad.

Joint compliance — Compliance model
Off (default) | On

Compliance model for the block:

- `Off` — Do not model stiffness and damping.
- `On` — Model stiffness and damping.

Dependencies

If this parameter is set `On`, related parameters are enabled.

Joint stiffness — Stiffness coefficient
`1e6 N*m/rad` (default) | scalar

Linear spring stiffness of the joint. The spring stiffness accounts for elastic energy storage in the joint due to material compliance.

Dependencies

This parameter is enabled when **Joint compliance** is set to `On`.

Joint damping — Damping coefficient
`1e3 N*m/(rad/s)` (default) | scalar

Linear damping coefficient of the joint. The damping coefficient accounts for energy dissipation in the joint due to material compliance.

Dependencies

This parameter is enabled when **Joint compliance** is set to `On`.

Initial base shaft angle — Initial base shaft angle
`0 rad` (default) | scalar

Rotation angle of the base shaft about its length axis at the beginning of simulation.

Initial torque from base to follower shaft — Initial torque
`0 N*m` (default) | scalar

Torque that the base shaft transfers to the follower shaft at the beginning of simulation. This torque determines the initial state of material compliance at the joint. Set this value to greater than zero to preload the shafts with torque. Changing this value alters the initial transient response due to material compliance.

Dependencies

This parameter is enabled when **Jointcompliance** is set to `On`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Belt Drive | Belt Pulley | Chain Drive | Flexible Shaft | Rope Drum

Introduced in R2013b

Variable Inertia

Time-varying inertia specified as physical signal input

Library

Inertias & Loads

Description

This block models the effects of a time-varying inertia. Components with variable inertia include satellites unfolding arrays of solar panels. A physical signal input port provides the instantaneous inertia at each time step throughout simulation. To minimize solver errors, avoid discontinuities and abrupt changes in the input signal.

As the component inertia changes, so does the inertial torque acting on that component:

$$T_I = I\dot{\omega},$$

where:

- T_I is the inertial torque.
- I is the variable inertia.
- ω is the angular velocity of the inertia.

The physical signal input saturates at the minimum inertia specified in the block dialog box. Whenever the input falls below the minimum inertia, the block performs its calculations on the minimum inertia instead. The inertial torque becomes:

$$T_I = I_{Min}\dot{\omega},$$

where:

- I_{Min} is the minimum inertia allowed.

The angular momentum of the variable inertia is conserved. In the absence of external torques, a decrease in the moment of inertia leads to a proportional increase in angular velocity. The angular momentum remains constant until a net external torque causes it to change.

Variables

Use the **Variables** tab to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Ports

The block has one conserving rotational port and one physical signal input port. The conserving rotational port identifies the driveline component whose inertia varies with time. The physical signal input port provides the means to specify the inertia as a function of time.

Parameters

Minimum inertia

Lowest value that the variable inertia can take on during simulation. The physical signal input saturates at this value. If the input falls below the minimum inertia, it is set equal to the minimum inertia value. The default value is `0.1 kg*m^2`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Unbalanced Load | Variable Mass

Introduced in R2015b

Variable Mass

Time-varying mass specified as physical signal input

Library

Inertias & Loads

Description

This block models the effects of a time-varying mass. Components with variable mass include fuel tanks with progressively lower fuel contents. A physical signal input port provides the instantaneous mass at each time step throughout simulation. To minimize solver errors, avoid discontinuities and abrupt changes in the input signal.

As the component mass changes, so does the inertial force acting on that component:

$$F_I = m\dot{v},$$

where:

- F_I is the inertial force.
- m is the variable mass.
- v is the velocity of the mass.

The physical signal input saturates at the minimum mass specified in the block dialog box. Whenever the input falls below the minimum mass, the block performs its calculations on the minimum mass instead. The inertial force becomes:

$$F_I = m_{Min}\dot{v},$$

where:

- m_{Min} is the minimum mass allowed.

The linear momentum of the variable mass is conserved. In the absence of external forces, a decrease in mass leads to a proportional increase in linear velocity. The linear momentum remains constant until a net external force causes it to change.

Variables

Use the **Variables** tab to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Ports

The block has one conserving translational port and one physical signal input port. The conserving translational port identifies the driveline component whose mass varies with time. The physical signal input port provides the means to specify the mass as a function of time.

Parameters

Minimum mass

Lowest value that the variable mass can take on during simulation. The physical signal input saturates at this value. If the input falls below the minimum mass, it is set equal to the minimum mass value. The default value is `0.1 kg`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Unbalanced Load | Variable Inertia

Introduced in R2015b

Variable Ratio Transmission

Dynamic gearbox with variable and controllable gear ratio, transmission compliance, and friction losses

Library: Simscape / Driveline / Couplings & Drives
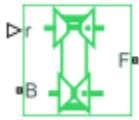
Description

The Variable Ratio Transmission block represents a gearbox that dynamically transfers motion and torque between the two connected driveshaft axes, the base and the follower.

When you choose to have the block ignore the dynamics of transmission compliance, it constrains the driveshafts to corotate with a variable gear ratio that you control. You can choose whether the follower axis rotates in the same or opposite direction as the base axis. If the follower and base axis rotate in the same direction, ω_F and ω_B have the same sign. If the follower and base axis rotate in opposite directions, ω_F and ω_B have opposite signs.

Transmission compliance introduces internal time delay between the axis motions. Unlike a gear, a variable ratio transmission does not act as a kinematic constraint. You can also control the torque loss caused by transmission and viscous losses.

Ideal Motion and Torque Transfer

The Variable Ratio Transmission block dynamically transfers motion and torque between the base shaft and the follower shaft.

If the relative compliance ϕ between the axes is absent, the block is equivalent to a gear with a variable ratio $g_{FB}(t)$. Such a gear imposes a time-dependent kinematic constraint on the motions of the two driveshafts:

$$\omega_B = \pm g_{FB}(t)\omega_F \tau_B = \pm g_{FB}(t)\tau_F$$

However, the Variable Ratio Transmission does include compliance between the axes. Dynamic motion and torque transfer replace the kinematic constraint, with a nonzero ϕ that dynamically responds through the base compliance parameters k_p and k_v:

$$\frac{d\phi}{dt} = \pm g_{FB}(t)\omega_F - \omega_B, \tau_B = -k_p\phi(t) - k_v\frac{d\phi}{dt},$$

$$\pm g_{FB}(t)\tau_B + \tau_F - \tau_{loss} = 0.$$

$\tau_{loss} = 0$ in the ideal case.

Estimating Compliance Parameters

- You can estimate the base angular compliance k_p from the transmission time constant t_c and inertia J.

$$k_p = J\left(\frac{2\pi}{t_c}\right)^2$$

- You can estimate the base angular velocity compliance k_v from the transmission time constant t_c, inertia J, and damping coefficient C.

$$k_v = 2C\frac{t_c}{2\pi} = 2C\sqrt{J/k_p}$$

Nonideal Torque Transfer and Losses

With nonideal torque transfer, $\tau_{loss} \neq 0$. The Variable Ratio Transmission block models losses similarly to nonideal gears. For general information about nonideal gear modeling, see "Model Gears with Losses".

In a nonideal gearbox, the angular velocity and compliance dynamics remain the same as in the ideal case. The transferred torque and power are reduced by:

- Coulomb friction, such as the friction between belt and wheel, or internal belt losses due to stretching, characterized by an efficiency η.
- Viscous coupling of driveshafts with bearings, parametrized by viscous friction coefficients μ.

$$\tau_{loss} = \tau_{Coul}\tanh\left(4\omega_{out}/\omega_{th}\right) + \mu_B\omega_B + \mu_F\omega_F$$

$$\tau_{Coul} = |\tau_F|(1 - \eta)$$

When the angular velocity changes sign, the hyperbolic tangent function smooths the change in the Coulomb friction torque.

Power Flow	Power Loss Condition	Output Driveshaft ω_{out}
Forward	$\omega_B \tau_B > \omega_F \tau_F$	Follower, ω_F
Reverse	$\omega_B \tau_B < \omega_F \tau_F$	Base, ω_B

Meshing Efficiency

The block only fully applies the friction loss represented by efficiency η if the absolute value of the follower angular velocity ω_F is greater than a velocity threshold ω_{th}.

If this absolute velocity is less than ω_{th}, the block smooths the efficiency to one at zero velocity.

