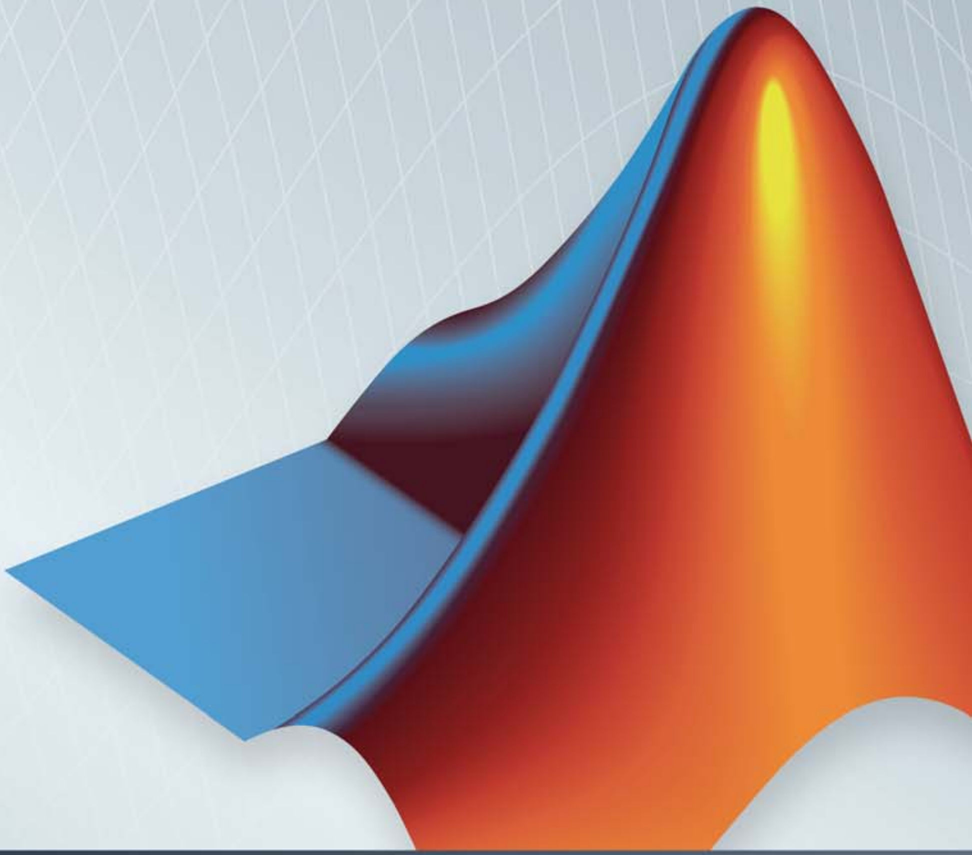
Simscape™

Reference

R2013**b**



MATLAB® & SIMULINK®

MathWorks®

How to Contact MathWorks

`www.mathworks.com`	Web
`comp.soft-sys.matlab`	Newsgroup
`www.mathworks.com/contact_TS.html`	Technical Support
`suggest@mathworks.com`	Product enhancement suggestions
`bugs@mathworks.com`	Bug reports
`doc@mathworks.com`	Documentation error reports
`service@mathworks.com`	Order status, license renewals, passcodes
`info@mathworks.com`	Sales, pricing, and general information

508-647-7000 (Phone)

508-647-7001 (Fax)

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

For contact information about worldwide offices, see the MathWorks Web site.

Simscape™ Reference

Trademarks

Patents

Revision History

March 2007	Online only	New for Version 1.0 (Release 2007a)
September 2007	Online only	Revised for Version 2.0 (Release 2007b)
March 2008	Online only	Revised for Version 2.1 (Release 2008a)
October 2008	Online only	Revised for Version 3.0 (Release 2008b)
March 2009	Online only	Revised for Version 3.1 (Release 2009a)
September 2009	Online only	Revised for Version 3.2 (Release 2009b)
March 2010	Online only	Revised for Version 3.3 (Release 2010a)
September 2010	Online only	Revised for Version 3.4 (Release 2010b)
April 2011	Online only	Revised for Version 3.5 (Release 2011a)
September 2011	Online only	Revised for Version 3.6 (Release 2011b)
March 2012	Online only	Revised for Version 3.7 (Release 2012a)
September 2012	Online only	Revised for Version 3.8 (Release 2012b)
March 2013	Online only	Revised for Version 3.9 (Release 2013a)
September 2013	Online only	Revised for Version 3.10 (Release 2013b)

Contents

1 Blocks — Alphabetical List

2 Functions — Alphabetical List

3 Configuration Parameters

4 Model Advisor Checks

1

Blocks — Alphabetical List

Absolute Reference (TL)

Purpose

Reference connection for thermal liquid ports

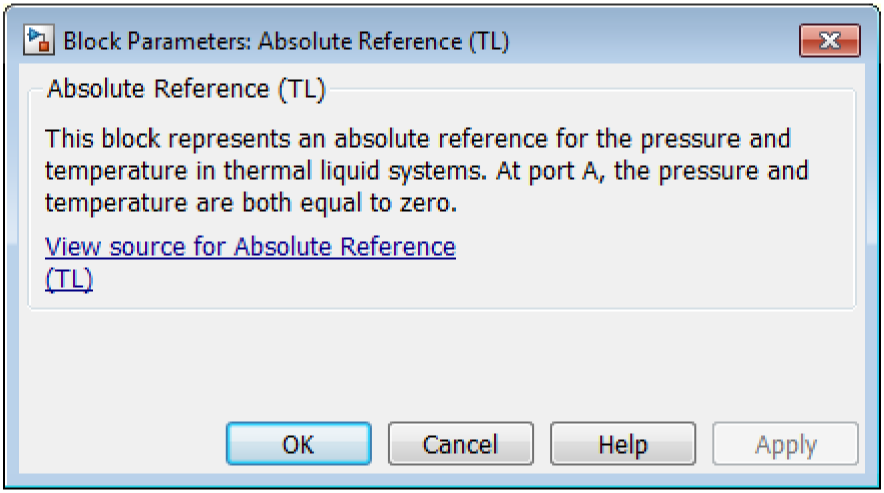
Library

Thermal Liquid/Elements

Description

The Absolute Reference (TL) block represents an absolute reference for the pressure and temperature in thermal liquid systems. At port A, the pressure and temperature are both equal to zero.

Dialog Box and Parameters



Block Parameters: Absolute Reference (TL)

Absolute Reference (TL)

This block represents an absolute reference for the pressure and temperature in thermal liquid systems. At port A, the pressure and temperature are both equal to zero.

View source for Absolute Reference (TL)

The block has no parameters.

Ports

The block has one thermal liquid conserving port.

See Also

Temperature Reservoir (TL)

Controlled Temperature Reservoir (TL)

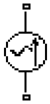
Purpose

Ideal sinusoidal current source

Library

Electrical Sources

Description



The AC Current Source block represents an ideal current source that maintains sinusoidal current through it, independent of the voltage across its terminals.

The output current is defined by the following equation:

$$I = I_0 \cdot \sin(2\pi \cdot f \cdot t + \varphi)$$

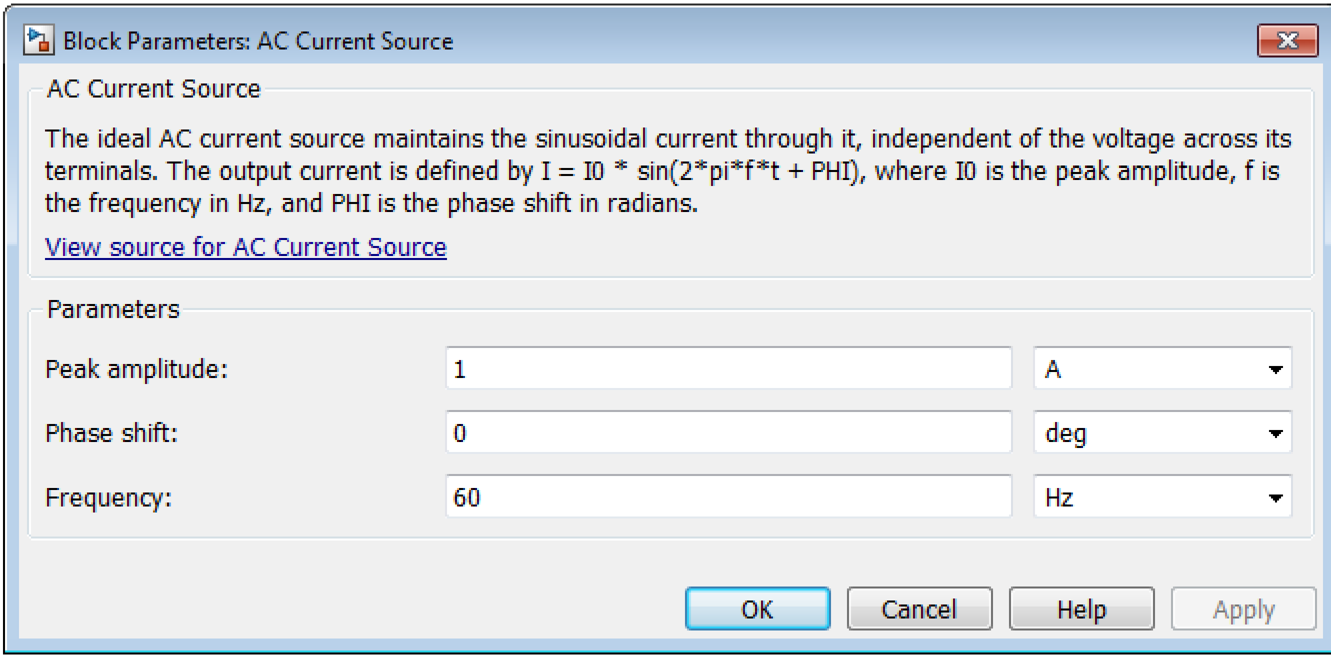
where

I	Current
I_0	Peak amplitude
f	Frequency
φ	Phase shift
t	Time

The positive direction of the current flow is indicated by the arrow.

Note For Release 2012b and earlier, the unit definition for Hz was `rev/s`, whereas in R2013a it was changed to be `1/s`, in compliance with the SI unit system. For this block it means that you must specify frequency in units of `Hz` or directly convertible to Hz, such as `1/s`, `kHz`, `MHz` and `GHz`. In 2012b and earlier you could also specify frequency in angular units (such as `rad/s` or `rpm`), but this is no longer possible because the internal equation of the block now uses the 2π conversion factor to account for the `1/s` unit definition. If you use this block in a model created prior to R2013a, update it by using the `slupdate` utility. For more information, see the R2013a Release Notes.

Dialog Box and Parameters



Block Parameters: AC Current Source

AC Current Source

The ideal AC current source maintains the sinusoidal current through it, independent of the voltage across its terminals. The output current is defined by I = I0 * sin(2*pi*f*t + PHI), where I0 is the peak amplitude, f is the frequency in Hz, and PHI is the phase shift in radians.

View source for AC Current Source

Parameters

Peak amplitude: 1 A

Phase shift: 0 deg

Frequency: 60 Hz

Peak amplitude

Peak current amplitude. The default value is 1 A.

Phase shift

Phase shift in angular units. The default value is 0.

Frequency

Current frequency, specified in Hz or units directly convertible to Hz (where Hz is defined as 1/s). For example, kHz and MHz are valid units, but rad/s is not. The default value is 60 Hz.

Ports

The block has two electrical conserving ports associated with its terminals.

See Also

AC Voltage Source

AC Voltage Source

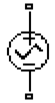
Purpose

Ideal constant voltage source

Library

Electrical Sources

Description



The AC Voltage Source block represents an ideal voltage source that maintains sinusoidal voltage across its output terminals, independent of the current flowing through the source.

The output voltage is defined by the following equation:

$$V = V_0 \cdot \sin(2\pi \cdot f \cdot t + \varphi)$$

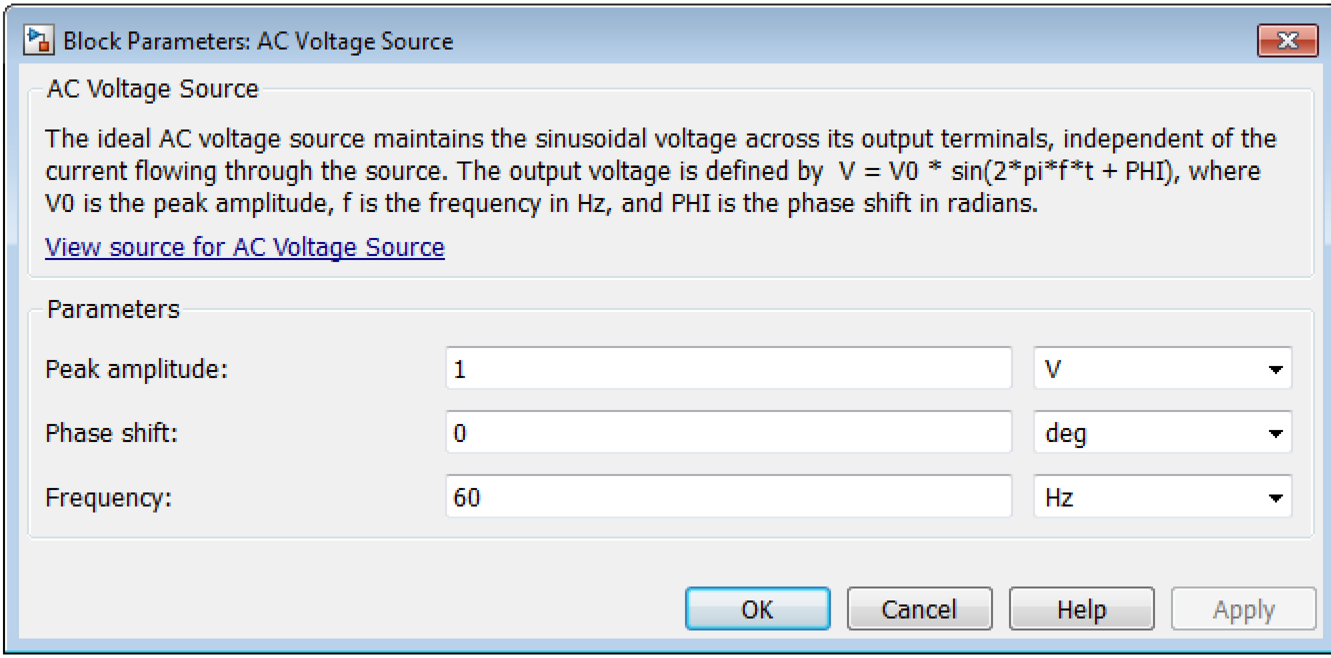
where

V	Voltage
V_0	Peak amplitude
f	Frequency
φ	Phase shift
t	Time

Connections + and – are conserving electrical ports corresponding to the positive and negative terminals of the voltage source, respectively. The current is positive if it flows from positive to negative, and the voltage across the source is equal to the difference between the voltage at the positive and the negative terminal, $V(+) - V(–)$.

Note For Release 2012b and earlier, the unit definition for Hz was rev/s, whereas in R2013a it was changed to be 1/s, in compliance with the SI unit system. For this block it means that you must specify frequency in units of Hz or directly convertible to Hz, such as 1/s, kHz, MHz and GHz. In 2012b and earlier you could also specify frequency in angular units (such as rad/s or rpm), but this is no longer possible because the internal equation of the block now uses the 2π conversion factor to account for the 1/s unit definition. If you use this block in a model created prior to R2013a, update it by using the `slupdate` utility. For more information, see the R2013a Release Notes.

Dialog Box and Parameters



Block Parameters: AC Voltage Source

AC Voltage Source

The ideal AC voltage source maintains the sinusoidal voltage across its output terminals, independent of the current flowing through the source. The output voltage is defined by V = V0 * sin(2*pi*f*t + PHI), where V0 is the peak amplitude, f is the frequency in Hz, and PHI is the phase shift in radians.

View source for AC Voltage Source

Parameters

Peak amplitude: 1 V

Phase shift: 0 deg

Frequency: 60 Hz

Peak amplitude
Peak voltage amplitude. The default value is 1 V.

Phase shift
Phase shift in angular units. The default value is 0.

Frequency
Voltage frequency, specified in Hz or units directly convertible to Hz (where Hz is defined as 1/s). For example, kHz and MHz are valid units, but rad/s is not. The default value is 60 Hz.

Ports

The block has the following ports:

+

Electrical conserving port associated with the source positive terminal.

Electrical conserving port associated with the source negative terminal.

See Also

AC Current Source

Adiabatic Cup

Purpose

Thermal element with no thermal mass and perfect insulation

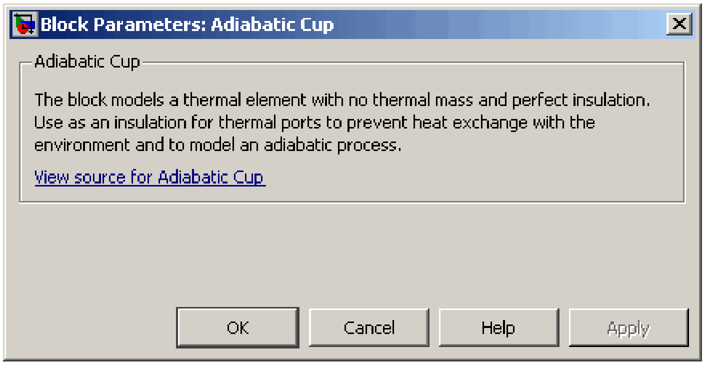
Library

Pneumatic Elements

Description

The Adiabatic Cup block models a thermal element with no thermal mass and perfect insulation. Use this block as an insulation for thermal ports to prevent heat exchange with the environment and to model an adiabatic process.

Dialog Box and Parameters



Block Parameters: Adiabatic Cup

Adiabatic Cup

The block models a thermal element with no thermal mass and perfect insulation. Use as an insulation for thermal ports to prevent heat exchange with the environment and to model an adiabatic process.

View source for Adiabatic Cup

The block has no parameters.

Ports

The block has one thermal conserving port.

Asynchronous Sample & Hold

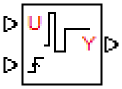
Purpose

Output sample-and-hold signal with external trigger

Library

Physical Signals/Discrete

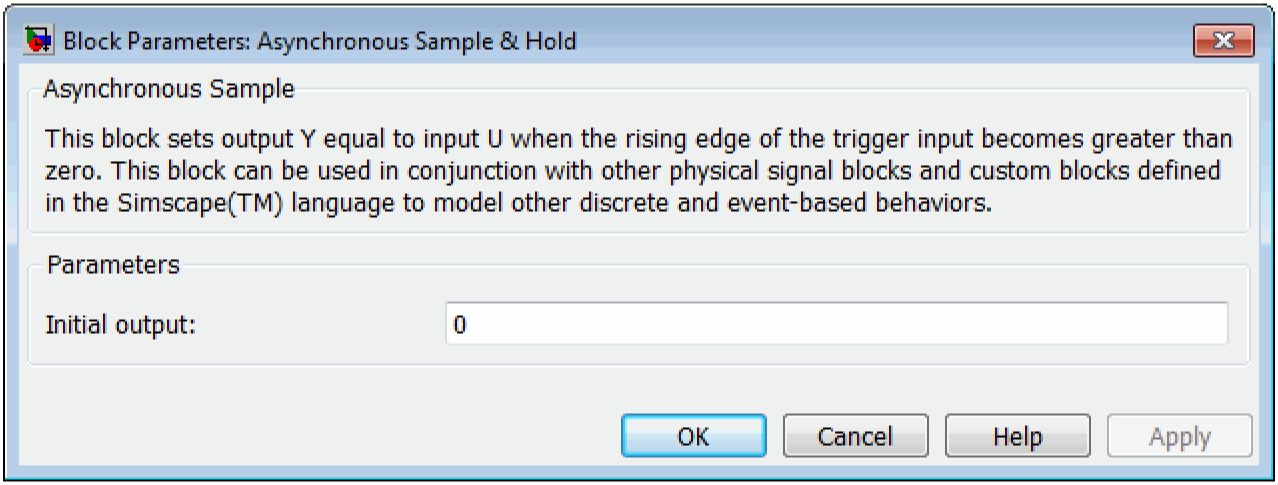
Description



The Asynchronous Sample & Hold block sets the output signal, *Y*, equal to the input signal, *U*, when the rising edge of the trigger input becomes greater than zero. Use this block, in conjunction with other physical signal blocks, to model discrete and event-based behaviors.

Both inputs and the output are physical signals.

Dialog Box and Parameters



Block Parameters: Asynchronous Sample & Hold

Asynchronous Sample

This block sets output Y equal to input U when the rising edge of the trigger input becomes greater than zero. This block can be used in conjunction with other physical signal blocks and custom blocks defined in the Simscape(TM) language to model other discrete and event-based behaviors.

Parameters

Initial output: 0

Initial output

The value of the output signal at time zero. The output of the block remains at this value until the block is triggered by a rising trigger signal becoming positive. The default value is `0`.

Ports

The block has two physical signal input ports and one physical signal output port.

Examples

The Asynchronous PWM Voltage Source example illustrates how you can use the Asynchronous Sample & Hold block to build components with more complex behaviors. For an alternative discrete-time implementation, see the Discrete-Time PWM Voltage Source example. The discrete-time version is better suited to fixed-step solvers and hardware-in-the-loop applications, whereas the asynchronous implementation is better suited to fast desktop simulation using variable-step solvers.

See Also

Counter

Purpose

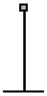
Liquid flow stop

Library

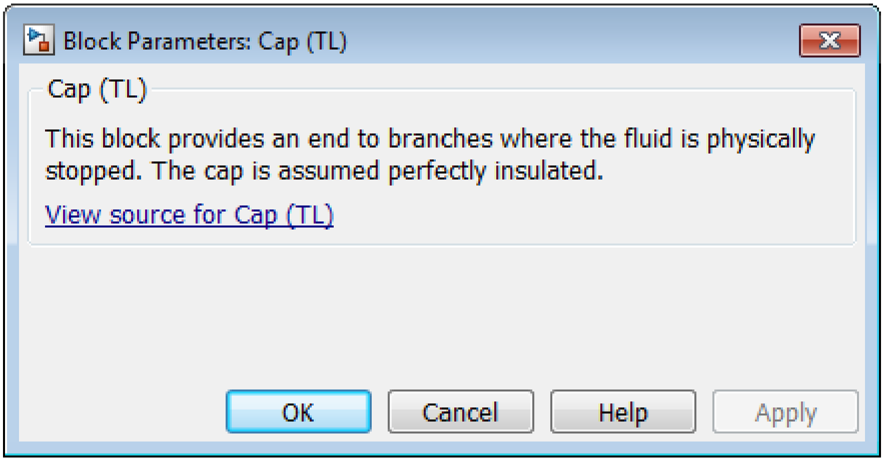
Thermal Liquid/Elements

Description

The Cap (TL) block represents a physical stop to liquid flow in a pipe network branch. The stop is perfectly insulated, preventing heat transfer with its surroundings.



Dialog Box and Parameters



Block Parameters: Cap (TL)

Cap (TL)

This block provides an end to branches where the fluid is physically stopped. The cap is assumed perfectly insulated.

View source for Cap (TL)

The block has no parameters.

Ports

The block has one thermal liquid conserving port.

Capacitor

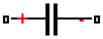
Purpose

Linear capacitor in electrical systems

Library

Electrical Elements

Description



The Capacitor block models a linear capacitor, described with the following equation:

$$I = C\frac{dV}{dt}$$

where

I	Current
V	Voltage
C	Capacitance
t	Time

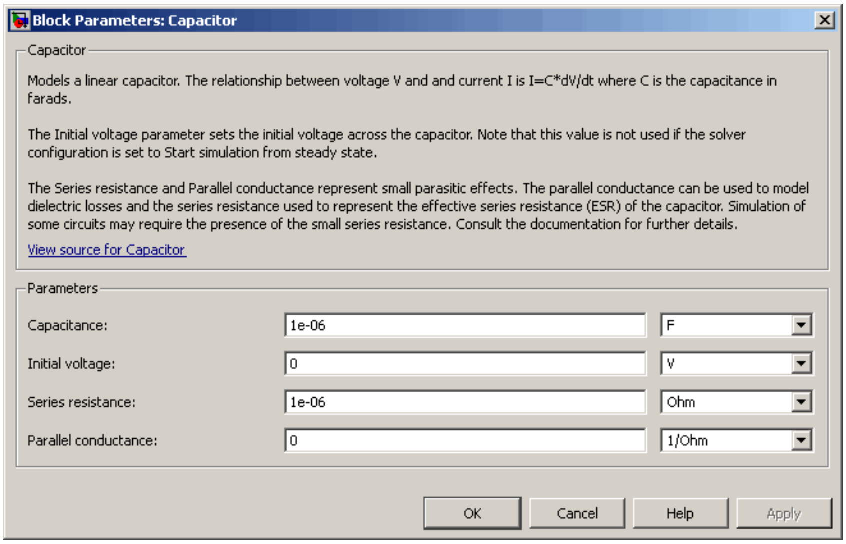
The **Initial voltage** parameter sets the initial voltage across the capacitor.

Note This value is not used if the solver configuration is set to **Start simulation from steady state**.

The **Series resistance** and **Parallel conductance** parameters represent small parasitic effects. The parallel conductance directly across the capacitor can be used to model dielectric losses, or equivalently leakage current per volt. The series resistance can be used to represent component effective series resistance (ESR) or connection resistance. Simulation of some circuits may require the presence of the small series resistance. For more information, see “Modeling Best Practices” in the Simscape™ User’s Guide.

Connections + and – are conserving electrical ports corresponding to the positive and negative terminals of the capacitor, respectively. The current is positive if it flows from positive to negative, and the voltage across the capacitor is equal to the difference between the voltage at the positive and the negative terminal, $V(+) - V(–)$.

Dialog Box and Parameters



Block Parameters: Capacitor

Capacitor

Models a linear capacitor. The relationship between voltage V and and current I is I=C*dV/dt where C is the capacitance in farads.

The Initial voltage parameter sets the initial voltage across the capacitor. Note that this value is not used if the solver configuration is set to Start simulation from steady state.

The Series resistance and Parallel conductance represent small parasitic effects. The parallel conductance can be used to model dielectric losses and the series resistance used to represent the effective series resistance (ESR) of the capacitor. Simulation of some circuits may require the presence of the small series resistance. Consult the documentation for further details.

View source for Capacitor

Parameters

Capacitance: 1e-06 F

Initial voltage: 0 V

Series resistance: 1e-06 Ohm

Parallel conductance: 0 1/Ohm

Capacitance

Capacitance, in farads. The default value is 1 µF.

Initial voltage

Initial voltage across the capacitor. This parameter is not used if the solver configuration is set to **Start simulation from steady state**. The default value is 0.

Series resistance

Represents small parasitic effects. The series resistance can be used to represent component internal resistance. Simulation of some circuits may require the presence of the small series resistance. The default value is 1 µΩ.

Parallel conductance

Represents small parasitic effects. The parallel conductance directly across the capacitor can be used to model leakage current per volt. The default value is `0`.

Ports

The block has the following ports:

+

Electrical conserving port associated with the capacitor positive terminal.

-

Electrical conserving port associated with the capacitor negative terminal.

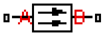
Purpose

Heat transfer by conduction

Library

Thermal Elements

Description



The Conductive Heat Transfer block represents a heat transfer by conduction between two layers of the same material. The transfer is governed by the Fourier law and is described with the following equation:

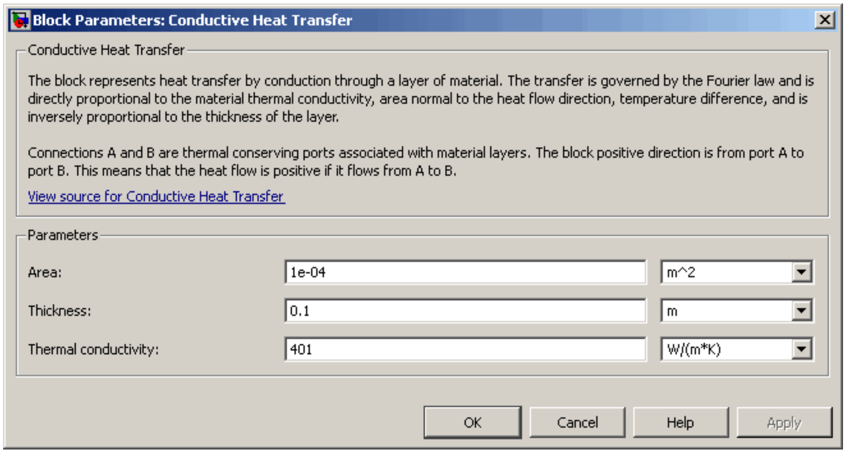
$$Q = k\square\frac{A}{D}(T_A - T_B)$$

where

Q	Heat flow
k	Material thermal conductivity
A	Area normal to the heat flow direction
D	Distance between layers
T_A, T_B	Temperatures of the layers

Connections A and B are thermal conserving ports associated with material layers. The block positive direction is from port A to port B. This means that the heat flow is positive if it flows from A to B.

Dialog Box and Parameters



Conductive Heat Transfer

The block represents heat transfer by conduction through a layer of material. The transfer is governed by the Fourier law and is directly proportional to the material thermal conductivity, area normal to the heat flow direction, temperature difference, and is inversely proportional to the thickness of the layer.

Connections A and B are thermal conserving ports associated with material layers. The block positive direction is from port A to port B. This means that the heat flow is positive if it flows from A to B.

View source for Conductive Heat Transfer

Parameters

Area:	1e-04	m^2
Thickness:	0.1	m
Thermal conductivity:	401	W/(m*K)

Area

Area of heat transfer, normal to the heat flow direction. The default value is `0.0001` m^2.

Thickness

Thickness between layers. The default value is `0.1` m.

Thermal conductivity

Thermal conductivity of the material. The default value is `401` W/m/K.

Ports

The block has the following ports:

A

Thermal conserving port associated with layer A.

B

Thermal conserving port associated with layer B.

See Also

Convective Heat Transfer

Radiative Heat Transfer

Purpose

Physical Modeling connector port for subsystem

Library

Utilities

Description

The Connection Port block transfers both the conserving and the physical signal connections to the outside boundary of a subsystem block. This transfer is similar to the Inport and Outport blocks in Simulink® models. A subsystem needs a Connection Port block for each physical connection line that crosses its boundary. You can manually place a Connection Port block inside a subsystem, or Simulink can automatically insert a Connection Port block when you create a subsystem within an existing network.

Port Appearance on Subsystem Boundary

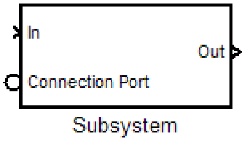
The ports on the subsystem boundary change their appearance depending on the type of port to which the Connection Port block is connected inside the subsystem.

Connection Port Block Inside a Subsystem Connects to ...		... and Appears on the Outside Boundary of the Subsystem as ...	
A Conserving port		A square Conserving port ▣	
A Physical Signal inport or outport		A triangular Physical Signal ▷ inport or outport	
A two-way connector port of the Two-Way Connection block		A two-way connector port ⊞	
A SimMechanics™ connector port, either:		A SimMechanics connector port, either:	
	Round connector port ○		Round connector port ○
	Body coordinate system port ◪		Body coordinate system port ◪

Port Location and Orientation on Subsystem Boundary

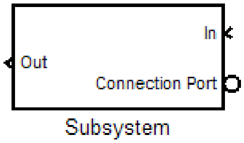
The orientation of the parent subsystem block and your choice of port location determine the Connection Port block port location on the parent subsystem boundary.

- A subsystem is in its fundamental orientation when its Simulink signal inports occur on its left side and its Simulink signal outports occur on its right side.

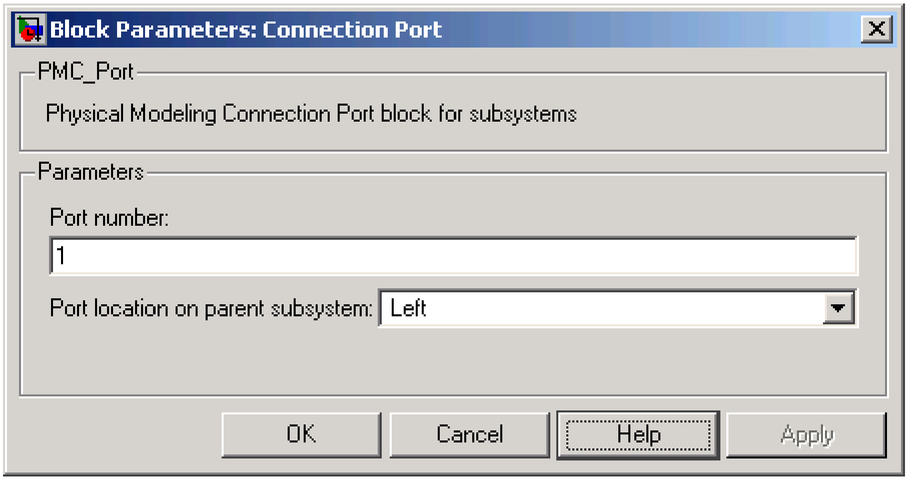
  When a subsystem is oriented in this way, the actual port location on the subsystem boundary respects your choice of port location (left or right) for the connector port.

- A subsystem orientation is reversed, with left and right interchanged, when its Simulink signal inports occur on its right side and its Simulink signal outports occur on its left side.

  When a subsystem is oriented in this way, the actual port location on the subsystem boundary reverses your choice of port location. If you choose left, the port appears on the right side. If you choose right, the port appears on the left side.

Dialog Box and Parameters



Block Parameters: Connection Port

PMC_Port

Physical Modeling Connection Port block for subsystems

Parameters

Port number: 1

Port location on parent subsystem: Left

Port number

Labels the subsystem connector port that this block creates. Each connector port on the boundary of a single subsystem requires a unique number as a label. The default value for the first port is `1`.

Port location on parent subsystem

Choose here which side of the parent subsystem boundary the port is located. The choices are `Left` or `Right`. The default choice is `Left`.

See "Port Location and Orientation on Subsystem Boundary" on page 1-20.

See Also

In the Simulink documentation, see "Create a Subsystem".

Constant Area Hydraulic Orifice

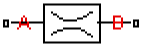
Purpose

Hydraulic orifice with constant cross-sectional area

Library

Hydraulic Elements

Description



The Constant Area Hydraulic Orifice block models a sharp-edged constant-area orifice. The model distinguishes between the laminar and turbulent flow regimes by comparing the Reynolds number with its critical value. The flow rate through the orifice is proportional to the pressure differential across the orifice, and is determined according to the following equations:

$$q = \begin{cases} C_D \cdot A \sqrt{\frac{2}{\rho} |\, p \,|} \cdot sign(p) & \text{for } Re >= Re_{\mathrm{cr}} \\ 2C_{DL} \cdot A \frac{D_H}{\nu \cdot \rho} p & \text{for } Re < Re_{\mathrm{cr}} \end{cases}$$

$$p = p_A - p_B$$

$$\mathrm{Re} = \frac{q \cdot D_H}{A \cdot \nu}$$

$$C_{DL} = \left(\frac{C_D}{\sqrt{\mathrm{Re}_{cr}}} \right)^2$$

$$D_H = \sqrt{\frac{4A}{\pi}}$$

where

q	Flow rate
p	Pressure differential
p_A, p_B	Gauge pressures at the block terminals

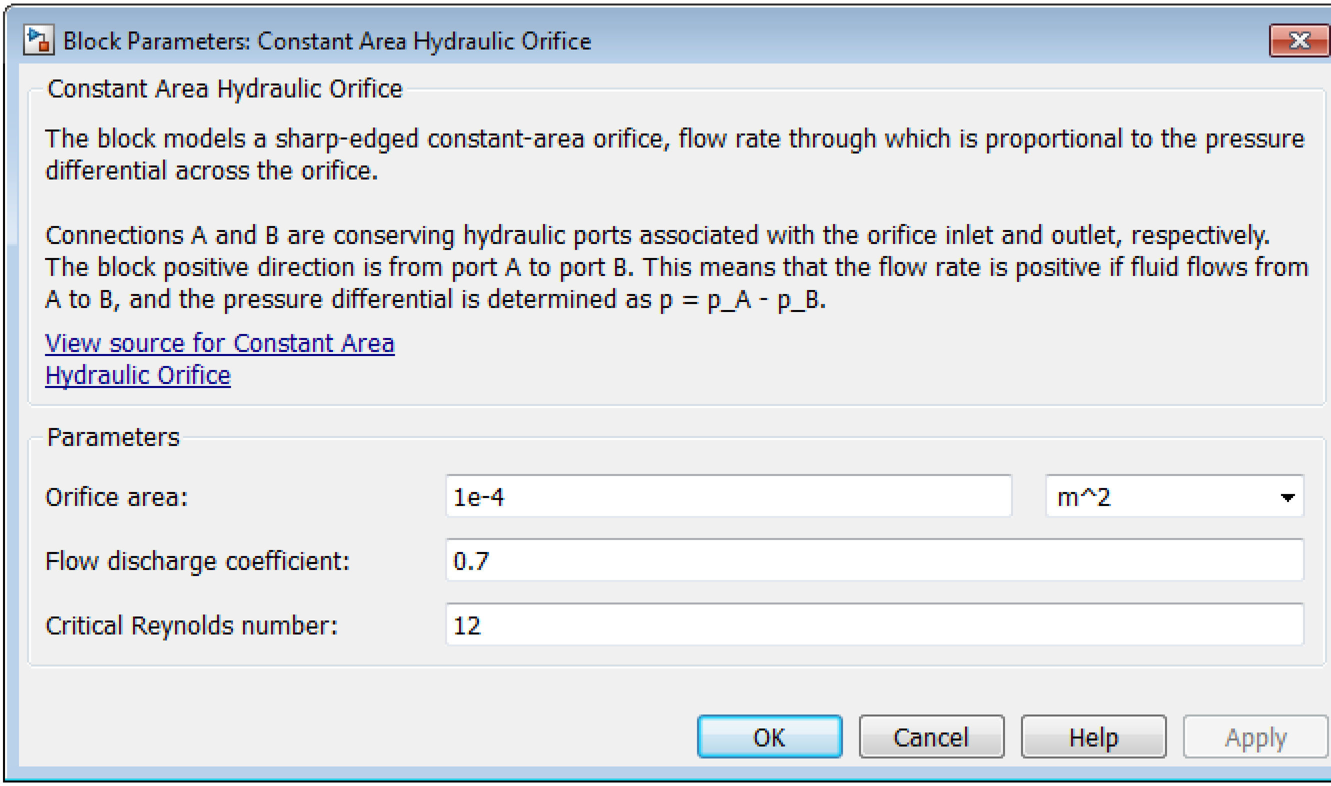
C_D	Flow discharge coefficient
A	Orifice passage area
D_H	Orifice hydraulic diameter
ρ	Fluid density
ν	Fluid kinematic viscosity
Re	Reynolds number
Re_{cr}	Critical Reynolds number

The block positive direction is from port A to port B. This means that the flow rate is positive if it flows from A to B, and the pressure differential is determined as $p = p_A - p_B$.

Basic Assumptions and Limitations

- Fluid inertia is not taken into account.
- The transition between laminar and turbulent regimes is assumed to be sharp and taking place exactly at Re=Re_{cr}.

Dialog Box and Parameters



Block Parameters: Constant Area Hydraulic Orifice

Constant Area Hydraulic Orifice

The block models a sharp-edged constant-area orifice, flow rate through which is proportional to the pressure differential across the orifice.

Connections A and B are conserving hydraulic ports associated with the orifice inlet and outlet, respectively. The block positive direction is from port A to port B. This means that the flow rate is positive if fluid flows from A to B, and the pressure differential is determined as p = p_A - p_B.

View source for Constant Area Hydraulic Orifice

Parameters

Orifice area: 1e-4 m^2

Flow discharge coefficient: 0.7

Critical Reynolds number: 12

Orifice area
Orifice passage area. The default value is `1e-4` m^2.

Flow discharge coefficient
Semi-empirical parameter for orifice capacity characterization. Its value depends on the geometrical properties of the orifice, and usually is provided in textbooks or manufacturer data sheets. The default value is `0.7`.

Critical Reynolds number

The maximum Reynolds number for laminar flow. The transition from laminar to turbulent regime is assumed to take place when the Reynolds number reaches this value. The value of the parameter depends on orifice geometrical profile, and the recommendations on the parameter value can be found in hydraulic textbooks. The default value is 12, which corresponds to a round orifice in thin material with sharp edges.

Global Parameters

Parameters determined by the type of working fluid:

- **Fluid density**
- **Fluid kinematic viscosity**

Use the Hydraulic Fluid block or the Custom Hydraulic Fluid block to specify the fluid properties.

Ports

The block has the following ports:

A

Hydraulic conserving port associated with the orifice inlet.

B

Hydraulic conserving port associated with the orifice outlet.

See Also

Variable Area Hydraulic Orifice

Constant Area Pneumatic Orifice

Purpose

Sharp-edged orifice in pneumatic systems

Library

Pneumatic Elements

Description

The Constant Area Pneumatic Orifice block models the flow rate of an ideal gas through a sharp-edged orifice.

The flow rate through the orifice is proportional to the orifice area and the pressure differential across the orifice.

$$G = C_d \cdot A \cdot p_i \sqrt{\frac{2\gamma}{\gamma - 1} \cdot \frac{1}{RT_i} \left[\left(\frac{p_o}{p_i} \right)^{\frac{2}{\gamma}} - \left(\frac{p_o}{p_i} \right)^{\frac{\gamma+1}{\gamma}} \right]}$$

where

G Mass flow rate

C_d Discharge coefficient, to account for effective loss of area due to orifice shape

A Orifice cross-sectional area

p_i, p_o Absolute pressures at the orifice inlet and outlet, respectively. The inlet and outlet change depending on flow direction. For positive flow ($G > 0$), $p_i = p_A$, otherwise $p_i = p_B$.

γ The ratio of specific heats at constant pressure and constant volume, c_p / c_v

R Specific gas constant

T Absolute gas temperature

The choked flow occurs at the critical pressure ratio defined by

$$\beta_{cr} = \frac{p_o}{p_i} = \left(\frac{2}{\gamma + 1} \right)^{\frac{\gamma}{\gamma - 1}}$$

after which the flow rate depends on the inlet pressure only and is computed with the expression

$$G = C_d \cdot A \cdot p_i \sqrt{\frac{\gamma}{RT_i} \cdot \beta_{cr}^{\frac{\gamma+1}{\gamma}}}$$

The square root relationship has infinite gradient at zero flow, which can present numerical solver difficulties. Therefore, for very small pressure differences, defined by p_o / p_i > 0.999, the flow equation is replaced by a linear flow-pressure relationship

$$G = kC_d \cdot A \cdot T_i^{-0.5}\left(p_i - p_o\right)$$

where k is a constant such that the flow predicted for p_o / p_i is the same as that predicted by the original flow equation for p_o / p_i = 0.999.

The heat flow out of the orifice is assumed equal to the heat flow into the orifice, based on the following considerations:

- The orifice is square-edged or sharp-edged, and as such is characterized by an abrupt change of the downstream area. This means that practically all the dynamic pressure is lost in the expansion.
- The lost energy appears in the form of internal energy that rises the output temperature and makes it very close to the inlet temperature.

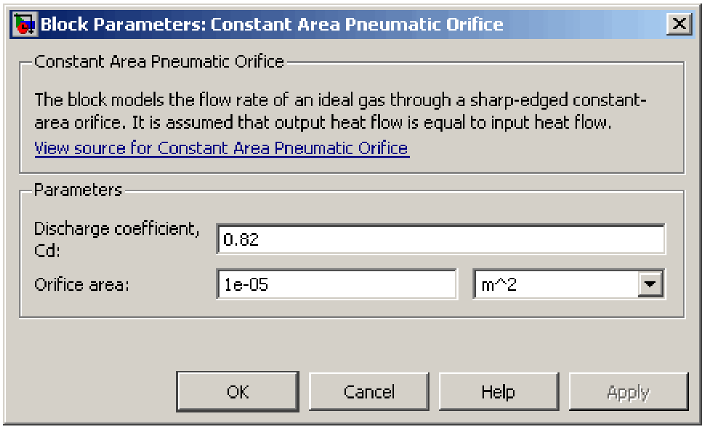
Therefore, $q_i = q_o$, where q_i and q_o are the input and output heat flows, respectively.

The block positive direction is from port A to port B. This means that the flow rate is positive if it flows from A to B.

Basic Assumptions and Limitations

- The gas is ideal.
- Specific heats at constant pressure and constant volume, c_p and c_v, are constant.
- The process is adiabatic, that is, there is no heat transfer with the environment.
- Gravitational effects can be neglected.
- The orifice adds no net heat to the flow.

Dialog Box and Parameters



Block Parameters: Constant Area Pneumatic Orifice

Constant Area Pneumatic Orifice

The block models the flow rate of an ideal gas through a sharp-edged constant-area orifice. It is assumed that output heat flow is equal to input heat flow.

View source for Constant Area Pneumatic Orifice

Parameters

Discharge coefficient, Cd: 0.82

Orifice area: 1e-05 m^2

Discharge coefficient, Cd

Semi-empirical parameter for orifice capacity characterization. Its value depends on the geometrical properties of the orifice, and usually is provided in textbooks or manufacturer data sheets. The default value is `0.82`.

Orifice area

Specify the orifice cross-sectional area. The default value is `1e-5` m^2.

Ports

The block has the following ports:

A

Pneumatic conserving port associated with the orifice inlet for positive flow.

B

Pneumatic conserving port associated with the orifice outlet for positive flow.

See Also

Constant Area Pneumatic Orifice (ISO 6358)

Variable Area Pneumatic Orifice

Constant Area Pneumatic Orifice (ISO 6358)

Purpose

Fixed-area pneumatic orifice complying with ISO 6358 standard

Library

Pneumatic Elements

Description

The Constant Area Pneumatic Orifice (ISO 6358) block models the flow rate of an ideal gas through a fixed-area sharp-edged orifice. The model conforms to the ISO 6358 standard and is based on the following flow equations, originally proposed by Sanville [1]:

$$G = \begin{cases} k_1 \cdot p_i \left(1 - \dfrac{p_o}{p_i}\right) \sqrt{\dfrac{T_{ref}}{T_i}} \cdot sign\left(p_i - p_o\right) & \text{if } \dfrac{p_o}{p_i} > \beta_{lam} \text{ (laminar)} \\ p_i \cdot C \cdot \rho_{ref} \sqrt{\dfrac{T_{ref}}{T_i}} \cdot \sqrt{1 - \left(\dfrac{\dfrac{p_o}{p_i} - b}{1 - b}\right)^2} & \text{if } \beta_{lam} > \dfrac{p_o}{p_i} > b \text{ (subsonic)} \\ p_i \cdot C \cdot \rho_{ref} \sqrt{\dfrac{T_{ref}}{T_i}} & \text{if } \dfrac{p_o}{p_i} <= b \text{ (choked)} \end{cases}$$

$$k_1 = \frac{1}{1 - \beta_{lam}} \cdot C \cdot \rho_{ref} \sqrt{1 - \left(\frac{\beta_{lam} - b}{1 - b}\right)^2}$$

where

G	Mass flow rate
β_{lam}	Pressure ratio at laminar flow, a value between 0.999 and 0.995
b	Critical pressure ratio, that is, the ratio between the outlet pressure p_o and inlet pressure p_i at which the gas velocity achieves sonic speed

C Sonic conductance of the component, that is, the ratio between the mass flow rate and the product of inlet pressure p_1 and the mass density at standard conditions when the flow is choked

ρ_{ref} Gas density at which the sonic conductance was measured (1.185 kg/m^3 for air)

p_i, p_o Absolute pressures at the orifice inlet and outlet, respectively. The inlet and outlet change depending on flow direction. For positive flow ($G > 0$), $p_i = p_A$, otherwise $p_i = p_B$.

T_i, T_o Absolute gas temperatures at the orifice inlet and outlet, respectively

T_{ref} Gas temperature at which the sonic conductance was measured (T_{ref} = 293.15 K)

The equation itself, parameters b and C, and the heuristic on how to measure these parameters experimentally form the basis for the standard ISO 6358 (1989). The values of the critical pressure ratio b and the sonic conductance C depend on a particular design of a component. Typically, they are determined experimentally and are sometimes given on a manufacturer data sheet.

The block can also be parameterized in terms of orifice effective area or flow coefficient, instead of sonic conductance. When doing so, block parameters are converted into an equivalent value for sonic conductance. When specifying effective area, the following formula proposed by Gidlund and detailed in [2] is used:

$$C = 0.128\, d^{2}$$

where

C Sonic conductance in dm^3/(s*bar)

d Inner diameter of restriction in mm

The effective area (whether specified directly, or calculated when the orifice is parameterized in terms of C_v or K_v, as described below) is used to determine the inner diameter d in the Gidlund formula, assuming a circular cross section.

Gidlund also gives an approximate formula for the critical pressure ratio in terms of the pneumatic line diameter D,

$$b = 0.41 + 0.272\ d \ / \ D$$

This equation is not used by the block and you must specify the critical pressure ratio directly.

If the orifice is parameterized in terms of the C_v [2] coefficient, then the C_v coefficient is turned into an equivalent effective orifice area for use in the Gidlund formula:

$$A = 1.6986\mathrm{e} - 5\ C_v$$

By definition, an opening or restriction has a C_v coefficient of 1 if it passes 1 gpm (gallon per minute) of water at pressure drop of 1 psi.

If the orifice is parameterized in terms of the K_v [2] coefficient, then the K_v coefficient is turned into an equivalent effective orifice area for use in the Gidlund formula:

$$A = 1.1785\mathrm{e} - 6\ C_v$$

K_v is the SI counterpart of C_v. An opening or restriction has a K_v coefficient of 1 if it passes 1 lpm (liter per minute) of water at pressure drop of 1 bar.

The heat flow out of the orifice is assumed equal to the heat flow into the orifice, based on the following considerations:

- The orifice is square-edged or sharp-edged, and as such is characterized by an abrupt change of the downstream area. This means that practically all the dynamic pressure is lost in the expansion.

- The lost energy appears in the form of internal energy that rises the output temperature and makes it very close to the inlet temperature.

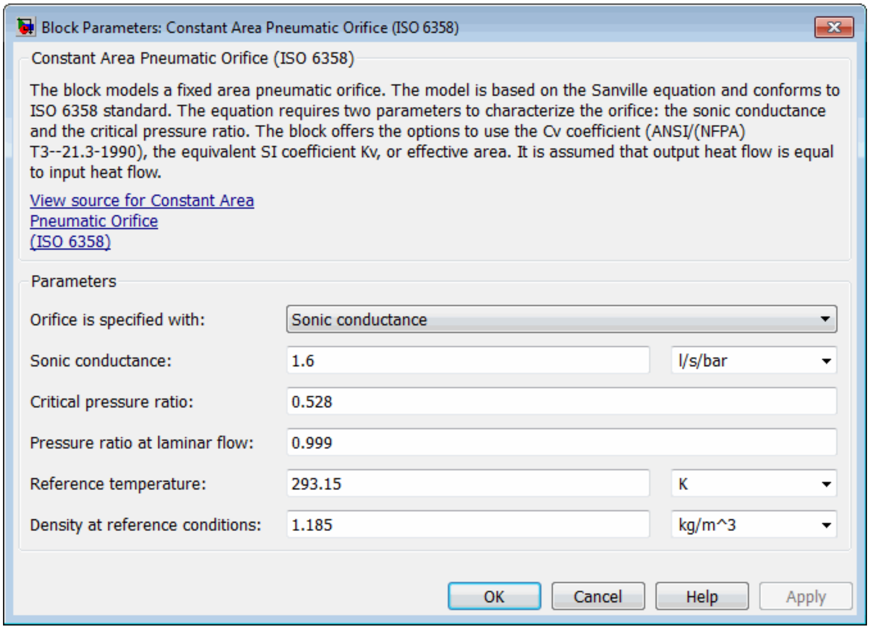
Therefore, $q_i = q_o$, where q_i and q_o are the input and output heat flows, respectively.

The block positive direction is from port A to port B. This means that the flow rate is positive if it flows from A to B.

Basic Assumptions and Limitations

- The gas is ideal.
- Specific heats at constant pressure and constant volume, c_p and c_v, are constant.
- The process is adiabatic, that is, there is no heat transfer with the environment.
- Gravitational effects can be neglected.
- The orifice adds no net heat to the flow.

Dialog Box and Parameters



Block Parameters: Constant Area Pneumatic Orifice (ISO 6358)

Constant Area Pneumatic Orifice (ISO 6358)

The block models a fixed area pneumatic orifice. The model is based on the Sanville equation and conforms to ISO 6358 standard. The equation requires two parameters to characterize the orifice: the sonic conductance and the critical pressure ratio. The block offers the options to use the Cv coefficient (ANSI/(NFPA) T3--21.3-1990), the equivalent SI coefficient Kv, or effective area. It is assumed that output heat flow is equal to input heat flow.

View source for Constant Area Pneumatic Orifice (ISO 6358)

Parameters

Orifice is specified with: Sonic conductance

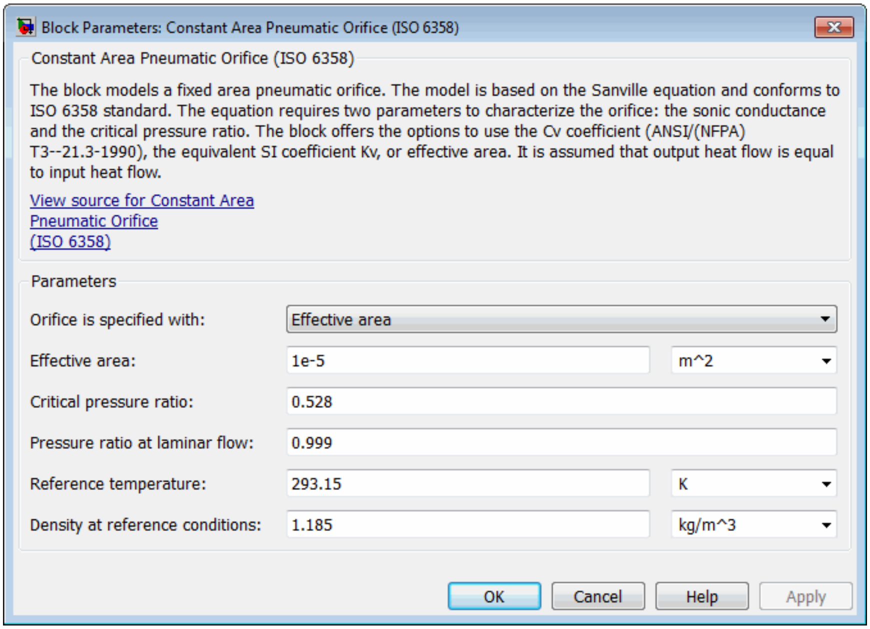
Sonic conductance: 1.6 l/s/bar

Critical pressure ratio: 0.528

Pressure ratio at laminar flow: 0.999

Reference temperature: 293.15 K

Density at reference conditions: 1.185 kg/m^3



Block Parameters: Constant Area Pneumatic Orifice (ISO 6358)

Constant Area Pneumatic Orifice (ISO 6358)

The block models a fixed area pneumatic orifice. The model is based on the Sanville equation and conforms to ISO 6358 standard. The equation requires two parameters to characterize the orifice: the sonic conductance and the critical pressure ratio. The block offers the options to use the Cv coefficient (ANSI/(NFPA) T3--21.3-1990), the equivalent SI coefficient Kv, or effective area. It is assumed that output heat flow is equal to input heat flow.

View source for Constant Area Pneumatic Orifice (ISO 6358)

Parameters

Orifice is specified with: Effective area

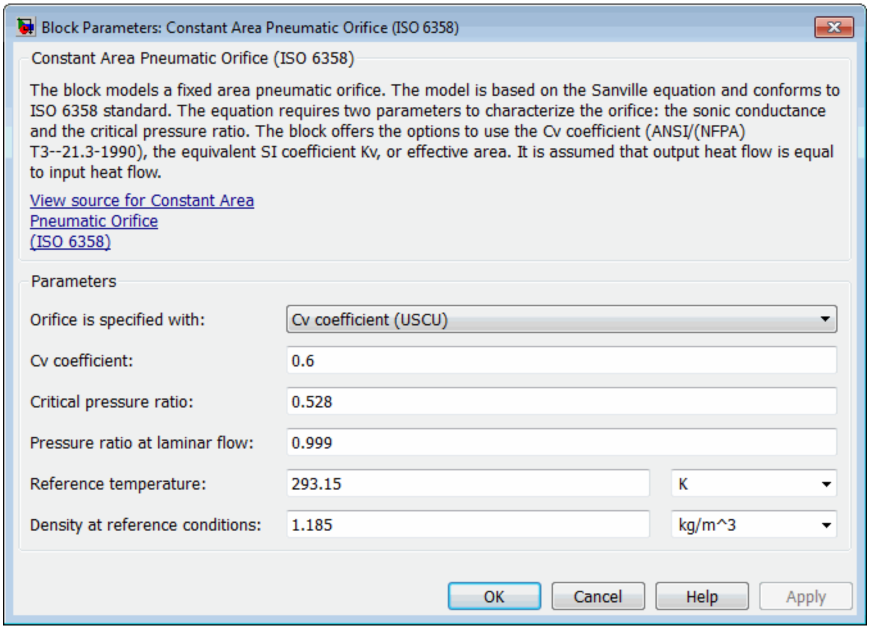
Effective area: 1e-5 m^2

Critical pressure ratio: 0.528

Pressure ratio at laminar flow: 0.999

Reference temperature: 293.15 K

Density at reference conditions: 1.185 kg/m^3



Block Parameters: Constant Area Pneumatic Orifice (ISO 6358)

Constant Area Pneumatic Orifice (ISO 6358)

The block models a fixed area pneumatic orifice. The model is based on the Sanville equation and conforms to ISO 6358 standard. The equation requires two parameters to characterize the orifice: the sonic conductance and the critical pressure ratio. The block offers the options to use the Cv coefficient (ANSI/(NFPA) T3--21.3-1990), the equivalent SI coefficient Kv, or effective area. It is assumed that output heat flow is equal to input heat flow.

View source for Constant Area Pneumatic Orifice (ISO 6358)

Parameters

Orifice is specified with: Cv coefficient (USCU)

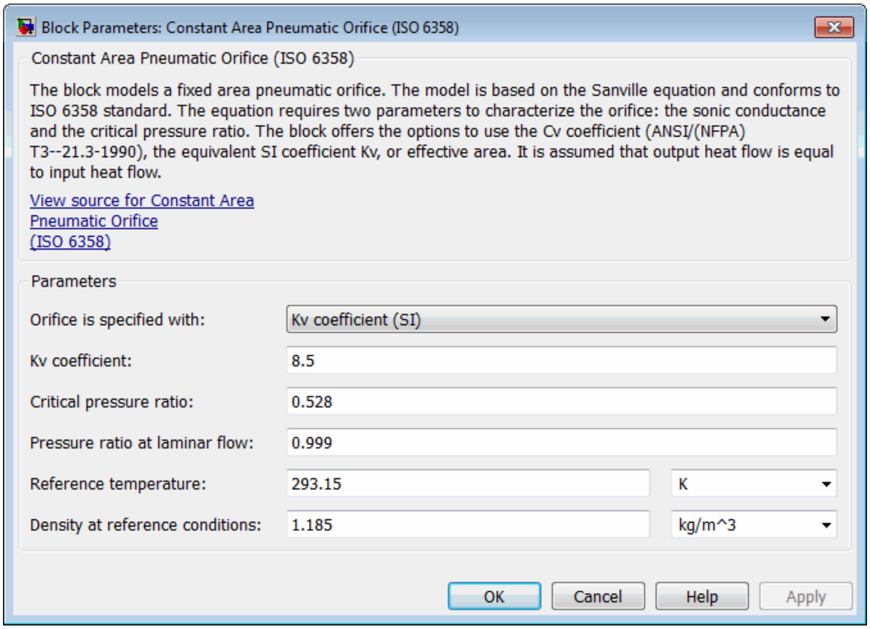
Cv coefficient: 0.6

Critical pressure ratio: 0.528

Pressure ratio at laminar flow: 0.999

Reference temperature: 293.15 K

Density at reference conditions: 1.185 kg/m^3



Block Parameters: Constant Area Pneumatic Orifice (ISO 6358)

Constant Area Pneumatic Orifice (ISO 6358)

The block models a fixed area pneumatic orifice. The model is based on the Sanville equation and conforms to ISO 6358 standard. The equation requires two parameters to characterize the orifice: the sonic conductance and the critical pressure ratio. The block offers the options to use the Cv coefficient (ANSI/(NFPA) T3--21.3-1990), the equivalent SI coefficient Kv, or effective area. It is assumed that output heat flow is equal to input heat flow.

View source for Constant Area Pneumatic Orifice (ISO 6358)

Parameters

Orifice is specified with: Kv coefficient (SI)

Kv coefficient: 8.5

Critical pressure ratio: 0.528

Pressure ratio at laminar flow: 0.999

Reference temperature: 293.15 K

Density at reference conditions: 1.185 kg/m^3

Orifice is specified with

Select one of the following model parameterization methods:

- `Sonic conductance` — Provide value for the sonic conductance of the orifice. The values of the sonic conductance and the critical pressure ratio form the basis for the ISO 6358 compliant flow equations for the orifice. This is the default method.
- `Effective area` — Provide value for the orifice effective area. This value is internally converted by the block into an equivalent value for sonic conductance.
- `Cv coefficient (USCU)` — Provide value for the flow coefficient specified in US units. This value is internally converted by the block into an equivalent value for the orifice effective area.

- `Kv coefficient (SI)` — Provide value for the flow coefficient specified in SI units. This value is internally converted by the block into an equivalent value for the orifice effective area.

Sonic conductance
Specify the sonic conductance of the orifice, that is, the ratio between the mass flow rate and the product of upstream pressure and the mass density at standard conditions when the flow is choked. This value depends on the geometrical properties of the orifice, and usually is provided in textbooks or manufacturer data sheets. The default value is `1.6` l/s/bar. This parameter appears in the dialog box if **Orifice is specified with** parameter is set to `Sonic conductance`.

Effective area
Specify the orifice cross-sectional area. The default value is `1e-5` m^2. This parameter appears in the dialog box if **Orifice is specified with** parameter is set to `Effective area`.

Cv coefficient
Specify the value for the flow coefficient in US units. The default value is `0.6`. This parameter appears in the dialog box if **Orifice is specified with** parameter is set to `Cv coefficient (USCU)`.

Kv coefficient
Specify the value for the flow coefficient in SI units. The default value is `8.5`. This parameter appears in the dialog box if **Orifice is specified with** parameter is set to `Kv coefficient (SI)`.

Critical pressure ratio
Specify the critical pressure ratio, that is, the ratio between the downstream pressure and the upstream pressure at which the gas velocity achieves sonic speed. The default value is `0.528`.

Pressure ratio at laminar flow
Specify the ratio between the downstream pressure and the upstream pressure at laminar flow. This value can be in the range between 0.995 and 0.999. The default value is `0.999`.

Reference temperature

Specify the gas temperature at which the sonic conductance was measured. The default value is `293.15` K.

Density at reference conditions

Specify the gas density at which the sonic conductance was measured. The default value is `1.185` kg/m^3.

Ports

The block has the following ports:

A

Pneumatic conserving port associated with the orifice inlet for positive flow.

B

Pneumatic conserving port associated with the orifice outlet for positive flow.

References

[1] Sanville, F. E. “A New Method of Specifying the Flow Capacity of Pneumatic Fluid Power Valves.” Paper D3, p.37-47. BHRA. Second International Fluid Power Symposium, Guildford, England, 1971.

[2] Beater, P. *Pneumatic Drives. System Design, Modeling, and Control.* New York: Springer, 2007.

See Also

Constant Area Pneumatic Orifice

Variable Area Pneumatic Orifice

Constant Volume Chamber (TL)

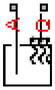
Purpose

Physical enclosure with fixed liquid volume

Library

Thermal Liquid/Elements

Description



The Constant Volume Chamber (TL) block represents a physical enclosure with a fixed volume of liquid. The enclosure can exchange mass and energy with its surroundings, allowing its internal pressure and temperature to evolve over time. Heat transfer occurs via convection, as liquid enters or exits the chamber, and conduction, as thermal energy flows through the chamber walls and the liquid itself.

Port A is a thermal liquid conserving port that corresponds to the chamber inlet. Pressure in the chamber equals the pressure at this port. Port Q is a thermal conserving port associated with the heat flux through the chamber wall. Temperature in the chamber equals the temperature at this port.

The following equations govern the dynamic evolution of liquid pressure and temperature in the chamber:

$$V \cdot \rho_{\text{int}} \cdot \left(\frac{1}{\beta_{\text{int}}} \cdot \frac{dp_{\text{int}}}{dt} + \alpha_{\text{int}} \cdot \frac{dT_{\text{int}}}{dt} \right) = \dot{m}_A$$

$$V \cdot \frac{d(\rho_{\text{int}} \cdot u_{\text{int}})}{dt} = \phi_A + \dot{Q}$$

where

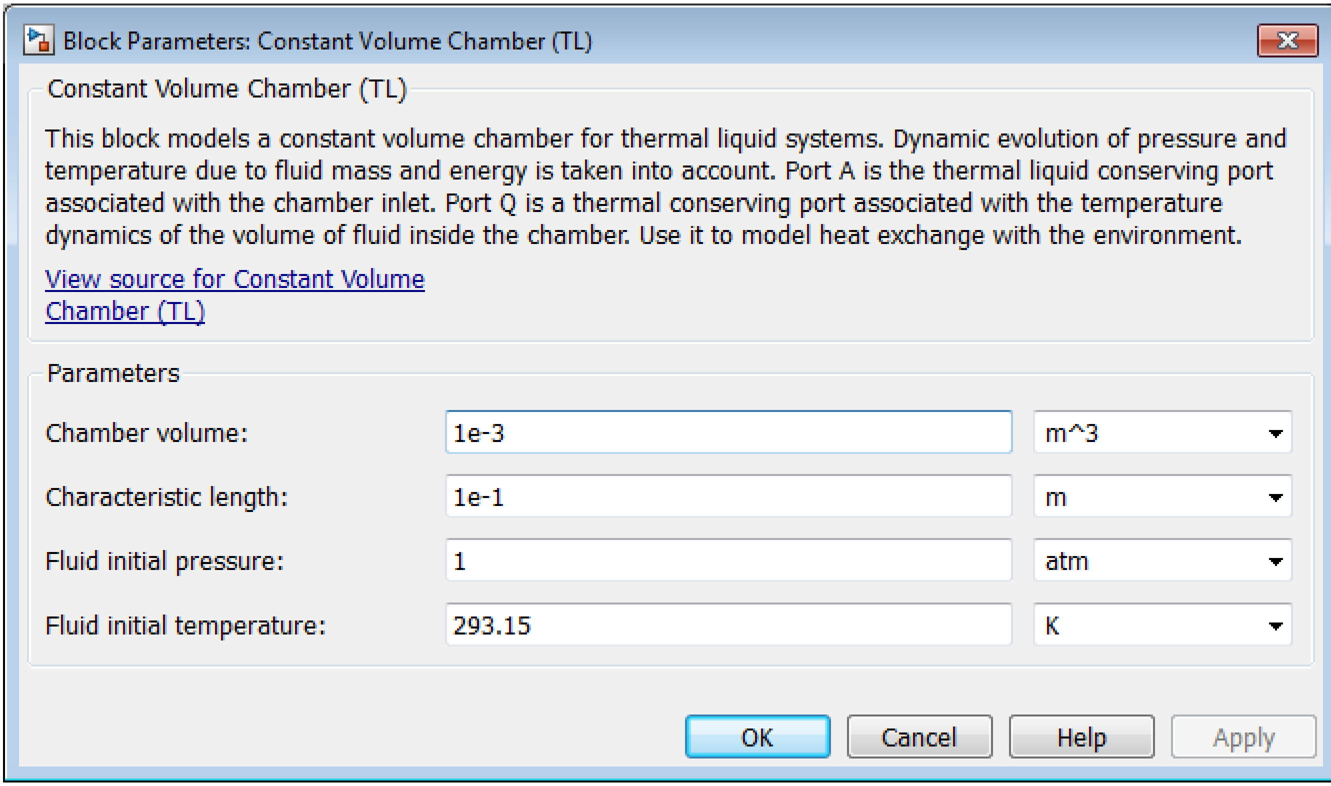
V	Chamber volume
p_{int}	Chamber pressure

T_{int}	Chamber temperature
$\dot{m}_A$	Mass flow rate into chamber
$_{\text{int}}$	Liquid isothermal bulk modulus
$_{\text{int}}$	Liquid isobaric thermal expansion coefficient
c_{int}	Liquid specific heat
u_{int}	Liquid specific internal energy
$_{\text{int}}$	Liquid mass density
$_{\text{A}}$	Thermal energy flux into chamber through port A
$\dot{Q}$	Thermal energy flux into chamber through port Q

Assumptions and Limitations

- Chamber walls are perfectly rigid.

Dialog Box and Parameters



Constant Volume Chamber (TL)

This block models a constant volume chamber for thermal liquid systems. Dynamic evolution of pressure and temperature due to fluid mass and energy is taken into account. Port A is the thermal liquid conserving port associated with the chamber inlet. Port Q is a thermal conserving port associated with the temperature dynamics of the volume of fluid inside the chamber. Use it to model heat exchange with the environment.

View source for Constant Volume Chamber (TL)

Parameters

Chamber volume: 1e-3 m^3

Characteristic length: 1e-1 m

Fluid initial pressure: 1 atm

Fluid initial temperature: 293.15 K

Chamber volume

Enter the volume of liquid present in the chamber. This volume remains constant throughout the simulation. Values must equal or exceed zero. The default value is `1e-3` m^3.

Characteristic length
Enter the characteristic length of the chamber. This length is the average distance that liquid must traverse as it leaves the chamber. The default value is `1e-1` m.

Fluid initial pressure
Enter the absolute pressure of the enclosed liquid at time zero. The default value is `1` atm.

Fluid initial temperature
Enter the temperature of the enclosed liquid at time zero. The default value is `293.15` K.

Ports

The block has one thermal liquid conserving port, A, and one thermal conserving port, Q.

See Also

Temperature Reservoir (TL)

Controlled Temperature Reservoir (TL)

Constant Volume Hydraulic Chamber

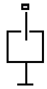
Purpose

Hydraulic capacity of constant volume

Library

Hydraulic Elements

Description



The Constant Volume Hydraulic Chamber block models a fixed-volume chamber with rigid or flexible walls, to be used in hydraulic valves, pumps, manifolds, pipes, hoses, and so on. Use this block in models where you have to account for some form of fluid compressibility. You can select the appropriate representation of fluid compressibility using the block parameters.

Fluid compressibility in its simplest form is simulated according to the following equations:

$$V_f = V_c + \frac{V_c}{E} p$$

$$q = \frac{dV_f}{dt}$$

where

q	Flow rate into the chamber
V_f	Volume of fluid in the chamber
V_c	Geometrical chamber volume
E	Fluid bulk modulus
p	Gauge pressure of fluid in the chamber

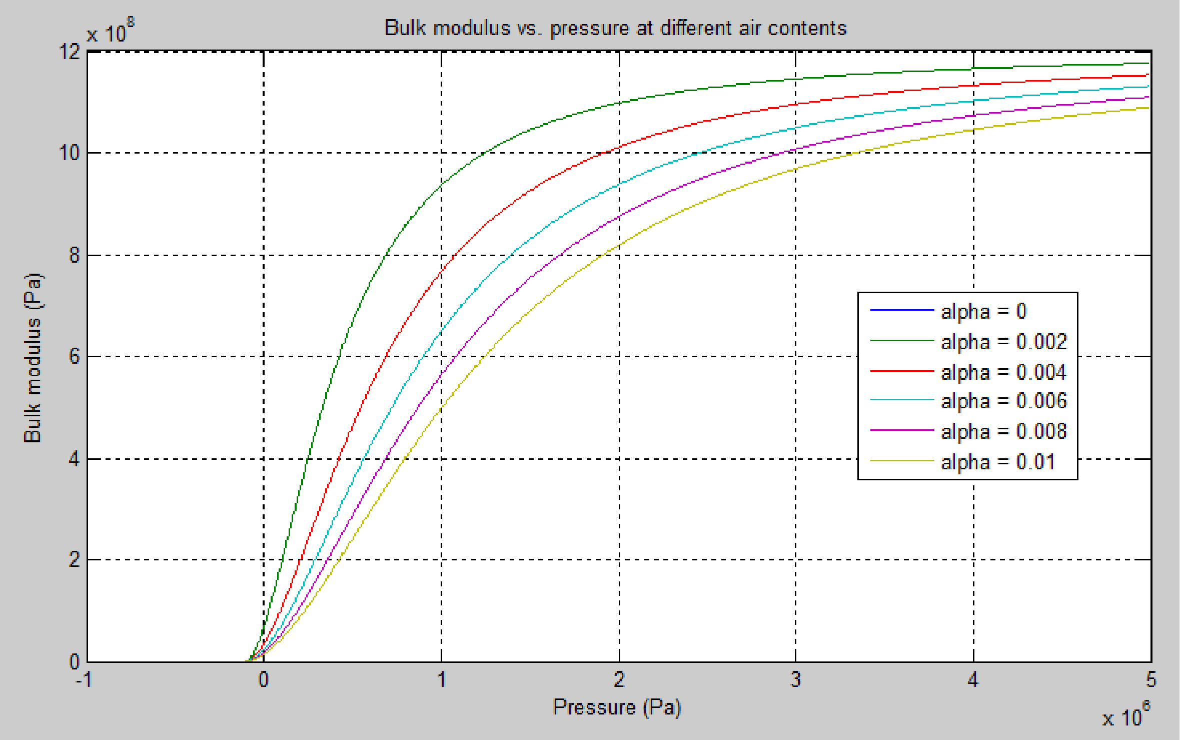
If pressure in the chamber is likely to fall to negative values and approach cavitation limit, the above equations must be enhanced. In this block, it is done by representing the fluid in the chamber as a mixture of liquid and a small amount of entrained, nondissolved gas (see [1, 2]). The mixture bulk modulus is determined as:

$$E = E_l \frac{1+\alpha\left(\dfrac{p_a}{p_a+p}\right)^{1/n}}{1+\alpha\dfrac{p_a^{1/n}}{n\cdot(p_a+p)^{\frac{n+1}{n}}}E_l}$$

where

E_1	Pure liquid bulk modulus
p_α	Atmospheric pressure
α	Relative gas content at atmospheric pressure, $\alpha = V_G/V_L$
V_G	Gas volume at atmospheric pressure
V_L	Volume of liquid
n	Gas-specific heat ratio

The main objective of representing fluid as a mixture of liquid and gas is to introduce an approximate model of cavitation, which takes place in a chamber if pressure drops below fluid vapor saturation level. As it is seen in the graph below, the bulk modulus of a mixture decreases at $p \to p_a$, thus considerably slowing down further pressure change. At high pressure, $p >> p_a$, a small amount of nondissolved gas has practically no effect on the system behavior.



Cavitation is an inherently thermodynamic process, requiring consideration of multiple-phase fluids, heat transfers, etc., and as such cannot be accurately simulated with Simscape software. But the simplified version implemented in the block is good enough to signal if pressure falls below dangerous level, and to prevent computation failure that normally occurs at negative pressures.

If pressure falls below absolute vacuum (–101325 Pa), the simulation stops and an error message is displayed.

If chamber walls have noticeable compliance, the above equations must be further enhanced by representing geometrical chamber volume as a function of pressure:

$$V_c = \pi d^2 / 4 \cdot L$$

$$d(s) = \frac{K_p}{1+\tau s} p(s)$$

where

d Internal diameter of the cylindrical chamber

L Length of the cylindrical chamber

K_p Proportionality coefficient (m/Pa)

τ Time constant

s Laplace operator

Coefficient K_p establishes relationship between pressure and the internal diameter at steady-state conditions. For metal tubes, the coefficient can be computed as (see [2]):

$$K_p = \frac{d}{E_M}\left(\frac{D^2+d^2}{D^2-d^2}+\nu\right)$$

where

D Pipe external diameter

E_M Modulus of elasticity (Young's modulus) for the pipe material

Poisson's ratio for the pipe material

For hoses, the coefficient can be provided by the manufacturer.

The process of expansion and contraction in pipes and especially in hoses is a complex combination of nonlinear elastic and viscoelastic deformations. This process is approximated in the block with the first-order lag, whose time constant is determined empirically (for example, see [3]).

As a result, by selecting appropriate values, you can implement four different models of fluid compressibility with this block:

- Chamber with rigid walls, no entrained gas in the fluid
- Cylindrical chamber with compliant walls, no entrained gas in the fluid
- Chamber with rigid walls, fluid with entrained gas
- Cylindrical chamber with compliant walls, fluid with entrained gas

The block allows two methods of specifying the chamber size:

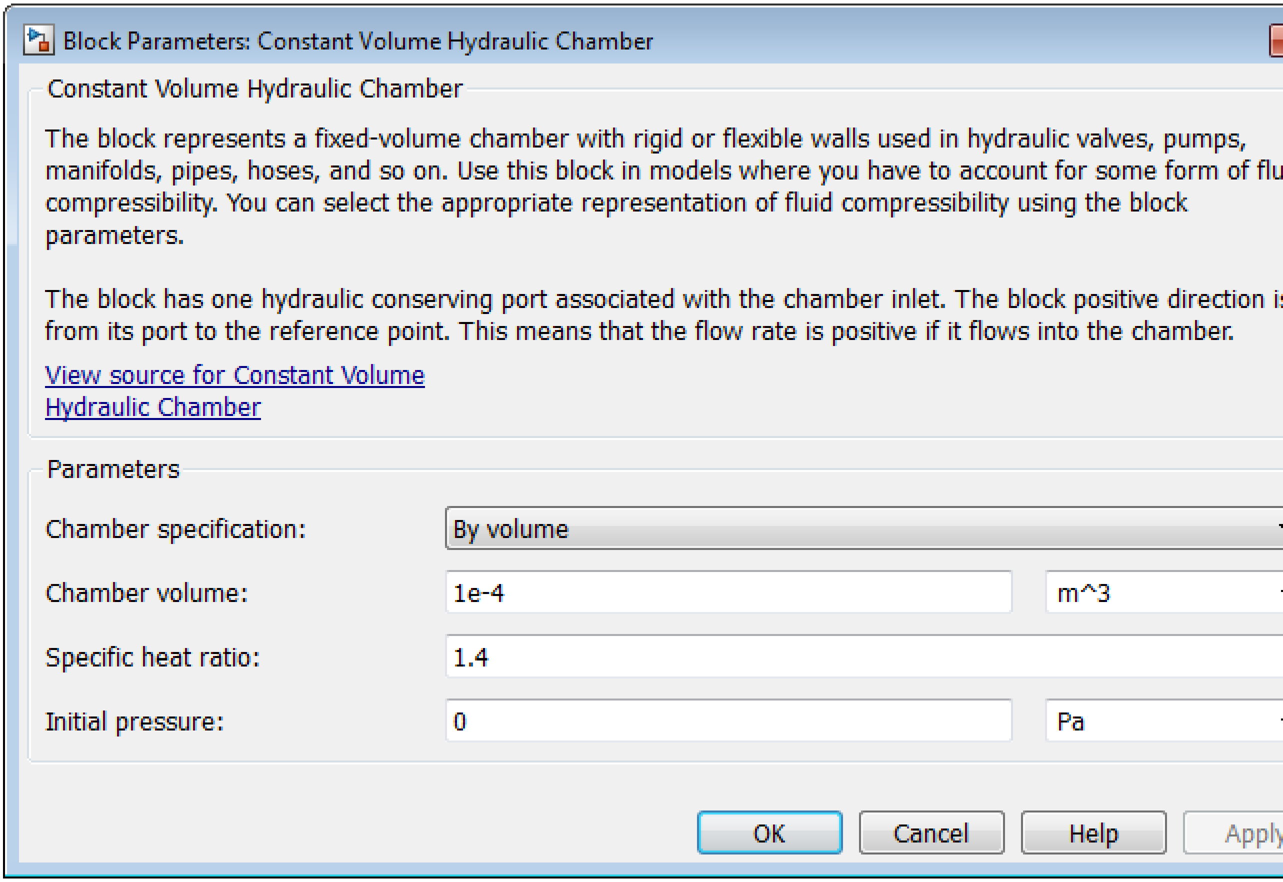
- By volume — Use this option for cylindrical or non-cylindrical chambers with rigid walls. You only need to know the volume of the chamber. This chamber type does not account for wall compliance.
- By length and diameter — Use this option for cylindrical chambers with rigid or compliant walls, such as circular pipes or hoses.

The block has one hydraulic conserving port associated with the chamber inlet. The block positive direction is from its port to the reference point. This means that the flow rate is positive if it flows into the chamber.

Basic Assumptions and Limitations

- No inertia associated with pipe walls is taken into account.
- Chamber with compliant walls is assumed to have a cylindrical shape. Chamber with rigid wall can have any shape.

Dialog Box and Parameters



Block Parameters: Constant Volume Hydraulic Chamber

Constant Volume Hydraulic Chamber

The block represents a fixed-volume chamber with rigid or flexible walls used in hydraulic valves, pumps, manifolds, pipes, hoses, and so on. Use this block in models where you have to account for some form of flu compressibility. You can select the appropriate representation of fluid compressibility using the block parameters.

The block has one hydraulic conserving port associated with the chamber inlet. The block positive direction i from its port to the reference point. This means that the flow rate is positive if it flows into the chamber.

View source for Constant Volume Hydraulic Chamber

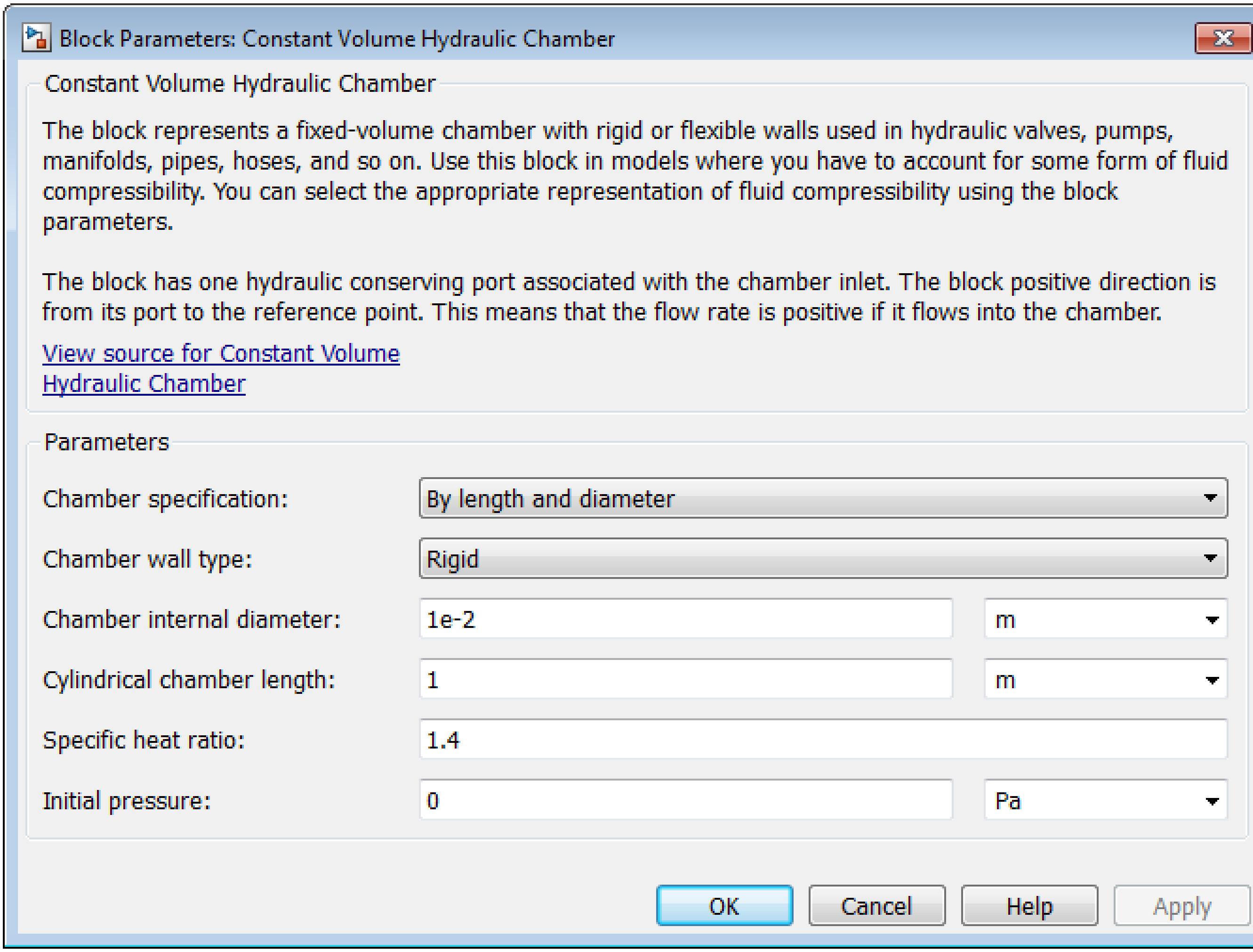
Parameters

Chamber specification: By volume

Chamber volume: 1e-4 m^3

Specific heat ratio: 1.4

Initial pressure: 0 Pa



Constant Volume Hydraulic Chamber

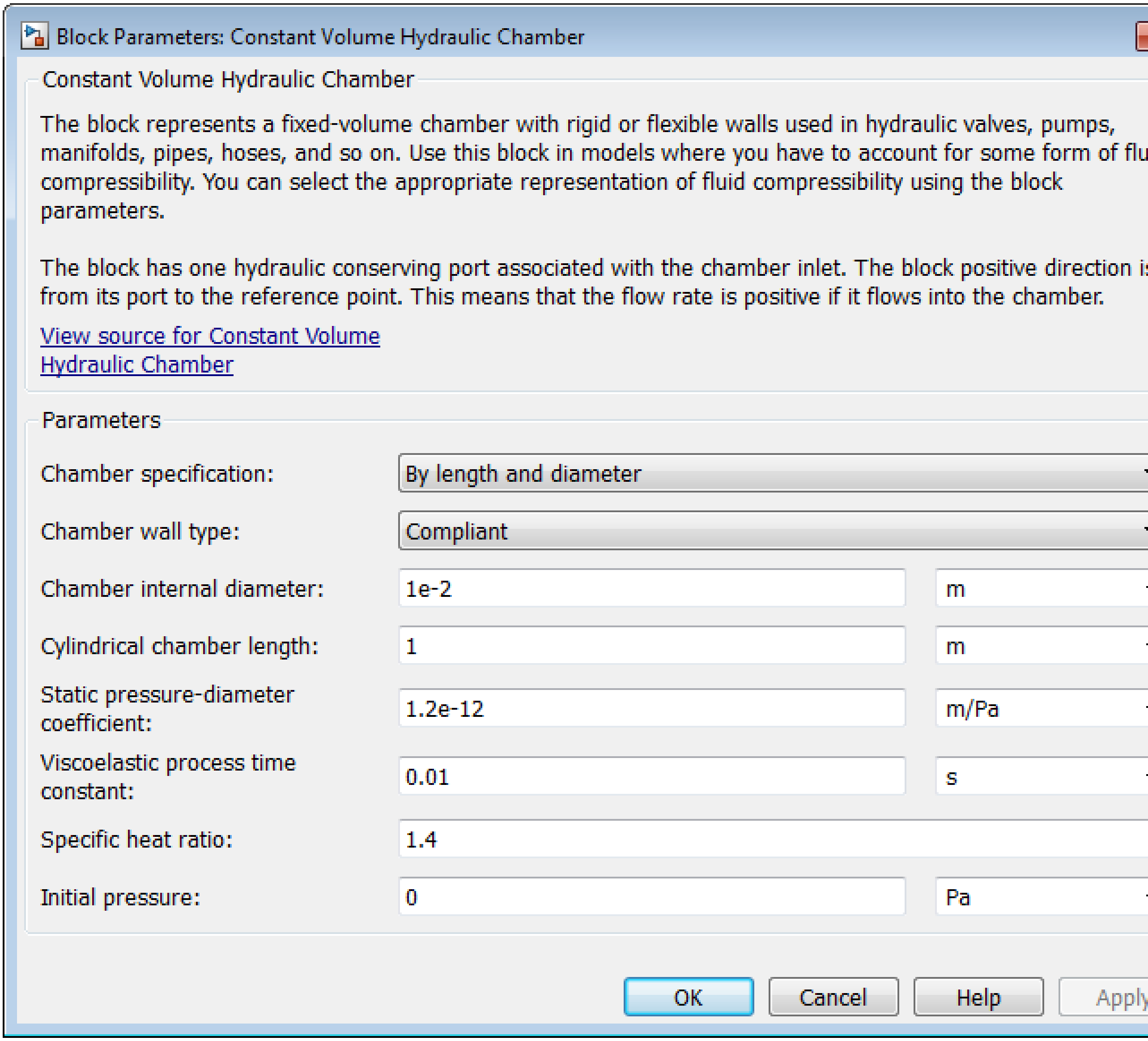
The block represents a fixed-volume chamber with rigid or flexible walls used in hydraulic valves, pumps, manifolds, pipes, hoses, and so on. Use this block in models where you have to account for some form of fluid compressibility. You can select the appropriate representation of fluid compressibility using the block parameters.