Ports

Input

r — Input-to-output shaft velocity ratio, unitless
physical signal

Physical signal port associated with the input to output shaft ratio.

Conserving

B — Base driveshaft
mechanical rotational

Mechanical rotational conserving port associated with the base driveshaft.

F — Follower driveshaft
mechanical rotational

Mechanical rotational conserving port associated with the follower driveshaft.

Parameters

Main

Output shaft rotates — Output shaft direction versus input shaft direction
In same direction as input shaft (default) | In opposite direction to input shaft

Output driveshaft rotation relative to the input driveshaft.

Compliance

Transmission stiffness at base (B) — Base spring rate coefficient
3e5 N*m/rad (default) | positive scalar

Reciprocal of the transmission angular compliance k_p measured at the base.

Transmission damping at base (B) — Base damping coefficient
0.05 N*m/(rad/s) (default) | positive scalar

Reciprocal of the transmission angular compliance damping k_v measured at the base.

Initial input torque at base (B) — Starting torque at base
0 N*m (default) | scalar

Torque applied at the base driveshaft at the start of simulation.

Transmission Losses

Losses model — Optional loss modeling
No losses — Suitable for HIL simulation (default) | Constant efficiency

Implementation of friction losses from nonideal torque transfer.

- No losses — Suitable for HIL simulation — Torque transfer is ideal.
- Constant efficiency — The block reduces gear box torque transfer by a constant efficiency η satisfying $0 < \eta \le 1$.

Efficiency — Base-to-follower torque transfer efficiency
0.8 (default) | positive scalar

Effective torque transfer efficiency η between the base and follower driveshafts.

Dependencies

To enable this parameter, set **Losses model** to Constant efficiency.

Follower angular velocity threshold — Point at which block applies full efficiency loss
0.01 rad/s (default) | positive scalar

Absolute angular velocity threshold ω_{th} above which full efficiency loss is applied, for follower velocity ω_F.

Dependencies

To enable this parameter, set **Losses model** to `Constant efficiency`.

Viscous Losses

Viscous friction coefficients at base (B) and follower (F) — Base and follower \viscous friction vector
[0, 0] N*m/(rad/s) (default) | vector of nonnegative elements

Vector of viscous damping coefficients [μ_B, μ_F] applied at the base and follower driveshafts, respectively.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Simple Gear

Introduced in R2011a

Variable Rotational Damper

Rotational damper with variable damping coefficient
Library: Simscape / Driveline / Couplings & Drives / Springs & Dampers
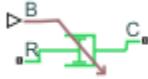
Description

The Variable Rotational Damper block represents a rotational viscous damper with a variable damping coefficient. A physical signal input port provides the magnitude of the damping coefficient. The magnitude of the damping torque is equal to the product of the physical signal input and the relative angular velocity between the two rotational conserving ports, **R** and **C**. A minimum damping coefficient prevents nonphysical negative damping values.

The damping torque satisfies the following expression:

$$T = \begin{cases} B \cdot \omega & B \geq B_{\min} \\ B_{\min} \cdot \omega & B < B_{\min} \end{cases},$$

where

- T is the torque transmitted through damper.
- B is the viscous damping coefficient.
- ω is the relative angular velocity measured between the two rotational conserving ports according to $\omega = \omega_R - \omega_C$.
- ω_R is the angular velocity of the rotational conserving port **R**.
- ω_C is the angular velocity of the rotational conserving port **C**.

The block applies equal and opposite damping torques on the rotational conserving ports. The sign of the damping torque acting on port **R** is equal to the sign of the relative angular velocity. A positive relative angular velocity corresponds to a positive damping torque acting on port **R**, and a negative damping torque of equal magnitude acting on port **C**.

Assumptions and Limitations

- The block represents only viscous damping.
- The value of the damping coefficient must be equal to or greater than zero.

Ports

Input

B — Variable damping coefficient
physical signal

Physical signal input port associated with the variable damping coefficient.

Conserving

C — Base port
rotational mechanical

Rotational mechanical conserving port associated with the base.

R — Follower port
rotational mechanical

Rotational mechanical conserving port associated with the follower.

Parameters

Minimum damping coefficient — Damping limit value
0 N*m/(rad/s) (default) | nonnegative scalar

Minimum value allowed for the damping coefficient. The physical signal input saturates below this level.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Nonlinear Rotational Damper | Nonlinear Translational Damper | Rotational Damper | Translational Damper | Variable Translational Damper

Introduced in R2013a

Variable Rotational Spring

Rotational spring with variable spring stiffness

Library

Couplings & Drives/Springs & Dampers

Description

The block represents a torsional spring with variable spring stiffness. A physical signal input port provides the value of the variable spring stiffness. The magnitude of the spring torque is equal to the product of the physical signal input and the relative angular displacement between the two rotational conserving ports. A minimum spring rate value prevents non-physical values.

The rotational spring torque satisfies the following expression:

$$T = \begin{cases} K \cdot \theta & K \geq K_{\min} \\ K_{\min} \cdot \theta & K < K_{\min} \end{cases},$$

The parameters are:

- T — Torque transmitted through the spring
- K — Spring rate (stiffness)
- K_{min} — Minimum allowable spring rate (stiffness)
- θ — Relative angular displacement measured between the two rotational conserving ports according to $\theta = \theta_{init} + \theta_R - \theta_C$
- θ_{init} — Initial angular deformation between the two rotational conserving ports
- θ_R — Absolute angular displacement of rotational conserving port R
- θ_C — Absolute angular displacement of rotational conserving port C

The block applies equal and opposite spring torques on the two rotational conserving ports. The sign of the spring torque acting on port R is equal to the sign of the relative angular velocity. A positive relative angular velocity corresponds to a positive damping torque acting on port R, and a negative damping torque of equal magnitude acting on port C.

Variables

Use the **Variables** tab to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Assumptions and Limitations

The value of the spring rate must be greater than zero.

Ports

K

Physical signal input port representing the variable spring stiffness.

C

Rotational conserving port.

R

Rotational conserving port.

Parameters

Minimum spring rate

Minimum value allowed for the spring rate (stiffness). The physical signal input saturates below the specified value. The parameter must be greater than zero. The default value is `0.01`. The default unit is `(N*m/rad)`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Nonlinear Rotational Spring | Variable Translational Spring | Nonlinear Translational Spring

Introduced in R2013a

Variable Translational Damper

Translational viscous damper with variable damping coefficient

Library

Couplings & Drives/Springs & Dampers

Description

The block represents a translational viscous damper with variable damping coefficient. A physical signal input port provides the magnitude of the translational damping coefficient. The magnitude of the damping force is equal to the product of the physical signal input and the relative linear velocity between the two translational conserving ports. A minimum damping coefficient prevents non-physical negative damping values.

The translational damping force satisfies the following expression:

$$F = \begin{cases} B \cdot v, & B \geq B_{\min} \\ B_{\min} \cdot v, & B < B_{\min} \end{cases},$$

The parameters are:

- F — Force transmitted through the translational damper between the two translational conserving ports
- B — Viscous damping coefficient
- B_{min} — Minimum allowed damping coefficient
- v — Relative linear velocity measured between the two translational conserving ports
- v_R — Linear velocity of port R
- v_C — Linear velocity of port C

The block applies equal and opposite damping forces on the two translational conserving ports. The sign of the damping force acting on port R is equal to the sign of the relative linear velocity. A positive linear velocity corresponds to a positive damping force acting on port R, and a negative damping force of equal magnitude acting on port C.

Assumptions and Limitations

- The block represents strictly viscous damping.
- The value of the damping coefficient must be greater than or equal to zero.

Ports

B

Physical signal input port representing the variable translational damping coefficient.

C

Translational conserving port.

R

Translational conserving port.

Parameters

Minimum damping coefficient

Minimum value allowed for the variable damping coefficient. The physical signal input saturates below the specified value. The parameter must be greater than or equal to zero. The default value is `0 N/(m/s)`.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Nonlinear Rotational Damper | Nonlinear Translational Damper | Rotational Damper | Translational Damper | Variable Rotational Damper

Introduced in R2013a

Variable Translational Spring

Translational spring with variable spring stiffness
Library: Simscape / Driveline / Couplings & Drives / Springs &
Dampers

Description

The Variable Translational Spring block represents a translational spring with variable stiffness. A physical signal input port provides the variable spring stiffness. The magnitude of the spring force is equal to the product of the physical signal input and the relative linear displacement between the two translational conserving ports. A minimum spring rate prevents nonphysical values.