The block has one hydraulic conserving port associated with the chamber inlet. The block positive direction is from its port to the reference point. This means that the flow rate is positive if it flows into the chamber.

View source for Constant Volume Hydraulic Chamber

Parameters

Chamber specification:	By length and diameter	
Chamber wall type:	Rigid	
Chamber internal diameter:	1e-2	m
Cylindrical chamber length:	1	m
Specific heat ratio:	1.4	
Initial pressure:	0	Pa



Block Parameters: Constant Volume Hydraulic Chamber

Constant Volume Hydraulic Chamber

The block represents a fixed-volume chamber with rigid or flexible walls used in hydraulic valves, pumps, manifolds, pipes, hoses, and so on. Use this block in models where you have to account for some form of flu compressibility. You can select the appropriate representation of fluid compressibility using the block parameters.

The block has one hydraulic conserving port associated with the chamber inlet. The block positive direction i from its port to the reference point. This means that the flow rate is positive if it flows into the chamber.

View source for Constant Volume Hydraulic Chamber

Parameters

Chamber specification:	By length and diameter	
Chamber wall type:	Compliant	
Chamber internal diameter:	1e-2	m
Cylindrical chamber length:	1	m
Static pressure-diameter coefficient:	1.2e-12	m/Pa
Viscoelastic process time constant:	0.01	s
Specific heat ratio:	1.4	
Initial pressure:	0	Pa

Chamber specification

The parameter can have one of two values: `By volume` or `By length and diameter`. The value `By length and diameter` is recommended if a chamber is formed by a circular pipe. If the parameter is set to `By volume`, wall compliance is not taken into account. The default value of the parameter is `By volume`.

Chamber wall type

The parameter can have one of two values: `Rigid` or `Compliant`. If the parameter is set to `Rigid`, wall compliance is not taken into account, which can improve computational efficiency. The value `Compliant` is recommended for hoses and metal pipes, where compliance can affect the system behavior. The default value of the parameter is `Rigid`. The parameter is used if the **Chamber specification** parameter is set to `By length and diameter`.

Chamber volume

Volume of fluid in the chamber. The default value is `1e-4` m^3. The parameter is used if the **Chamber specification** parameter is set to `By volume`.

Chamber internal diameter

Internal diameter of the cylindrical chamber. The default value is `0.01` m. The parameter is used if the **Chamber specification** parameter is set to `By length and diameter`.

Cylindrical chamber length

Length of the cylindrical chamber. The default value is `1` m. The parameter is used if the **Chamber specification** parameter is set to `By length and diameter`.

Static pressure-diameter coefficient

Coefficient K_p that establishes relationship between pressure and the internal diameter at steady-state conditions. The parameter can be determined analytically or experimentally. The default value is `1.2e-12` m/Pa. The parameter is used if **Chamber wall type** is set to `Compliant`.

Viscoelastic process time constant

Time constant in the transfer function relating pipe internal diameter to pressure variations. With this parameter, the simulated elastic or viscoelastic process is approximated with the first-order lag. The parameter is determined experimentally or provided by the manufacturer. The default value is `0.01` s. The parameter is used if **Chamber wall type** is set to `Compliant`.

Specific heat ratio

Gas-specific heat ratio. The default value is `1.4`.

Initial pressure

Initial pressure in the chamber. This parameter specifies the initial condition for use in computing the block's initial state at the beginning of a simulation run. For more information, see "Initial Conditions Computation". The default value is `0`.

Restricted Parameters

When your model is in Restricted editing mode, you cannot modify the following parameters:

- **Chamber specification**
- **Chamber wall type**

All other block parameters are available for modification. The actual set of modifiable block parameters depends on the values of the **Tube cross section type** and **Chamber wall type** parameters at the time the model entered Restricted mode.

Global Parameters

Parameters determined by the type of working fluid:

- **Fluid bulk modulus**
- **Nondissolved gas ratio** — Nondissolved gas relative content determined as a ratio of gas volume to the liquid volume.

Use the Hydraulic Fluid block or the Custom Hydraulic Fluid block to specify the fluid properties.

Ports

The block has one hydraulic conserving port associated with the chamber inlet.

References

[1] Manring, N.D., *Hydraulic Control Systems*, John Wiley & Sons, New York, 2005

[2] Meritt, H.E., *Hydraulic Control Systems*, John Wiley & Sons, New York, 1967

[3] Holcke, Jan, *Frequency Response of Hydraulic Hoses*, RIT, FTH, Stockholm, 2002

See Also

Hydraulic Piston Chamber

Variable Hydraulic Chamber

Constant Volume Pneumatic Chamber

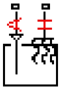
Purpose

Constant volume pneumatic chamber based on ideal gas law

Library

Pneumatic Elements

Description



The Constant Volume Pneumatic Chamber block models a constant volume pneumatic chamber based on the ideal gas law and assuming constant specific heats.

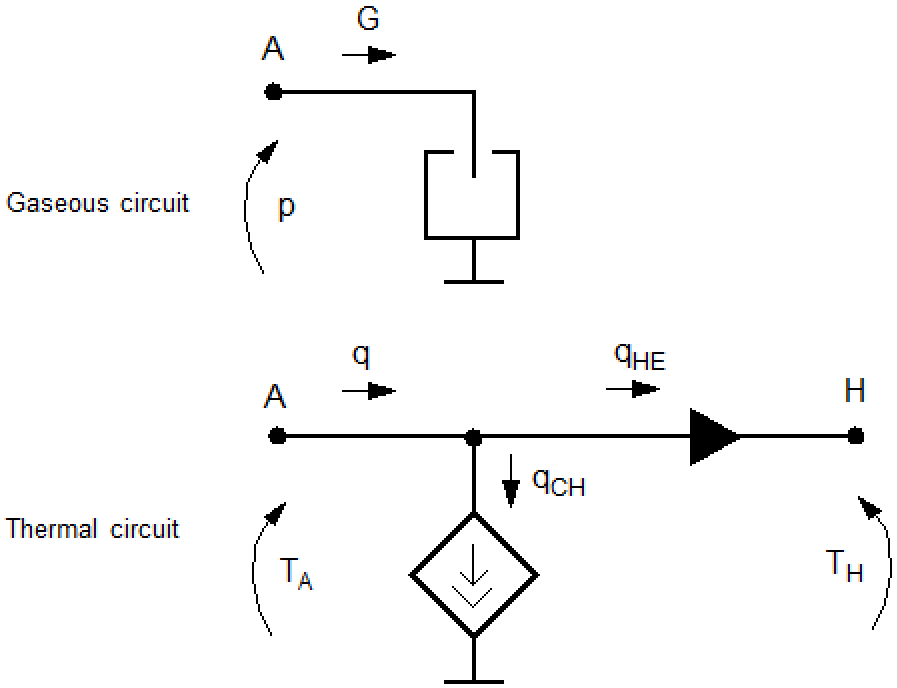
The continuity equation for the network representation of the constant chamber is

$$G = \frac{V}{RT}\left(\frac{dp}{dt} - \frac{p}{T}\frac{dT}{dt}\right)$$

where

G	Mass flow rate at input port
V	Chamber volume
p	Absolute pressure in the chamber
R	Specific gas constant
T	Absolute gas temperature
t	Time

The equivalent circuit of the Constant Volume Pneumatic Chamber block model is shown in the following illustration. Port A is the pneumatic conserving port associated with the chamber inlet. Port A connects both to the gaseous and the thermal circuit. Port H is a thermal conserving port through which heat exchange with the environment takes place. Port H connects only to the thermal circuit.



The diagram shows that the heat flow *q* to the chamber consists of two components:

- Heat flow q_{CH}, associated with the gaseous process
- Heat flow q_{HE}, associated with the heat exchange with the environment

The heat flow due to gas inflow is

$$q_{CH} = \frac{c_v V}{R} \frac{dp}{dt}$$

where c_{v} is specific heat at constant volume.

The heat exchange with the environment happens through port H, connected to thermal components. To determine the value of the heat exchange flow, the model contains a short-circuit element, resulting in the equation

$$T_{\mathrm{A}} = T_{\mathrm{H}}$$

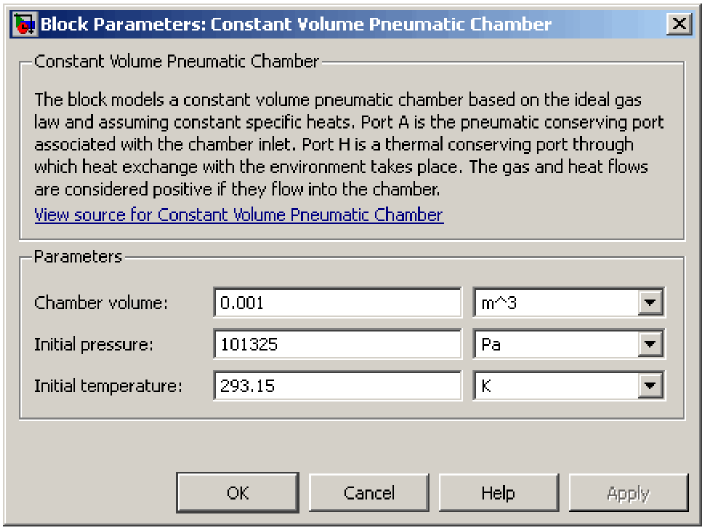
where both T_{A} and T_{H} represent the gas temperature.

The gas flow and the heat flow are considered positive if they flow into the chamber.

Basic Assumptions and Limitations

- The gas is ideal.
- Specific heats at constant pressure and constant volume, c_p and c_v, are constant.

Dialog Box and Parameters



Block Parameters: Constant Volume Pneumatic Chamber

Constant Volume Pneumatic Chamber

The block models a constant volume pneumatic chamber based on the ideal gas law and assuming constant specific heats. Port A is the pneumatic conserving port associated with the chamber inlet. Port H is a thermal conserving port through which heat exchange with the environment takes place. The gas and heat flows are considered positive if they flow into the chamber.

View source for Constant Volume Pneumatic Chamber

Parameters

Chamber volume: 0.001 m^3

Initial pressure: 101325 Pa

Initial temperature: 293.15 K

Chamber volume

Specify the volume of the chamber. The default value is `.001` m^3.

Initial pressure

Specify the initial pressure in the chamber. This parameter specifies the initial condition for use in computing the initial state at the beginning of a simulation run. For more information, see "Initial Conditions Computation". The default value is `101235` Pa.

Initial temperature

Specify the initial temperature of the gas in the chamber. This parameter specifies the initial condition for use in computing the initial state at the beginning of a simulation run. For more information, see "Initial Conditions Computation". The default value is `293.15` K.

Ports

The block has the following ports:

`A`

Pneumatic conserving port associated with the chamber inlet.

`H`

Thermal conserving port through which heat exchange with the environment takes place.

See Also

Pneumatic Piston Chamber

Rotary Pneumatic Piston Chamber

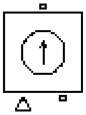
Purpose

Ideal current source driven by input signal

Library

Electrical Sources

Description

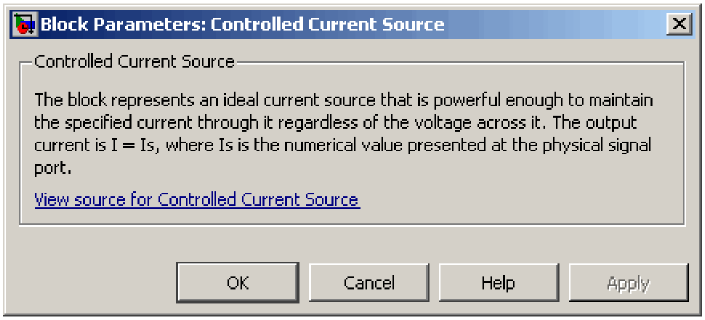


The Controlled Current Source block represents an ideal current source that is powerful enough to maintain the specified current through it regardless of the voltage across the source.

The output current is *I* = *Is*, where *Is* is the numerical value presented at the physical signal port.

The positive direction of the current flow is indicated by the arrow.

Dialog Box and Parameters



Controlled Current Source

The block represents an ideal current source that is powerful enough to maintain the specified current through it regardless of the voltage across it. The output current is I = Is, where Is is the numerical value presented at the physical signal port.

View source for Controlled Current Source

The block has no parameters.

Ports

The block has one physical signal input port and two electrical conserving ports associated with its electrical terminals.

See Also

Controlled Voltage Source

Controlled Flux Source

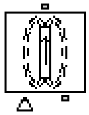
Purpose

Ideal flux source driven by input signal

Library

Magnetic Sources

Description

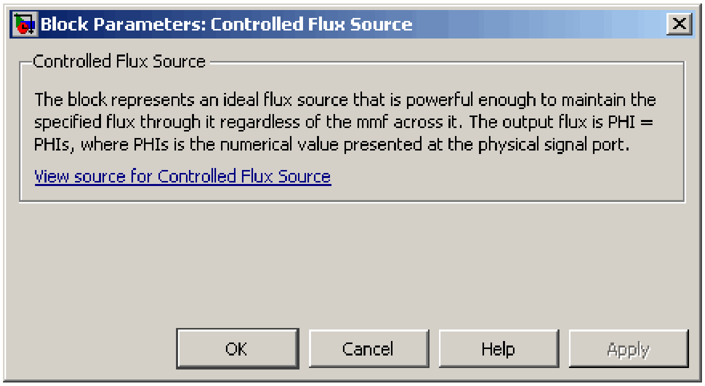


The Controlled Flux Source block represents an ideal flux source that is powerful enough to maintain the specified flux through it regardless of the mmf across the source.

The output flux is *PHI* = *PHIs*, where *PHIs* is the numerical value presented at the physical signal port.

The positive direction of the flux flow is indicated by the arrow.

Dialog Box and Parameters



The block represents an ideal flux source that is powerful enough to maintain the specified flux through it regardless of the mmf across it. The output flux is PHI = PHIs, where PHIs is the numerical value presented at the physical signal port.

View source for Controlled Flux Source

The block has no parameters.

Ports

The block has one physical signal input port and two magnetic conserving ports associated with its magnetic terminals.

See Also

Controlled MMF Source

Flux Source

MMF Source

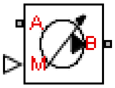
Purpose

Ideal compressor that generates a variable mass flow rate through its inlets

Library

Thermal Liquid/Sources

Description



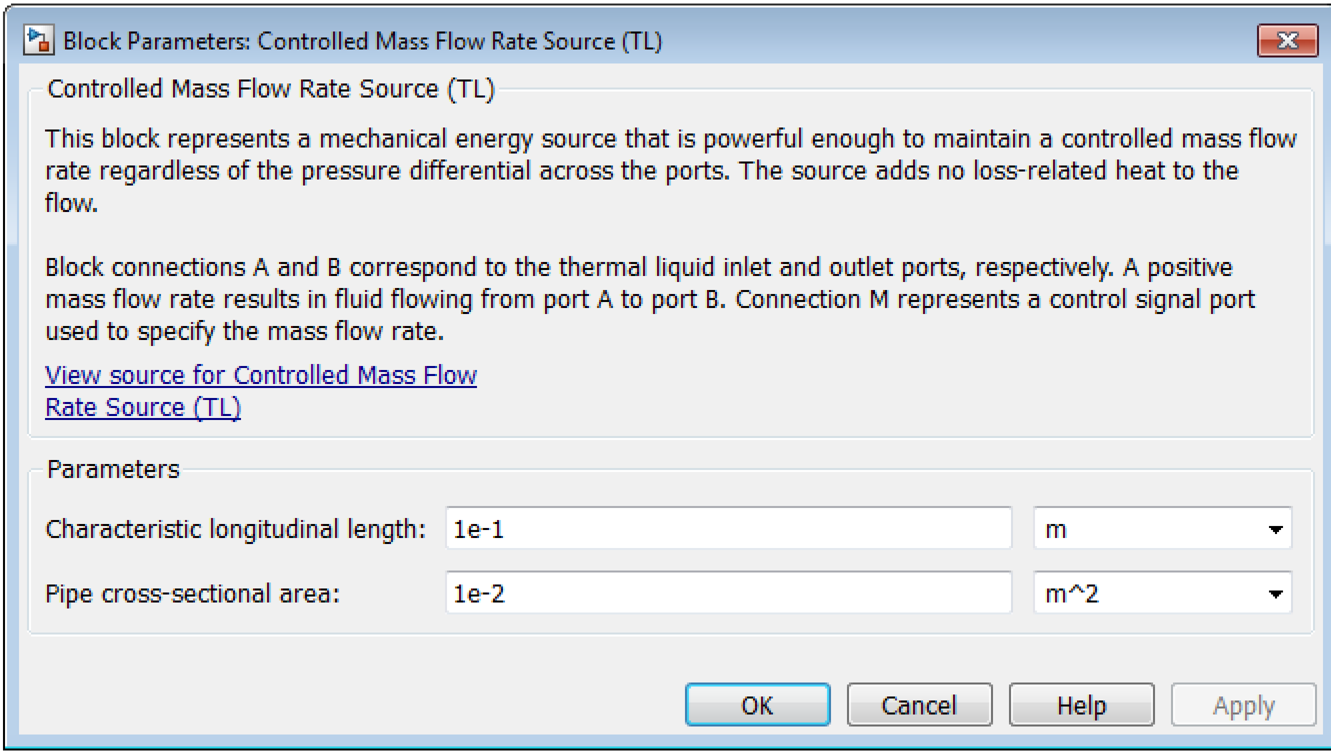
The Controlled-Mass Flow Rate Source (TL) block represents an ideal compressor generating a variable mass flow rate through its inlets. The compressor can hold the specified mass flow rate regardless of the pressure differential across its inlets. No heat exchange occurs between the compressor and the flowing liquid.

Ports A and B represent the compressor inlets. Both ports are thermal liquid conserving ports. The mass flow rate is positive when liquid flows from inlet A to inlet B. The block provides physical signal port M so that you can specify the mass flow rate through the compressor inlets.

Assumptions and Limitations

- No heat exchange occurs between the compressor and the flowing liquid.
- The pressure differential between the compressor inlets is independent of the mass flow rate through those inlets.

Dialog Box and Parameters



Block Parameters: Controlled Mass Flow Rate Source (TL)

Controlled Mass Flow Rate Source (TL)

This block represents a mechanical energy source that is powerful enough to maintain a controlled mass flow rate regardless of the pressure differential across the ports. The source adds no loss-related heat to the flow.

Block connections A and B correspond to the thermal liquid inlet and outlet ports, respectively. A positive mass flow rate results in fluid flowing from port A to port B. Connection M represents a control signal port used to specify the mass flow rate.

View source for Controlled Mass Flow Rate Source (TL)

Parameters

Characteristic longitudinal length: 1e-1 m

Pipe cross-sectional area: 1e-2 m^2

Characteristic longitudinal length

Enter the mean path length the liquid must flow through to go from inlet A to inlet B. The default value is `1e-1` m.

Pipe cross-sectional area

Enter the cross-sectional area of the adjoining pipes. The default value is `1e-2` m^2.

Ports

The block has the following ports.

A	Thermal liquid conserving port associated with compressor inlet A
B	Thermal liquid conserving port associated with compressor inlet B
M	Physical signal port providing the mass flow rate between inlets A and B

See Also

Pressure Source (TL)

Controlled Pressure Source (TL)

Mass Flow Rate Source (TL)

Controlled MMF Source

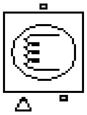
Purpose

Ideal magnetomotive force source driven by input signal

Library

Magnetic Sources

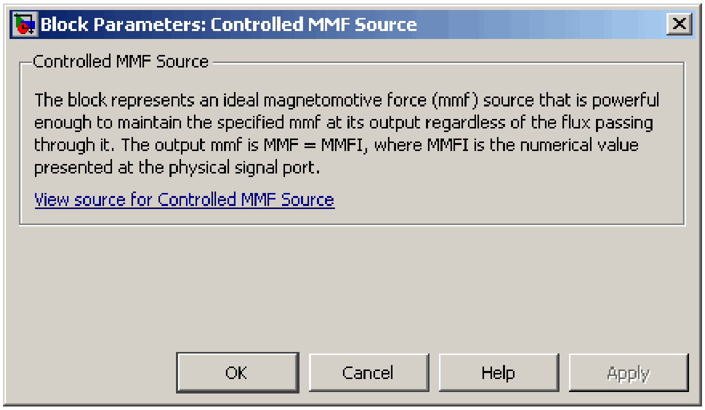
Description



The Controlled MMF Source block represents an ideal magnetomotive force (mmf) source that is powerful enough to maintain the specified mmf at its output regardless of the flux passing through it.

The output mmf is $MMF = MMFI$, where $MMFI$ is the numerical value presented at the physical signal port.

Dialog Box and Parameters



Block Parameters: Controlled MMF Source

Controlled MMF Source

The block represents an ideal magnetomotive force (mmf) source that is powerful enough to maintain the specified mmf at its output regardless of the flux passing through it. The output mmf is MMF = MMFI, where MMFI is the numerical value presented at the physical signal port.

View source for Controlled MMF Source

The block has no parameters.

Ports

The block has one physical signal input port and two magnetic conserving ports associated with its magnetic terminals.

See Also

Controlled Flux Source

Flux Source

MMF Source

Controlled Pneumatic Flow Rate Source

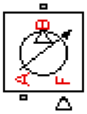
Purpose

Ideal compressor with signal-controlled mass flow rate

Library

Pneumatic Sources

Description



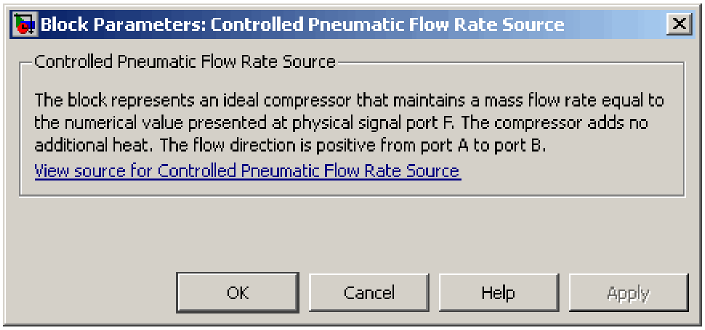
The Controlled Pneumatic Flow Rate Source block represents an ideal compressor that maintains a mass flow rate equal to the numerical value presented at physical signal port F. The compressor adds no heat. Block connections A and B correspond to the pneumatic inlet and outlet ports, respectively, and connection F represents a control signal port.

The block positive direction is from port A to port B. This means that the flow rate is positive if it flows from A to B. The pressure differential is determined as $p = p_A - p_B$ and is negative if pressure at the source outlet is greater than pressure at its inlet. The power generated by the source is negative if the source adds energy to the flow.

Warning

Be careful when driving an orifice directly from a flow rate source. The choked flow condition limits the flow that is possible through an orifice as a function of upstream pressure and temperature. Hence the flow rate value produced by the flow rate source must be compatible with upstream pressure and temperature. Specifying a flow rate that is too high will result in an unsolvable set of equations.

Dialog Box and Parameters


Block Parameters: Controlled Pneumatic Flow Rate Source
Controlled Pneumatic Flow Rate Source

The block represents an ideal compressor that maintains a mass flow rate equal to the numerical value presented at physical signal port F. The compressor adds no additional heat. The flow direction is positive from port A to port B.

View source for Controlled Pneumatic Flow Rate Source

The block has no parameters.

Ports

The block has the following ports:

A
: Pneumatic conserving port associated with the source inlet.

B
: Pneumatic conserving port associated with the source outlet.

F
: Control signal port.

See Also

Pneumatic Flow Rate Source

Pneumatic Mass & Heat Flow Sensor

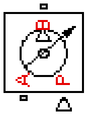
Purpose

Ideal compressor with signal-controlled pressure difference

Library

Pneumatic Sources

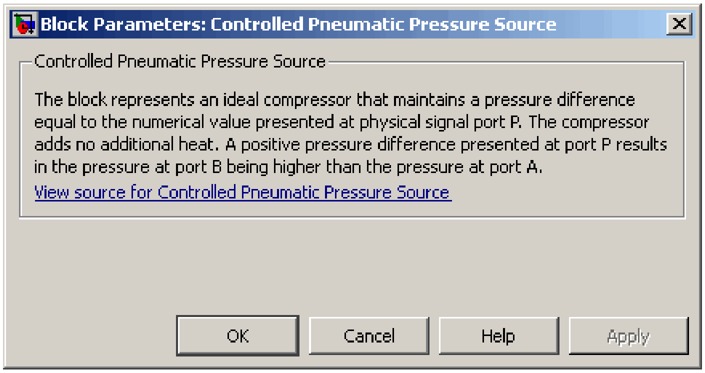
Description



The Controlled Pneumatic Pressure Source block represents an ideal compressor that maintains a pressure difference equal to the numerical value presented at physical signal port F. The compressor adds no heat. Block connections A and B correspond to the pneumatic inlet and outlet ports, respectively, and connection F represents a control signal port.

A positive pressure difference results in the pressure at port B being higher than the pressure at port A.

Dialog Box and Parameters



Block Parameters: Controlled Pneumatic Pressure Source

Controlled Pneumatic Pressure Source

The block represents an ideal compressor that maintains a pressure difference equal to the numerical value presented at physical signal port P. The compressor adds no additional heat. A positive pressure difference presented at port P results in the pressure at port B being higher than the pressure at port A.

View source for Controlled Pneumatic Pressure Source

The block has no parameters.

Ports

The block has the following ports:

A

Pneumatic conserving port associated with the source inlet.

B

Pneumatic conserving port associated with the source outlet.

F

Control signal port.

See Also

Pneumatic Pressure Source

Pneumatic Pressure & Temperature Sensor

Controlled Pressure Source (TL)

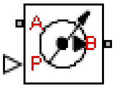
Purpose

Ideal compressor that generates a variable pressure differential across its inlets

Library

Thermal Liquid/Sources

Description



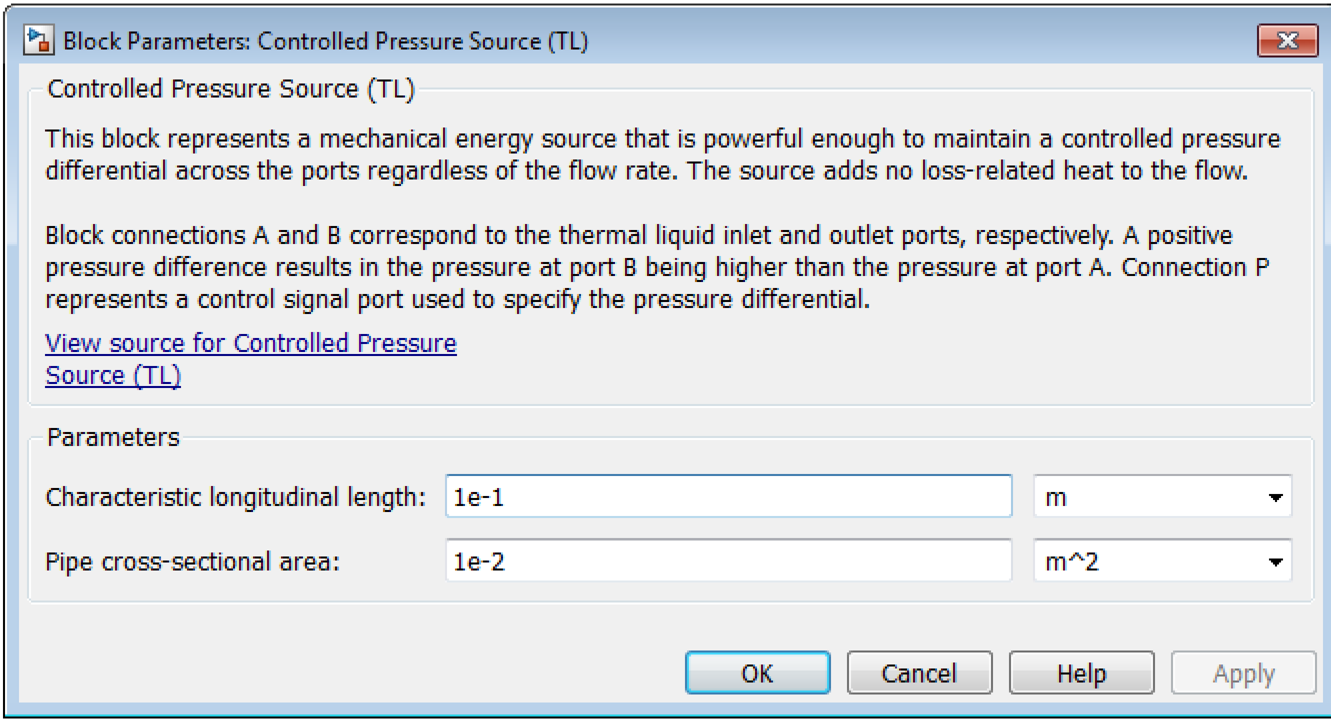
The Controlled-Pressure Source (TL) block represents an ideal compressor generating a variable pressure differential across its inlets. The compressor can hold the specified pressure differential regardless of its mass flow rate. No heat exchange occurs between the compressor and the flowing liquid.

Ports A and B represent the compressor inlets. Both ports are thermal liquid conserving ports. The pressure differential is positive when the pressure at inlet B is greater than the pressure at inlet A. The block provides physical signal port P so that you can specify the pressure differential across the compressor inlets.

Assumptions and Limitations

- No heat exchange occurs between the compressor and the flowing liquid.
- The pressure differential between the compressor inlets is independent of the mass flow rate through those inlets.

Dialog Box and Parameters



Block Parameters: Controlled Pressure Source (TL)

Controlled Pressure Source (TL)

This block represents a mechanical energy source that is powerful enough to maintain a controlled pressure differential across the ports regardless of the flow rate. The source adds no loss-related heat to the flow.

Block connections A and B correspond to the thermal liquid inlet and outlet ports, respectively. A positive pressure difference results in the pressure at port B being higher than the pressure at port A. Connection P represents a control signal port used to specify the pressure differential.

View source for Controlled Pressure Source (TL)

Parameters

Characteristic longitudinal length: 1e-1 m

Pipe cross-sectional area: 1e-2 m^2

Characteristic longitudinal length

Enter the mean path length the liquid must flow through to go from inlet A to inlet B. The default value is `1e-1` m.

Pipe cross-sectional area

Enter the cross-sectional area of the adjoining pipes. The default value is `1e-2` m^2.

Ports

The block has the following ports.

A Thermal liquid conserving port associated with compressor inlet A

B Thermal liquid conserving port associated with compressor inlet B

P Physical signal port providing the pressure differential between inlets A and B

See Also

Pressure Source (TL)

Mass Flow Rate Source (TL)

Controlled Mass Flow Rate Source (TL)

Controlled Temperature Reservoir (TL)

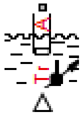
Purpose

Infinite open reservoir at variable temperature

Library

Thermal Liquid/Elements

Description



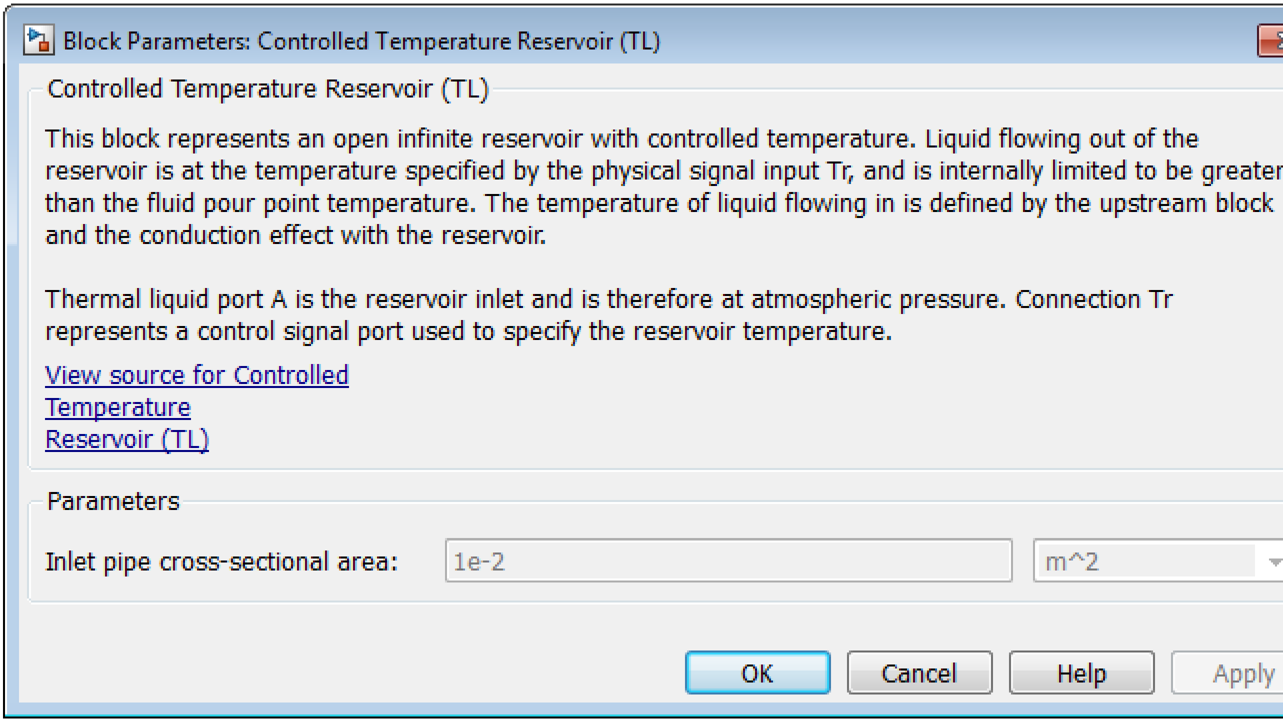
The Controlled-Temperature Reservoir (TL) block represents an infinite open reservoir at variable temperature. Because it is open, the reservoir and its inlet are at atmospheric pressure. Port A, a thermal liquid conserving port, represents the reservoir inlet. Port Tr, a physical signal port, provides the reservoir temperature control signal.

The inlet temperature depends on the direction of liquid flow. If the liquid flows into the reservoir, the inlet temperature equals that of the upstream liquid. The reservoir acts as a heat sink. If liquid flows out of the reservoir, the inlet temperature equals that of the reservoir. The reservoir acts as a heat source.

To ensure a smooth temperature change at the reservoir inlet during liquid flow reversal, the block includes heat conduction along a length equal to the effective diameter of the pipe. This diameter is a function of the specified cross-sectional area of the inlet pipe.

This block also functions as a reference point for pressure and temperature measurements in a pipe network. These measurements are relative to atmospheric pressure and reservoir temperature, respectively.

Dialog Box and Parameters



Block Parameters: Controlled Temperature Reservoir (TL)

Controlled Temperature Reservoir (TL)

This block represents an open infinite reservoir with controlled temperature. Liquid flowing out of the reservoir is at the temperature specified by the physical signal input Tr, and is internally limited to be greater than the fluid pour point temperature. The temperature of liquid flowing in is defined by the upstream block and the conduction effect with the reservoir.

Thermal liquid port A is the reservoir inlet and is therefore at atmospheric pressure. Connection Tr represents a control signal port used to specify the reservoir temperature.

View source for Controlled Temperature Reservoir (TL)

Parameters

Inlet pipe cross-sectional area: 1e-2 m^2

Inlet pipe cross-sectional area

Enter the cross-sectional area of the reservoir inlet pipe. The block uses this area to determine the characteristic length of the pipe along which heat conduction occurs. The default value is `1e-2` m^2.

Ports

The block has one thermal liquid conserving port, A, and one physical signal port, Tr.

See Also

Temperature Reservoir (TL)

Constant Volume Chamber (TL)

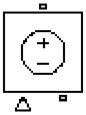
Purpose

Ideal voltage source driven by input signal

Library

Electrical Sources

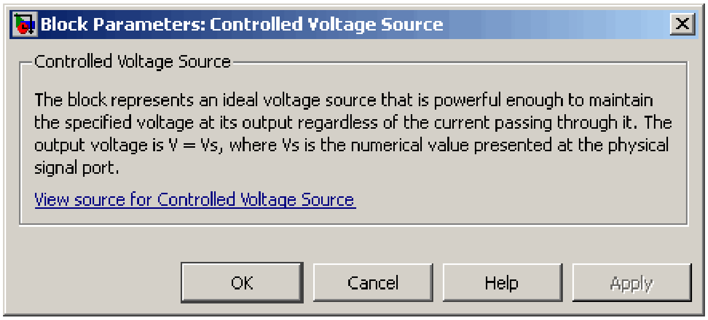
Description



The Controlled Voltage Source block represents an ideal voltage source that is powerful enough to maintain the specified voltage at its output regardless of the current flowing through the source.

The output voltage is $V = Vs$, where Vs is the numerical value presented at the physical signal port.

Dialog Box and Parameters



Block Parameters: Controlled Voltage Source
Controlled Voltage Source

The block represents an ideal voltage source that is powerful enough to maintain the specified voltage at its output regardless of the current passing through it. The output voltage is V = Vs, where Vs is the numerical value presented at the physical signal port.

View source for Controlled Voltage Source
OK
Cancel
Help
Apply

The block has no parameters.

Ports

The block has one physical signal input port and two electrical conserving ports associated with its electrical terminals.

See Also

Controlled Current Source

Convective Heat Transfer

Purpose

Heat transfer by convection

Library

Thermal Elements

Description

The Convective Heat Transfer block represents a heat transfer by convection between two bodies by means of fluid motion. The transfer is governed by the Newton law of cooling and is described with the following equation:

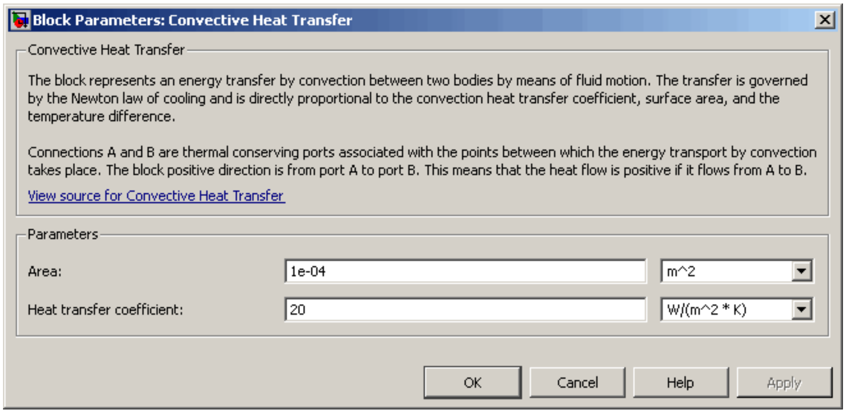
$$Q = k \cdot A \cdot (T_A - T_B)$$

where

Q	Heat flow
k	Convection heat transfer coefficient
A	Surface area
T_A, T_B	Temperatures of the bodies

Connections A and B are thermal conserving ports associated with the points between which the heat transfer by convection takes place. The block positive direction is from port A to port B. This means that the heat flow is positive if it flows from A to B.

Dialog Box and Parameters



Block Parameters: Convective Heat Transfer

Convective Heat Transfer

The block represents an energy transfer by convection between two bodies by means of fluid motion. The transfer is governed by the Newton law of cooling and is directly proportional to the convection heat transfer coefficient, surface area, and the temperature difference.

Connections A and B are thermal conserving ports associated with the points between which the energy transport by convection takes place. The block positive direction is from port A to port B. This means that the heat flow is positive if it flows from A to B.

View source for Convective Heat Transfer

Parameters

Area: 1e-04 m^2

Heat transfer coefficient: 20 W/(m^2 * K)

Area

Surface area of heat transfer. The default value is `0.0001` m^2.

Heat transfer coefficient

Convection heat transfer coefficient. The default value is `20` W/m^2/K.

Ports

The block has the following ports:

A

Thermal conserving port associated with body A.

B

Thermal conserving port associated with body B.

See Also

Conductive Heat Transfer

Radiative Heat Transfer

Counter

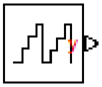
Purpose

Increment output signal by 1 with every time step

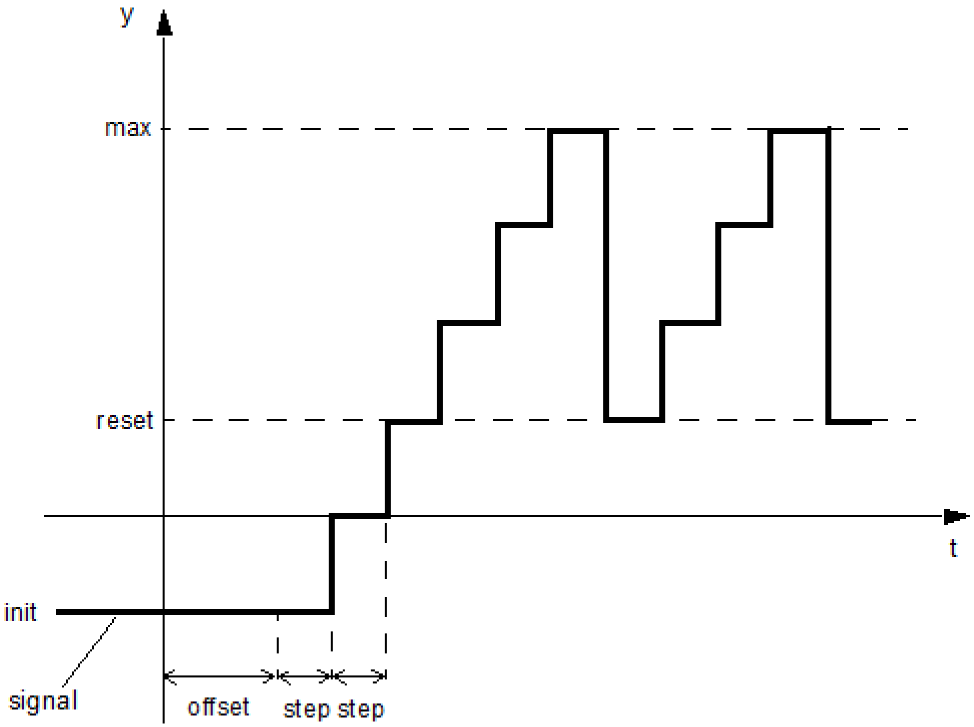
Library

Physical Signals/Sources

Description



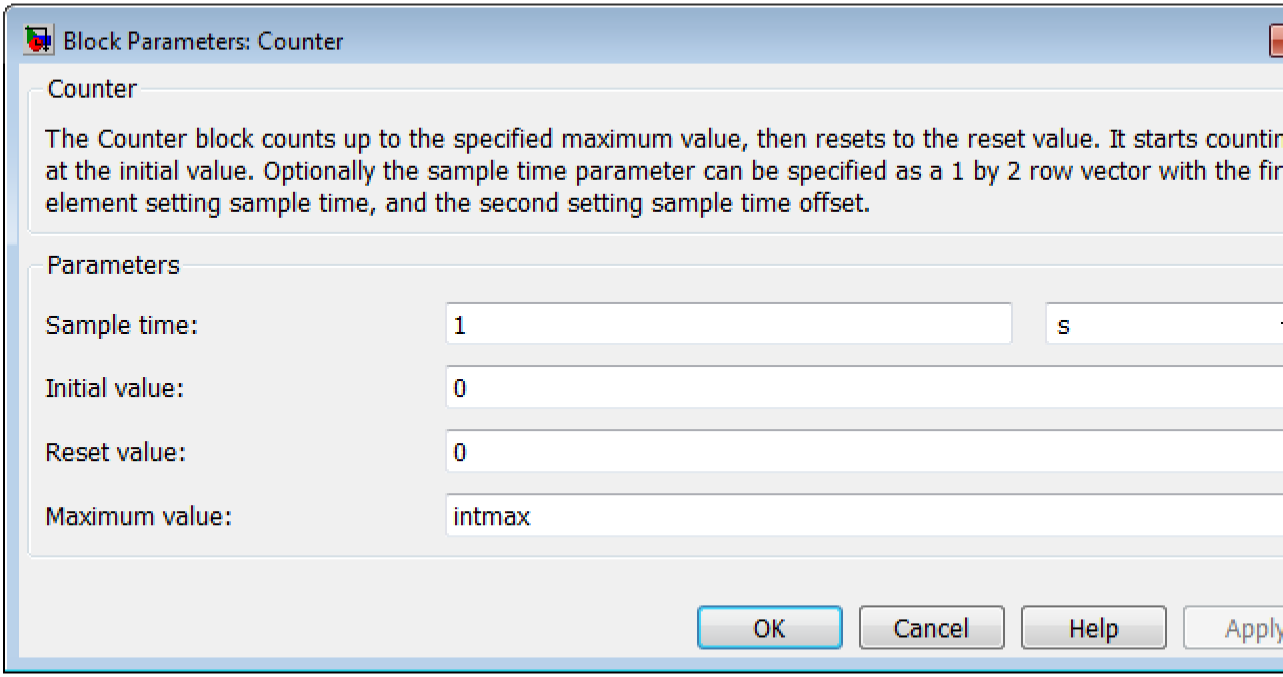
The Counter block increments the output signal, *y*, by 1 with every time step repeatedly between the minimum (reset) value and the maximum value. You can optionally specify an initial signal value, different from the reset value, and an initial time offset. The output signal generated by the block is shown in the following diagram.



If the initial time offset is specified, the block outputs the initial signal value *init* until the simulation time reaches the *offset* value, at which point the counting cycle starts. The block outputs the current value for one time step, then repeatedly increments the signal value by 1 and outputs it for one time step, until it reaches the maximum value *`max`*. The block outputs the *`max`* value for one time step, then returns to the *`reset`* value, and the counting cycle starts again.

Use this block, in conjunction with other physical signal blocks, to model discrete behaviors.

Dialog Box and Parameters



Counter

The Counter block counts up to the specified maximum value, then resets to the reset value. It starts countin at the initial value. Optionally the sample time parameter can be specified as a 1 by 2 row vector with the fir element setting sample time, and the second setting sample time offset.

Parameters

Sample time: 1 s

Initial value: 0

Reset value: 0

Maximum value: intmax

Sample time

The value of the time *`step`* interval. The default *`step`* value is `1` s. To specify an initial time offset, enter the parameter value as `[` *`step`* `,` *`offset`* `]`, otherwise the *`offset`* value is assumed to be `0`.

Initial value

The value of the output signal at the beginning of the first counting cycle. If you specify an initial time offset by using the **Sample time** parameter, the output of the block remains at this value until the simulation time reaches the *`offset`* value, after which the first counting cycle starts. The value must be an integer. The default value is `0`.

Reset value

The value of the output signal at the beginning of each counting cycle except the first one. The output of the block remains at this value for one time *`step`*, specified by the **Sample time** parameter. The value must be an integer. The default value is `0`.

Maximum value

The value of the output signal at the end of the counting cycle. The output of the block remains at this value for one time *`step`*, specified by the **Sample time** parameter, at which point the signal returns to the **Reset value** and the cycle starts again. The value must be an integer. The default value is `intmax` (2147483647, the largest positive value that can be represented in the MATLAB® software with a 32-bit integer).

Ports

The block has one physical signal output port.

Examples

The Discrete-Time PWM Voltage Source example illustrates how you can use the Counter block to build components with more complex behaviors. For an alternative asynchronous implementation, see the Asynchronous PWM Voltage Source example. The discrete-time version is better suited to fixed-step solvers and hardware-in-the-loop applications, whereas the asynchronous implementation is better suited to fast desktop simulation using variable-step solvers.

See Also

Asynchronous Sample & Hold

Current-Controlled Current Source

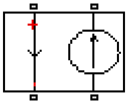
Purpose

Linear current-controlled current source

Library

Electrical Sources

Description



The Current-Controlled Current Source block models a linear current-controlled current source, described with the following equation:

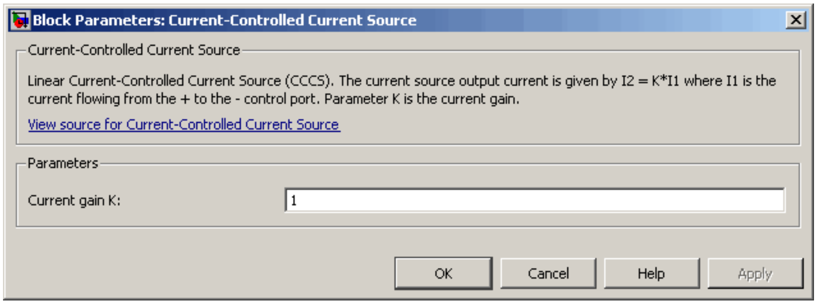
$$I2 = K \cdot I1$$

where

$I2$	Output current
K	Current gain
$I1$	Current flowing from the + to the – control port

To use the block, connect the + and – ports on the left side of the block (the control ports) to the control current source. The arrow between these ports indicates the positive direction of the control current flow. The two ports on the right side of the block (the output ports) generate the output current, with the arrow between them indicating the positive direction of the output current flow.

Dialog Box and Parameters



Block Parameters: Current-Controlled Current Source

Current-Controlled Current Source

Linear Current-Controlled Current Source (CCCS). The current source output current is given by I2 = K*I1 where I1 is the current flowing from the + to the - control port. Parameter K is the current gain.

View source for Current-Controlled Current Source

Parameters

Current gain K: 1

Current gain K

Ratio of the current between the two output terminals to the current passing between the two control terminals. The default value is 1.

Ports

The block has four electrical conserving ports. Connections + and – on the left side of the block are the control ports. The other two ports are the electrical terminals that provide the output current. The arrows between each pair of ports indicate the positive direction of the current flow.

See Also

Current-Controlled Voltage Source

Voltage-Controlled Current Source

Voltage-Controlled Voltage Source

Current-Controlled Voltage Source

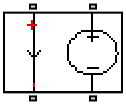
Purpose

Linear current-controlled voltage source

Library

Electrical Sources

Description



The Current-Controlled Voltage Source block models a linear current-controlled voltage source, described with the following equation:

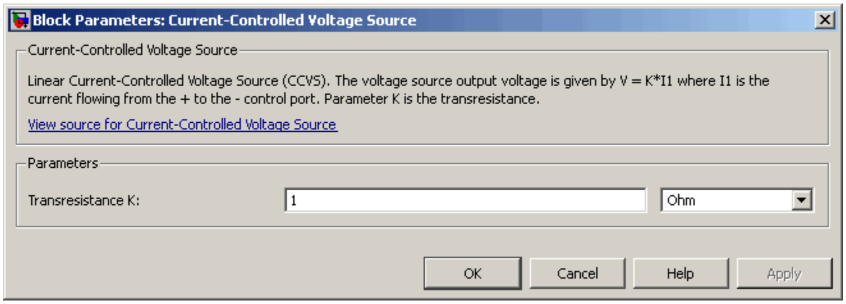
$$V = K \cdot I1$$

where

V	Voltage
K	Transresistance
I1	Current flowing from the + to the – control port

To use the block, connect the + and – ports on the left side of the block (the control ports) to the control current source. The arrow indicates the positive direction of the current flow. The two ports on the right side of the block (the output ports) generate the output voltage. Polarity is indicated by the + and – signs.

Dialog Box and Parameters



Block Parameters: Current-Controlled Voltage Source

Current-Controlled Voltage Source

Linear Current-Controlled Voltage Source (CCVS). The voltage source output voltage is given by V = K*I1 where I1 is the current flowing from the + to the - control port. Parameter K is the transresistance.

View source for Current-Controlled Voltage Source

Parameters

Transresistance K: 1 Ohm

Transresistance K

Ratio of the voltage between the two output terminals to the current passing between the two control terminals. The default value is 1 Ω.

Ports

The block has four electrical conserving ports. Connections + and – on the left side of the block are the control ports. The arrow indicates the positive direction of the current flow. The other two ports are the electrical terminals that provide the output voltage. Polarity is indicated by the + and – signs.

See Also

Current-Controlled Current Source

Voltage-Controlled Current Source

Voltage-Controlled Voltage Source

Current Sensor

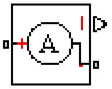
Purpose

Current sensor in electrical systems

Library

Electrical Sensors

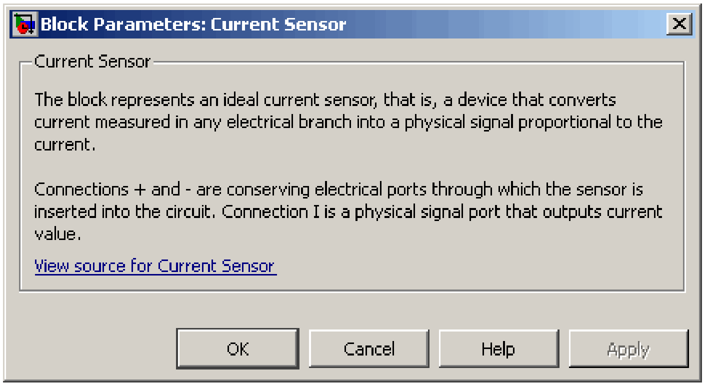
Description



The Current Sensor block represents an ideal current sensor, that is, a device that converts current measured in any electrical branch into a physical signal proportional to the current.

Connections + and – are electrical conserving ports through which the sensor is inserted into the circuit. Connection I is a physical signal port that outputs the measurement result.

Dialog Box and Parameters


Block Parameters: Current Sensor

Current Sensor

The block represents an ideal current sensor, that is, a device that converts current measured in any electrical branch into a physical signal proportional to the current.

Connections + and - are conserving electrical ports through which the sensor is inserted into the circuit. Connection I is a physical signal port that outputs current value.

View source for Current Sensor

The block has no parameters.

Ports

The block has the following ports:

+

Electrical conserving port associated with the sensor positive terminal.

-

Electrical conserving port associated with the sensor negative terminal.

`I`

Physical signal output port for current.

See Also

Voltage Sensor

Custom Hydraulic Fluid

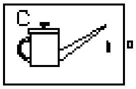
Purpose

Working fluid properties, set by specifying parameter values

Library

Hydraulic Utilities

Description



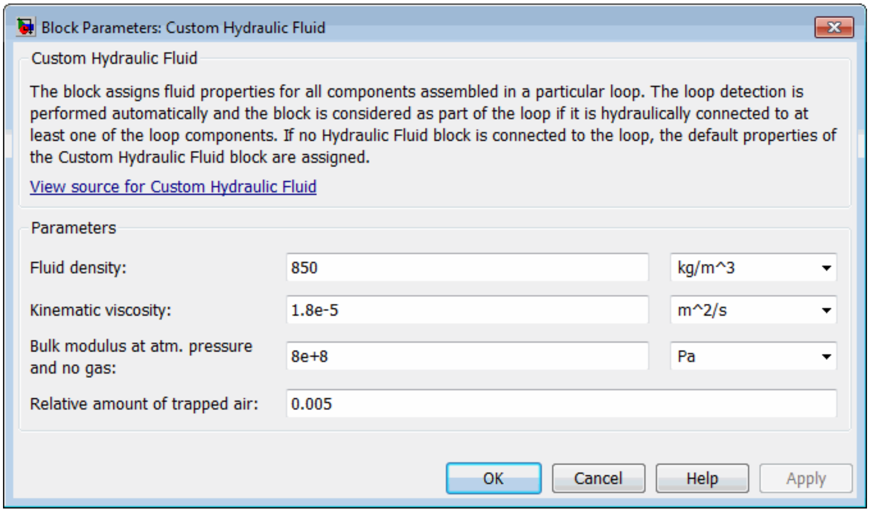
The Custom Hydraulic Fluid block lets you specify the type of hydraulic fluid used in a loop of hydraulic blocks. It provides the hydraulic fluid properties, such as kinematic viscosity, density, and bulk modulus, for all the hydraulic blocks in the loop. These fluid properties are assumed to be constant during simulation time.

The Custom Hydraulic Fluid block lets you specify the fluid properties, such as kinematic viscosity, density, bulk modulus, and relative amount of entrapped air, as block parameters.

The Custom Hydraulic Fluid block has one port. You can connect it to a hydraulic diagram by branching a connection line off the main line and connecting it to the port. When you connect the Custom Hydraulic Fluid block to a hydraulic line, the software automatically identifies the hydraulic blocks connected to the particular loop and propagates the hydraulic fluid properties to all the hydraulic blocks in the loop.

Each topologically distinct hydraulic loop in a diagram requires the properties of the working fluid to be specified. You can specify these properties by using either a Custom Hydraulic Fluid block or a Hydraulic Fluid block, which is available with SimHydraulics® libraries. If no Hydraulic Fluid block or Custom Hydraulic Fluid block is attached to a loop, the hydraulic blocks in this loop use the default fluid, which is equivalent to fluid defined by a Custom Hydraulic Fluid block with the default parameter values.

Dialog Box and Parameters



Custom Hydraulic Fluid

The block assigns fluid properties for all components assembled in a particular loop. The loop detection is performed automatically and the block is considered as part of the loop if it is hydraulically connected to at least one of the loop components. If no Hydraulic Fluid block is connected to the loop, the default properties of the Custom Hydraulic Fluid block are assigned.

View source for Custom Hydraulic Fluid

Parameters

Fluid density:	850	kg/m^3
Kinematic viscosity:	1.8e-5	m^2/s
Bulk modulus at atm. pressure and no gas:	8e+8	Pa
Relative amount of trapped air:	0.005	

Fluid density

Density of the working fluid. The default value is 850 kg/m^3.

Kinematic viscosity

Kinematic viscosity of the working fluid. The default value is 1.8e-5 m^2/s.

Bulk modulus at atm. pressure and no gas

Bulk modulus of the working fluid, at atmospheric pressure and with no entrapped air. The default value is 8e8 Pa.

Relative amount of trapped air

Amount of entrained, nondissolved gas in the fluid. The amount is specified as the ratio of gas volume at normal conditions to the fluid volume in the chamber. In practice, the relative amount of trapped air is always greater than 0. If set to 0, ideal fluid is assumed, but you will get a warning upon simulation. The default value is 0.005.

Ports

The block has one hydraulic conserving port.

See Also

Hydraulic Fluid

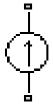
Purpose

Ideal constant current source

Library

Electrical Sources

Description

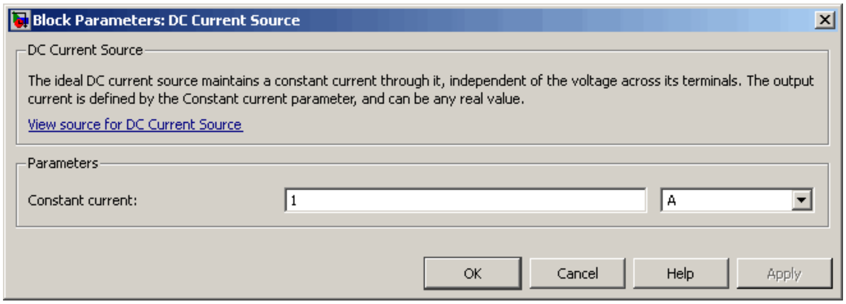


The DC Current Source block represents an ideal current source that is powerful enough to maintain specified current through it regardless of the voltage across the source.

You specify the output current by using the **Constant current** parameter, which can be positive or negative.

The positive direction of the current flow is indicated by the arrow.

Dialog Box and Parameters



Block Parameters: DC Current Source

DC Current Source

The ideal DC current source maintains a constant current through it, independent of the voltage across its terminals. The output current is defined by the Constant current parameter, and can be any real value.

View source for DC Current Source

Parameters

Constant current: 1 A

Constant current

Output current. You can specify positive or negative values. The default value is 1 A.

Ports

The block has two electrical conserving ports associated with its terminals.

See Also

DC Voltage Source

DC Voltage Source

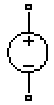
Purpose

Ideal constant voltage source

Library

Electrical Sources

Description

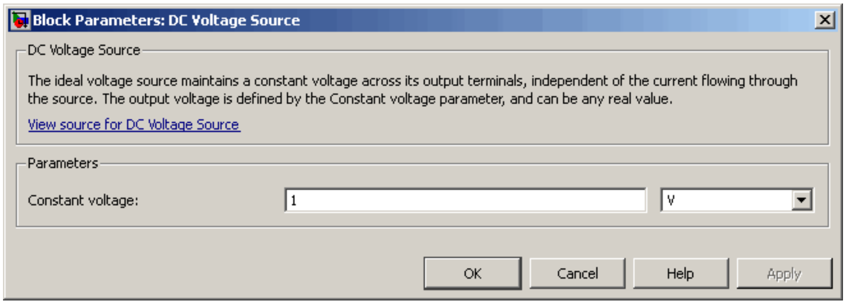


The DC Voltage Source block represents an ideal voltage source that is powerful enough to maintain specified voltage at its output regardless of the current flowing through the source.

You specify the output voltage by using the **Constant voltage** parameter, which can be positive or negative.

Connections + and – are conserving electrical ports corresponding to the positive and negative terminals of the voltage source, respectively. The current is positive if it flows from positive to negative, and the voltage across the source is equal to the difference between the voltage at the positive and the negative terminal, *V*(+) – *V*(–).

Dialog Box and Parameters


Block Parameters: DC Voltage Source

DC Voltage Source

The ideal voltage source maintains a constant voltage across its output terminals, independent of the current flowing through the source. The output voltage is defined by the Constant voltage parameter, and can be any real value.

View source for DC Voltage Source

Parameters

Constant voltage: 1 V

Constant voltage

Output voltage. You can specify positive or negative values. The default value is 1 V.

Ports

The block has the following ports:

+

Electrical conserving port associated with the source positive terminal.

-
    Electrical conserving port associated with the source negative terminal.

See Also

DC Current Source

Diode

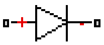
Purpose

Piecewise linear diode in electrical systems

Library

Electrical Elements

Description



The Diode block models a piecewise linear diode. If the voltage across the diode is bigger than the **Forward voltage** parameter value, then the diode behaves like a linear resistor with low resistance, given by the **On resistance** parameter value, plus a series voltage source. If the voltage across the diode is less than the forward voltage, then the diode behaves like a linear resistor with low conductance given by the **Off conductance** parameter value.

When forward biased, the series voltage source is described with the following equation:

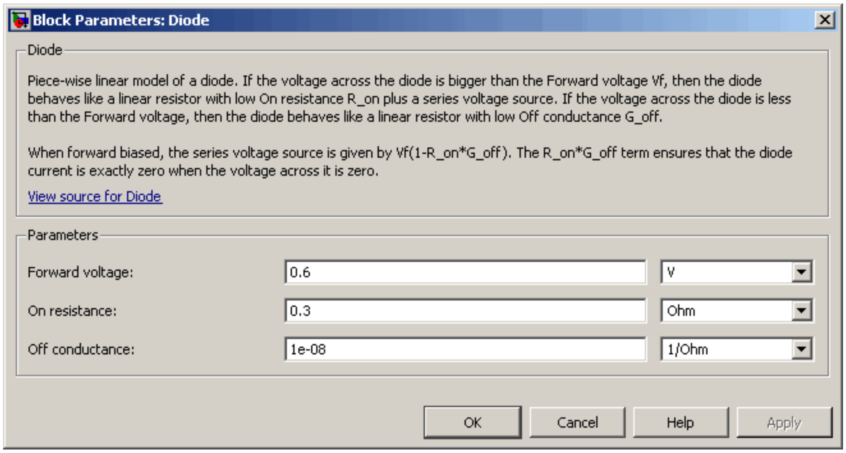
$$V = V_f(1 - R_{on}G_{off}),$$

where

V	Voltage
Vf	Forward voltage
R_{on}	On resistance
G_{off}	Off conductance

The $R_{on} \cdot G_{off}$ term ensures that the diode current is exactly zero when the voltage across it is zero.

Dialog Box and Parameters



Block Parameters: Diode

Diode

Piece-wise linear model of a diode. If the voltage across the diode is bigger than the Forward voltage Vf, then the diode behaves like a linear resistor with low On resistance R_on plus a series voltage source. If the voltage across the diode is less than the Forward voltage, then the diode behaves like a linear resistor with low Off conductance G_off.

When forward biased, the series voltage source is given by Vf(1-R_on*G_off). The R_on*G_off term ensures that the diode current is exactly zero when the voltage across it is zero.

View source for Diode

Parameters

Forward voltage: 0.6 V

On resistance: 0.3 Ohm

Off conductance: 1e-08 1/Ohm

Forward voltage

Minimum voltage that needs to be applied for the diode to become forward-biased. The default value is `0.6` V.

On resistance

The resistance of a forward-biased diode. The default value is `0.3` Ω.

Off conductance

The conductance of a reverse-biased diode. The default value is `1e-8` 1/Ω.

Ports

The block has the following ports:

+

Electrical conserving port associated with the diode positive terminal.

-

Electrical conserving port associated with the diode negative terminal.

Electrical Reference

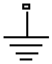
Purpose

Connection to electrical ground

Library

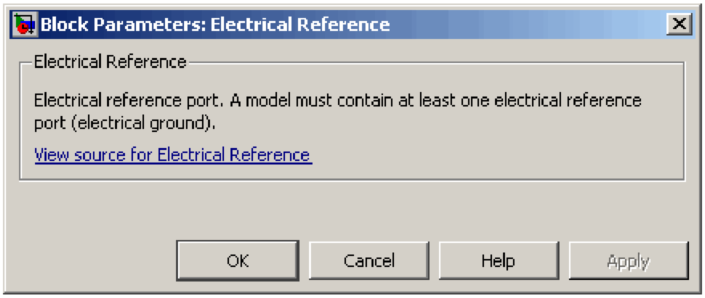
Electrical Elements

Description



The Electrical Reference block represents an electrical ground. Electrical conserving ports of all the blocks that are directly connected to ground must be connected to an Electrical Reference block. A model with electrical elements must contain at least one Electrical Reference block.

Dialog Box and Parameters



Block Parameters: Electrical Reference

Electrical Reference

Electrical reference port. A model must contain at least one electrical reference port (electrical ground).

View source for Electrical Reference

The Electrical Reference block has no parameters.

Ports

The block has one electrical conserving port.

See Also

Hydraulic Reference

Mechanical Rotational Reference

Mechanical Translational Reference

Thermal Reference

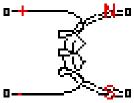
Purpose

Lossless electromagnetic energy conversion device

Library

Magnetic Elements

Description



The Electromagnetic Converter block provides a generic interface between the electrical and magnetic domains.

The block is based on the following equations:

$$\mathtt{F} = N \cdot I$$

$$V = -N \cdot \frac{d\Phi}{dt}$$

where

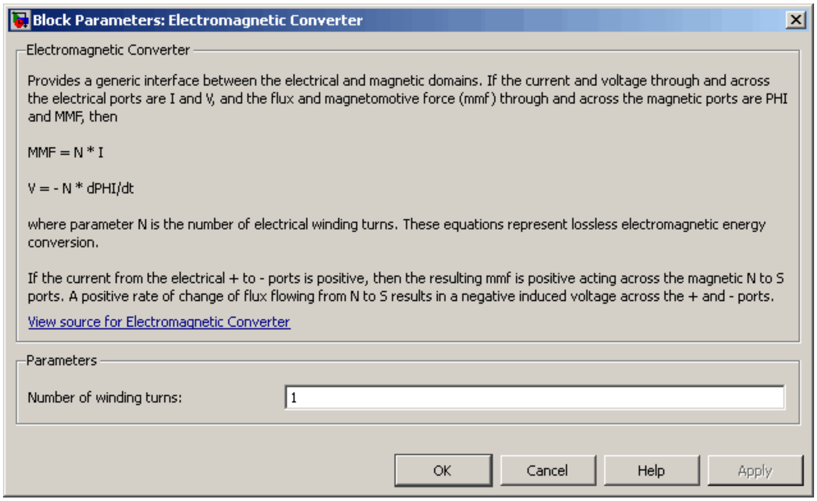
$\mathtt{F}$	Magnetomotive force (mmf) across the magnetic ports
Φ	Flux through the magnetic ports
I	Current through the electrical ports
V	Voltage across the electrical ports
N	Number of electrical winding turns
t	Simulation time

Connections N and S are magnetic conserving ports, and connections + and – are electrical conserving ports. If the current from the electrical + to – ports is positive, then the resulting mmf is positive acting across the magnetic N to S ports. A positive rate of change of flux flowing from N to S results in a negative induced voltage across the + and – ports.

Basic Assumptions and Limitations

Electromagnetic energy conversion is lossless.

Dialog Box and Parameters



Block Parameters: Electromagnetic Converter

Electromagnetic Converter

Provides a generic interface between the electrical and magnetic domains. If the current and voltage through and across the electrical ports are I and V, and the flux and magnetomotive force (mmf) through and across the magnetic ports are PHI and MMF, then

MMF = N * I

V = - N * dPHI/dt

where parameter N is the number of electrical winding turns. These equations represent lossless electromagnetic energy conversion.

If the current from the electrical + to - ports is positive, then the resulting mmf is positive acting across the magnetic N to S ports. A positive rate of change of flux flowing from N to S results in a negative induced voltage across the + and - ports.

View source for Electromagnetic Converter

Parameters

Number of winding turns: 1

Number of winding turns

Number of electrical winding turns. The default value is 1.

Ports

The block has the following ports:

N

Magnetic conserving port associated with the block North terminal.

S

Magnetic conserving port associated with the block South terminal.

+

Positive electrical conserving port.

-

Negative electrical conserving port.

See Also

Reluctance Force Actuator

Purpose

Pressure differential across tube or channel due to change in fluid velocity

Library

Hydraulic Elements

Description

The Fluid Inertia block models pressure differential, due to change in fluid velocity, across a fluid passage of constant cross-sectional area. The pressure differential is determined according to the following equation:

$$p = \rho \frac{L}{A} \frac{dq}{dt}$$

where

p	Pressure differential
ρ	Fluid density
L	Passage length
A	Passage area
q	Flow rate
t	Time

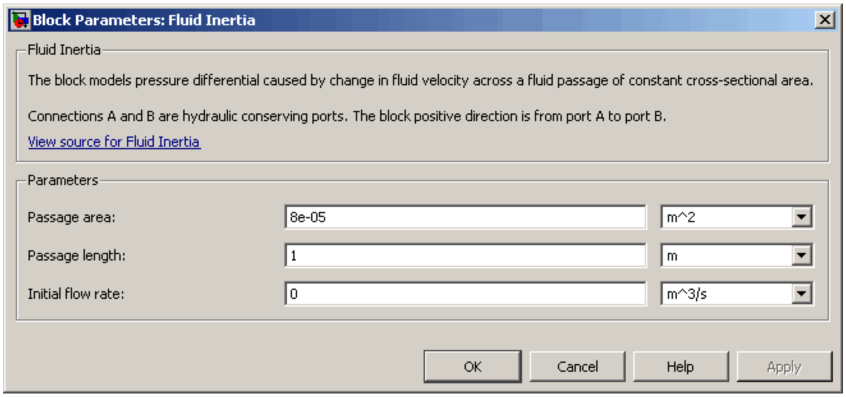
Use this block in various pipe or channel models that require fluid inertia to be accounted for.

The block positive direction is from port A to port B. This means that the flow rate is positive if it flows from A to B.

Assumptions and Limitations

Fluid density remains constant.

Dialog Box and Parameters



Block Parameters: Fluid Inertia

Fluid Inertia

The block models pressure differential caused by change in fluid velocity across a fluid passage of constant cross-sectional area.

Connections A and B are hydraulic conserving ports. The block positive direction is from port A to port B.

View source for Fluid Inertia

Parameters

Passage area: 8e-05 m^2

Passage length: 1 m

Initial flow rate: 0 m^3/s

Passage area

Fluid passage cross-sectional area. The default value is 8e-5 m^2.

Passage length

Length of the fluid passage. The default value is 1 m.

Initial flow rate

Initial flow rate through the passage. This parameter specifies the initial condition for use in computing the block's initial state at the beginning of a simulation run. For more information, see "Initial Conditions Computation". The default value is 0.

Global Parameters

Parameter determined by the type of working fluid:

- **Fluid density**

Use the Hydraulic Fluid block or the Custom Hydraulic Fluid block to specify the fluid properties.

Ports

The block has the following ports:

A

Hydraulic conserving port associated with the passage inlet.

B

Hydraulic conserving port associated with the passage outlet.

Flux Sensor

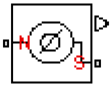
Purpose

Ideal flux sensor

Library

Magnetic Sensors

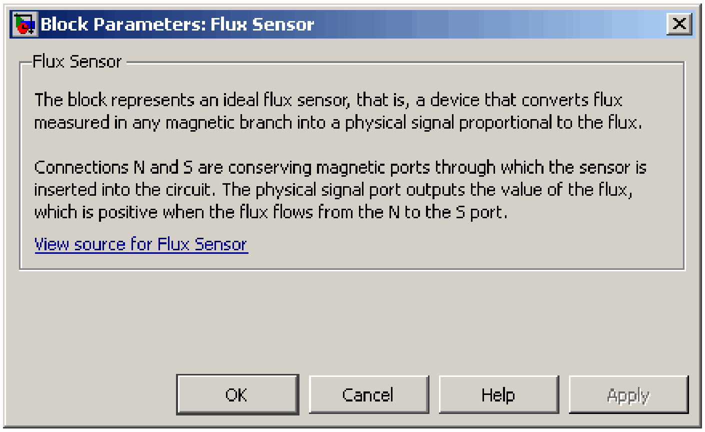
Description



The Flux Sensor block represents an ideal flux sensor, that is, a device that converts flux measured in any magnetic branch into a physical signal proportional to the flux.

Connections N and S are conserving magnetic ports through which the sensor is inserted into the circuit. The physical signal port outputs the value of the flux, which is positive when the flux flows from the N to the S port.

Dialog Box and Parameters



The block represents an ideal flux sensor, that is, a device that converts flux measured in any magnetic branch into a physical signal proportional to the flux.

Connections N and S are conserving magnetic ports through which the sensor is inserted into the circuit. The physical signal port outputs the value of the flux, which is positive when the flux flows from the N to the S port.

View source for Flux Sensor

The block has no parameters.

Ports

The block has the following ports:

N

Magnetic conserving port associated with the sensor North terminal.

S

Magnetic conserving port associated with the sensor South terminal.

The block also has a physical signal output port, which outputs the value of the flux.

See Also

Controlled Flux Source

Flux Source

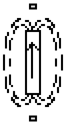
Purpose

Ideal flux source

Library

Magnetic Sources

Description

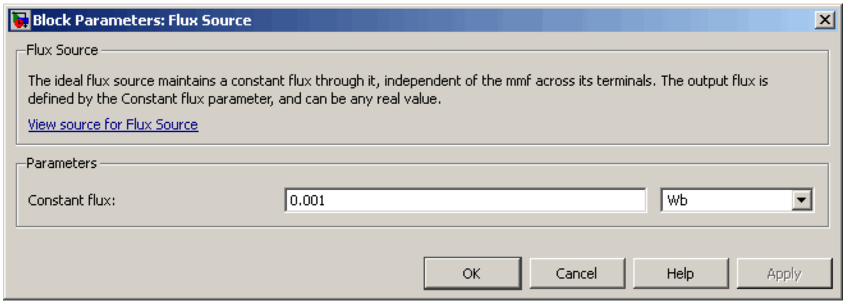


The Flux Source block represents an ideal flux source that is powerful enough to maintain specified constant flux through it, regardless of the mmf across its terminals.

You specify the output flux by using the **Constant flux** parameter, which can be positive, negative, or zero.

The positive direction of the flux flow is indicated by the arrow.

Dialog Box and Parameters



Block Parameters: Flux Source

Flux Source

The ideal flux source maintains a constant flux through it, independent of the mmf across its terminals. The output flux is defined by the Constant flux parameter, and can be any real value.

View source for Flux Source

Parameters

Constant flux: 0.001 Wb

Constant flux

Output flux. You can specify any real value. The default value is `0.001` Wb.

Ports

The block has two magnetic conserving ports associated with its terminals.

See Also

Controlled Flux Source

Controlled MMF Source

MMF Source

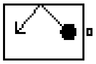
Purpose

Pneumatic domain properties for attached circuit

Library

Pneumatic Utilities

Description



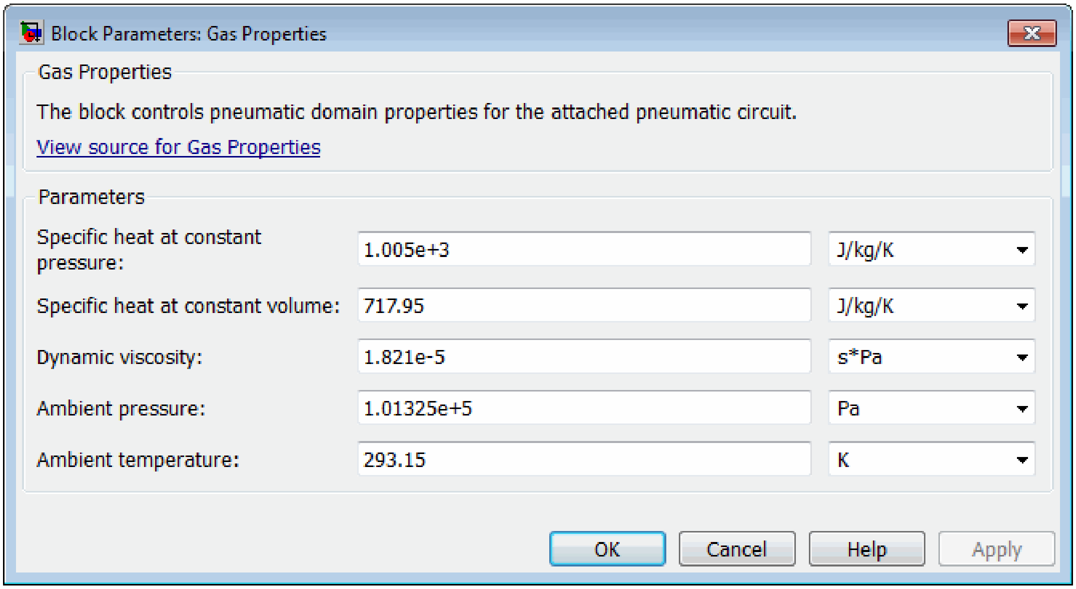
The Gas Properties block defines pneumatic domain properties for a circuit, that is, the gas properties that act as global parameters for all the blocks connected to the pneumatic circuit. These gas properties are assumed to be constant during simulation time.

The Gas Properties block lets you specify the gas properties, such as specific heat at constant pressure and constant volume, as well as viscosity, as block parameters. It also lets you specify ambient pressure and ambient temperature.

The Gas Properties block has one port. You can connect it to a pneumatic diagram by branching a connection line off the main line and connecting it to the port. When you connect the Gas Properties block to a pneumatic line, the software automatically identifies the pneumatic blocks connected to the particular circuit and propagates the gas properties to all the pneumatic blocks in the circuit.

Each topologically distinct pneumatic circuit in a diagram requires exactly one Gas Properties block to be connected to it. Therefore, there must be as many Gas Properties blocks as there are pneumatic circuits in the system. If no Gas Properties block is attached to a circuit, the pneumatic blocks in this circuit use the gas properties corresponding to the default Gas Properties block parameter values.

Dialog Box and Parameters



Block Parameters: Gas Properties

Gas Properties

The block controls pneumatic domain properties for the attached pneumatic circuit.

View source for Gas Properties

Parameters

Specific heat at constant pressure:	1.005e+3	J/kg/K
Specific heat at constant volume:	717.95	J/kg/K
Dynamic viscosity:	1.821e-5	s*Pa
Ambient pressure:	1.01325e+5	Pa
Ambient temperature:	293.15	K

Specific heat at constant pressure

Specify the gas specific heat at constant pressure. The default value is `1.005e3` J/kg/K.

Specific heat at constant volume

Specify the gas specific heat at constant volume. The default value is `717.95` J/kg/K.

Dynamic viscosity

Specify the gas dynamic viscosity. The default value is `1.821e-5` s*Pa.

Ambient pressure

Specify the gas ambient pressure. The default value is `101325` Pa.

Ambient temperature

Specify the gas ambient temperature. The default value is `293.15` K.

Ports

The block has one pneumatic conserving port.

Gear Box

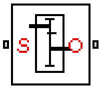
Purpose

Gear box in mechanical systems

Library

Mechanisms

Description



The Gear Box block represents an ideal, nonplanetary, fixed gear ratio gear box. The gear ratio is determined as the ratio of the input shaft angular velocity to that of the output shaft.

The gear box is described with the following equations:

$$\omega_1 = N \cdot \omega_2$$

$$T_2 = N \cdot T_1$$

$$P_1 = \omega_1 \cdot T_1$$

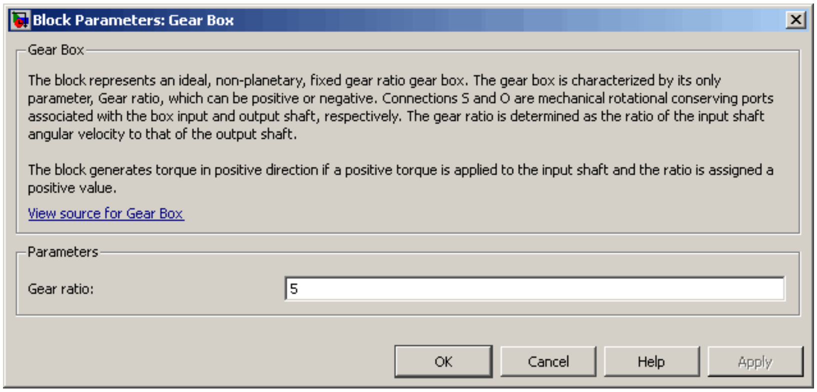
$$P_2 = -\omega_2 \cdot T_2$$

where

ω_1	Input shaft angular velocity
ω_2	Output shaft angular velocity
N	Gear ratio
T_1	Torque on the input shaft
T_2	Torque on the output shaft
P_1	Power on the input shaft
P_2	Power on the output shaft. Notice the minus sign in computing P_2. One of the network rules is that the power flowing through a conserving port is positive if it is removed (dissipated) from the circuit, and is negative if the component generates power into the system.

Connections S and O are mechanical rotational conserving ports associated with the box input and output shaft, respectively. The block positive directions are from S to the reference point and from the reference point to O.

Dialog Box and Parameters



Block Parameters: Gear Box

Gear Box

The block represents an ideal, non-planetary, fixed gear ratio gear box. The gear box is characterized by its only parameter, Gear ratio, which can be positive or negative. Connections S and O are mechanical rotational conserving ports associated with the box input and output shaft, respectively. The gear ratio is determined as the ratio of the input shaft angular velocity to that of the output shaft.

The block generates torque in positive direction if a positive torque is applied to the input shaft and the ratio is assigned a positive value.

View source for Gear Box

Parameters

Gear ratio: 5

Gear ratio

The ratio of the input shaft angular velocity to that of the output shaft. You can specify both positive and negative values. The default value is 5.

Ports

The block has the following ports:

S

Mechanical rotational conserving port associated with input shaft.

O

Mechanical rotational conserving port associated with the output shaft.

Gyrator

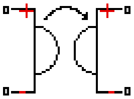
Purpose

Ideal gyrator in electrical systems

Library

Electrical Elements

Description



Gyrators can be used to implement an inductor with a capacitor. The main benefit is that an equivalent inductance can be created with a much smaller physically sized capacitance. In practice, a gyrator is implemented with an op-amp plus additional passive components.

The Gyrator block models an ideal gyrator with no losses, described with the following equations:

$$I1 = G\square V2$$

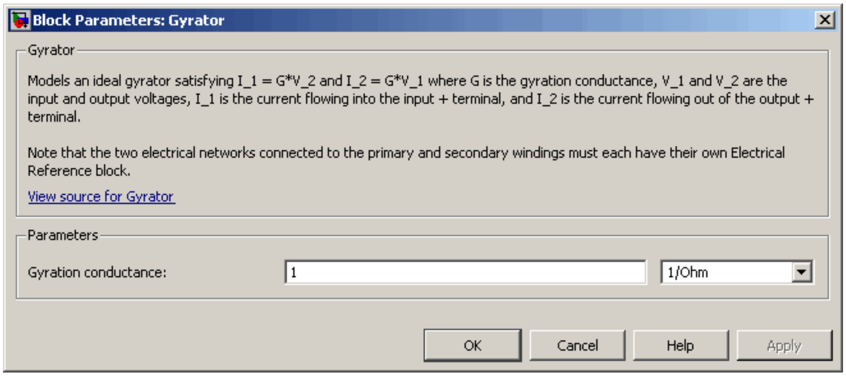
$$I2 = G\square V1$$

where

`V1`	Input voltage
`V2`	Output voltage
`I1`	Current flowing into the input + terminal
`I2`	Current flowing out of the output + terminal
`G`	Gyration conductance

The two electrical networks connected to the primary and secondary windings must each have their own Electrical Reference block.

Dialog Box and Parameters



Block Parameters: Gyrator

Gyrator

Models an ideal gyrator satisfying I_1 = G*V_2 and I_2 = G*V_1 where G is the gyration conductance, V_1 and V_2 are the input and output voltages, I_1 is the current flowing into the input + terminal, and I_2 is the current flowing out of the output + terminal.

Note that the two electrical networks connected to the primary and secondary windings must each have their own Electrical Reference block.

View source for Gyrator

Parameters

Gyration conductance: 1 1/Ohm

Gyration conductance

The gyration conductance constant *G*. The default value is 1.

Ports

The block has four electrical conserving ports. Polarity is indicated by the + and – signs.

Hydraulic Cap

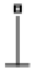
Purpose

Hydraulic port terminator with zero flow

Library

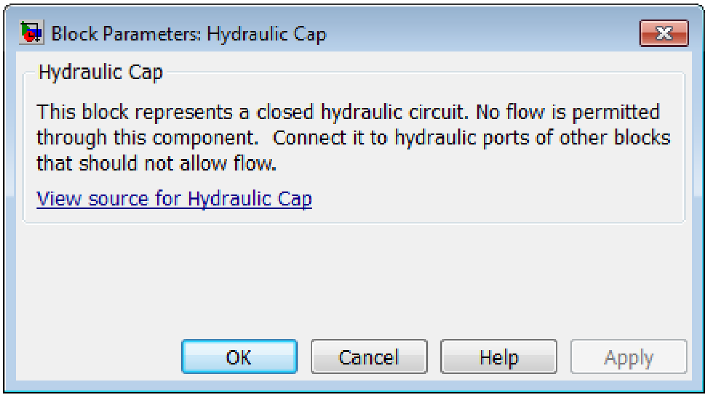
Hydraulic Elements

Description



The Hydraulic Cap block represents a hydraulic plug, that is, a hydraulic port with zero flow through it. Physical Network block diagrams do not allow unconnected Conserving ports. Use this block to terminate hydraulic ports (on other blocks) that you wish to cap.

Dialog Box and Parameters



Block Parameters: Hydraulic Cap

Hydraulic Cap

This block represents a closed hydraulic circuit. No flow is permitted through this component. Connect it to hydraulic ports of other blocks that should not allow flow.

View source for Hydraulic Cap

The Hydraulic Cap block has no parameters.

Ports

The block has one hydraulic conserving port.

See Also

Open Circuit

Rotational Free End

Translational Free End

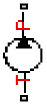
Purpose

Ideal source of hydraulic energy, characterized by constant flow rate

Library

Hydraulic Sources

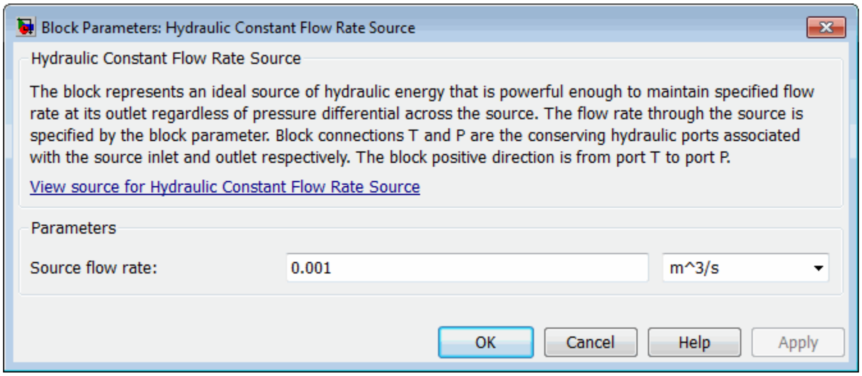
Description



The Hydraulic Constant Flow Rate Source block represents an ideal source of hydraulic energy that is powerful enough to maintain specified flow rate at its outlet regardless of the pressure differential across the source. The **Source flow rate** parameter specifies the flow rate through the source.

Block connections T and P correspond to the hydraulic inlet and outlet ports, respectively. The block positive direction is from port T to port P.

Dialog Box and Parameters



Block Parameters: Hydraulic Constant Flow Rate Source

Hydraulic Constant Flow Rate Source

The block represents an ideal source of hydraulic energy that is powerful enough to maintain specified flow rate at its outlet regardless of pressure differential across the source. The flow rate through the source is specified by the block parameter. Block connections T and P are the conserving hydraulic ports associated with the source inlet and outlet respectively. The block positive direction is from port T to port P.

View source for Hydraulic Constant Flow Rate Source

Parameters

Source flow rate: 0.001 m^3/s

Source flow rate

Specifies the flow rate through the source. The default value is `0.001` m^3/s.

Ports

The block has the following ports:

T

Hydraulic conserving port associated with the source inlet.

P

Hydraulic conserving port associated with the source outlet.

See Also

Hydraulic Flow Rate Sensor

Hydraulic Flow Rate Source

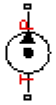
Purpose

Ideal source of hydraulic energy, characterized by constant pressure

Library

Hydraulic Sources

Description



The Hydraulic Constant Pressure Source block represents an ideal source of hydraulic energy that is powerful enough to maintain the specified pressure differential between its inlet and outlet regardless of the flow rate through the source.

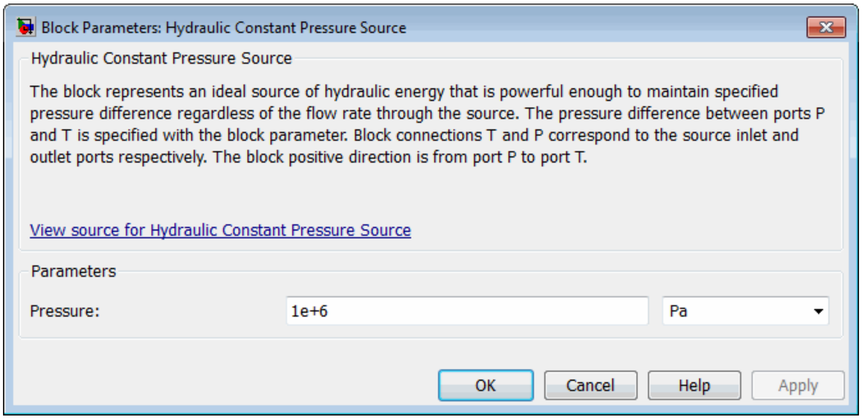
The **Pressure** parameter specifies the pressure differential across the source

$$p = p_P - p_T$$

where p_P, p_T are the gauge pressures at the source ports.

Block connections T and P correspond to the hydraulic inlet and outlet ports, respectively. The block positive direction is from port P to port T.

Dialog Box and Parameters



Block Parameters: Hydraulic Constant Pressure Source

Hydraulic Constant Pressure Source

The block represents an ideal source of hydraulic energy that is powerful enough to maintain specified pressure difference regardless of the flow rate through the source. The pressure difference between ports P and T is specified with the block parameter. Block connections T and P correspond to the source inlet and outlet ports respectively. The block positive direction is from port P to port T.

View source for Hydraulic Constant Pressure Source

Parameters

Pressure: 1e+6 Pa

Pressure

Specifies the pressure difference between the source inlet and outlet. The default value is `1e6` Pa.

Ports

The block has the following ports:

T

Hydraulic conserving port associated with the source inlet.

P

Hydraulic conserving port associated with the source outlet.

See Also

Hydraulic Pressure Source

Hydraulic Pressure Sensor

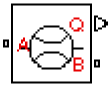
Purpose

Ideal flow meter

Library

Hydraulic Sensors

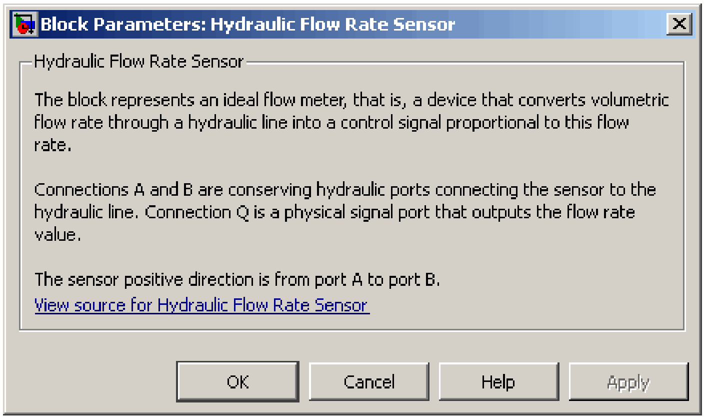
Description



The Hydraulic Flow Rate Sensor block represents an ideal flow meter, that is, a device that converts volumetric flow rate through a hydraulic line into a control signal proportional to this flow rate. The sensor is ideal because it does not account for inertia, friction, delays, pressure loss, and so on.

Connections A and B are conserving hydraulic ports connecting the sensor to the hydraulic line. Connection Q is a physical signal port that outputs the flow rate value. The sensor positive direction is from A to B. This means that the flow rate is positive if it flows from A to B.

Dialog Box and Parameters



Hydraulic Flow Rate Sensor

The block represents an ideal flow meter, that is, a device that converts volumetric flow rate through a hydraulic line into a control signal proportional to this flow rate.

Connections A and B are conserving hydraulic ports connecting the sensor to the hydraulic line. Connection Q is a physical signal port that outputs the flow rate value.

The sensor positive direction is from port A to port B.

View source for Hydraulic Flow Rate Sensor

The block has no parameters.

Ports

The block has the following ports:

A

Hydraulic conserving port associated with the sensor positive probe.

B

Hydraulic conserving port associated with the sensor negative (reference) probe.

Q

Physical signal port that outputs the flow rate value.

See Also

Hydraulic Constant Flow Rate Source

Hydraulic Flow Rate Source

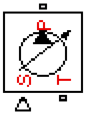
Purpose

Ideal source of hydraulic energy, characterized by flow rate

Library

Hydraulic Sources

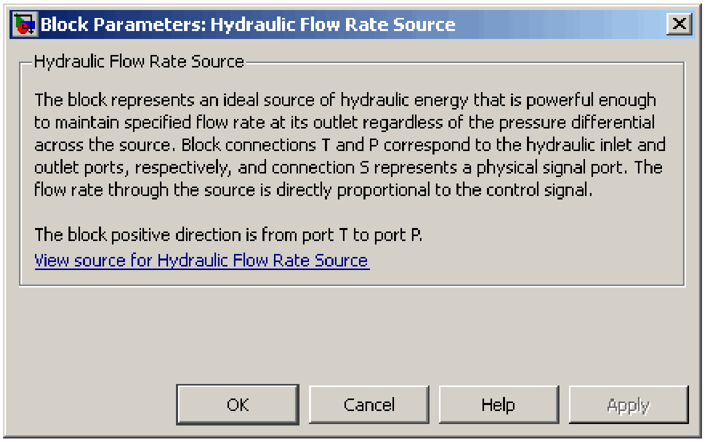
Description



The Hydraulic Flow Rate Source block represents an ideal source of hydraulic energy that is powerful enough to maintain specified flow rate at its outlet regardless of the pressure differential across the source. Block connections T and P correspond to the hydraulic inlet and outlet ports, respectively, and connection S represents a control signal port. The flow rate through the source is directly proportional to the signal at the control port S. The entire variety of Simulink signal sources can be used to generate desired flow rate variation profile.

The block positive direction is from port T to port P. This means that the flow rate is positive if it flows from T to P. The pressure differential is determined as $p = p_T - p_P$ and is negative if pressure at the source outlet is greater than pressure at its inlet. The power generated by the source is negative if the source delivers energy to port P.

Dialog Box and Parameters


Block Parameters: Hydraulic Flow Rate Source
Hydraulic Flow Rate Source

The block represents an ideal source of hydraulic energy that is powerful enough to maintain specified flow rate at its outlet regardless of the pressure differential across the source. Block connections T and P correspond to the hydraulic inlet and outlet ports, respectively, and connection S represents a physical signal port. The flow rate through the source is directly proportional to the control signal.

The block positive direction is from port T to port P.

View source for Hydraulic Flow Rate Source

The block has no parameters.

Ports

The block has the following ports:

T

Hydraulic conserving port associated with the source inlet.

P

Hydraulic conserving port associated with the source outlet.

S

Control signal port.

See Also

Hydraulic Constant Flow Rate Source

Hydraulic Flow Rate Sensor

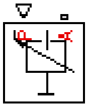
Purpose

Variable volume hydraulic capacity in cylinders

Library

Hydraulic Elements

Description



The Hydraulic Piston Chamber block models fluid compressibility in a chamber created by a piston of a cylinder. The fluid is considered to be a mixture of liquid and a small amount of entrained, nondissolved gas. Use this block together with the Translational Hydro-Mechanical Converter block.

Note The Hydraulic Piston Chamber block takes into account only the flow rate caused by fluid compressibility. The fluid volume consumed to create piston velocity is accounted for in the Translational Hydro-Mechanical Converter block.

The chamber is simulated according to the following equations (see [1, 2]):

$$q = \frac{V_0 + A(x_0 + x \cdot or)}{E} \cdot \frac{dp}{dt}$$

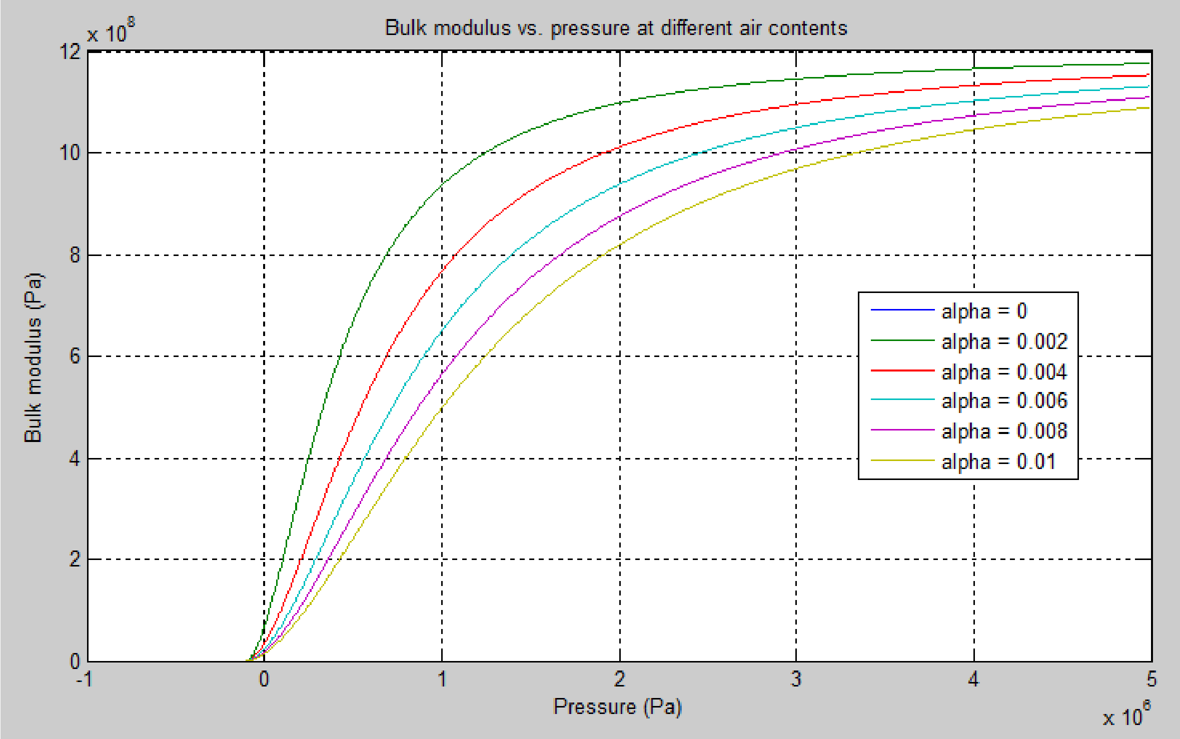
$$E = E_l \frac{1 + \alpha \left(\frac{p_a}{p_a + p} \right)^{1/n}}{1 + \alpha \frac{p_a^{1/n}}{n \cdot (p_a + p)^{\frac{n+1}{n}}} E_l}$$

where

q	Flow rate due to fluid compressibility
V_0	Dead volume
A	Effective piston area

x_0	Piston initial position
x	Piston displacement from initial position
or	Chamber orientation with respect to the globally assigned positive direction. If displacement in positive direction increases the volume of the chamber, or equals 1. If displacement in positive direction decreases the volume of the chamber, or equals –1.
E	Fluid bulk modulus
E_1	Pure liquid bulk modulus
p	Gauge pressure of fluid in the chamber
p_α	Atmospheric pressure
α	Relative gas content at atmospheric pressure, $\alpha = V_G/V_L$
V_G	Gas volume at atmospheric pressure
V_L	Volume of liquid
n	Gas-specific heat ratio

The main objective of representing fluid as a mixture of liquid and gas is to introduce an approximate model of cavitation, which takes place in a chamber if pressure drops below fluid vapor saturation level. As it is seen in the graph below, the bulk modulus of a mixture decreases at $p \to p_a$, thus considerably slowing down further pressure change. At high pressure, $p >> p_a$, a small amount of nondissolved gas has practically no effect on the system behavior.



Cavitation is an inherently thermodynamic process, requiring consideration of multiple-phase fluids, heat transfers, etc., and as such cannot be accurately simulated with Simscape software. But the simplified version implemented in the block is good enough to signal if pressure falls below dangerous level, and to prevent computation failure that normally occurs at negative pressures.

If pressure falls below absolute vacuum (–101325 Pa), the simulation stops and an error message is displayed.

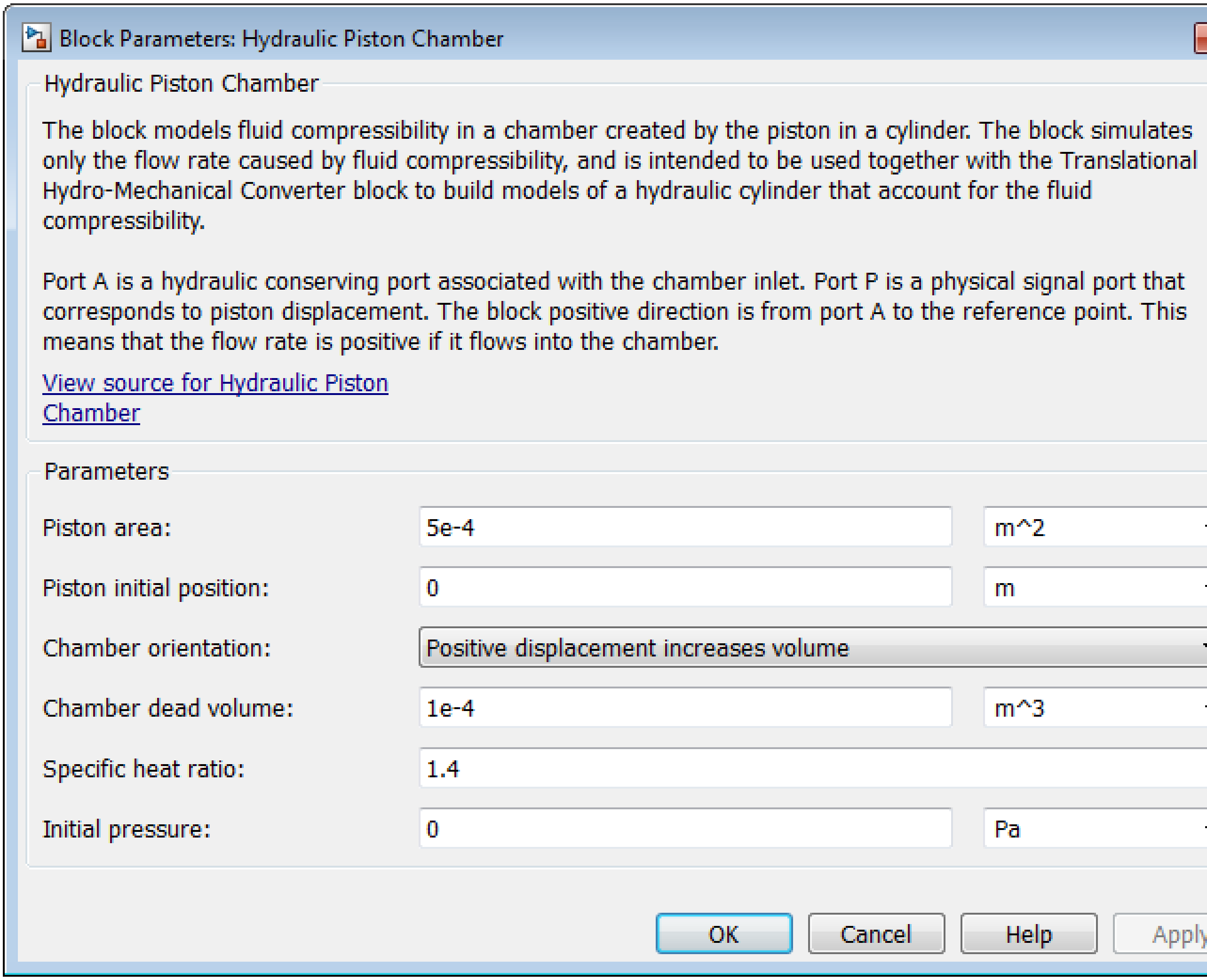
Port A is a hydraulic conserving port associated with the chamber inlet. Port P is a physical signal port that controls piston displacement.

The block positive direction is from port A to the reference point. This means that the flow rate is positive if it flows into the chamber.

Basic Assumptions and Limitations

- Fluid density remains constant.
- Chamber volume can not be less that the dead volume.
- Fluid fills the entire chamber volume.

Dialog Box and Parameters



Block Parameters: Hydraulic Piston Chamber

Hydraulic Piston Chamber

The block models fluid compressibility in a chamber created by the piston in a cylinder. The block simulates only the flow rate caused by fluid compressibility, and is intended to be used together with the Translational Hydro-Mechanical Converter block to build models of a hydraulic cylinder that account for the fluid compressibility.

Port A is a hydraulic conserving port associated with the chamber inlet. Port P is a physical signal port that corresponds to piston displacement. The block positive direction is from port A to the reference point. This means that the flow rate is positive if it flows into the chamber.

View source for Hydraulic Piston Chamber

Parameters

Piston area: 5e-4 m^2

Piston initial position: 0 m

Chamber orientation: Positive displacement increases volume

Chamber dead volume: 1e-4 m^3

Specific heat ratio: 1.4

Initial pressure: 0 Pa

Piston area
Effective piston area. The default value is `5e-4` m^2.

Piston initial position
Initial offset of the piston from the cylinder cap. The default value is 0.

Chamber orientation
Specifies chamber orientation with respect to the globally assigned positive direction. The chamber can be installed in two different ways, depending upon whether the piston motion in the positive direction increases or decreases the volume of the chamber. If piston motion in the positive direction decreases the chamber volume, set the parameter to `Positive displacement decreases volume`. The default value is `Positive displacement increases volume`.

Chamber dead volume
Volume of fluid in the chamber at zero piston position. The default value is `1e-4` m^3.

Specific heat ratio
Gas-specific heat ratio. The default value is `1.4`.

Initial pressure
Initial pressure in the chamber. This parameter specifies the initial condition for use in computing the block's initial state at the beginning of a simulation run. For more information, see "Initial Conditions Computation". The default value is `0`.

Restricted Parameters

When your model is in Restricted editing mode, you cannot modify the following parameter:

- **Chamber orientation**

All other block parameters are available for modification.

Global Parameters

Parameters determined by the type of working fluid:

- **Fluid density**
- **Fluid kinematic viscosity**

Use the Hydraulic Fluid block or the Custom Hydraulic Fluid block to specify the fluid properties.

Ports

The block has the following ports:

A

Hydraulic conserving port associated with the chamber inlet.

P

Physical signal port that controls piston displacement.

References

[1] Manring, N.D., *Hydraulic Control Systems*, John Wiley & Sons, New York, 2005

[2] Meritt, H.E., *Hydraulic Control Systems*, John Wiley & Sons, New York, 1967

See Also

Constant Volume Hydraulic Chamber

Translational Hydro-Mechanical Converter

Variable Hydraulic Chamber

Hydraulic Pressure Sensor

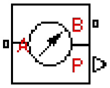
Purpose

Ideal pressure sensing device

Library

Hydraulic Sensors

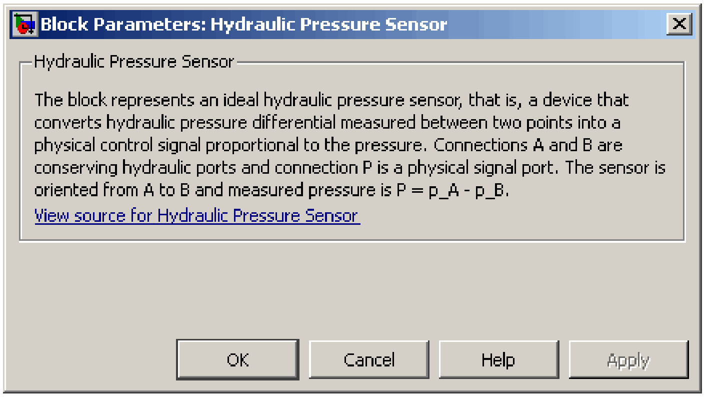
Description



The Hydraulic Pressure Sensor block represents an ideal hydraulic pressure sensor, that is, a device that converts hydraulic pressure differential measured between two points into a control signal proportional to this pressure. The sensor is ideal because it does not account for inertia, friction, delays, pressure loss, and so on.

Connections A and B are conserving hydraulic ports connecting the sensor to the hydraulic line. Connection P is a physical signal port that outputs the pressure value. The sensor positive direction is from A to B. This means that the pressure differential is determined as $p = p_A - p_B$.

Dialog Box and Parameters



Block Parameters: Hydraulic Pressure Sensor

Hydraulic Pressure Sensor

The block represents an ideal hydraulic pressure sensor, that is, a device that converts hydraulic pressure differential measured between two points into a physical control signal proportional to the pressure. Connections A and B are conserving hydraulic ports and connection P is a physical signal port. The sensor is oriented from A to B and measured pressure is P = p_A - p_B.

View source for Hydraulic Pressure Sensor

The block has no parameters.

Ports

The block has the following ports:

A

Hydraulic conserving port associated with the sensor positive probe.

B

Hydraulic conserving port associated with the sensor negative (reference) probe.

P

Physical signal port that outputs the pressure value.

See Also

Hydraulic Constant Pressure Source

Hydraulic Pressure Source

Hydraulic Pressure Source

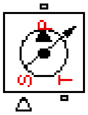
Purpose

Ideal source of hydraulic energy, characterized by pressure

Library

Hydraulic Sources

Description



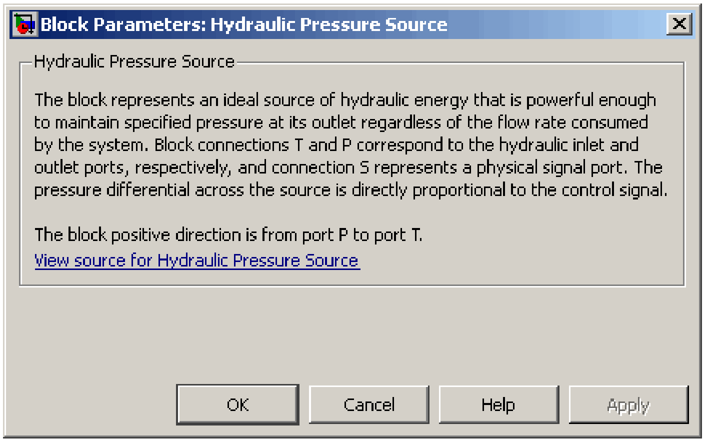
The Hydraulic Pressure Source block represents an ideal source of hydraulic energy that is powerful enough to maintain specified pressure at its outlet regardless of the flow rate consumed by the system. Block connections T and P correspond to the hydraulic inlet and outlet ports, respectively, and connection S represents a control signal port. The pressure differential across the source

$$p = p_P - p_T$$

where p_P, p_T are the gauge pressures at the source ports, is directly proportional to the signal at the control port S. The entire variety of Simulink signal sources can be used to generate desired pressure variation profile.

The block positive direction is from port P to port T. This means that the flow rate is positive if it flows from P to T. The power generated by the source is negative if the source delivers energy to port P.

Dialog Box and Parameters



Block Parameters: Hydraulic Pressure Source

Hydraulic Pressure Source

The block represents an ideal source of hydraulic energy that is powerful enough to maintain specified pressure at its outlet regardless of the flow rate consumed by the system. Block connections T and P correspond to the hydraulic inlet and outlet ports, respectively, and connection S represents a physical signal port. The pressure differential across the source is directly proportional to the control signal.

The block positive direction is from port P to port T.

View source for Hydraulic Pressure Source

The block has no parameters.

Ports

The block has the following ports:

T

Hydraulic conserving port associated with the source inlet.

P

Hydraulic conserving port associated with the source outlet.

S

Control signal port.

See Also

Hydraulic Constant Pressure Source

Hydraulic Pressure Sensor

Hydraulic Reference

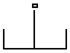
Purpose

Connection to atmospheric pressure

Library

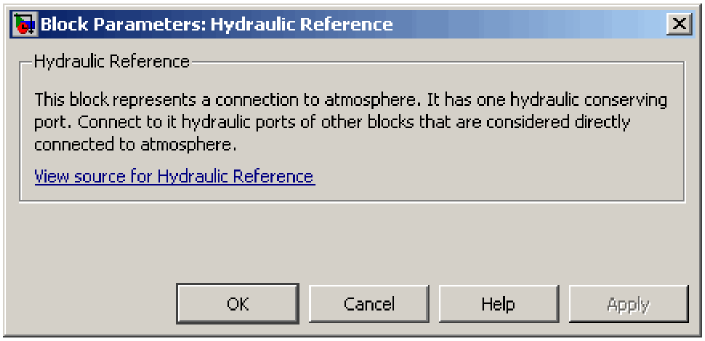
Hydraulic Elements

Description



The Hydraulic Reference block represents a connection to atmospheric pressure. Hydraulic conserving ports of all the blocks that are referenced to atmosphere (for example, suction ports of hydraulic pumps, or return ports of valves, cylinders, pipelines, if they are considered directly connected to atmosphere) must be connected to a Hydraulic Reference block.

Dialog Box and Parameters



Block Parameters: Hydraulic Reference

Hydraulic Reference

This block represents a connection to atmosphere. It has one hydraulic conserving port. Connect to it hydraulic ports of other blocks that are considered directly connected to atmosphere.

View source for Hydraulic Reference

The Hydraulic Reference block has no parameters.

Ports

The block has one hydraulic conserving port.

See Also

Electrical Reference

Mechanical Rotational Reference

Mechanical Translational Reference

Thermal Reference

Hydraulic Resistive Tube

Purpose

Hydraulic pipeline which accounts for friction losses only

Library

Hydraulic Elements

Description

The Hydraulic Resistive Tube block models hydraulic pipelines with circular and noncircular cross sections and accounts for resistive property only. In other words, the block is developed with the basic assumption of the steady state fluid momentum conditions. Neither fluid compressibility nor fluid inertia is considered in the model, meaning that features such as water hammer cannot be investigated. If necessary, you can add fluid compressibility, fluid inertia, and other effects to your model using other blocks, thus producing a more comprehensive model.

The end effects are also not considered, assuming that the flow is fully developed along the entire pipe length. To account for local resistances, such as bends, fittings, inlet and outlet losses, and so on, all the resistances are converted into their equivalent lengths, and then the total length of all the resistances is added to the pipe geometrical length.

Pressure loss due to friction is computed with the Darcy equation, in which losses are proportional to the flow regime-dependable friction factor and the square of the flow rate. The friction factor in turbulent regime is determined with the Haaland approximation (see [1]). The friction factor during transition from laminar to turbulent regimes is determined with the linear interpolation between extreme points of the regimes. As a result of these assumptions, the tube is simulated according to the following equations:

$$p = f \frac{\left(L + L_{eq}\right)}{D_H} \frac{\rho}{2A^2} q \cdot | q |$$

$$f = \begin{cases} K_s / Re & \text{for } Re <= Re_L \\ f_L + \dfrac{f_T - f_L}{Re_T - Re_L}(Re - Re_L) & \text{for } Re_L < Re < Re_T \\ \dfrac{1}{\left(-1.8\log_{10}\left(\dfrac{6.9}{Re} + \left(\dfrac{r / D_H}{3.7}\right)^{1.11}\right)\right)^2} & \text{for } Re >= Re_T \end{cases}$$

$$\mathrm{Re} = \frac{q \cdot D_H}{A \cdot \nu}$$

where

p	Pressure loss along the pipe due to friction
q	Flow rate through the pipe
Re	Reynolds number
Re_L	Maximum Reynolds number at laminar flow
Re_T	Minimum Reynolds number at turbulent flow
K_s	Shape factor that characterizes the pipe cross section
f_L	Friction factor at laminar border
f_T	Friction factor at turbulent border
A	Pipe cross-sectional area
D_H	Pipe hydraulic diameter
L	Pipe geometrical length
L_{eq}	Aggregate equivalent length of local resistances

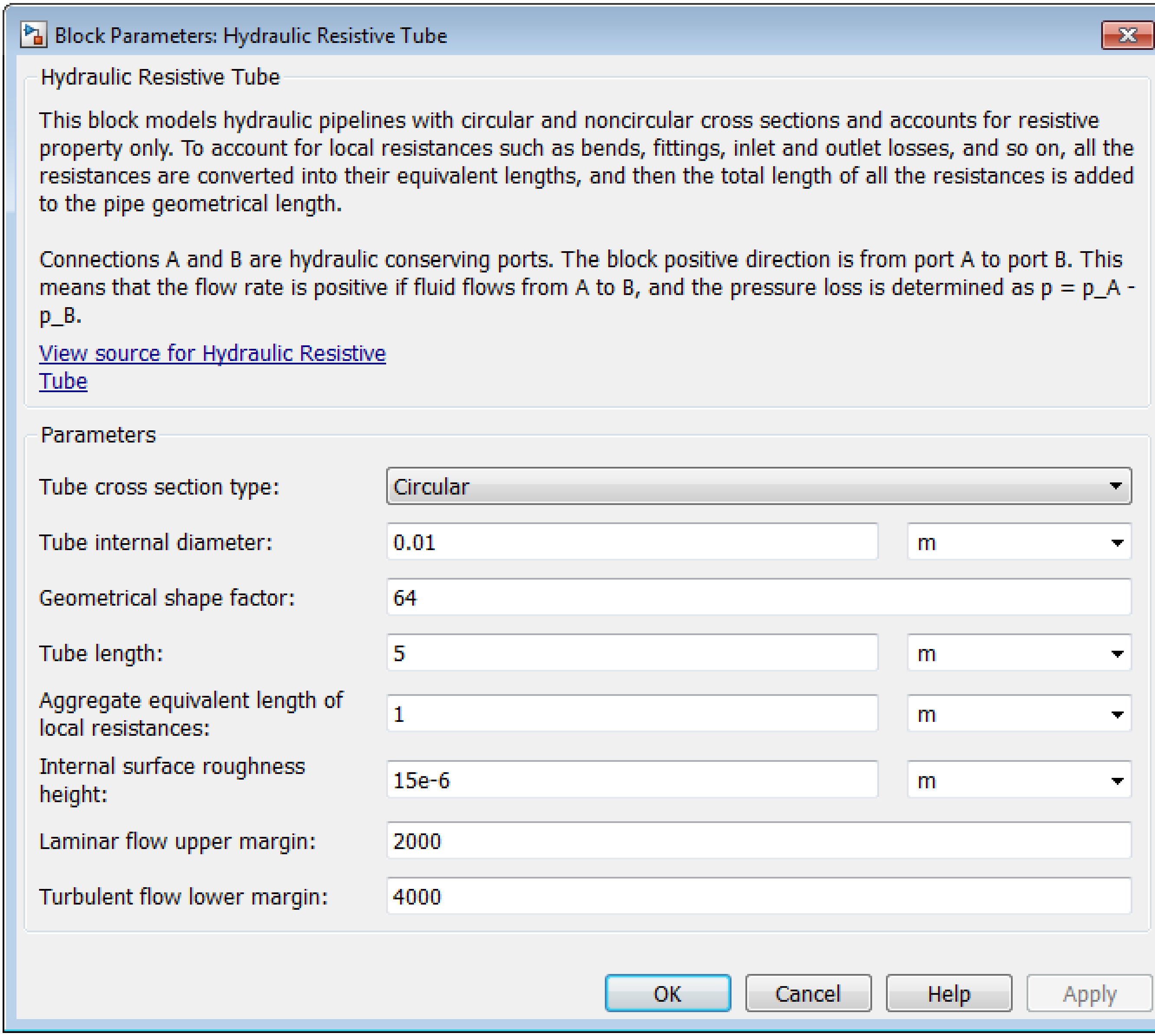
r	Height of the roughness on the pipe internal surface
v	Fluid kinematic viscosity

The block positive direction is from port A to port B. This means that the flow rate is positive if it flows from A to B, and the pressure loss is determined as $p = p_A - p_B$.

Basic Assumptions and Limitations

- Flow is assumed to be fully developed along the pipe length.
- Fluid inertia, fluid compressibility, and wall compliance are not taken into account.

Dialog Box and Parameters



Hydraulic Resistive Tube

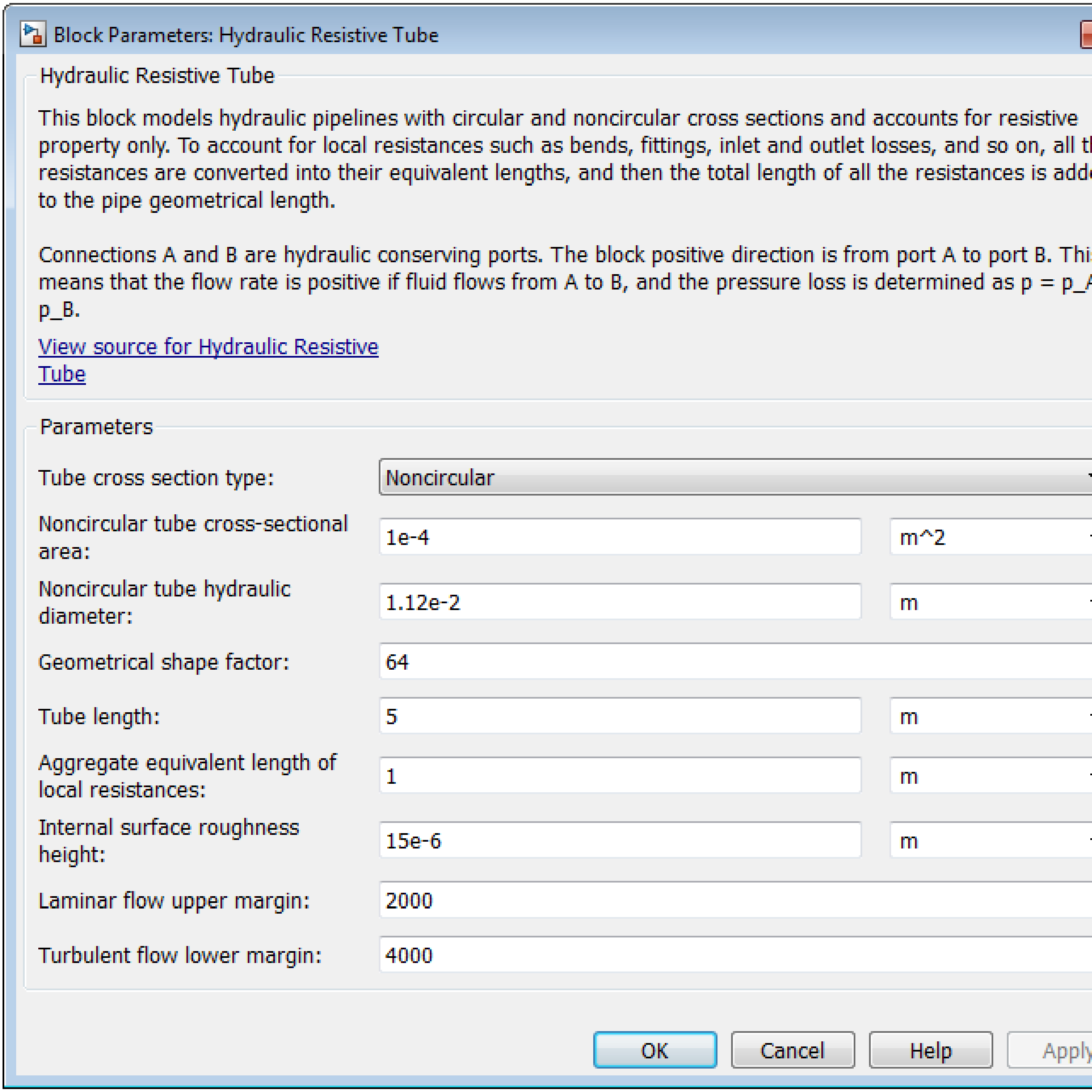
This block models hydraulic pipelines with circular and noncircular cross sections and accounts for resistive property only. To account for local resistances such as bends, fittings, inlet and outlet losses, and so on, all the resistances are converted into their equivalent lengths, and then the total length of all the resistances is added to the pipe geometrical length.

Connections A and B are hydraulic conserving ports. The block positive direction is from port A to port B. This means that the flow rate is positive if fluid flows from A to B, and the pressure loss is determined as p = p_A - p_B.

View source for Hydraulic Resistive Tube

Parameters

Tube cross section type:	Circular	
Tube internal diameter:	0.01	m
Geometrical shape factor:	64	
Tube length:	5	m
Aggregate equivalent length of local resistances:	1	m
Internal surface roughness height:	15e-6	m
Laminar flow upper margin:	2000	
Turbulent flow lower margin:	4000	



Block Parameters: Hydraulic Resistive Tube

Hydraulic Resistive Tube

This block models hydraulic pipelines with circular and noncircular cross sections and accounts for resistive property only. To account for local resistances such as bends, fittings, inlet and outlet losses, and so on, all tl resistances are converted into their equivalent lengths, and then the total length of all the resistances is add to the pipe geometrical length.

Connections A and B are hydraulic conserving ports. The block positive direction is from port A to port B. Thi means that the flow rate is positive if fluid flows from A to B, and the pressure loss is determined as p = p_A p_B.

View source for Hydraulic Resistive Tube

Parameters

Tube cross section type: Noncircular

Noncircular tube cross-sectional area: 1e-4 m^2

Noncircular tube hydraulic diameter: 1.12e-2 m

Geometrical shape factor: 64

Tube length: 5 m

Aggregate equivalent length of local resistances: 1 m

Internal surface roughness height: 15e-6 m

Laminar flow upper margin: 2000

Turbulent flow lower margin: 4000

Tube cross section type

The type of tube cross section: `Circular` or `Noncircular`. For a circular tube, you specify its internal diameter. For a noncircular tube, you specify its hydraulic diameter and tube cross-sectional area. The default value of the parameter is `Circular`.

Tube internal diameter

Tube internal diameter. The parameter is used if **Tube cross section type** is set to `Circular`. The default value is `0.01` m.

Noncircular tube cross-sectional area

Tube cross-sectional area. The parameter is used if **Tube cross section type** is set to `Noncircular`. The default value is `1e-4` m^2.

Noncircular tube hydraulic diameter

Hydraulic diameter of the tube cross section. The parameter is used if **Tube cross section type** is set to `Noncircular`. The default value is `0.0112` m.

Geometrical shape factor

Used for computing friction factor at laminar flow. The shape of the tube cross section determines the value. For a tube with a noncircular cross section, set the factor to an appropriate value, for example, 56 for a square, 96 for concentric annulus, 62 for rectangle (2:1), and so on [1]. The default value is `64`, which corresponds to a tube with a circular cross section.

Tube length

Tube geometrical length. The default value is `5` m.

Aggregate equivalent length of local resistances

This parameter represents total equivalent length of all local resistances associated with the tube. You can account for the pressure loss caused by local resistances, such as bends, fittings, armature, inlet/outlet losses, and so on, by adding to the pipe geometrical length an aggregate equivalent length of all the local resistances. The default value is `1` m.

Internal surface roughness height
Roughness height on the tube internal surface. The parameter is typically provided in data sheets or manufacturer's catalogs. The default value is `1.5e-5` m, which corresponds to drawn tubing.

Laminar flow upper margin
Specifies the Reynolds number at which the laminar flow regime is assumed to start converting into turbulent. Mathematically, this is the maximum Reynolds number at fully developed laminar flow. The default value is `2000`.

Turbulent flow lower margin
Specifies the Reynolds number at which the turbulent flow regime is assumed to be fully developed. Mathematically, this is the minimum Reynolds number at turbulent flow. The default value is `4000`.

Restricted Parameters

When your model is in Restricted editing mode, you cannot modify the following parameter:

- **Tube cross section type**

All other block parameters are available for modification. The actual set of modifiable block parameters depends on the value of the **Tube cross section type** parameter at the time the model entered Restricted mode.

Global Parameters

Parameters determined by the type of working fluid:

- **Fluid density**
- **Fluid kinematic viscosity**

Use the Hydraulic Fluid block or the Custom Hydraulic Fluid block to specify the fluid properties.

Ports

The block has the following ports:

A

Hydraulic conserving port associated with the tube inlet.

B

Hydraulic conserving port associated with the tube outlet.

References

[1] White, F.M., *Viscous Fluid Flow*, McGraw-Hill, 1991

See Also

Linear Hydraulic Resistance

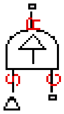
Purpose

Ideal angular velocity source in mechanical rotational systems

Library

Mechanical Sources

Description

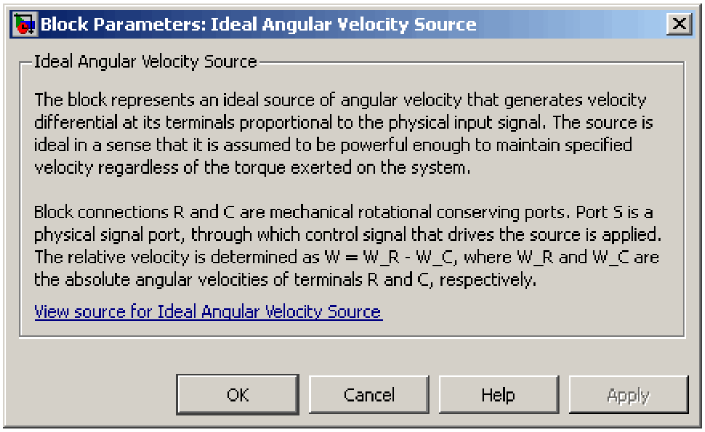


The Ideal Angular Velocity Source block represents an ideal source of angular velocity that generates velocity differential at its terminals proportional to the input physical signal. The source is ideal in a sense that it is assumed to be powerful enough to maintain specified velocity regardless of the torque exerted on the system.

Connections R and C are mechanical rotational conserving ports. Port S is a physical signal port, through which the control signal that drives the source is applied. The relative velocity (velocity differential) across the source is directly proportional to the signal at the control port S. The entire variety of Simulink signal sources can be used to generate the desired velocity variation profile.

The block positive direction is from port R to port C. This means that the velocity is measured as $\omega = \omega_R - \omega_C$, where ω_R, ω_C are the absolute angular velocities at ports R and C, respectively, and torque through the source is positive if it is directed from R to C. The power generated by the source is negative if the source delivers energy to port R.

Dialog Box and Parameters



Block Parameters: Ideal Angular Velocity Source

Ideal Angular Velocity Source

The block represents an ideal source of angular velocity that generates velocity differential at its terminals proportional to the physical input signal. The source is ideal in a sense that it is assumed to be powerful enough to maintain specified velocity regardless of the torque exerted on the system.

Block connections R and C are mechanical rotational conserving ports. Port S is a physical signal port, through which control signal that drives the source is applied. The relative velocity is determined as W = W_R - W_C, where W_R and W_C are the absolute angular velocities of terminals R and C, respectively.

View source for Ideal Angular Velocity Source

The block has no parameters.

Ports

The block has the following ports:

`R`

Mechanical rotational conserving port.

`C`

Mechanical rotational conserving port associated with the source reference point (case).

`S`

Physical signal input port, through which the control signal that drives the source is applied.

See Also

Ideal Force Source

Ideal Torque Source

Ideal Translational Velocity Source

Ideal Force Sensor

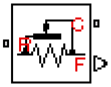
Purpose

Force sensor in mechanical translational systems

Library

Mechanical Sensors

Description

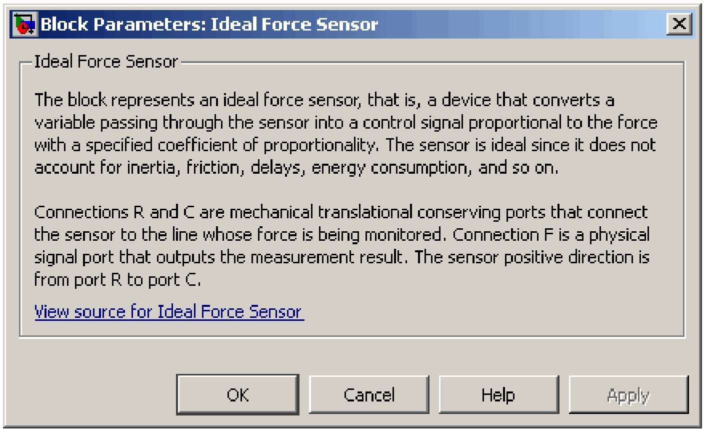


The Ideal Force Sensor block represents a device that converts a variable passing through the sensor into a control signal proportional to the force. The sensor is ideal since it does not account for inertia, friction, delays, energy consumption, and so on.

Connections R and C are mechanical translational conserving ports that connect the block to the line where force is being monitored. Connection F is a physical signal port that outputs the measurement result.

The block positive direction is from port R to port C. This means that positive force applied to port R (the sensor positive probe) generates a positive output signal.

Dialog Box and Parameters



The block represents an ideal force sensor, that is, a device that converts a variable passing through the sensor into a control signal proportional to the force with a specified coefficient of proportionality. The sensor is ideal since it does not account for inertia, friction, delays, energy consumption, and so on.

Connections R and C are mechanical translational conserving ports that connect the sensor to the line whose force is being monitored. Connection F is a physical signal port that outputs the measurement result. The sensor positive direction is from port R to port C.

The block has no parameters.

Ports

The block has the following ports:

R

Mechanical translational conserving port associated with the sensor positive probe.

C

Mechanical translational conserving port associated with the sensor negative (reference) probe.

F

Physical signal output port for force.

See Also

Ideal Rotational Motion Sensor

Ideal Torque Sensor

Ideal Translational Motion Sensor

Ideal Force Source

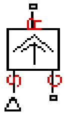
Purpose

Ideal source of mechanical energy that generates force proportional to the input signal

Library

Mechanical Sources

Description

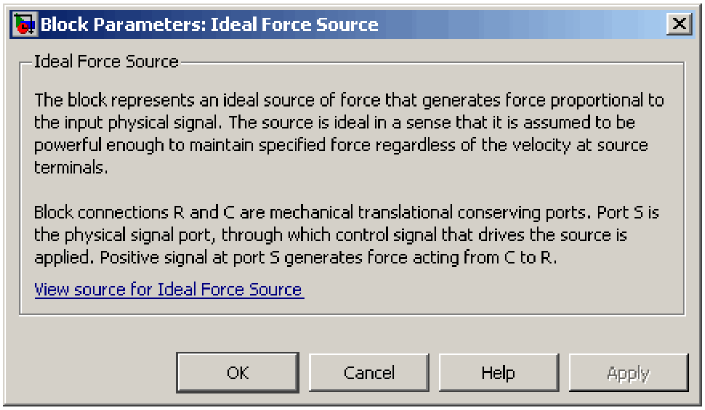


The Ideal Force Source block represents an ideal source of mechanical energy that generates force proportional to the input physical signal. The source is ideal in a sense that it is assumed to be powerful enough to maintain specified force at its output regardless of the velocity at source terminals.

Connections R and C are mechanical translational conserving ports. Port S is a physical signal port, through which the control signal that drives the source is applied. You can use the entire variety of Simulink signal sources to generate the desired force variation profile. Positive signal at port S generates force acting from C to R. The force generated by the source is directly proportional to the signal at the control port S.

The block positive direction is from port C to port R. This means that the force is positive if it acts in the direction from C to R. The relative velocity is determined as $v = v_C - v_R$, where v_R, v_C are the absolute velocities at ports R and C, respectively, and it is negative if velocity at port R is greater than that at port C. The power generated by the source is negative if the source delivers energy to port R.

Dialog Box and Parameters



Block Parameters: Ideal Force Source

Ideal Force Source

The block represents an ideal source of force that generates force proportional to the input physical signal. The source is ideal in a sense that it is assumed to be powerful enough to maintain specified force regardless of the velocity at source terminals.

Block connections R and C are mechanical translational conserving ports. Port S is the physical signal port, through which control signal that drives the source is applied. Positive signal at port S generates force acting from C to R.

View source for Ideal Force Source

The block has no parameters.

Ports

The block has the following ports:

`R`

Mechanical translational conserving port.

`C`

Mechanical translational conserving port associated with the source reference point (case).

`S`

Physical signal input port, through which the control signal that drives the source is applied.

See Also

Ideal Angular Velocity Source

Ideal Torque Source

Ideal Translational Velocity Source

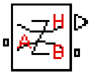
Purpose

Ideal heat flow meter

Library

Thermal Sensors

Description

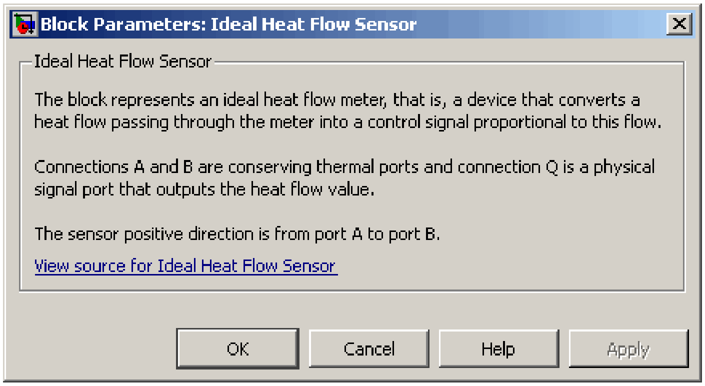


The Ideal Heat Flow Sensor block represents an ideal heat flow meter, that is, a device that converts a heat flow passing through the meter into a control signal proportional to this flow. The meter must be connected in series with the component whose heat flow is being monitored.

Connections A and B are thermal conserving ports. Port Q is a physical signal port that outputs the heat flow value.

The block positive direction is from port A to port B.

Dialog Box and Parameters


Block Parameters: Ideal Heat Flow Sensor

Ideal Heat Flow Sensor

The block represents an ideal heat flow meter, that is, a device that converts a heat flow passing through the meter into a control signal proportional to this flow.

Connections A and B are conserving thermal ports and connection Q is a physical signal port that outputs the heat flow value.

The sensor positive direction is from port A to port B.

View source for Ideal Heat Flow Sensor

The block has no parameters.

Ports

The block has the following ports:

A

Thermal conserving port associated with the sensor positive probe.

B

Thermal conserving port associated with the sensor negative probe.

`Q`

Physical signal output port for heat flow.

See Also

Ideal Heat Flow Source

Ideal Temperature Sensor

Ideal Temperature Source

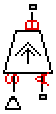
Purpose

Ideal source of thermal energy, characterized by heat flow

Library

Thermal Sources

Description

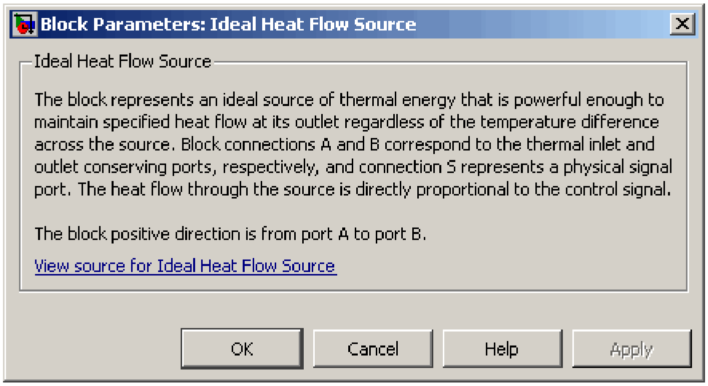


The Ideal Heat Flow Source block represents an ideal source of thermal energy that is powerful enough to maintain specified heat flow at its outlet regardless of the temperature difference across the source.

Connections A and B are thermal conserving ports corresponding to the source inlet and outlet, respectively. Port S is a physical signal port, through which the control signal that drives the source is applied. You can use the entire variety of Simulink signal sources to generate the desired heat flow variation profile. The heat flow through the source is directly proportional to the signal at the control port S.

The block positive direction is from port A to port B. This means that positive signal at port S generates heat flow in the direction from A to B.

Dialog Box and Parameters



Block Parameters: Ideal Heat Flow Source

Ideal Heat Flow Source

The block represents an ideal source of thermal energy that is powerful enough to maintain specified heat flow at its outlet regardless of the temperature difference across the source. Block connections A and B correspond to the thermal inlet and outlet conserving ports, respectively, and connection S represents a physical signal port. The heat flow through the source is directly proportional to the control signal.

The block positive direction is from port A to port B.

View source for Ideal Heat Flow Source

The block has no parameters.

Ports

The block has the following ports:

A

Thermal conserving port associated with the source inlet.

B

Thermal conserving port associated with the source outlet.

S

Physical signal input port, through which the control signal that drives the source is applied.

See Also

Ideal Heat Flow Sensor

Ideal Temperature Sensor

Ideal Temperature Source

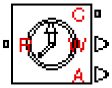
Purpose

Motion sensor in mechanical rotational systems

Library

Mechanical Sensors

Description



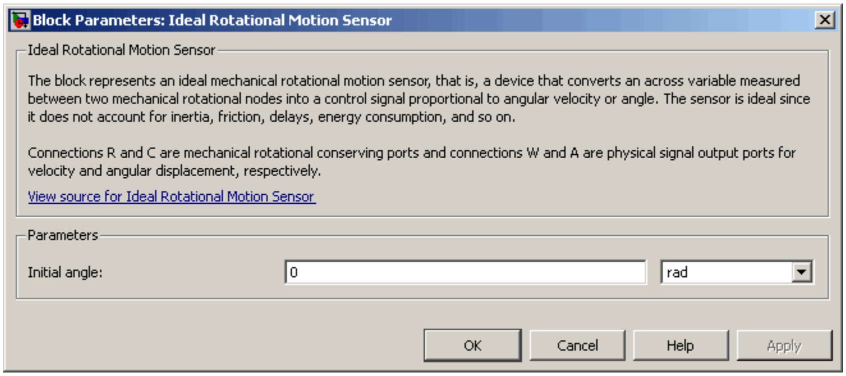
The Ideal Rotational Motion Sensor block represents an ideal mechanical rotational motion sensor, that is, a device that converts an across variable measured between two mechanical rotational nodes into a control signal proportional to angular velocity or angle. You can specify the initial angular position (offset) as a block parameter.

The sensor is ideal since it does not account for inertia, friction, delays, energy consumption, and so on.

Connections R and C are mechanical rotational conserving ports that connect the block to the nodes whose motion is being monitored. Connections W and A are physical signal output ports for velocity and angular displacement, respectively.

The block positive direction is from port R to port C. This means that the velocity is measured as $\omega = \omega_R - \omega_C$, where ω_R, ω_C are the absolute angular velocities at ports R and C, respectively.

Dialog Box and Parameters


Block Parameters: Ideal Rotational Motion Sensor

Ideal Rotational Motion Sensor

The block represents an ideal mechanical rotational motion sensor, that is, a device that converts an across variable measured between two mechanical rotational nodes into a control signal proportional to angular velocity or angle. The sensor is ideal since it does not account for inertia, friction, delays, energy consumption, and so on.

Connections R and C are mechanical rotational conserving ports and connections W and A are physical signal output ports for velocity and angular displacement, respectively.

View source for Ideal Rotational Motion Sensor

Parameters

Initial angle: 0 rad

Initial angle

Sensor initial angle, or offset (rad). The default value is 0.

Ports

The block has the following ports:

`R`

Mechanical rotational conserving port associated with the sensor positive probe.

`C`

Mechanical rotational conserving port associated with the sensor negative (reference) probe.

`W`

Physical signal output port for angular velocity.

`A`

Physical signal output port for angular displacement.

See Also

Ideal Force Sensor

Ideal Torque Sensor

Ideal Translational Motion Sensor

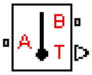
Purpose

Ideal temperature sensor

Library

Thermal Sensors

Description

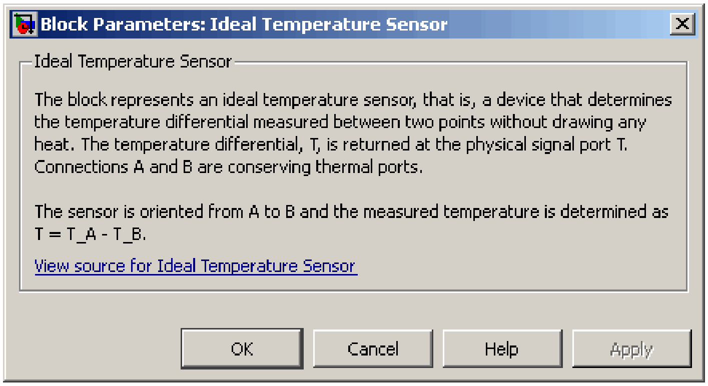


The Ideal Temperature Sensor block represents an ideal temperature sensor, that is, a device that determines the temperature differential measured between two points without drawing any heat.

Connections A and B are thermal conserving ports that connect to the two points where temperature is being monitored. Port T is a physical signal port that outputs the temperature differential value.

The block positive direction is from port A to port B. The measured temperature is determined as $T = T_A - T_B$.

Dialog Box and Parameters



The block represents an ideal temperature sensor, that is, a device that determines the temperature differential measured between two points without drawing any heat. The temperature differential, T, is returned at the physical signal port T. Connections A and B are conserving thermal ports.

The sensor is oriented from A to B and the measured temperature is determined as T = T_A - T_B.

View source for Ideal Temperature Sensor

The block has no parameters.

Ports

The block has the following ports:

A

Thermal conserving port associated with the sensor positive probe.

B

Thermal conserving port associated with the sensor negative probe.

T

Physical signal output port for temperature.

See Also

Ideal Heat Flow Sensor

Ideal Heat Flow Source

Ideal Temperature Source

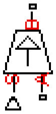
Purpose

Ideal source of thermal energy, characterized by temperature

Library

Thermal Sources

Description

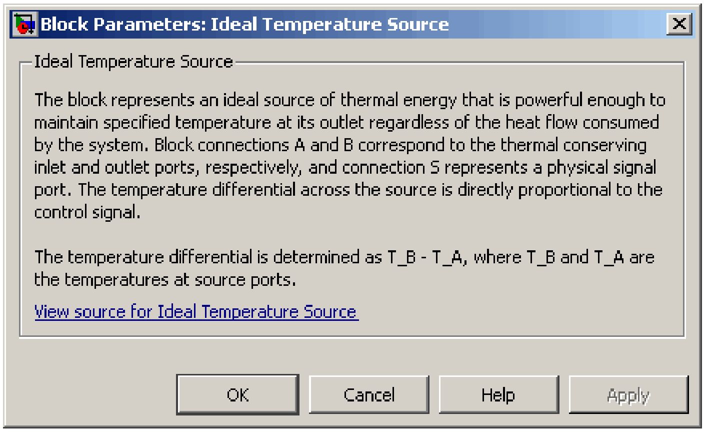


The Ideal Temperature Source block represents an ideal source of thermal energy that is powerful enough to maintain specified temperature at its outlet regardless of the heat flow consumed by the system.

Connections A and B are thermal conserving ports corresponding to the source inlet and outlet, respectively. Port S is a physical signal port, through which the control signal that drives the source is applied. You can use the entire variety of Simulink signal sources to generate the desired heat flow variation profile. The temperature differential across the source is directly proportional to the signal at the control port S.

The block positive direction is from port A to port B. This means that the temperature differential is determined as $T_B - T_A$, where T_B and T_A are the temperatures at source ports.

Dialog Box and Parameters



Block Parameters: Ideal Temperature Source

Ideal Temperature Source

The block represents an ideal source of thermal energy that is powerful enough to maintain specified temperature at its outlet regardless of the heat flow consumed by the system. Block connections A and B correspond to the thermal conserving inlet and outlet ports, respectively, and connection S represents a physical signal port. The temperature differential across the source is directly proportional to the control signal.

The temperature differential is determined as T_B - T_A, where T_B and T_A are the temperatures at source ports.

View source for Ideal Temperature Source

The block has no parameters.

Ports

The block has the following ports:

A

Thermal conserving port associated with the source inlet.

B

Thermal conserving port associated with the source outlet.

S

Physical signal input port, through which the control signal that drives the source is applied.

See Also

Ideal Heat Flow Sensor

Ideal Heat Flow Source

Ideal Temperature Sensor

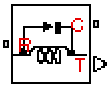
Purpose

Torque sensor in mechanical rotational systems

Library

Mechanical Sensors

Description

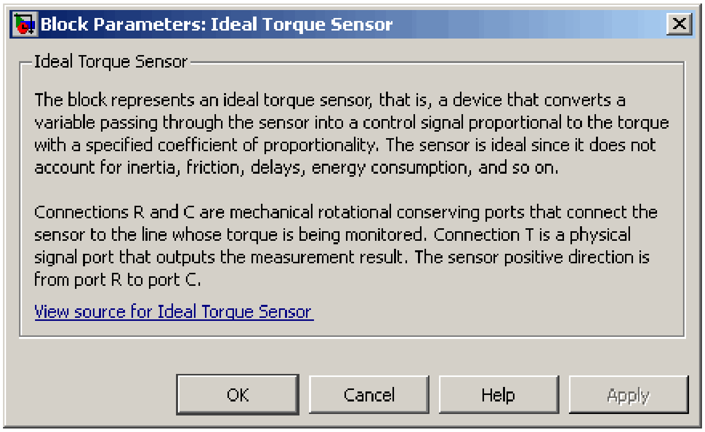


The Ideal Torque Sensor block represents a device that converts a variable passing through the sensor into a control signal proportional to the torque. The sensor is ideal because it does not account for inertia, friction, delays, energy consumption, and so on.

Connections R and C are mechanical rotational conserving ports that connect the block to the line where torque is being monitored. Connection T is a physical signal port that outputs the measurement result.

The block positive direction is from port R to port C.

Dialog Box and Parameters



Block Parameters: Ideal Torque Sensor

Ideal Torque Sensor

The block represents an ideal torque sensor, that is, a device that converts a variable passing through the sensor into a control signal proportional to the torque with a specified coefficient of proportionality. The sensor is ideal since it does not account for inertia, friction, delays, energy consumption, and so on.

Connections R and C are mechanical rotational conserving ports that connect the sensor to the line whose torque is being monitored. Connection T is a physical signal port that outputs the measurement result. The sensor positive direction is from port R to port C.

View source for Ideal Torque Sensor

The block has no parameters.

Ports

The block has the following ports:

R

Mechanical rotational conserving port associated with the sensor positive probe.

C

Mechanical rotational conserving port associated with the sensor negative (reference) probe.

T

Physical signal output port for torque.

See Also

Ideal Force Sensor

Ideal Rotational Motion Sensor

Ideal Translational Motion Sensor

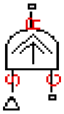
Purpose

Ideal source of mechanical energy that generates torque proportional to the input signal

Library

Mechanical Sources

Description

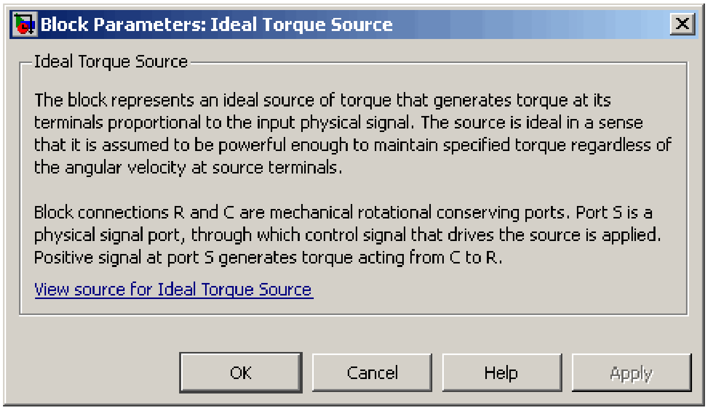


The Ideal Torque Source block represents an ideal source of mechanical energy that generates torque proportional to the input physical signal. The source is ideal in a sense that it is assumed to be powerful enough to maintain specified torque regardless of the angular velocity at source terminals.

Connections R and C are mechanical rotational conserving ports. Port S is a physical signal port, through which the control signal that drives the source is applied. You can use the entire variety of Simulink signal sources to generate the desired torque variation profile. Positive signal at port S generates torque acting from C to R. The torque generated by the source is directly proportional to the signal at the control port S.

The block positive direction is from port C to port R. This means that the torque is positive if it acts in the direction from C to R. The relative velocity is determined as $\omega = \omega_R - \omega_C$, where ω_R, ω_C are the absolute angular velocities at ports R and C, respectively, and it is negative if velocity at port R is greater than that at port C. The power generated by the source is negative if the source delivers energy to port R.

Dialog Box and Parameters



Ideal Torque Source

The block represents an ideal source of torque that generates torque at its terminals proportional to the input physical signal. The source is ideal in a sense that it is assumed to be powerful enough to maintain specified torque regardless of the angular velocity at source terminals.

Block connections R and C are mechanical rotational conserving ports. Port S is a physical signal port, through which control signal that drives the source is applied. Positive signal at port S generates torque acting from C to R.

View source for Ideal Torque Source

The block has no parameters.

Ports

The block has the following ports:

`R`
: Mechanical rotational conserving port.

`C`
: Mechanical rotational conserving port associated with the source reference point (case).

`S`
: Physical signal input port, through which the control signal that drives the source is applied.

See Also

Ideal Angular Velocity Source

Ideal Force Source

Ideal Translational Velocity Source

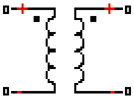
Purpose

Ideal transformer in electrical systems

Library

Electrical Elements

Description



The Ideal Transformer block models an ideal power-conserving transformer, described with the following equations:

$$V1 = N \cdot V2$$

$$I2 = N \cdot I1$$

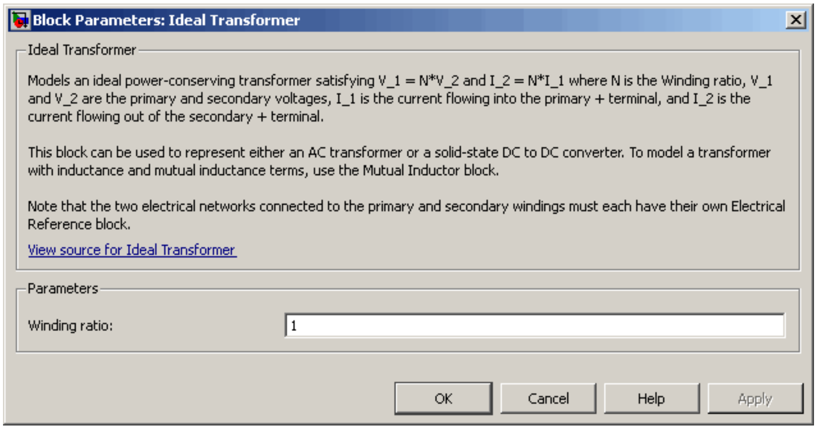
where

`V1`	Primary voltage
`V2`	Secondary voltage
`I1`	Current flowing into the primary + terminal
`I2`	Current flowing out of the secondary + terminal
`N`	Winding ratio

This block can be used to represent either an AC transformer or a solid-state DC to DC converter. To model a transformer with inductance and mutual inductance terms, use the Mutual Inductor block.

The two electrical networks connected to the primary and secondary windings must each have their own Electrical Reference block.

Dialog Box and Parameters



Block Parameters: Ideal Transformer

Ideal Transformer

Models an ideal power-conserving transformer satisfying V_1 = N*V_2 and I_2 = N*I_1 where N is the Winding ratio, V_1 and V_2 are the primary and secondary voltages, I_1 is the current flowing into the primary + terminal, and I_2 is the current flowing out of the secondary + terminal.

This block can be used to represent either an AC transformer or a solid-state DC to DC converter. To model a transformer with inductance and mutual inductance terms, use the Mutual Inductor block.

Note that the two electrical networks connected to the primary and secondary windings must each have their own Electrical Reference block.

View source for Ideal Transformer

Parameters

Winding ratio: 1

Winding ratio

Winding ratio of the transformer, or ratio of primary coil turns to secondary coil turns. The default value is 1.

Ports

The block has four electrical conserving ports. Polarity is indicated by the + and – signs.

See Also

Mutual Inductor

Ideal Translational Motion Sensor

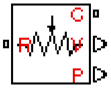
Purpose

Motion sensor in mechanical translational systems

Library

Mechanical Sensors

Description



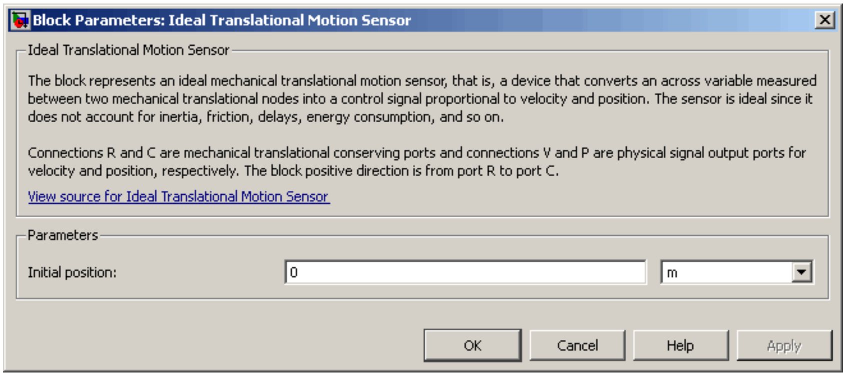
The Ideal Translational Motion Sensor block represents a device that converts an across variable measured between two mechanical translational nodes into a control signal proportional to velocity or position. You can specify the initial position (offset) as a block parameter.

The sensor is ideal since it does not account for inertia, friction, delays, energy consumption, and so on.

Connections R and C are mechanical translational conserving ports that connect the block to the nodes whose motion is being monitored. Connections V and P are physical signal output ports for velocity and position, respectively.

The block positive direction is from port R to port C. This means that the velocity is measured as $v = v_R - v_C$, where v_R,v_C are the absolute velocities at ports R and C, respectively.

Dialog Box and Parameters



Block Parameters: Ideal Translational Motion Sensor

Ideal Translational Motion Sensor

The block represents an ideal mechanical translational motion sensor, that is, a device that converts an across variable measured between two mechanical translational nodes into a control signal proportional to velocity and position. The sensor is ideal since it does not account for inertia, friction, delays, energy consumption, and so on.

Connections R and C are mechanical translational conserving ports and connections V and P are physical signal output ports for velocity and position, respectively. The block positive direction is from port R to port C.

View source for Ideal Translational Motion Sensor

Parameters

Initial position: 0 m

Initial position

Sensor initial position, or offset (m). The default value is `0`.

Ports

The block has the following ports:

`R`
: Mechanical translational conserving port associated with the sensor positive probe.

`C`
: Mechanical translational conserving port associated with the sensor negative (reference) probe.

`V`
: Physical signal output port for velocity.

`P`
: Physical signal output port for position.

See Also

Ideal Force Sensor

Ideal Rotational Motion Sensor

Ideal Torque Sensor

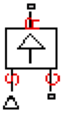
Purpose

Ideal velocity source in mechanical translational systems

Library

Mechanical Sources

Description

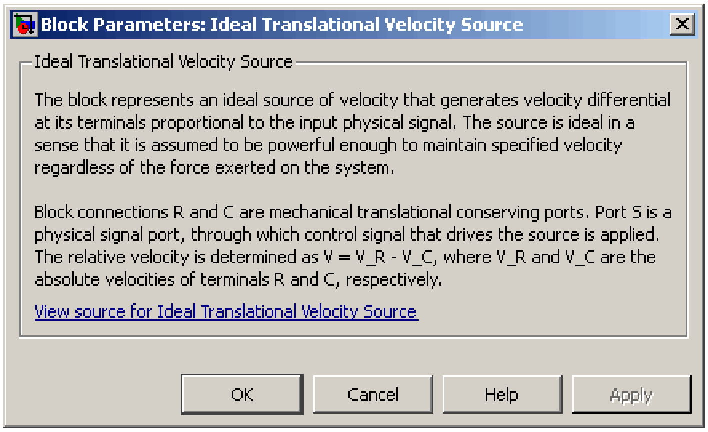


The Ideal Translational Velocity Source block represents an ideal source of velocity that generates velocity differential at its terminals proportional to the input physical signal. The source is ideal in a sense that it is assumed to be powerful enough to maintain specified velocity regardless of the force exerted on the system.

Connections R and C are mechanical translational conserving ports. Port S is a physical signal port, through which the control signal that drives the source is applied. The relative velocity (velocity differential) across the source is directly proportional to the signal at the control port S. The entire variety of Simulink signal sources can be used to generate the desired velocity variation profile.

The block positive direction is from port R to port C. This means that the velocity is measured as $v = v_R - v_C$, where v_R, v_C are the absolute velocities at ports R and C, respectively, and force through the source is negative if it is acts from C to R. The power generated by the source is negative if the source delivers energy to port R.

Dialog Box and Parameters



Ideal Translational Velocity Source

The block represents an ideal source of velocity that generates velocity differential at its terminals proportional to the input physical signal. The source is ideal in a sense that it is assumed to be powerful enough to maintain specified velocity regardless of the force exerted on the system.

Block connections R and C are mechanical translational conserving ports. Port S is a physical signal port, through which control signal that drives the source is applied. The relative velocity is determined as V = V_R - V_C, where V_R and V_C are the absolute velocities of terminals R and C, respectively.

View source for Ideal Translational Velocity Source

The block has no parameters.

Ports

The block has the following ports:

`R`
: Mechanical translational conserving port.

`C`
: Mechanical translational conserving port associated with the source reference point (case).

`S`
: Physical signal input port, through which the control signal that drives the source is applied.

See Also

Ideal Angular Velocity Source

Ideal Force Source

Ideal Torque Source

Purpose

Linear inductor in electrical systems

Library

Electrical Elements

Description

The Inductor block models a linear inductor, described with the following equation:

$$V = L\frac{dI}{dt}$$

where

I	Current
V	Voltage
L	Inductance
t	Time

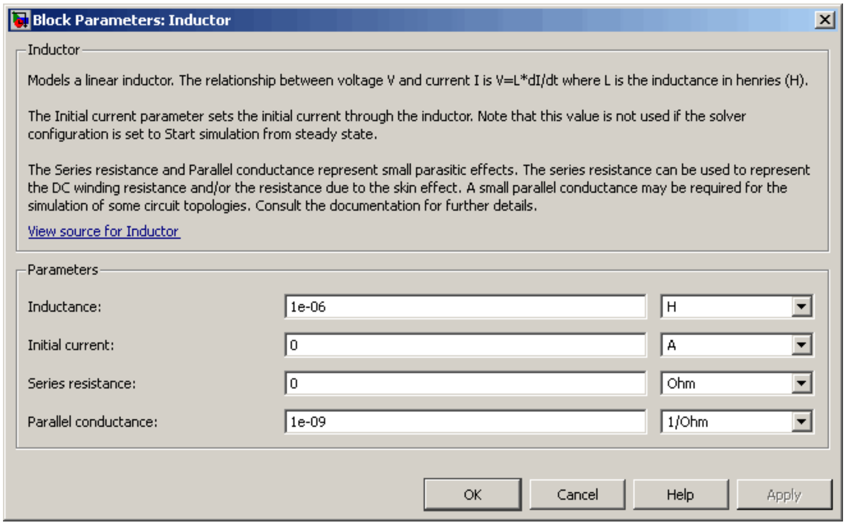
The **Initial current** parameter sets the initial current through the inductor.

Note This value is not used if the solver configuration is set to **Start simulation from steady state**.

The **Series resistance** and **Parallel conductance** parameters represent small parasitic effects. The series resistance can be used to represent the DC winding resistance or the resistance due to the skin effect. Simulation of some circuits may require the presence of a small parallel conductance. For more information, see “Modeling Best Practices”.

Connections + and – are conserving electrical ports corresponding to the positive and negative terminals of the inductor, respectively. The current is positive if it flows from positive to negative, and the voltage across the inductor is equal to the difference between the voltage at the positive and the negative terminal, *V*(+) – *V*(–).

Dialog Box and Parameters



Block Parameters: Inductor

Inductor

Models a linear inductor. The relationship between voltage V and current I is V=L*dI/dt where L is the inductance in henries (H).

The Initial current parameter sets the initial current through the inductor. Note that this value is not used if the solver configuration is set to Start simulation from steady state.

The Series resistance and Parallel conductance represent small parasitic effects. The series resistance can be used to represent the DC winding resistance and/or the resistance due to the skin effect. A small parallel conductance may be required for the simulation of some circuit topologies. Consult the documentation for further details.

View source for Inductor

Parameters

Inductance:	1e-06	H
Initial current:	0	A
Series resistance:	0	Ohm
Parallel conductance:	1e-09	1/Ohm

Inductance

Inductance, in henries. The default value is 1 µH.

Initial current

Initial current through the inductor. This parameter is not used if the solver configuration is set to **Start simulation from steady state**. The default value is 0.

Series resistance

Represents small parasitic effects. The series resistance can be used to represent the DC winding resistance. The default value is 0.

Parallel conductance

Represents small parasitic effects. The parallel conductance across the inductor can be used to model insulation conductance. Simulation of some circuits may require the presence of a small parallel conductance. The default value is 1e-9 1/Ω.

Ports

The block has the following ports:

+

Electrical conserving port associated with the inductor positive terminal.

-

Electrical conserving port associated with the inductor negative terminal.

Inertia

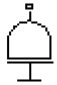
Purpose

Ideal mechanical rotational inertia

Library

Mechanical Rotational Elements

Description



The Inertia block represents an ideal mechanical rotational inertia, described with the following equation:

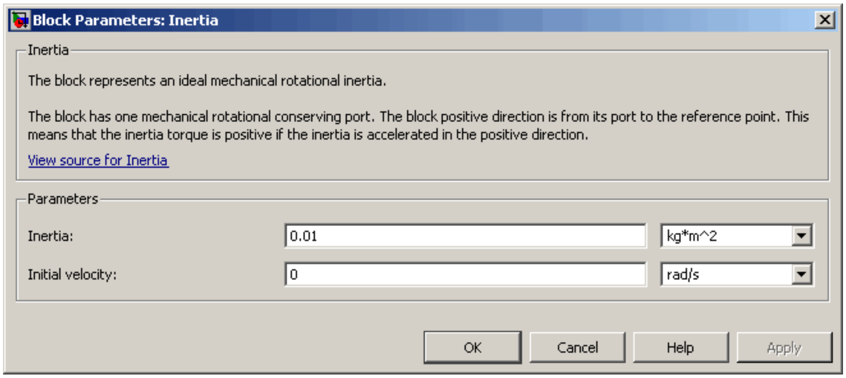
$$T = J\frac{d\omega}{dt}$$

where

T	Inertia torque
J	Inertia
ω	Angular velocity
t	Time

The block has one mechanical rotational conserving port. The block positive direction is from its port to the reference point. This means that the inertia torque is positive if inertia is accelerated in positive direction.

Dialog Box and Parameters



Block Parameters: Inertia

Inertia

The block represents an ideal mechanical rotational inertia.

The block has one mechanical rotational conserving port. The block positive direction is from its port to the reference point. This means that the inertia torque is positive if the inertia is accelerated in the positive direction.

View source for Inertia

Parameters

Inertia: 0.01 kg*m^2

Initial velocity: 0 rad/s

Inertia

Inertia. The default value is `0.001` kg*m^2.

Initial velocity

Initial angular velocity of the inertia. This parameter specifies the initial condition for use in computing the block's initial state at the beginning of a simulation run. For more information, see "Initial Conditions Computation". The default value is `0`.

Ports

The block has one mechanical rotational conserving port, associated with the inertia connection to the system.

See Also

Mass

Lever

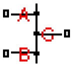
Purpose

Generic mechanical lever

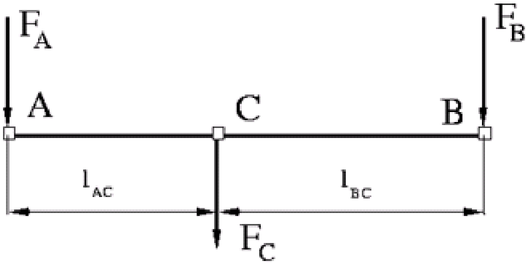
Library

Mechanisms

Description



The Lever block represents a mechanical lever in its generic form, known as a free or summing lever, shown in the following schematic.