The translational spring force satisfies the following expression:

$$F = \begin{cases} K \cdot x, & K \geq K_{\min} \\ K_{\min} \cdot x, & K < K_{\min} \end{cases}$$

where:

- F is the force transmitted through the spring between the two translational conserving ports.
- K is the spring rate coefficient.
- K_{min} is the minimum allowed spring rate.
- x is the relative displacement between the two translational conserving ports according to $x = x_{init} + x_R - x_C$.
- x_{init} is the initial spring deformation.
- x_R is the absolute displacement of the translational conserving port **R**.
- x_C is the absolute displacement of the translational conserving port **C**.

The block applies equal and opposite spring forces on the two translational conserving ports. The sign of the spring force acting on port **R** is equal to the sign of the relative linear displacement between the two ports. A positive linear velocity corresponds to a positive spring force acting on port **R**, and a negative spring force of equal magnitude acting on port **C**.

Assumptions and Limitations

The block limits the value of the variable spring stiffness to remain above zero.

Variables

Use the **Variables** tab to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Ports

Input

K — Variable spring coefficient, N/m
physical signal

Physical signal input port associated with variable spring stiffness.

Conserving

C — Base port
mechanical translational

Mechanical translational conserving port associated with the negative spring end. The positive direction is from port **R** to port **C**.

R — Follower port
mechanical translational

Mechanical translational conserving port associated with the positive spring end. The positive direction is from port **R** to port **C**.

Parameters

`Minimum spring rate` — Spring rate limit value
1 N/m (default) | positive scalar

Minimum value allowed for the spring rate. The physical signal input saturates below the specified value.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Nonlinear Rotational Spring | Nonlinear Translational Spring | Variable Rotational Spring

Introduced in R2013a

Vehicle Body

Two-axle vehicle with longitudinal dynamics and motion and adjustable mass, geometry, and drag properties

Library: Simscape / Driveline / Tires & Vehicles
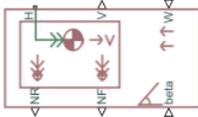
Description

The Vehicle Body block represents a two-axle vehicle body in longitudinal motion. The vehicle can have the same or a different number of wheels on each axle. For example, two wheels on the front axle and one wheel on the rear axle. The vehicle wheels are assumed identical in size. The vehicle can also have a center of gravity (CG) that is at or below the plane of travel.

The block accounts for body mass, aerodynamic drag, road incline, and weight distribution between axles due to acceleration and road profile. Optionally include pitch and suspension dynamics. The vehicle does not move vertically relative to the ground.

The block has an option to include an externally-defined mass and an externally-defined inertia. The mass, inertia, and center of gravity of the vehicle body can vary over the course of simulation in response to system changes.

Model

The vehicle axles are parallel and form a plane. The longitudinal, x, direction lies in this plane and perpendicular to the axles. If the vehicle is traveling on an incline slope, β, the normal, z, direction is not parallel to gravity but is always perpendicular to the axle-longitudinal plane.

This figure and table define the vehicle motion model variables.
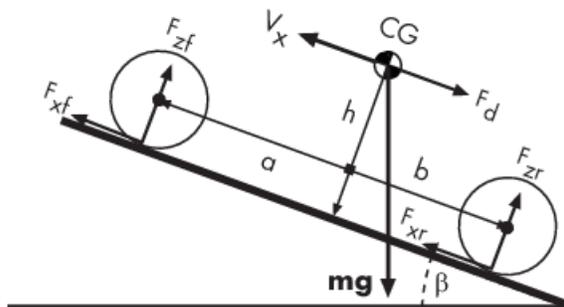
Vehicle Dynamics and Motion

Vehicle Model Variables

Symbol	Description
g	Gravitational acceleration
β	Incline angle
m	Mass of the vehicle
h	Height of vehicle center of gravity (CG) above the ground
a, b	Distance of front and rear axles, respectively, from the normal projection point of vehicle CG onto the common axle plane
V_x	Velocity of the vehicle. When $V_x > 0$, the vehicle moves forward. When $V_x < 0$, the vehicle moves backward.
V_w	Wind speed. When $V_w > 0$, the wind is headwind. When $V_w < 0$, the wind is tailwind.
n	Number of wheels on each axle
F_{xf}, F_{xr}	Longitudinal forces on each wheel at the front and rear ground contact points, respectively
F_{zf}, F_{zr}	Normal load forces on each wheel at the front and rear ground contact points, respectively
A	Effective frontal vehicle cross-sectional area
C_d	Aerodynamic drag coefficient
ρ	Mass density of air
F_d	Aerodynamic drag force

Equations

Vehicle Dynamics

The vehicle motion is a result of the net effect of all the forces and torques acting on it. The longitudinal tire forces push the vehicle forward or backward. The weight mg of the vehicle acts through its center of gravity (CG). Depending on the incline angle, the weight pulls the vehicle to the ground and pulls it either backward or forward. Whether the vehicle travels forward or backward, aerodynamic drag slows it down. For simplicity, the drag is assumed to act through the CG.

$$m\dot{V}_x = F_x - F_d - mg \cdot \sin\beta$$

$$F_x = n(F_{xf} + F_{xr})$$

$$F_d = \frac{1}{2}C_d\rho A(V_x + V_w)^2 \cdot \mathrm{sgn}(V_x + V_w)$$

Zero normal acceleration and zero pitch torque determine the normal force on each front and rear wheel.

$$F_{zf} = \frac{-h(F_d + mg\sin\beta + m\dot{V}_x) + b \cdot mg\cos\beta}{n(a + b)}$$

$$F_{zr} = \frac{+h(F_d + mg\sin\beta + m\dot{V}_x) + a \cdot mg\cos\beta}{n(a+b)}$$

The wheel normal forces satisfy $F_{zf} + F_{zr} = mg\frac{\cos\beta}{n}$.

If you include an externally-defined mass or inertia, the equations are shifted by the weighted value of the input.

Pitch dynamics

Pitch acceleration depends on three torque components and the inertia of the vehicle:

$$\alpha = \frac{(f \cdot h) + (F_{zf}a) - (F_{zr}b)}{J}$$

Where:

- α is the pitch acceleration.
- f is the longitudinal force.
- h is the height of the center of gravity when measured parallel to the z-axis.
- J is the inertia.

If you choose a linear model for suspension stiffness and damping, the block uses small angle approximation for pitch calculations. If you choose a table lookup model, the block uses the vectors that you specify calculate pitch dynamics. For hard stop equations, see Translational Hard Stop.

Variables

Use the **Variables** tab to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Limitations and Assumptions

The Vehicle Body block lets you model only longitudinal dynamics, parallel to the ground and oriented along the direction of motion. The vehicle is assumed to be in pitch and normal equilibrium. The block does not model pitch or vertical movement. As such, the equations assume that the wheels never lose contact. This constraint can result in negative normal forces.

Ports

Input

W — Headwind speed, m/s
scalar

Physical signal input port for headwind speed.

beta — Road incline angle, rad
scalar

Physical signal input port for road incline angle.

CG — Center of gravity, m
two-element vector

Physical signal input port for the center of gravity, in m, of the externally-defined mass relative to the CG of the vehicle body.

Dependencies

This port is visible only when the **Externally-defined additional mass** parameter in the **Main** settings is set to On.

M — Mass, kg
scalar

Physical signal input port for the mass, in kg, of the externally-defined mass.

Dependencies

This port is visible only when the **Externally-defined additional mass** parameter in the **Main** settings is set to On.

J — External moment of inertia, kg*m^2
scalar

Physical signal input port for the moment of inertia, in kg*m^2, of the externally-defined mass.

Dependencies

This port is visible only when the **Externally-defined additional mass** parameter in the **Main** settings and the **Pitch dynamics** parameter in the **Pitch** settings are both set to On.

Output

V — Longitudinal velocity, m/s
scalar

Physical signal output port for vehicle longitudinal velocity.

NF — Front axle normal force, N
scalar

Physical signal output port for normal force on the front axle. Wheel forces are considered positive if acting downwards.

NR — Rear axle normal force, N
scalar

Physical signal output port for normal force on the rear axle. Wheel forces are considered positive if acting downwards.