The summing lever equations are derived with the assumption of small angle deviation from initial position:

$$v_C = K_{AC} \cdot v_A + K_{BC} \cdot v_B$$

$$F_A = K_{AC} \cdot F_C$$

$$F_B = K_{BC} \cdot F_C$$

$$K_{AC} = \frac{l_{BC}}{l_{AC} + l_{BC}}$$

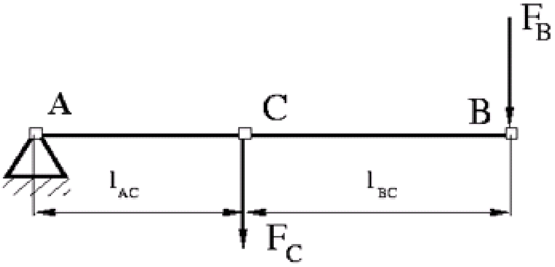
$$K_{BC} = \frac{l_{AC}}{l_{AC} + l_{BC}}$$

where

v_A, v_B, v_C	Lever joints velocities
F_A, F_B, F_C	Lever joints forces
l_{AC}, l_{BC}	Arm lengths

The above equations were derived with the assumption that the lever sums forces and motions at node C. The assumption was arbitrary and does not impose any limitations on how the forces or motions are applied to the lever. In other words, any of the lever nodes can be “input” or “output” nodes, depending on the value of the force. Moreover, any of the block nodes can be connected to the reference point, thus converting a three-node lever into a first-class lever, with the fulcrum at the end, or a second-class lever, with the fulcrum in the middle.

The following illustration shows a schematic of a two-node first-class lever, with the fulcrum at node A.

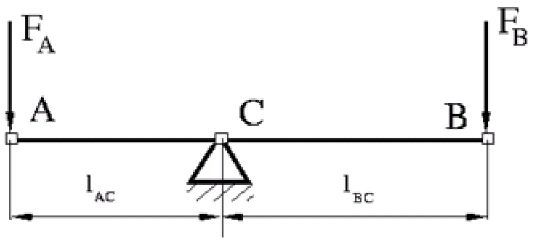


It is described with the following equations:

$$v_C = K_{BC} \cdot v_B$$

$$F_B = K_{BC} \cdot F_C$$

The next illustration shows a schematic of a second-class lever, with the fulcrum in the middle.



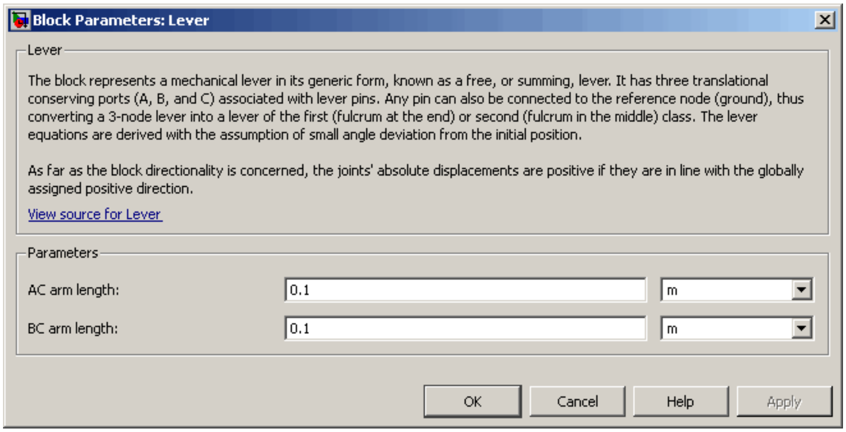
It is described with the following equations:

$$v_A = -\frac{l_{AC}}{l_{BC}} \cdot v_B$$

$$F_B = -\frac{l_{AC}}{l_{BC}} \cdot F_A$$

As far as the block directionality is concerned, the joints' absolute displacements are positive if they are in line with the globally assigned positive direction.

Dialog Box and Parameters



Block Parameters: Lever

Lever

The block represents a mechanical lever in its generic form, known as a free, or summing, lever. It has three translational conserving ports (A, B, and C) associated with lever pins. Any pin can also be connected to the reference node (ground), thus converting a 3-node lever into a lever of the first (fulcrum at the end) or second (fulcrum in the middle) class. The lever equations are derived with the assumption of small angle deviation from the initial position.

As far as the block directionality is concerned, the joints' absolute displacements are positive if they are in line with the globally assigned positive direction.

View source for Lever

Parameters

AC arm length:	0.1	m
BC arm length:	0.1	m

AC arm length

Arm length between nodes A and C. The default value is `0.1` m.

BC arm length

Arm length between nodes B and C. The default value is `0.1` m.

Ports

The block has the following ports:

`A`

Mechanical translational conserving port associated with the node A of the lever.

`B`

Mechanical translational conserving port associated with the node B of the lever.

`C`

Mechanical translational conserving port associated with the node C of the lever.

Examples

The Linkage Mechanism example illustrates the use of the Lever block in three different modes. Linkages L_1 and L_4 simulate first-class levers with the fulcrum at the end. Linkage L_2 represents a summing lever. Linkage L_3 simulates a second-class lever with the fulcrum in the middle.

Linear Hydraulic Resistance

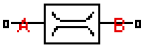
Purpose

Hydraulic pipeline with linear resistance losses

Library

Hydraulic Elements

Description

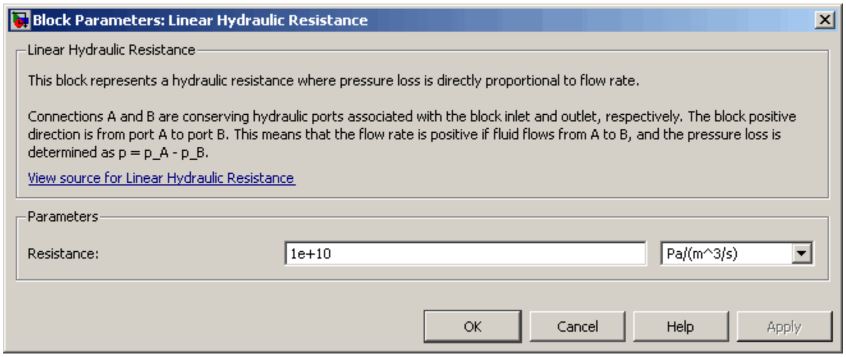


The Linear Hydraulic Resistance block represents a hydraulic resistance where pressure loss is directly proportional to flow rate. This block can be useful at preliminary stages of development, or as a powerful means to speed up the simulation, especially if the flow rate varies insignificantly with respect to the operating point.

Connections A and B are conserving hydraulic ports associated with the block inlet and outlet, respectively.

The block positive direction is from port A to port B. This means that the flow rate is positive if fluid flows from A to B, and the pressure loss is determined as $p = p_A - p_B$.

Dialog Box and Parameters


Block Parameters: Linear Hydraulic Resistance

Linear Hydraulic Resistance

This block represents a hydraulic resistance where pressure loss is directly proportional to flow rate.

Connections A and B are conserving hydraulic ports associated with the block inlet and outlet, respectively. The block positive direction is from port A to port B. This means that the flow rate is positive if fluid flows from A to B, and the pressure loss is determined as p = p_A - p_B.

View source for Linear Hydraulic Resistance

Parameters

Resistance: 1e+10 Pa/(m^3/s)

Resistance

The linear resistance coefficient. The default value is `10e9` Pa/(m^3/s).

Ports

The block has the following ports:

A

Hydraulic conserving port associated with the resistance inlet.

B

Hydraulic conserving port associated with the resistance outlet.

See Also

Hydraulic Resistive Tube

Local Restriction (TL)

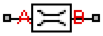
Purpose

Narrow opening with constant cross-sectional area

Library

Thermal Liquid/Elements

Description



The Local Restriction (TL) block represents a narrow opening with constant cross-sectional area. The restriction causes a pressure drop and temperature gain in the liquid flowing through it. Common restrictions include valves and orifices.

To compute the pressure drop across the restriction, the block uses a discharge coefficient. This coefficient relates the pressure drop to the kinetic energy of the upstream liquid. The restriction is adiabatic. It does not exchange heat with the environment.

The liquid volume in the local restriction is generally small. With this assumption, the dynamic compressibility and thermal inertia of the liquid are negligible. The block ignores both of these effects.

The following equations govern the behavior of the local restriction:

$$0 = \dot{m}_A + \dot{m}_B$$

where

A	Cross-sectional area of the pipes adjacent to the restriction
A_r	Cross-sectional area of the local restriction
C_d	Flow discharge coefficient
p_Int	Liquid pressure in the local restriction
$\dot{m}_A$, $\dot{m}_B$	Mass flow rates of liquid into the local restriction at inlets A and B
v_A, v_B	Liquid velocity into the local restriction at inlets A and B
Re	Reynolds number
Re_c	Critical Reynolds number

$_{\mathrm{u}}$ Upstream liquid density

μ_{u} Upstream fluid dynamic viscosity

$\dot{}_{\mathrm{A}}$, $\dot{}_{\mathrm{B}}$ Thermal fluxes into the local restriction at inlets A and B

The liquid velocities at inlets A and B follow from the mass flow rates at those inlets:

$$v_A = \frac{\dot{m}_A}{A \cdot \rho_{A,u}}$$

$$v_B = \frac{\dot{m}_B}{A \cdot \rho_{B,u}}$$

where $_{\mathrm{A,u}}$ and $_{\mathrm{B,u}}$ are the liquid mass densities at inlets A and B.

The Reynolds number in the restriction satisfies the expression

$$\mathrm{Re} = \frac{\left|\dot{m}_A\right|}{A_r \cdot \mu_u}$$

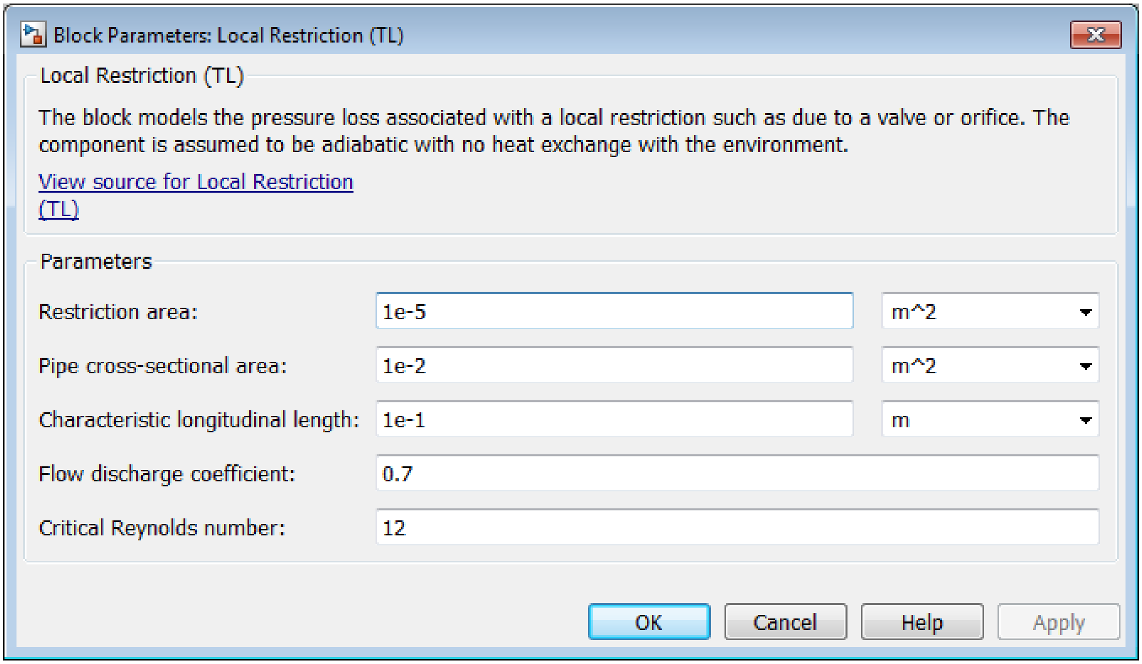
The block smooths the transition between laminar and turbulent flow regimes ($\mathrm{Re} \leq \mathrm{Re}_c$ and $\mathrm{Re} \geq \mathrm{Re}_c$, respectively). Smoothing occurs in a way that avoids zero-crossing events in both the flow regime transition and at zero flow.

In its default configuration, the block represents a sharp orifice with a cross-sectional area of 1e-5 m^2 and a discharge coefficient of 0.7.

Assumptions and Limitations

- Restriction is adiabatic. No heat exchange occurs between the restriction and its surroundings.
- Liquid dynamic compressibility and thermal inertia are negligibly small.

Dialog Box and Parameters



Block Parameters: Local Restriction (TL)

Local Restriction (TL)

The block models the pressure loss associated with a local restriction such as due to a valve or orifice. The component is assumed to be adiabatic with no heat exchange with the environment.

View source for Local Restriction (TL)

Parameters

Restriction area:	1e-5	m^2
Pipe cross-sectional area:	1e-2	m^2
Characteristic longitudinal length:	1e-1	m
Flow discharge coefficient:	0.7	
Critical Reynolds number:	12	

Restriction Area

Enter the cross-sectional area of the local restriction. This is the area the liquid in the restriction flows through. The default value is `1e-5` m^2 .

Pipe cross-sectional area

Enter the cross-sectional area of the adjoining pipes. This is the area the liquid in the pipes flows through. The default value is `1e-2` m^2 .

Characteristic longitudinal length

Enter the restriction length along the flow direction. The default value is `1e-1` m.

Flow discharge coefficient

Enter the discharge coefficient associated with the minor loss of the restriction. The default value is `0.7`, corresponding to a sharp orifice.

Critical Reynolds number

Enter the Reynolds number at which flow transitions from laminar to turbulent. The default value is `12`, corresponding to a sharp orifice.

Ports

The block has two thermal liquid conserving ports, A and B. These ports represent the inlet and outlet of the local restriction.

See Also

Variable Local Restriction (TL)

Pipe (TL)

Magnetic Reference

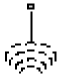
Purpose

Reference connection for magnetic ports

Library

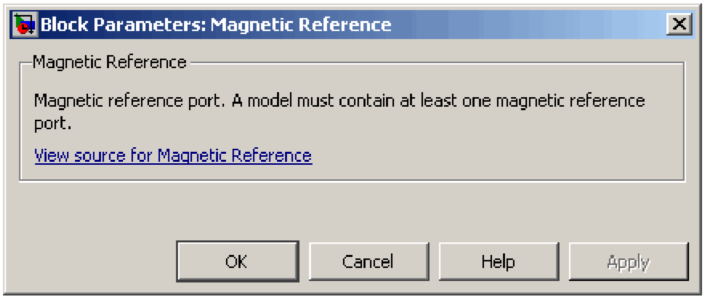
Magnetic Elements

Description



The Magnetic Reference block represents a reference point for all magnetic conserving ports. A model with magnetic elements must contain at least one Magnetic Reference block.

Dialog Box and Parameters



Block Parameters: Magnetic Reference

Magnetic Reference

Magnetic reference port. A model must contain at least one magnetic reference port.

View source for Magnetic Reference

The Magnetic Reference block has no parameters.

Ports

The block has one magnetic conserving port.

See Also

Electrical Reference

Hydraulic Reference

Mechanical Rotational Reference

Mechanical Translational Reference

Thermal Reference

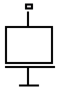
Purpose

Ideal mechanical translational mass

Library

Mechanical Translational Elements

Description



The Mass block represents an ideal mechanical translational mass, described with the following equation:

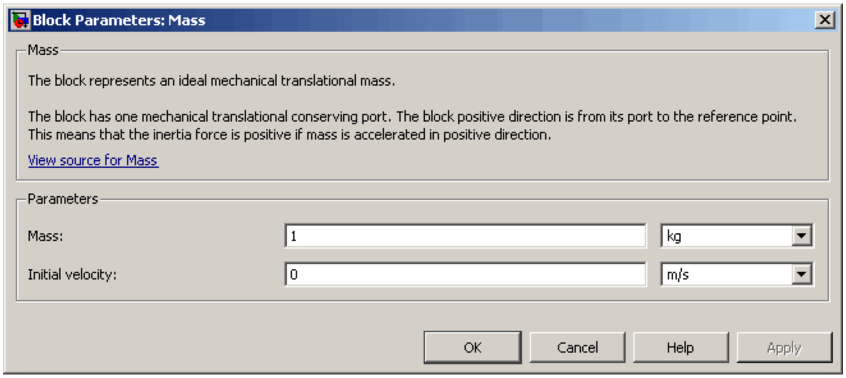
$$F = m\frac{dv}{dt}$$

where

F	Inertia force
m	Mass
v	Velocity
t	Time

The block has one mechanical translational conserving port. The block positive direction is from its port to the reference point. This means that the inertia force is positive if mass is accelerated in positive direction.

Dialog Box and Parameters



Block Parameters: Mass

Mass

The block represents an ideal mechanical translational mass.

The block has one mechanical translational conserving port. The block positive direction is from its port to the reference point. This means that the inertia force is positive if mass is accelerated in positive direction.

View source for Mass

Parameters

Mass: 1 kg

Initial velocity: 0 m/s

Mass

Mass. The default value is `1` kg.

Initial velocity

Initial velocity of the mass. This parameter specifies the initial condition for use in computing the block's initial state at the beginning of a simulation run. For more information, see "Initial Conditions Computation". The default value is `0`.

Ports

The block has one mechanical translational conserving port, associated with the mass connection to the system.

See Also

Inertia

Mass Flow Rate & Thermal Flux Sensor (TL)

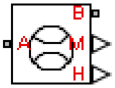
Purpose

Ideal sensor that measures mass flow rate and thermal flux

Library

Thermal Liquid/Sensors

Description



The Mass Flow Rate & Thermal Flux Sensor (TL) block represents an ideal sensor that measures mass and thermal energy flow rates through a thermal liquid node. Because the flow rates are Through variables, the block must connect in series with the component being measured.

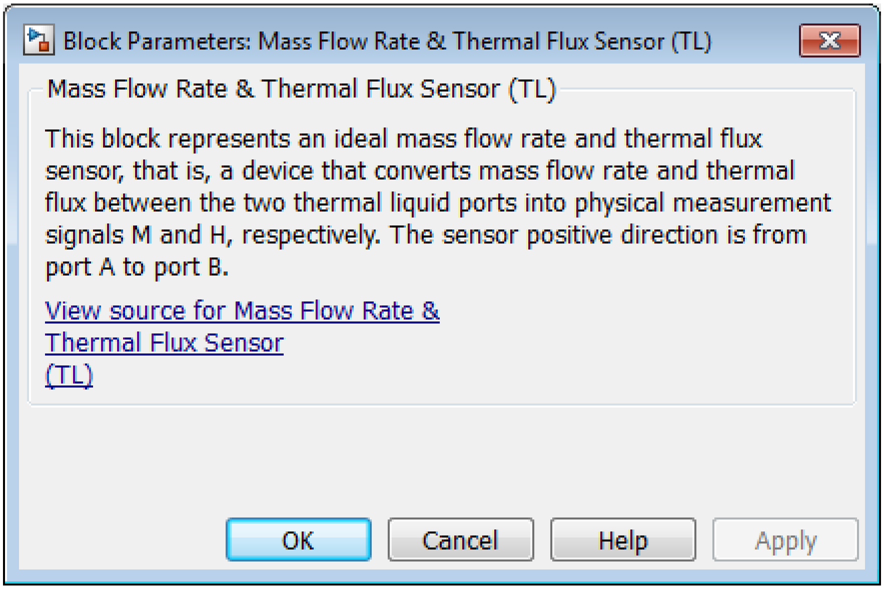
The relative orientation of ports A and B establishes the measurement sign. The sign is positive if flow occurs from port A to port B. Switching port connections reverses the measurement sign.

Two physical signal ports output the measurement data. Port M outputs the mass flow rate. Port H outputs the thermal energy flow rate. Connect the ports to PS-Simulink Converter blocks to transform the output physical signals into Simulink signals, e.g., for plotting or additional data processing.

Assumptions and Limitations

- Sensor inertia is negligible.

Dialog Box and Parameters



Block Parameters: Mass Flow Rate & Thermal Flux Sensor (TL)

Mass Flow Rate & Thermal Flux Sensor (TL)

This block represents an ideal mass flow rate and thermal flux sensor, that is, a device that converts mass flow rate and thermal flux between the two thermal liquid ports into physical measurement signals M and H, respectively. The sensor positive direction is from port A to port B.

View source for Mass Flow Rate & Thermal Flux Sensor (TL)

The block has no parameters.

Ports

The block has the following ports.

A, B	Thermal liquid conserving ports
M	Physical signal output port for mass flow rate measurement
H	Physical signal output port for thermal flux measurement

See Also

Pressure & Temperature Sensor (TL)

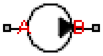
Purpose

Ideal compressor that generates a constant mass flow rate through its inlets

Library

Thermal Liquid/Sources

Description



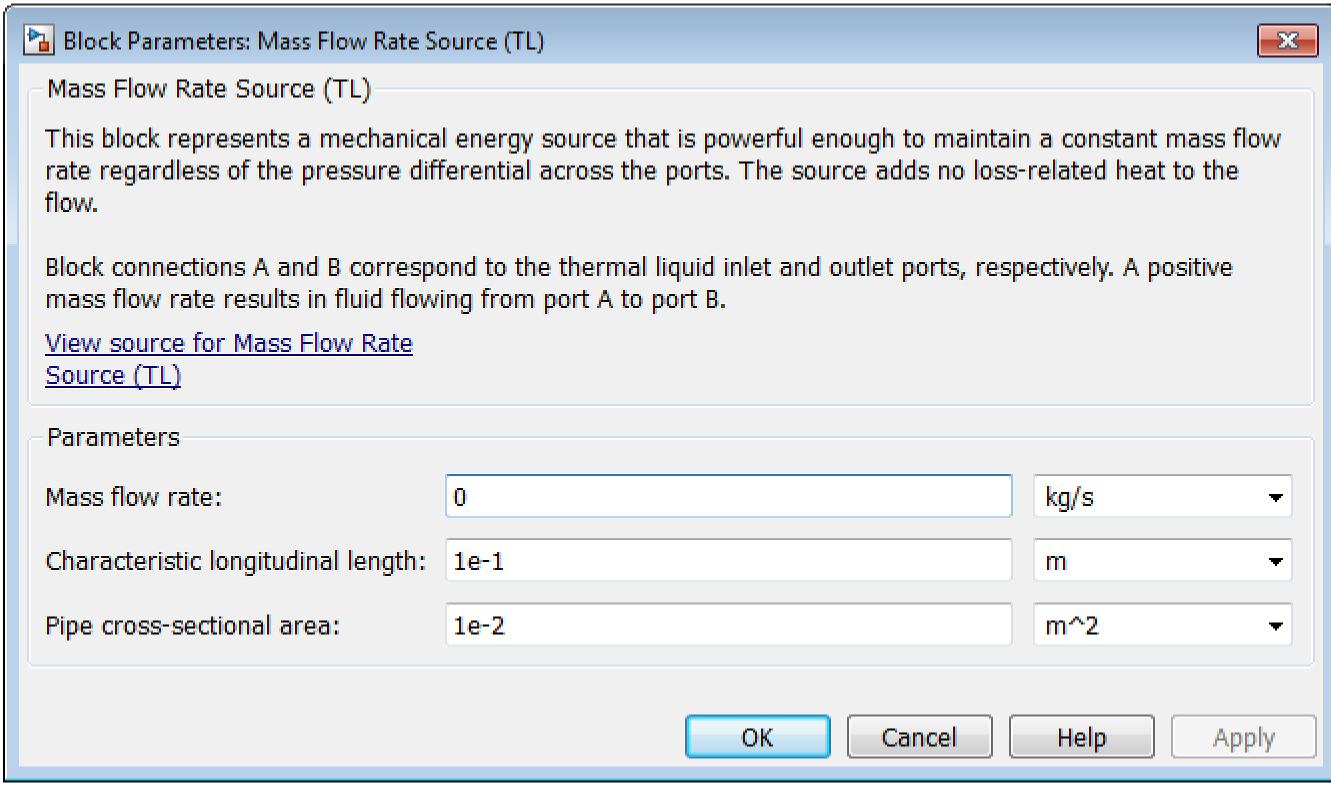
The Mass Flow Rate Source (TL) block represents an ideal compressor generating a constant mass flow rate through its inlets. The compressor can hold the specified mass flow rate regardless of the pressure differential between its inlets. No heat exchange occurs between the compressor and the flowing liquid.

Ports A and B represent the compressor inlets. Both ports are thermal liquid conserving ports. The mass flow rate is positive when liquid flows from inlet A to inlet B.

Assumptions and Limitations

- No heat exchange occurs between the compressor and the flowing liquid.
- The mass flow rate through the compressor inlets is independent of the pressure differential between those inlets.

Dialog Box and Parameters



Block Parameters: Mass Flow Rate Source (TL)

Mass Flow Rate Source (TL)

This block represents a mechanical energy source that is powerful enough to maintain a constant mass flow rate regardless of the pressure differential across the ports. The source adds no loss-related heat to the flow.

Block connections A and B correspond to the thermal liquid inlet and outlet ports, respectively. A positive mass flow rate results in fluid flowing from port A to port B.

View source for Mass Flow Rate Source (TL)

Parameters

Mass flow rate:	0	kg/s
Characteristic longitudinal length:	1e-1	m
Pipe cross-sectional area:	1e-2	m^2

Mass flow rate

Enter the rate at which mass flows through the compressor inlets. The default value is `0` kg/s.

Characteristic longitudinal length

Enter the mean path length the liquid must flow through to go from inlet A to inlet B. The default value is `1e-1` m.

Pipe cross-sectional area

Enter the cross-sectional area of the adjoining pipes. The default value is `1e-2` m^2.

Ports

This block has two thermal liquid conserving ports, A and B.

See Also

Controlled Mass Flow Rate Source (TL)

Pressure Source (TL)

Controlled Pressure Source (TL)

Mechanical Rotational Reference

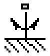
Purpose

Reference connection for mechanical rotational ports

Library

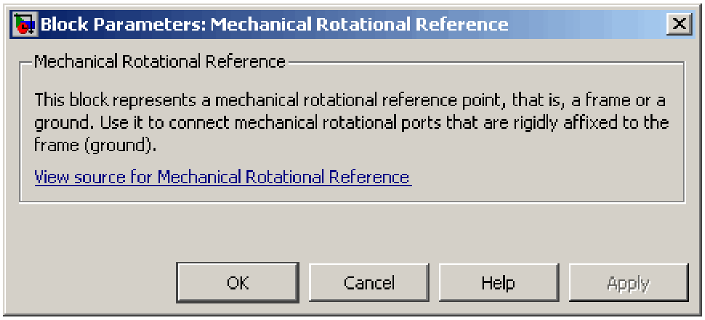
Mechanical Rotational Elements

Description



The Mechanical Rotational Reference block represents a reference point, or frame, for all mechanical rotational ports. All rotational ports that are rigidly clamped to the frame (ground) must be connected to a Mechanical Rotational Reference block.

Dialog Box and Parameters



Block Parameters: Mechanical Rotational Reference

Mechanical Rotational Reference

This block represents a mechanical rotational reference point, that is, a frame or a ground. Use it to connect mechanical rotational ports that are rigidly affixed to the frame (ground).

View source for Mechanical Rotational Reference

The Mechanical Rotational Reference block has no parameters.

Ports

The block has one mechanical rotational port.

See Also

Electrical Reference

Hydraulic Reference

Mechanical Translational Reference

Thermal Reference

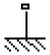
Purpose

Reference connection for mechanical translational ports

Library

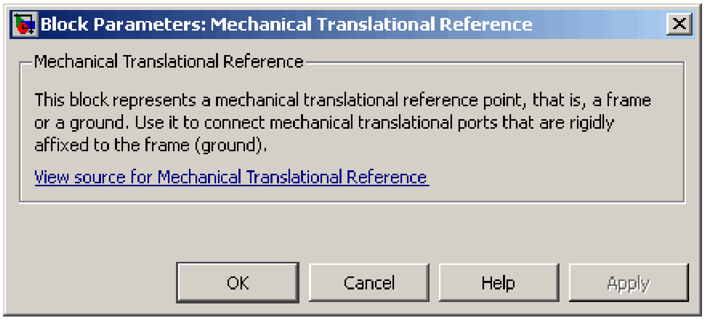
Mechanical Translational Elements

Description



The Mechanical Translational Reference block represents a reference point, or frame, for all mechanical translational ports. All translational ports that are rigidly clamped to the frame (ground) must be connected to a Mechanical Translational Reference block.

Dialog Box and Parameters



Block Parameters: Mechanical Translational Reference

Mechanical Translational Reference

This block represents a mechanical translational reference point, that is, a frame or a ground. Use it to connect mechanical translational ports that are rigidly affixed to the frame (ground).

View source for Mechanical Translational Reference

The Mechanical Translational Reference block has no parameters.

Ports

The block has one mechanical translational port.

See Also

Electrical Reference

Hydraulic Reference

Mechanical Rotational Reference

Thermal Reference

MMF Sensor

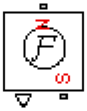
Purpose

Ideal magnetomotive force sensor

Library

Magnetic Sensors

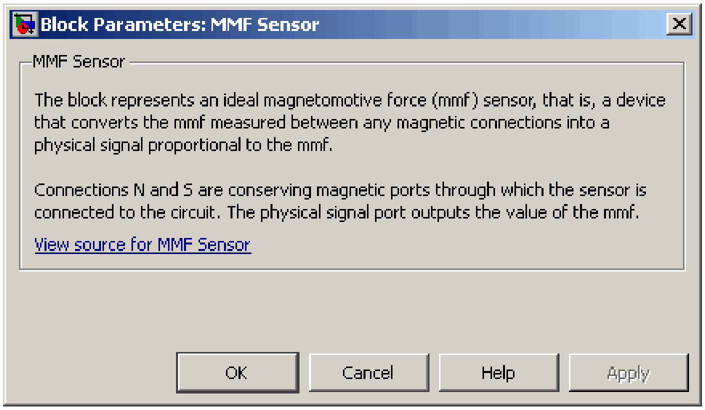
Description



The MMF Sensor block represents an ideal magnetomotive force (mmf) sensor, that is, a device that converts the mmf measured between any magnetic connections into a physical signal proportional to the mmf.

Connections N and S are conserving magnetic ports through which the sensor is connected to the circuit. The physical signal port outputs the value of the mmf.

Dialog Box and Parameters



The block represents an ideal magnetomotive force (mmf) sensor, that is, a device that converts the mmf measured between any magnetic connections into a physical signal proportional to the mmf.

Connections N and S are conserving magnetic ports through which the sensor is connected to the circuit. The physical signal port outputs the value of the mmf.

View source for MMF Sensor

The block has no parameters.

Ports

The block has the following ports:

N

Magnetic conserving port associated with the sensor North terminal.

S

Magnetic conserving port associated with the sensor South terminal.

The block also has a physical signal output port, which outputs the value of the mmf.

See Also

Controlled MMF Source

MMF Source

MMF Source

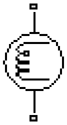
Purpose

Ideal magnetomotive force source

Library

Magnetic Sources

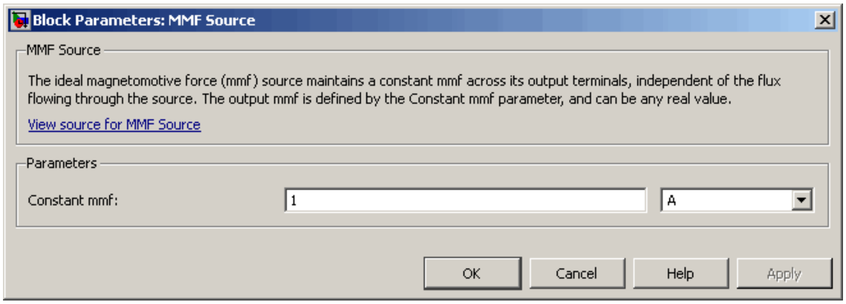
Description



The MMF Source block represents an ideal magnetomotive force (mmf) source that is powerful enough to maintain specified constant mmf across its output terminals, regardless of the flux flowing through the source.

You specify the output mmf by using the **Constant mmf** parameter, which can be positive, negative, or zero.

Dialog Box and Parameters



Block Parameters: MMF Source

MMF Source

The ideal magnetomotive force (mmf) source maintains a constant mmf across its output terminals, independent of the flux flowing through the source. The output mmf is defined by the Constant mmf parameter, and can be any real value.

View source for MMF Source

Parameters

Constant mmf: 1 A

Constant mmf

Output mmf. You can specify any real value. The default value is 1 A.

Ports

The block has two magnetic conserving ports associated with its terminals.

See Also

Controlled Flux Source

Controlled MMF Source

Flux Source

Mutual Inductor

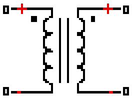
Purpose

Mutual inductor in electrical systems

Library

Electrical Elements

Description



The Mutual Inductor block models a mutual inductor, described with the following equations:

$$V1 = L1\frac{dI1}{dt} + M\frac{dI2}{dt}$$

$$V2 = L2\frac{dI2}{dt} + M\frac{dI1}{dt}$$

$$M = k\sqrt{L1 \cdot L2}$$

where

`V1`	Voltage across winding 1
`V2`	Voltage across winding 2
`I1`	Current flowing into the + terminal of winding 1
`I2`	Current flowing into the + terminal of winding 2
`L1`, `L2`	Winding self-inductances
`M`	Mutual inductance
`k`	Coefficient of coupling, 0 < `k` < 1
`t`	Time

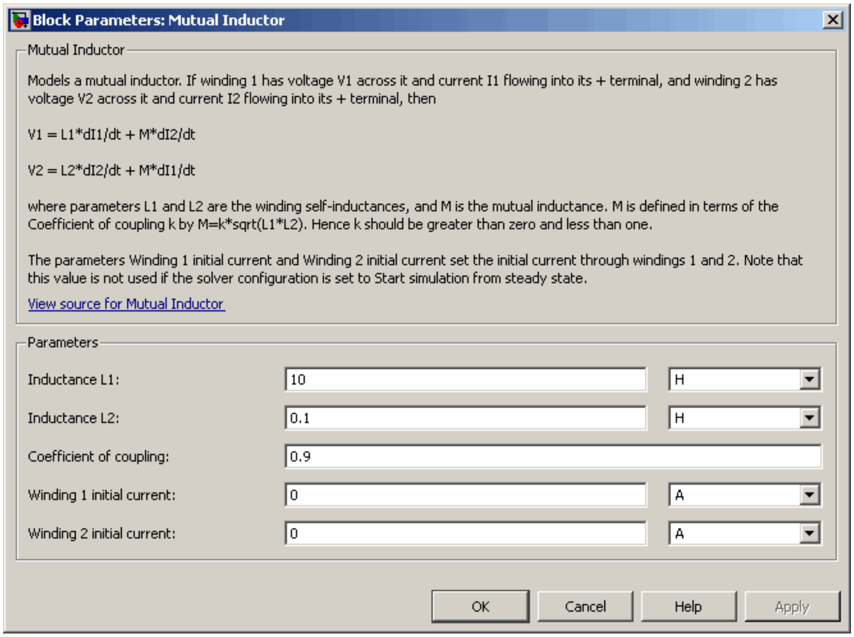
The **Winding 1 initial current** and **Winding 2 initial current** parameters set the initial current through windings 1 and 2.

Note These values are not used if the solver configuration is set to **Start simulation from steady state**.

This block can be used to represent an AC transformer. If inductance and mutual inductance terms are not important in a model, or are unknown, you can use the Ideal Transformer block instead.

The two electrical networks connected to the primary and secondary windings must each have their own Electrical Reference block.

Dialog Box and Parameters



Block Parameters: Mutual Inductor

Mutual Inductor

Models a mutual inductor. If winding 1 has voltage V1 across it and current I1 flowing into its + terminal, and winding 2 has voltage V2 across it and current I2 flowing into its + terminal, then

V1 = L1*dI1/dt + M*dI2/dt

V2 = L2*dI2/dt + M*dI1/dt

where parameters L1 and L2 are the winding self-inductances, and M is the mutual inductance. M is defined in terms of the Coefficient of coupling k by M=k*sqrt(L1*L2). Hence k should be greater than zero and less than one.

The parameters Winding 1 initial current and Winding 2 initial current set the initial current through windings 1 and 2. Note that this value is not used if the solver configuration is set to Start simulation from steady state.

View source for Mutual Inductor

Parameters

Inductance L1:	10	H
Inductance L2:	0.1	H
Coefficient of coupling:	0.9	
Winding 1 initial current:	0	A
Winding 2 initial current:	0	A

Inductance L1

Self-inductance of the first winding. The default value is `10` H.

Inductance L2

Self-inductance of the second winding. The default value is `0.1` H.

Coefficient of coupling

Coefficient of coupling, which defines the mutual inductance. The parameter value should be greater than zero and less than 1. The default value is `0.9`.

Winding 1 initial current

Initial current through the first winding. This parameter is not used if the solver configuration is set to **Start simulation from steady state**. The default value is `0`.

Winding 2 initial current

Initial current through the second winding. This parameter is not used if the solver configuration is set to **Start simulation from steady state**. The default value is `0`.

Ports

The block has four electrical conserving ports. Polarity is indicated by the + and – signs.

See Also

Ideal Transformer

Op-Amp

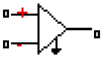
Purpose

Ideal operational amplifier

Library

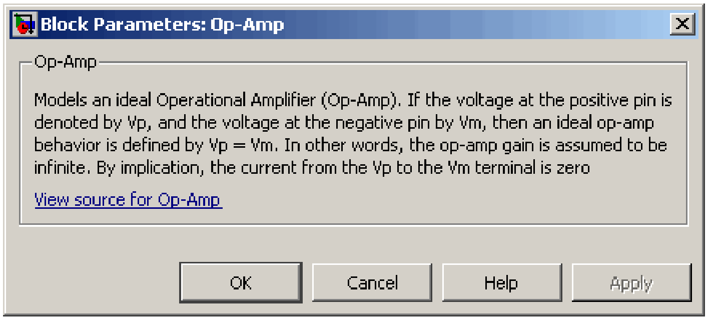
Electrical Elements

Description



The Op-Amp block models an ideal operational amplifier (op-amp). If the voltage at the positive pin is denoted by *Vp*, and the voltage at the negative pin by *Vm*, then an ideal op-amp behavior is defined by *Vp* = *Vm*. In other words, the op-amp gain is assumed to be infinite. By implication, the current from the Vp to the Vm terminal is zero.

Dialog Box and Parameters



Models an ideal Operational Amplifier (Op-Amp). If the voltage at the positive pin is denoted by Vp, and the voltage at the negative pin by Vm, then an ideal op-amp behavior is defined by Vp = Vm. In other words, the op-amp gain is assumed to be infinite. By implication, the current from the Vp to the Vm terminal is zero

The Op-Amp block has no parameters.

Ports

The block has three electrical conserving ports.

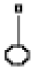
Purpose

Electrical port terminator that draws no current

Library

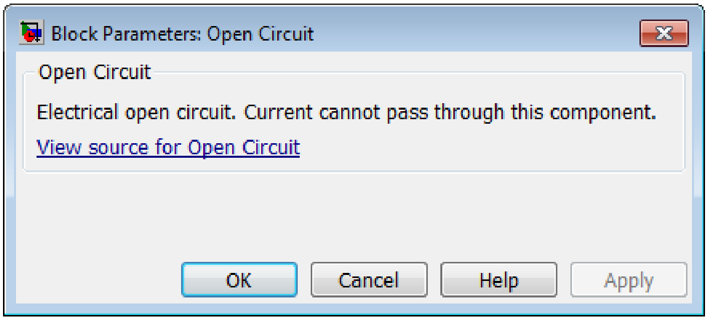
Electrical Elements

Description



The Open Circuit block represents an electrical terminal that draws no current. Physical Network block diagrams do not allow unconnected Conserving ports. Use this block to terminate electrical ports (on other blocks) that you wish to leave open-circuit.

Dialog Box and Parameters



Block Parameters: Open Circuit

Open Circuit

Electrical open circuit. Current cannot pass through this component.

View source for Open Circuit

The Open Circuit block has no parameters.

Ports

The block has one electrical conserving port.

See Also

Hydraulic Cap

Rotational Free End

Translational Free End

Perfect Insulator

Purpose

Thermal element with no thermal mass and perfect insulation

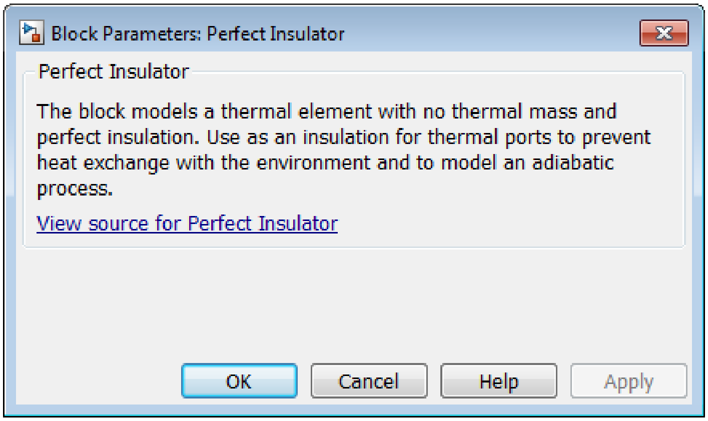
Library

Thermal Elements

Description

The Perfect Insulator block models a thermal element with no thermal mass and perfect insulation. Physical Network block diagrams do not allow unconnected Conserving ports. Use this block as an insulation for thermal ports to prevent heat exchange with the environment and to model an adiabatic process.

Dialog Box and Parameters



Block Parameters: Perfect Insulator

Perfect Insulator

The block models a thermal element with no thermal mass and perfect insulation. Use as an insulation for thermal ports to prevent heat exchange with the environment and to model an adiabatic process.

View source for Perfect Insulator

The Perfect Insulator block has no parameters.

Ports

The block has one thermal conserving port.

See Also

Hydraulic Cap

Open Circuit

Rotational Free End

Translational Free End

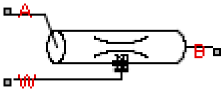
Purpose

Rigid conduit for liquid transport

Library

Thermal Liquid/Elements

Description



The Pipe (TL) block represents a pipeline segment with a fixed volume of liquid. The liquid experiences pressure losses and heating due to, respectively, viscous friction and conductive heat transfer with the pipe wall. Viscous friction follows from the Darcy-Weisbach law, while the heat exchange coefficient follows from Nusselt number correlations. Heat transfer can occur in an unsteady manner.

The block includes parameters to account for the dynamic compressibility and inertia of liquid in a pipe. However, by default the block treats liquid flow through the pipe as steady and liquid mass within the pipe as constant. In this mode, the momentum and mass equations of this block are in their steady states. The liquid behaves as if it were incompressible. Pressure waves due to liquid inertia are absent in the pipe.

Depending on the effects you include, the block can function in three configurations: resistive tube, compressible resistive tube, and pipeline segment. The table summarizes the effects present in each configuration.

Configuration	Dynamic Compressibility	Flow Inertia	Thermal Dynamics
Resistive tube	Off	Off	On
Compressible resistive tube	On	Off	On
Pipeline segment	On	On	On

The configuration to use depends on the relevant effects the model must capture. The pipeline segment configuration provides the greatest accuracy. However, this configuration also increases model complexity, raising the simulation computational cost and challenging the convergence to a numerical solution in rapid transient processes. As the simplest in the list, the resistive tube configuration provides a good starting point in a model. This is the default configuration of the block.

To view the block source code, at the MATLAB command line enter:

```
edit <matlabroot>\toolbox\physmod\simscape\library\m\...
+foundation\+thermal_liquid\+elements\CONFIGURATION.ssc
```

Replace `<matlabroot>` with the output of the `matlabroot` command and `CONFIGURATION` with one of the three configuration names:

- resistive_tube
- compressible_resistive_tube
- pipeline_segment

Use this block in the resistive tube configuration when:

- Thermal dynamic effects are important but flow dynamic effects, which have a smaller time scale, are not.
- Liquid mass in the pipe is a negligible fraction of the total liquid mass in the system.

The resistive tube configuration is the recommended starting point for this block, even if fluid dynamic compressibility and flow inertia are important. The simulation results using this configuration provides reasonable initial conditions for more advanced configurations in which fluid dynamic compressibility and flow inertia are important—e.g. compressible resistive tube and pipeline segment configurations.

A typical application of the resistive tube configuration is the thermal analysis of an open-loop circuit linked to reservoirs or of a closed-loop circuit in which the majority of the liquid mass exists outside the pipe.

Use this block in the compressible resistive tube configuration when:

- Thermal dynamic effects are important but flow dynamic effects, which have a smaller time scale, are not.
- Liquid mass in the pipe is not negligible with respect to the total liquid mass in the system

A typical application of the compressible resistive tube configuration include the thermal analysis of a closed-loop circuit, not linked to any reservoir, in which thermal expansion raises liquid pressure—e.g. a car cooling system.

Use this block in the pipeline segment configuration when the characteristic time of the thermal liquid system is close to the liquid compressibility time scale:

$$\tau = \frac{L}{a}$$

where L is the characteristic longitudinal length of the pipe in which waves can develop and a is the speed of sound in the liquid. A typical application of the pipeline segment configuration is the study of the water hammer effect due to fast-shutting valves.

The following equations govern the behavior of liquid in the pipe:

$$\dot{m}_A + \dot{m}_B = \begin{cases} 0, & \text{if fluid dynamic compressibility is ’off’} \\ V \cdot \rho_{\text{int}} \cdot \left(\dfrac{1}{\beta_{\text{int}}} \cdot \dfrac{dp_{\text{int}}}{dt} + \alpha_{\text{int}} \cdot \dfrac{dT_{\text{int}}}{dt} \right), & \text{if fluid dynamic compressibility is ’on’} \end{cases}$$

$$A \cdot (p_{\text{int}} - p_A) + F_{vd,A} = \begin{cases} 0, & \text{if flow inertia is ’off’} \\ -\dfrac{L}{2} \cdot \ddot{m}_A, & \text{if flow inertia is ’on’} \end{cases}$$

$$A \cdot (p_{\text{int}} - p_B) + F_{vd,B} = \begin{cases} 0, & \text{if flow inertia is ’off’} \\ -\dfrac{L}{2} \cdot \ddot{m}_B, & \text{if flow inertia is ’on’} \end{cases}$$

$$V \cdot \frac{d(\rho_{\text{int}} \cdot u_{\text{int}})}{dt} = \phi_A + \phi_B + \phi_W + v_A \cdot F_{vd,A} + v_B \cdot F_{vd,B} + p_{\text{int}} \cdot A \cdot (v_A + v_B)$$

where

A	Cross-sectional area of the pipe
V	Volume of liquid in the pipe
L	Length of the pipe
p_{int}, p_{A}, p_{B}	Pressures of liquid in the pipe, at port A, and at port B
T_{int}, T_{A}, T_{B}	Temperatures of liquid in the pipe, at port A, and at port B
$\dot{m}_A$, $\dot{m}_B$	Mass flow rates of liquid into the pipe at port A and at port B
v_{A}, v_{B}	Velocities of liquid into the pipe at ports A and B
$F_{\text{vd,A}}$, $F_{\text{vd,B}}$	Viscous dissipations between the pipe volume center and ports A and B
$_{\text{int}}$	Isothermal bulk modulus of liquid in the pipe

a_{int}	Isobaric coefficient of thermal expansion of liquid in the pipe
c_{int}	Specific heat of liquid in the pipe
u_{int}	Internal energy of liquid in the pipe
$_{\text{int}}$	Density of liquid in the pipe
$_{\text{A}}$, $_{\text{B}}$, $_{\text{W}}$	Thermal fluxes into the pipe at ports A, B, and W

In the resistive tube configuration, the two momentum equations (second and third in the preceding system of equations) combine into a single pressure difference equation relating the liquid pressures at ports A and B. The liquid pressure inside the pipe equals the average of these two liquid pressures. Variations in liquid density are small, making momentum fluxes insignificant with respect to viscous forces.

Liquid velocities at ports A and B follow from the mass flow rates into the pipe through the same ports:

$$v_A = \frac{\dot{m}_A}{A \cdot \rho_{A,u}}$$

$$v_B = \frac{\dot{m}_B}{A \cdot \rho_{B,u}}$$

where $_{\text{A,u}}$ and $_{\text{B,u}}$ are the upwind mass densities of liquid at ports A and B.

Viscous forces at pipe inlets A and B depend on the flow regime (laminar or turbulent):

$$F_{vd,A} = \begin{cases} K_S \cdot \nu_{A,u} \cdot \left(\dfrac{L + L_{eq}}{2}\right) \cdot \dfrac{\dot{m}_A}{2 \cdot {D_H}^2}, & \mathit{if}\ \mathrm{Re}_A < \mathrm{Re}_l \\ -f_A \cdot \left(\dfrac{L + L_{eq}}{2}\right) \cdot \dfrac{\dot{m}_A \cdot |\dot{m}_A|}{2 \cdot \rho_{A,u} \cdot D_H \cdot A^2}, & \mathit{if}\ \mathrm{Re}_A \geq \mathrm{Re}_t \end{cases}$$

$$F_{vd,B} = \begin{cases} K_S \cdot \nu_{B,u} \cdot \left(\dfrac{L + L_{eq}}{2} \right) \cdot \dfrac{\dot{m}_B}{2 \cdot {D_H}^2}, & if \ \mathrm{Re}_B < \mathrm{Re}_l \\ -f_B \cdot \left(\dfrac{L + L_{eq}}{2} \right) \cdot \dfrac{\dot{m}_B \cdot \left| \dot{m}_B \right|}{2 \cdot \rho_{B,u} \cdot D_H \cdot A^2}, & if \ \mathrm{Re}_B \geq \mathrm{Re}_t \end{cases}$$

where

D_{H}	Hydraulic diameter of the pipe
f_{A}, f_{B}	Darcy friction factors in the two pipe halves adjacent to pipe inlets A and B
K_{S}	Shape factor of the pipe
L_{eq}	Aggregate equivalent length of local pipe resistances
Re_{A}, Re_{B}	Reynolds numbers at pipe inlets A and B
Re_{l}	Maximum Reynolds number corresponding to laminar flow in the pipe
Re_{B}	Minimum Reynolds number corresponding to turbulent flow in the pipe
$_{\mathrm{A,u}}$, $_{\mathrm{B,u}}$	Upwind liquid dynamic viscosities at pipe inlets A and B

The block smooths the transition between laminar and turbulent flow regimes ($\mathrm{Re}_{\mathrm{t}} > \mathrm{Re} > \mathrm{Re}_{\mathrm{l}}$) based on the Reynolds number. At pipe inlets A and B, the Reynolds numbers are

$$\mathrm{Re}_A = \frac{D_H \cdot \dot{m}_A}{\mu_{A,u}}$$

$$\mathrm{Re}_A = \frac{D_H \quad \dot{m}_A}{\mu_{A,u}}$$

where $\mu_{A,u}$ and $\mu_{B,u}$ are the upwind liquid dynamic viscosities at pipe inlets A and B. The Darcy friction factor, D_H, satisfies the Haaland approximation in the turbulent flow regime:

$$f = \frac{1}{\left[-1.8\log_{10}\left(\frac{6.9}{\mathrm{Re}} + \left(\frac{r/D_H}{3.7}\right)^{1.11}\right)\right]^2}$$

where r is the roughness of the internal pipe surface.

The following equation governs the convective heat transfer between the pipe wall and the liquid it encloses:

$$\phi_W = h \cdot \left(T_W - T_{MTD}\right) \cdot P \cdot L$$

where

D_H	Hydraulic diameter of the pipe
T_W	Temperature of the pipe wall (port W)
T_{MTD}	Mean liquid temperature difference between pipe inlets A and B
P	Cross-sectional perimeter of the pipe

The cross-sectional perimeter of the pipe, P, follows from the hydraulic diameter:

$$P = \frac{4 \cdot A}{D_H}$$

The heat transfer coefficient follows from empirical correlations involving the Nusselt number. These correlations relate the Nusselt number to powers of Reynolds and Prandtl numbers, the ratio of the pipe hydraulic diameter to its height, and the ratio of the liquid dynamic viscosities at the inlet and wall temperatures. The correlation used depends on the flow regime in the pipe—laminar or turbulent.

The block smooths the transition between flow regimes based on the Reynolds number:

$$Nu(\mathrm{Re}) = \begin{cases} a_l \cdot \mathrm{Re}^{b_l} \cdot \mathrm{Pr}^{c_l} \cdot \left(\dfrac{D_H}{L}\right)^{d_l} \cdot \left(\dfrac{\mu_{Int}}{\mu_W}\right)^{e_l}, & \mathrm{Re} \le \mathrm{Re}_l \\ Nu_l + \left(Nu_l - Nu_t\right) \cdot \dfrac{\mathrm{Re} - \mathrm{Re}_l}{\mathrm{Re}_t - \mathrm{Re}_l}, & \mathrm{Re}_l < \mathrm{Re} < \mathrm{Re}_t \\ a_t \cdot \mathrm{Re}^{b_t} \cdot \mathrm{Pr}^{c_t} \cdot \left(\dfrac{D_H}{L}\right)^{d_t} \cdot \left(\dfrac{\mu_{Int}}{\mu_W}\right)^{e_t}, & \mathrm{Re} \ge \mathrm{Re}_t \end{cases}$$

$$Nu_l = Nu(\mathrm{Re}_l)$$

$$Nu_t = Nu(\mathrm{Re}_t)$$

$$Nu = \frac{h\ D_H}{k_{Int}}$$

$$\mathrm{Pr} = \frac{c_{Int} \cdot \mu_{Int}}{k_{Int}}$$

where

Nu	Nusselt number
Nu_l	Nusselt number at the maximum Reynolds number corresponding to laminar flow
Nu_t	Nusselt number at the minimum Reynolds number corresponding to turbulent flow
Pr	Prandtl number
Re	Reynolds number

a_l, b_l, c_l, d_l, e_l Empirical correlation coefficients for Nusselt number computation in the laminar flow regime

a_t, b_t, c_t, d_t, e_t Empirical correlation coefficients for Nusselt number computation in the turbulent flow regime

k_{Int} Thermal conductivity of liquid in the pipe

μ_{Int} Dynamic viscosity of liquid in the pipe

In the laminar regime, the default coefficient values follow from the Sieder and Tate correlation:

$$Nu = 1.86 \cdot \left(\mathrm{Re} \cdot \mathrm{Pr} \cdot \frac{D_H}{L} \right)^{1/3} \cdot \left(\frac{\mu_{Int}}{\mu_W} \right)^{0.14}$$

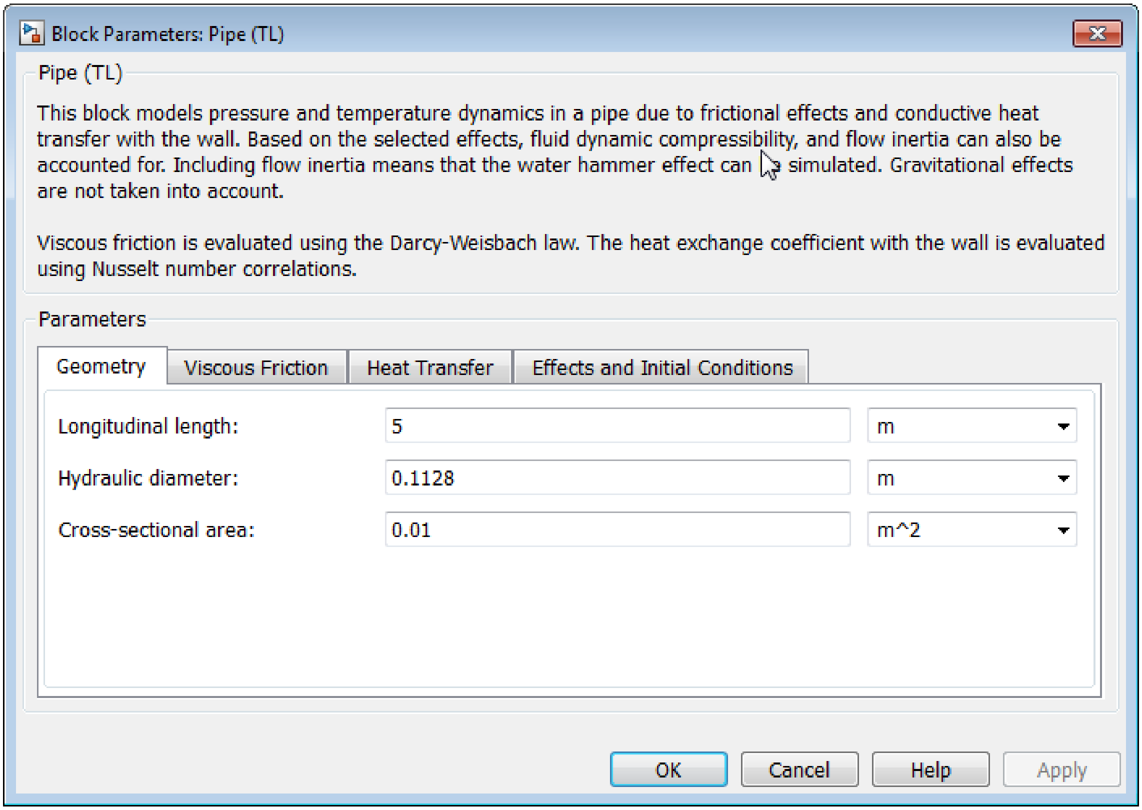
In the turbulent flow regime, the default coefficient values follow from the Colburn correlation:

$$Nu = 0.023 \cdot \mathrm{Re}^{0.8} \cdot \mathrm{Pr}^{0.33}$$

Assumptions and Limitations

- Pipe walls are not compliant.
- Flow through the pipe is fully developed.
- Gravitational effects on liquid pressure are negligible.

Dialog Box and Parameters



Block Parameters: Pipe (TL)

Pipe (TL)

This block models pressure and temperature dynamics in a pipe due to frictional effects and conductive heat transfer with the wall. Based on the selected effects, fluid dynamic compressibility, and flow inertia can also be accounted for. Including flow inertia means that the water hammer effect can be simulated. Gravitational effects are not taken into account.

Viscous friction is evaluated using the Darcy-Weisbach law. The heat exchange coefficient with the wall is evaluated using Nusselt number correlations.

Parameters

Longitudinal length:	5	m
Hydraulic diameter:	0.1128	m
Cross-sectional area:	0.01	m^2

Geometry

Longitudinal length

Enter the longitudinal length of the pipe. This is the length of the pipe along the direction of flow. The default value is 5 m.

Hydraulic diameter

Enter the hydraulic diameter of the pipe. This is the diameter of a cylindrical pipe with the same cross-sectional area. The default value is `0.1128` m.

Cross-sectional area

Enter the cross-sectional area of the pipe. This is the area of the pipe normal to the direction of flow. The default value is `0.01` m^2.

Viscous Friction

Aggregate equivalent length of local resistances

Enter the combined length of all local resistances present in the pipe. Local resistances include bends, fittings, armatures, and pipe inlets and outlets. The effect of the local resistances is to increase the effective length of the pipe segment. The default value is `1` m.

Shape factor

Enter the shape factor of the pipe. This dimensionless factor encodes the ratio between the height and width of the pipe, correcting for noncircular cross-sectional shapes. The block uses this factor to determine pressure losses in the laminar flow regime. The default value is `64`, corresponding to a pipe with circular cross-section.

Internal surface absolute roughness

Enter the absolute roughness of the internal surface of the pipe. This roughness equals the average height of surface defects inside the pipe. The block uses the absolute roughness to determine pressure losses in the turbulent flow regime. The default value is `1.5e-5` m, corresponding to drawn tubing.

Laminar flow upper margin

Enter the Reynolds number above which flow begins to transition from laminar to turbulent. This number equals the maximum Reynolds number corresponding to fully developed laminar flow. The default value is `2000`.

Turbulent flow lower margin

Enter the Reynolds number below which flow begins to transition from turbulent to laminar. This number equals to the minimum Reynolds number corresponding to fully developed turbulent flow. The default value is `4000`.

Heat Transfer

Laminar regime Nusselt number correlation coefficients

Enter a vector with the empirical correlation coefficients for convective heat transfer in the laminar flow regime. The coefficients must appear in the order `[a`$_l$ `b`$_l$ `c`$_l$ `d`$_l$ `e`$_l$`]`, corresponding to the empirical correlation

$$Nu(\mathrm{Re}) = a_l \ \mathrm{Re}^{b_l} \ \mathrm{Pr}^{c_l} \left(\frac{D_H}{L}\right)^{d_l} \left(\frac{\mu_{Int}}{\mu_W}\right)^{e_l}$$

The block uses the empirical correlation to determine heat transfer between the liquid and the pipe surface in the laminar regime. The default vector is `[1.86 1/3 1/3 1/3 0.14]`, from the Sieder and Tate correlation in the laminar regime.

Turbulent regime Nusselt number correlation coefficients

Enter a vector with the empirical correlation coefficients for convective heat transfer in the turbulent flow regime. The coefficients must appear in the order `[a`$_t$ `b`$_t$ `c`$_t$ `d`$_t$ `e`$_t$`]`, corresponding to the empirical correlation

$$Nu(\mathrm{Re}) = a_t \cdot \mathrm{Re}^{b_t} \cdot \mathrm{Pr}^{c_t} \cdot \left(\frac{D_H}{L}\right)^{d_t} \cdot \left(\frac{\mu_{Int}}{\mu_W}\right)^{e_t}$$

The block uses the empirical correlation to determine heat transfer between the liquid and the pipe surface in the turbulent regime. The default vector is `[0.023 0.8 1/3 0 0]`, from the Colburn correlation in the turbulent regime.

Effects and Initial Conditions

Fluid dynamic compressibility

Select whether to account for the dynamic compressibility of the liquid. Dynamic compressibility gives the liquid density a dependence on pressure and temperature, impacting the transient response of the system at small time scales. Selecting `On` displays the additional parameter **Initial fluid pressure in the pipe**. The default setting is `Off`.

Flow inertia

Select whether to account for the flow inertia of the liquid. Flow inertia gives the liquid a resistance to changes in mass flow rate. Selecting `On` displays the additional parameter **Initial mass flow rate oriented from A to B**. The default setting is `Off`.

Initial fluid temperature inside the pipe

Enter the liquid temperature in the pipe at time zero. The default value is `293.15` K.

Initial fluid pressure inside the pipe

Enter the liquid pressure in the pipe at time zero. This parameter appears only when **Fluid dynamic compressibility** is `On`. The default value is `1` atm.

Initial mass flow rate oriented from A to B

Enter the mass flow rate from port A to port B at time zero. This parameter is visible only when **Flow inertia** is `On`. The default value is `0.1` kg/s.

Ports

The block has two thermal liquid conserving ports, A and B, and one thermal conserving port, W.

See Also

Local Restriction (TL)

Variable Local Restriction (TL)

Pneumatic Absolute Reference

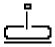
Purpose

Reference connection to zero absolute pressure and temperature for pneumatic ports

Library

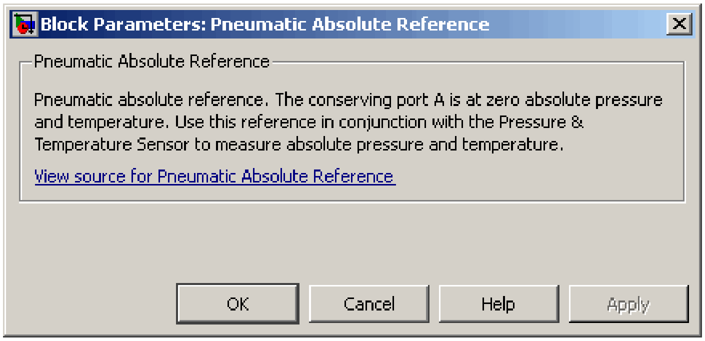
Pneumatic Elements

Description



The Pneumatic Absolute Reference block provides a pneumatic reference port at zero absolute pressure and temperature. Use this block with the Pneumatic Pressure & Temperature Sensor block to create Physical Signals corresponding to absolute pressure and temperature.

Dialog Box and Parameters



Block Parameters: Pneumatic Absolute Reference

Pneumatic Absolute Reference

Pneumatic absolute reference. The conserving port A is at zero absolute pressure and temperature. Use this reference in conjunction with the Pressure & Temperature Sensor to measure absolute pressure and temperature.

View source for Pneumatic Absolute Reference

The block has no parameters.

Ports

The block has one pneumatic conserving port, which is at zero absolute pressure and temperature.

See Also

Pneumatic Atmospheric Reference

Pneumatic Pressure & Temperature Sensor

Pneumatic Atmospheric Reference

Purpose

Reference connection to ambient pressure and temperature for pneumatic ports

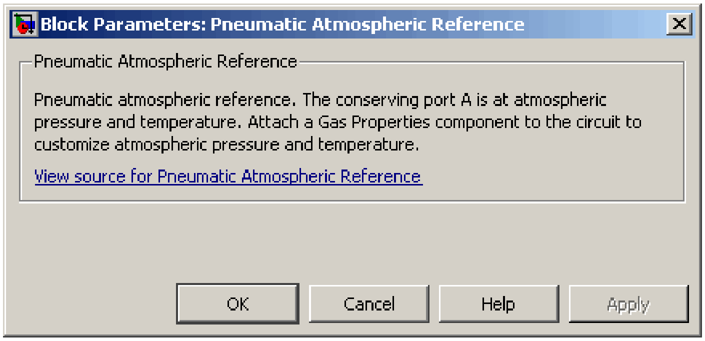
Library

Pneumatic Elements

Description

The Pneumatic Atmospheric Reference block provides a pneumatic reference port with pressure and temperature values set to the ambient temperature and pressure. The Gas Properties block, if present, specifies the values for ambient temperature and pressure for all pneumatic blocks in the circuit. If a pneumatic circuit does not contain a Gas Properties block, ambient temperature and pressure are set to default values of 293.15 K and 101,325 Pa. Use the Pneumatic Atmospheric Reference block with the Pneumatic Pressure Source block to model an ideal pressure source that takes atmospheric air, and increases the pressure by a constant amount.

Dialog Box and Parameters


Block Parameters: Pneumatic Atmospheric Reference
Pneumatic Atmospheric Reference
OK
Cancel
Help
Apply

Pneumatic atmospheric reference. The conserving port A is at atmospheric pressure and temperature. Attach a Gas Properties component to the circuit to customize atmospheric pressure and temperature.

View source for Pneumatic Atmospheric Reference

The block has no parameters.

Ports

The block has one pneumatic conserving port.

See Also

Gas Properties

Pneumatic Absolute Reference

Pneumatic Pressure Source

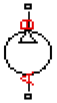
Purpose

Ideal compressor with constant mass flow rate

Library

Pneumatic Sources

Description



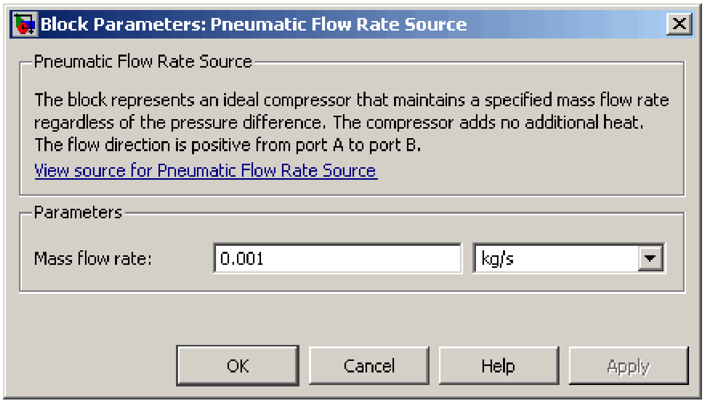
The Pneumatic Flow Rate Source block represents an ideal compressor that maintains a specified mass flow rate regardless of the pressure difference. Use this block when delivery of an actual device is practically independent of the source pressure, for example, in positive displacement compressors. The compressor adds no heat. Block connections A and B correspond to the pneumatic inlet and outlet ports, respectively.

The block positive direction is from port A to port B. This means that the flow rate is positive if it flows from A to B. The pressure differential is determined as $p = p_A - p_B$ and is negative if pressure at the source outlet is greater than pressure at its inlet. The power generated by the source is negative if the source adds energy to the flow.

Warning

Be careful when driving an orifice directly from a flow rate source. The choked flow condition limits the flow that is possible through an orifice as a function of upstream pressure and temperature. Hence the flow rate value produced by the flow rate source must be compatible with upstream pressure and temperature. Specifying a flow rate that is too high will result in an unsolvable set of equations.

Dialog Box and Parameters



Block Parameters: Pneumatic Flow Rate Source

Pneumatic Flow Rate Source

The block represents an ideal compressor that maintains a specified mass flow rate regardless of the pressure difference. The compressor adds no additional heat. The flow direction is positive from port A to port B.

View source for Pneumatic Flow Rate Source

Parameters

Mass flow rate: 0.001 kg/s

Mass flow rate

Specify the mass flow rate of the source. The default value is `0.001` kg/s.

Ports

The block has the following ports:

A

Pneumatic conserving port associated with the source inlet.

B

Pneumatic conserving port associated with the source outlet.

See Also

Controlled Pneumatic Flow Rate Source

Pneumatic Mass & Heat Flow Sensor

Pneumatic Mass & Heat Flow Sensor

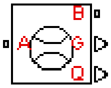
Purpose

Ideal mass flow and heat flow sensor

Library

Pneumatic Sensors

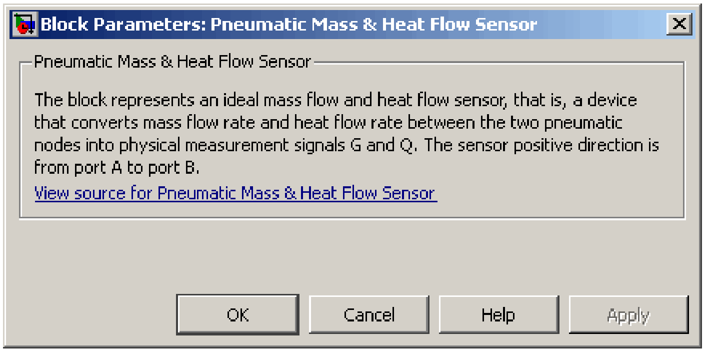
Description



The Pneumatic Mass & Heat Flow Sensor block represents an ideal mass flow and heat flow sensor, that is, a device that converts mass flow rate and heat flow rate between the two pneumatic nodes into physical measurement signals G and Q, respectively.

The sensor positive direction is from port A to port B.

Dialog Box and Parameters


Block Parameters: Pneumatic Mass & Heat Flow Sensor
Pneumatic Mass & Heat Flow Sensor
OK
Cancel
Help
Apply

The block represents an ideal mass flow and heat flow sensor, that is, a device that converts mass flow rate and heat flow rate between the two pneumatic nodes into physical measurement signals G and Q. The sensor positive direction is from port A to port B.

View source for Pneumatic Mass & Heat Flow Sensor

The block has no parameters.

Ports

The block has the following ports:

A

Pneumatic conserving port associated with the sensor inlet.

B

Pneumatic conserving port associated with the sensor outlet.

See Also

Controlled Pneumatic Flow Rate Source

Pneumatic Flow Rate Source

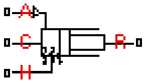
Purpose

Translational pneumatic piston chamber based on ideal gas law

Library

Pneumatic Elements

Description



The Pneumatic Piston Chamber block models a pneumatic piston chamber based on the ideal gas law and assuming constant specific heats. Use this model as a building block for pneumatic translational actuators. The piston can exert force in one direction only, and the direction is set by the **Chamber orientation** parameter.

The continuity equation for the network representation of the piston chamber is

$$G = \frac{V_0 + A \cdot x}{RT}\left(\frac{dp}{dt} - \frac{p}{T}\frac{dT}{dt}\right) + \frac{A}{RT} \cdot p \cdot \frac{dx}{dt}$$

where

G	Mass flow rate at input port
V_0	Initial chamber volume
A	Piston effective area
x	Piston displacement
p	Absolute pressure in the chamber
R	Specific gas constant
T	Absolute gas temperature
t	Time

The energy equation is

$$q = \frac{c_v}{R}\left(V_0 + A \cdot x\right)\frac{dp}{dt} + \frac{c_p \cdot A}{R}\, p \frac{dx}{dt} - q_w$$

where

q Heat flow due to gas inflow in the chamber (through the pneumatic port)

q_w Heat flow through the chamber walls (through the thermal port)

c_v Specific heat at constant volume

c_p Specific heat at constant pressure

The force equation is

$$F = (p - p_a) \cdot A$$

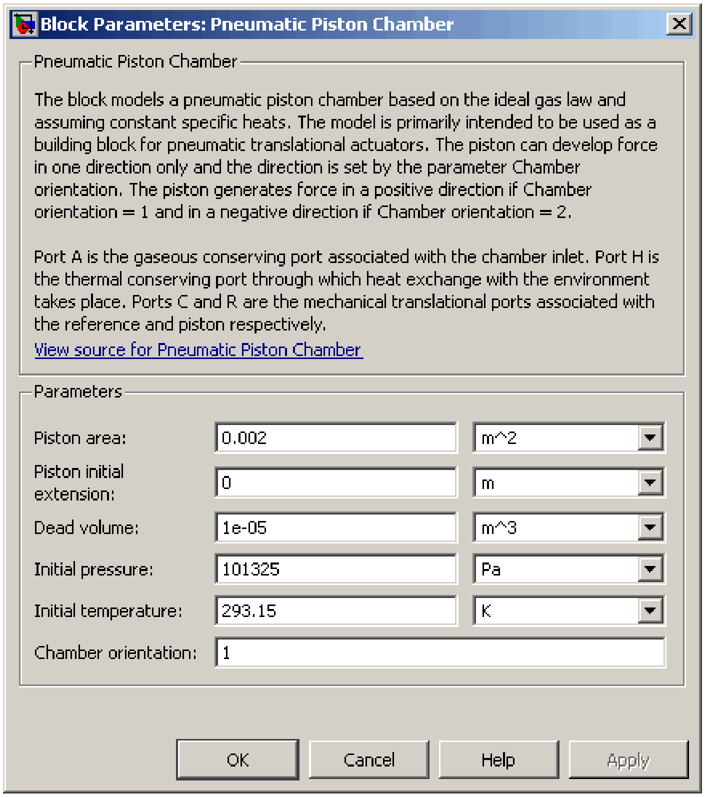
where p_a is the atmospheric pressure acting on the outside of the piston.

Port A is the pneumatic conserving port associated with the chamber inlet. Port H is a thermal conserving port through which heat exchange with the environment takes place. Ports C and R are mechanical translational conserving ports associated with the piston case and rod, respectively. The gas flow and the heat flow are considered positive if they flow into the chamber.

Basic Assumptions and Limitations

- The gas is ideal.
- Specific heats at constant pressure and constant volume, c_p and c_v, are constant.

Dialog Box and Parameters



Block Parameters: Pneumatic Piston Chamber

Pneumatic Piston Chamber

The block models a pneumatic piston chamber based on the ideal gas law and assuming constant specific heats. The model is primarily intended to be used as a building block for pneumatic translational actuators. The piston can develop force in one direction only and the direction is set by the parameter Chamber orientation. The piston generates force in a positive direction if Chamber orientation = 1 and in a negative direction if Chamber orientation = 2.

Port A is the gaseous conserving port associated with the chamber inlet. Port H is the thermal conserving port through which heat exchange with the environment takes place. Ports C and R are the mechanical translational ports associated with the reference and piston respectively.

View source for Pneumatic Piston Chamber

Parameters

Piston area:	0.002	m^2
Piston initial extension:	0	m
Dead volume:	1e-05	m^3
Initial pressure:	101325	Pa
Initial temperature:	293.15	K
Chamber orientation:	1	

Piston area

Specify the effective piston area. The default value is `.002` m^2.

Piston initial extension

Specify the initial offset of the piston from the cylinder cap. The default value is `0`.

Dead volume

Specify the volume of gas in the chamber at zero piston position. The default value is `1e-5` m^3.

Initial pressure

Specify the initial pressure in the chamber. This parameter specifies the initial condition for use in computing the initial state at the beginning of a simulation run. For more information, see "Initial Conditions Computation". The default value is `101235` Pa.

Initial temperature

Specify the initial temperature of the gas in the chamber. This parameter specifies the initial condition for use in computing the initial state at the beginning of a simulation run. For more information, see "Initial Conditions Computation". The default value is `293.15` K.

Chamber orientation

Specify the direction of force generation. The piston generates force in a positive direction if this parameter is set to `1` (the default). If you set this parameter to `2`, the piston generates force in a negative direction.

Ports

The block has the following ports:

`A`

Pneumatic conserving port associated with the chamber inlet.

`H`

Thermal conserving port through which heat exchange with the environment takes place.

`R`

Mechanical translational conserving port associated with the piston (rod).

`C`

Mechanical translational conserving port associated with the reference (case).

See Also

Constant Volume Pneumatic Chamber

Rotary Pneumatic Piston Chamber

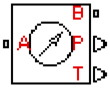
Purpose

Ideal pressure and temperature sensor

Library

Pneumatic Sensors

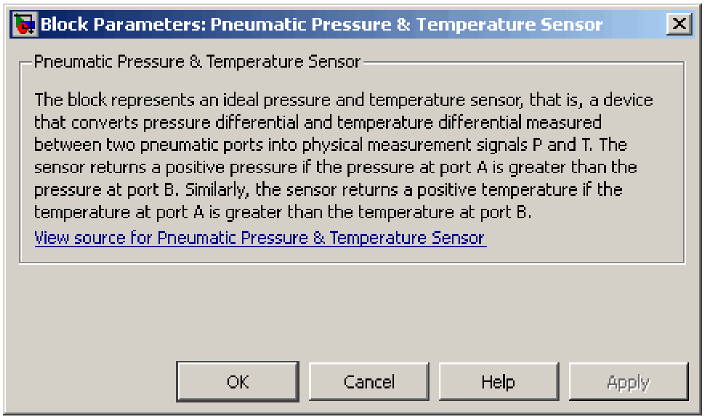
Description



The Pneumatic Pressure & Temperature Sensor block represents an ideal pressure and temperature sensor, that is, a device that converts pressure differential and temperature differential measured between two pneumatic ports into physical measurement signals P and T, respectively.

The sensor positive direction is from port A to port B. This means that the sensor returns a positive pressure if the pressure at port A is greater than the pressure at port B. Similarly, the sensor returns a positive temperature if the temperature at port A is greater than the temperature at port B.

Dialog Box and Parameters


Block Parameters: Pneumatic Pressure & Temperature Sensor
Pneumatic Pressure & Temperature Sensor
OK
Cancel
Help
Apply

The block represents an ideal pressure and temperature sensor, that is, a device that converts pressure differential and temperature differential measured between two pneumatic ports into physical measurement signals P and T. The sensor returns a positive pressure if the pressure at port A is greater than the pressure at port B. Similarly, the sensor returns a positive temperature if the temperature at port A is greater than the temperature at port B.

View source for Pneumatic Pressure & Temperature Sensor

The block has no parameters.

Ports

The block has the following ports:

A

Pneumatic conserving port associated with the sensor inlet.

B

Pneumatic conserving port associated with the sensor outlet.

See Also

Controlled Pneumatic Pressure Source

Pneumatic Pressure Source

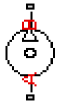
Purpose

Ideal compressor with constant pressure difference

Library

Pneumatic Sources

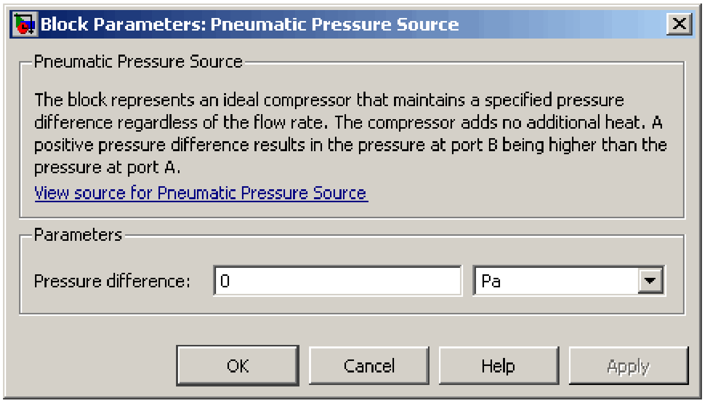
Description



The Pneumatic Pressure Source block represents an ideal compressor that maintains a specified pressure difference regardless of the flow rate. Use this block when pressure of an actual device is practically independent of the source flow rate, for example, in factory network outlets or large capacity receivers. The compressor adds no heat. Block connections A and B correspond to the pneumatic inlet and outlet ports, respectively.

A positive pressure difference results in the pressure at port B being higher than the pressure at port A.

Dialog Box and Parameters



Block Parameters: Pneumatic Pressure Source

Pneumatic Pressure Source

The block represents an ideal compressor that maintains a specified pressure difference regardless of the flow rate. The compressor adds no additional heat. A positive pressure difference results in the pressure at port B being higher than the pressure at port A.

View source for Pneumatic Pressure Source

Parameters

Pressure difference: 0 Pa

Pressure difference

Specify the pressure difference across the source. The default value is 0.

Ports

The block has the following ports:

A

Pneumatic conserving port associated with the source inlet.

B

Pneumatic conserving port associated with the source outlet.

See Also

Controlled Pneumatic Pressure Source

Pneumatic Pressure & Temperature Sensor

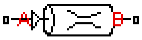
Purpose

Pneumatic pipe accounting for pressure loss and added heat due to flow resistance

Library

Pneumatic Elements

Description



The Pneumatic Resistive Tube block models the loss in pressure and heating due to viscous friction along a short stretch of pipe with circular cross section. Use this block with the Constant Volume Pneumatic Chamber block to build a model of a pneumatic transmission line.

The tube is simulated according to the following equations:

$$p_i - p_o = \begin{cases} \frac{RT_i}{p_i} \square \frac{32\mu L}{AD^2} \square G & \text{for } Re < Re_{lam} \text{(laminar flow)} \\ f \square \frac{RT_i}{p_i} \square \frac{L}{D} \square \frac{G^2}{2A^2} & \text{for } Re > Re_{turb} \text{(turbulent flow)} \end{cases}$$

where

p_i, p_o	Absolute pressures at the tube inlet and outlet, respectively. The inlet and outlet change depending on flow direction. For positive flow ($G > 0$), $p_i = p_A$, otherwise $p_i = p_B$.
T_i, T_o	Absolute gas temperatures at the tube inlet and outlet, respectively
G	Mass flow rate
μ	Gas viscosity
f	Friction factor for turbulent flow
D	Tube internal diameter
A	Tube cross-sectional area
L	Tube length
Re	Reynolds number

The friction factor for turbulent flow is approximated by the Haaland function

$$f = \left(-1.8 \log_{10} \left(\frac{6.9}{\mathrm{Re}} + \left(\frac{e}{3.7D} \right)^{1.11} \right) \right)^{-2}$$

where e is the surface roughness for the pipe material.

The Reynolds number is defined as:

$$\mathrm{Re} = \rho v D / \mu$$

where ρ is the gas density and v is the gas velocity. Gas velocity is related to mass flow rate by

$$G = \rho v A$$

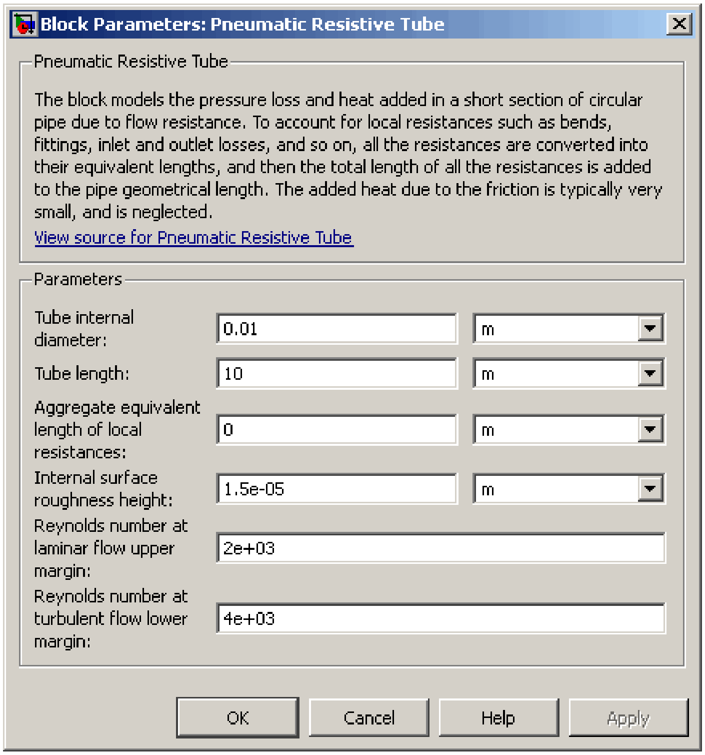
For flows between Re_{lam} and Re_{turb}, a linear blend is implemented between the flow predicted by the two equations.

In a real pipe, loss in kinetic energy due to friction is turned into added heat energy. However, the amount of heat is very small, and is neglected in the Pneumatic Resistive Tube block. Therefore, $q_i = q_o$, where q_i and q_o are the input and output heat flows, respectively.

Basic Assumptions and Limitations

- The gas is ideal.
- The pipe has a circular cross section.
- The process is adiabatic, that is, there is no heat transfer with the environment.
- Gravitational effects can be neglected.
- The flow resistance adds no net heat to the flow.

Dialog Box and Parameters



Block Parameters: Pneumatic Resistive Tube

Pneumatic Resistive Tube

The block models the pressure loss and heat added in a short section of circular pipe due to flow resistance. To account for local resistances such as bends, fittings, inlet and outlet losses, and so on, all the resistances are converted into their equivalent lengths, and then the total length of all the resistances is added to the pipe geometrical length. The added heat due to the friction is typically very small, and is neglected.

View source for Pneumatic Resistive Tube

Parameters

Tube internal diameter: 0.01 m

Tube length: 10 m

Aggregate equivalent length of local resistances: 0 m

Internal surface roughness height: 1.5e-05 m

Reynolds number at laminar flow upper margin: 2e+03

Reynolds number at turbulent flow lower margin: 4e+03

Tube internal diameter

Internal diameter of the tube. The default value is `0.01` m.

Tube length

Tube geometrical length. The default value is `10` m.

Aggregate equivalent length of local resistances

This parameter represents total equivalent length of all local resistances associated with the tube. You can account for the pressure loss caused by local resistances, such as bends, fittings, armature, inlet/outlet losses, and so on, by adding to the pipe geometrical length an aggregate equivalent length of all the local resistances. The default value is `0`.

Internal surface roughness height
Roughness height on the tube internal surface. The parameter is typically provided in data sheets or manufacturer catalogs. The default value is `1.5e-5` m, which corresponds to drawn tubing.

Reynolds number at laminar flow upper margin
Specifies the Reynolds number at which the laminar flow regime is assumed to start converting into turbulent flow. Mathematically, this value is the maximum Reynolds number at fully developed laminar flow. The default value is `2000`.