Conserving

H — Horizontal motion
mechanical translational

Conserving port associated with the horizontal motion of the vehicle body. Connect tire traction motion to this port.

Parameters

Main

Mass — Vehicle mass
1200 kg (default)

Mass of the vehicle.

Number of wheels per axle — Wheel count
2 (default) | scalar number or a two-element array

Wheel counts on the front and rear axles. If the input is a scalar number, the wheel counts of the front and rear axles are assumed the same. For example, if the input is 2, then the front and rear axles are each assumed to have two wheels.

If the input is a two-element array, the first number is the front-axle wheel count and the second number the rear-axle wheel count. For example, if the input is the array [2,1], then the front axle is assumed to have two wheels and the rear axle one wheel.

Horizontal distance from CG to front axle — CG-to-front axle distance
1.4 m (default)

Horizontal distance, *a*, from the center of gravity to the front wheel axle of the vehicle.

Horizontal distance from CG to rear axle — CG-to-rear axle distance
1.6 m (default)

Horizontal distance, *b*, from the center of gravity to the rear wheel axle of the vehicle.

CG height above ground — CG-to-ground distance
0.5 m (default)

Distance, *h*, between the center of gravity of the vehicle and the ground.

Gravitational acceleration — Gravitational acceleration
9.81 m/s^2 (default)

Acceleration due to gravitational force acting at the center of gravity of the vehicle.

Externally-defined additional mass — Variable mass option
Off (default) | On

Option to include as time- or event-variant mass that affects the mass and CG of the vehicle body.

Dependencies

The **CG**, **M**, and **J** ports are visible only when this parameter is set to On. Exposure of the **J** port also requires that you select On for the **Pitch dynamics** parameter in the **Pitch** settings.

Negative normal force warning — Warning
Off (default) | On

To see a warning when the block calculates negative values for normal force during simulation, select On. The block can generate negative forces because the equations assume that the wheels never lose contact.

Drag

`Frontal area` — Effective cross-sectional area
`3 m^2` (default)

Effective cross-sectional area, *A*, presented by the vehicle in longitudinal motion. The block uses this value to calculate the aerodynamic drag force on the vehicle.

`Drag coefficient` — Aerodynamic drag coefficient
`0.4` (default)

Aerodynamic drag coefficient, C_d. The block uses this value to calculate the aerodynamic drag force on the vehicle.

`Air density` — Ambient air density
`1.18 kg/m^3` (default)

Density of the air that surrounds the vehicle.

Pitch

The table shows how the visibility of some parameters depends on the option that you choose for other parameters. To learn how to read the table, see "Parameter Dependencies" on page B-2.

Pitch Parameter Dependencies

Pitch	
Pitch dynamics — Choose `Off` or `On`	
Off	**On**
	Pitch moment of inertia
	Suspension model — Choose `Linear` or `By table lookup`

	Linear	**By table lookup**
	Front suspension stiffness	Front suspension stiffness force vector
	Front suspension damping	Front suspension deformation vector
	Rear suspension stiffness	Front suspension damping force vector
	Rear suspension damping	Front suspension velocity vector
		Rear suspension stiffness force vector
		Rear suspension deformation vector
		Rear suspension damping force vector
		Rear suspension velocity vector
		Interpolation method
		Extrapolation method

	Hard stop — Choose `Off` or `On`	
	Off	**On**
		Front upper bound
		Front lower bound
		Front contact stiffness
		Front contact damping
		Rear upper bound
		Rear lower bound
		Rear contact stiffness
		Rear contact damping

	Initial pitch
	Initial pitch rate

Pitch dynamics — Pitch dynamics modelling
`Off` (default) | `On`

Option to model pitch dynamics.

Dependencies

Selecting `On` for this parameter enables parameters for:

- Suspension model
- Hard-stop model

- Initial pitch conditions

The **J** port is visible only when this parameter is set to On. Exposure of the **J** port also requires that you select On for the **Externally-defined additional mass** parameter in the **Main** settings.

For more information, see the Pitch Parameter Dependencies table.

Pitch moment of inertia — Pitch moment of inertia
4000 kg/m^2 (default) | positive scalar

Moment of inertia for pitch calculations.

Dependencies

This parameter is enabled when you set **Pitch dynamics** to On.

For more information, see the Pitch Parameter Dependencies table.

Suspension model — Suspension parameterization method
Linear (default) | Table lookup

Parameterization method for suspension modeling.

You parameterize the suspension using stiffness and damping data. To specify the data using scalar values, select Linear. To specify the data using vector values, select By table lookup.

Dependencies

This parameter is enabled when you set **Pitch dynamics** to On.

For more information, see the Pitch Parameter Dependencies table.

Front suspension stiffness — Front suspension stiffness
1e4 N/m (default) | positive scalar

Front suspension stiffness per axle.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to On
- **Suspension model** to Linear

For more information, see the Pitch Parameter Dependencies table.

Front suspension damping — Front suspension damping
1e4N/(m/s) (default) | positive scalar

Front suspension damping per axle.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to On
- **Suspension model** to Linear

For more information, see the Pitch Parameter Dependencies table.

Rear suspension stiffness — Rear suspension stiffness
1e4 N/m (default) | positive scalar

Rear suspension stiffness per axle.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to On
- **Suspension model** to Linear

For more information, see the Pitch Parameter Dependencies table.

Rear suspension damping — Rear suspension damping
1e4 N/(m/s) (default) | positive scalar

Rear suspension damping per axle.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to On
- **Suspension model** to Linear

For more information, see the Pitch Parameter Dependencies table.

Front suspension stiffness force vector — Front suspension stiffness force
[-2000, -1000, 0, 1000, 2000] N (default) | increasing vector

Stiffness force of the front suspension. Specify the output values for the lookup table as a vector. The number of elements in the output vector must be the same as the number of elements in the input vector. The input vector parameter is the **Front suspension deformation vector** parameter.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to On
- **Suspension model** to By table lookup

For more information, see the Pitch Parameter Dependencies table.

Front suspension deformation vector — Front suspension deformation
[-.4, -.2, 0, .2, .4] m (default) | increasing vector

Deformation of the front suspension specified in terms of displacement. Specify the input values for the lookup table as a vector. The values of the elements in the vector must increase from left to right. The minimum number of elements in the vector depends on the interpolation method that you select. For linear interpolation, provide at least two elements. For smooth interpolation, provide at least three elements. The output vector parameter is the **Front suspension stiffness force vector** parameter.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to On
- **Suspension model** to By table lookup

For more information, see the Pitch Parameter Dependencies table.

Front suspension damping force vector — Front suspension damping force
`[-200, -100, 0, 100, 200]` N (default) | increasing vector

Damping force of the front suspension. Specify the output values for the lookup table as a vector. The number of elements in the output vector must be the same as the number of elements in the input vector. The input vector parameter is the **Front suspension velocity vector** parameter.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to On
- **Suspension model** to By table lookup

For more information, see the Pitch Parameter Dependencies table.

Front suspension velocity vector — Front suspension velocity
`[-4, -2, 0, 2, 4]` m/s (default) | increasing vector

Velocity of the front suspension. Specify the input values for the lookup table as a vector. The values of the elements in the vector must increase from left to right. The minimum number of elements in the vector depends on the interpolation method that you select. For linear interpolation, provide at least two elements. For smooth interpolation, provide at least three elements. The output vector parameter is the **Front suspension damping force vector** parameter.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to On
- **Suspension model** to By table lookup

For more information, see the Pitch Parameter Dependencies table.

Rear suspension stiffness force vector — Rear suspension stiffness force
`[-2000, -1000, 0, 1000, 2000]` N (default) | increasing vector

Stiffness force of the rear suspension. Specify the output values for the lookup table as a vector. The number of elements in the output vector must be the same as the number of elements in the input vector. The input vector parameter is the **Rear suspension deformation vector** parameter.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to On

- **Suspension model** to By table lookup

For more information, see the Pitch Parameter Dependencies table.

Rear suspension deformation vector — Rear suspension deformation
[-.4, -.2, 0, .2, .4] m (default) | increasing vector

Deformation of the rear suspension specified in terms of displacement. Specify the input values for the lookup table as a vector. The values of the elements in the vector must increase from left to right. The minimum number of elements in the vector depends on the interpolation method that you select. For linear interpolation, provide at least two elements. For smooth interpolation, provide at least three elements. The output vector parameter is the **Rear suspension stiffness force vector** parameter.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to On
- **Suspension model** to By table lookup

For more information, see the Pitch Parameter Dependencies table.