Reynolds number at turbulent flow lower margin
Specifies the Reynolds number at which the turbulent flow regime is assumed to be fully developed. Mathematically, this value is the minimum Reynolds number at turbulent flow. The default value is `4000`.

Ports

The block has the following ports:

A
Pneumatic conserving port associated with the tube inlet for positive flow.

B
Pneumatic conserving port associated with the tube outlet for positive flow.

See Also

Constant Volume Pneumatic Chamber

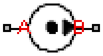
Purpose

Ideal compressor that generates a constant pressure differential across its inlets

Library

Thermal Liquid/Sources

Description



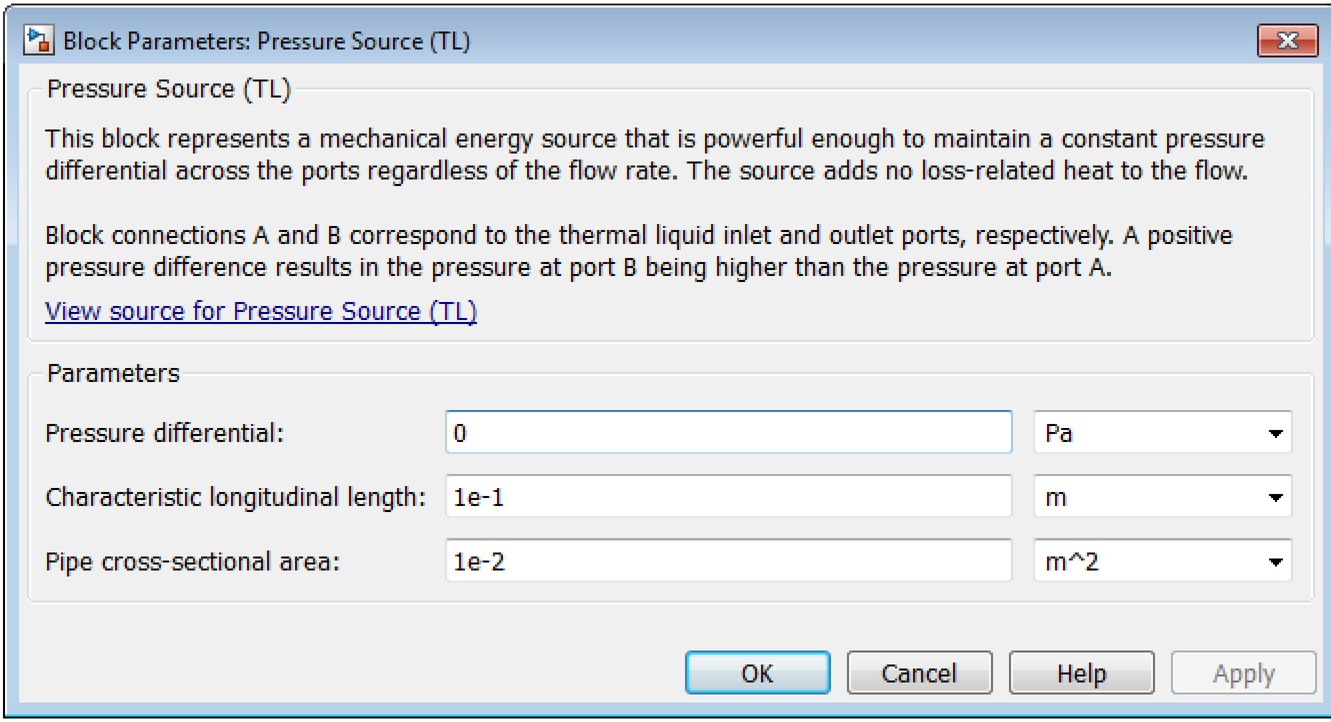
The Pressure Source (TL) block represents an ideal compressor generating a constant pressure differential across its inlets. The compressor can hold the specified pressure differential regardless of its mass flow rate. No heat exchange occurs between the compressor and the flowing liquid.

Ports A and B represent the compressor inlets. Both ports are thermal liquid conserving ports. The pressure differential is positive when the pressure at inlet B is greater than the pressure at inlet A.

Assumptions and Limitations

- No heat exchange occurs between the compressor and the flowing liquid.
- The pressure differential between the compressor inlets is independent of the mass flow rate through those inlets.

Dialog Box and Parameters



Block Parameters: Pressure Source (TL)

Pressure Source (TL)

This block represents a mechanical energy source that is powerful enough to maintain a constant pressure differential across the ports regardless of the flow rate. The source adds no loss-related heat to the flow.

Block connections A and B correspond to the thermal liquid inlet and outlet ports, respectively. A positive pressure difference results in the pressure at port B being higher than the pressure at port A.

View source for Pressure Source (TL)

Parameters

Pressure differential:	0	Pa
Characteristic longitudinal length:	1e-1	m
Pipe cross-sectional area:	1e-2	m^2

Pressure differential

Enter the pressure difference between compressor inlets A and B. The default value is 0 Pa.

Characteristic longitudinal length

Enter the mean path length the liquid must flow through to go from inlet A to inlet B. The default value is 1e-1 m.

Pipe cross-sectional area

Enter the cross-sectional area of the adjoining pipes. The default value is `1e-2` m^2.

Ports

The block has two thermal liquid conserving ports, A and B.

See Also

Controlled Pressure Source (TL)

Mass Flow Rate Source (TL)

Controlled Mass Flow Rate Source (TL)

Pressure & Temperature Sensor (TL)

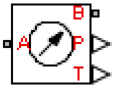
Purpose

Ideal sensor for measuring the relative pressure and temperature between two points

Library

Thermal Liquid/Sensors

Description



The Pressure & Temperature Sensor (TL) block represents an ideal sensor that measures pressure and temperature differences between two thermal liquid nodes. Because pressure and temperature are Across variables, the sensor block must connect in parallel with the component being measured.

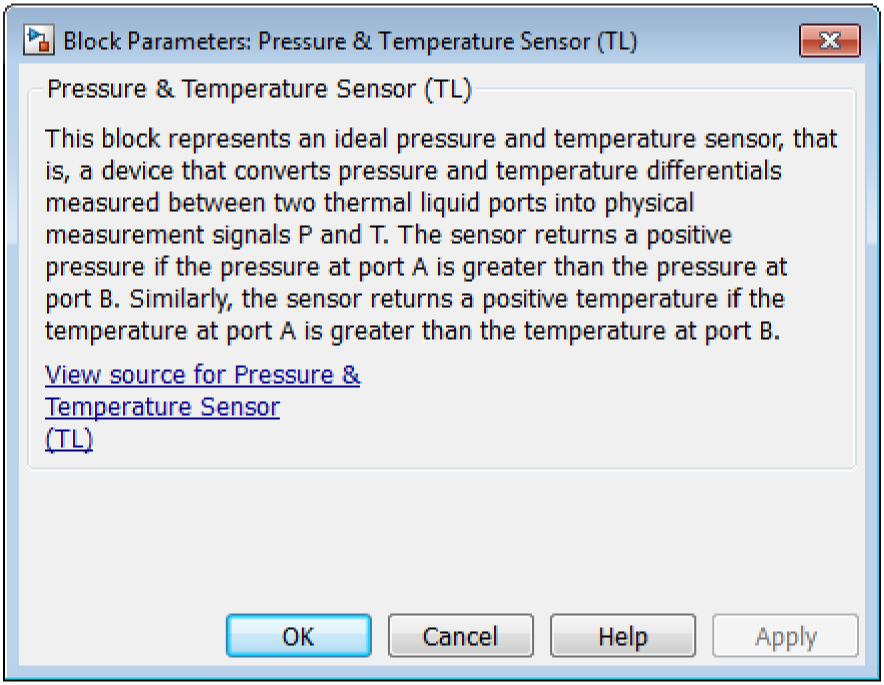
The relative orientation of ports A and B determines the measurement sign. The sign is positive if the measured quantity is greater at port B than it is at port A. Switching port connections reverses the measurement sign.

Ports P and T output the pressure and temperature measurements as physical signals. Connect the ports to PS-Simulink Converter blocks to transform the output physical signals into Simulink signals, e.g., for plotting or additional data processing.

Assumptions and Limitations

- Sensor inertia is negligible

Dialog Box and Parameters



Block Parameters: Pressure & Temperature Sensor (TL)

Pressure & Temperature Sensor (TL)

This block represents an ideal pressure and temperature sensor, that is, a device that converts pressure and temperature differentials measured between two thermal liquid ports into physical measurement signals P and T. The sensor returns a positive pressure if the pressure at port A is greater than the pressure at port B. Similarly, the sensor returns a positive temperature if the temperature at port A is greater than the temperature at port B.

View source for Pressure & Temperature Sensor (TL)

The block has no parameters.

Ports

The block has the following ports.

A, B	Thermal liquid conserving ports
P	Physical signal output port for pressure measurement
T	Physical signal output port for temperature measurement

See Also

Mass Flow Rate & Thermal Flux Sensor (TL)

PS Abs

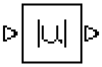
Purpose

Output absolute value of input physical signal

Library

Physical Signals/Nonlinear Operators

Description



The PS Abs block returns the absolute value of the input physical signal:

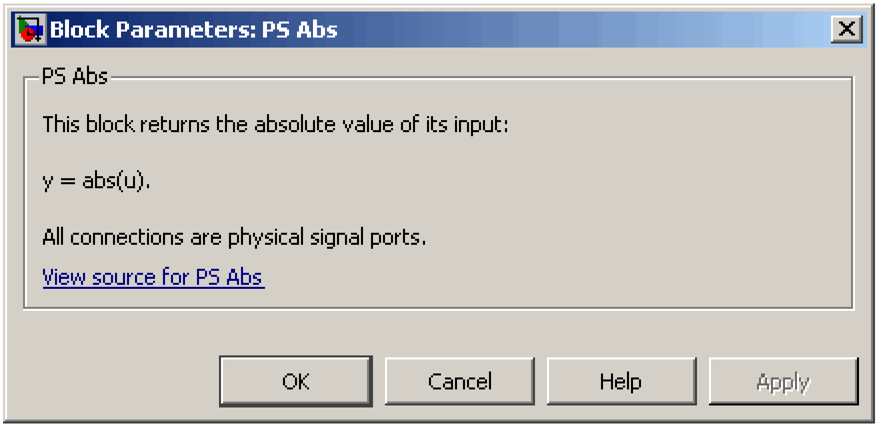
$$y = |u|$$

where

u	Physical signal at the input port
y	Physical signal at the output port

Both the input and the output are physical signals.

Dialog Box and Parameters



This block returns the absolute value of its input:

y = abs(u).

All connections are physical signal ports.

The PS Abs block has no parameters.

Ports

The block has one physical signal input port and one physical signal output port.

See Also

PS Dead Zone

PS Max

PS Min

PS Saturation

PS Sign

PS Add

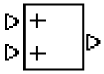
Purpose

Add two physical signal inputs

Library

Physical Signals/Functions

Description



The PS Add block outputs the sum of two input physical signals:

$$y = u_1 + u_2$$

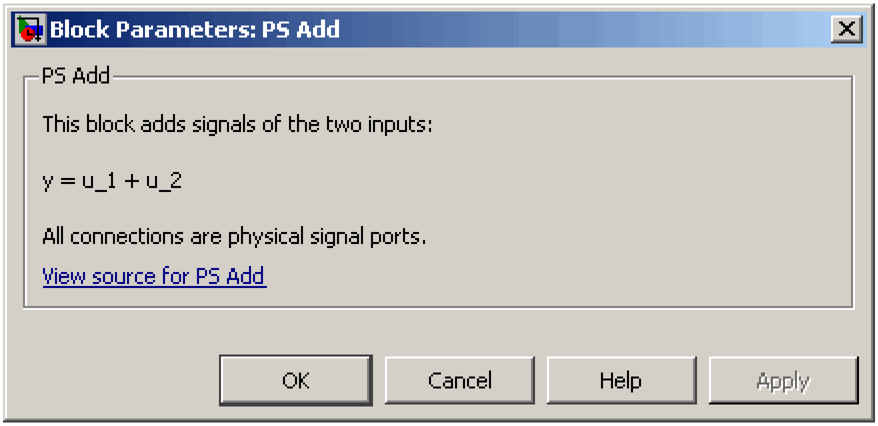
where

u_1 Physical signal at the first input port

u_2 Physical signal at the second input port

y Physical signal at the output port

Dialog Box and Parameters



Block Parameters: PS Add

PS Add

This block adds signals of the two inputs:

y = u_1 + u_2

All connections are physical signal ports.

View source for PS Add

The PS Add block has no parameters.

Ports

The block has two physical signal input ports and one physical signal output port.

See Also

PS Divide

PS Gain

PS Math Function

PS Product

PS Subtract

PS Ceil

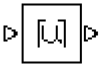
Purpose

Output the smallest integer larger than or equal to input physical signal

Library

Physical Signals/Nonlinear Operators

Description



The PS Ceil block rounds the input physical signal toward positive infinity, that is, to the nearest integer larger than or equal to the input value:

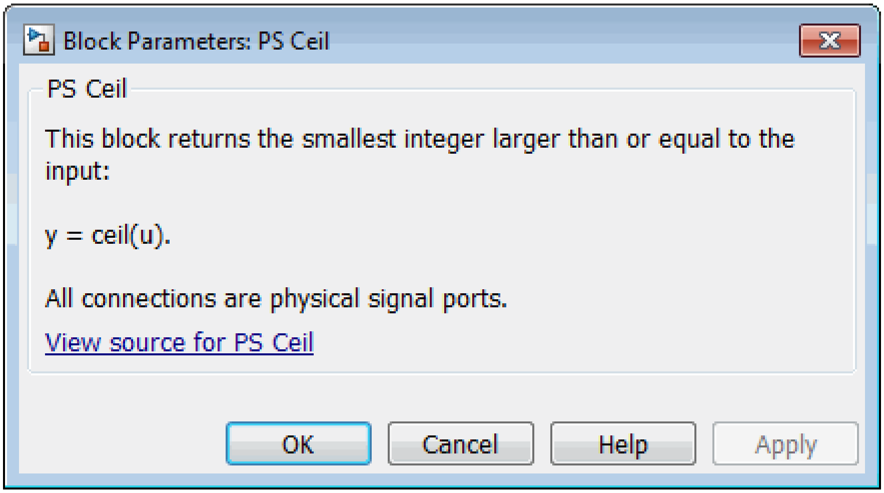
$$y = ceil(u)$$

where

u	Physical signal at the input port
y	Physical signal at the output port

Both the input and the output are physical signals.

Dialog Box and Parameters



Block Parameters: PS Ceil

PS Ceil

This block returns the smallest integer larger than or equal to the input:

y = ceil(u).

All connections are physical signal ports.

View source for PS Ceil

The PS Ceil block has no parameters.

Ports

The block has one physical signal input port and one physical signal output port.

See Also

`ceil`

PS Fix

PS Floor

PS Round

PS Constant

Purpose

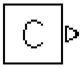
Generate constant physical signal

Library

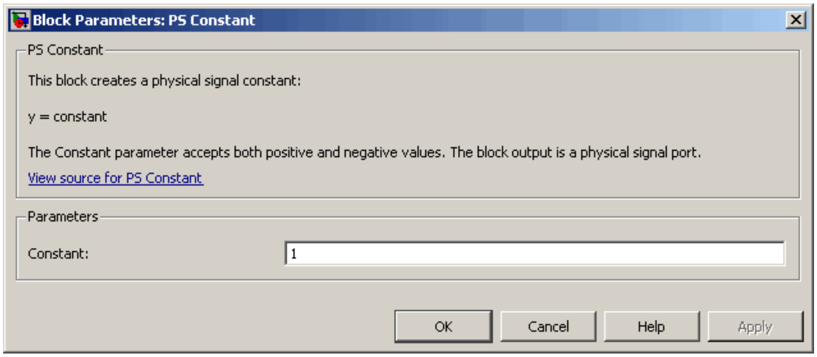
Physical Signals/Sources

Description

The PS Constant block generates a physical signal of a constant value. You specify the value of the signal as the **Constant** parameter.



Dialog Box and Parameters



Block Parameters: PS Constant

PS Constant

This block creates a physical signal constant:

y = constant

The Constant parameter accepts both positive and negative values. The block output is a physical signal port.

View source for PS Constant

Parameters

Constant: 1

Constant

The signal value. You can specify both positive and negative values.

Ports

The block has one physical signal output port.

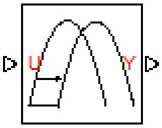
Purpose

Delay input physical signal by specified time

Library

Physical Signals/Delays

Description



The PS Constant Delay block generates the output physical signal, y, by delaying the input physical signal, u:

$$y = u\ (\ t - \tau\ )$$

where τ is the delay time.

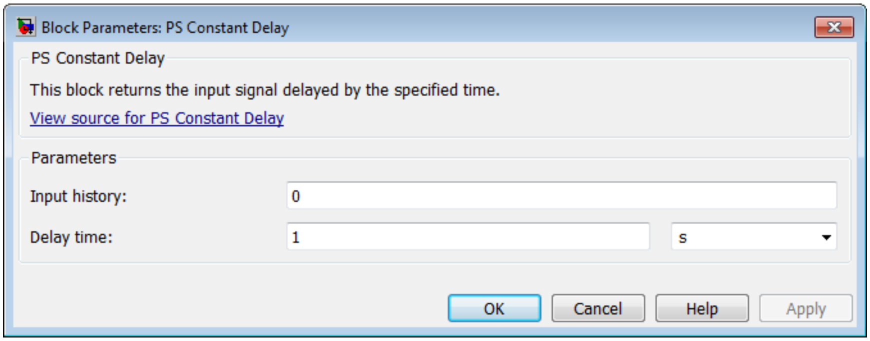
The delay time is constant throughout the simulation. You specify the value of the delay time as the **Delay time** parameter.

For the initial time interval, when t <= StartTime + τ, the block outputs the **Input history** parameter value.

Note

- When simulating a model that contains blocks with delays, memory allocation for storing the data history is controlled by the **Delay memory budget [kB]** parameter in the Solver Configuration block. If this budget is exceeded, simulation errors out. You can adjust this parameter value based on your available memory resources.
- For recommendation on how to linearize a model that contains blocks with delays, see “Linearizing with Simulink Linearization Blocks”.

Dialog Box and Parameters



Block Parameters: PS Constant Delay

PS Constant Delay

This block returns the input signal delayed by the specified time.

View source for PS Constant Delay

Parameters

Input history: 0

Delay time: 1 s

Input history

The output signal value during the initial time interval, until the specified delay time elapses after the start of simulation. The default value is `0`.

Delay time

The delay time for the signal. The parameter value must be positive. The default value is `1` s.

Ports

The block has one physical signal input port and one physical signal output port.

See Also

PS Variable Delay

PS Dead Zone

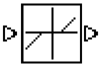
Purpose

Provide region of zero output for physical signals

Library

Physical Signals/Nonlinear Operators

Description

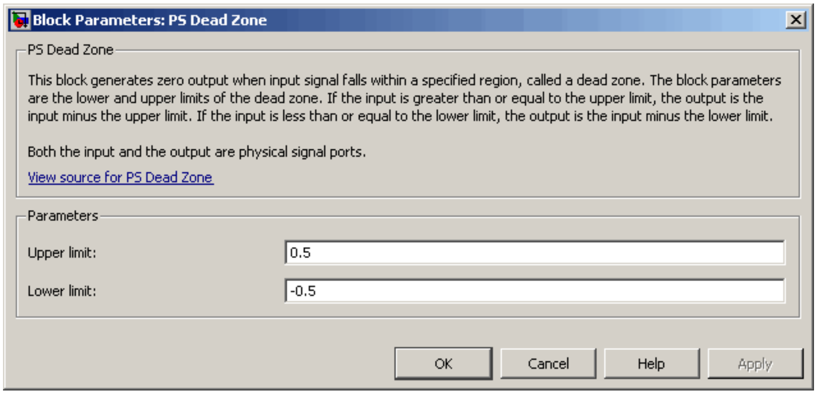


The PS Dead Zone block generates zero output when input signal falls within a specified region, called a dead zone. You can specify the lower and upper limits of the dead zone as block parameters. The block output depends on the input and dead zone:

- If the input is within the dead zone (greater than the lower limit and less than the upper limit), the output is zero.
- If the input is greater than or equal to the upper limit, the output is the input minus the upper limit.
- If the input is less than or equal to the lower limit, the output is the input minus the lower limit.

Both the input and the output are physical signals.

Dialog Box and Parameters



Block Parameters: PS Dead Zone

PS Dead Zone

This block generates zero output when input signal falls within a specified region, called a dead zone. The block parameters are the lower and upper limits of the dead zone. If the input is greater than or equal to the upper limit, the output is the input minus the upper limit. If the input is less than or equal to the lower limit, the output is the input minus the lower limit.

Both the input and the output are physical signal ports.

View source for PS Dead Zone

Parameters

Upper limit: 0.5

Lower limit: -0.5

Upper limit

The upper limit, or end, of the dead zone. The default value is `0.5`.

Lower limit

The lower limit, or start, of the dead zone. The default value is `-0.5`.

Ports

The block has one physical signal input port and one physical signal output port.

See Also

PS Abs

PS Max

PS Min

PS Saturation

PS Sign

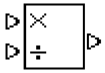
Purpose

Compute simple division of two input physical signals

Library

Physical Signals/Functions

Description



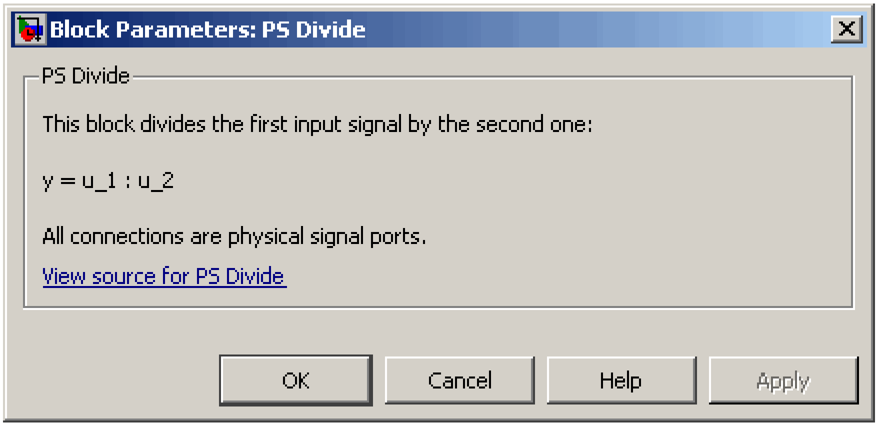
The PS Divide block divides one physical signal input by another and outputs the difference:

$$y = u_1 \div u_2$$

where

u_1	Physical signal at the first input port (marked with the x sign)
u_2	Physical signal at the second input port (marked with the ÷ sign)
y	Physical signal at the output port

Dialog Box and Parameters



This block divides the first input signal by the second one:

y = u_1 : u_2

All connections are physical signal ports.

View source for PS Divide

The PS Divide block has no parameters.

Ports

The block has two physical signal input ports and one physical signal output port.

See Also

PS Add

PS Gain

PS Math Function

PS Product

PS Subtract

PS Fix

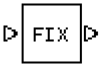
Purpose

Round input physical signal toward zero

Library

Physical Signals/Nonlinear Operators

Description



The PS Fix block rounds the input physical signal toward zero, that is, for a positive signal returns the nearest integer smaller than or equal to the input value, and for a negative signal returns the nearest integer larger than or equal to the input value:

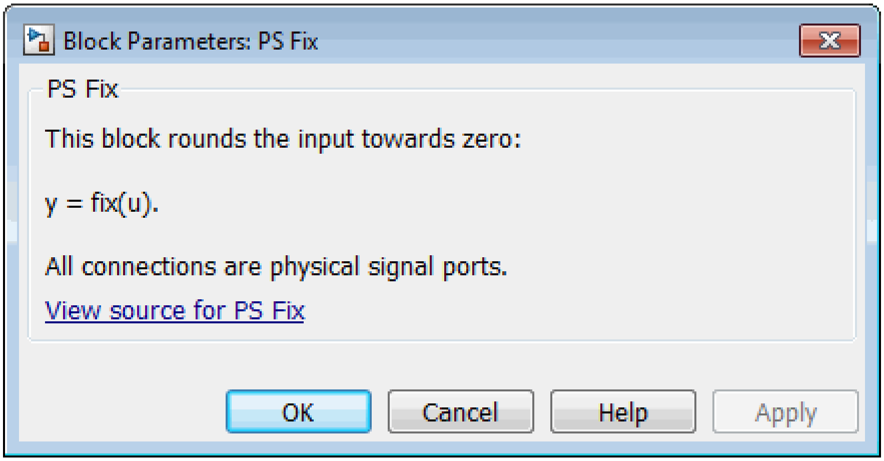
$$y = fix(u)$$

where

u	Physical signal at the input port
y	Physical signal at the output port

Both the input and the output are physical signals.

Dialog Box and Parameters



This block rounds the input towards zero:

y = fix(u).

All connections are physical signal ports.

View source for PS Fix

The PS Fix block has no parameters.

Ports

The block has one physical signal input port and one physical signal output port.

See Also

`fix`

PS Ceil

PS Floor

PS Round

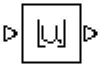
Purpose

Output the largest integer smaller than or equal to input physical signal

Library

Physical Signals/Nonlinear Operators

Description



The PS Floor block rounds the input physical signal toward negative infinity, that is, to the nearest integer smaller than or equal to the input value:

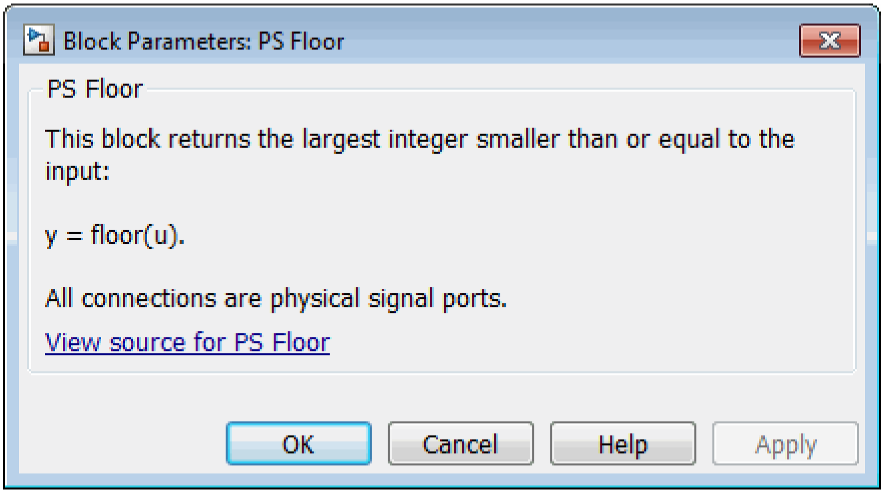
$$y = floor(u)$$

where

u	Physical signal at the input port
y	Physical signal at the output port

Both the input and the output are physical signals.

Dialog Box and Parameters



PS Floor

This block returns the largest integer smaller than or equal to the input:

y = floor(u).

All connections are physical signal ports.

View source for PS Floor

The PS Floor block has no parameters.

Ports

The block has one physical signal input port and one physical signal output port.

See Also

`floor`

PS Ceil

PS Fix

PS Round

Purpose

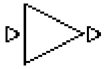
Multiply input physical signal by constant

Library

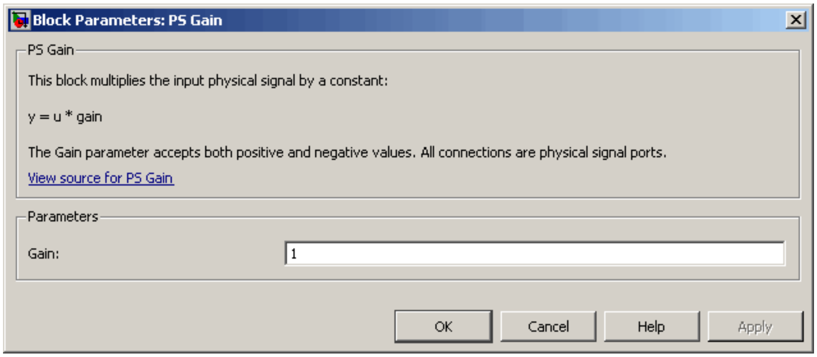
Physical Signals/Functions

Description

The PS Gain block multiplies the input physical signal by a constant value (gain). You specify the value of the gain as the **Gain** parameter.



Dialog Box and Parameters



Block Parameters: PS Gain

PS Gain

This block multiplies the input physical signal by a constant:

y = u * gain

The Gain parameter accepts both positive and negative values. All connections are physical signal ports.

View source for PS Gain

Parameters

Gain: 1

Gain

The multiplication coefficient. You can specify both positive and negative values.

Ports

The block has one physical signal input port and one physical signal output port.

See Also

PS Add

PS Divide

PS Math Function

PS Product

PS Subtract

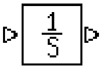
Purpose

Integrate physical signal

Library

Physical Signals/Linear Operators

Description



The PS Integrator block outputs the integral of its input at the current time step. The following equation represents the output of the block:

$$y(t) = \int_{t_0}^{t} u(t)dt + y_0$$

where

u	Physical signal at the input port
y_0	Initial condition
y	Physical signal at the output port
t	Time

The PS Integrator block is a dynamic system with one state, its output. The PS Integrator block's input is the state's time derivative:

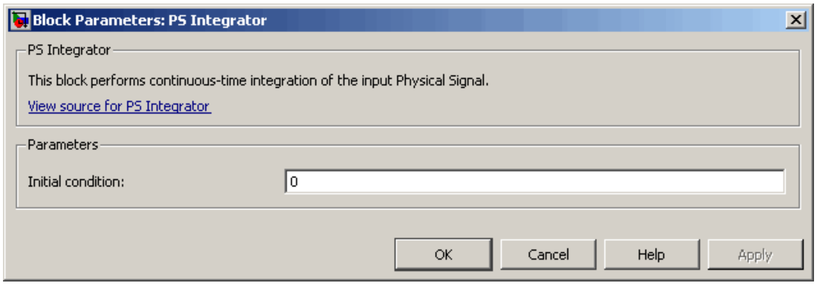
$$x = y(t)$$

$$x_0 = y_0$$

$$\dot{x} = u(t)$$

The solver computes the output of the PS Integrator block at the current time step, using the current input value and the value of the state at the previous time step. To support this computational model, the PS Integrator block saves its output at the current time step for use by the solver to compute its output at the next time step. The block also provides the solver with an initial condition for use in computing the block's initial state at the beginning of a simulation run. The default value of the initial condition is 0. You can specify another value for the initial condition as a parameter on the block dialog box.

Dialog Box and Parameters



Block Parameters: PS Integrator

PS Integrator

This block performs continuous-time integration of the input Physical Signal.

View source for PS Integrator

Parameters

Initial condition: 0

Initial Condition

Specify the initial condition for use in computing the block's initial state at the beginning of a simulation run. For more information, see "Initial Conditions Computation". The default value is `0`.

Ports

The block has one physical signal input port and one physical signal output port.

PS Lookup Table (1D)

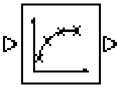
Purpose

Approximate one-dimensional function using specified lookup method

Library

Physical Signals/Lookup Tables

Description



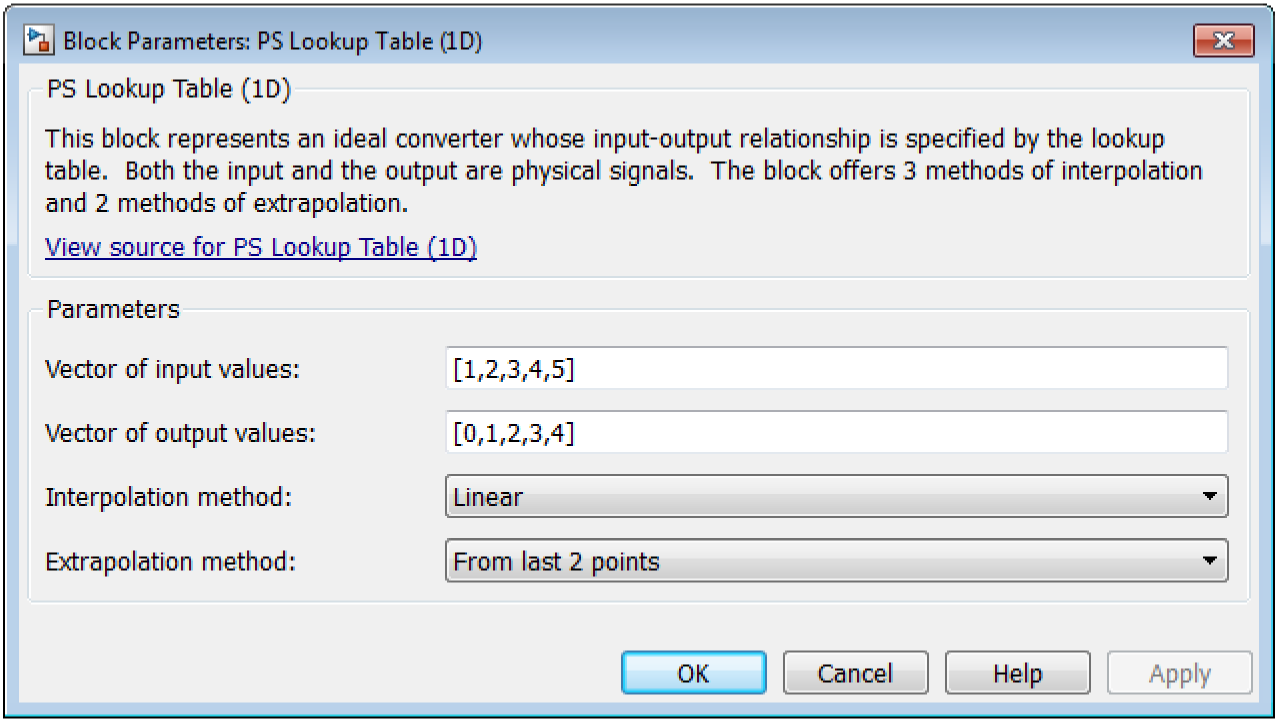
The PS Lookup Table (1D) block computes an approximation to some function `y=f(x)` given data vectors `x` and `y`. Both the input and the output are physical signals.

Note To map two physical signal inputs to an output, use the PS Lookup Table (2-D) block.

The length of the `x` and `y` data vectors provided to this block must match. Also, the `x` data vector must be *strictly increasing* (i.e., the value of the next element in the vector is greater than the value of the preceding element).

You define the lookup table by specifying the **Vector of input values** parameter as a 1-by-`n` vector and the **Vector of output values** parameter as a 1-by-`n` vector. The block generates output based on the input values using the selected interpolation and extrapolation methods. You have a choice of three interpolation methods and two extrapolation methods.

Dialog Box and Parameters



Block Parameters: PS Lookup Table (1D)

PS Lookup Table (1D)

This block represents an ideal converter whose input-output relationship is specified by the lookup table. Both the input and the output are physical signals. The block offers 3 methods of interpolation and 2 methods of extrapolation.

View source for PS Lookup Table (1D)

Parameters

Vector of input values: [1,2,3,4,5]

Vector of output values: [0,1,2,3,4]

Interpolation method: Linear

Extrapolation method: From last 2 points

Vector of input values

Specify the vector of input values as a one-dimensional array. The input values vector must be strictly increasing. The values can be nonuniformly spaced.

Vector of output values

Specify the vector of output values as a one-dimensional array. The output values vector must be of the same size as the input values vector.

Interpolation method

Select one of the following interpolation methods for approximating the output value when the input value is between two consecutive grid points:

- `Linear` — Uses a linear function.
- `Cubic` — Uses the Piecewise Cubic Hermite Interpolation Polinomial (PCHIP). For more information, see [1] and the `pchip` MATLAB function.
- `Spline` — Uses the cubic spline interpolation algorithm described in [2].

Extrapolation method

Select one of the following extrapolation methods for determining the output value when the input value is outside the range specified in the argument list:

- `From last 2 points` — Extrapolates using the linear method (regardless of the interpolation method specified), based on the last two output values at the appropriate end of the range. That is, the block uses the first and second specified output values if the input value is below the specified range, and the two last specified output values if the input value is above the specified range.
- `From last point` — Uses the last specified output value at the appropriate end of the range. That is, the block uses the last specified output value for all input values greater than the last specified input argument, and the first specified output value for all input values less than the first specified input argument.

Restricted Parameters

When your model is in Restricted editing mode, you cannot modify the following parameters:

- **Interpolation method**
- **Extrapolation method**

All other block parameters are available for modification.

Ports

The block has one physical signal input port and one physical signal output port.

References

[1] D. Kahaner, Cleve Moler, Stephen Nash, *Numerical Methods and Software*, Prentice Hall, 1988

[2] W.H. Press, B.P. Flannery, S.A. Teulkolsky, W.T. Wetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1992

See Also

PS Lookup Table (2D)

PS Lookup Table (2D)

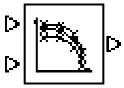
Purpose

Approximate two-dimensional function using specified lookup method

Library

Physical Signals/Lookup Tables

Description

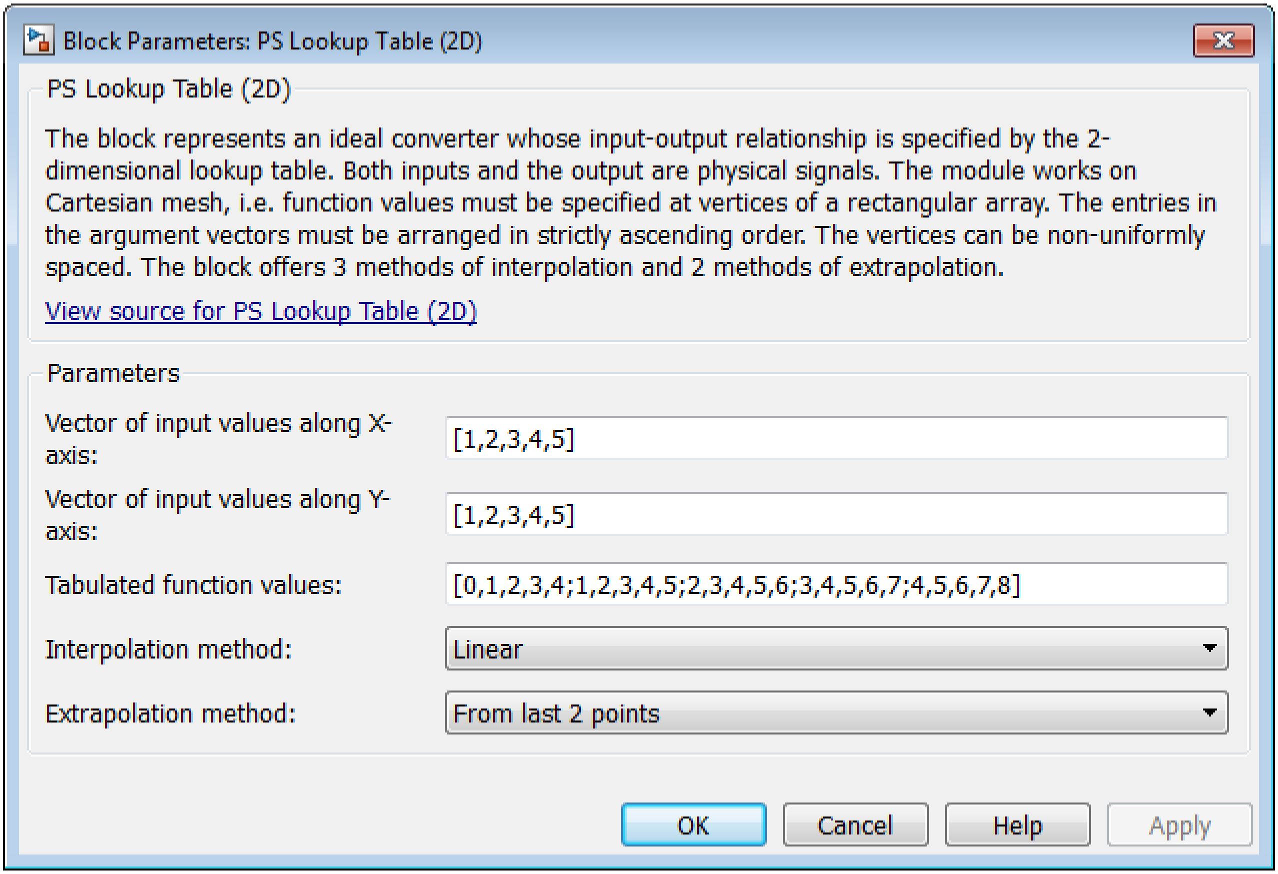


The PS Lookup Table (2D) block computes an approximation to some function `z=f(x,y)` given the `x`, `y`, `z` data points. The two inputs and the output are physical signals.

The `x` and `y` data vectors must be *strictly increasing* (i.e., the value of the next element in the vector is greater than the value of the preceding element). The matrix size of the tabulated function values must match the dimensions defined by the input vectors.

You define the lookup table by specifying the **Vector of input values along X-axis** parameter as a 1-by-`m` vector of `x` data points, the **Vector of input values along Y-axis** parameter as a 1-by-`n` vector of `y` data points, and the **Tabulated function values** as an `m`-by-`n` matrix of `z` data points. The block works on Cartesian mesh, i.e., function values must be specified at vertices of a rectangular array. The block generates output based on the input grid lookup using the selected interpolation and extrapolation methods. You have a choice of three interpolation methods and two extrapolation methods.

Dialog Box and Parameters



PS Lookup Table (2D)

The block represents an ideal converter whose input-output relationship is specified by the 2-dimensional lookup table. Both inputs and the output are physical signals. The module works on Cartesian mesh, i.e. function values must be specified at vertices of a rectangular array. The entries in the argument vectors must be arranged in strictly ascending order. The vertices can be non-uniformly spaced. The block offers 3 methods of interpolation and 2 methods of extrapolation.

View source for PS Lookup Table (2D)

Parameters

Vector of input values along X-axis: [1,2,3,4,5]

Vector of input values along Y-axis: [1,2,3,4,5]

Tabulated function values: [0,1,2,3,4;1,2,3,4,5;2,3,4,5,6;3,4,5,6,7;4,5,6,7,8]

Interpolation method: Linear

Extrapolation method: From last 2 points

Vector of input values along X-axis

Specify the vector of input values along the *x*-axis as a 1-by-`m` array. The input values vector must be strictly increasing. The values can be nonuniformly spaced.

Vector of input values along Y-axis

Specify the vector of input values along the *y*-axis as a 1-by-`n` array. The input values vector must be strictly increasing. The values can be nonuniformly spaced.

Tabulated function values

Specify the output values as an `m`-by-`n` matrix, defining the function values at the input grid vertices. The matrix size must match the dimensions defined by the input vectors.

Interpolation method

Select one of the following interpolation methods for approximating the output value when the input value is between two consecutive grid points:

- `Linear` — Uses a bilinear interpolation algorithm, which is an extension of linear interpolation for functions in two variables. The method performs linear interpolation first in *x*-direction and then in *y*-direction.
- `Cubic` — Uses the bicubic interpolation algorithm described in [1].
- `Spline` — Uses the bicubic spline interpolation algorithm described in [1].

Extrapolation method

Select one of the following extrapolation methods for determining the output value when the input value is outside the range specified in the argument list:

- `From last 2 points` — Extrapolates using the linear method (regardless of the interpolation method specified) based on the last two output values at the appropriate grid location, similar to PS Lookup Table (1D) block.
- `From last point` — Uses the last specified output value at the appropriate grid location, similar to PS Lookup Table (1D) block..

Restricted Parameters

When your model is in Restricted editing mode, you cannot modify the following parameters:

- **Interpolation method**
- **Extrapolation method**

All other block parameters are available for modification.

Ports

The block has two physical signal input ports and one physical signal output port.

References

[1] W.H.Press, B.P.Flannery, S.A.Teulkolsky, W.T.Wetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1992

See Also

PS Lookup Table (1D)

PS Math Function

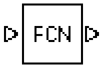
Purpose

Apply mathematical function to input physical signal

Library

Physical Signals/Functions

Description

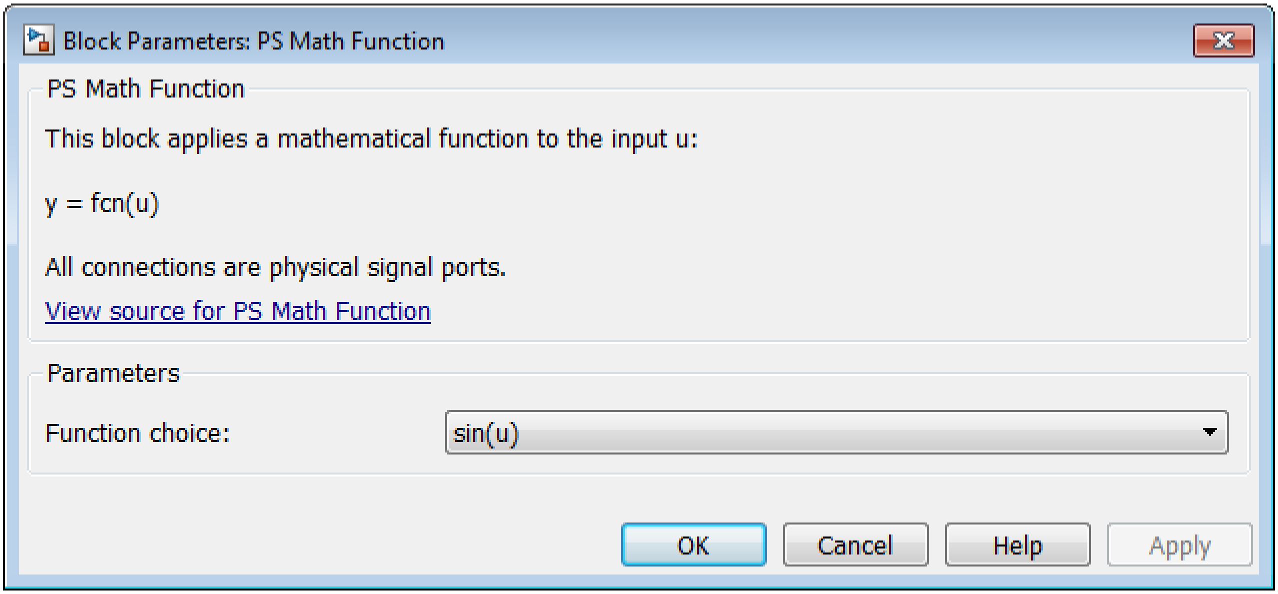


The PS Math Function block applies a mathematical function to the input physical signal, u. The block output is the result of the operation of the function on the input. You can select one of the following functions from the **Function choice** parameter list.

Function	Description	Mathematical Expression
`sin(u)`	Sinus	$\sin(u)$
`cos(u)`	Cosinus	$\cos(u)$
`exp(u)`	Exponential	e^u
`log(u)`	Natural logarithm	$\ln(u)$
`10^u`	Power of base 10	10^u
`log10(u)`	Common (base 10) logarithm	$\log(u)$
`u^2`	Power 2	u^2
`sqrt(u)`	Square root	$u^{0.5}$
`1/u`	Reciprocal	$1/u$

The PS Math Function block issues a simulation-time error when the input falls out of the expected domain for the particular function used. For example, if set to `sqrt`, the PS Math Function block issues an error if it receives negative input during simulation.

Dialog Box and Parameters



Block Parameters: PS Math Function

PS Math Function

This block applies a mathematical function to the input u:

y = fcn(u)

All connections are physical signal ports.

View source for PS Math Function

Parameters

Function choice: sin(u)

Function choice

Select the function to perform. The block output is the result of the operation of the function on the input.

Ports

The block has one physical signal input port and one physical signal output port.

See Also

PS Add

PS Divide

PS Gain

PS Product

PS Subtract

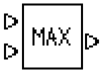
Purpose

Output maximum of two input physical signals

Library

Physical Signals/Nonlinear Operators

Description



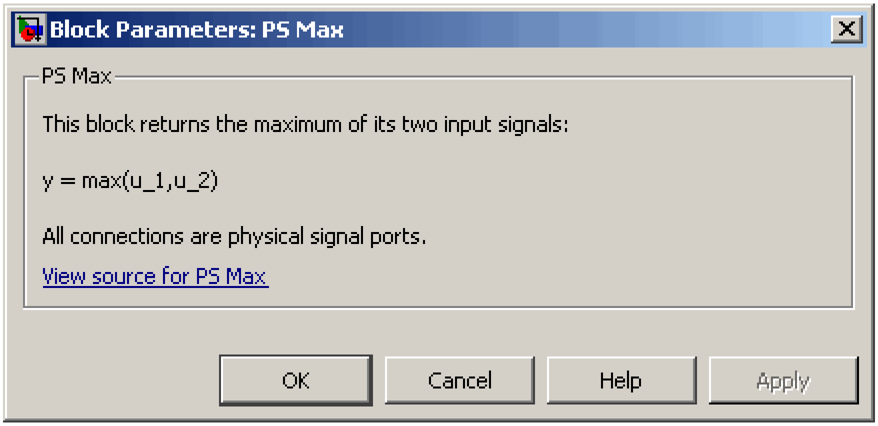
The PS Max block outputs the maximum of its two input physical signals:

$$y = \max(u_1, u_2)$$

where

u_1	Physical signal at the first input port
u_2	Physical signal at the second input port
y	Physical signal at the output port

Dialog Box and Parameters



This block returns the maximum of its two input signals:

y = max(u_1,u_2)

All connections are physical signal ports.

View source for PS Max

The PS Max block has no parameters.

Ports

The block has two physical signal input ports and one physical signal output port.

See Also

PS Abs

PS Dead Zone

PS Min

PS Saturation

PS Sign

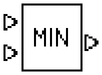
Purpose

Output minimum of two input physical signals

Library

Physical Signals/Nonlinear Operators

Description



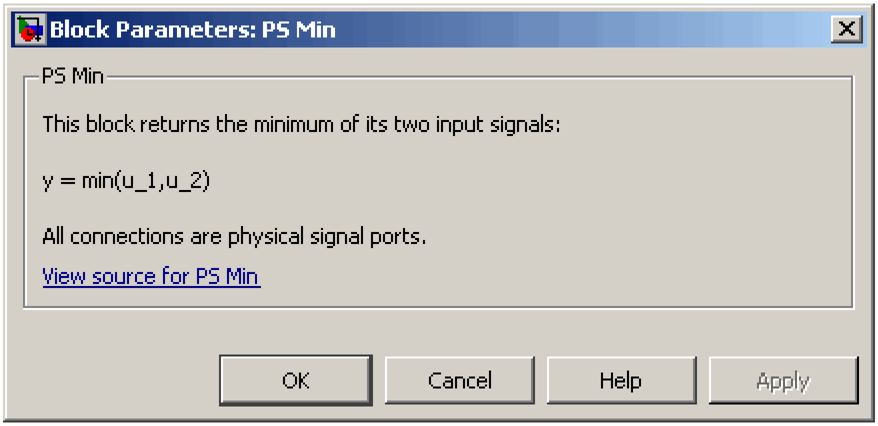
The PS Min block outputs the minimum of its two input physical signals:

$$y = \min(u_1, u_2)$$

where

u_1	Physical signal at the first input port
u_2	Physical signal at the second input port
y	Physical signal at the output port

Dialog Box and Parameters



This block returns the minimum of its two input signals:

y = min(u_1,u_2)

All connections are physical signal ports.

View source for PS Min

The PS Min block has no parameters.

Ports

The block has two physical signal input ports and one physical signal output port.

See Also

PS Abs

PS Dead Zone

PS Max

PS Saturation

PS Sign

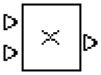
Purpose

Multiply two physical signal inputs

Library

Physical Signals/Functions

Description



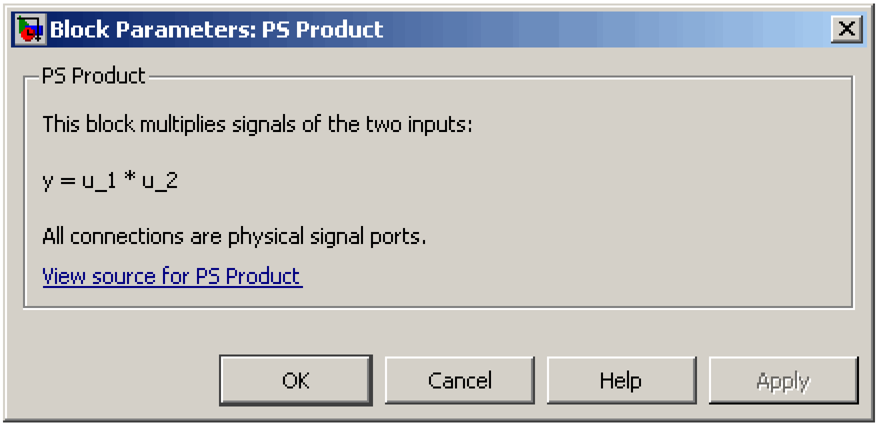
The PS Product block outputs the product of two input physical signals:

$$y = u_1 \Box u_2$$

where

u_1	Physical signal at the first input port
u_2	Physical signal at the second input port
y	Physical signal at the output port

Dialog Box and Parameters



Block Parameters: PS Product

PS Product

This block multiplies signals of the two inputs:

y = u_1 * u_2

All connections are physical signal ports.

View source for PS Product

The PS Product block has no parameters.

Ports

The block has two physical signal input ports and one physical signal output port.

See Also

PS Add

PS Divide

PS Gain

PS Math Function

PS Subtract

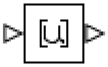
Purpose

Round input physical signal toward nearest integer

Library

Physical Signals/Nonlinear Operators

Description



The PS Round block rounds the input physical signal toward the nearest integer:

$$y = \mathrm{round}(\ u\ )$$

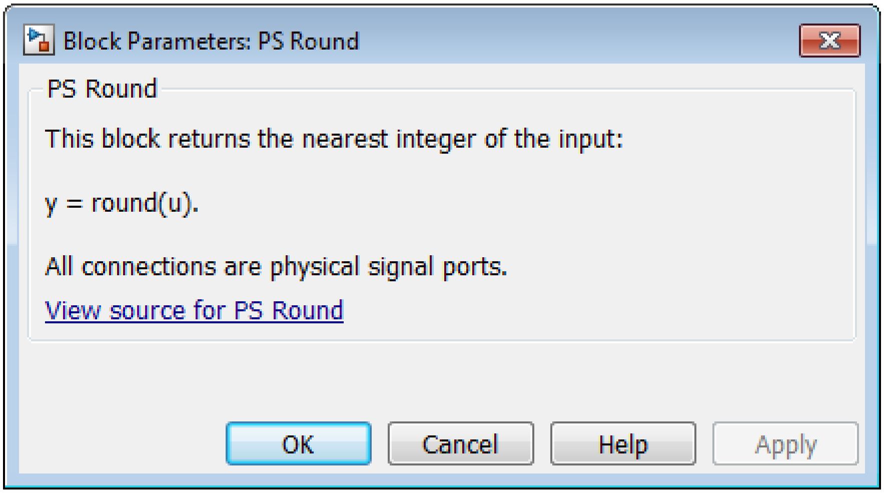
where

u	Physical signal at the input port
y	Physical signal at the output port

Positive signals with a fractional part of 0.5 round up to the nearest positive integer. Negative signals with a fractional part of -0.5 round down to the nearest negative integer.

Both the input and the output are physical signals.

Dialog Box and Parameters



Block Parameters: PS Round

PS Round

This block returns the nearest integer of the input:

y = round(u).

All connections are physical signal ports.

View source for PS Round

The PS Round block has no parameters.

Ports

The block has one physical signal input port and one physical signal output port.

See Also

`round`

PS Ceil

PS Fix

PS Floor

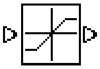
Purpose

Limit range of physical signal

Library

Physical Signals/Nonlinear Operators

Description

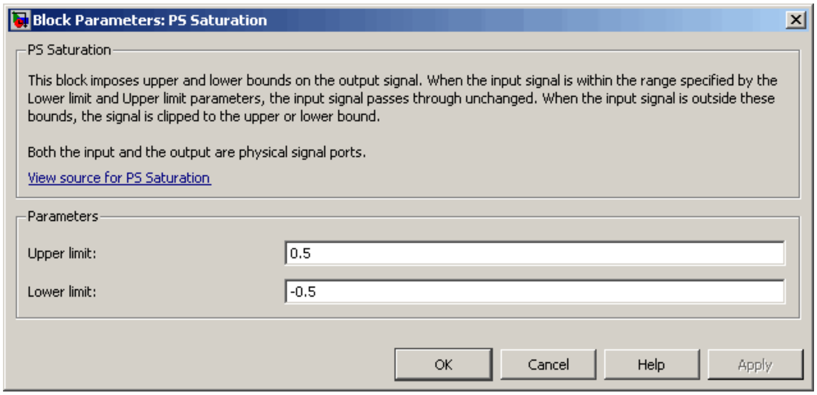


The PS Saturation block imposes upper and lower bounds on a physical signal. When the input signal is within the range specified by the **Lower limit** and **Upper limit** parameters, the input signal passes through unchanged. When the input signal is outside these bounds, the signal is clipped to the upper or lower bound.

When the **Lower limit** and **Upper limit** parameters are set to the same value, the block outputs that value.

Both the input and the output are physical signals.

Dialog Box and Parameters



PS Saturation

This block imposes upper and lower bounds on the output signal. When the input signal is within the range specified by the Lower limit and Upper limit parameters, the input signal passes through unchanged. When the input signal is outside these bounds, the signal is clipped to the upper or lower bound.

Both the input and the output are physical signal ports.

View source for PS Saturation

Parameters

Upper limit: 0.5

Lower limit: -0.5

Upper limit

The upper bound on the input signal. When the input signal to the PS Saturation block is above this value, the output of the block is clipped to this value. The default is `0.5`.

Lower limit

The lower bound on the input signal. When the input signal to the PS Saturation block is below this value, the output of the block is clipped to this value. The default is `-0.5`.

Ports

The block has one physical signal input port and one physical signal output port.

See Also

PS Abs

PS Dead Zone

PS Max

PS Min

PS Sign

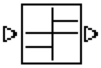
Purpose

Output sign of input physical signal

Library

Physical Signals/Nonlinear Operators

Description

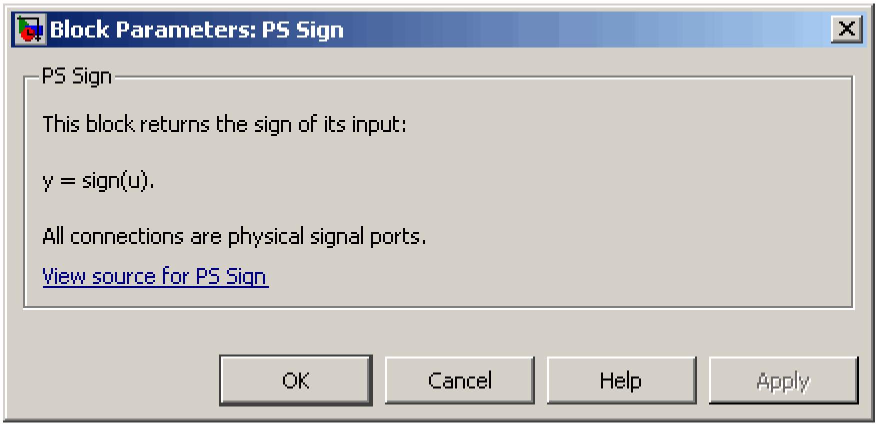


The PS Sign block returns the sign of the input physical signal:

- The output is 1 when the input is greater than zero.
- The output is 0 when the input is equal to zero.
- The output is –1 when the input is less than zero.

Both the input and the output are physical signals.

Dialog Box and Parameters



This block returns the sign of its input:

y = sign(u).

All connections are physical signal ports.

View source for PS Sign

The PS Sign block has no parameters.

Ports

The block has one physical signal input port and one physical signal output port.

See Also

PS Abs

PS Dead Zone

PS Max

PS Min

PS Saturation

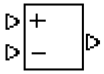
Purpose

Compute simple subtraction of two input physical signals

Library

Physical Signals/Functions

Description



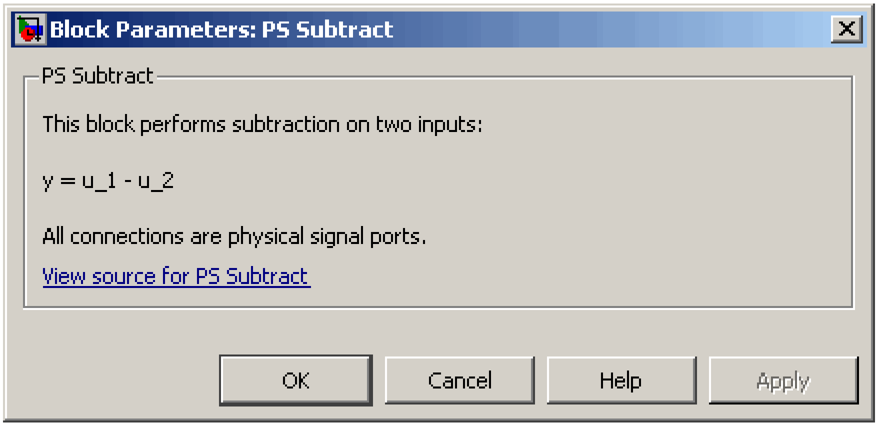
The PS Subtract block subtracts one physical signal input from another and outputs the difference:

$$y = u_1 - u_2$$

where

u_1	Physical signal at the first input port (marked with the plus sign)
u_2	Physical signal at the second input port (marked with the minus sign)
y	Physical signal at the output port

Dialog Box and Parameters



This block performs subtraction on two inputs:

y = u_1 - u_2

All connections are physical signal ports.

View source for PS Subtract

The PS Subtract block has no parameters.

Ports

The block has two physical signal input ports and one physical signal output port.

See Also

PS Add

PS Divide

PS Gain

PS Math Function

PS Product

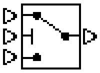
Purpose

Single-pole double-throw switch controlled by external physical signal

Library

Physical Signals/Nonlinear Operators

Description

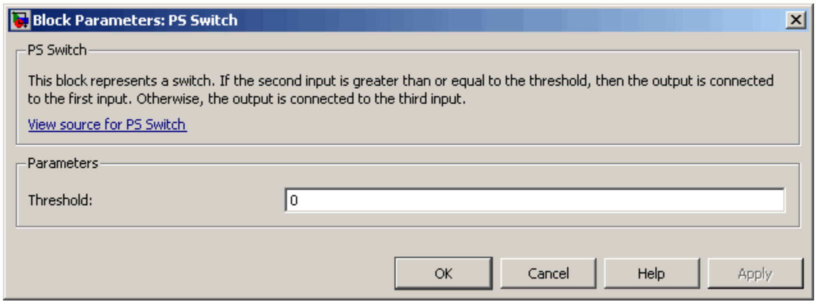


The PS Switch block compares the value of the physical signal presented at the second (middle) input port to the threshold value:

- If the control input value is greater than or equal to the threshold, the output is connected to the first input. This is the default connection shown in the block icon.
- If the control input value is less than the threshold, the output is connected to the third input.

The second (middle) input port is the control port and it never connects to the output. All the inputs and the output are physical signals.

Dialog Box and Parameters



Block Parameters: PS Switch

PS Switch

This block represents a switch. If the second input is greater than or equal to the threshold, then the output is connected to the first input. Otherwise, the output is connected to the third input.

View source for PS Switch

Parameters

Threshold: 0

Threshold

The threshold value for opening and closing the switch. If the control physical signal, presented at the second (middle) input port, is greater than or equal to this value, then the output is connected to the first input. Otherwise, the output is connected to the third input. The default value is `0`.

Ports

The block has three physical signal input ports and one physical signal output port.

See Also

Switch

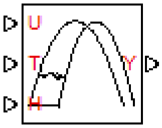
Purpose

Delay input physical signal by variable time

Library

Physical Signals/Delays

Description



The PS Variable Delay block generates the output physical signal, y, by delaying the input physical signal, u:

$$y = u\ (\ t - \tau\ )$$

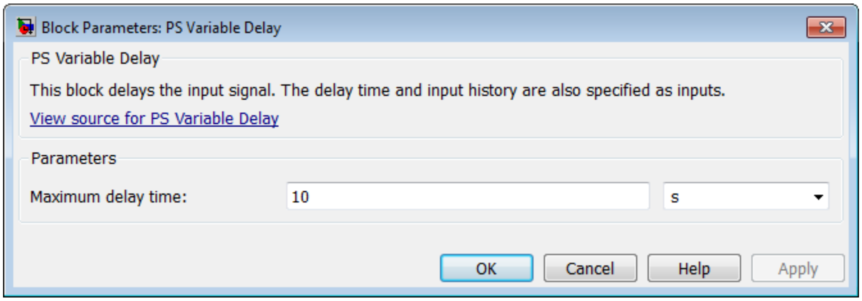
where τ is the delay time, which can vary throughout the simulation. You supply the delay time as a signal through the input port T.

For the initial time interval, when $t <=$ StartTime + τ, the block outputs the value of the signal supplied through the input port H.

Note

- When simulating a model that contains blocks with delays, memory allocation for storing the data history is controlled by the **Delay memory budget [kB]** parameter in the Solver Configuration block. If this budget is exceeded, simulation errors out. You can adjust this parameter value based on your available memory resources.
- For recommendation on how to linearize a model that contains blocks with delays, see "Linearizing with Simulink Linearization Blocks".

Dialog Box and Parameters



Block Parameters: PS Variable Delay

PS Variable Delay

This block delays the input signal. The delay time and input history are also specified as inputs.

View source for PS Variable Delay

Parameters

Maximum delay time: 10 s

Maximum delay time

The upper limit for the delay time. Exceeding the maximum delay time during simulation results in a runtime error. The parameter value must be positive. The default value is 10 s.

Ports

The block has the following ports:

U

Physical signal input port for the original signal.

T

Physical signal input port that supplies the delay time.

H

Physical signal input port that supplies the output signal for the initial time interval, when time since the start of simulation is less than or equal to the delay time.

Y

Physical signal output port for the delayed signal.

See Also

PS Constant Delay

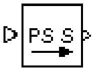
Purpose

Convert physical signal into Simulink output signal

Library

Utilities

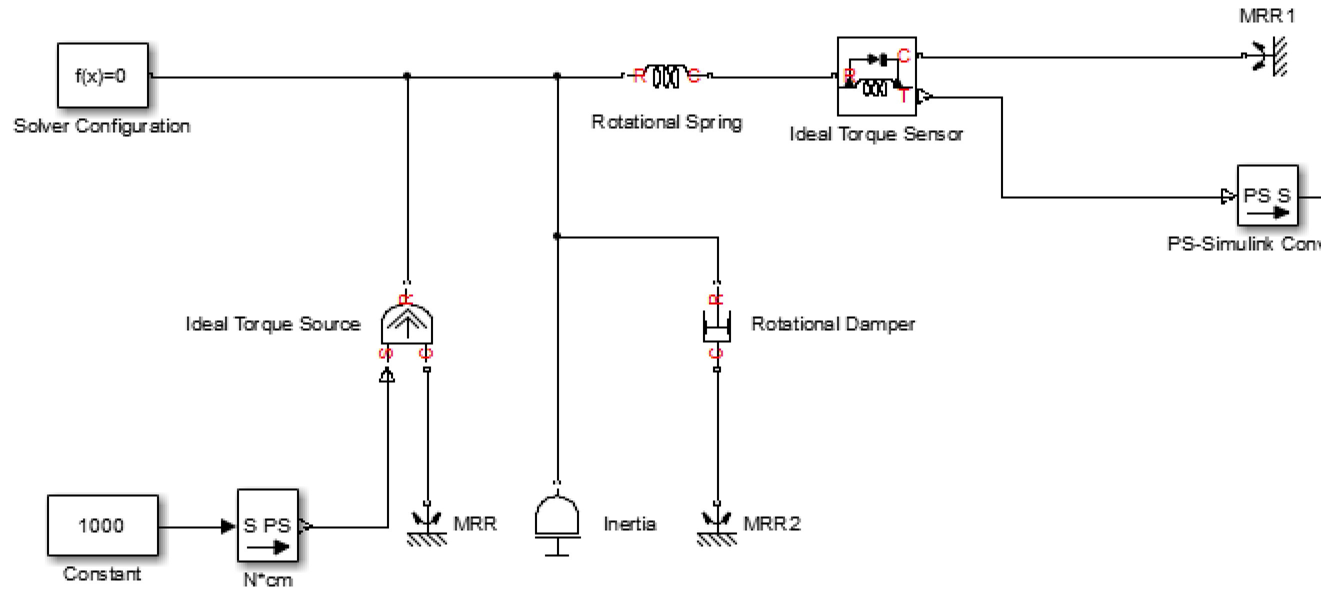
Description



The PS-Simulink Converter block converts a physical signal into a Simulink output signal. Use this block to connect outputs of a Physical Network diagram to Simulink scopes or other Simulink blocks.

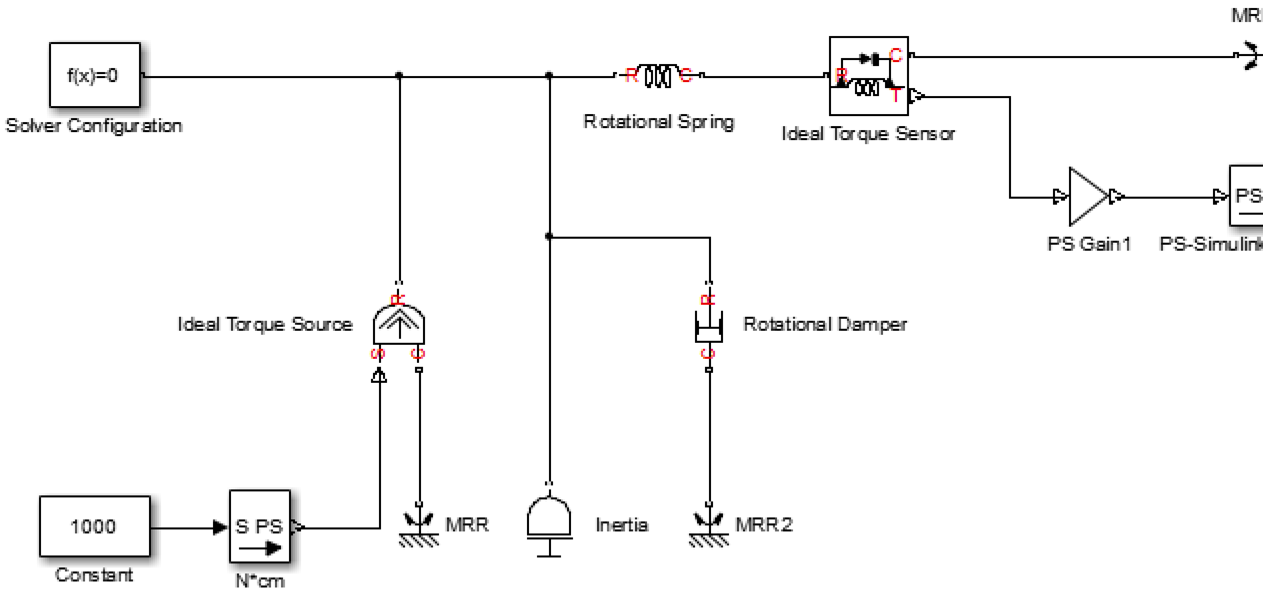
The **Output signal unit** parameter lets you specify the desired units for the output signal. These units must be commensurate with the units of the input physical signal coming into the block. The Simulink output signal is unitless, but if you specify a desired output unit, the block applies a gain equal to the conversion factor before outputting the Simulink signal. For example, if the input physical signal coming into the block is displacement, in meters, and you set **Output signal unit** to `mm`, the block multiplies the value of the input signal by 10e3 before outputting it.

In the diagram below, the input signal for the PS-Simulink Converter block is torque in N*m, and if you do not specify the output signal unit, the Display block shows the value of 10. If you change the **Output signal unit** parameter value in the PS-Simulink Converter block to `N*cm`, the torque value in the Display block changes to 1000, as shown in the diagram.



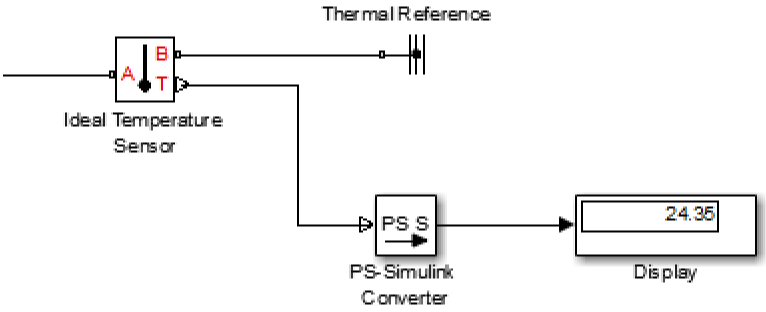
Note Currently, physical units are not propagated through the blocks in the Physical Signals library, such as PS Add, PS Gain, and so on. If your diagram contains a Physical Signals block before a PS-Simulink Converter block, the unit specification in the PS-Simulink Converter block is ignored.

In the following example, the PS-Simulink Converter block is installed after the PS Gain1 block. The display reading will remain the same regardless of the **Output signal unit** parameter setting in the PS-Simulink Converter block.

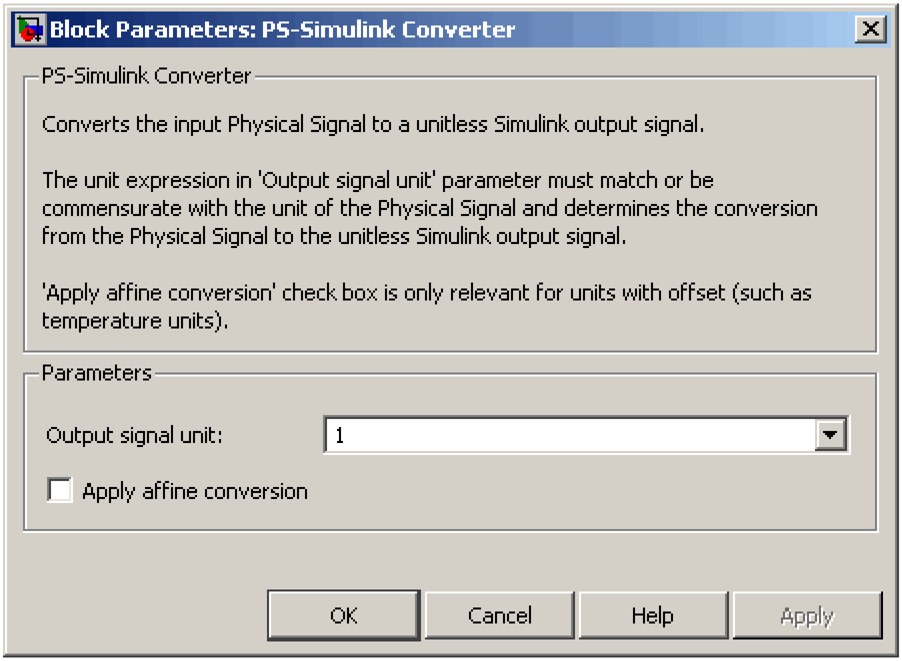


When the output signal is related to thermodynamic variables and contains units of temperature, you must decide whether affine conversion needs to be applied. For more information, see "When to Apply Affine Conversion". Usually, if the output signal represents a relative temperature, that is, a change in temperature, you need to apply linear conversion, $\Delta T_{new} = L * \Delta T_{old}$ (the default method). However, if the output signal represents an absolute temperature, you need to apply affine conversion, $T_{new} = L * T_{old} + O$.

In the following diagram, the Display block shows the room temperature. If you want to display it in degrees Celsius, open the PS-Simulink Converter block, type `C` in the **Output signal unit** field, and select the **Apply affine conversion** check box. The display reading is 24.35. However, if you leave the **Apply affine conversion** check box clear, the Display block would show 297.5.



Dialog Box and Parameters



PS-Simulink Converter

Converts the input Physical Signal to a unitless Simulink output signal.

The unit expression in 'Output signal unit' parameter must match or be commensurate with the unit of the Physical Signal and determines the conversion from the Physical Signal to the unitless Simulink output signal.

'Apply affine conversion' check box is only relevant for units with offset (such as temperature units).

Parameters

Output signal unit: 1

□ Apply affine conversion

Output signal unit

Specify the desired units for the output signal. These units must be commensurate with the units of the input physical signal coming into the block. The system compares the units you specified with the actual units of the input physical signal and applies a gain equal to the conversion factor before outputting the Simulink signal. You can select a unit from the drop-down list, or type the desired unit name, such as `rpm`, or a valid expression, such as `rad/s`. For more information and a list of unit abbreviations, see "How to Specify Units in Block Dialogs" and "Unit Definitions". The default value is `1`, which means that the unit is not specified. If you do not specify a unit, or if the unit matches the actual units of the input physical signal, no gain is applied.

Apply affine conversion

This check box is applicable only for units that can be converted either with or without an affine offset, such as thermal units. For more information, see "Thermal Unit Conversions".

Restricted Parameters

When your model is in Restricted editing mode, you cannot modify any of the block parameters.

Ports

The block has a physical signal input port, located on its left side, and a Simulink output port, located on its right side (in the block default orientation).

See Also

Simulink-PS Converter

Radiative Heat Transfer

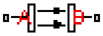
Purpose

Heat transfer by radiation

Library

Thermal Elements

Description



The Radiative Heat Transfer block represents a heat transfer by radiation between two bodies. The transfer is governed by the Stefan-Boltzmann law and is described with the following equation:

$$Q = k \cdot A \cdot (T_A^4 - T_B^4)$$

where

Q	Heat flow
k	Radiation coefficient
A	Emitting body surface area
T_{A}, T_{B}	Temperatures of the bodies

The radiation coefficient is determined by geometrical shapes, dimensions, and surface emissivity. For example, the radiation constant for the heat transfer between two parallel plates is computed as

$$k = \frac{\sigma}{\frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} - 1}$$

where

σ	Stefan-Boltzmann constant
ε_1, ε_2	Surface emissivity for the emitting and receiving plate, respectively

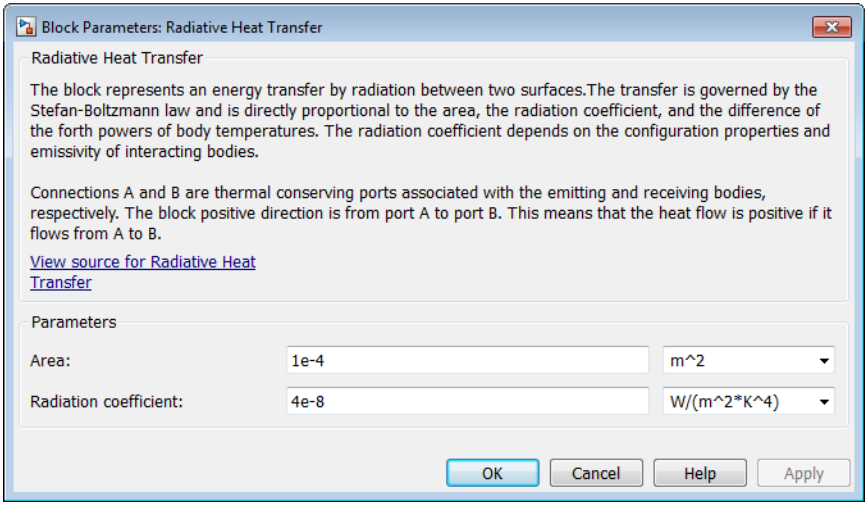
Similarly, the radiation coefficient for concentric cylinders is determined with the formula

$$k = \frac{\sigma}{\frac{1}{\varepsilon_1} + \frac{1-\varepsilon_2}{\varepsilon_2}\frac{r_1}{r_2}}$$

where r_1 and r_2 are the emitting and receiving cylinder radii, respectively. Reference [1] contains formulas for a wide variety of shapes.

Connections A and B are thermal conserving ports associated with the emitting and receiving bodies, respectively. The block positive direction is from port A to port B. This means that the heat flow is positive if it flows from A to B.

Dialog Box and Parameters



Block Parameters: Radiative Heat Transfer

Radiative Heat Transfer

The block represents an energy transfer by radiation between two surfaces.The transfer is governed by the Stefan-Boltzmann law and is directly proportional to the area, the radiation coefficient, and the difference of the forth powers of body temperatures. The radiation coefficient depends on the configuration properties and emissivity of interacting bodies.

Connections A and B are thermal conserving ports associated with the emitting and receiving bodies, respectively. The block positive direction is from port A to port B. This means that the heat flow is positive if it flows from A to B.

View source for Radiative Heat Transfer

Parameters

Area: 1e-4 m^2

Radiation coefficient: 4e-8 W/(m^2*K^4)

Area

Radiating body area of heat transfer. The default value is `0.0001` m^2.

Radiation coefficient

Radiation coefficient of the two bodies, based on their geometrical shapes, dimensions, and surface emissivity. See [1] for more information. The default value is `4e-8` W/m^2/K^4.

Ports

The block has the following ports:

A
: Thermal conserving port associated with body A.

B
: Thermal conserving port associated with body B.

References

[1] Siegel, R. and J.R. Howell. *Thermal Radiation Heat Transfer*. New York: Taylor and Francis, 2002.

See Also

Conductive Heat Transfer

Convective Heat Transfer

Purpose

Generate normally distributed random numbers for physical modeling

Library

Physical Signals/Sources

Description

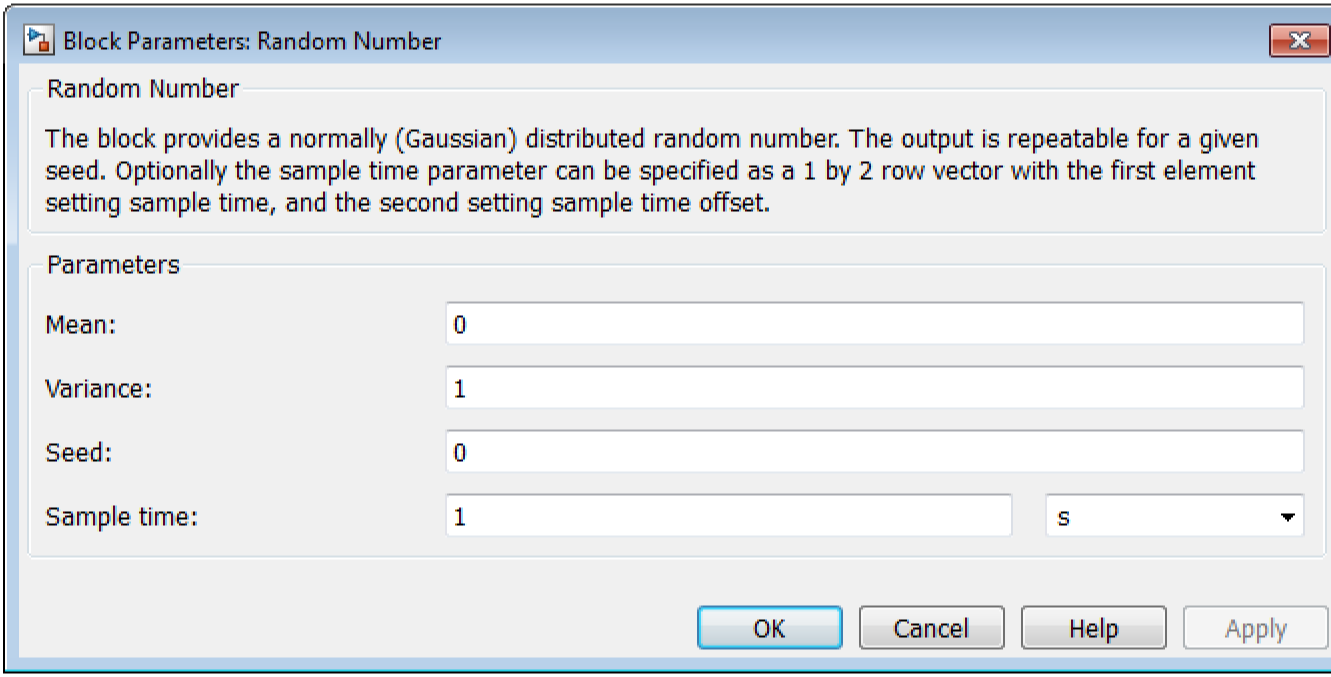
The Random Number block generates normally (Gaussian) distributed random numbers. To generate uniformly distributed random numbers, use the Uniform Random Number block.

The block behavior is the same as the Simulink Random Number block (except that it generates a physical signal rather than a unitless Simulink signal) and is based on the polar rejection method [1, 2].

You have an option to specify an initial time offset as part of the **Sample time** parameter. In this case, the block outputs 0 until the simulation time reaches the *`offset`* value, at which point the random sequence starts.

Random Number blocks that use the same seed and parameters generate a repeatable sequence. The seed resets to the specified value each time a simulation starts. By default, the block produces a sequence that has a mean of 0 and a variance of 1.

Dialog Box and Parameters



Block Parameters: Random Number

Random Number

The block provides a normally (Gaussian) distributed random number. The output is repeatable for a given seed. Optionally the sample time parameter can be specified as a 1 by 2 row vector with the first element setting sample time, and the second setting sample time offset.

Parameters

Mean: 0

Variance: 1

Seed: 0

Sample time: 1 s

Mean

Specify the mean of the random numbers. The default is 0.

Variance

Specify the variance of the random numbers. The default is 1.

Seed

Specify the starting seed for the random number generator. Output is repeatable for a given seed. The seed must be an integer in the range of 0 to (2^32 – 1). The default is 0.

Sample time

The value of the time `step` interval. The default `step` value is `1` s. To specify an initial time offset, enter the parameter value as `[step, offset]`, otherwise the `offset` value is assumed to be `0`. The offset must be less than the step size.

Ports

The block has one physical signal output port.

References

[1] Bell, J. R. “Algorithm 334: Normal random deviates.” *Communications of the ACM*. Vol. 11, Number 7, 1968, p. 498.

[2] Knop, R. “Remark on Algorithm 334 [G5]: normal random deviates.” *Communications of the ACM*. Vol. 12, Number 5, 1969, p. 281.

See Also

Uniform Random Number

Reluctance

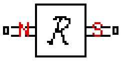
Purpose

Magnetic reluctance

Library

Magnetic Elements

Description



The Reluctance block models a magnetic reluctance, that is, a component that resists flux flow. The ratio of the magnetomotive force (mmf) across the component to the resulting flux that flows through the component is constant, and the ratio value is defined as the reluctance. Reluctance depends on the geometry of the section being modeled.