Rear suspension damping force vector — Rear suspension damping force
[-200, -100, 0, 100, 200] N (default) | increasing vector

Damping force of the rear suspension. Specify the output values for the lookup table as a vector. The number of elements in the output vector must be the same as the number of elements in the input vector. The input vector parameter is the **Rear suspension velocity vector** parameter.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to On
- **Suspension model** to By table lookup

For more information, see the Pitch Parameter Dependencies table.

Rear suspension velocity vector — Rear suspension velocity
[-4, -2, 0, 2, 4] m/s (default) | increasing vector

Velocity of the rear suspension. Specify the input values for the lookup table as a vector. The values of the elements in the vector must increase from left to right. The minimum number of elements in the vector depends on the interpolation method that you select. For linear interpolation, provide at least two elements. For smooth interpolation, provide at least three elements. The output vector parameter is the **Rear suspension damping force vector** parameter.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to On
- **Suspension model** to By table lookup

For more information, see the Pitch Parameter Dependencies table.

`Interpolation method` — **Interpolation method**
`linear` (default) | `smooth`

Interpolation methods for approximating the output value when the input value is between two consecutive grid points are:

- `Linear` — Select this default option to get the best performance. Provide at least two values per dimension.
- `Smooth` — Select this option to produce a continuous curve with continuous first-order derivatives. Provide at least three values per dimension.

For more information on interpolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to `On`
- **Suspension model** to `By table lookup`

For more information, see the Pitch Parameter Dependencies table.

`Extrapolation method` — **Extrapolation method**
`linear` (default) | `smooth`

Extrapolation method for determining the output value when the input value is outside the range specified in the argument list are:

- `Linear` — Produces a curve with continuous first-order derivatives in the extrapolation region and at the boundary with the interpolation region.
- `Nearest` — Produces an extrapolation that does not go above the highest point in the data or below the lowest point in the data.

For more information on extrapolation algorithms, see the PS Lookup Table (1D) block reference page.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to `On`
- **Suspension model** to `By table lookup`

For more information, see the Pitch Parameter Dependencies table.

`Hard stop` — **Hard-stop modeling option**
`Off` (default) | `On`

Option to model hardstops for the front and rear suspension.

Dependencies

This parameter is enabled when you set the **Pitch dynamics** to `On`:

Selecting `On` for this parameter enables parameters for the front and rear hard stops:

- Upper bound
- Lower bound
- Contact stiffness
- Contact damping

For more information, see the Pitch Parameter Dependencies table.

`Front upper bound` — **Hardstop front suspension upper bound**
`.25` (default)

Upper bound of the hard stop for the front suspension.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to `On`
- **Hard stop** to `On`

For more information, see the Pitch Parameter Dependencies table.

`Front lower bound` — **Vehicle front lower bound**
`-.25` (default)

Lower bound of the hard stop for the front suspension.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to `On`
- **Hard stop** to `On`

For more information, see the Pitch Parameter Dependencies table.

`Front contact stiffness` — **Vehicle front contact stiffness**
`1e6` (default)

Contact stiffness of the hard stop for the front suspension.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to `On`
- **Hard stop** to `On`

For more information, see the Pitch Parameter Dependencies table.

`Front contact damping` — **Vehicle front contact damping**
`150` (default)

Contact damping of the hard stop for the front suspension.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to On
- **Hard stop** to On

For more information, see the Pitch Parameter Dependencies table.

Rear upper bound — Vehicle rear upper bound
`.25` (default)

Upper bound of the hard stop for the rear suspension.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to On
- **Hard stop** to On

For more information, see the Pitch Parameter Dependencies table.

Rear lower bound — Vehicle rear lower bound
`-.25` (default)

Lower bound of the hard stop for the rear suspension.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to On
- **Hard stop** to On

For more information, see the Pitch Parameter Dependencies table.

Rear contact stiffness — Vehicle rear contact stiffness
`1e6` (default)

Contact stiffness of the hard stop for the rear suspension.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to On
- **Hard stop** to On

For more information, see the Pitch Parameter Dependencies table.

Rear contact damping — Vehicle rear contact damping
`150` (default)

Contact damping of the hard stop for the rear suspension.

Dependencies

This parameter is enabled when you set both of these options:

- **Pitch dynamics** to On
- **Hard stop** to On

For more information, see the Pitch Parameter Dependencies table.

`Initial pitch` — Pitch at the beginning of the simulation
`0 rad` (default)

Pitch at the beginning of the simulation.

Dependencies

This parameter is enabled when you set **Pitch dynamics** to On.

For more information, see the Pitch Parameter Dependencies table.

`Initial pitch rate` — Pitch rate at the beginning of the simulation
`0 rad/s` (default)

Pitch rate at the beginning of the simulation.

Dependencies

This parameter is enabled when you set **Pitch dynamics** to On.

For more information, see the Pitch Parameter Dependencies table.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Tire (Friction Parameterized) | Tire-Road Interaction (Magic Formula) | Tire (Magic Formula) | Tire (Simple) | Translational Hard Stop

Topics
"Complete Vehicle Model"

Introduced in R2011a

Worm Gear

Worm gear with adjustable gear ratio and friction losses
Library: Simscape / Driveline / Gears

Description

The block represents a rotational gear that constrains the two connected driveline axes, worm (W) and gear (G), to rotate together in a fixed ratio that you specify. You can choose whether the gear rotates in a positive or negative direction. Right-hand rotation is the positive direction. If the worm thread is right-hand, ω_W and ω_G have the same sign. If the worm thread is left-hand, ω_W and ω_G have opposite signs.

Thermal Model

You can model the effects of heat flow and temperature change by enabling the optional thermal port. To enable the port, set **Friction model** to `Temperature-dependent efficiency`.

Model Variables

R_{WG}	Gear ratio
ω_W	Worm angular velocity
ω_G	Gear angular velocity
α	Normal pressure angle
λ	Worm lead angle
L	Worm lead
d	Worm pitch diameter
τ_G	Gear torque
τ_W	Torque on the worm
τ_{loss}	Torque loss due to meshing friction. The loss depends on the device efficiency and the power flow direction. To avoid abrupt change of the friction torque at $\omega_G = 0$, the friction torque is introduced via the hyperbolic function.
τ_{fr}	Steady-state value of the friction torque at $\omega_G \to \infty$.
k	Friction coefficient
η_{WG}	Torque transfer efficiency from worm to gear
η_{GW}	Torque transfer efficiency from gear to worm
p_{th}	Power threshold
$[\mu_W \ \mu_G]$	Vector of viscous friction coefficients for the worm and gear

Ideal Gear Constraint and Gear Ratio

Worm gear imposes one kinematic constraint on the two connected axes:

$$\omega_W = R_{WG}\omega_G \ . \tag{1-11}$$

The two degrees of freedom are reduced to one independent degree of freedom. The forward-transfer gear pair convention is (1,2) = (W,G).

The torque transfer is:

$$R_{WG}\tau_W - \tau_G - \tau_{loss} = 0 \ , \tag{1-12}$$

with $\tau_{loss} = 0$ in the ideal case.

Nonideal Gear Constraint

In the nonideal case, $\tau_{loss} \neq 0$. For general considerations on nonideal gear modeling, see "Model Gears with Losses".

Geometric Surface Contact Friction

In the contact friction case, η_{WG} and η_{GW} are determined by:

- The worm-gear threading geometry, specified by lead angle λ and normal pressure angle α.
- The surface contact friction coefficient k.

$$\eta_{WG} = (\cos\alpha - k \cdot \tan\lambda)/(\cos\alpha + k/\tan\lambda) \ , \tag{1-13}$$

$$\eta_{GW} = (\cos\alpha - k/\tan\lambda)/(\cos\alpha + k \cdot \tan\lambda) \ . \tag{1-14}$$

Constant Efficiencies

In the constant friction case, you specify η_{WG} and η_{GW}, independently of geometric details.

Self-Locking and Negative Efficiency

η_{GW} has two distinct regimes, depending on lead angle λ, separated by the *self-locking point* at which $\eta_{GW} = 0$ and $\cos\alpha = k/\tan\lambda$.