The block is based on the following equations:

$$\mathtt{F} = \Phi \cdot \Re$$

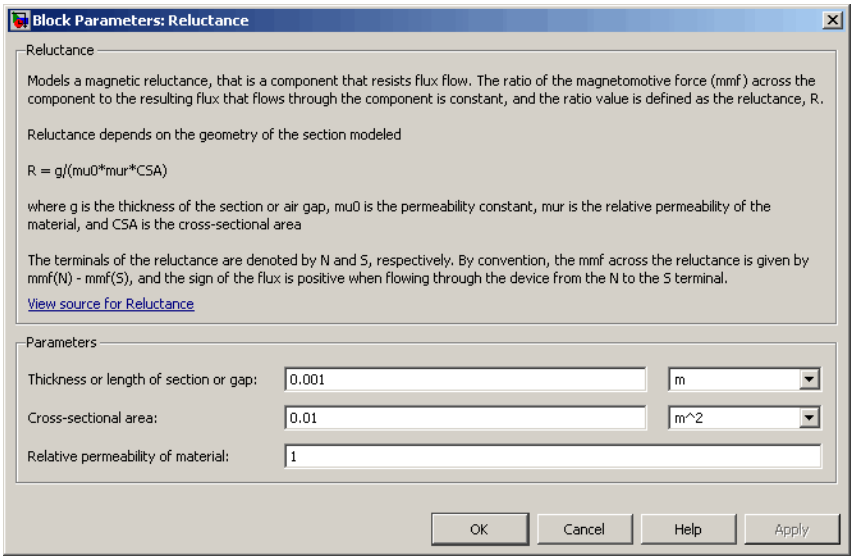
$$\Re = \frac{g}{\mu_0 \cdot \mu_r \cdot A}$$

where

$\mathtt{F}$	Magnetomotive force (mmf) across the component
Φ	Flux through the component
$\Re$	Reluctance
g	Thickness of the section being modeled, or length of air gap
μ_0	Permeability constant
μ_r	Relative permeability of the material
A	Cross-sectional area of the section being modeled

Connections N and S are magnetic conserving ports. The mmf across the reluctance is given by $\mathtt{F}(N) - \mathtt{F}(S)$, and the sign of the flux is positive when flowing through the device from N to S.

Dialog Box and Parameters



Block Parameters: Reluctance

Reluctance

Models a magnetic reluctance, that is a component that resists flux flow. The ratio of the magnetomotive force (mmf) across the component to the resulting flux that flows through the component is constant, and the ratio value is defined as the reluctance, R.

Reluctance depends on the geometry of the section modeled

R = g/(mu0*mur*CSA)

where g is the thickness of the section or air gap, mu0 is the permeability constant, mur is the relative permeability of the material, and CSA is the cross-sectional area

The terminals of the reluctance are denoted by N and S, respectively. By convention, the mmf across the reluctance is given by mmf(N) - mmf(S), and the sign of the flux is positive when flowing through the device from the N to the S terminal.

View source for Reluctance

Parameters

Thickness or length of section or gap: 0.001 m

Cross-sectional area: 0.01 m^2

Relative permeability of material: 1

Thickness or length of section or gap

Thickness of the section being modeled, or length of air gap. The default value is 0.001 m.

Cross-sectional area

Area of the section being modeled. The default value is 0.01 m^2.

Relative permeability of material

Relative permeability of the section material. The default value is 1.

Ports

The block has the following ports:

N

Magnetic conserving port associated with the block North terminal.

S

Magnetic conserving port associated with the block South terminal.

See Also

Variable Reluctance

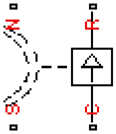
Purpose

Magnetomotive device based on reluctance force

Library

Magnetic Elements

Description



The Reluctance Force Actuator block models a generic magnetomotive device based on reluctance force.

The block is based on the following equations:

$$F = -0.5 \cdot \Phi^2 \cdot \frac{d\Re}{dx}$$

$$\Re(x) = \frac{x}{\mu_0 \cdot \mu_r \cdot A}$$

$$u = dx$$

where

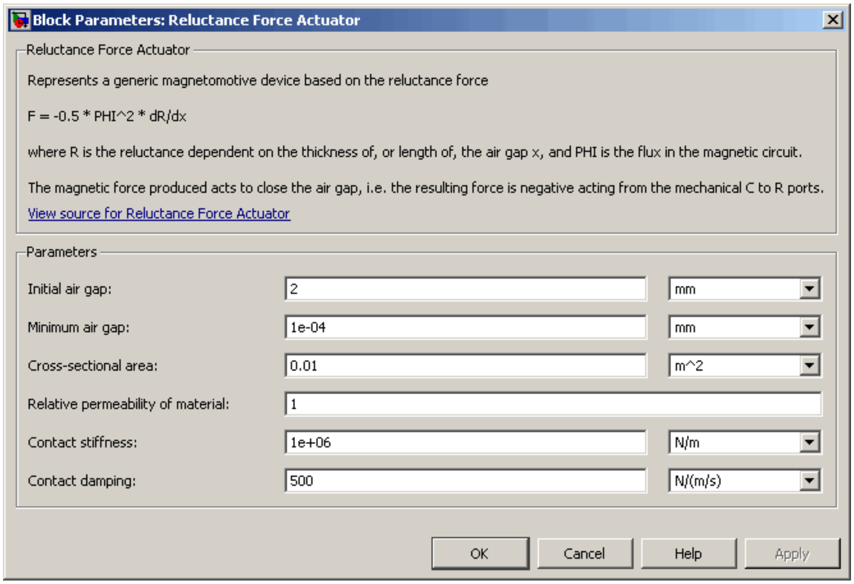
F	Reluctance force
Φ	Flux in the magnetic circuit
$\Re$	Reluctance
x	Thickness or length of the air gap
μ_0	Permeability constant
μ_r	Relative permeability of the material
A	Cross-sectional area of the section being modeled
u	Velocity

Connections N and S are magnetic conserving ports, and connections C and R are mechanical translational conserving ports. The magnetic force produced by the actuator acts to close the gap, therefore the resulting force is negative when it acts from C to R.

Basic Assumptions and Limitations

- The current excitation in the system is constant.
- Only axial reluctance is modeled.

Dialog Box and Parameters



Block Parameters: Reluctance Force Actuator

Reluctance Force Actuator

Represents a generic magnetomotive device based on the reluctance force

F = -0.5 * PHI^2 * dR/dx

where R is the reluctance dependent on the thickness of, or length of, the air gap x, and PHI is the flux in the magnetic circuit.

The magnetic force produced acts to close the air gap, i.e. the resulting force is negative acting from the mechanical C to R ports.

View source for Reluctance Force Actuator

Parameters

Initial air gap:	2	mm
Minimum air gap:	1e-04	mm
Cross-sectional area:	0.01	m^2
Relative permeability of material:	1	
Contact stiffness:	1e+06	N/m
Contact damping:	500	N/(m/s)

Initial air gap

Thickness or length of air gap at the beginning of simulation. The default value is `2` mm.

Minimum air gap

Minimal value of air gap, with the reluctance force acting to close the air gap. The parameter value has to be greater than 0. The default value is `1e-4` mm.

Cross-sectional area

Area of the section being modeled. The default value is `0.01` m^2.

Relative permeability of material

Relative permeability of the section material. The default value is `1`.

Contact stiffness

Stiffness that models the hard stop at the minimum air gap position. The default value is `10e6` N/m.

Contact damping

Damping that models the hard stop at the minimum air gap position. The default value is 500 N/(m/s).

Ports

The block has the following ports:

`N`

Magnetic conserving port associated with the block North terminal.

`S`

Magnetic conserving port associated with the block South terminal.

`R`

Mechanical translational conserving port associated with the rod.

`C`

Mechanical translational conserving port associated with the case.

See Also

Reluctance

Variable Reluctance

Repeating Sequence

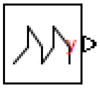
Purpose

Output periodic piecewise linear signal

Library

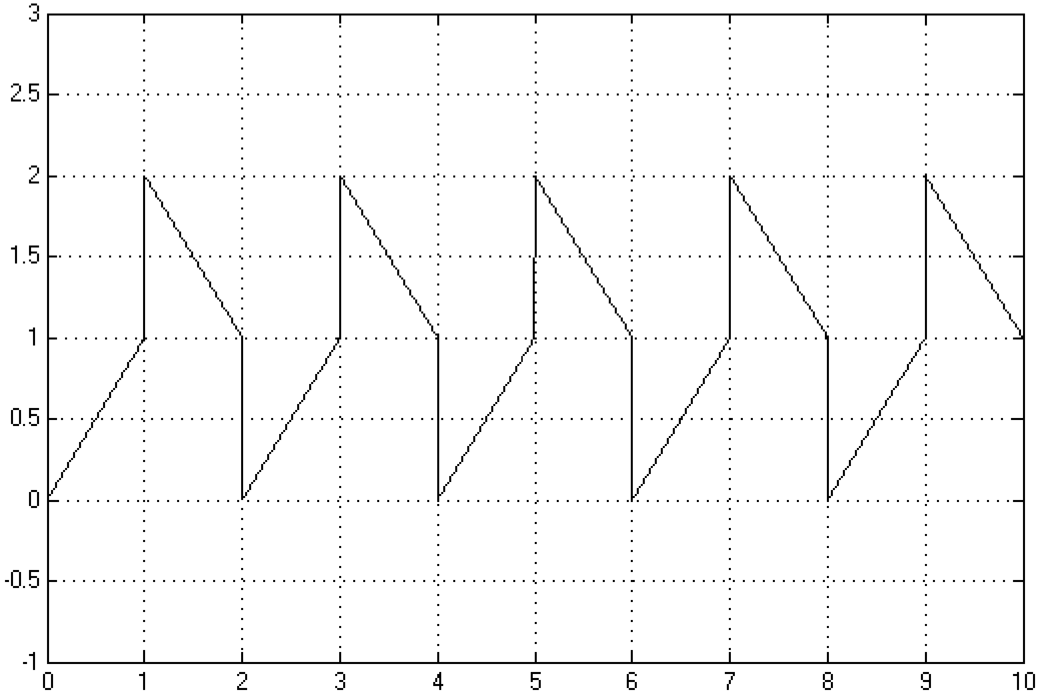
Physical Signals/Sources

Description

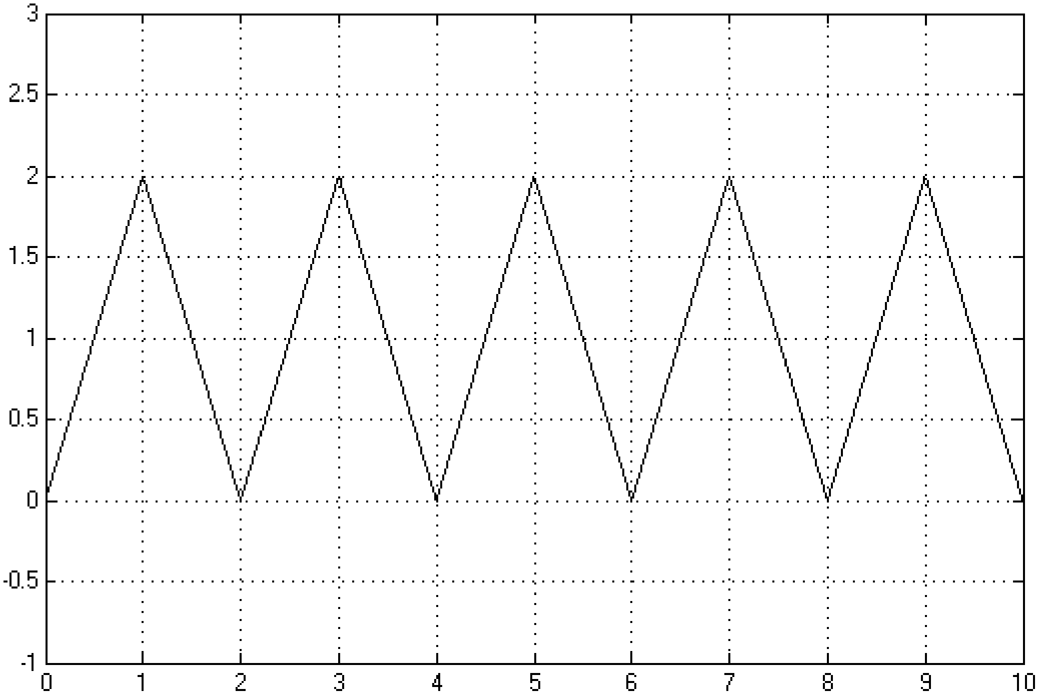


The Repeating Sequence block outputs a periodic piecewise linear signal, `y`. You can optionally specify an initial signal value and an initial time offset. The repeating sequence consists of a number of linear segments, connected to each other. The number of segments must be no greater than 20. You specify how to connect the segments by choosing a signal type. For the same set of block parameter values, the resulting output signal will be different depending on the signal type:

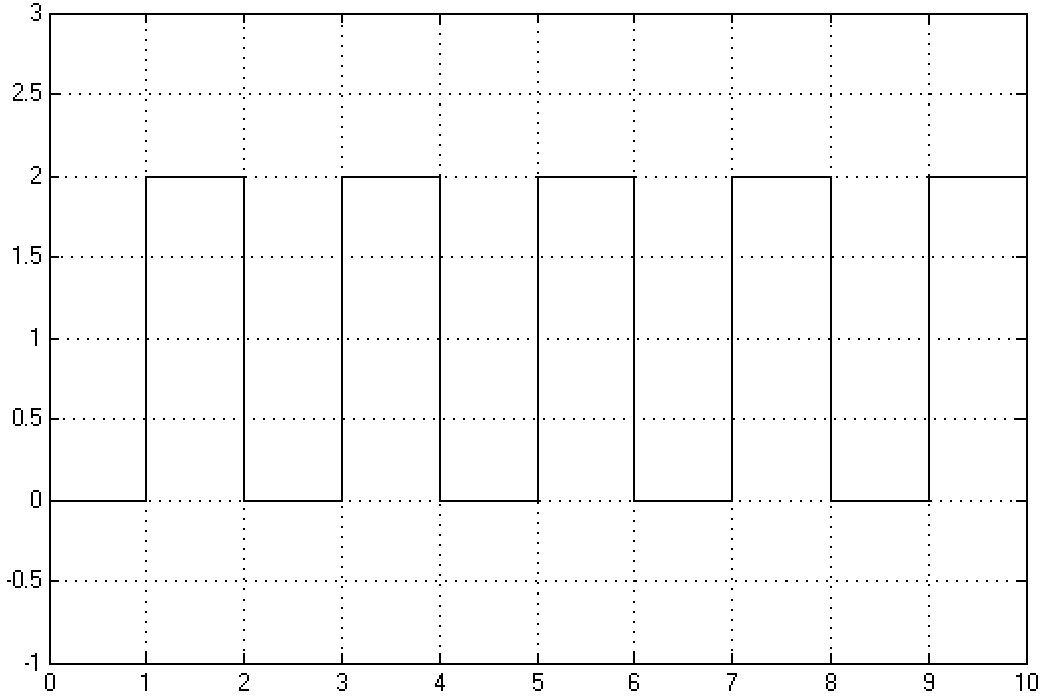
- `Discontinuous` — Each linear segment in the repeating sequence is defined by its duration, start value, and end value. If the end value of a segment is not the same as the start value of the next segment, they are connected by a vertical line.



- `Continuous` — Each linear segment in the repeating sequence is defined by its duration and start value. The end value of a segment is the same as the start value of the next segment.

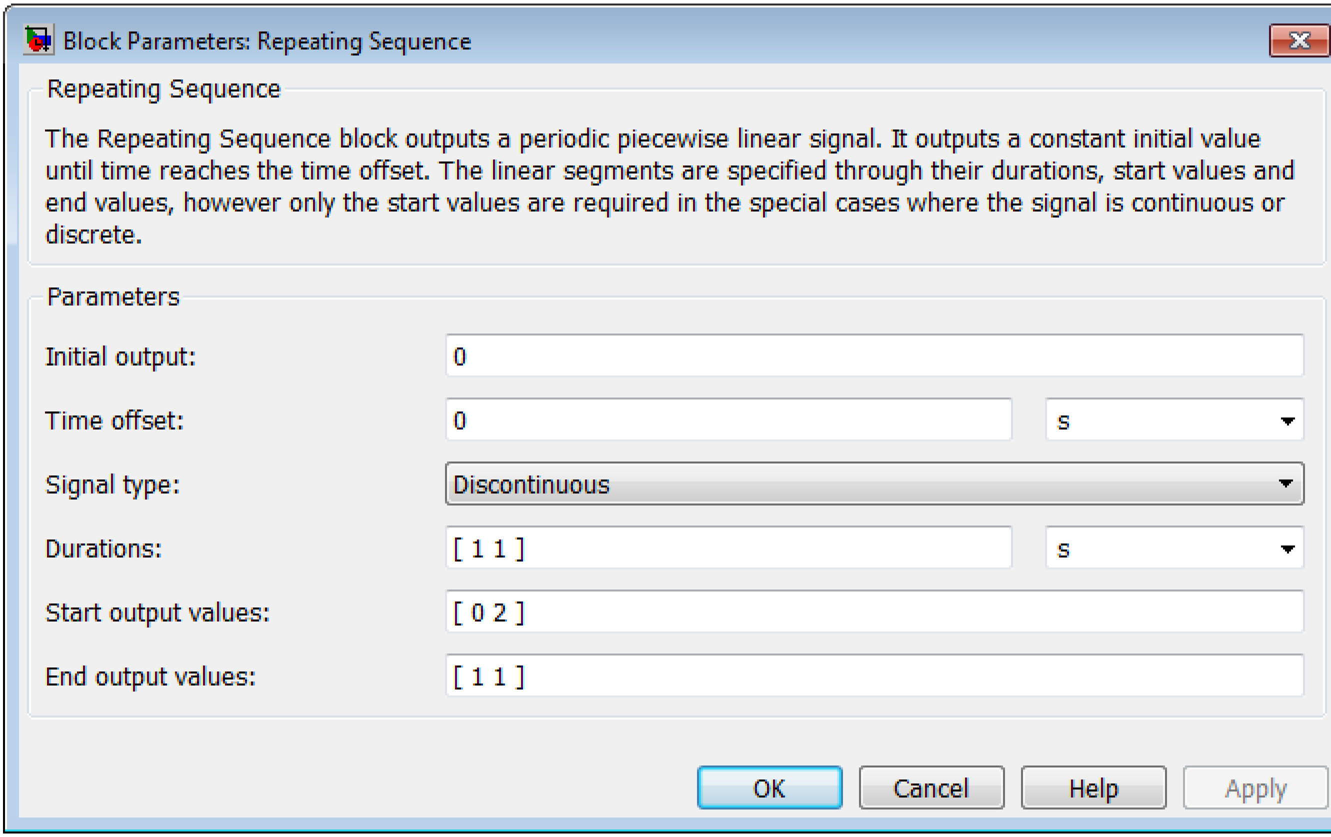


- `Discrete` — Each linear segment in the repeating sequence is defined by its duration and start value. The end value of a segment is the same as its start value.



Use this block to generate various types of physical signals, such as pulse, sawtooth, stair, and so on.

Dialog Box and Parameters



Block Parameters: Repeating Sequence

Repeating Sequence

The Repeating Sequence block outputs a periodic piecewise linear signal. It outputs a constant initial value until time reaches the time offset. The linear segments are specified through their durations, start values and end values, however only the start values are required in the special cases where the signal is continuous or discrete.

Parameters

Initial output: 0

Time offset: 0 s

Signal type: Discontinuous

Durations: [1 1] s

Start output values: [0 2]

End output values: [1 1]

Initial output

The value of the output signal at time zero. The output of the block remains at this value until the simulation time reaches the **Time offset** value. The default value is `0`.

Time offset

The value of the initial time offset, before the start of the repeating sequence. During this time, the output of the block remains at the **Initial output** value. The default value is `0`.

Signal type

Select one of the following signal types:

- `Discontinuous` — For each linear segment in the repeating sequence, define its duration, start value, and end value. If the end value of a segment is not the same as the start value of the next segment, they are connected by a vertical line. This is the default method.
- `Continuous` — For each linear segment in the repeating sequence, define its duration and start value. The end value of a segment is the same as the start value of the next segment.
- `Discrete` — For each linear segment in the repeating sequence, define its duration and start value. The end value of a segment is the same as its start value.

Durations

Specify the linear segment durations as a 1–by–*n* row vector, where *n* is the number of linear segments in the repeating sequence. *n* must be no greater than 20. The default is `[ 1 1 ]`s, which means two linear segments, each lasting 1 second.

Start output values

Specify the start values of the output signal for each linear segment as a 1–by–*n* row vector, where *n* is the number of linear segments in the repeating sequence. The size of the vector must be no greater than 20 and must match the size of the **Durations** row vector. The default is `[ 0 2 ]`, which means that the first of the two linear segments starts at 0, and the second one starts at 2.

End output values

Specify the end values of the output signal for each linear segment as a 1–by–*n* row vector, where *n* is the number of linear segments in the repeating sequence. The size of the vector must be no greater than 20 and must match the size of the **Durations** row vector. The default is `[ 1 1 ]`. This parameter is only visible if the **Signal type** parameter is set to `Discontinuous`. For other signal types, the end value of a segment is defined either by the start value of the next segment (`Continuous`) or the start value of the same segment (`Discrete`).

Ports

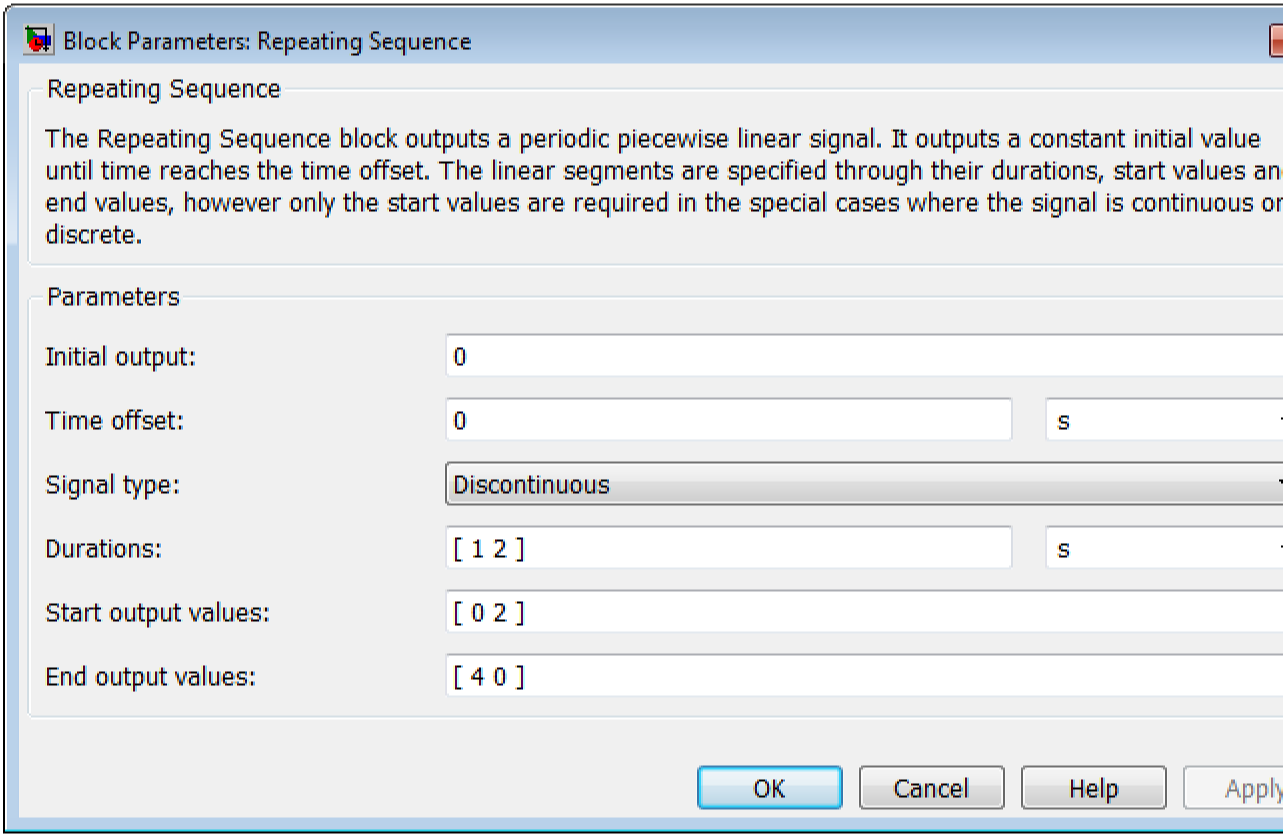
The block has one physical signal output port.

Examples

Discontinuous Repeating Sequence

This example shows the mapping between the block parameter values and the resulting output signal.

Set the block parameters as shown:



Block Parameters: Repeating Sequence

Repeating Sequence

The Repeating Sequence block outputs a periodic piecewise linear signal. It outputs a constant initial value until time reaches the time offset. The linear segments are specified through their durations, start values an end values, however only the start values are required in the special cases where the signal is continuous or discrete.

Parameters

Initial output: 0

Time offset: 0 s

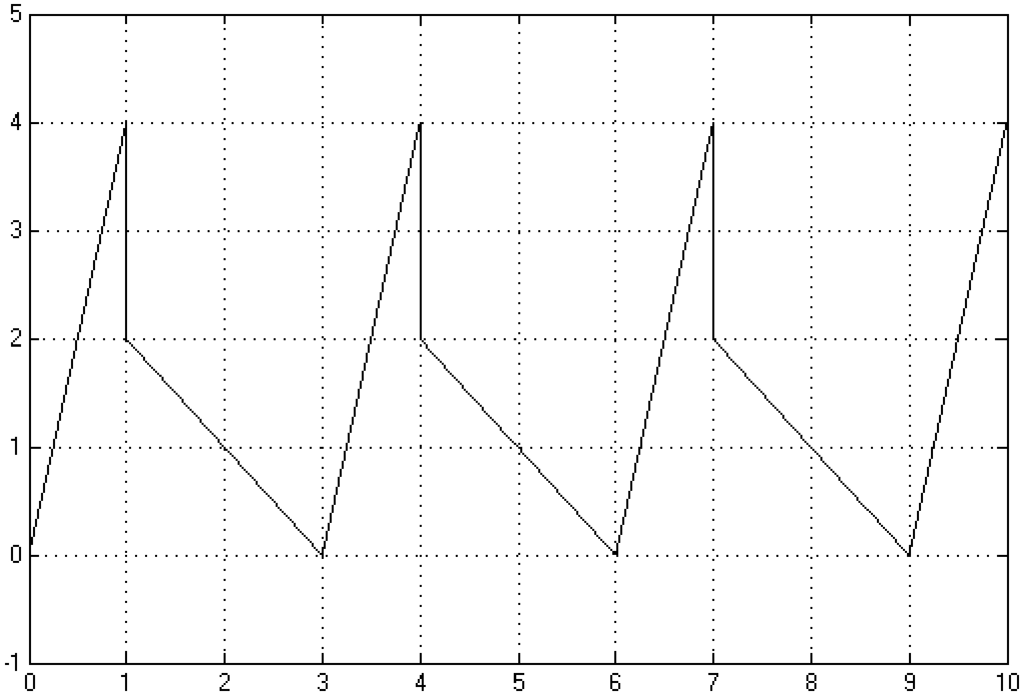
Signal type: Discontinuous

Durations: [1 2] s

Start output values: [0 2]

End output values: [4 0]

The following plot shows the resulting block output.



The signal starts at 0 and consists of two linear segments. The duration of the first segment is 1 second, the segment starts at 0 and ends at 4. The signal is discontinuous, and the end value of the first segment is different than the start value of the second segment, therefore they are connected by a vertical line. The second segment starts at 2, lasts for 2 seconds, and ends at 0, after which the sequence repeats.

See Also

Counter

Resistor

Purpose

Linear resistor in electrical systems

Library

Electrical Elements

Description

The Resistor block models a linear resistor, described with the following equation:

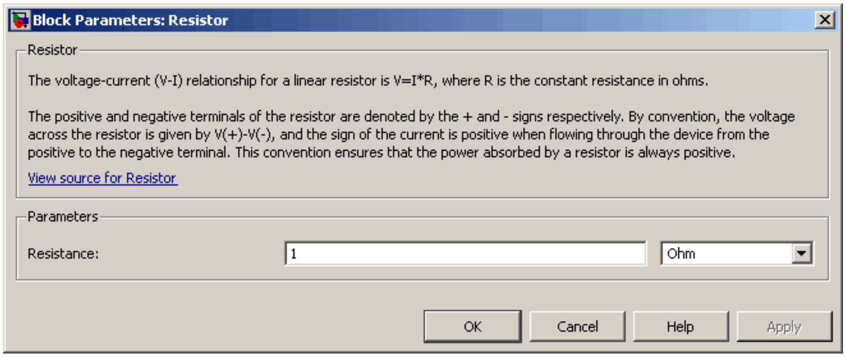
$$V = I\square R$$

where

V	Voltage
I	Current
R	Resistance

Connections + and – are conserving electrical ports corresponding to the positive and negative terminals of the resistor, respectively. By convention, the voltage across the resistor is given by $V(+) – V(–)$, and the sign of the current is positive when flowing through the device from the positive to the negative terminal. This convention ensures that the power absorbed by a resistor is always positive.

Dialog Box and Parameters



Block Parameters: Resistor

Resistor

The voltage-current (V-I) relationship for a linear resistor is V=I*R, where R is the constant resistance in ohms.

The positive and negative terminals of the resistor are denoted by the + and - signs respectively. By convention, the voltage across the resistor is given by V(+)-V(-), and the sign of the current is positive when flowing through the device from the positive to the negative terminal. This convention ensures that the power absorbed by a resistor is always positive.

View source for Resistor

Parameters

Resistance: 1 Ohm

Resistance

Resistance, in ohms. The default value is 1 Ω.

Ports

The block has the following ports:

+
: Electrical conserving port associated with the resistor positive terminal.

-
: Electrical conserving port associated with the resistor negative terminal.

See Also

Variable Resistor

Rotary Pneumatic Piston Chamber

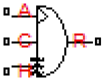
Purpose

Rotational pneumatic piston chamber based on ideal gas law

Library

Pneumatic Elements

Description



The Rotary Pneumatic Piston Chamber block models a pneumatic rotary piston chamber based on the ideal gas law and assuming constant specific heats. Use this model as a building block for pneumatic rotational actuators. The piston can generate torque in one direction only, and the direction is set by the **Chamber orientation** parameter.

The continuity equation for the network representation of the piston chamber is

$$G = \frac{V_0 + D \cdot \theta}{RT}\left(\frac{dp}{dt} - \frac{p}{T}\frac{dT}{dt}\right) + \frac{D}{RT} \cdot p \cdot \frac{d\theta}{dt}$$

where

G	Mass flow rate at input port
V_0	Initial chamber volume
D	Piston displacement (volume per unit angle)
Θ	Piston angle
p	Absolute pressure in the chamber
R	Specific gas constant
T	Absolute gas temperature
t	Time

The energy equation is

$$q = \frac{c_v}{R}(V_0 + D \cdot \theta)\frac{dp}{dt} + \frac{c_p \cdot D}{R} p \frac{d\theta}{dt} - q_w$$

where

q Heat flow due to gas inflow in the chamber (through the pneumatic port)

q_w Heat flow through the chamber walls (through the thermal port)

c_v Specific heat at constant volume

c_p Specific heat at constant pressure

The torque equation is

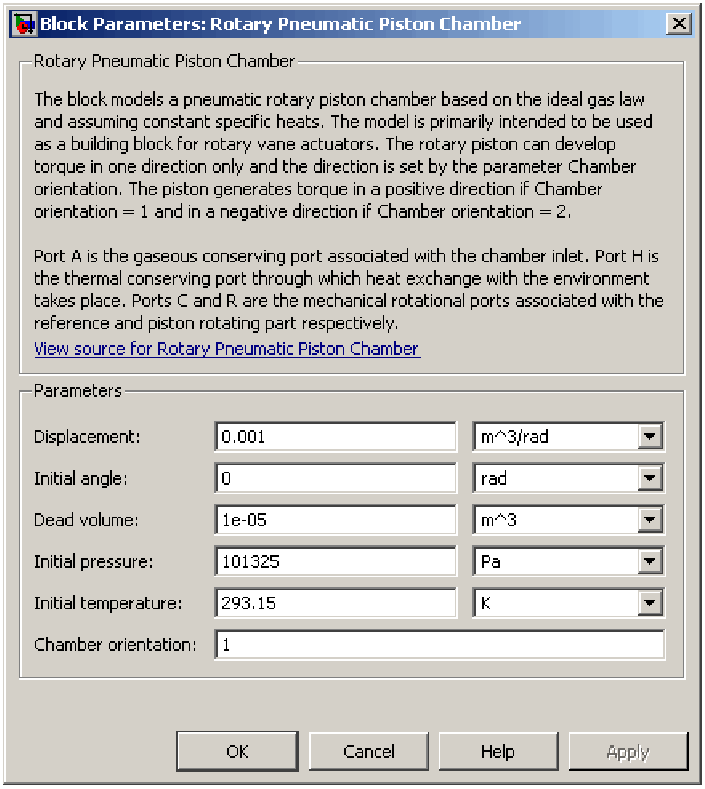
$$\tau = p \square D$$

Port A is the pneumatic conserving port associated with the chamber inlet. Port H is a thermal conserving port through which heat exchange with the environment takes place. Ports C and R are mechanical rotational conserving ports associated with the piston case and rod, respectively. The gas flow and the heat flow are considered positive if they flow into the chamber.

Basic Assumptions and Limitations

- The gas is ideal.
- Specific heats at constant pressure and constant volume, c_p and c_v, are constant.

Dialog Box and Parameters



Rotary Pneumatic Piston Chamber

The block models a pneumatic rotary piston chamber based on the ideal gas law and assuming constant specific heats. The model is primarily intended to be used as a building block for rotary vane actuators. The rotary piston can develop torque in one direction only and the direction is set by the parameter Chamber orientation. The piston generates torque in a positive direction if Chamber orientation = 1 and in a negative direction if Chamber orientation = 2.

Port A is the gaseous conserving port associated with the chamber inlet. Port H is the thermal conserving port through which heat exchange with the environment takes place. Ports C and R are the mechanical rotational ports associated with the reference and piston rotating part respectively.

View source for Rotary Pneumatic Piston Chamber

Parameters

Displacement: 0.001 m^3/rad

Initial angle: 0 rad

Dead volume: 1e-05 m^3

Initial pressure: 101325 Pa

Initial temperature: 293.15 K

Chamber orientation: 1

Displacement

Specify the effective piston displacement, as volume per unit angle. The default value is `.001` m^3/rad.

Initial angle

Specify the initial piston angle. The default value is `0`.

Dead volume

Specify the volume of gas in the chamber at zero piston position. The default value is `1e-5` m^3.

Initial pressure

Specify the initial pressure in the chamber. This parameter specifies the initial condition for use in computing the initial state at the beginning of a simulation run. For more information, see "Initial Conditions Computation". The default value is `101235` Pa.

Initial temperature

Specify the initial temperature of the gas in the chamber. This parameter specifies the initial condition for use in computing the initial state at the beginning of a simulation run. For more information, see "Initial Conditions Computation". The default value is `293.15` K.

Chamber orientation

Specify the direction of torque generation. The piston generates torque in a positive direction if this parameter is set to `1` (the default). If you set this parameter to `2`, the piston generates torque in a negative direction.

Ports

The block has the following ports:

`A`

Pneumatic conserving port associated with the chamber inlet.

`H`

Thermal conserving port through which heat exchange with the environment takes place.

`R`

Mechanical rotational conserving port associated with the piston (rod).

`C`

Mechanical rotational conserving port associated with the reference (case).

See Also

Constant Volume Pneumatic Chamber

Pneumatic Piston Chamber

Purpose

Viscous damper in mechanical rotational systems

Library

Mechanical Rotational Elements

Description

The Rotational Damper block represents an ideal mechanical rotational viscous damper described with the following equations:

$$T = D\square\omega$$

$$\omega = \omega_R - \omega_C$$

where

T Torque transmitted through the damper

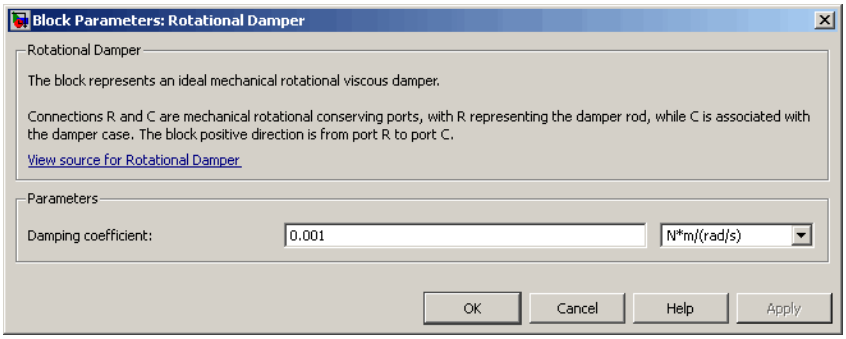
D Damping (viscous friction) coefficient

ω Relative angular velocity

ω_R,ω_C Absolute angular velocities of terminals R and C, respectively

The block positive direction is from port R to port C. This means that the torque is positive if it acts in the direction from R to C.

Dialog Box and Parameters



Block Parameters: Rotational Damper

Rotational Damper

The block represents an ideal mechanical rotational viscous damper.

Connections R and C are mechanical rotational conserving ports, with R representing the damper rod, while C is associated with the damper case. The block positive direction is from port R to port C.

View source for Rotational Damper

Parameters

Damping coefficient: 0.001 N*m/(rad/s)

Damping coefficient

Damping coefficient, defined by viscous friction. The default value is `0.001` N*m/(rad/s).

Ports

The block has the following ports:

`R`

Mechanical rotational conserving port.

`C`

Mechanical rotational conserving port.

See Also

Rotational Friction

Rotational Hard Stop

Rotational Spring

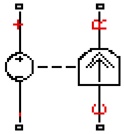
Purpose

Interface between electrical and mechanical rotational domains

Library

Electrical Elements

Description



The Rotational Electromechanical Converter block provides an interface between the electrical and mechanical rotational domains. It converts electrical energy into mechanical energy in the form of rotational motion, and vice versa. The converter is described with the following equations:

$$T = K \cdot I$$

$$V = K \cdot \omega$$

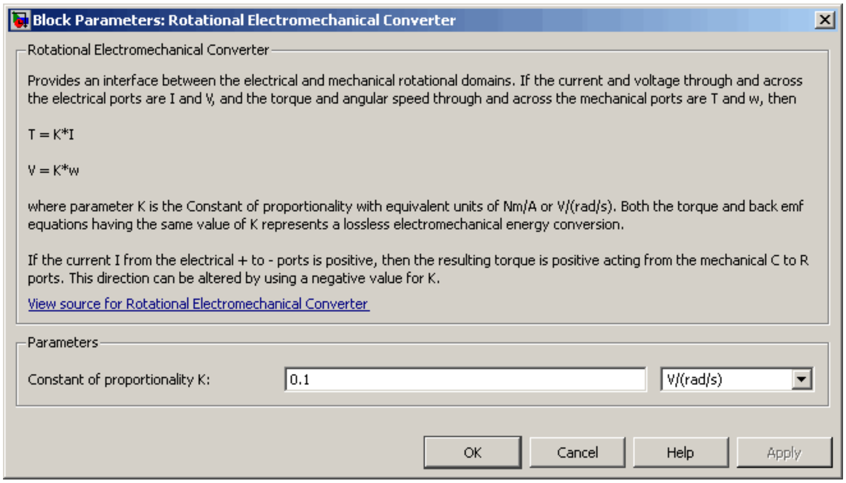
where

V	Voltage across the electrical ports of the converter
I	Current through the electrical ports of the converter
T	Torque
ω	Angular speed
K	Constant of proportionality

The Rotational Electromechanical Converter block represents a lossless electromechanical energy conversion, therefore the same constant of proportionality is used in both equations.

Connections + and – are conserving electrical ports corresponding to the positive and negative terminals of the converter, respectively. Connections C and R are conserving mechanical rotational ports. If the current flowing from the positive to the negative terminal is positive, then the resulting torque is positive acting from port C to port R. This direction can be altered by using a negative value for K.

Dialog Box and Parameters



Rotational Electromechanical Converter

Provides an interface between the electrical and mechanical rotational domains. If the current and voltage through and across the electrical ports are I and V, and the torque and angular speed through and across the mechanical ports are T and w, then

T = K*I

V = K*w

where parameter K is the Constant of proportionality with equivalent units of Nm/A or V/(rad/s). Both the torque and back emf equations having the same value of K represents a lossless electromechanical energy conversion.

If the current I from the electrical + to - ports is positive, then the resulting torque is positive acting from the mechanical C to R ports. This direction can be altered by using a negative value for K.

View source for Rotational Electromechanical Converter

Parameters

Constant of proportionality K: 0.1 V/(rad/s)

Constant of proportionality K

Constant of proportionality for electromechanical conversions. The default value is `0.1` V/(rad/s).

Ports

The block has the following ports:

+

Electrical conserving port associated with the converter positive terminal.

-

Electrical conserving port associated with the converter negative terminal.

C

Mechanical rotational conserving port.

R

Mechanical rotational conserving port.

See Also

Translational Electromechanical Converter

Rotational Free End

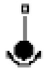
Purpose

Rotational port terminator with zero torque

Library

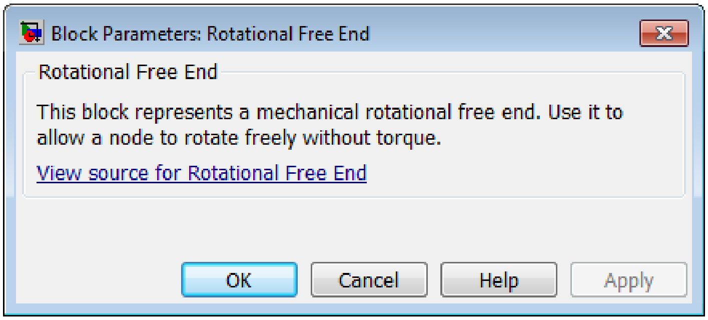
Mechanical Rotational Elements

Description



The Rotational Free End block represents a mechanical rotational port that rotates freely, without torque. Physical Network block diagrams do not allow unconnected Conserving ports. Use this block to terminate mechanical rotational ports (on other blocks) that you wish to leave unconnected.

Dialog Box and Parameters



Block Parameters: Rotational Free End

Rotational Free End

This block represents a mechanical rotational free end. Use it to allow a node to rotate freely without torque.

View source for Rotational Free End

The Rotational Free End block has no parameters.

Ports

The block has one mechanical rotational conserving port.

See Also

Hydraulic Cap

Open Circuit

Translational Free End

Rotational Friction

Purpose

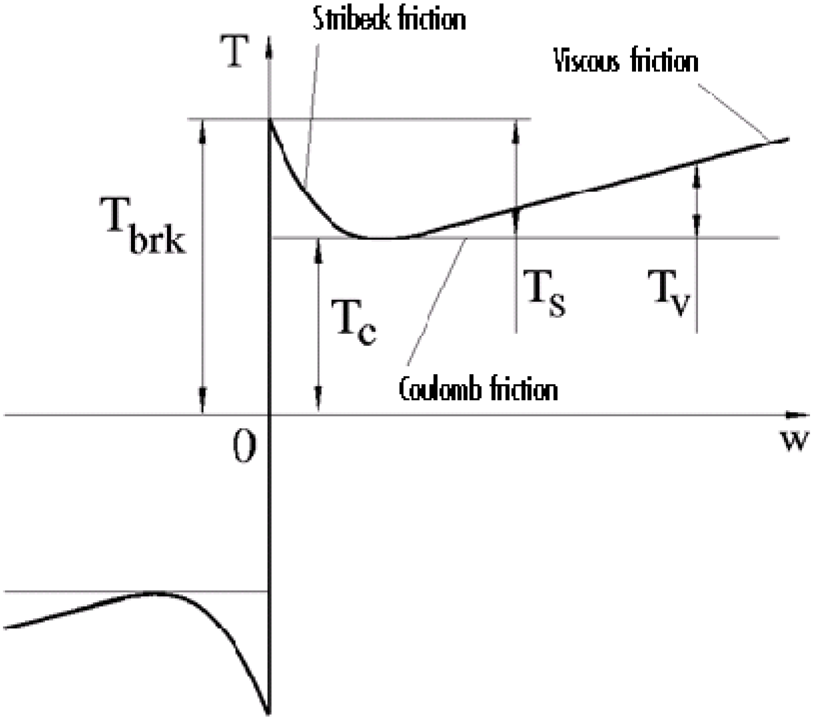
Friction in contact between rotating bodies

Library

Mechanical Rotational Elements

Description

The Rotational Friction block represents friction in contact between rotating bodies. The friction torque is simulated as a function of relative velocity and is assumed to be the sum of Stribeck, Coulomb, and viscous components, as shown in the following figure.



The Stribeck friction, T_S, is the negatively sloped characteristics taking place at low velocities (see [1]). The Coulomb friction, T_C, results in a constant torque at any velocity. The viscous friction, T_V, opposes motion with the torque directly proportional to the relative velocity. The sum of the Coulomb and Stribeck frictions at the vicinity of zero velocity is often referred to as the breakaway friction, T_{brk}. The friction is approximated with the following equations:

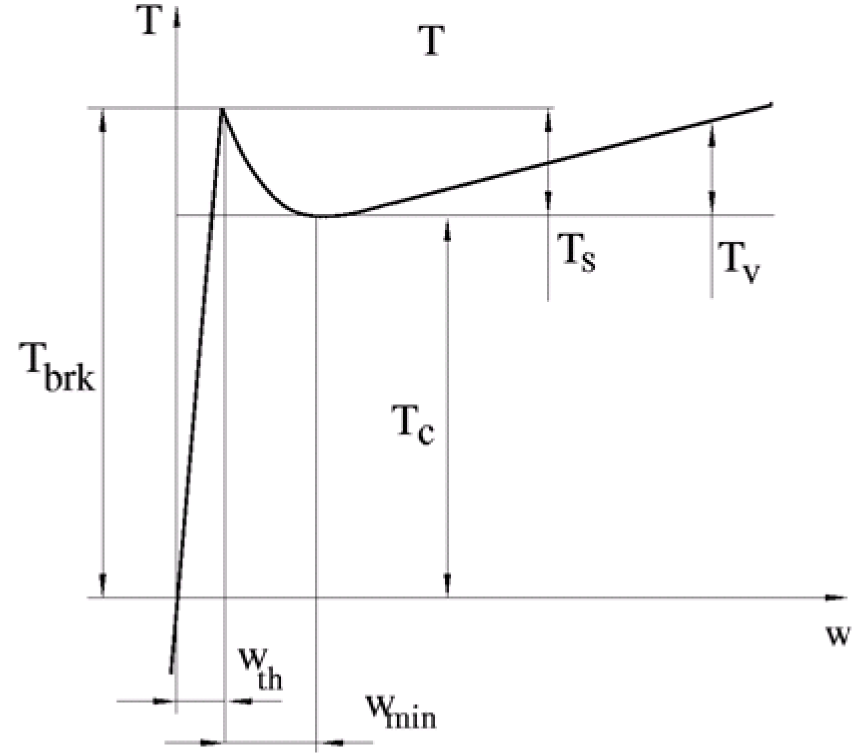
$$T = \left(T_C + (T_{brk} - T_C) \cdot \exp(-c_v\,|\,\omega\,|)\right) sign(\omega) + f\omega$$

$$\omega = \omega_R - \omega_C$$

where

T	Friction torque
T_C	Coulomb friction torque
T_{brk}	Breakaway friction torque
c_v	Coefficient
ω	Relative velocity
ω_R,ω_C	Absolute angular velocities of terminals R and C, respectively
f	Viscous friction coefficient

The approximation above is too idealistic and has a substantial drawback. The characteristic is discontinuous at $\omega = 0$, which creates considerable computational problems. It has been proven that the discontinuous friction model is a nonphysical simplification in the sense that the mechanical contact with distributed mass and compliance cannot exhibit an instantaneous change in torque (see [1]). There are numerous models of friction without discontinuity. The Rotational Friction block implements one of the simplest versions of continuous friction models. The friction torque-relative velocity characteristic of this approximation is shown in the following figure.



The discontinuity is eliminated by introducing a very small, but finite, region in the zero velocity vicinity, within which friction torque is assumed to be linearly proportional to velocity, with the proportionality coefficient T_{brk}/ω_{th}, where ω_{th} is the velocity threshold. It has been proven experimentally that the velocity threshold in the range between 10^{-3} and 10^{-5} rad/s is a good compromise between the accuracy and computational robustness and effectiveness. Notice that friction torque computed with this approximation does not actually stop relative motion when an acting torque drops below breakaway friction level. The bodies will creep relative to each other at a very small velocity proportional to acting torque.

As a result of introducing the velocity threshold, the block equations are slightly modified:

- If $|\omega| >= \omega_{th}$,

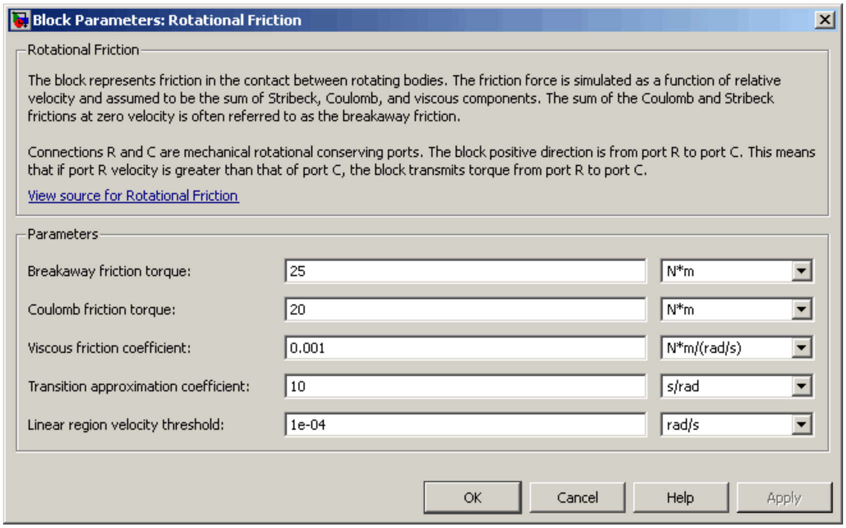
$$T = \left(T_C + (T_{brk} - T_C)\cdot\exp(-c_v\,|\omega|)\right) sign(\omega) + f\omega$$

- If $|\omega| < \omega_{th}$,

$$T = \omega\frac{\left(f\omega_{th} + \left(T_C + (T_{brk} - T_C)\cdot\exp(-c_v\omega_{th})\right)\right)}{\omega_{th}}$$

The block positive direction is from port R to port C. This means that if the port R velocity is greater than that of port C, the block transmits torque from R to C.

Dialog Box and Parameters



Block Parameters: Rotational Friction

Rotational Friction

The block represents friction in the contact between rotating bodies. The friction force is simulated as a function of relative velocity and assumed to be the sum of Stribeck, Coulomb, and viscous components. The sum of the Coulomb and Stribeck frictions at zero velocity is often referred to as the breakaway friction.

Connections R and C are mechanical rotational conserving ports. The block positive direction is from port R to port C. This means that if port R velocity is greater than that of port C, the block transmits torque from port R to port C.

View source for Rotational Friction

Parameters

Breakaway friction torque:	25	N*m
Coulomb friction torque:	20	N*m
Viscous friction coefficient:	0.001	N*m/(rad/s)
Transition approximation coefficient:	10	s/rad
Linear region velocity threshold:	1e-04	rad/s

Breakaway friction torque

Breakaway friction torque, which is the sum of the Coulomb and the static frictions. It must be greater than or equal to the Coulomb friction torque value. The default value is `25` N*m.

Coulomb friction torque

Coulomb friction torque, which is the friction that opposes rotation with a constant torque at any velocity. The default value is `20` N*m.

Viscous friction coefficient

Proportionality coefficient between the friction torque and the relative angular velocity. The parameter value must be greater than or equal to zero. The default value is `0.001` N*m/(rad/s).

Transition approximation coefficient

The parameter sets the value of coefficient c_v, which is used for the approximation of the transition between the static and the Coulomb frictions. Its value is assigned based on the following considerations: the static friction component reaches approximately 95% of its steady-state value at velocity $3/c_v$, and 98% at velocity $4/c_v$, which makes it possible to develop an approximate relationship c_v ~= $4/\omega_{min,}$ where ω_{min} is the relative velocity at which friction torque has its minimum value. By default, c_v is set to `10` rad/s, which corresponds to a minimum friction at velocity of about `0.4` s/rad.

Linear region velocity threshold

The parameter sets the small vicinity near zero velocity, within which friction torque is considered to be linearly proportional to the relative velocity. MathWorks recommends that you use values in the range between `1e-5` and `1e-3` rad/s. The default value is `1e-4` rad/s.

Ports

The block has the following ports:

`R`

Mechanical rotational conserving port.

`C`

Mechanical rotational conserving port.

Examples

The Mechanical Rotational System with Stick-Slip Motion example illustrates the use of the Rotational Friction block in mechanical systems. The friction element is installed between the load and the velocity source, and there is a difference between the breakaway and the Coulomb frictions. As a result, stick-slip motion is developed in the regions of constant velocities.

References

[1] B. Armstrong, C.C. de Wit, *Friction Modeling and Compensation*, The Control Handbook, CRC Press, 1995

See Also

Rotational Damper

Rotational Hard Stop

Rotational Spring

Rotational Hard Stop

Purpose

Double-sided rotational hard stop

Library

Mechanical Rotational Elements

Description

The Rotational Hard Stop block represents a double-sided mechanical rotational hard stop that restricts motion of a body between upper and lower bounds. Both ports of the block are of mechanical rotational type. The impact interaction between the slider and the stops is assumed to be elastic. The stop is implemented as a spring that comes into contact with the slider as the gap is cleared. The spring opposes slider penetration into the stop with the force linearly proportional to this penetration. To account for energy dissipation and nonelastic effects, the damping is introduced as a block parameter, thus making it possible to account for energy loss.

The hard stop is described with the following equations:

$$T = \begin{cases} K_p \cdot \delta + D_p (\omega_R - \omega_C) & \text{for } \delta >= g_p \\ 0 & \text{for } g_n < \delta < g_p \\ K_n \cdot \delta + D_n (\omega_R - \omega_C) & \text{for } \delta <= g_n \end{cases}$$

$$\delta = \varphi_R - \varphi_C$$

$$\omega_R = \frac{d\varphi_R}{dt}$$

$$\omega_C = \frac{d\varphi_C}{dt}$$

where

T Interaction torque between the slider and the case

Relative angular displacement between the slider and the case

g_p Gap between the slider and the case in positive direction

g_n Gap between the slider and the case in negative direction

ω_R, ω_C Absolute angular velocities of terminals R and C, respectively

φ_R, φ_C Absolute angular displacements of terminals R and C, respectively

K_p Contact stiffness at positive restriction

K_n Contact stiffness at negative restriction

D_p Damping coefficient at positive restriction

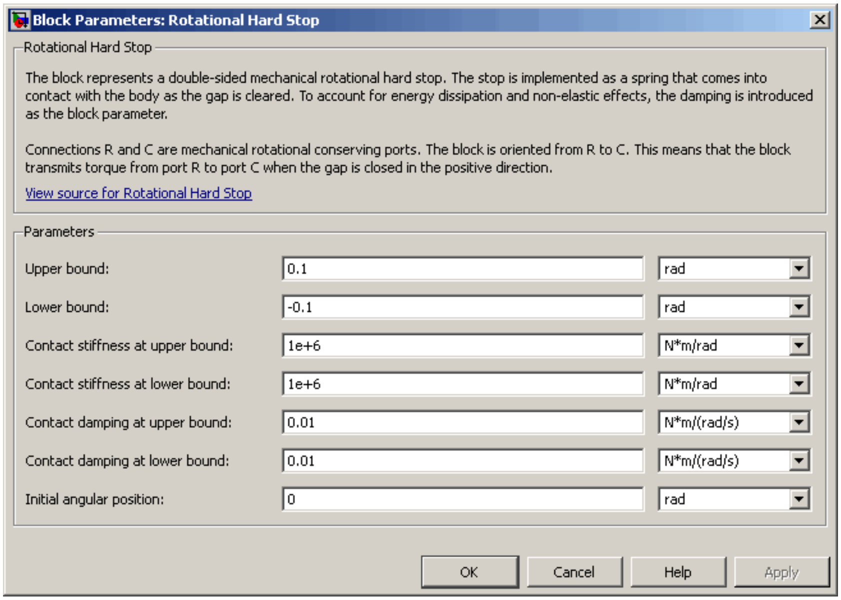
D_n Damping coefficient at negative restriction

t Time

The equations are derived with respect to the local coordinate system whose axis is directed clockwise from port R to port C. The terms “positive” and “negative” in the variable descriptions refer to this coordinate system, and the gap in negative direction must be specified with negative value. If the local coordinate system is not aligned with the globally assigned positive direction, the gaps interchange their values with respective sign adjustment.

The block is oriented from R to C. This means that the block transmits torque from port R to port C when the gap in positive direction is cleared up.

Dialog Box and Parameters



Rotational Hard Stop

The block represents a double-sided mechanical rotational hard stop. The stop is implemented as a spring that comes into contact with the body as the gap is cleared. To account for energy dissipation and non-elastic effects, the damping is introduced as the block parameter.

Connections R and C are mechanical rotational conserving ports. The block is oriented from R to C. This means that the block transmits torque from port R to port C when the gap is closed in the positive direction.

View source for Rotational Hard Stop

Parameters

Upper bound:	0.1	rad
Lower bound:	-0.1	rad
Contact stiffness at upper bound:	1e+6	N*m/rad
Contact stiffness at lower bound:	1e+6	N*m/rad
Contact damping at upper bound:	0.01	N*m/(rad/s)
Contact damping at lower bound:	0.01	N*m/(rad/s)
Initial angular position:	0	rad

Upper bound

Gap between the slider and the upper bound. The direction is specified with respect to the local coordinate system, with the slider located in the origin. A positive value of the parameter specifies the gap between the slider and the upper bound. A negative value sets the slider as penetrating into the upper bound. The default value is `0.1` rad.

Lower bound

Gap between the slider and the lower bound. The direction is specified with respect to the local coordinate system, with the slider located in the origin. A negative value of the parameter specifies the gap between the slider and the lower bound. A positive value sets the slider as penetrating into the lower bound. The default value is `-0.1` rad.

Contact stiffness at upper bound

The parameter specifies the elastic property of colliding bodies when the slider hits the upper bound. The greater the value of the parameter, the less the bodies penetrate into each other, the more rigid the impact becomes. Lesser value of the parameter makes contact softer, but generally improves convergence and computational efficiency. The default value is `1e6` N*m/rad.

Contact stiffness at lower bound

The parameter specifies the elastic property of colliding bodies when the slider hits the lower bound. The greater the value of the parameter, the less the bodies penetrate into each other, the more rigid the impact becomes. Lesser value of the parameter makes contact softer, but generally improves convergence and computational efficiency. The default value is `1e6` N*m/rad.

Contact damping at upper bound

The parameter specifies dissipating property of colliding bodies when the slider hits the upper bound. At zero damping, the impact is close to an absolutely elastic one. The greater the value of the parameter, the more energy dissipates during an interaction. Keep in mind that damping affects slider motion as long as the slider is in contact with the stop, including the period when slider is pulled back from the contact. For computational efficiency and convergence reasons, MathWorks recommends that you assign a nonzero value to this parameter. The default value is 0.01 N*m*s/rad.

Contact damping at lower bound

The parameter specifies dissipating property of colliding bodies when the slider hits the lower bound. At zero damping, the impact is close to an absolutely elastic one. The greater the value of the parameter, the more energy dissipates during an interaction. Keep in mind that damping affects slider motion as long as the slider is in contact with the stop, including the period when slider is pulled back from the contact. For computational efficiency and convergence reasons, MathWorks recommends that you assign a nonzero value to this parameter. The default value is 0.01 N*m*s/rad.

Initial angular position

The initial position of the slider, in angular units, with respect to the local coordinate system that is used for specifying upper and lower bounds. This parameter specifies the initial condition for use in computing the block's initial state at the beginning of a simulation run. For more information, see "Initial Conditions Computation". The default value is `0`.

Ports

The block has the following ports:

`R`

Mechanical rotational conserving port associated with the slider that travels between stops installed on the case.

`C`

Mechanical rotational conserving port associated with the case.

See Also

Rotational Damper

Rotational Friction

Rotational Spring

Rotational Hydro-Mechanical Converter

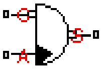
Purpose

Ideal hydro-mechanical transducer as building block for rotary actuators

Library

Hydraulic Elements

Description



The Rotational Hydro-Mechanical Converter block models an ideal transducer that converts hydraulic energy into mechanical energy, in the form of rotational motion of the converter shaft, and vice versa. Physically, the converter represents the main component of a single-acting rotary vane actuator. Using this block as a basic element, you can build a large variety of rotary actuators by adding application-specific effects, such as fluid compressibility, leakage, friction, hard stops, and so on.

The converter is simulated according to the following equations:

$$q = D(\omega_S - \omega_C) \cdot or$$

$$T = D \cdot p \cdot or$$

where

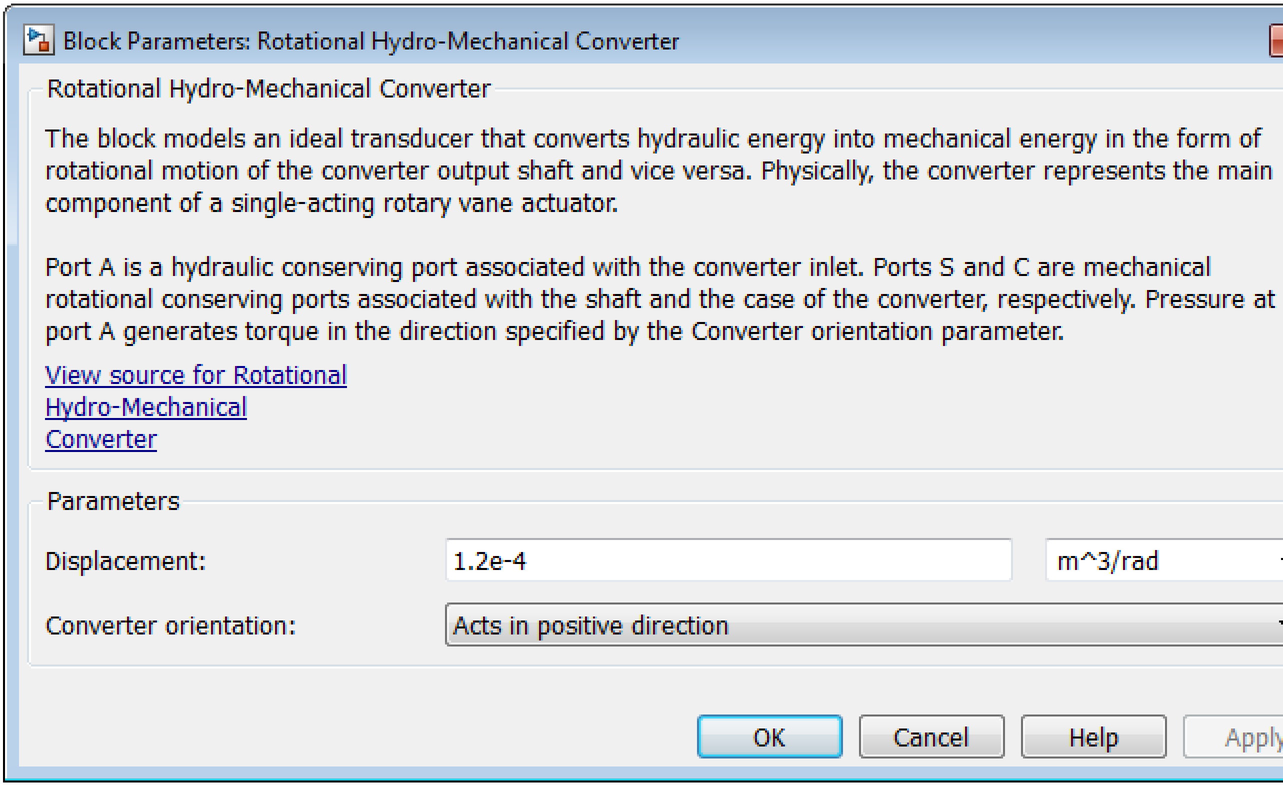
q	Flow rate to the converter chamber
D	Converter displacement, or fluid volume needed to rotate the shaft per angle unit
ω_S	Converter shaft angular velocity
ω_C	Converter case angular velocity
F	Torque on the shaft
p	Gauge pressure of fluid in the converter chamber
or	Converter orientation with respect to the globally assigned positive direction. If pressure applied at port A generates torque in positive direction, *or* equals 1. If pressure applied at port A generates torque in negative direction, *or* equals –1.

Port A is a hydraulic conserving port associated with the converter inlet. Ports S and C are mechanical rotational conserving ports associated with the shaft and the case of the converter, respectively. Pressure at port A generates torque in the direction specified by the **Converter orientation** parameter.

Basic Assumptions and Limitations

The block simulates an ideal converter, with only the transduction property considered. No inertia, friction, leakage, or other effects are taken into account.

Dialog Box and Parameters



Rotational Hydro-Mechanical Converter

The block models an ideal transducer that converts hydraulic energy into mechanical energy in the form of rotational motion of the converter output shaft and vice versa. Physically, the converter represents the main component of a single-acting rotary vane actuator.

Port A is a hydraulic conserving port associated with the converter inlet. Ports S and C are mechanical rotational conserving ports associated with the shaft and the case of the converter, respectively. Pressure at port A generates torque in the direction specified by the Converter orientation parameter.

View source for Rotational Hydro-Mechanical Converter

Parameters

Displacement: 1.2e-4 m^3/rad

Converter orientation: Acts in positive direction

Displacement

Effective converter displacement. The default value is `1.2e-4` m^3/rad.

Converter orientation

Specifies converter orientation with respect to the globally assigned positive direction. The converter can be installed in two different ways, depending upon whether it generates torque in the positive or in the negative direction when pressure is applied at its inlet. If pressure applied at port A generates torque in negative direction, set the parameter to `Acts in negative direction`. The default value is `Acts in positive direction`.

Restricted Parameters

When your model is in Restricted editing mode, you cannot modify the following parameter:

- **Converter orientation**

All other block parameters are available for modification.

Ports

The block has the following ports:

`A`

Hydraulic conserving port associated with the converter inlet.

`S`

Mechanical rotational conserving port associated with the shaft of the converter.

`C`

Mechanical rotational conserving port associated with the case of the converter.

See Also

Translational Hydro-Mechanical Converter

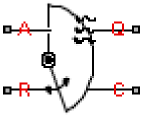
Purpose

Interface between liquid and rotational mechanical subsystems

Library

Thermal Liquid/Elements

Description



The Rotational Mechanical Converter (TL) block represents the liquid side of a rotational mechanical interface. This interface converts liquid pressure into torque and vice versa. The output torque acts in a single direction, set using a **Mechanical orientation** parameter.

The rotational mechanical interface contains no hard stops. To include hard stops, use the Simscape Rotational Hard Stop block. A model of a rotational hydraulic actuator, for example, requires both blocks.

Port A is a thermal liquid conserving port corresponding to the converter inlet. Liquid pressure in the converter equals that at port A. Port Q is a thermal conserving port for modeling heat exchange between the converter liquid and the converter housing. Liquid temperature in the converter equals that at port Q.

The block models the dynamic evolution of temperature in the converter. The block can also model dynamic compressibility effects in the liquid. The following equations govern the dynamic behavior of liquid at the interface:

$$\dot{m}_A = \varepsilon \cdot \rho_{\text{int}} \cdot D \cdot \Omega_{\text{int}\mathit{erface}} + \begin{cases} 0, & \text{if fluid dynamic compressibility is 'Of} \\ V \cdot \rho_{\text{int}} \cdot \left(\dfrac{1}{\beta_{\text{int}}} \cdot \dfrac{dp_{\text{int}}}{dt} + \alpha_{\text{int}} \cdot \dfrac{dT_{\text{int}}}{dt} \right), & \text{if fluid dynamic compressibility is 'Or} \end{cases}$$

where

D	Converter displacement
	Mechanical orientation (1 for Positive, -1 for Negative)
	Torque the liquid exerts on the converter interface
V	Liquid volume in the converter
v_A	Liquid velocity into the converter at port A

$_{\text{interface}}$ Interface angular velocity (positive for converter expansion, negative for converter contraction)

p_{int} Liquid pressure in the converter

T_{int} Liquid temperature in the converter

$\dot{m}_A$ Mass flow rate into the converter at port A

$_{\text{int}}$ Liquid bulk modulus in the converter

$_{\text{int}}$ Liquid coefficient of expansion in the converter

u_{int} Liquid internal energy in the converter

$_{\text{int}}$ Liquid density in the converter

$\dot{}_{\text{A}}$, $\dot{}_{\text{Q}}$ Thermal fluxes into the converter at ports A and Q

To view the block source code, at the MATLAB command line enter:

```
edit <matlabroot>\toolbox\physmod\simscape\library\m\
+foundation\+thermal_liquid\+elements\<converter>.ssc
```

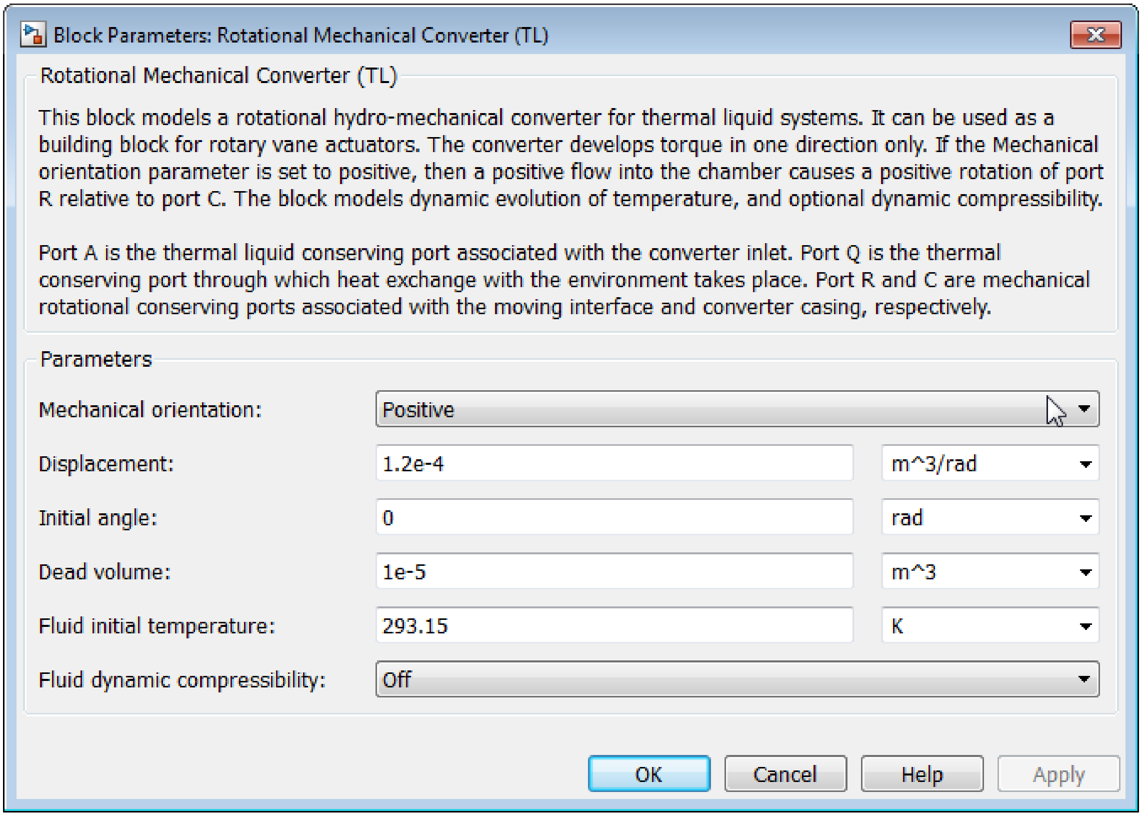
Replace `<matlabroot>` with the output of the `matlabroot` command, and `<converter>` with the compressibility configuration mode:

- `rotational_converter_dynamic_compressibility` for the code corresponding to fluid dynamic compressibility ‘On’
- `rotational_converter_steady_compressibility` for the code corresponding to fluid dynamic compressibility ‘Off’

Assumptions and Limitations

- Converter walls are not compliant. They cannot deform regardless of internal pressure and temperature.
- The converter contains no mechanical hard stop.

Dialog Box and Parameters



Rotational Mechanical Converter (TL)

This block models a rotational hydro-mechanical converter for thermal liquid systems. It can be used as a building block for rotary vane actuators. The converter develops torque in one direction only. If the Mechanical orientation parameter is set to positive, then a positive flow into the chamber causes a positive rotation of port R relative to port C. The block models dynamic evolution of temperature, and optional dynamic compressibility.

Port A is the thermal liquid conserving port associated with the converter inlet. Port Q is the thermal conserving port through which heat exchange with the environment takes place. Port R and C are mechanical rotational conserving ports associated with the moving interface and converter casing, respectively.

Parameters

Mechanical orientation: Positive

Displacement: 1.2e-4 m^3/rad

Initial angle: 0 rad

Dead volume: 1e-5 m^3

Fluid initial temperature: 293.15 K

Fluid dynamic compressibility: Off

Mechanical orientation

Select the relative orientation of the converter with respect to the thermal liquid system. The relative orientation determines the rotation direction associated with positive flow into the converter. That direction is positive if the mechanical orientation of the converter is positive. It is negative if the mechanical orientation of the converter is negative. The default setting is `Positive`.

Displacement
Enter the displaced liquid volume corresponding to a unit rotation angle of the spinning converter interface. The default value is `1.2e-4` m^3/rad.

Initial angle
Enter the rotation angle between the spinning converter interface and the clamping structure at time zero. The default value is `0` rad.

Dead volume
Enter the liquid volume remaining in the converter at a zero rotation angle. The default value is `1e-5` m^3.

Fluid initial temperature
Enter the liquid temperature in the converter at time zero. The default value is `293.15` K.

Fluid dynamic compressibility
Select whether to include the effect of fluid dynamic compressibility on the transient response of the converter model. Selecting `On` exposes an additional parameter. The default setting is `Off`.

Fluid initial pressure
Enter the liquid pressure in the converter at time zero. This parameter is visible only if **Fluid dynamic compressibility** is `On`. The default value is `1` atm.

Ports

This block has the following ports.

A	Thermal liquid conserving port
Q	Thermal conserving port

R Rotational mechanical conserving port associated with the moving interface

C Rotational mechanical conserving port associated with the converter casing

See Also

Translational Mechanical Converter (TL)

Rotational Pneumatic-Mechanical Converter

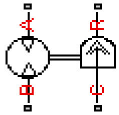
Purpose

Interface between pneumatic and mechanical rotational domains

Library

Pneumatic Elements

Description



The Rotational Pneumatic-Mechanical Converter block provides an interface between the pneumatic and the mechanical rotational domains. Use it as a building block for modeling pneumatic pumps and motors.

The pneumatic flow rate and mechanical rotation are related by the following equations:

$$Q = D \cdot \omega$$

$$T = \begin{cases} D \cdot (p_A - p_B) \cdot \eta & \text{for } (p_A - p_B) \cdot \omega >= 0 \\ D \cdot (p_A - p_B) / \eta & \text{for } (p_A - p_B) \cdot \omega < 0 \end{cases}$$

where

Q	Volumetric flow rate flowing from port A to port B
p_A	Pressure at port A
p_B	Pressure at port B
ω	Shaft angular rotational speed
T	Mechanical torque
D	Volumetric displacement per unit rotation
η	Converter efficiency

The torque equation depends on the direction of power flow, and is always such that the conversion results in some thermal losses.

From considering energy flow, the heat flow out (q_o) of the converter must equate to the heat flow in (q_i) minus mechanical work done. Therefore, the heat equations are:

$$q_i = |G| \cdot c_p \cdot T_i$$

$$q_o = \begin{cases} q_i - D \cdot (p_A - p_B) \cdot \omega \cdot \eta & \text{for } (p_A - p_B) \cdot \omega >= 0 \\ q_i - D \cdot (p_A - p_B) \cdot \omega / \eta & \text{for } (p_A - p_B) \cdot \omega < 0 \end{cases}$$

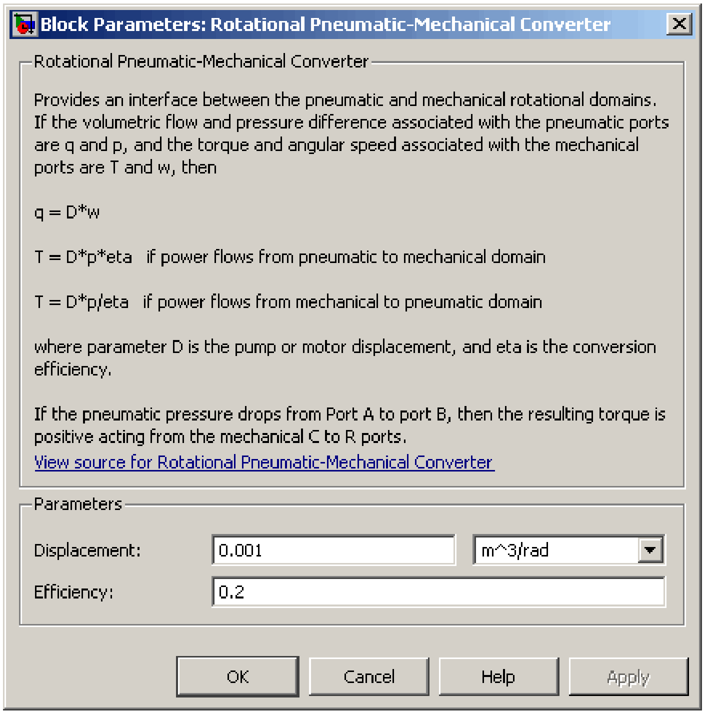
where G is the mass flow rate.

If the pneumatic pressure drops from port A to port B, then the resulting torque is positive acting from the mechanical port C to port R.

Basic Assumptions and Limitations

- Conversion efficiency is constant, that is, it does not depend on torque or speed.
- Gas flow rate is linearly dependent of pump speed.
- The process is adiabatic, that is, there is no heat transfer with the environment.
- Gravitational effects can be neglected.

Dialog Box and Parameters



Rotational Pneumatic-Mechanical Converter

Provides an interface between the pneumatic and mechanical rotational domains. If the volumetric flow and pressure difference associated with the pneumatic ports are q and p, and the torque and angular speed associated with the mechanical ports are T and w, then

q = D*w

T = D*p*eta   if power flows from pneumatic to mechanical domain

T = D*p/eta   if power flows from mechanical to pneumatic domain

where parameter D is the pump or motor displacement, and eta is the conversion efficiency.

If the pneumatic pressure drops from Port A to port B, then the resulting torque is positive acting from the mechanical C to R ports.

View source for Rotational Pneumatic-Mechanical Converter

Parameters

Displacement: 0.001 m^3/rad

Efficiency: 0.2

Displacement

Specify the effective piston displacement, as volume per unit angle. The default value is `.001` m^3/rad.

Efficiency

Specify the converter efficiency. The default value is `0.2`.

Ports

The block has the following ports:

A

Pneumatic conserving port associated with the converter inlet.

B

Pneumatic conserving port associated with the converter outlet.

`R`

Mechanical rotational conserving port associated with the piston (rod).

`C`

Mechanical rotational conserving port associated with the reference (case).

See Also

Rotary Pneumatic Piston Chamber

Rotational Spring

Purpose

Ideal spring in mechanical rotational systems

Library

Mechanical Rotational Elements

Description

The Rotational Spring block represents an ideal mechanical rotational linear spring, described with the following equations:

$$T = K \cdot \varphi$$

$$\varphi = \varphi_{init} + \varphi_R - \varphi_C$$

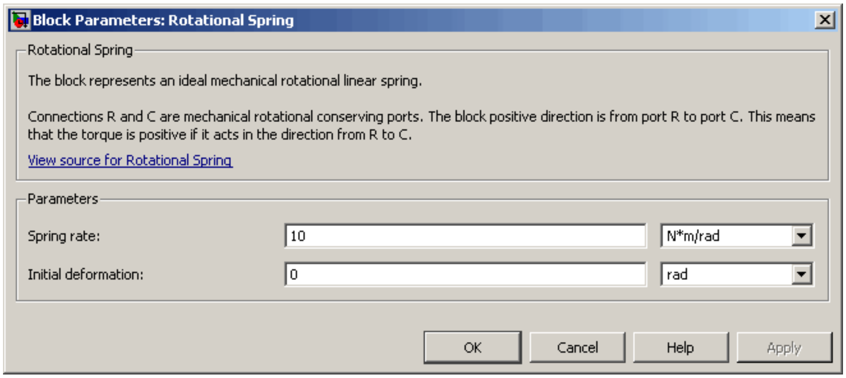
$$\omega = \frac{d\varphi}{dt}$$

where

T	Torque transmitted through the spring
K	Spring rate
φ	Relative displacement angle (spring deformation)
φ_{init}	Spring preliminary winding (spring offset)
φ_{R},φ_{C}	Absolute angular displacements of terminals R and C, respectively
ω	Relative angular velocity
t	Time

The block positive direction is from port R to port C. This means that the torque is positive if it acts in the direction from R to C.

Dialog Box and Parameters



Block Parameters: Rotational Spring

Rotational Spring

The block represents an ideal mechanical rotational linear spring.

Connections R and C are mechanical rotational conserving ports. The block positive direction is from port R to port C. This means that the torque is positive if it acts in the direction from R to C.

View source for Rotational Spring

Parameters

Spring rate: 10 N*m/rad

Initial deformation: 0 rad

Spring rate

Spring rate. The default value is 10 N*m/rad.

Initial deformation

Spring initial deformation, or offset, in angular units. The deformation is determined as $\varphi = \varphi_{init} + \varphi_R - \varphi_C$, where φ_{init} is the initial deformation, and φ_R,$\varphi_C$are the absolute angular displacements of terminals R and C in the globally assigned positive direction. The spring can be initially compressed ($\varphi_{init} > 0$) or stretched ($\varphi_{init} < 0$). This parameter specifies the initial condition for use in computing the block's initial state at the beginning of a simulation run. For more information, see "Initial Conditions Computation". The default value is 0.

Ports

The block has the following ports:

R

Mechanical rotational conserving port.

C

Mechanical rotational conserving port.

See Also

Rotational Damper

Rotational Friction

Rotational Hard Stop

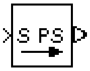
Purpose

Convert Simulink input signal into physical signal

Library

Utilities

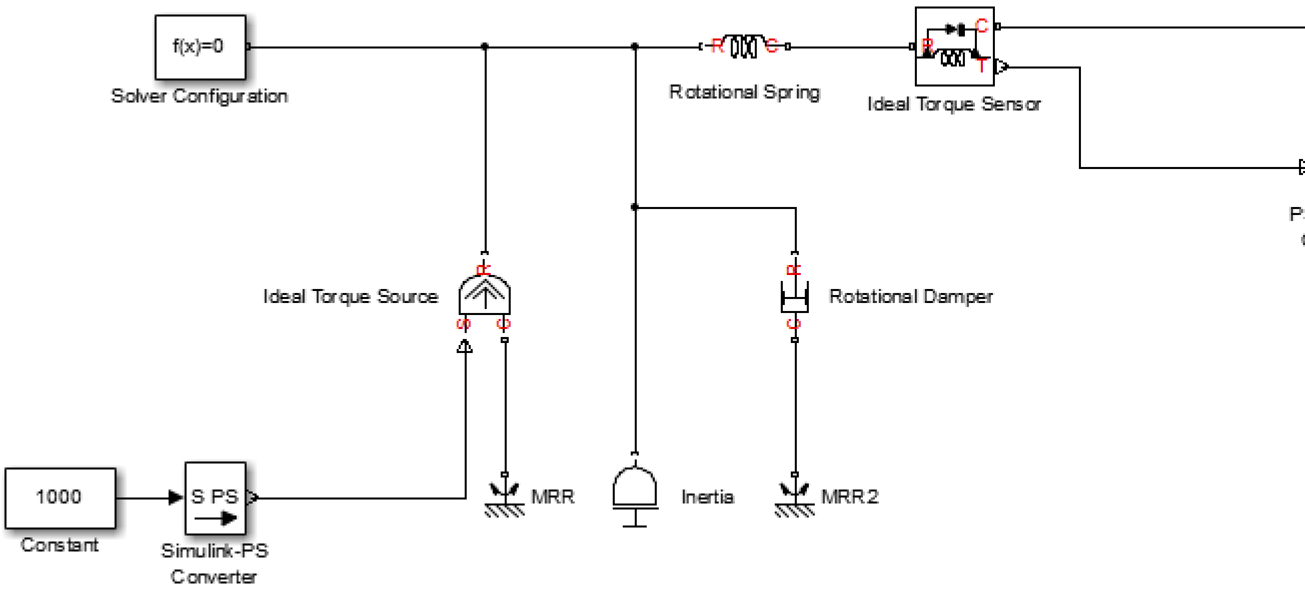
Description



The Simulink-PS Converter block converts the input Simulink signal into a physical signal. Use this block to connect Simulink sources or other Simulink blocks to the inputs of a Physical Network diagram.