- In the *overhauling regime*, $\eta_{GW} > 0$, and the force acting on the nut can rotate the screw.
- In the *self-locking regime*, $\eta_{GW} < 0$, and an external torque must be applied to the screw to release an otherwise locked mechanism. The more negative is η_{GW}, the larger the torque must be to release the mechanism.

η_{WG} is conventionally positive.

Meshing Efficiency

The efficiencies η of meshing between worm and gear are fully active only if the transmitted power is greater than the power threshold.

If the power is less than the threshold, the actual efficiency is automatically regularized to unity at zero velocity.

You can set the meshing losses friction model to:

- `No meshing losses - suitable for HIL simulation`.
- `Constant efficiency`, which is the default friction setting for block versions prior to R2020b.
- `Temperature-dependent efficiency`, which models variability in the base-shaft efficiencies calculated in the `Constant efficiency` setting according to a user-supplied look-up table. The temperature-dependency setting enables a thermal conserving port **H**. This port receives the heat flow into the block, which is translated into the block temperature according to the gear **Thermal mass**.

Viscous Friction Force

The viscous friction coefficient μ_W controls the viscous friction torque experienced by the worm from lubricated, nonideal gear threads and viscous bearing losses. The viscous friction torque on a worm driveline axis is $-\mu_W\omega_W$. ω_W is the angular velocity of the worm with respect to its mounting.

The viscous friction coefficient μ_G controls the viscous friction torque experienced by the gear, mainly from viscous bearing losses. The viscous friction torque on a gear driveline axis is $-\mu_G\omega_G$. ω_G is the angular velocity of the gear with respect to its mounting.

Hardware-in-the-Loop Simulation

For optimal performance of your real-time simulation, set the **Friction model** to `No meshing losses - Suitable for HIL simulation` on the **Meshing Losses** tab.

Variables

Use the **Variables** settings to set the priority and initial target values for the block variables before simulating. For more information, see "Set Priority and Initial Target for Block Variables".

Limitations

- Gear inertia is assumed to be negligible.
- Gears are treated as rigid components.
- Coulomb friction slows down simulation. For more information, see "Adjust Model Fidelity".

Ports

Conserving

W — Worm
mechanical rotational

Rotational conserving port representing the worm torque and angular velocity.

G — Gear
mechanical rotational

Rotational conserving port representing the gear torque and angular velocity.

H — Optional heat transfer port
mechanical translational

Thermal conserving port for thermal modeling.

Dependencies

To enable this port, set **Friction model** to either:

- `Temperature-dependent efficiency`
- `Temperature and load-dependent efficiency`

.

Parameters

Main

Gear ratio — Ratio of angular velocities
25 (default) | positive scalar greater than 1

Gear or transmission ratio R_{WG} determined as the ratio of the worm angular velocity to the gear angular velocity.

Worm thread type — Rotational orientation
Right-hand (default) | Left-hand

Choose the directional sense of gear rotation corresponding to positive worm rotation. If you select `Left-hand`, rotation of the worm in the generally-assigned positive direction results in the gear rotation in negative direction.

Meshing Losses

Friction model — Gear friction model
No meshing losses — Suitable for HIL simulation (default) | Constant efficiency | Temperature-dependent efficiency

- `No meshing losses — Suitable for HIL simulation` — Gear meshing is ideal.
- `Constant efficiency` — Transfer of torque between worm and gear is reduced by friction.
- `Temperature-dependent efficiency` — Torque transfer is determined from user-supplied data for worm-gear efficiency, gear-worm efficiency, and temperature.

Friction parametrization — Friction losses from nonideal meshing of gear threads
Friction coefficient and geometrical parameters (default) | Efficiencies

- `Friction coefficient and geometrical parameters` — Friction is determined by contact friction between surfaces.
- `Efficiencies` — Friction is determined by constant efficiencies $0 < \eta < 1$.

Dependencies

To enable this parameter, set **Friction model** to `Constant efficiency`.

Normal pressure angle — Thread pressure angle
17.5 deg (default) | positive scalar

Thread pressure angle α in the normal plane. The value must be greater than zero and less than 90 degrees.

Dependencies

To enable this parameter, set **Friction model** to Constant efficiency and **Friction parameterization** to Friction coefficient and geometrical parameters.

Lead angle — Thread helix angle
20 deg (default) | positive scalar

Thread helix angle $\lambda = \arctan[L/(\pi d)]$, where:

- L is the worm lead.
- d is the worm pitch diameter.

The value must be greater than zero.

Dependencies

To enable this parameter, set **Friction model** to Constant efficiency and **Friction parameterization** to Friction coefficient and geometrical parameters.

Friction coefficient — Thread friction coefficient
0.08 (default) | positive scalar

Dimensionless coefficient of normal friction in the thread. The value must be greater than zero.

Dependencies

To enable this parameter, set **Friction model** to Constant efficiency and **Friction parameterization** to Friction coefficient and geometrical parameters.

Worm-gear efficiency — Efficiency from worm to gear
0.74 (default) | positive scalar in the range of (0,1]

Efficiency η_{WG} of the power transfer from worm to gear.

Dependencies

To enable this parameter, set **Friction model** to Constant efficiency and **Friction parameterization** to Efficiencies.

Gear-worm efficiency — Efficiency from gear to worm
0.65 (default) | positive scalar in the range of (0,1]

Efficiency η_{GW} of the power transfer from gear to worm.

Dependencies

To enable this parameter, set **Friction model** to Constant efficiency and **Friction parameterization** to Efficiencies.

Temperature — Array for tabular parameterization of efficiency
[280 300 320] K (default) | vector

Array of temperatures used to construct a 1-D temperature-efficiency lookup table. The array values must increase left to right. The temperature array must be the same size as the **Worm-gear efficiency** and **Gear-worm efficiency** arrays.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

`Worm-gear efficiency` — **Array of worm-to-gear efficiencies**
`[0.75 0.7 0.65]` (default) | vector

Array of component efficiencies with the worm as the driver— that is, with power flowing from the worm to the gear. The array values are the efficiencies at the temperatures in the **Temperature** array. The two arrays must be the same size.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

`Gear-worm efficiency` — **Array of gear-to-worm efficiencies**
`[0.5 0.45 0.4]` (default) | vector

Array of component efficiencies with the gear as the driver— that is, with power flowing from the gear to the worm. The array values are the efficiencies at the temperatures in the **Temperature** array. The two arrays must be the same size.

Dependencies

To enable this parameter, set **Friction model** to `Temperature-dependent efficiency`.

`Power threshold` — **Power below which numerical smoothing is applied**
`0.001` W (default) | positive scalar

Power threshold above which full efficiency factor is in effect. A hyperbolic tangent function smooths the efficiency factor between zero at rest and the current efficiency set point.

Viscous Losses

`Viscous friction coefficients at worm (W) and gear (G)` — **Fluid dynamic friction coefficient**
`[0 0]` N*m/(rad/s) (default) | positive two-element vector

Vector of viscous friction coefficients $[\mu_W \ \mu_G]$, for the worm and gear, respectively.

Thermal Port

`Thermal mass` — **Thermal mass**
`50` J/K (default) | positive scalar

Thermal energy required to change the component temperature by a single degree. The greater the thermal mass, the more resistant the component is to temperature change.

`Initial temperature` — **Initial temperature**
`300` K (default) | positive scalar

Component temperature at the start of simulation. The initial temperature alters the component efficiency according to an efficiency vector that you specify, affecting the starting meshing or friction losses.

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also

Simscape Blocks
Leadscrew | Sun-Planet | Sun-Planet Worm Gear

Topics
"Basic Motion, Torque, and Force Modeling"
"Model Thermal Losses in Driveline Components"
"Prepare Simscape Driveline Models for Real-Time Simulation Using Simscape Checks"

Introduced in R2011a

Technical Conventions

Driveline Abbreviations and Conventions

In this section...
"Abbreviations" on page A-2
"Angular Motion" on page A-2
"Gear Ratios" on page A-2

Abbreviations

An important abbreviation is DoF, which means *degree of freedom* and refers to one coordinate of angular motion. All driveline rotational DoFs are measured with respect to a single absolute coordinate system at rest.