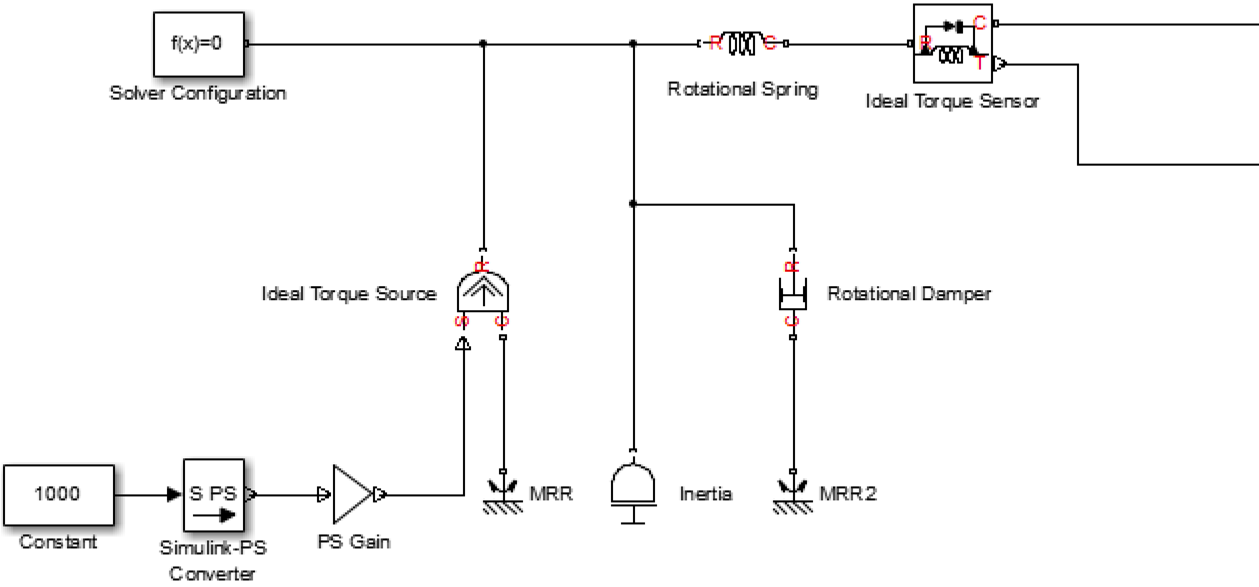
You specify the desired units as the **Input signal unit** parameter. If you leave the block unitless, with the **Input signal unit** parameter set to `1`, then the physical signal units are inferred from the destination block. The default destination block units are meter-kilogram-second or MKS (SI). If you specify different units, commensurate with the expected default units of the destination block input, then the unit manager attaches these units to the input Simulink signal value and performs the necessary unit conversion when providing the signal to the destination block.

In the diagram below, the Ideal Torque Source block expects a torque signal, in N*m, on its S port. The Constant source block provides the value for this input signal. If you left the Simulink-PS Converter block unitless, the Ideal Torque Source block would generate torque of 1000 N*m. The parameters of other blocks in this example are chosen so that the output value of the Ideal Torque Sensor block is equal to the torque generated by the Ideal Torque Source block, and therefore the Display block would show the value of 1000. If you change the **Input signal unit** parameter value in the Simulink-PS Converter block to `N*cm`, the unit manager performs the conversion and the Ideal Torque Source block generates torque of 10 N*m; the torque value in the Display block changes to 10, as shown in the diagram.



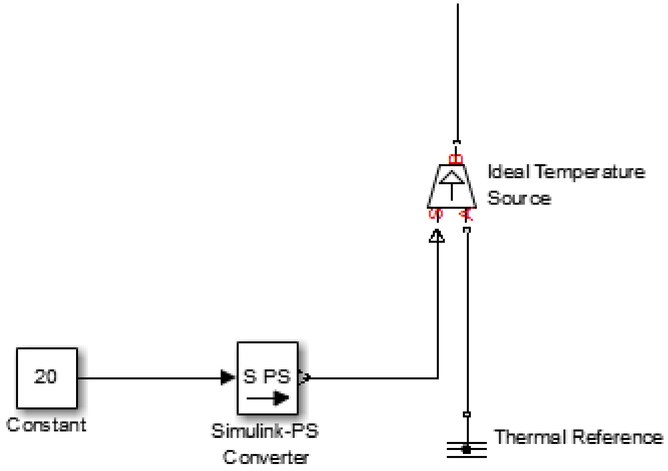
Note Currently, physical units are not propagated through the blocks in the Physical Signals library, such as PS Add, PS Gain, and so on. If your diagram contains a Physical Signals block after a Simulink-PS Converter block, the unit specification in the Simulink-PS Converter block does not propagate to the rest of the network.

In the following example, the PS Gain block is installed after the Simulink-PS Converter block. It stops the unit propagation to the rest of the physical network, and the Ideal Torque Source block will generate torque of 1000 N*m regardless of the **Input signal unit** parameter setting in the Simulink-PS Converter block.



When the input signal is related to thermodynamic variables and contains units of temperature, you must decide whether affine conversion needs to be applied. For more information, see “When to Apply Affine Conversion”. Usually, if the input signal represents a relative temperature, that is, a change in temperature, you need to apply linear conversion, $\Delta T_{new} = L * \Delta T_{old}$ (the default method). However, if the input signal represents an absolute temperature, you need to apply affine conversion, $T_{new} = L * T_{old} + O$.

For example, in the Simulink-PS Converter block shown in the following diagram, if you type `C` in the **Input signal unit** field and select the **Apply affine conversion** check box, the temperature generated by the Ideal Temperature Source block is equal to 293.15 K. However, if you leave the **Apply affine conversion** check box clear, the output of the Ideal Temperature Source block is 20 K.

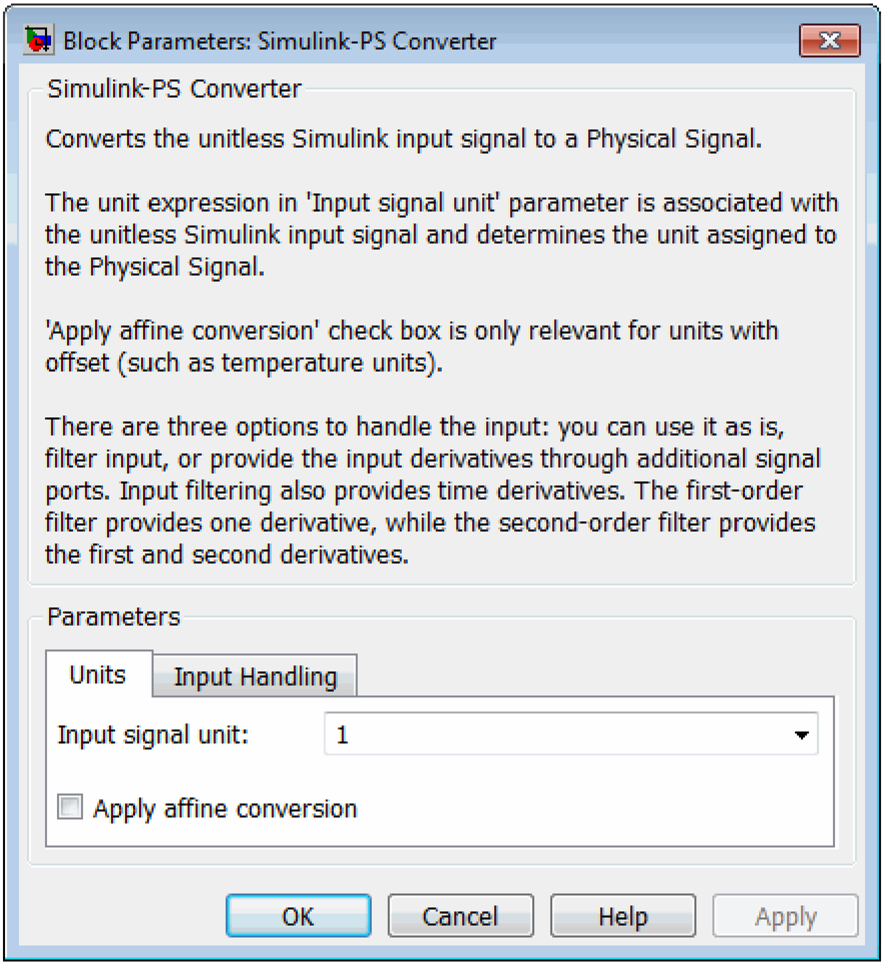


Dialog Box and Parameters

The block dialog box has two tabs:

- “Units” on page 1-353
- “Input Handling” on page 1-355

Units



Block Parameters: Simulink-PS Converter

Simulink-PS Converter

Converts the unitless Simulink input signal to a Physical Signal.

The unit expression in 'Input signal unit' parameter is associated with the unitless Simulink input signal and determines the unit assigned to the Physical Signal.

'Apply affine conversion' check box is only relevant for units with offset (such as temperature units).

There are three options to handle the input: you can use it as is, filter input, or provide the input derivatives through additional signal ports. Input filtering also provides time derivatives. The first-order filter provides one derivative, while the second-order filter provides the first and second derivatives.

Parameters

Input signal unit: 1

□ Apply affine conversion

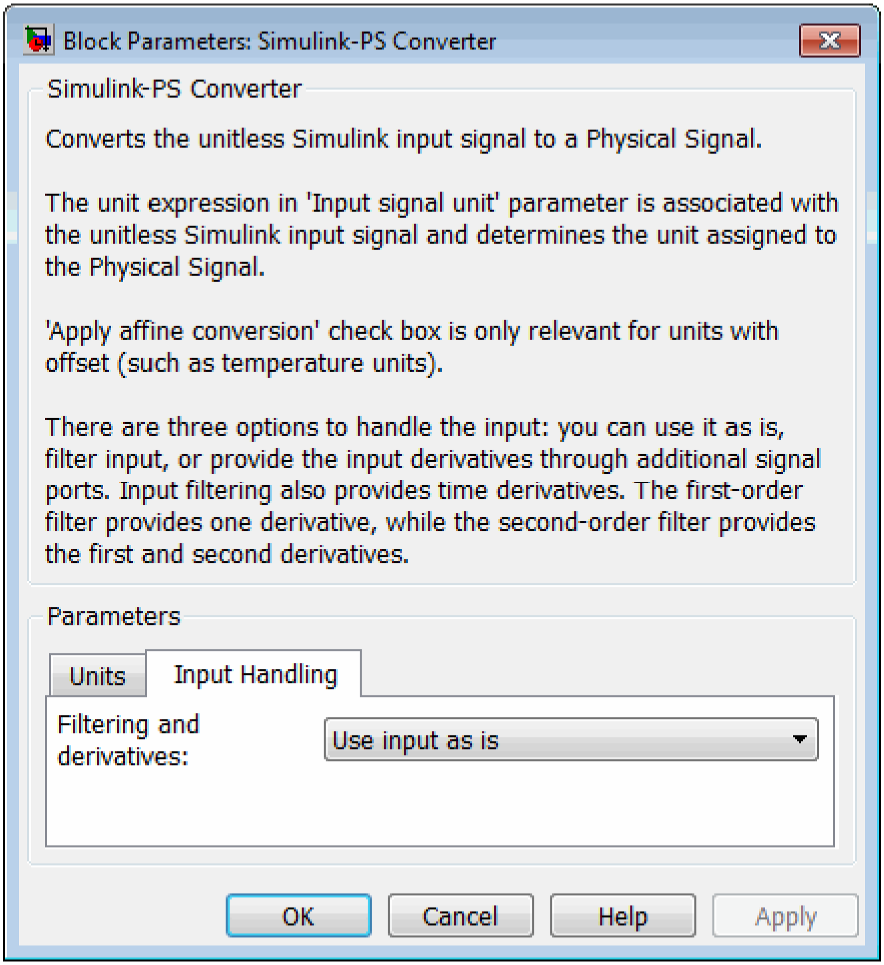
Input signal unit

Units to be assigned to the physical signal. These units must be commensurate with the expected default units of the destination block input. You can select a unit from the drop-down list, or type the desired unit name, such as `rpm`, or a valid expression, such as `rad/s`. For more information and a list of unit abbreviations, see "How to Specify Units in Block Dialogs"and "Unit Definitions". The default value is `1`, which means that the units of the physical signal at the block output match the expected default units of the destination block input.

Apply affine conversion

This check box is applicable only for units that can be converted either with or without an affine offset, such as thermal units. For more information, see "Thermal Unit Conversions".

Input Handling



Block Parameters: Simulink-PS Converter

Simulink-PS Converter

Converts the unitless Simulink input signal to a Physical Signal.

The unit expression in 'Input signal unit' parameter is associated with the unitless Simulink input signal and determines the unit assigned to the Physical Signal.

'Apply affine conversion' check box is only relevant for units with offset (such as temperature units).

There are three options to handle the input: you can use it as is, filter input, or provide the input derivatives through additional signal ports. Input filtering also provides time derivatives. The first-order filter provides one derivative, while the second-order filter provides the first and second derivatives.

Parameters

Filtering and derivatives: Use input as is

Filtering and derivatives

This parameter lets you filter input and thus provide time derivatives of the input signal, or provide the time derivatives through additional input ports on the Simulink-PS Converter block:

- `Use input as is` — Do not perform input filtering or otherwise provide time derivatives of the input signal. This is the default method. If you use an explicit solver, MathWorks recommends that you provide input derivatives by selecting one of the other options for this parameter. For more information, see “Harmonizing Simulink and Simscape Solvers”.
- `Filter input` — Provide input derivatives by Filter the input through a low-pass filter, which also provides input derivatives. In this case, the input signal is modified (through filtering) before being converted to a physical signal. The first-order filter provides one derivative, while the second-order filter provides the first and second derivatives. If you use this option, set the appropriate **Input filtering time constant** parameter value, as described below.
- `Provide input derivative(s)` — Provide time derivatives of the input signal as additional input signals to the Simulink-PS Converter block. If you select this option, the input signal is not modified. You can provide just the first derivative, or both the first and second derivatives, by using the **Input derivatives** parameter, as described below.

Input filtering order

This parameter is applicable only if the **Filtering and derivatives** parameter is set to `Filter input`. It lets you specify the number of time derivatives computed for the input signal by selecting the filter order:

- `First-order filtering` — Provides only the first derivative.
- `Second-order filtering` — Provides the first and second derivatives.

Input filtering time constant (in seconds)

This parameter is applicable only if the **Filtering and derivatives** parameter is set to `Filter input`. It specifies the filter time constant, which controls the filtering of the input signal. The filtered input follows the true input but is smoothed, with a lag on the order of the time constant chosen. You should set the time constant to a value no larger than the smallest time interval of interest in the system. The trade-off in choosing a very small time constant is that the filtered input signal will be closer to the true input signal, at the cost of increasing the stiffness of the system and slowing down the simulation. The default value is `.001` s.

Input derivatives

This parameter is applicable only if the **Filtering and derivatives** parameter is set to `Provide input derivative(s)`. It lets you provide time derivatives of the input signal as additional input signals:

- `Provide first derivative` — If you select this option, an additional Simulink input port appears on the Simulink-PS Converter block, to let you connect the signal providing input derivatives.
- `Provide first and second derivatives` — If you select this option, two additional Simulink input ports appear on the Simulink-PS Converter block, to let you connect the signals providing input derivatives.

Restricted Parameters

When your model is in Restricted editing mode, you cannot modify any of the block parameters, with the following exception: if the **Filtering and derivatives** parameter is set to `Filter input` prior to entering Restricted mode, you can change the value of the **Input filtering time constant** parameter.

Ports

The block has one or two Simulink input ports, depending on the **Input derivatives** parameter value, located on its left side, and a physical signal output port, located on its right side (in the block default orientation).

See Also

PS-Simulink Converter

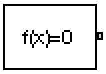
Purpose

Physical Networks environment and solver configuration

Library

Utilities

Description



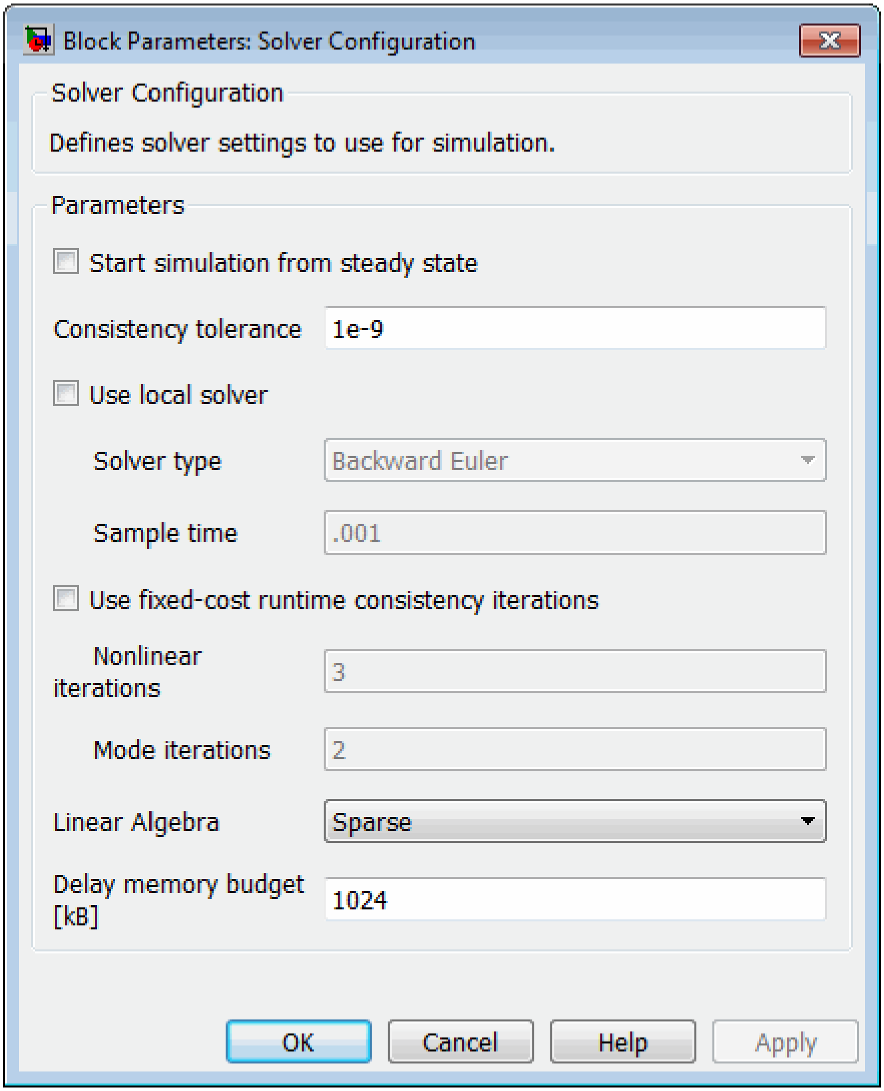
Each physical network represented by a connected Simscape block diagram requires solver settings information for simulation. The Solver Configuration block specifies the solver parameters that your model needs before you can begin simulation.

Each topologically distinct Simscape block diagram requires exactly one Solver Configuration block to be connected to it.

Ports

The block has one conserving port. You can add this block anywhere on a physical network circuit by creating a branching point and connecting it to the only port of the Solver Configuration block.

Dialog Box and Parameters



Block Parameters: Solver Configuration

Solver Configuration

Defines solver settings to use for simulation.

Parameters

□ Start simulation from steady state

Consistency tolerance 1e-9

□ Use local solver

Solver type Backward Euler

Sample time .001

□ Use fixed-cost runtime consistency iterations

Nonlinear iterations 3

Mode iterations 2

Linear Algebra Sparse

Delay memory budget [kB] 1024

Start simulation from steady state

When this box is selected, the solver attempts to find the steady state that would result if the inputs to the system were held constant for a sufficiently large time, starting from the initial state obtained from the initial conditions computation. For more information, see "Initial Conditions Computation". Simulation then starts from this steady state.

Note Using the **Initial state** option on the **Data Import/Export** pane of the Configuration Parameters dialog box overrides the **Start simulation from steady state** option.

Consistency tolerance

This parameter affects the nonlinear solver used for computing initial conditions and for transient initialization. It determines how accurately the algebraic constraints are to be satisfied at the beginning of simulation and after every discrete event (for example, a discontinuity resulting from a valve opening, a hard stop, and so on). Decrease the parameter value (that is, tighten tolerance) to obtain a more reliable time simulation. Increase the parameter value (that is, relax the tolerance) if solving for initial conditions failed to converge, or to reduce the computation time.

The default value is `1e-9`, which is applicable to most cases.

Use local solver

Lets you use a sample-based local solver with a sample time specified by the **Sample time** parameter. In sample-based simulation, all the Physical Network states, which are otherwise continuous, become represented to Simulink as discrete states. The solver updates the states once per time step. This option is especially useful for generated code or hardware-in-the-loop (HIL) simulations.

Note If you use a local solver, simultaneous use of Simulink or Simulink Control Design™ linearization tools is not recommended.

Solver type
Select the solver type used for updating the states: `Backward Euler` or `Trapezoidal Rule`. The **Use local solver** check box must be selected. The default is `Backward Euler`.

Sample time
Specify the sample time for the local solver. The **Use local solver** check box must be selected. The sample time must be positive. The default is `.001` s.

Use fixed-cost runtime consistency iterations
Lets you perform transient initialization at a fixed computational cost.

If you select this check box, you can specify the maximum number of nonlinear and mode iterations for transient initialization. If the system does not converge upon reaching these numbers, it ignores the failure and goes to the next step.

If you clear the check box, the system uses a more robust and time-consuming algorithm, and errors out if it fails to reach convergence at the time of transient initialization.

Nonlinear iterations
Specify the maximum number of Newton iterations at the time of transient initialization. The **Use fixed-cost runtime consistency iterations** check box must be selected. The default number is `3`.

Mode iterations
Specify the maximum number of mode iterations at the time of transient initialization. The **Use fixed-cost runtime**

consistency iterations check box must be selected. The default number is 2.

Only one major mode update per step is performed when using local solvers, therefore this parameter is not available if the **Use local solver** check box is selected.

Linear Algebra

Specifies how the solver treats matrices. The parameter can have one of two values: `Sparse` or `Full`. The default value of the parameter is `Sparse`.

Your choice of this parameter value, `Sparse` or `Full`, is implemented in both model simulation and code generated from your model.

Delay memory budget [kB]

Specify the maximum memory budget, in kB, allowed for processing delays when simulating models that contain either blocks from the Delays library, or custom blocks using the `delay` Simscape language construct. The purpose of this parameter is to protect against excessive memory swapping. If this budget is exceeded, simulation errors out. You can adjust this value based on your available memory resources. The default number is 1024 kB.

See Also

- “How Simscape Simulation Works”
- “Setting Up Solvers for Physical Models”
- “Customizing Solvers for Physical Models”
- “Code Generation”
- “Real-Time Simulation”
- “Finding an Operating Point”
- “Linearizing at an Operating Point”

Switch

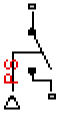
Purpose

Switch controlled by external physical signal

Library

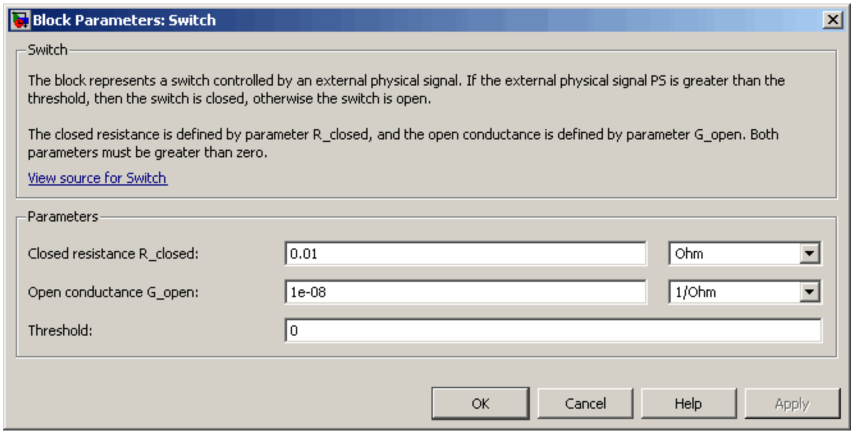
Electrical Elements

Description



The Switch block models a switch controlled by an external physical signal. If the external physical signal PS is greater than the value specified in the **Threshold** parameter, then the switch is closed, otherwise the switch is open.

Dialog Box and Parameters



Block Parameters: Switch

Switch

The block represents a switch controlled by an external physical signal. If the external physical signal PS is greater than the threshold, then the switch is closed, otherwise the switch is open.

The closed resistance is defined by parameter R_closed, and the open conductance is defined by parameter G_open. Both parameters must be greater than zero.

View source for Switch

Parameters

Closed resistance R_closed: 0.01 Ohm

Open conductance G_open: 1e-08 1/Ohm

Threshold: 0

Closed resistance R_closed

The resistance of the switch when it is closed. The parameter value must be greater than zero. The default value is `0.01` Ω.

Open conductance G_open

The conductance of the switch when it is open. The parameter value must be greater than zero. The default value is `1e-8` 1/Ω.

Threshold

The threshold value for opening and closing the switch. If the external physical signal PS is greater than this value, then the switch is closed, otherwise the switch is open. The default value is 0.

Ports

The block has two electrical conserving ports and one physical signal port PS.

See Also

PS Switch

Temperature Reservoir (TL)

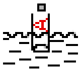
Purpose

Infinite open reservoir at a fixed temperature

Library

Thermal Liquid/Elements

Description



The Temperature Reservoir (TL) block represents an infinite open reservoir at a fixed temperature. Because it is open, the reservoir and its inlet are at atmospheric pressure. Port A, a thermal liquid conserving port, represents that inlet.

The inlet temperature depends on the direction of liquid flow. If liquid flows into the reservoir, the inlet temperature equals that of the upstream liquid. The reservoir acts as a heat sink. If liquid flows out of the reservoir, the inlet temperature equals that of the reservoir. The reservoir acts as a heat source.

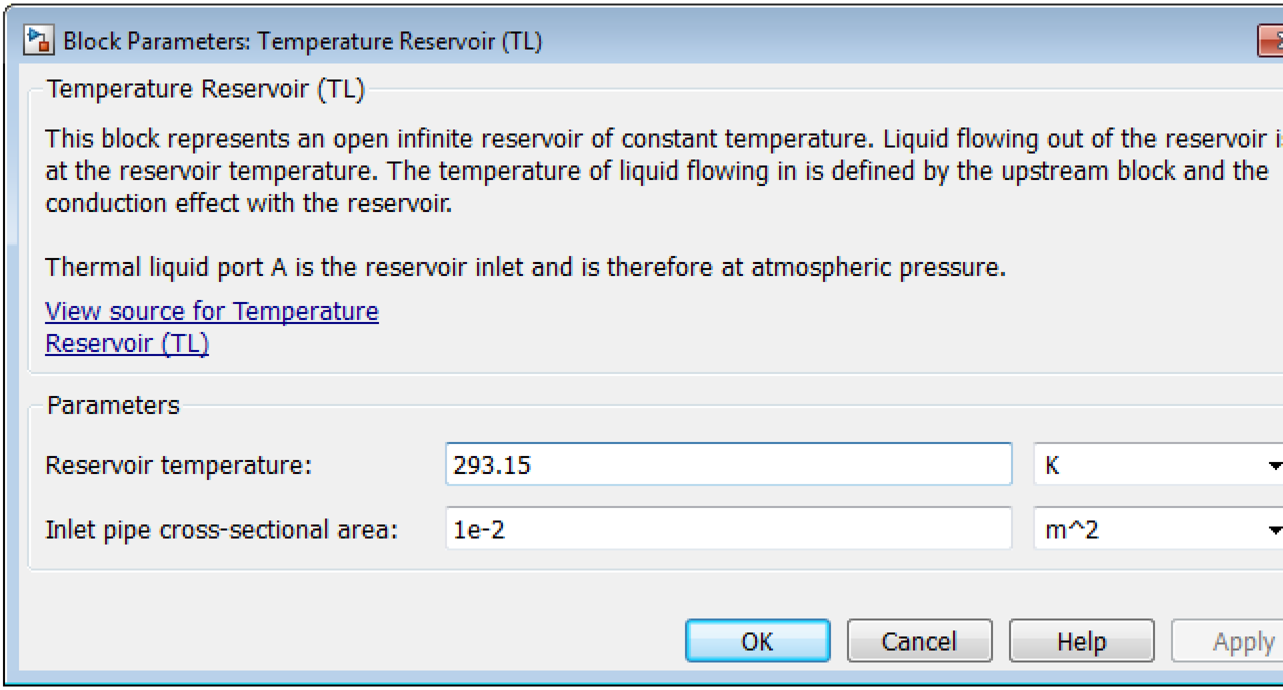
To ensure a smooth temperature change at the reservoir inlet during liquid flow reversal, the block includes heat conduction along a length equal to the effective diameter of the pipe. This diameter is a function of the specified cross-sectional area of the inlet pipe.

This block also functions as a reference point for pressure and temperature measurements in a pipe network. These measurements are relative to atmospheric pressure and reservoir temperature, respectively.

Assumptions and Limitations

- Reservoir temperature is constant.

Dialog Box and Parameters



Block Parameters: Temperature Reservoir (TL)

Temperature Reservoir (TL)

This block represents an open infinite reservoir of constant temperature. Liquid flowing out of the reservoir i at the reservoir temperature. The temperature of liquid flowing in is defined by the upstream block and the conduction effect with the reservoir.

Thermal liquid port A is the reservoir inlet and is therefore at atmospheric pressure.

View source for Temperature Reservoir (TL)

Parameters

Reservoir temperature:	293.15	K
Inlet pipe cross-sectional area:	1e-2	m^2

Reservoir temperature

Enter the temperature of the liquid reservoir. During simulation, the block holds this temperature constant. The default value is `293.15` K.

Inlet pipe cross-sectional area

Enter the cross-sectional area of the reservoir inlet pipe. The block uses this area to determine the characteristic length of the pipe along which heat conduction occurs. The default value is `1e-2` m^2.

Ports

The block has one thermal liquid conserving port, A.

See Also

Constant Volume Chamber (TL)

Controlled Temperature Reservoir (TL)

Thermal Liquid Settings (TL)

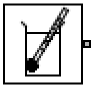
Purpose

Thermodynamic properties of the liquid medium

Library

Thermal Liquid/Utilities

Description

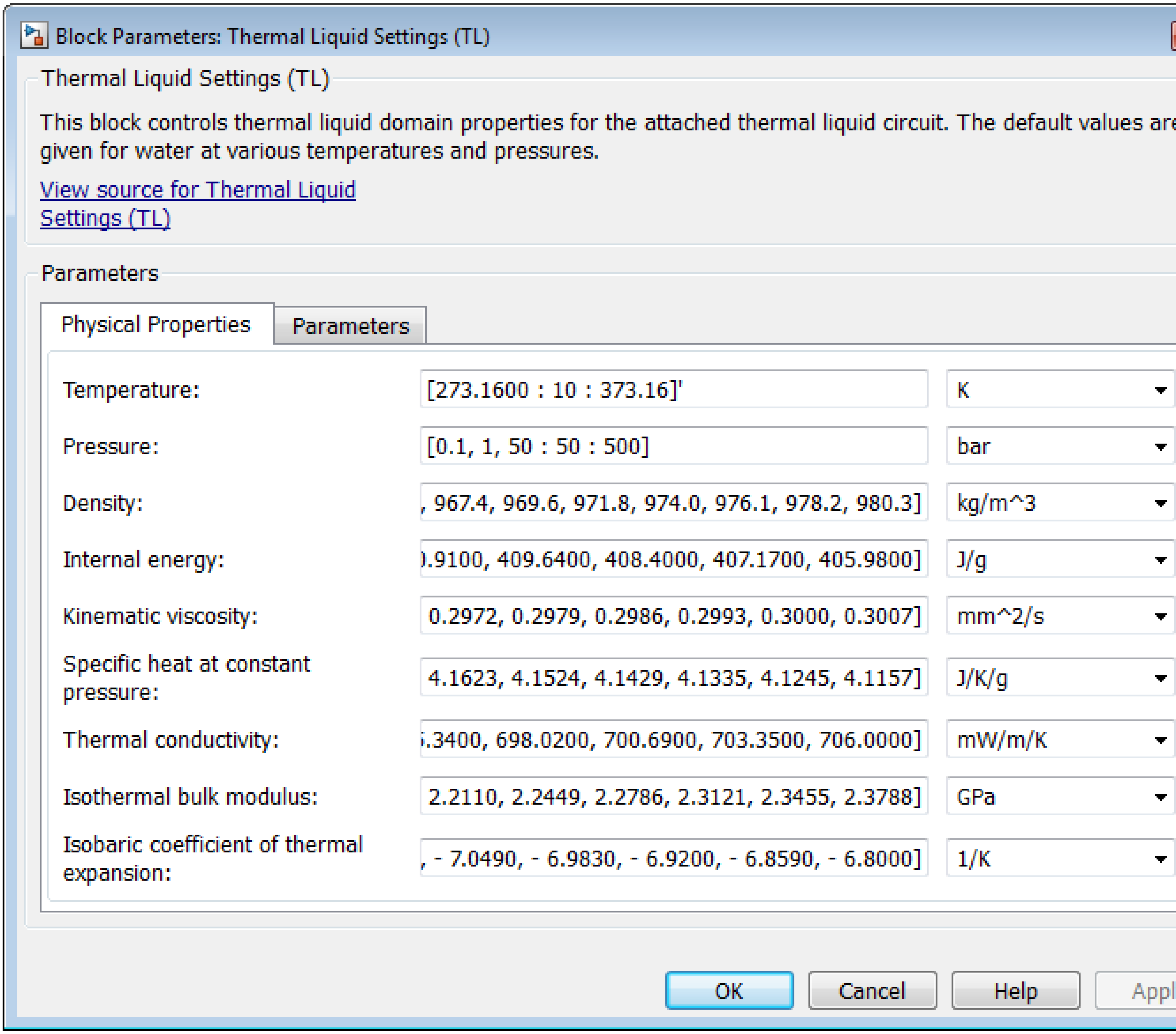


The Thermal Liquid Settings (TL) block represents the liquid medium in a thermal liquid system. A set of physical properties governs the thermodynamic behavior of the liquid medium. The properties are global: they apply to the entire thermal liquid system containing this block. Each topologically distinct thermal liquid block network must contain exactly one Thermal Liquid Settings (TL) block.

Inputs for the physical properties of the liquid medium are lookup tables. The tables provide the numerical values of the physical properties at discrete temperatures and pressures. For temperatures and pressures not included in the table but within the data range, the block applies linear interpolation between the two nearest data points.

The block accepts the temperature-pressure validity region of a thermal liquid model as input. Adjust this region to ensure the model simulates only at acceptable temperature and pressure values, e.g., to avoid phase change or component failure.

Dialog Box and Parameters



Block Parameters: Thermal Liquid Settings (TL)

Thermal Liquid Settings (TL)

This block controls thermal liquid domain properties for the attached thermal liquid circuit. The default values are given for water at various temperatures and pressures.

View source for Thermal Liquid Settings (TL)

Parameters

Temperature:	[273.1600 : 10 : 373.16]'	K
Pressure:	[0.1, 1, 50 : 50 : 500]	bar
Density:	, 967.4, 969.6, 971.8, 974.0, 976.1, 978.2, 980.3]	kg/m^3
Internal energy:	).9100, 409.6400, 408.4000, 407.1700, 405.9800]	J/g
Kinematic viscosity:	0.2972, 0.2979, 0.2986, 0.2993, 0.3000, 0.3007]	mm^2/s
Specific heat at constant pressure:	4.1623, 4.1524, 4.1429, 4.1335, 4.1245, 4.1157]	J/K/g
Thermal conductivity:	;.3400, 698.0200, 700.6900, 703.3500, 706.0000]	mW/m/K
Isothermal bulk modulus:	2.2110, 2.2449, 2.2786, 2.3121, 2.3455, 2.3788]	GPa
Isobaric coefficient of thermal expansion:	, - 7.0490, - 6.9830, - 6.9200, - 6.8590, - 6.8000]	1/K

Physical Properties

Temperature
Enter an N×1 column vector with the temperature values for the liquid property lookup tables. Each temperature value corresponds to a lookup table row. For smooth interpolation between lookup table data points, keep N large. The default vector is `[273.1600 : 10 : 373.16]'` K.

Pressure
Enter a 1×M row vector with the pressure values for the liquid property lookup tables. Each pressure value corresponds to a lookup table column. For smooth interpolation between lookup table data points, keep M large. The default vector is `[0.1, 1, 50 : 50 : 500]` bar.

Density
Enter an N×M matrix with the mass density lookup table values. The lookup table is two-sided: temperature varies from row to row, while pressure varies from column to column. The **Temperature** vector provides the temperature values for the N lookup table rows. The **Pressure** vector provides the pressure values for the M lookup table columns. The default matrix is 11×12. It provides values for water.

Internal energy
Enter an N×M matrix with the internal energy lookup table values. The lookup table is two-sided: temperature varies from row to row, while pressure varies from column to column. The **Temperature** vector provides the temperature values for the N lookup table rows. The **Pressure** vector provides the pressure values for the `M` lookup table columns. The default matrix is 11×12. It provides values for water.

Kinematic viscosity
Enter an N×M matrix with the kinematic viscosity lookup-table values. The lookup table is two-sided: temperature varies from row to row, while pressure varies from column to column. The **Temperature** vector provides the temperature values for the N

lookup-table rows. The **Pressure** vector provides the pressure values for the M lookup table columns. The default matrix is 11×12. It provides values for water.

Specific heat at constant pressure

Enter an N×M matrix with the specific heat lookup table values at constant pressure. The lookup table is two-sided: temperature varies from row to row, while pressure varies from column to column. The **Temperature** vector provides the temperature values for the N lookup table rows. The **Pressure** vector provides the pressure values for the M lookup table columns. The default matrix is 11×12. It provides values for water.

Thermal conductivity

Enter an N×M matrix with the thermal conductivity lookup table values. The lookup table is two-sided: temperature varies from row to row, while pressure varies from column to column. The **Temperature** vector provides the temperature values for the N lookup table rows. The **Pressure** vector provides the pressure values for the M lookup table columns. The default matrix is 11×12. It provides values for water.

Isothermal bulk modulus

Enter an N×M matrix with the isothermal bulk modulus lookup table values. The lookup table is two-sided: temperature varies from row to row, while pressure varies from column to column. The **Temperature** vector provides the temperature values for the N lookup table rows. The **Pressure** vector provides the pressure values for the M lookup table columns. The default matrix is 11×12. It provides values for water.

Isobaric coefficient of thermal expansion

Enter an N×M matrix with the lookup table values of the isobaric thermal expansion coefficient. The lookup table is two-sided: temperature varies from row to row, while pressure varies from column to column. The **Temperature** vector provides the temperature values for the N lookup table rows. The **Pressure** vector provides the pressure values for the M lookup table columns. The default matrix is 11×12. It provides values for water.

Parameters

Valid pressure-temperature region parameterization
Select the parameterization for the valid pressure-temperature region. Simulation stops if pressure or temperature fall outside this valid range. Parameterization options include `By minimum and maximum values`, which provides a rectangular pressure-temperature region, and `By validity matrix`, which provides an arbitrarily shaped pressure-temperature region. The default parameterization is `By minimum and maximum values`.

Minimum valid temperature
Enter the lowest allowable value the liquid temperature can reach. This parameter appears when you select `By minimum and maximum values` in **Valid pressure-temperature region parameterization**. The default value is `273.16` K.

Maximum valid temperature
Enter the highest allowable value the liquid temperature can reach. This parameter appears when you select `By minimum and maximum values` in **Valid pressure-temperature region parameterization**. The default value is `373.16` K.

Minimum valid pressure
Enter the lowest allowable value the liquid pressure can reach. This parameter appears when you select `By minimum and maximum values` in **Valid pressure-temperature region parameterization**. The default value is `0.1` bar.

Maximum valid pressure
Enter the highest allowable value the liquid pressure can reach. This parameter appears when you select `By minimum and maximum values` in **Valid pressure-temperature region parameterization**. The default value is `500` bar.

Validity Matrix

Enter an N×M matrix that specifies which pressure-temperature value pairs in the physical property lookup tables are valid. In this matrix, enter `1` for valid pressure-temperature pairs and `-1` for invalid pairs. This parameter is visible when you select `By validity matrix` in **Valid pressure-temperature region parameterization**. The default matrix is `ones(11,12)`, denoting that all temperature-pressure value pairs in the physical property lookup tables are valid.

Minimum thermal conductance

Lowest allowable value the thermal conductance of the liquid medium can reach. This parameter ensures that, even during strong flow reversal, thermal energy flows smoothly and continuously between blocks. The default value is `1e-3` W/K.

Atmospheric pressure

Value of the absolute pressure in the atmosphere. The default value is `1` atm.

Ports

The block has one thermal liquid conserving port, A.

Purpose

Mass in thermal systems

Library

Thermal Elements

Description

The Thermal Mass block represents a thermal mass, which reflects the ability of a material or a combination of materials to store internal energy. The property is characterized by mass of the material and its specific heat. The thermal mass is described with the following equation:

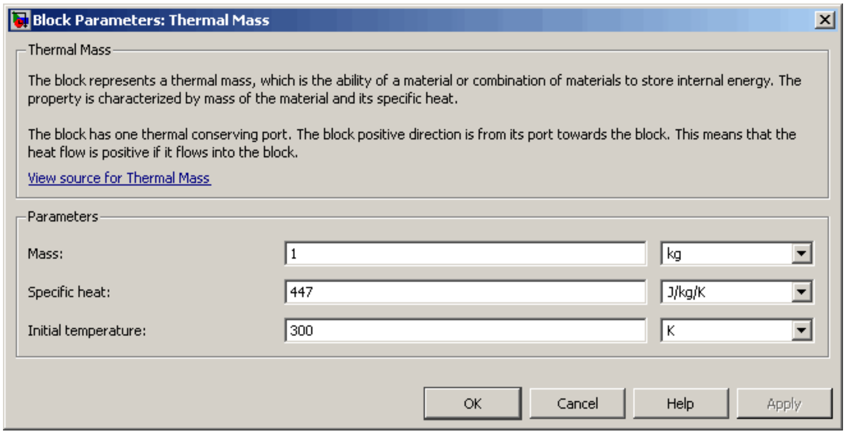
$$Q = c \cdot m \frac{dT}{dt}$$

where

Q	Heat flow
c	Specific heat of mass material
m	Mass
T	Temperature
t	Time

The block has one thermal conserving port. The block positive direction is from its port towards the block. This means that the heat flow is positive if it flows into the block.

Dialog Box and Parameters



Thermal Mass

The block represents a thermal mass, which is the ability of a material or combination of materials to store internal energy. The property is characterized by mass of the material and its specific heat.

The block has one thermal conserving port. The block positive direction is from its port towards the block. This means that the heat flow is positive if it flows into the block.

View source for Thermal Mass

Parameters

Mass: 1 kg

Specific heat: 447 J/kg/K

Initial temperature: 300 K

Mass

Mass. The default value is 1 kg.

Specific heat

Specific heat of the material. The default value is 447 J/kg/K.

Initial temperature

Initial temperature of the mass. This parameter specifies the initial condition for use in computing the block's initial state at the beginning of a simulation run. For more information, see "Initial Conditions Computation". The default value is 300 K.

Ports

The block has one thermal conserving port, associated with the mass connection to the system.

See Also

Mass

Thermal Reference

Purpose

Reference connection for thermal ports

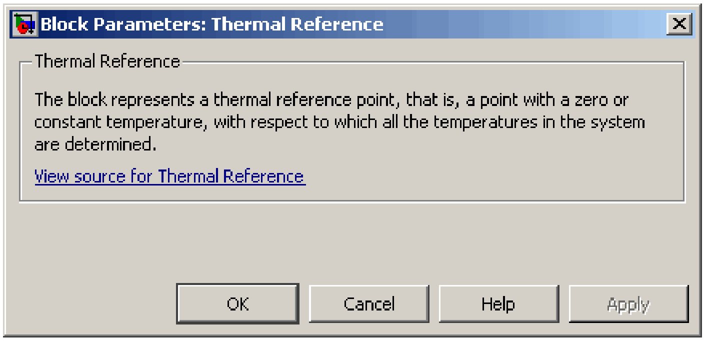
Library

Thermal Elements

Description

The Thermal Reference block represents a thermal reference point, that is, a point with an absolute zero temperature, with respect to which all the temperatures in the system are determined.

Dialog Box and Parameters



Block Parameters: Thermal Reference

Thermal Reference

The block represents a thermal reference point, that is, a point with a zero or constant temperature, with respect to which all the temperatures in the system are determined.

View source for Thermal Reference

The Thermal Reference block has no parameters.

Ports

The block has one thermal conserving port.

See Also

Electrical Reference

Hydraulic Reference

Mechanical Rotational Reference

Mechanical Translational Reference

Translational Damper

Purpose

Viscous damper in mechanical translational systems

Library

Mechanical Translational Elements

Description

The Translational Damper block represents an ideal mechanical translational viscous damper, described with the following equations:

$$F = Dv$$

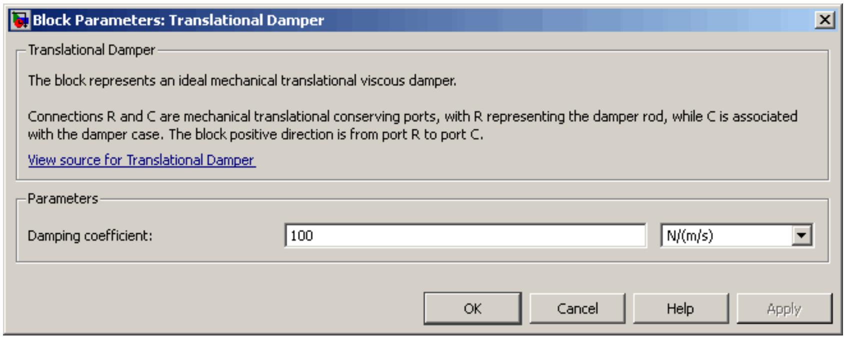
$$v = v_R - v_C$$

where

F	Force transmitted through the damper
D	Damping (viscous friction) coefficient
v	Relative velocity
v_R, v_C	Absolute velocities of terminals R and C, respectively

The block positive direction is from port R to port C. This means that the force is positive if it acts in the direction from R to C.

Dialog Box and Parameters



Translational Damper

The block represents an ideal mechanical translational viscous damper.

Connections R and C are mechanical translational conserving ports, with R representing the damper rod, while C is associated with the damper case. The block positive direction is from port R to port C.

View source for Translational Damper

Parameters

Damping coefficient: 100 N/(m/s)

Damping coefficient

Damping coefficient, defined by viscous friction. The default value is 100 N/(m/s).

Ports

The block has the following ports:

`R`
: Mechanical translational conserving port associated with the damper rod.

`C`
: Mechanical translational conserving port associated with the damper case.

See Also

Translational Friction

Translational Hard Stop

Translational Spring

Translational Electromechanical Converter

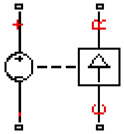
Purpose

Interface between electrical and mechanical translational domains

Library

Electrical Elements

Description



The Translational Electromechanical Converter block provides an interface between the electrical and mechanical translational domains. It converts electrical energy into mechanical energy in the form of translational motion, and vice versa. The converter is described with the following equations:

$$F = K \cdot I$$

$$V = K \cdot U$$

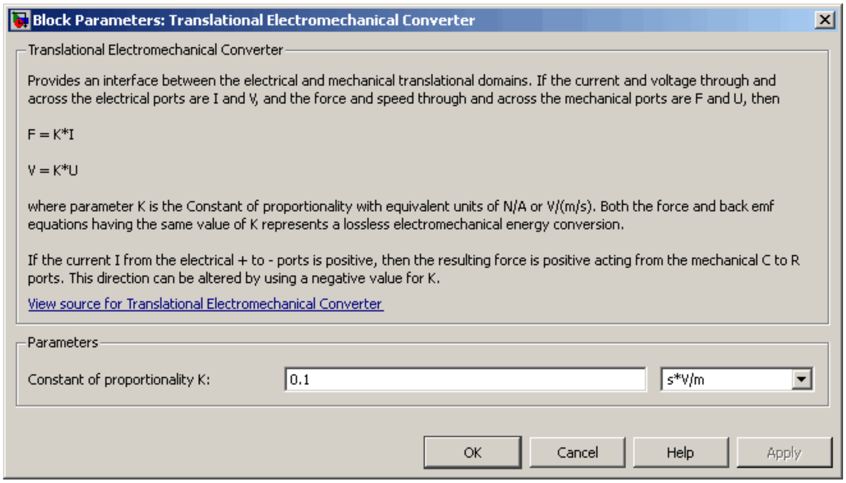
where

V	Voltage across the electrical ports of the converter
I	Current through the electrical ports of the converter
F	Force
U	Speed
K	Constant of proportionality

The Translational Electromechanical Converter block represents a lossless electromechanical energy conversion, therefore the same constant of proportionality is used in both equations.

Connections + and – are conserving electrical ports corresponding to the positive and negative terminals of the converter, respectively. Connections C and R are conserving mechanical translational ports. If the current flowing from the positive to the negative terminal is positive, then the resulting force is positive acting from port C to port R. This direction can be altered by using a negative value for K.

Dialog Box and Parameters



Block Parameters: Translational Electromechanical Converter

Translational Electromechanical Converter

Provides an interface between the electrical and mechanical translational domains. If the current and voltage through and across the electrical ports are I and V, and the force and speed through and across the mechanical ports are F and U, then

F = K*I

V = K*U

where parameter K is the Constant of proportionality with equivalent units of N/A or V/(m/s). Both the force and back emf equations having the same value of K represents a lossless electromechanical energy conversion.

If the current I from the electrical + to - ports is positive, then the resulting force is positive acting from the mechanical C to R ports. This direction can be altered by using a negative value for K.

View source for Translational Electromechanical Converter

Parameters

Constant of proportionality K: 0.1 s*V/m

Constant of proportionality K

Constant of proportionality for electromechanical conversions. The default value is `0.1` V/(m/s).

Ports

The block has the following ports:

`+`

Electrical conserving port associated with the converter positive terminal.

`-`

Electrical conserving port associated with the converter negative terminal.

`C`

Mechanical translational conserving port.

`R`

Mechanical translational conserving port.

See Also

Rotational Electromechanical Converter

Translational Free End

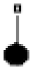
Purpose

Translational port terminator with zero force

Library

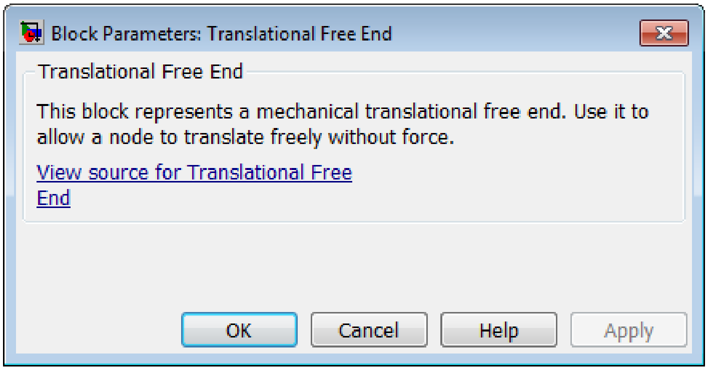
Mechanical Translational Elements

Description



The Translational Free End block represents a mechanical translational port that moves freely, without force. Physical Network block diagrams do not allow unconnected Conserving ports. Use this block to terminate mechanical translational ports (on other blocks) that you wish to leave unconnected.

Dialog Box and Parameters



Block Parameters: Translational Free End

Translational Free End

This block represents a mechanical translational free end. Use it to allow a node to translate freely without force.

View source for Translational Free End

The Translational Free End block has no parameters.

Ports

The block has one mechanical translational conserving port.

See Also

Hydraulic Cap

Open Circuit

Rotational Free End

Purpose

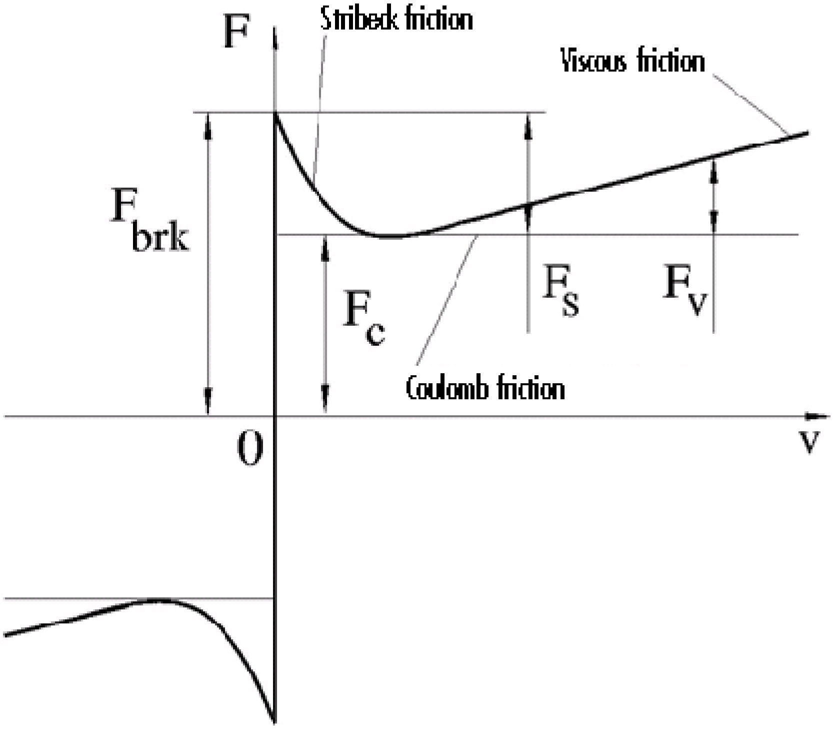
Friction in contact between moving bodies

Library

Mechanical Translational Elements

Description

The Translational Friction block represents friction in contact between moving bodies. The friction force is simulated as a function of relative velocity and is assumed to be the sum of Stribeck, Coulomb, and viscous components, as shown in the following figure.



The Stribeck friction, F_S, is the negatively sloped characteristics taking place at low velocities (see [1]). The Coulomb friction, F_C, results in a constant force at any velocity. The viscous friction, F_V, opposes motion with the force directly proportional to the relative velocity. The sum of the Coulomb and Stribeck frictions at the vicinity of zero velocity is often referred to as the breakaway friction, F_{brk}. The friction is approximated with the following equations:

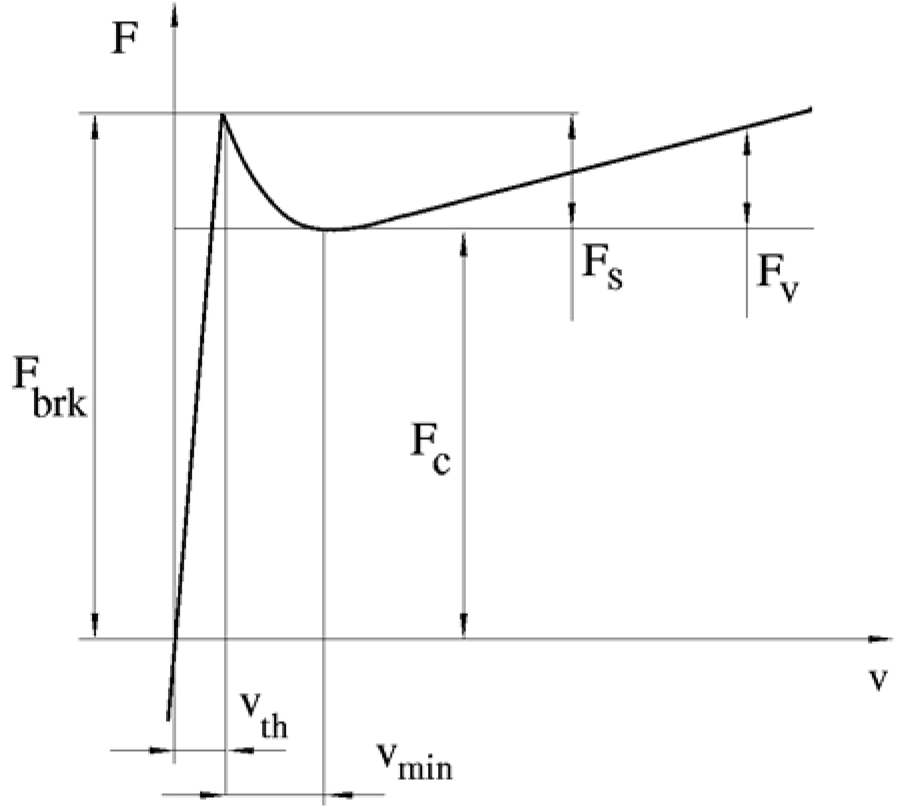
$$F = \left(F_C + \left(F_{brk} - F_C\right) \cdot \exp\left(-c_v \mid v \mid\right)\right) sign\left(v\right) + fv$$

$$v = v_R - v_C$$

where

F	Friction force
F_C	Coulomb friction
F_{brk}	Breakaway friction
c_v	Coefficient
v	Relative velocity
$v_{R,} v_C$	Absolute velocities of terminals R and C, respectively
f	Viscous friction coefficient

The approximation above is too idealistic and has a substantial drawback. The characteristic is discontinuous at $v = 0$, which creates considerable computational problems. It has been proven that the discontinuous friction model is a nonphysical simplification in the sense that the mechanical contact with distributed mass and compliance cannot exhibit an instantaneous change in force (see [1]). There are numerous models of friction without discontinuity. The Translational Friction block implements one of the simplest versions of continuous friction models. The friction force-relative velocity characteristic of this approximation is shown in the following figure.



The discontinuity is eliminated by introducing a very small, but finite, region in the zero velocity vicinity, within which friction force is assumed to be linearly proportional to velocity, with the proportionality coefficient F_{brk}/v_{th}, where v_{th} is the velocity threshold. It has been proven experimentally that the velocity threshold in the range between 10^{-4} and 10^{-6} m/s is a good compromise between the accuracy and computational robustness and effectiveness. Notice that friction force computed with this approximation does not actually stop relative motion when an acting force drops below breakaway friction level. The bodies will creep relative to each other at a very small velocity proportional to acting force.

As a result of introducing the velocity threshold, the block equations are slightly modified:

- If $|v| >= v_{th}$,

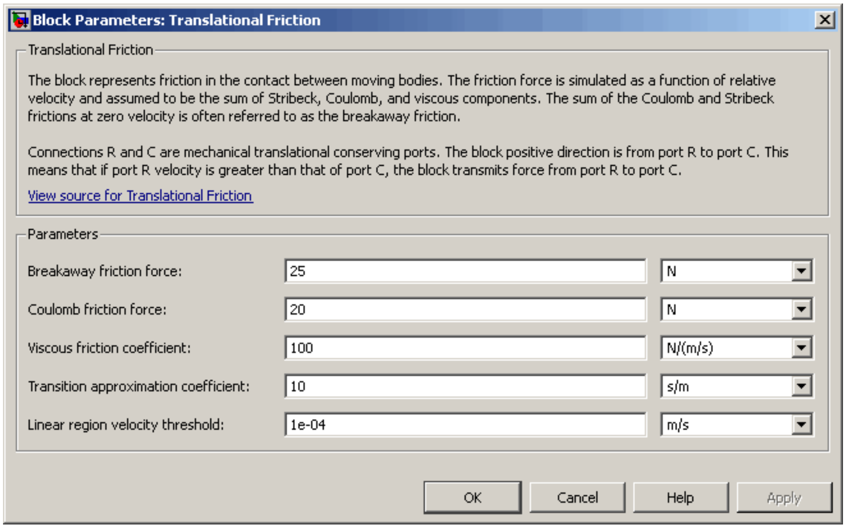
$$F = \left(F_C + (F_{brk} - F_C) \cdot \exp(-c_v\,|v|)\right) sign(v) + fv$$

- If $|v| < v_{th}$,

$$F = v\frac{\left(fv_{th} + \left(F_C + (F_{brk} - F_C) \cdot \exp(-c_v v_{th})\right)\right)}{v_{th}}$$

The block positive direction is from port R to port C. This means that if the port R velocity is greater than that of port C, the block transmits force from R to C.

Dialog Box and Parameters



Translational Friction

The block represents friction in the contact between moving bodies. The friction force is simulated as a function of relative velocity and assumed to be the sum of Stribeck, Coulomb, and viscous components. The sum of the Coulomb and Stribeck frictions at zero velocity is often referred to as the breakaway friction.

Connections R and C are mechanical translational conserving ports. The block positive direction is from port R to port C. This means that if port R velocity is greater than that of port C, the block transmits force from port R to port C.

View source for Translational Friction

Parameters

Breakaway friction force:	25	N
Coulomb friction force:	20	N
Viscous friction coefficient:	100	N/(m/s)
Transition approximation coefficient:	10	s/m
Linear region velocity threshold:	1e-04	m/s

Breakaway friction force

Breakaway friction force, which is the sum of the Coulomb and the static frictions. It must be greater than or equal to the Coulomb friction force value. The default value is 25 N.

Coulomb friction force

Coulomb friction force, which is the friction that opposes motion with a constant force at any velocity. The default value is 20 N.

Viscous friction coefficient

Proportionality coefficient between the friction force and the relative velocity. The parameter value must be greater than or equal to zero. The default value is 100 N/(m/s).

Transition approximation coefficient

The parameter sets the value of coefficient c_v, which is used for the approximation of the transition between the static and the Coulomb frictions. Its value is assigned based on the following considerations: the static friction component reaches approximately 95% of its steady-state value at velocity $3/c_v$, and 98% at velocity $4/c_v$, which makes it possible to develop an approximate relationship c_v ~= 4/$v_{min,}$ where v_{min} is the relative velocity at which friction force has its minimum value. By default, c_v is set to 10 s/m, which corresponds to a minimum friction at velocity of about 0.4 m/s.

Linear region velocity threshold
The parameter sets the small vicinity near zero velocity, within which friction force is considered to be linearly proportional to the relative velocity. MathWorks recommends that you use values in the range between 1e-6 and 1e-4 m/s. The default value is 1e-4 m/s.

Ports

The block has the following ports:

R
Mechanical translational conserving port.

C
Mechanical translational conserving port.

References

[1] B. Armstrong, C.C. de Wit, *Friction Modeling and Compensation*, The Control Handbook, CRC Press, 1995

See Also

Translational Damper

Translational Hard Stop

Translational Spring

Purpose

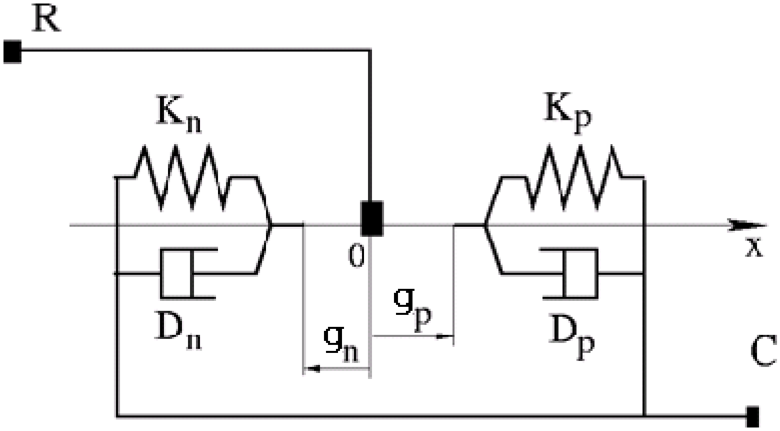
Double-sided translational hard stop

Library

Mechanical Translational Elements

Description

The Translational Hard Stop block represents a double-sided mechanical translational hard stop that restricts motion of a body between upper and lower bounds. Both ports of the block are of mechanical translational type. The impact interaction between the slider and the stops is assumed to be elastic. This means that the stop is represented as a spring that comes into contact with the slider as the gap is cleared and opposes slider penetration into the stop with the force linearly proportional to this penetration. To account for energy dissipation and nonelastic effects, the damping is introduced as the block's parameter, thus making it possible to account for energy loss. The following schematic shows the idealization of the mechanical translational hard stop adopted in the block:



The hard stop is described with the following equations:

$$F = \begin{cases} K_p \cdot \delta + D_p \left(v_R - v_C \right) & \text{for } \delta >= g_p \\ 0 & \text{for } g_n < \delta < g_p \\ K_n \cdot \delta + D_n \left(v_R - v_C \right) & \text{for } \delta <= g_n \end{cases}$$

$$\delta = x_R - x_C$$

$$v_R = \frac{dx_R}{dt}$$

$$v_C = \frac{dx_C}{dt}$$

where

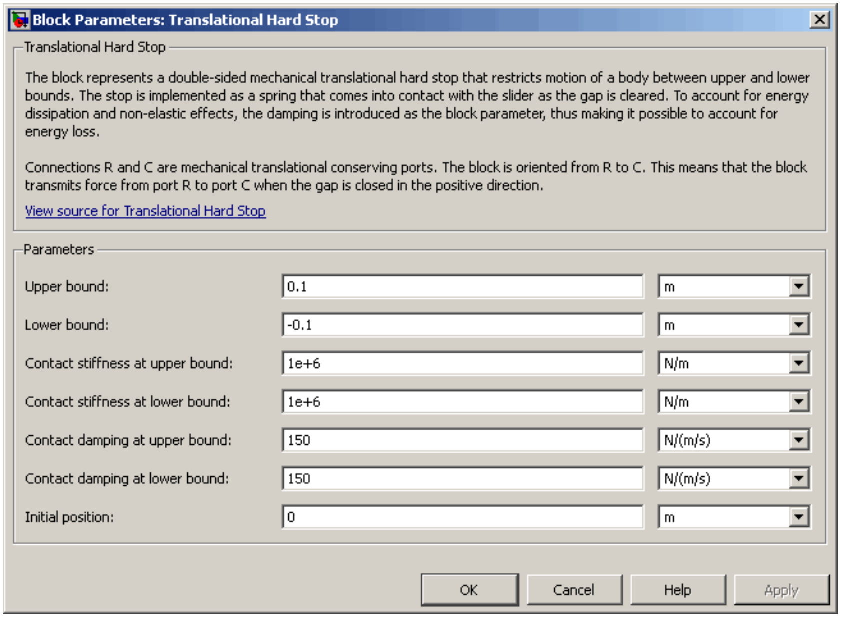
F	Interaction force between the slider and the case
δ	Relative displacement between the slider and the case
g_p	Gap between the slider and the case in positive direction
g_n	Gap between the slider and the case in negative direction
$v_{R,} v_C$	Absolute velocities of terminals R and C, respectively
$x_{R,} x_C$	Absolute displacements of terminals R and C, respectively
K_p	Contact stiffness at positive restriction
K_n	Contact stiffness at negative restriction
D_p	Damping coefficient at positive restriction
D_n	Damping coefficient at negative restriction
t	Time

The equations are derived with respect to the local coordinate system whose axis is directed from port R to port C. The terms “positive” and “negative” in the variable descriptions refer to this coordinate system, and the gap in negative direction must be specified with negative value.

If the local coordinate system is not aligned with the globally assigned positive direction, the gaps interchange their values with respective sign adjustment.

The block is oriented from R to C. This means that the block transmits force from port R to port C when the gap in positive direction is cleared up.

Dialog Box and Parameters



Block Parameters: Translational Hard Stop

Translational Hard Stop

The block represents a double-sided mechanical translational hard stop that restricts motion of a body between upper and lower bounds. The stop is implemented as a spring that comes into contact with the slider as the gap is cleared. To account for energy dissipation and non-elastic effects, the damping is introduced as the block parameter, thus making it possible to account for energy loss.

Connections R and C are mechanical translational conserving ports. The block is oriented from R to C. This means that the block transmits force from port R to port C when the gap is closed in the positive direction.

View source for Translational Hard Stop

Parameters

Upper bound:	0.1	m
Lower bound:	-0.1	m
Contact stiffness at upper bound:	1e+6	N/m
Contact stiffness at lower bound:	1e+6	N/m
Contact damping at upper bound:	150	N/(m/s)
Contact damping at lower bound:	150	N/(m/s)
Initial position:	0	m

Upper bound

Gap between the slider and the upper bound. The direction is specified with respect to the local coordinate system, with the slider located in the origin. A positive value of the parameter specifies the gap between the slider and the upper bound. A negative value sets the slider as penetrating into the upper bound. The default value is `0.1` m.

Lower bound

Gap between the slider and the lower bound. The direction is specified with respect to the local coordinate system, with the slider located in the origin. A negative value of the parameter specifies the gap between the slider and the lower bound. A positive value sets the slider as penetrating into the lower bound. The default value is `-0.1` m.

Contact stiffness at upper bound

The parameter specifies the elastic property of colliding bodies when the slider hits the upper bound. The greater the value of the parameter, the less the bodies penetrate into each other, the more rigid the impact becomes. Lesser value of the parameter makes contact softer, but generally improves convergence and computational efficiency. The default value is `1e6` N/m.

Contact stiffness at lower bound

The parameter specifies the elastic property of colliding bodies when the slider hits the lower bound. The greater the value of the parameter, the less the bodies penetrate into each other, the more rigid the impact becomes. Lesser value of the parameter makes contact softer, but generally improves convergence and computational efficiency. The default value is `1e6` N/m.

Contact damping at upper bound

The parameter specifies dissipating property of colliding bodies when the slider hits the upper bound. At zero damping, the impact is close to an absolutely elastic one. The greater the value of the parameter, the more energy dissipates during an interaction. Keep in mind that damping affects slider motion as long as the slider is in contact with the stop, including the period when slider is pulled back from the contact. For computational efficiency and convergence reasons, MathWorks recommends that you assign a nonzero value to this parameter. The default value is 150 N*s/m.

Contact damping at lower bound

The parameter specifies dissipating property of colliding bodies when the slider hits the lower bound. At zero damping, the impact is close to an absolutely elastic one. The greater the value of the parameter, the more energy dissipates during an interaction. Keep in mind that damping affects slider motion as long as the slider is in contact with the stop, including the period when slider is pulled back from the contact. For computational efficiency and convergence reasons, MathWorks recommends that you assign a nonzero value to this parameter. The default value is 150 N*s/m.

Initial position

The initial position of the slider, with respect to the local coordinate system that is used for specifying upper and lower bounds. This parameter specifies the initial condition for use in computing the block's initial state at the beginning of a simulation run. For more information, see "Initial Conditions Computation". The default value is `0`.

Ports

The block has the following ports:

`R`

Mechanical translational conserving port associated with the slider that travels between stops installed on the case.

`C`

Mechanical translational conserving port associated with the case.

Examples

The Mechanical System with Translational Hard Stop example illustrates the use of the Translational Hard Stop block in mechanical systems. Two masses are interacting through a hard stop. The mass on the left is driven by an ideal velocity source. Plotting the displacement of the second mass against the displacement of the first mass produces a typical hysteresis curve.

See Also

Translational Damper

Translational Friction

Translational Spring

Translational Hydro-Mechanical Converter

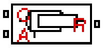
Purpose

Single chamber of hydraulic cylinder for use as building block for various cylinder models

Library

Hydraulic Elements

Description



The Translational Hydro-Mechanical Converter block models an ideal transducer that converts hydraulic energy into mechanical energy in the form of translational motion of the converter output member. Using this block as a basic element, you can build a large variety of hydraulic cylinder models by adding application-specific effects, such as fluid compressibility, leakage, friction, hard stops, and so on.

The converter is simulated according to the following equations:

$$q = A(v_R - v_C) \cdot or$$

$$F = A \cdot p \cdot or$$

where

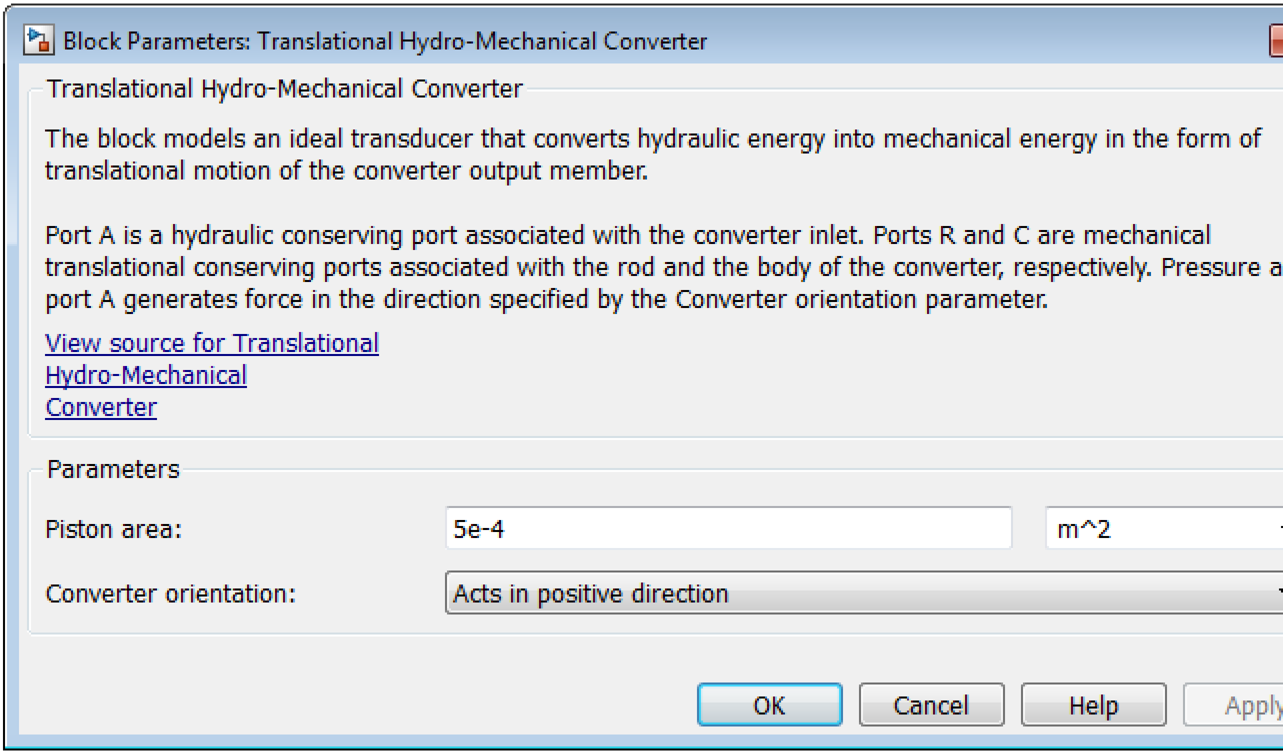
q	Flow rate due to fluid compressibility
A	Effective piston area
v_R	Converter rod velocity
v_C	Converter case velocity
F	Force developed by the converter
p	Gauge pressure of fluid in the converter chamber
or	Converter orientation with respect to the globally assigned positive direction. If pressure applied at port A exerts force in positive direction, *or* equals 1. If pressure applied at port A exerts force in negative direction, *or* equals –1.

Port A is a hydraulic conserving port associated with the converter inlet. Ports R and C are translational mechanical conserving ports associated with the rod and the case of the converter, respectively.

Basic Assumptions and Limitations

The block simulates an ideal converter, with only the transduction property considered. No inertia, friction, leakage, or other effects are taken into account.

Dialog Box and Parameters


Block Parameters: Translational Hydro-Mechanical Converter

Translational Hydro-Mechanical Converter

The block models an ideal transducer that converts hydraulic energy into mechanical energy in the form of translational motion of the converter output member.

Port A is a hydraulic conserving port associated with the converter inlet. Ports R and C are mechanical translational conserving ports associated with the rod and the body of the converter, respectively. Pressure a port A generates force in the direction specified by the Converter orientation parameter.

View source for Translational Hydro-Mechanical Converter

Parameters

Piston area: 5e-4 m^2

Converter orientation: Acts in positive direction

Piston area

Effective piston area. The default value is `5e-4` m^2.

Converter orientation

Specifies converter orientation with respect to the globally assigned positive direction. The converter can be installed in two different ways, depending upon whether it exerts force in the positive or in the negative direction when pressure is applied at its inlet. If pressure applied at port A exerts force in negative direction, set the parameter to `Acts in negative direction`. The default value is `Acts in positive direction`.

Restricted Parameters

When your model is in Restricted editing mode, you cannot modify the following parameter:

- **Converter orientation**

All other block parameters are available for modification.

Ports

The block has the following ports:

`A`

Hydraulic conserving port associated with the converter inlet.

`R`

Mechanical translational conserving port associated with the rod of the converter.

`C`

Mechanical translational conserving port associated with the case of the converter.

See Also

Rotational Hydro-Mechanical Converter

Translational Mechanical Converter (TL)

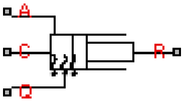
Purpose

Interface between liquid and translational mechanical subsystems

Library

Thermal Liquid/Elements

Description



The Translational Mechanical Converter (TL) block represents the liquid side of a translational mechanical interface. This interface converts liquid pressure into force and vice versa. The output force acts in a single direction, set using a **Mechanical orientation** parameter.

The translational mechanical interface contains no hard stops. To include hard stops, use the Simscape Translational Hard Stop block. A model of a translational hydraulic actuator, for example, requires both blocks.

Port A is a thermal liquid conserving port corresponding to the converter inlet. Liquid pressure in the converter equals that at port A. Port Q is a thermal conserving port for modeling heat exchange between the converter liquid and the converter housing. Liquid temperature in the converter equals that at port Q.

The block models the dynamic evolution of temperature in the converter. The block can also model dynamic compressibility effects in the enclosed liquid. The following equations govern the dynamic behavior of liquid at the interface:

$$\dot{m}_A = \varepsilon \cdot \rho_{\mathrm{int}} \cdot A \cdot v_{\mathrm{int}erface} + \begin{cases} 0, & \text{if fluid dynamic compressibility is 'Off} \\ V \cdot \rho_{\mathrm{int}} \cdot \left(\dfrac{1}{\beta_{\mathrm{int}}} \cdot \dfrac{dp_{\mathrm{int}}}{dt} + \alpha_{\mathrm{int}} \cdot \dfrac{dT_{\mathrm{int}}}{dt} \right), & \text{if fluid dynamic compressibility is 'On} \end{cases}$$

where

A	Cross-sectional area of the interface
	Mechanical orientation (1 for Positive, -1 for Negative)
F	Force the liquid exerts on the interface
V	Liquid volume in the converter
v_A	Liquid velocity into the converter at port A

$v_{\text{interface}}$ Interface velocity (positive for converter expansion, negative for converter contraction)

p_{int} Liquid pressure in the converter

T_{int} Liquid temperature in the converter

$\dot{m}_A$ Mass flow rate into the converter at port A

$_{\text{int}}$ Liquid bulk modulus in the converter

$_{\text{int}}$ Liquid coefficient of thermal expansion in the converter

u_{int} Liquid internal energy in the converter

$_{\text{int}}$ Liquid density in the converter

$\cdot_{\text{A}}$, $\cdot_{\text{Q}}$ Thermal fluxes into the converter at ports A and Q

The liquid velocity follows from the mass flow rate into the converter:

$$v_A = \frac{\dot{m}_A}{A \cdot \rho_{A,u}}$$

where $_{\text{A,u}}$ is the upwind liquid density at port A.

To view the block source code, at the MATLAB command line enter:

```
edit <matlabroot>\toolbox\physmod\simscape\library\m\
+foundation\+thermal_liquid\+elements\<converter>.ssc
```

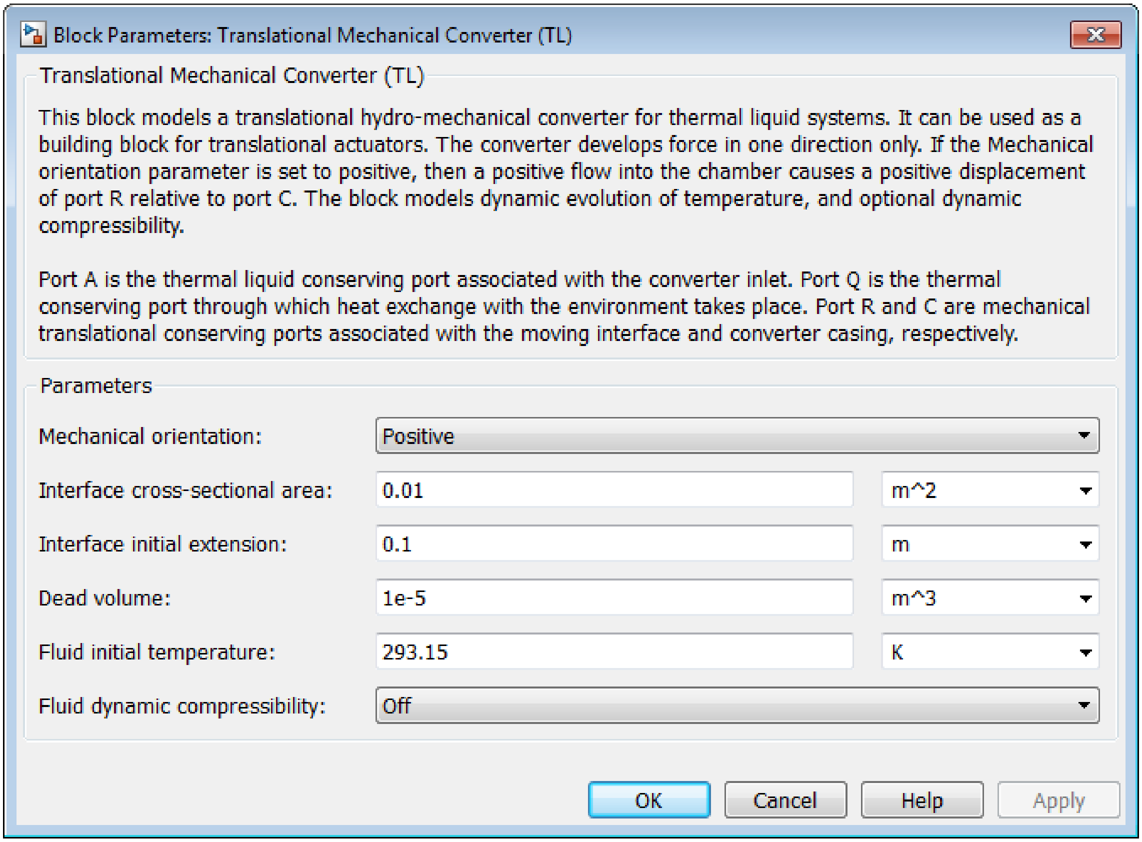
Replace `<matlabroot>` with the output of the `matlabroot` command, and `<converter>` with the compressibility configuration mode using this syntax:

- `translational_converter_dynamic_compressibility` for the code corresponding to fluid dynamic compressibility 'On'
- `translational_converter_steady_compressibility` for the code corresponding to fluid dynamic compressibility 'Off'

Assumptions and Limitations

- Converter walls are not compliant. They cannot deform, regardless of internal pressure and temperature.
- The converter contains no mechanical hard stop.

Dialog Box and Parameters



Block Parameters: Translational Mechanical Converter (TL)

Translational Mechanical Converter (TL)

This block models a translational hydro-mechanical converter for thermal liquid systems. It can be used as a building block for translational actuators. The converter develops force in one direction only. If the Mechanical orientation parameter is set to positive, then a positive flow into the chamber causes a positive displacement of port R relative to port C. The block models dynamic evolution of temperature, and optional dynamic compressibility.

Port A is the thermal liquid conserving port associated with the converter inlet. Port Q is the thermal conserving port through which heat exchange with the environment takes place. Port R and C are mechanical translational conserving ports associated with the moving interface and converter casing, respectively.

Parameters

Mechanical orientation:	Positive	
Interface cross-sectional area:	0.01	m^2
Interface initial extension:	0.1	m
Dead volume:	1e-5	m^3
Fluid initial temperature:	293.15	K
Fluid dynamic compressibility:	Off	

Mechanical orientation

Select the relative orientation of the converter with respect to the thermal liquid system. The relative orientation determines the translation direction associated with positive flow into the converter. That direction is positive if the mechanical orientation of the converter is positive. It is negative if the mechanical orientation of the converter is negative. The default setting is `Positive`.

Interface cross-sectional area
Enter the cross-sectional area of the converter interface. This is the area the liquid must push to generate a force. The default value is `0.01` m^2.

Interface initial extension
Enter the offset distance between the translating converter interface and the converter cap at time zero. The default value is `0`.

Dead volume
Enter the liquid volume remaining in the converter at a zero offset distance. The default value is `1e-5` m^3.

Fluid initial temperature
Enter the liquid temperature in the converter at time zero. The default value is `293.15` K.

Fluid dynamic compressibility
Select whether to include the effect of fluid dynamic compressibility on the transient response of the converter model. Selecting `On` exposes an additional parameter. The default setting is `Off`.

Fluid initial pressure
Enter the liquid pressure in the converter at time zero. This parameter is visible only if **Fluid dynamic compressibility** is `On`. The default value is `1` atm.

Ports

This block has the following ports.

A	Thermal liquid conserving port
Q	Thermal conserving port

R	Translational mechanical conserving port associated with the moving interface
C	Translational mechanical conserving port associated with the converter casing

See Also

Rotational Mechanical Converter (TL)

Purpose

Ideal spring in mechanical translational systems

Library

Mechanical Translational Elements

Description

The Translational Spring block represents an ideal mechanical linear spring, described with the following equations:

$$F = Kx$$

$$x = x_{init} + x_R - x_C$$

$$v = \frac{dx}{dt}$$

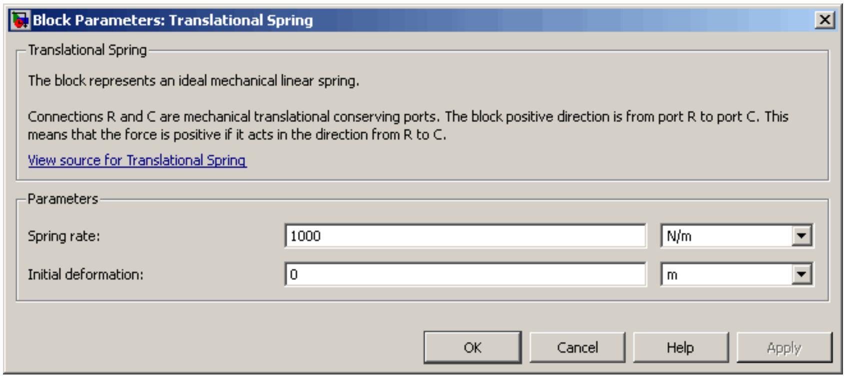
where

F	Force transmitted through the spring
K	Spring rate
x	Relative displacement (spring deformation)
x_{init}	Spring initial displacement (spring offset)
$x_{R,} x_C$	Absolute displacements of terminals R and C, respectively
v	Relative velocity
t	Time

The block positive direction is from port R to port C. This means that the force is positive if it acts in the direction from R to C.

Translational Spring

Dialog Box and Parameters



Block Parameters: Translational Spring

Translational Spring

The block represents an ideal mechanical linear spring.

Connections R and C are mechanical translational conserving ports. The block positive direction is from port R to port C. This means that the force is positive if it acts in the direction from R to C.

View source for Translational Spring

Parameters

Spring rate: 1000 N/m

Initial deformation: 0 m

Spring rate

Spring rate. The default value is `1000` N/m.