Angular Motion

Standard symbols for angular motion analysis include the following:

Symbol	Meaning (Units)
r	Gear radius
N	Number of gear teeth
g	Gear ratio
θ	Angle
ω	Angular velocity
τ	Torque

Gear Ratios

For a pair of coupled, coplanar gear wheels, the gear ratio g_{21} of gear 2 to gear 1 is defined as the ratio of the second gear wheel radius to the first. This definition is equivalent to the ratio of the number of teeth on the second gear wheel to the number of teeth on the first.

$$g_{21} \equiv r_2/r_1 = N_2/N_1$$

The gear ratio is the ratio of torques and the reciprocal of the angular velocity ratio.

$$g_{21} = \tau_2/\tau_1 = \omega_1/\omega_2$$

For gear boxes made of more than two gear wheels, the gear ratio is defined to be the ratio of torques or the reciprocal of the ratio of angular velocities, between the output and input shafts.

If the gear is reversing, the output ω and τ have opposite signs from the input ω and τ.

Parameter Dependencies

Parameter Dependencies

A Simscape block parameter is considered *visible* when it appears in the Simulink Property Inspector, in a block dialog box, or as a block choice on the Simscape context menu. A block parameter is considered *configurable* or *enabled* when you can configure it by selecting an option, entering a value, or selecting or clearing a check box. Parameters that are visible but are not enabled are typically dimmed.

The visibility and configurability of some parameters depend on the options that you select for other parameters. Parameter dependencies are typically listed in the parameter description in the documentation for the block. For some blocks, the documentation also includes a parameter dependency table.

Parameter Dependency Tables

Parameter dependency tables show how the visibility and configurability of some parameters depend on the options that you select for other parameters.

Parameter dependency tables indicate which parameter options affect the *visibility* of other parameters. For example, the next table shows the dependencies for the **Effects and Initial Conditions** parameters for the Pipe (2P) block from the **Simscape > Foundation Library > Two-Phase Fluid > Elements** library.

The row numbers in this table are for reference only. The first column of a parameter dependency table typically includes all parameters that are visible by default.

Parameters that affect the visibility or configurability of other parameters are shown in bold text. The options for the parameters that affect other parameters are shown in italic text in the following row of the table. Parameters that do not affect other parameters are shown in plain text.

Effects and Initial Conditions Parameter Dependencies

Row	Parameters and Options					
1	**Fluid Inertia**					
2	*Off*			*On*		
3	**Initial fluid regime**			**Initial fluid regime**		
4	*Subcooled liquid*	*Two-phase mixture*	*Superheated vapor*	*Subcooled liquid*	*Two-phase mixture*	*Superheated vapor*
5	Initial fluid pressure	Initial fluid pressure	Initial fluid pressure	Initial fluid pressure	Initial fluid pressure	Initial fluid pressure
6	Initial fluid temperature	Initial fluid vapor quality	Initial fluid temperature	Initial fluid temperature	Initial fluid vapor quality	Initial fluid temperature
7				Initial mass flow rate from port A to port B	Initial mass flow rate from port A to port B	Initial mass flow rate from port A to port B
8	Phase change time constant	Phase change time constant	Phase change time constant	Phase change time constant	Phase change time constant	Phase change time constant

The figure shows the **Effects and Initial Conditions** parameters that are visible by default on the block dialog box for the Pipe (2P) block.


If you change the **Fluid Inertia** parameter from `Off` to `On`, the **Initial mass flow rate from port A to port B** parameter becomes visible. The table shows this dependency in rows 1, 2, and 7.

If you then change the **Initial fluid regime** parameter from `Subcooled liquid` to `Two-phase mixture`, the **Initial fluid temperature** parameter is hidden and **Initial fluid vapor quality** becomes visible. The table shows these dependencies in rows 3, 4, and 6.
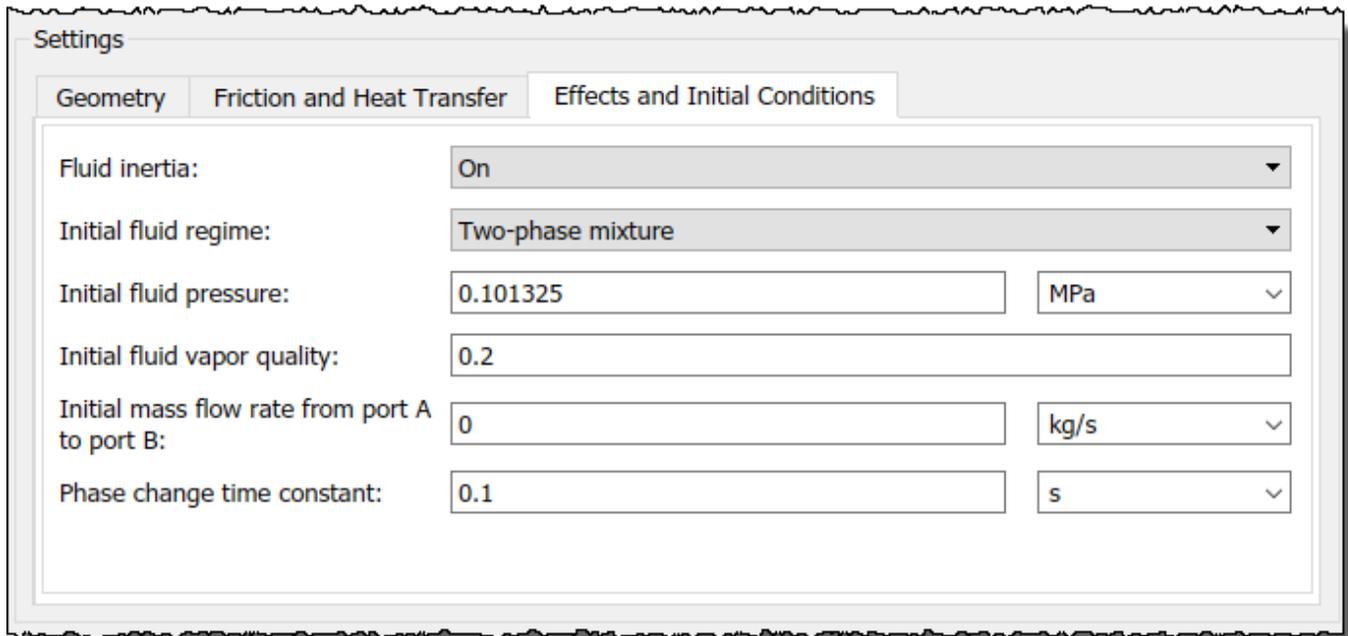
Parameter dependency tables also indicate which parameter options *enable* other parameters. For example, the next table shows the dependencies for the parameters for the Solver Configuration block from the **Simscape > Foundation Library > Utilities** library. The first column in the table contains row numbers, which are for reference only. The second column includes all parameters that are visible by default.

Parameters that affect the visibility and configurability of other parameters are shown in bold text. The options for the parameters that affect other parameters are shown as selected and cleared check boxes in the following row of the table. Parameters that are shown in plain text are enabled but do not affect other parameters. Parameters that are not enabled are not shown.

Solver Configuration Block Parameter Dependencies

Row	Parameters and Options			
1	Start simulation from steady state			
2	Consistency tolerance			
3	**Use local Solver**			
4	☐		☑	
5			Solver type	
6			Sample time	
7	**Use fixed-cost runtime consistency iterations**		**Use fixed-cost runtime consistency iterations**	
8	☐	☑	☑	☐
9		Nonlinear iterations	Nonlinear iterations	
10		Mode iterations		
11	Linear Algebra			
12	Equation formulation			
13	Delay memory budget [kB]			
14	**Apply filtering at 1-D/3-D connections when needed**			
15	☑		☐	
16	Filtering time constant			

The figure shows the parameters that are visible by default on the block dialog box for the Solver Configuration block. The dimmed parameters, for example, **Solver type**, are not enabled.
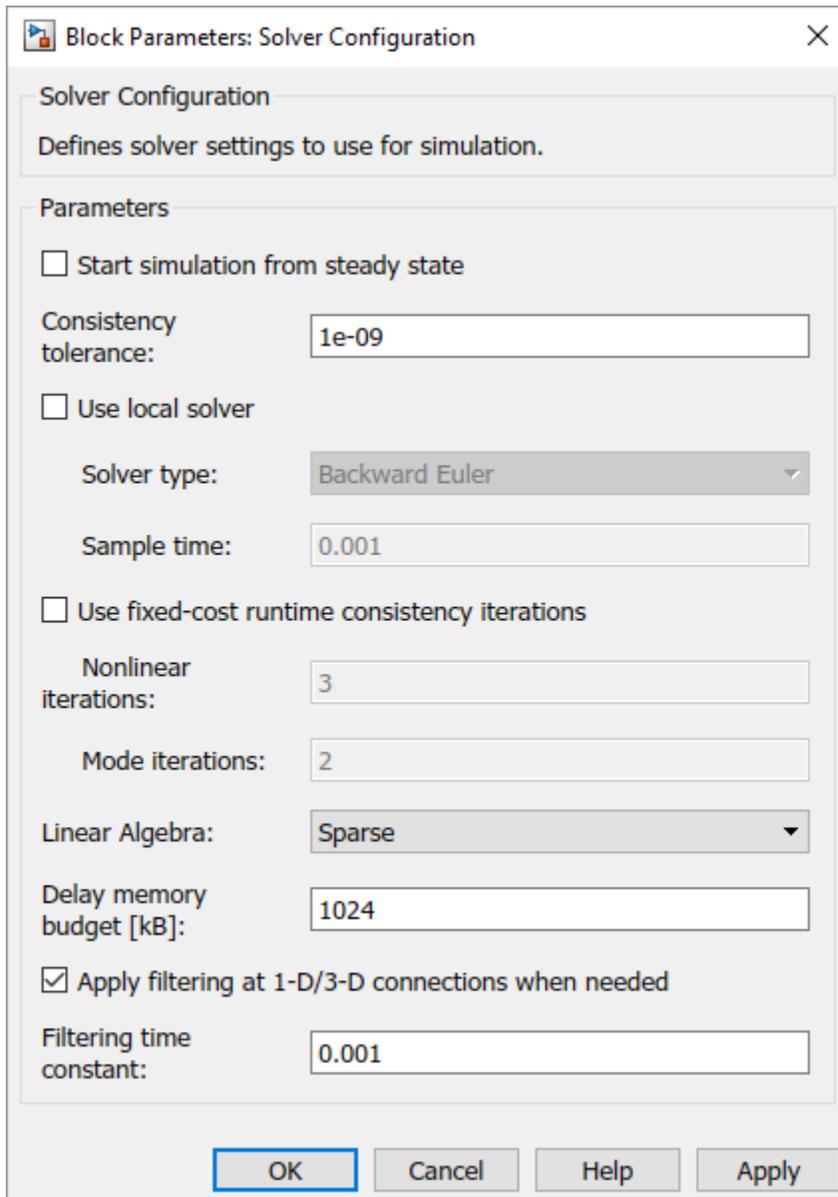
If you select the **Use local solver** check box, the **Use fixed-cost runtime consistency** check box becomes selected and these parameters become enabled.

- **Solver type**
- **Nonlinear iterations**
- **Sample time**

Selecting **Use local solver** does not enable the **Mode iterations** parameter. The **Mode iterations** parameter is only enabled when the **Use local solver** check box is cleared while **Use fixed-cost runtime consistency** check box is selected.

The table shows these dependencies in rows 3–10.
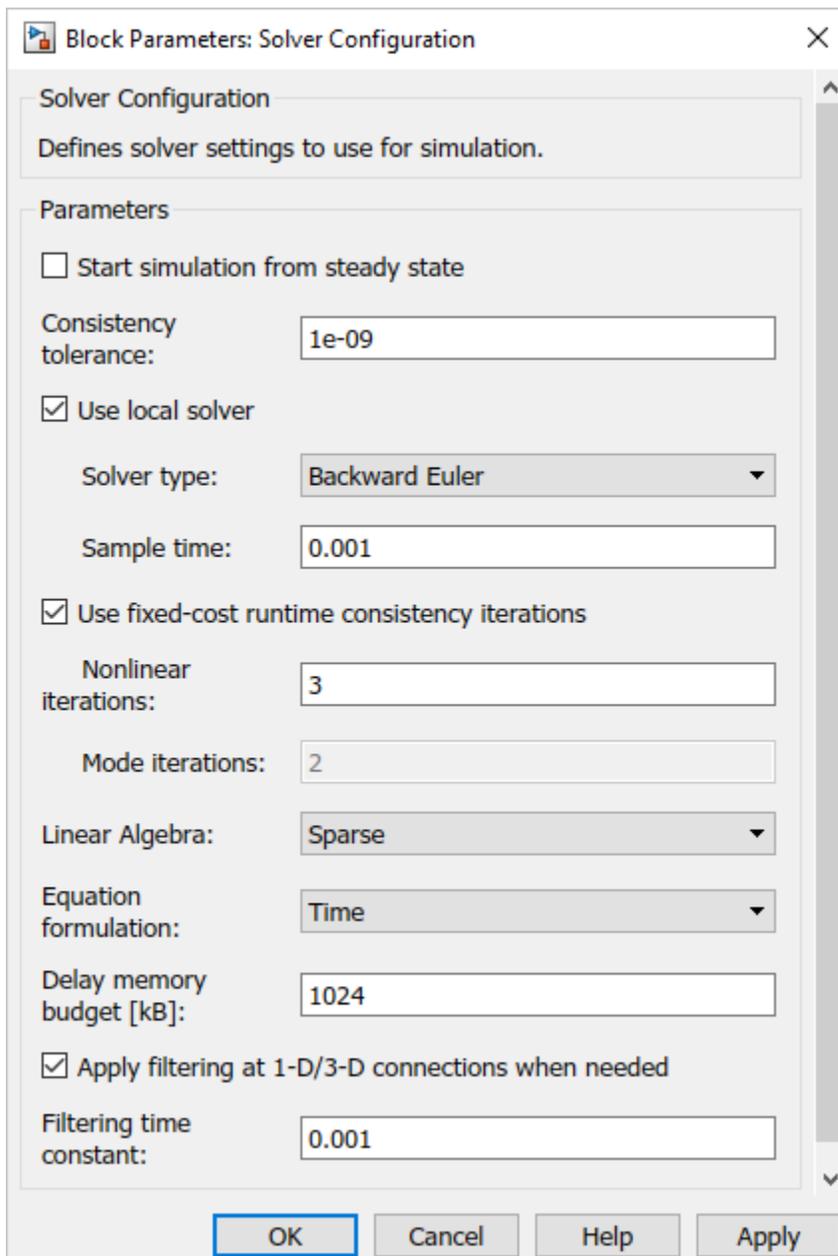
Block Parameters: Solver Configuration ✕

Solver Configuration

Defines solver settings to use for simulation.

Parameters

☐ Start simulation from steady state

Consistency
tolerance: `1e-09`

☑ Use local solver

    Solver type: Backward Euler ▼

    Sample time: `0.001`

☑ Use fixed-cost runtime consistency iterations

    Nonlinear
    iterations: `3`

    Mode iterations: `2`

Linear Algebra: Sparse ▼

Equation
formulation: Time ▼

Delay memory
budget [kB]:   `1024`

☑ Apply filtering at 1-D/3-D connections when needed

Filtering time
constant: `0.001`

[OK]   [Cancel]   [Help]   [Apply]

Bibliography

[1] Genta, G., *Motor Vehicle Dynamics: Modeling and Simulation,* Singapore, World Scientific, 1997, chapters 2, 3, and 4

[2] Jurgen, R. K., *Electronic Transmission Controls,* Troy, Michigan, Society of Automotive Engineers, 2000

[3] Juvinall, R. C., *Fundamentals of Machine Component Design,* New York, John Wiley & Sons, 1983

[4] The MathWorks Inc., https://www.mathworks.com/industries/auto/, Industries: Automotive

[5] Meriam, J. L., and L. G. Kraige, *Dynamics*, Volume 2 of *Engineering Mechanics,* New York, John Wiley & Sons, 1987

[6] Nwagboso, C. O., *Automotive Sensory Systems,* London, Chapman & Hall, 1993

[7] Pacejka, H. B. *Tire and Vehicle Dynamics,* Society of Automotive Engineers and Butterworth-Heinemann, Oxford, 2002, chapters 1,4,7, and 8

[8] Shingley, J. E., C. R. Mishke, and R. G. Budynas, *Mechanical Engineering Design,* McGraw-Hill, 2004

[9] Society of Automotive Engineers, https://www.sae.org, see "SAE Store"

[10] Wong, J. Y., *Theory of Ground Vehicles,* 3rd Ed., New York, Interscience, 2001