Initial deformation

Spring initial deformation, or offset, in length units. The deformation is determined as $x = x_{init} + x_R - x_C$, where x_{init} is the initial deformation, andx_R, x_C are the absolute displacements of terminals R and C in the globally assigned positive direction. The spring can be initially compressed ($x_{init} > 0$) or stretched ($x_{init} < 0$). This parameter specifies the initial condition for use in computing the block's initial state at the beginning of a simulation run. For more information, see "Initial Conditions Computation". The default value is `0`.

Ports

The block has the following ports:

`R`

Mechanical translational conserving port.

`C`

Mechanical translational conserving port.

See Also

Translational Damper

Translational Friction

Translational Hard Stop

Two-Way Connection

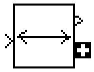
Purpose

Two-way connector port for subsystem

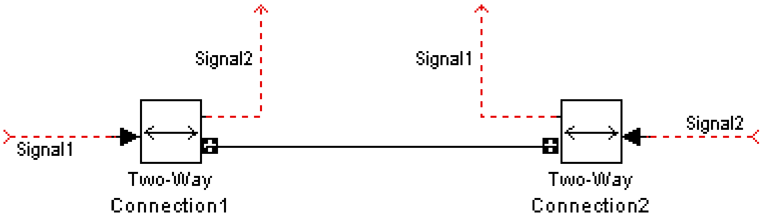
Library

Utilities

Description



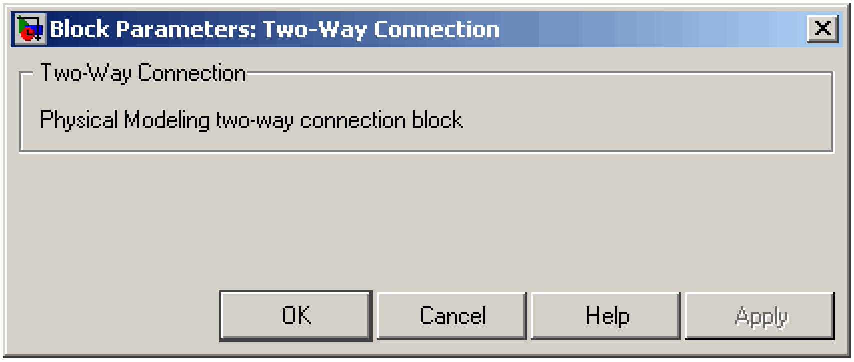
The Two-Way Connection block has a two-way connector port, which transports Simulink signals both ways. You connect this port to another two-way connector port. The schematic below illustrates how the two-way connection works. It carries the signal `Signal1` from the input port of the first Two-Way Connection block to the output port of the second Two-Way Connection block, and at the same time carries the signal `Signal2` from the input port of the second Two-Way Connection block to the output port of the first Two-Way Connection block.



The Two-Way Connection block supports invariant model architecture for top-down or bottom-up design. It lets you build subsystems as Simulink models, based on signals, and then connect them as if they are physical systems. Place the Two-Way Connection blocks inside the subsystem and connect them to the Connection Port blocks. Then the ports on the subsystem boundary appear as two-way connector ports ⊞.

Note Two-Way Connection blocks cannot be connected across nonvirtual subsystems.

Dialog Box and Parameters



Block Parameters: Two-Way Connection

Two-Way Connection

Physical Modeling two-way connection block

The block has no parameters.

Ports

The block has a Simulink input port, a Simulink output port, and a two-way connector port.

Uniform Random Number

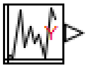
Purpose

Generate uniformly distributed random numbers for physical modeling

Library

Physical Signals/Sources

Description



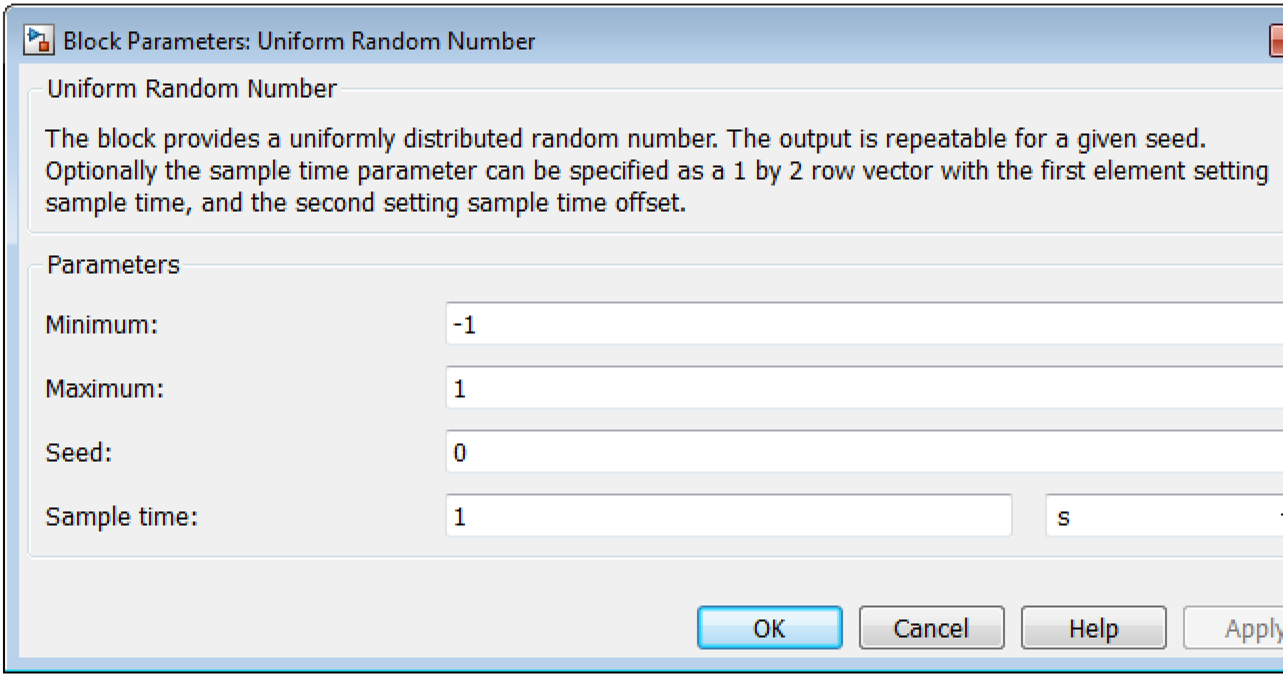
The Uniform Random Number block generates uniformly distributed random numbers over the interval you specify. To generate normally (Gaussian) distributed random numbers, use the Random Number block.

The block behavior is the same as the Simulink Uniform Random Number block (except that it generates a physical signal rather than a unitless Simulink signal).

You have an option to specify an initial time offset as part of the **Sample time** parameter. In this case, the block outputs 0 until the simulation time reaches the *`offset`* value, at which point the random sequence starts.

Uniform Random Number blocks that use the same seed and parameters generate a repeatable sequence. The seed resets to the specified value each time a simulation starts.

Dialog Box and Parameters



Block Parameters: Uniform Random Number

Uniform Random Number

The block provides a uniformly distributed random number. The output is repeatable for a given seed. Optionally the sample time parameter can be specified as a 1 by 2 row vector with the first element setting sample time, and the second setting sample time offset.

Parameters

Minimum: -1

Maximum: 1

Seed: 0

Sample time: 1 s

Minimum

Specify the minimum output value. The default is -1.

Maximum

Specify the maximum output value. The default is 1.

Seed

Specify the starting seed for the random number generator. Output is repeatable for a given seed. The seed must be an integer in the range of 0 to (2^32 – 1). The default is 0.

Sample time

The value of the time *`step`* interval. The default *`step`* value is `1` s. To specify an initial time offset, enter the parameter value as `[`*`step, offset`*`]`, otherwise the *`offset`* value is assumed to be `0`. The offset must be less than the step size.

Ports

The block has one physical signal output port.

See Also

Random Number

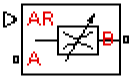
Purpose

Hydraulic variable orifice created by cylindrical spool and sleeve

Library

Hydraulic Elements

Description



The Variable Area Hydraulic Orifice block models a variable orifice created by a cylindrical sharp-edged spool and a variable-area slot in a sleeve. The area of the orifice is expected to be computed outside the block and imported via the AR physical signal connection. The **Minimum area** parameter specifies the minimum orifice area value. If the input signal falls below this level (for example, turns negative), the area is saturated to this value. The flow rate through the orifice is proportional to the orifice area and the pressure differential across the orifice.

The flow rate is determined according to the following equations:

$$q = \begin{cases} C_D \cdot A\sqrt{\frac{2}{\rho}|p|} \cdot sign(p) & \text{for } Re >= Re_{\mathrm{cr}} \\ 2C_{DL} \cdot A\frac{D_H}{\nu \cdot \rho}p & \text{for } Re < Re_{\mathrm{cr}} \end{cases}$$

$$p = p_A - p_B$$

$$\mathrm{Re} = \frac{q \cdot D_H}{A \cdot \nu}$$

$$C_{DL} = \left(\frac{C_D}{\sqrt{\mathrm{Re}_{cr}}}\right)^2$$

$$D_H = \sqrt{\frac{4A}{\pi}}$$

where

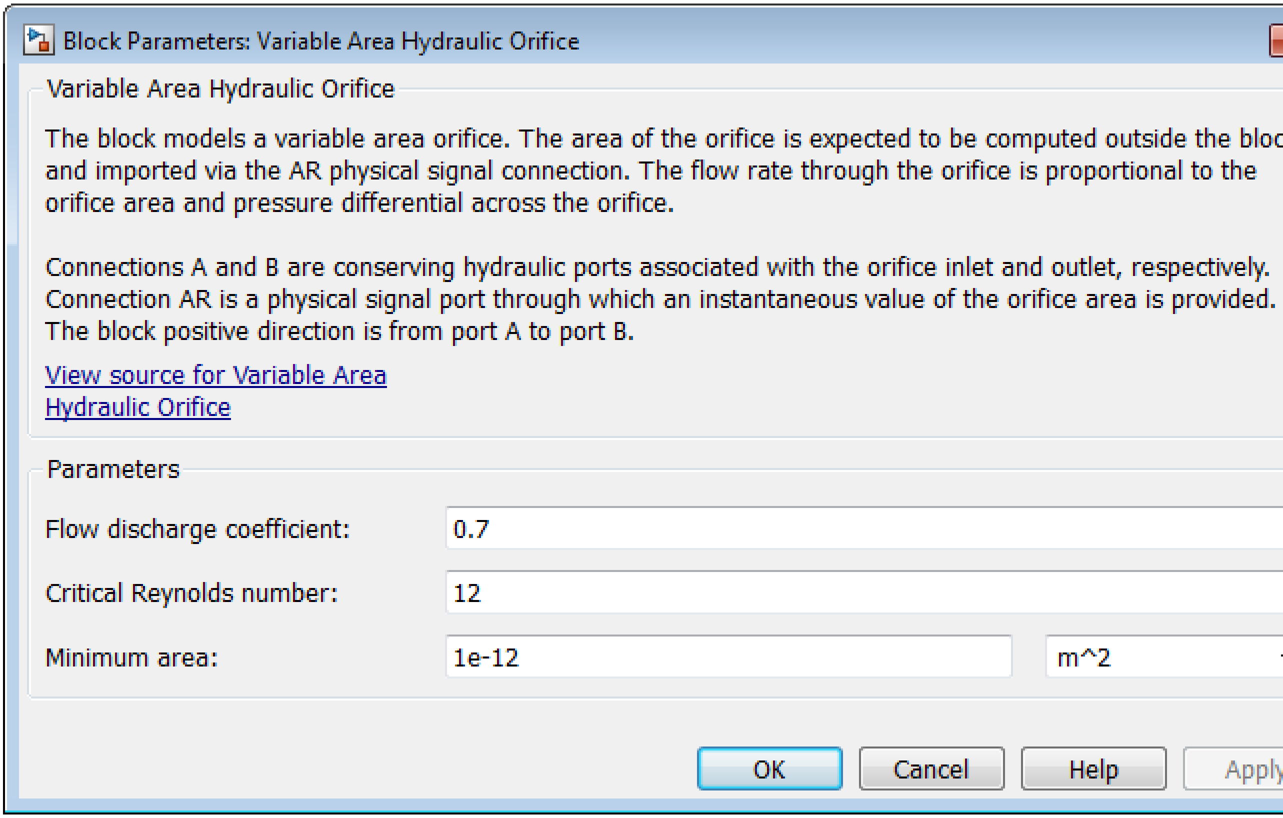
q	Flow rate
p	Pressure differential
$p_{A,}p_B$	Gauge pressures at the block terminals
C_D	Flow discharge coefficient
A	Orifice passage area, provided through the signal port
D_H	Orifice hydraulic diameter
ρ	Fluid density
ν	Fluid kinematic viscosity

The block positive direction is from port A to port B. This means that the flow rate is positive if it flows from A to B and the pressure differential is determined as $p = p_A - p_B$.

Basic Assumptions and Limitations

- Fluid inertia is not taken into account.
- The transition between laminar and turbulent regimes is assumed to be sharp and taking place exactly at Re=Re_{cr}.

Dialog Box and Parameters



Block Parameters: Variable Area Hydraulic Orifice

Variable Area Hydraulic Orifice

The block models a variable area orifice. The area of the orifice is expected to be computed outside the bloc and imported via the AR physical signal connection. The flow rate through the orifice is proportional to the orifice area and pressure differential across the orifice.

Connections A and B are conserving hydraulic ports associated with the orifice inlet and outlet, respectively. Connection AR is a physical signal port through which an instantaneous value of the orifice area is provided. The block positive direction is from port A to port B.

View source for Variable Area Hydraulic Orifice

Parameters

Flow discharge coefficient: 0.7

Critical Reynolds number: 12

Minimum area: 1e-12 m^2

Flow discharge coefficient

Semi-empirical parameter for orifice capacity characterization. Its value depends on the geometrical properties of the orifice, and usually is provided in textbooks or manufacturer data sheets. The default value is `0.7`.

Critical Reynolds number

The maximum Reynolds number for laminar flow. The transition from laminar to turbulent regime is supposed to take place when the Reynolds number reaches this value. The value of the parameter depends on orifice geometrical profile, and the recommendations on the parameter value can be found in hydraulic textbooks. The default value is `12`, which corresponds to a round orifice in thin material with sharp edges.

Minimum area

Leakage area of the completely closed orifice. If the input signal falls below this level (for example, turns negative), the area is saturated to this value. The parameter value must be greater than or equal to zero. The default value is `1e-12` m^2.

Global Parameters

Parameters determined by the type of working fluid:

- **Fluid density**
- **Fluid kinematic viscosity**

Use the Hydraulic Fluid block or the Custom Hydraulic Fluid block to specify the fluid properties.

Ports

The block has the following ports:

A

Hydraulic conserving port associated with the orifice inlet.

B

Hydraulic conserving port associated with the orifice outlet.

AR

Physical signal port that provides the value of the orifice area.

See Also

Constant Area Hydraulic Orifice

Variable Area Pneumatic Orifice

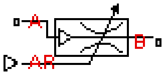
Purpose

Sharp-edged variable-area orifice in pneumatic systems

Library

Pneumatic Elements

Description



The Variable Area Pneumatic Orifice block models the flow rate of an ideal gas through a sharp-edged variable-area orifice. The area of the orifice is expected to be computed outside the block and imported via the AR physical signal connection. The **Minimum area** parameter specifies the minimum orifice area value. If the input signal falls below this level (for example, turns negative), the area is saturated to this value.

The flow rate through the orifice is proportional to the orifice area and the pressure differential across the orifice.

$$G = C_d \cdot A \cdot p_i \sqrt{\frac{2\gamma}{\gamma - 1} \cdot \frac{1}{RT_i}\left[\left(\frac{p_o}{p_i}\right)^{\frac{2}{\gamma}} - \left(\frac{p_o}{p_i}\right)^{\frac{\gamma+1}{\gamma}}\right]}$$

where

G Mass flow rate

C_d Discharge coefficient, to account for effective loss of area due to orifice shape

A Orifice cross-sectional area

p_i, p_o Absolute pressures at the orifice inlet and outlet, respectively. The inlet and outlet change depending on flow direction. For positive flow ($G > 0$), $p_i = p_A$, otherwise $p_i = p_B$.

γ The ratio of specific heats at constant pressure and constant volume, $c_p \ / \ c_v$

R Specific gas constant

T Absolute gas temperature

The choked flow occurs at the critical pressure ratio defined by

$$\beta_{cr} = \frac{p_o}{p_i} = \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}}$$

after which the flow rate depends on the inlet pressure only and is computed with the expression

$$G = C_d \cdot A \cdot p_i \sqrt{\frac{\gamma}{RT_i} \cdot \beta_{cr}{}^{\frac{\gamma+1}{\gamma}}}$$

The square root relationship has infinite gradient at zero flow, which can present numerical solver difficulties. Therefore, for very small pressure differences, defined by p_o / p_i > 0.999, the flow equation is replaced by a linear flow-pressure relationship

$$G = kC_d \cdot A \cdot T_i^{-0.5}\left(p_i - p_o\right)$$

where k is a constant such that the flow predicted for p_o / p_i is the same as that predicted by the original flow equation for p_o / p_i = 0.999.

The heat flow out of the orifice is assumed equal to the heat flow into the orifice, based on the following considerations:

- The orifice is square-edged or sharp-edged, and as such is characterized by an abrupt change of the downstream area. This means that practically all the dynamic pressure is lost in the expansion.
- The lost energy appears in the form of internal energy that rises the output temperature and makes it very close to the inlet temperature.

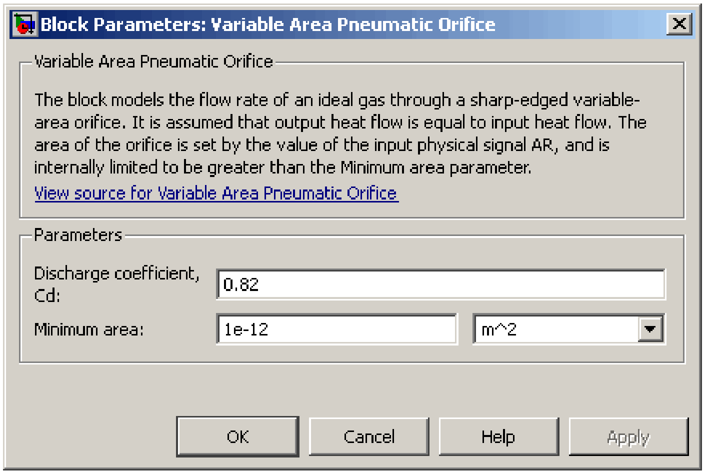
Therefore, $q_i = q_o$, where q_i and q_o are the input and output heat flows, respectively.

The block positive direction is from port A to port B. This means that the flow rate is positive if it flows from A to B.

Basic Assumptions and Limitations

- The gas is ideal.
- Specific heats at constant pressure and constant volume, c_p and c_v, are constant.
- The process is adiabatic, that is, there is no heat transfer with the environment.
- Gravitational effects can be neglected.
- The orifice adds no net heat to the flow.

Dialog Box and Parameters



Block Parameters: Variable Area Pneumatic Orifice

Variable Area Pneumatic Orifice

The block models the flow rate of an ideal gas through a sharp-edged variable-area orifice. It is assumed that output heat flow is equal to input heat flow. The area of the orifice is set by the value of the input physical signal AR, and is internally limited to be greater than the Minimum area parameter.

View source for Variable Area Pneumatic Orifice

Parameters

Discharge coefficient, Cd: 0.82

Minimum area: 1e-12 m^2

Discharge coefficient, Cd

Semi-empirical parameter for orifice capacity characterization. Its value depends on the geometrical properties of the orifice, and usually is provided in textbooks or manufacturer data sheets. The default value is `0.82`.

Minimum area

Specifies the minimum orifice area value. If the input signal falls below this level (for example, turns negative), the area is saturated to this value. The default value is `1e-12` m^2.

Ports

The block has the following ports:

A
: Pneumatic conserving port associated with the orifice inlet for positive flow.

B
: Pneumatic conserving port associated with the orifice outlet for positive flow.

AR
: Physical signal port that provides the value of the orifice area.

See Also

Constant Area Pneumatic Orifice

Constant Area Pneumatic Orifice (ISO 6358)

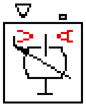
Purpose

Hydraulic capacity of variable volume with compressible fluid

Library

Hydraulic Elements

Description



The Variable Hydraulic Chamber block models fluid compressibility in variable volume chambers. The fluid is considered to be a mixture of liquid and a small amount of entrained, nondissolved gas. Use this block together with the Translational Hydro-Mechanical Converter block.

Note The Variable Hydraulic Chamber block takes into account only the flow rate caused by fluid compressibility. The fluid volume consumed to create piston velocity is accounted for in the Translational Hydro-Mechanical Converter block.

The chamber is simulated according to the following equations (see [1, 2]):

$$q = \frac{V_0 + V}{E} \cdot \frac{dp}{dt}$$

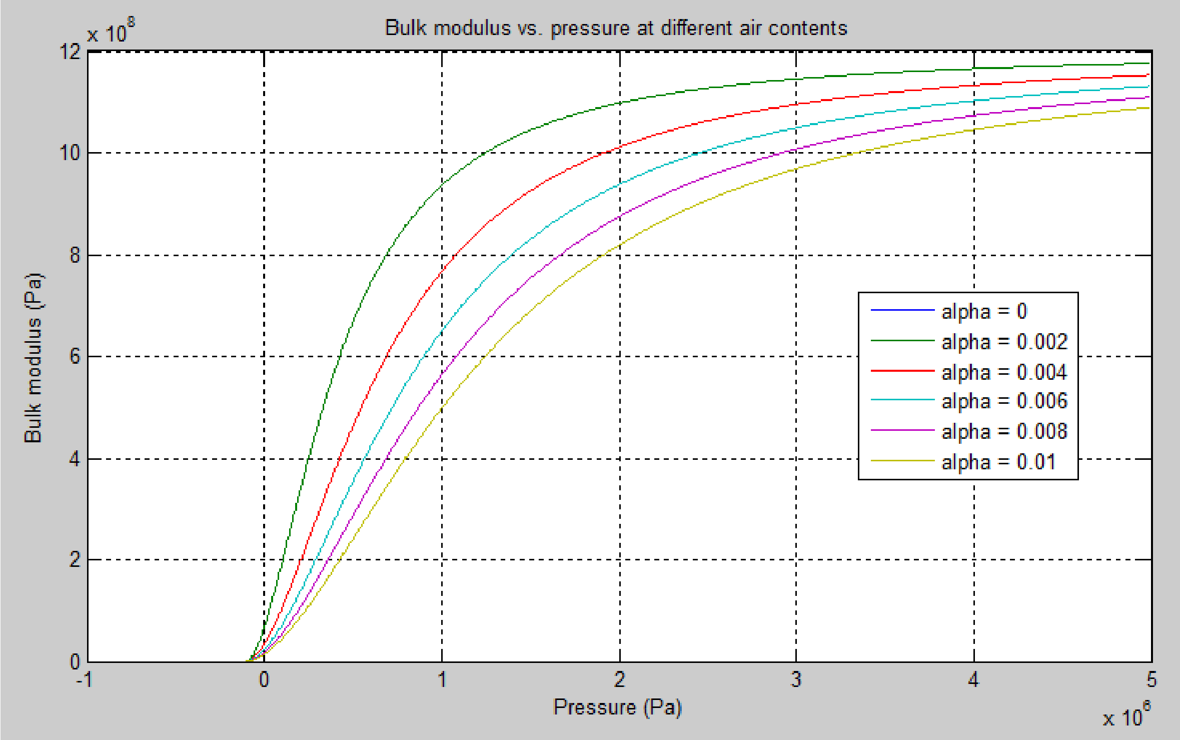
$$E = E_l \frac{1 + \alpha \left(\frac{p_a}{p_a + p} \right)^{1/n}}{1 + \alpha \frac{p_a^{1/n}}{n \cdot (p_a + p)^{\frac{n+1}{n}}} E_l}$$

where

q	Flow rate due to fluid compressibility
V_0	Initial volume of fluid in the chamber
V	Chamber volume change, provided through port V
E	Fluid bulk modulus

E_1	Pure liquid bulk modulus
p	Gauge pressure of fluid in the chamber
p_α	Atmospheric pressure
α	Relative gas content at atmospheric pressure, $\alpha = V_G/V_L$
V_G	Gas volume at atmospheric pressure
V_L	Volume of liquid
n	Gas-specific heat ratio

The main objective of representing fluid as a mixture of liquid and gas is to introduce an approximate model of cavitation, which takes place in a chamber if pressure drops below fluid vapor saturation level. As it is seen in the graph below, the bulk modulus of a mixture decreases at $p \to p_a$, thus considerably slowing down further pressure change. At high pressure, $p >> p_a$, a small amount of nondissolved gas has practically no effect on the system behavior.



Cavitation is an inherently thermodynamic process, requiring consideration of multiple-phase fluids, heat transfers, etc., and as such cannot be accurately simulated with Simscape software. But the simplified version implemented in the block is good enough to signal if pressure falls below dangerous level, and to prevent computation failure that normally occurs at negative pressures.

If pressure falls below absolute vacuum (–101325 Pa), the simulation stops and an error message is displayed.

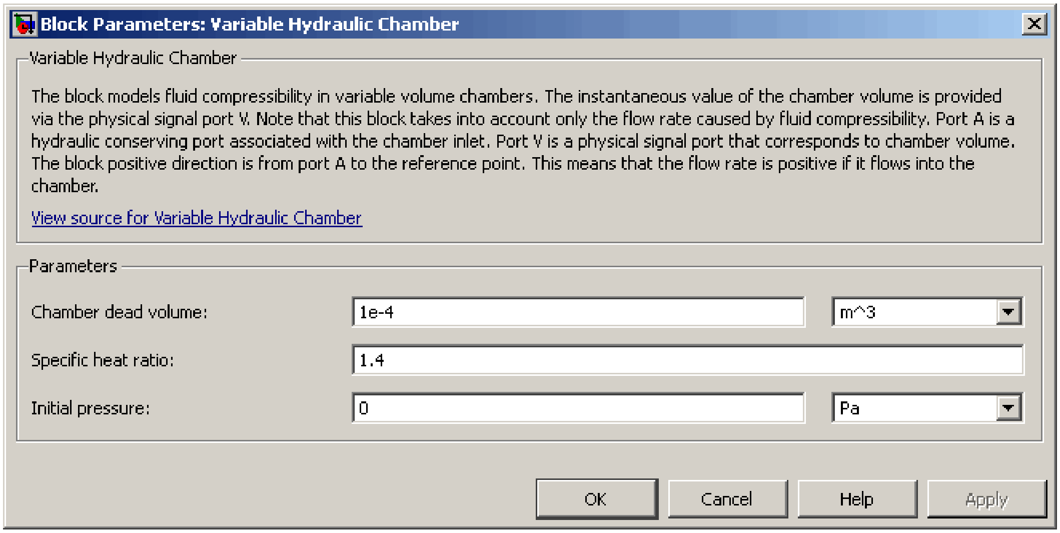
Port A is a hydraulic conserving port associated with the chamber inlet. Port V is a physical signal port that provides the chamber volume variation.

The block positive direction is from port A to the reference point. This means that the flow rate is positive if it flows into the chamber.

Basic Assumptions and Limitations

- Fluid density remains constant.
- Chamber volume can not be less that the dead volume.
- Fluid fills the entire chamber volume.

Dialog Box and Parameters



Block Parameters: Variable Hydraulic Chamber

Variable Hydraulic Chamber

The block models fluid compressibility in variable volume chambers. The instantaneous value of the chamber volume is provided via the physical signal port V. Note that this block takes into account only the flow rate caused by fluid compressibility. Port A is a hydraulic conserving port associated with the chamber inlet. Port V is a physical signal port that corresponds to chamber volume. The block positive direction is from port A to the reference point. This means that the flow rate is positive if it flows into the the chamber.

View source for Variable Hydraulic Chamber

Parameters

Chamber dead volume: 1e-4 m^3

Specific heat ratio: 1.4

Initial pressure: 0 Pa

Chamber dead volume
Minimal volume of fluid in the chamber. The default value is `1e-4` m^3.

Specific heat ratio
Gas-specific heat ratio. The default value is `1.4`.

Initial pressure

Initial pressure in the chamber. This parameter specifies the initial condition for use in computing the block's initial state at the beginning of a simulation run. For more information, see "Initial Conditions Computation". The default value is `0`.

Restricted Parameters

When your model is in Restricted editing mode, you cannot modify the following parameter:

- **Chamber orientation**

All other block parameters are available for modification.

Global Parameters

Parameters determined by the type of working fluid:

- **Fluid density**
- **Fluid kinematic viscosity**

Use the Hydraulic Fluid block or the Custom Hydraulic Fluid block to specify the fluid properties.

Ports

The block has the following ports:

A

Hydraulic conserving port associated with the chamber inlet.

V

Physical signal port that provides the chamber volume variation.

References

[1] Manring, N.D., *Hydraulic Control Systems*, John Wiley & Sons, New York, 2005

[2] Meritt, H.E., *Hydraulic Control Systems*, John Wiley & Sons, New York, 1967

See Also

Constant Volume Hydraulic Chamber

Hydraulic Piston Chamber

Variable Local Restriction (TL)

Purpose

Narrow opening with variable cross-sectional area

Library

Thermal Liquid/Elements

Description

The Variable Local Restriction (TL) block represents a narrow opening with variable cross-sectional area. The restriction causes a pressure drop and temperature gain in the liquid flowing through it. Common restrictions include valves and orifices.

To compute the pressure drop across the restriction, the block uses a discharge coefficient. This coefficient relates the pressure drop to the kinetic energy of the upstream liquid. The restriction is adiabatic. It does not exchange heat with the environment.

The block provides physical signal port AR so that you can specify the restriction cross-sectional area. This signal saturates at the value set in the **Restriction minimum area** parameter of the dialog box.

The liquid volume in the local restriction is assumed small. As a result, the dynamic compressibility and thermal inertia of the liquid are negligible. The block ignores both of these effects.

The following equations govern the behavior of the local restriction:

$$0 = \dot{m}_A + \dot{m}_B$$

where

A	Cross-sectional area of the pipes adjacent to the restriction
A_r	Cross-sectional area of the local restriction
C_d	Flow discharge coefficient
p_Int	Liquid pressure in the local restriction
$\dot{m}_A$, $\dot{m}_B$	Mass flow rates of liquid into the local restriction at inlets A and B
v_A, v_B	Liquid velocity into the local restriction at inlets A and B

Re	Reynolds number
Re_{c}	Critical Reynolds number
$_{\mathrm{u}}$	Upstream liquid density
μ_{u}	Upstream fluid dynamic viscosity
$\dot{}_{\mathrm{A}}$, $\dot{}_{\mathrm{B}}$	Thermal fluxes into the local restriction at inlets A and B

The liquid velocities at inlets A and B follow from the mass flow rates at those inlets:

$$v_A = \frac{\dot{m}_A}{A \cdot \rho_{A,u}}$$

$$v_B = \frac{\dot{m}_B}{A \cdot \rho_{B,u}}$$

where $_{\mathrm{A,u}}$ and $_{\mathrm{B,u}}$ are the liquid mass densities at inlets A and B.

The Reynolds number in the restriction satisfies the expression:

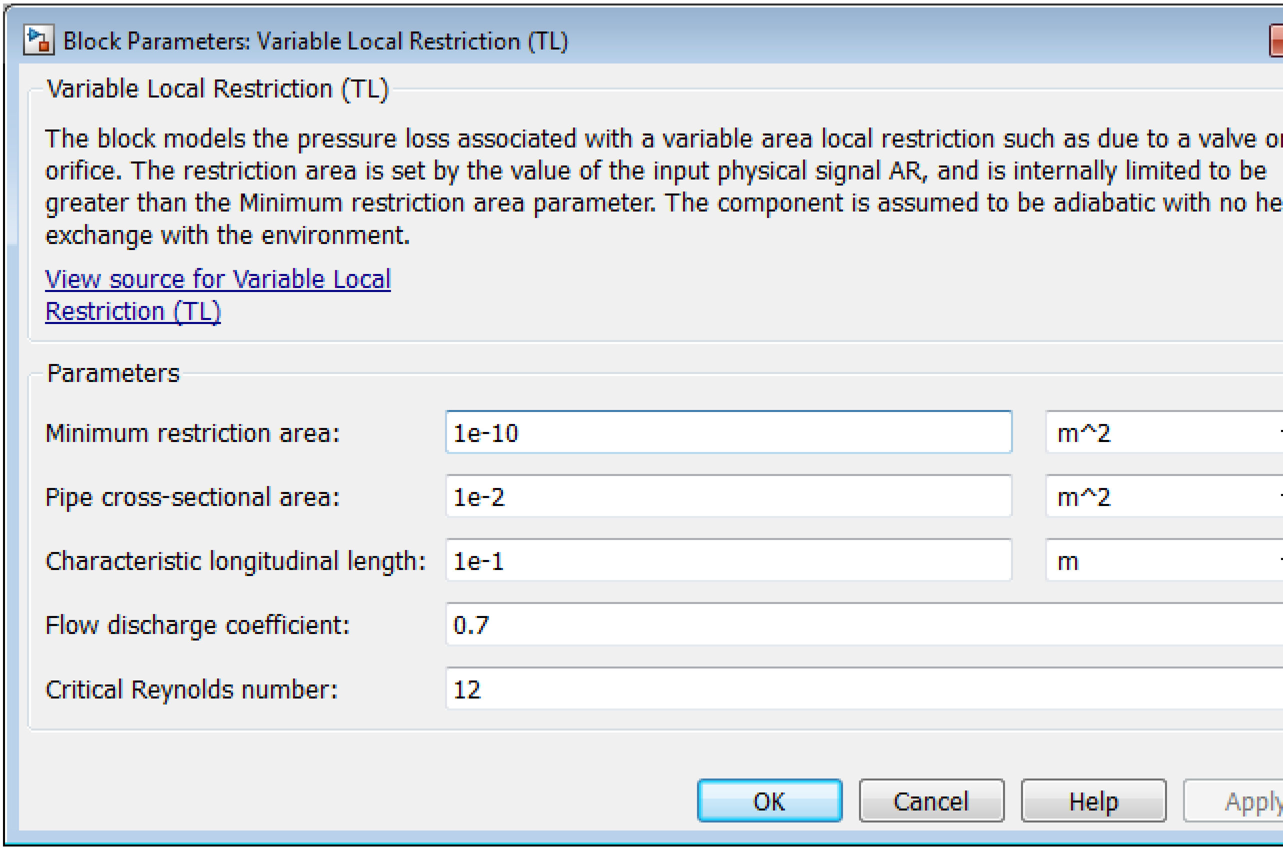
$$\mathrm{Re} = \frac{|\dot{m}_A|}{A_r \cdot \mu_u}$$

The block smooths the transition between laminar and turbulent flow regimes ($\mathrm{Re} \leq \mathrm{Re}_{\mathrm{c}}$ and $\mathrm{Re} \geq \mathrm{Re}_{\mathrm{c}}$, respectively). Smoothing occurs in a way that avoids zero-crossing events in both the flow regime transition and at zero flow.

Assumptions and Limitations

- Restriction is adiabatic. It exchanges no heat with the environment.
- Fluid dynamic compressibility and thermal inertia are negligible.

Dialog Box and Parameters



Block Parameters: Variable Local Restriction (TL)

Variable Local Restriction (TL)

The block models the pressure loss associated with a variable area local restriction such as due to a valve or orifice. The restriction area is set by the value of the input physical signal AR, and is internally limited to be greater than the Minimum restriction area parameter. The component is assumed to be adiabatic with no he exchange with the environment.

View source for Variable Local Restriction (TL)

Parameters

Minimum restriction area:	1e-10	m^2
Pipe cross-sectional area:	1e-2	m^2
Characteristic longitudinal length:	1e-1	m
Flow discharge coefficient:	0.7	
Critical Reynolds number:	12	

Minimum restriction area

Enter the smallest cross-sectional area for the local restriction. The restriction area physical signal saturates at this value. The default value is `1e-10` m^2 .

Pipe cross-sectional area
Enter the cross-sectional area of the adjoining pipes. The default value is `1e-2` m^2 .

Characteristic longitudinal length
Enter the restriction length along the flow direction. The default value is `1e-1` m.

Flow discharge coefficient
Enter the discharge coefficient associated with the minor loss of the restriction. The default value is `0.7`.

Critical Reynolds number
Enter the Reynolds number at which flow transitions from laminar to turbulent. The default value is `12`.

Ports

The block has two thermal liquid conserving ports, A and B, and one physical signal port, AR.

See Also

Local Restriction (TL)

Pipe (TL)

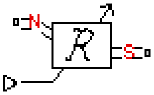
Purpose

Variable reluctance in electromagnetic systems

Library

Magnetic Elements

Description



The Variable Reluctance block models a variable reluctance, that is, a component that resists flux flow. The ratio of the magnetomotive force (mmf) across the component to the resulting flux that flows through the component is defined as the reluctance, and is dependent on the value of the input physical signal.

The block is based on the following equations:

$$\mathrm{F} = \Phi \cdot \Re$$

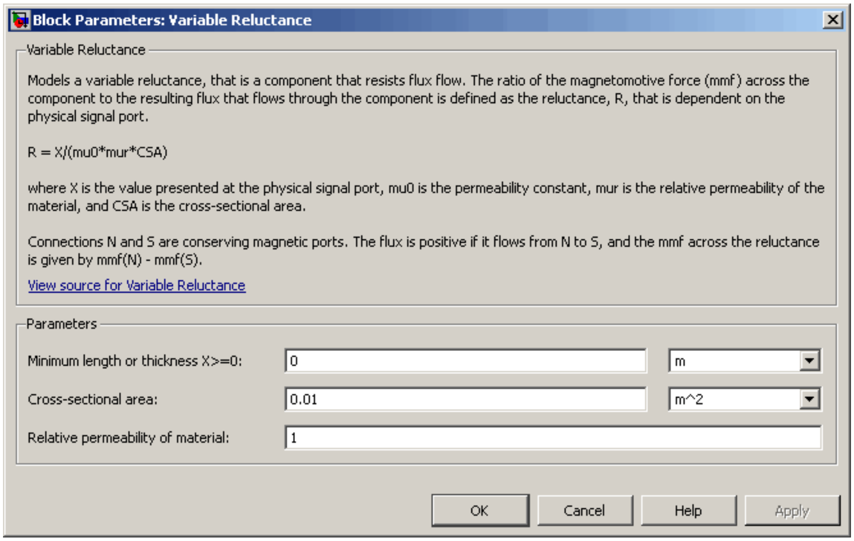
$$\Re = \frac{X}{\mu_0 \cdot \mu_r \cdot A}$$

where

F	Magnetomotive force (mmf) across the component
Φ	Flux through the component
$\Re$	Reluctance
X	Value presented at the physical signal port
μ_0	Permeability constant
μ_r	Relative permeability of the material
A	Cross-sectional area of the section being modeled

Connections N and S are magnetic conserving ports. The mmf across the reluctance is given by $\mathrm{F}(N) - \mathrm{F}(S)$, and the sign of the flux is positive when flowing through the device from N to S.

Dialog Box and Parameters



Block Parameters: Variable Reluctance

Variable Reluctance

Models a variable reluctance, that is a component that resists flux flow. The ratio of the magnetomotive force (mmf) across the component to the resulting flux that flows through the component is defined as the reluctance, R, that is dependent on the physical signal port.

R = X/(mu0*mur*CSA)

where X is the value presented at the physical signal port, mu0 is the permeability constant, mur is the relative permeability of the material, and CSA is the cross-sectional area.

Connections N and S are conserving magnetic ports. The flux is positive if it flows from N to S, and the mmf across the reluctance is given by mmf(N) - mmf(S).

View source for Variable Reluctance

Parameters

Minimum length or thickness X>=0: 0 m

Cross-sectional area: 0.01 m^2

Relative permeability of material: 1

Minimum length or thickness X>=0

The minimum value of length of air gap or thickness of section. If the input signal falls below this level (for example, turns negative), this minimum value is used. The parameter value must be nonnegative. The default value is 0.

Cross-sectional area

Area of the section being modeled. The default value is 0.01 m^2.

Relative permeability of material

Relative permeability of the section material. The default value is 1.

Ports

The block has the following ports:

N

Magnetic conserving port associated with the block North terminal.

S

Magnetic conserving port associated with the block South terminal.

The block also has one physical signal input port that provides the value of the length of air gap or thickness of section.

See Also

Reluctance

Variable Resistor

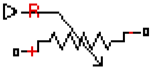
Purpose

Linear variable resistor in electrical systems

Library

Electrical Elements

Description



The Variable Resistor block models a linear variable resistor, described with the following equation:

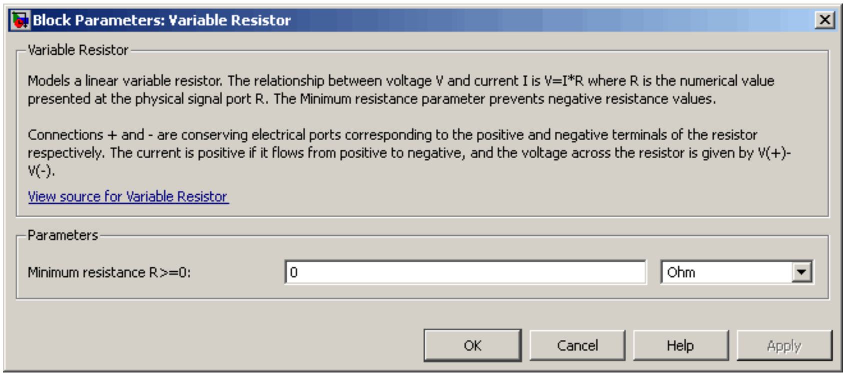
$$V = I□R$$

where

V	Voltage
I	Current
R	Resistance, i.e., the value presented at the control port

Connections + and – are conserving electrical ports corresponding to the positive and negative terminals of the resistor, respectively. P is a physical signal input port that controls the resistance value. The current is positive if it flows from positive to negative, and the voltage across the resistor is equal to the difference between the voltage at the positive and the negative terminal, $V(+) - V(-)$.

Dialog Box and Parameters



Variable Resistor

Models a linear variable resistor. The relationship between voltage V and current I is V=I*R where R is the numerical value presented at the physical signal port R. The Minimum resistance parameter prevents negative resistance values.

Connections + and - are conserving electrical ports corresponding to the positive and negative terminals of the resistor respectively. The current is positive if it flows from positive to negative, and the voltage across the resistor is given by V(+)-V(-).

View source for Variable Resistor

Parameters

Minimum resistance R>=0: 0 Ohm

Minimum resistance R>=0

The minimum resistance value. If the input signal falls below this level (for example, turns negative), this minimum resistance value is used. The parameter value must be nonnegative. The default value is `0`.

Ports

The block has the following ports:

`+`

Electrical conserving port associated with the resistor positive terminal.

`-`

Electrical conserving port associated with the resistor negative terminal.

`P`

Physical signal input port that provides the resistance value.

See Also

Resistor

Variable Volume Chamber

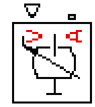
Purpose

Hydraulic capacity of variable volume with compressible fluid

Library

None (kept for compatibility purposes only)

Description



The Variable Volume Chamber block has been deprecated and removed from the library as of Version 3.0 (R2008b). Documentation is kept for compatibility reasons. If you use this block in your older models, it will still work. However, support may be discontinued in a future version. Replace this block with the Hydraulic Piston Chamber block.

See Also

Constant Volume Hydraulic Chamber

Hydraulic Piston Chamber

Translational Hydro-Mechanical Converter

Variable Hydraulic Chamber

Purpose

Linear voltage-controlled current source

Library

Electrical Sources

Description

The Voltage-Controlled Current Source block models a linear voltage-controlled current source, described with the following equation:

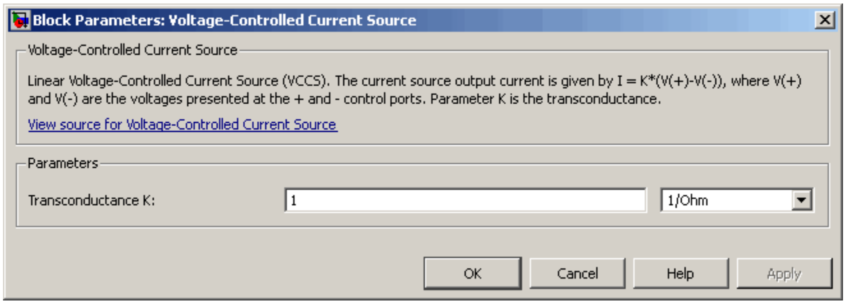
$$I = K\cdot(V(+) - V(-))$$

where

I	Current
K	Transconductance
V(+),*V(-)*	Voltages presented at the + and – control ports

To use the block, connect the + and – ports on the left side of the block (the control ports) to the control voltage source. The two ports on the right side of the block (the output ports) generate the output current. The arrow indicates the positive direction of the current flow.

Dialog Box and Parameters



Block Parameters: Voltage-Controlled Current Source

Voltage-Controlled Current Source

Linear Voltage-Controlled Current Source (VCCS). The current source output current is given by I = K*(V(+)-V(-)), where V(+) and V(-) are the voltages presented at the + and - control ports. Parameter K is the transconductance.

View source for Voltage-Controlled Current Source

Parameters

Transconductance K: 1 1/Ohm

Transconductance K

Transconductance, or the change in output current divided by the change in input voltage that causes it. The default value is 1 1/Ω.

Ports

The block has four electrical conserving ports. Connections + and – on the left side of the block are the control ports. The other two ports are the electrical terminals that provide the output current. The arrow indicates the positive direction of the current flow.

See Also

Current-Controlled Current Source

Current-Controlled Voltage Source

Voltage-Controlled Voltage Source

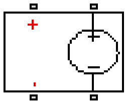
Purpose

Linear voltage-controlled voltage source

Library

Electrical Sources

Description



The Voltage-Controlled Voltage Source block models a linear voltage-controlled voltage source, described with the following equation:

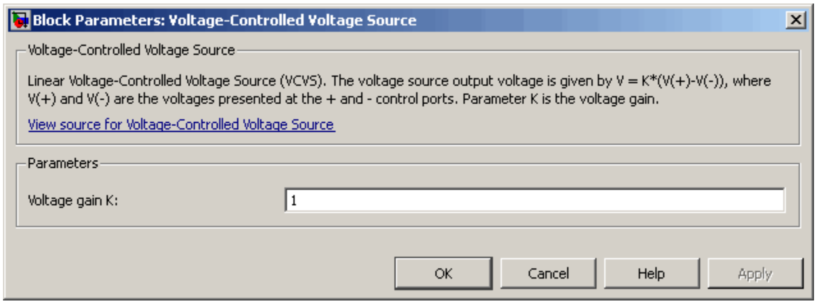
$$V = K \cdot (V(+) - V(-))$$

where

`V`	Output voltage
`K`	Voltage gain
`V(+),V(-)`	Voltages presented at the + and – control ports

To use the block, connect the + and – ports on the left side of the block (the control ports) to the control voltage source. The two ports on the right side of the block (the output ports) generate the output voltage. Polarity is indicated by the + and – signs.

Dialog Box and Parameters



Block Parameters: Voltage-Controlled Voltage Source

Voltage-Controlled Voltage Source

Linear Voltage-Controlled Voltage Source (VCVS). The voltage source output voltage is given by V = K*(V(+)-V(-)), where V(+) and V(-) are the voltages presented at the + and - control ports. Parameter K is the voltage gain.

View source for Voltage-Controlled Voltage Source

Parameters

Voltage gain K: 1

Voltage gain K

The change in the output voltage divided by the change in the control voltage that causes it. The default value is `1`.

Ports

The block has four electrical conserving ports. Connections + and – on the left side of the block are the control ports. The other two ports are the electrical terminals that provide the output voltage. Polarity is indicated by the + and – signs.

See Also

Current-Controlled Current Source

Current-Controlled Voltage Source

Voltage-Controlled Current Source

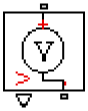
Purpose

Voltage sensor in electrical systems

Library

Electrical Sensors

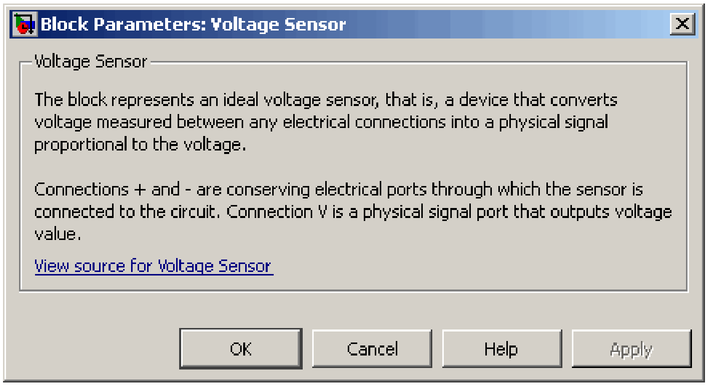
Description



The Voltage Sensor block represents an ideal voltage sensor, that is, a device that converts voltage measured between two points of an electrical circuit into a physical signal proportional to the voltage.

Connections + and – are electrical conserving ports through which the sensor is connected to the circuit. Connection V is a physical signal port that outputs the measurement result.

Dialog Box and Parameters



Block Parameters: Voltage Sensor

Voltage Sensor

The block represents an ideal voltage sensor, that is, a device that converts voltage measured between any electrical connections into a physical signal proportional to the voltage.

Connections + and - are conserving electrical ports through which the sensor is connected to the circuit. Connection V is a physical signal port that outputs voltage value.

View source for Voltage Sensor

The block has no parameters.

Ports

The block has the following ports:

+

Electrical conserving port associated with the sensor positive terminal.

-

Electrical conserving port associated with the sensor negative terminal.

`V`

Physical signal output port for voltage.

See Also

Current Sensor

Purpose

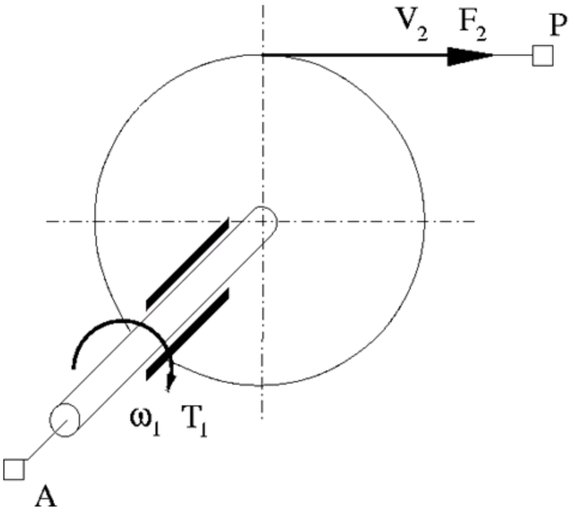
Wheel and axle mechanism in mechanical systems

Library

Mechanisms

Description

The Wheel and Axle block represents a wheel and axle mechanism shown in the following schematic.



The wheel and the axle have the same axis, and the axis is assumed to be rigidly connected to the frame, thus making this mechanism an ideal converter of mechanical rotational into mechanical translational motion. The mechanism has two connections: a mechanical rotational port A, which corresponds to the axle, and a mechanical translational port P, which corresponds to the wheel periphery. The mechanism is described with the following equations:

$$T = r \cdot F \cdot or$$

$$v = r \cdot \omega \cdot or$$

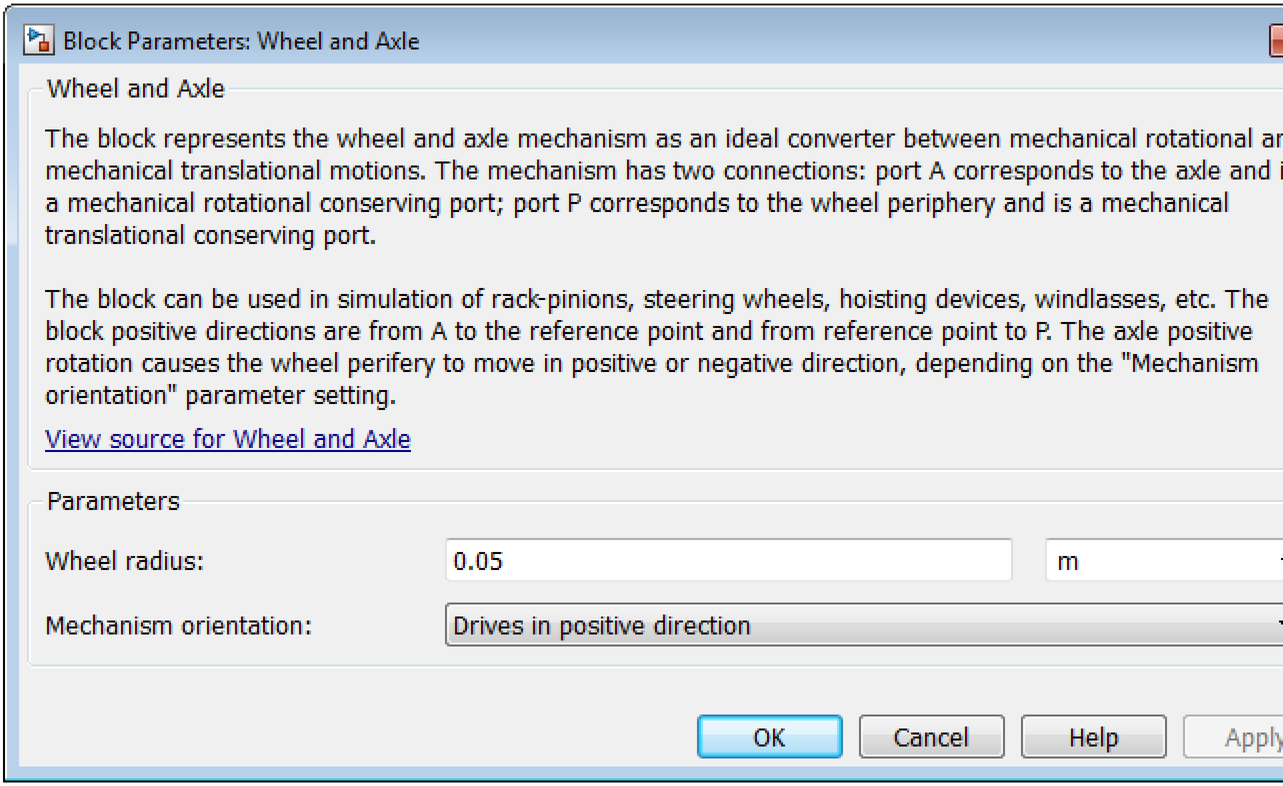
where

T	Torque on the axle
F	Force on the wheel periphery
ω	Angular velocity
v	Linear velocity on the wheel periphery
r	Wheel radius
or	Mechanism orientation indicator. The variable assumes +1 value if axle rotation in the globally assigned positive direction is converted into translational motion in positive direction, and –1 if positive rotation results in translational motion in negative direction.

The block can be used in simulation of rack-pinions, steering wheels, hoisting devices, windlasses, and so on.

The block positive directions are from A to the reference point and from the reference point to P.

Dialog Box and Parameters



Block Parameters: Wheel and Axle

Wheel and Axle

The block represents the wheel and axle mechanism as an ideal converter between mechanical rotational ar mechanical translational motions. The mechanism has two connections: port A corresponds to the axle and i a mechanical rotational conserving port; port P corresponds to the wheel periphery and is a mechanical translational conserving port.

The block can be used in simulation of rack-pinions, steering wheels, hoisting devices, windlasses, etc. The block positive directions are from A to the reference point and from reference point to P. The axle positive rotation causes the wheel perifery to move in positive or negative direction, depending on the "Mechanism orientation" parameter setting.

View source for Wheel and Axle

Parameters

Wheel radius: 0.05 m

Mechanism orientation: Drives in positive direction

Wheel radius

Radius of the wheel. The default value is `0.05` m.

Mechanism orientation

The parameter can be set to one of two options: `Drives in positive direction` or `Drives in negative direction`. The value `Drives in positive direction` specifies a mechanism where axle rotation in the globally assigned positive direction is converted into translational motion in positive direction. The value `Drives in negative direction` specifies a mechanism where axle rotation in the globally assigned positive direction is converted into translational motion in negative direction. The default value is `Drives in positive direction`.

Restricted Parameters

When your model is in Restricted editing mode, you cannot modify the following parameter:

- **Mechanism orientation**

All other block parameters are available for modification.

Ports

The block has the following ports:

`A`

Mechanical rotational conserving port associated with the axle.

`P`

Mechanical translational conserving port associated with the wheel periphery.

Examples

The Simple Mechanical System example illustrates the use of the Wheel and Axle block in mechanical systems.

See Also

Gear Box

2

Functions — Alphabetical List

pm_adddimension

Purpose

Adds new dimension to unit registry

Syntax

```
pm_adddimension(dimension,unitname)
```

Description

`pm_adddimension(dimension,unitname)` adds a new unit dimension with a fundamental unit, `unitname`.

Input Arguments

dimension - Name of dimension to add to the unit registry

string

Name of dimension to add to the unit registry, specified as a string. You can specify any string.

Data Types
`char`

unitname - Fundamental unit for new dimension

string

Fundamental unit used for the new dimension, specified as a string. The string must be a valid unit name: it must begin with a letter and contain only letters and numbers.

Data Types
`char`

Examples

Add Unit Dimension

Add a new unit dimension.

```
pm_adddimension('length','m');
```

The unit registry contains a new dimension, `length`, with a fundamental unit of meter, `m`.

See Also

`pm_addunit` | `pm_getdimensions` | `pm_getunits`

Concepts

- "Unit Definitions"

Purpose

Add new unit to unit registry

Syntax

```
pm_addunit(unitname, conversion, unitexpression)
```

Description

`pm_addunit(unitname, conversion, unitexpression)` introduces a new unit, `unitname`, defined as `conversion * unitexpression`.

The first argument, `unitname`, must be a valid unit name, that is, it must begin with a letter and contain only letters and numbers.

The second argument, `conversion`, may be either a positive real scalar or a 1x2 array. If this argument has two elements, then it is specifying an affine conversion, with the first element (a positive real number) being the linear conversion coefficient, and the second being the offset. For more information, see "Thermal Unit Conversions".

The third argument, `unitexpression`, must be a valid unit expression in terms of units already defined in the unit registry.

The following operators are supported in the unit mathematical expressions:

`*`	Multiplication
`/`	Division
`^`	Power
`+, -`	Plus, minus — for exponents only
`()`	Brackets to specify evaluation order

Examples

Add a new unit centimeter, `cm`, in terms of meter, `m`:

```
pm_addunit('cm', 0.01, 'm');
```

Add a new unit newton, `N`, in terms of kilograms, meters, and seconds:

```
pm_addunit('N', 1, 'kg*m/s^2');
```

Add a new unit Fahrenheit, Fh, in terms of Celsius:

```
pm_addunit('Fh', [5/9 -32*5/9], 'C');
```

See Also

pm_adddimension | pm_getdimensions | pm_getunits

Purpose

Get information about all dimensions in unit registry

Syntax

```
[dimensions, units] = pm_getdimensions
```

Description

`[dimensions, units] = pm_getdimensions` returns all dimensions registered in the unit registry in a cell array , `dimensions`. Their corresponding units are returned in the `units` cell array.

Examples

List all dimensions currently defined in the registry:

```
pm_getdimensions

ans =

    'charge'
    'length'
    'mass'
    'mole'
    'temperature'
    'time'
```

See Also

`pm_adddimension` | `pm_addunit` | `pm_getunits`

pm_getunits

Purpose

Get information about all units in unit registry

Syntax

```
[units, conversions, expressions] = pm_getunits
```

Description

`[units, conversions, expressions] = pm_getunits` returns all units in the registry in a cell array , `units`. Their corresponding conversions and base expressions are returned in `conversions` and `expressions`, respectively. For fundamental units, the conversion is `1.0` and the base expression is the unit itself.

Examples

List all units currently defined in the registry:

```
pm_getunits

ans =

    'm'
    'kg'
    's'
    'c'
    'K'
    'mol'
    'cm'
    'mm'
    'km'
    'um'
    'C'
    'Fh'
    'R'
    'in'
    'ft'
    'mi'
    'yd'
    'l'
    'gal'
    'igal'
    'g'
```

```
'mg'
'lbm'
'oz'
'slug'
'N'
'lbf'
'dyn'
'lb'
'mN'
'min'
'hr'
'ms'
'us'
'ns'
'rad'
'deg'
'rev'
'mph'
'fpm'
'fps'
'rpm'
'Hz'
'kHz'
'MHz'
'GHz'
'gee'
'J'
'Btu'
'eV'
'W'
'HP'
'V'
'A'
'F'
'H'
'Ohm'
'S'
```

```
    'Wb'
    'T'
    'G'
    'mV'
    'kV'
    'pA'
    'nA'
    'uA'
    'mA'
    'kA'
    'pF'
    'nF'
    'uF'
    'uH'
    'mH'
    'kOhm'
    'MOhm'
    'GOhm'
    'nS'
    'uS'
    'mS'
    'Pa'
    'bar'
    'psi'
    'atm'
    'lpm'
    'gpm'
    'Poise'
    'cP'
    'reyn'
    'St'
    'cSt'
    'Newt'
```

See Also

pm_adddimension | pm_addunit | pm_getdimensions

simscape.dependency.file

Purpose

Check dependencies for single file

Syntax

```
[fn_list, missing] = simscape.dependency.file('fileName')
[fn_list, missing] = simscape.dependency.file('fileName',
   dependencyType)
[fn_list, missing] = simscape.dependency.file('fileName',
   dependencyType, isRecursive)
[fn_list, missing] = simscape.dependency.file('fileName',
   dependencyType, isRecursive, doTMWFile)
```

Description

`[fn_list, missing] = simscape.dependency.file('fileName')` returns two cell arrays of strings: full path names of existing dependency files, `fn_list`, and missing files, `missing`. These cell arrays list the existing and missing files that are needed for the specified Simscape file to build successfully, or to correctly visualize and execute in MATLAB.

`[fn_list, missing] = simscape.dependency.file('fileName', dependencyType)` returns dependency files of the specified type.

`[fn_list, missing] = simscape.dependency.file('fileName', dependencyType, isRecursive)` lets you specify whether analysis is recursive on the generated dependency files. By default, returns only the top-level dependency files.

`[fn_list, missing] = simscape.dependency.file('fileName', dependencyType, isRecursive, doTMWFile)` lets you specify whether to include files inside the MATLAB root folder (installation directory) in the analysis.

Input Arguments

dependencyType

Enumerated value of type `Simscape.DependencyType`, which specifies the type of returned files:

`All` (default)	All the dependency files
`Auxiliary`	Files that are not necessary to convert the file and use it in block diagrams, but are needed to visualize it correctly, for example, block icon images
`Core`	Files necessary to convert the file and use it in block diagrams, for example, a domain file referenced by the component file being analyzed
`Derived`	Internally generated files that are not necessary for sharing the component file being analyzed, but including them will avoid rebuilding the library on the same platform.
`Simulink`	Additional files that help visualize the block generated from the component file being analyzed. These files are not necessary for simulation.

These enumerated values have the following order: `Core`, `Derived`, `Auxiliary`, `Simulink`, `All`. The return is accumulative. This means that for a requested file type, all earlier file types are also returned. For example, if you specify *`dependencyType`* as `Simscape.DependencyType.Derived`, the analysis returns both Core and Derived files.

doTMWFile

Logical value that indicates whether the file analysis includes files inside the MATLAB root folder (installation directory):

`true` (default)

```
false
```

fileName

The name of the Simscape file (with path), or class method, for which the dependencies are checked. In case of multiple files with the same name, only the first file of the specified name on the MATLAB path is analyzed.

isRecursive

Logical value that indicates whether the analysis is recursive on the generated dependency files:

`true`
`false` (default)

See Also

`simscape.dependency.lib` | `simscape.dependency.model`

How To

- "Checking File and Model Dependencies"

simscape.dependency.lib

Purpose

Check dependencies for library package

Syntax

```
[fn_list, missing] = simscape.dependency.lib('libName')
[fn_list, missing] = simscape.dependency.lib('libName',
   dependencyType)
[fn_list, missing] = simscape.dependency.lib('libName',
   dependencyType, 'mdlFileName')
[fn_list, missing] = simscape.dependency.lib('libName',
   dependencyType, 'mdlFileName', isRecursive)
[fn_list, missing] = simscape.dependency.lib('libName',
   dependencyType, 'mdlFileName', isRecursive, doTMWFile)
```

Description

`[fn_list, missing] = simscape.dependency.lib('libName')` returns two cell arrays of strings: full path names of existing dependency files, `fn_list`, and missing files, `missing`. These cell arrays list the existing and missing files that are needed for the specified Simscape library package to build successfully, or to correctly visualize and execute in MATLAB.

`[fn_list, missing] = simscape.dependency.lib('libName', dependencyType)` returns dependency files of the specified type.

`[fn_list, missing] = simscape.dependency.lib('libName', dependencyType, 'mdlFileName')` lets you specify the name of the library model. When not specified, or specified as an empty string (`''`), `libName_lib` is used.

`[fn_list, missing] = simscape.dependency.lib('libName', dependencyType, 'mdlFileName', isRecursive)` lets you specify whether analysis is recursive on the generated dependency files. By default, returns only the top-level dependency files.

`[fn_list, missing] = simscape.dependency.lib('libName', dependencyType, 'mdlFileName', isRecursive, doTMWFile)` lets you specify whether to include files inside the MATLAB root folder (installation directory) in the analysis.

If the package contains Simscape protected files, with the corresponding Simscape source files in the same folder, the analysis returns the names of protected files and then analyzes the source files for further dependencies. If the package contains Simscape protected files without the corresponding source files, the protected file names are returned without further analysis.

Input Arguments

dependencyType

Enumerated value of type `Simscape.DependencyType`, which specifies the type of returned files:

`All` (default)	All the dependency files
`Auxiliary`	Files that are not necessary to build the library, or run the models built from its blocks, but are needed to visualize it correctly, for example, block icon images or `lib.m` files.
`Core`	Files necessary to build the library or run the models built from its blocks, such as Simscape files or MATLAB files.
`Derived`	Internally generated files that are not necessary for sharing the library, but including them will avoid rebuilding the library on the same platform.
`Simulink`	Additional files that help visualize the blocks generated from the library components. These files are not necessary for simulation.

These enumerated values have the following order: `Core`, `Derived`, `Auxiliary`, `Simulink`, `All`. The return is accumulative. This means that for a requested file type, all earlier file types are also returned. For example, if you specify *`dependencyType`* as

Simscape.DependencyType.Derived, the analysis returns both Core and Derived files.

doTMWFile

Logical value that indicates whether the file analysis includes files inside the MATLAB root folder (installation directory):

true (default)
false

isRecursive

Logical value that indicates whether the analysis is recursive on the generated dependency files:

true
false (default)

libName

The name of a Simscape library package. The package folder name begins with a leading + character, whereas the argument to simscape.dependency.lib must omit the + character. You must run the command from the folder containing the top-level package, or from inside the package folder. In the latter case, you can omit the name of the library package if it is the only argument.

mdlFileName

The name of the library model (either without path, or with relative path, or with absolute path). The model file extension (.slx or .mdl) is optional.

Default: *libName*_lib

See Also

simscape.dependency.file | simscape.dependency.model

How To

- "Checking File and Model Dependencies"

Purpose

Check dependencies for model

Syntax

```
[fn_list, missing, reference2fnList,
   reference2missing] =
simscape.dependency.model('modelName')
```

Description

`[fn_list, missing, reference2fnList, reference2missing] = simscape.dependency.model('modelName')` checks dependencies for a model containing Simscape and Simulink blocks. *`modelName`* specifies the name of the model (either without path, or with relative path, or with absolute path). The model file extension (`.slx` or `.mdl`) is optional.

You must open the model first.

This command returns dependency information regarding Simscape files and blocks only. To perform a complete dependencies check for a model, use the Simulink Manifest Tools. For more information, see "Analyze Model Dependencies" in the *Simulink User's Guide*.

If during the analysis this command encounters a Simscape file located inside the MATLAB root folder, it returns the file name without performing any further analysis on this file, because all the dependent files in this case are part of standard MathWorks installation.

Output Arguments

fn_list

A cell array of strings containing the full paths of all existing files referenced by the model *`modelName`*.

missing

A cell array of strings containing the names of all files that are referenced by the model *`modelName`* but cannot be found.

reference2fnList

A list of structures, each of which includes a field `'names'` as a list of file names causing the reference, and a field `'type'` as the reference type for each file. Two reference types are used: `'Simscape component'` indicates reference from a model block. `'Simscape'` indicates reference from a file.

reference2missing

A list of structures, each of which includes a field `'names'` as a list of missing file names, and a field `'type'` as the reference type for each file. Two reference types are used: `'Simscape component'` indicates reference from a model block. `'Simscape'` indicates reference from a file.

See Also

`simscape.dependency.file` | `simscape.dependency.lib`

How To

- "Checking File and Model Dependencies"

Purpose

Plot logged simulation data for node or series

Syntax

```
h = simscape.logging.plot(obj,Name,Value)
```

Description

`h = simscape.logging.plot(obj,Name,Value)` plots the simulation series values along the *y*-axis, with time along the *x*-axis. `obj` is an object (or a homogeneous cell array of objects) of class `simscape.logging.Node` or `simscape.logging.Series`. If `obj` is a node, plots all nonempty series associated with the specified node and its children. You can filter data being plotted by using the name-value pair arguments. Depending on the type of `obj`, `h` is a structure (for a node) or a cell array (for a series) of handles to the resulting figures.

Tips

- Plotting simulation data for a high-level node and its children can generate a large number of plots. By default, the plots are not docked in the desktop, which results in a multitude of separate figure windows. To avoid this inconvenience, you can issue a command to make figures automatically dock in the desktop. For more information, see "Docking Figures Automatically" in the MATLAB documentation.

Input Arguments

obj

An object of class `simscape.logging.Node` or `simscape.logging.Series`. Can also be a homogeneous cell array of objects of either of these two classes. `obj` must include a full identifier path to the node or series, starting with the workspace log variable name.

The following table describes the resulting plots based on the type of the `obj` argument:

Scalar series object	Plots the simulation series values along the *y*-axis, with time along the *x*-axis.
Nonscalar series object	Plots each dimension of the series values on a different axis in the same figure window.

Cell array of series objects	Plots all series objects with commensurate units on the same axis (superimposed), and each dimension for a nonscalar series on a different axis in the same figure window. The input arguments are binned based on commensurate units. For each bin, all series objects with the same dimension as the first series object in that bin are plotted and others are ignored.
Node object	Plots all nonempty series associated with the node and its children (up to the level defined by the `depth`). If a node has multiple children, at level 1, that are simulation variable nodes, these children are plotted in the same figure window but on a different axis. Descendants at other levels are plotted in different figure windows. All dimensions of a nonscalar series are plotted on the same axis.
Cell array of node objects	Plots commensurate series superimposed on the same axis. Intended for use to compare simulation data from different runs. All entries of the cell array are required to be equivalent to each other, meaning that the node objects must have same hierarchy, and the series objects for each node must have the same dimensions and commensurate units.

Name-Value Pair Arguments

Specify optional comma-separated pairs of `Name,Value` arguments. `Name` is the argument name and `Value` is the corresponding value. `Name` must appear inside single quotes (`' '`). You can specify several name and value pair arguments in any order as `Name1,Value1,...,NameN,ValueN`.

'depth'

Plot data for children at *n* levels, where *n* is a nonnegative integer, for example:

0	No children; plot the nonempty series of the specified node only.
1	Plot the nonempty series of the specified node and its children.
2	Plot the nonempty series of the specified node, its children, and their children.

This argument is ignored if `obj` is a series.

> **Default:** Plots all descendants of the node object that have nonempty series.

'names'

Adds the plot legend. The number of elements must be same as the number of elements of `obj`.

> **Default:** No legend.

'time'

Plot data in the specified time range only. Provide a 1x2 vector `[`*`start_time end_time`*`]` to specify the time range. `[]` plots all data.

> **Default:** Plots all data.

'units'

Plot the series values in the specified units. This argument filters the data to plot only nodes and series that are commensurate with the specified unit. The value can be a single unit or a cell array of units. Unit names must appear inside single quotes (`''`).

> **Default:** Plots all data.

Output Arguments

h

A structure or a cell array of handles to the resulting plot figure windows, depending on the type of `obj`. If `obj` is a series, `h` is a cell array. If `obj` is a node, `h` is a structure with the same hierarchy as the object being plotted. For example, if a specific child is not plotted then that field in the output structure is empty.

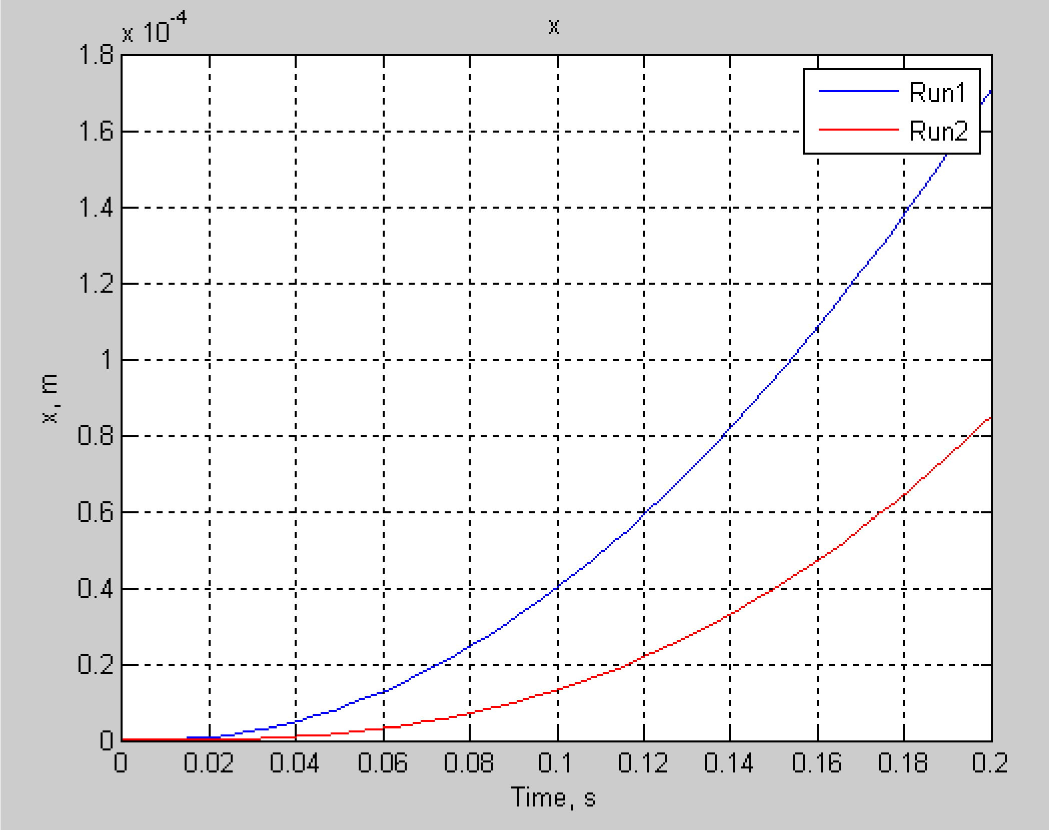
Examples

Plot all positions and velocities (series that are commensurate with units of `mm` and `mm/s`) in those units, respectively, for the top-level model node (with the default workspace variable name, `simlog`), its children and their children, within the time range between 1 and 3 seconds:

```
h = simscape.logging.plot(simlog, 'units', {'mm', 'mm/s'}, 'time', [1 3],
```

Compare data from two simulation runs. Supposing you use the workspace variable name `simlog1` to log the data from the first run, and the workspace variable name `simlog2` to log the data from the second run, the following command plots deformation of the Translational Spring block TS from both runs on the same axis, with the corresponding legend:

```
simscape.logging.plot({simlog1.TS.x simlog2.TS.x}, 'names', {'Run1' 'Run2
```



See Also

`simscape.logging.plotxy` | `simscape.logging.Node.plot` | `simscape.logging.Series.plot`

How To

- "Log and Plot Simulation Data"

simscape.logging.plotxy

Purpose

Plot logged simulation data for one node or series against another

Syntax

```
h = simscape.logging.plotxy(x,y,Name,Value)
```

Description

`h = simscape.logging.plotxy(x,y,Name,Value)` plots the simulation series values of object `y` along the *y*-axis, with series values of object `x` along the *x*-axis. `x` and `y` are objects (or homogeneous cell arrays of objects) of class `simscape.logging.Series` or `simscape.logging.Node`. If `x` or `y` is a node, it must be a simulation variable node (one that has a direct child series). The values of this child series are then plotted along the respective axis.

If `x` and `y` are cell arrays, they must be of the same size, or one of them can be a scalar. `x` and `y` must have the same time vectors. The remaining arguments are optional and provided as name-value pairs.

`h` is a cell array of figure handles, one for each `y` versus `x` plot generated.

Input Arguments

x

An object of class `simscape.logging.Series`. Can also be an object of class `simscape.logging.Node`, in which case it must be a simulation variable node (one that has a direct child series). The values of this series are plotted along the *x*-axis. Can also be a homogeneous cell array of objects of either of these two classes. `x` must include a full identifier path to the node or series, starting with the workspace log variable name.

y

An object of class `simscape.logging.Series`. Can also be an object of class `simscape.logging.Node`, in which case it must be a simulation variable node (one that has a direct child series). The values of this series are plotted along the *y*-axis. Can also be a homogeneous cell array of objects of either of these two classes. `y` must include a full identifier path to the node or series, starting with the workspace log variable name. If `x` and `y` are cell arrays, they must be of the same size, or one of them can be a scalar. `x` and `y` must have the same time vectors.

Name-Value Pair Arguments

Specify optional comma-separated pairs of `Name,Value` arguments. `Name` is the argument name and `Value` is the corresponding value. `Name` must appear inside single quotes (`' '`). You can specify several name and value pair arguments in any order as `Name1,Value1,...,NameN,ValueN`.

'time'

Plot data in the specified time range only. Provide a 1x2 vector `[`*`start_time end_time`*`]` to specify the time range. `[]` plots all data.

Default: Plots all data.

'xname'

Adds the *x*-axis name to the plot. Must be either a scalar or a cell array of the same size as `x`. Axis names must appear inside single quotes (`''`).

Default: Variable name.

'yname'

Adds the *y*-axis name to the plot. Must be either a scalar or a cell array of the same size as `y`. Axis names must appear inside single quotes (`''`).

Default: Variable name.

'xunit'

Plot the series values along the *x*-axis in the specified units. The specified unit must be commensurate with the unit of the series values. Unit name must appear inside single quotes (`''`).

Default: Default unit of the series values.

'yunit'

Plot the series values along the *y*-axis in the specified units. The specified unit must be commensurate with the unit of the series values. Unit name must appear inside single quotes ('').

Default: Default unit of the series values.

Output Arguments

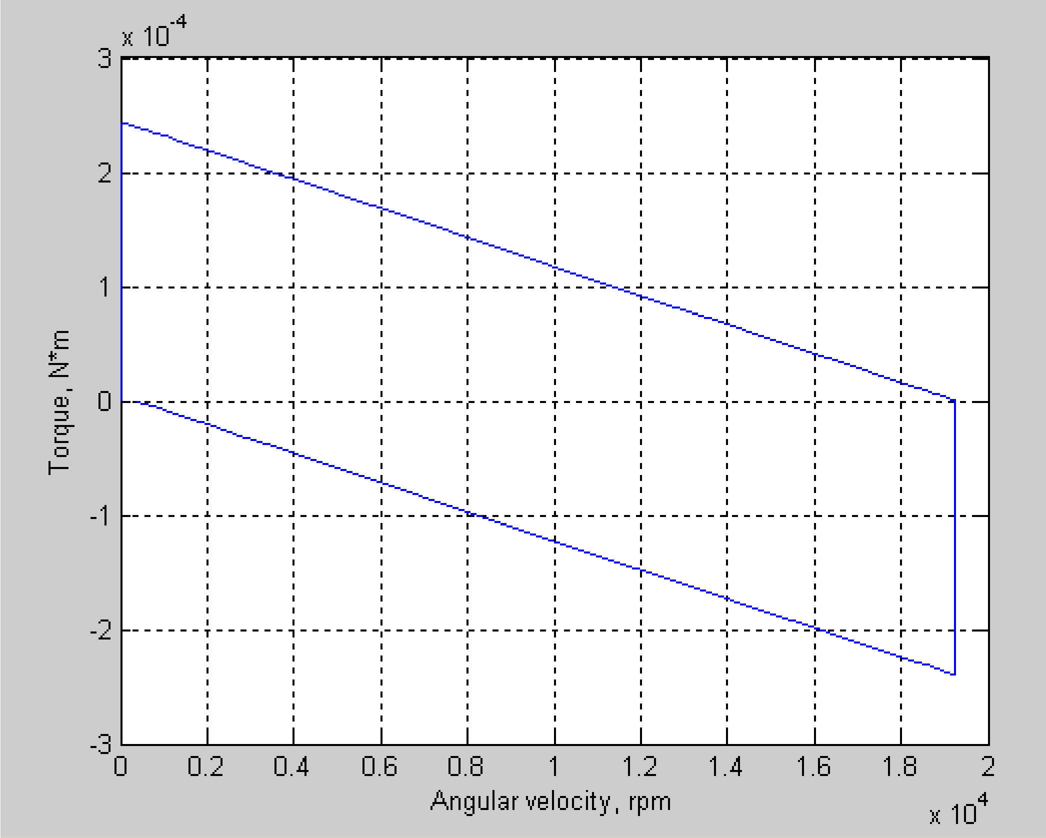
h

A cell array of handles to the resulting plot figure windows.

Examples

Plot the motor torque, in default units, against its angular velocity, in `rpm`, and add axis names:

```
plotxy(simlog.Rotational_Electromechanical_Converter.R.w, simlog.Motor_In
   'xunit', 'rpm', 'xname', 'Angular velocity', 'yname', 'Torque')
```



See Also

`simscape.logging.plot` | `simscape.logging.Node.plotxy` | `simscape.logging.Series.plotxy`

How To

- "Log and Plot Simulation Data"

simscape.logging.Node

Purpose

Represent hierarchy tree for simulation data

Description

This class represents the hierarchy of nodes for logging simulation data in a model. The tree starts with the workspace variable, which represents simulation data for the whole model, and recursively creates nodes for each of the children. The children are defined depending on the type of the parent node:

- For the top-level simulation log workspace variable, the children are all the Simscape blocks (and subsystems containing Simscape blocks) in the top-level model diagram.
- For a subsystem or a structural block, the children are all the constituent Simscape blocks and subsystems.
- For a block, the children are all its physical ports, Through and Across variables, and all internal variables defined in the block's Simscape file.
- For a physical port, the children are all its Across variables.

Final nodes in this recursion correspond to all the variables logged for the model. Final nodes do not have children nodes, and contain the series data logged during simulation.

You cannot construct an object of this class. The object is constructed automatically during simulation, as part of the simulation log workspace variable, if you enable data logging for the model.

Properties

id

The string identifying the `Node` object. For the simulation log workspace variable, this is the name of the top-level block diagram. For blocks and subsystems, the `id` is constructed automatically as a valid MATLAB identifier based on the name of the block or subsystem. For other types, the `id` is the name of the corresponding port or variable.

series

For `Node` objects that do not have children nodes, and therefore correspond to the logged variables, the `series` property returns an object of the `simscape.logging.Series` class that contains the simulation series data for this variable. For nodes that do not represent variables, the `series` property is hidden. If you access the hidden `series` property for such node, the property returns an object of the `simscape.logging.Series` class representing an empty series (with zero points).

The other properties are dynamic, and represent all the children of the `Node` object.

Methods

plot	Plot all series associated with node object
plotxy	Plot series associated with two node objects against each other
print	Print complete logging tree of node object

Copy Semantics

Handle. To learn how handle classes affect copy operations, see Copying Objects in the MATLAB Programming Fundamentals documentation.

See Also

`simscape.logging.Series`

Tutorials

- "Log and Plot Simulation Data"

How To

- "How to Log Simulation Data"

simscape.logging.Node.plot

Purpose

Plot all series associated with node object

Syntax

```
h = plot(node,Name,Value)
```

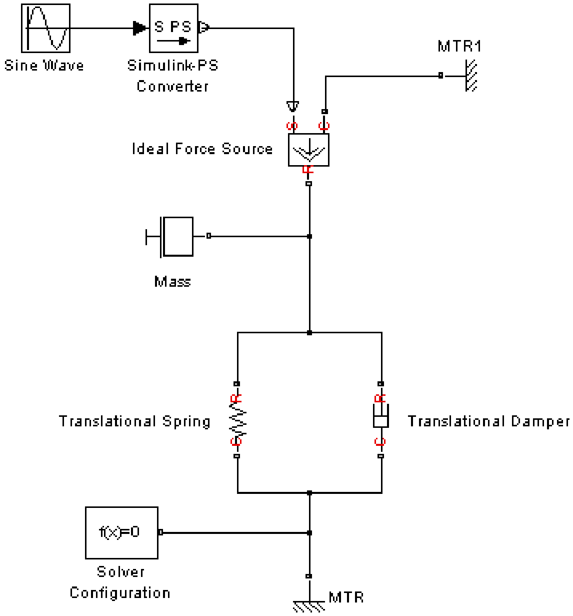
Description

`h = plot(node,Name,Value)` plots all nonempty series associated with the specified node and its children. You can filter data being plotted by using the name-value pair arguments. `h` is a structure of handles to the resulting figures. `node` is an object of class `simscape.logging.Node`. `node` must include a full identifier path to the node, starting with the workspace log variable name.

For more information, including the descriptions of name-value pair arguments, see the `simscape.logging.plot` reference page.

Examples

Consider the following model. The model name is `simple_mech2`, and data logging is enabled with the default workspace variable name, `simlog`.



The following is a complete logging tree for the model:

```
simlog.print
     simple_mech2
     +-Ideal_Force_Source
     | +-C
     | | +-v
     | +-R
     | | +-v
     | +-S
     | +-f
     | +-v
     +-MTR
     | +-V
     | | +-v
     | +-f
```

```
+-MTR1
| +-V
| | +-v
| +-f
+-Mass
| +-M
| | +-v
| +-f
+-Simulink_PS_Converter
+-Translational_Damper
| +-C
| | +-v
| +-R
| | +-v
| +-f
| +-v
+-Translational_Spring
  +-C
  | +-v
  +-R
  | +-v
  +-f
  +-v
  +-x
```

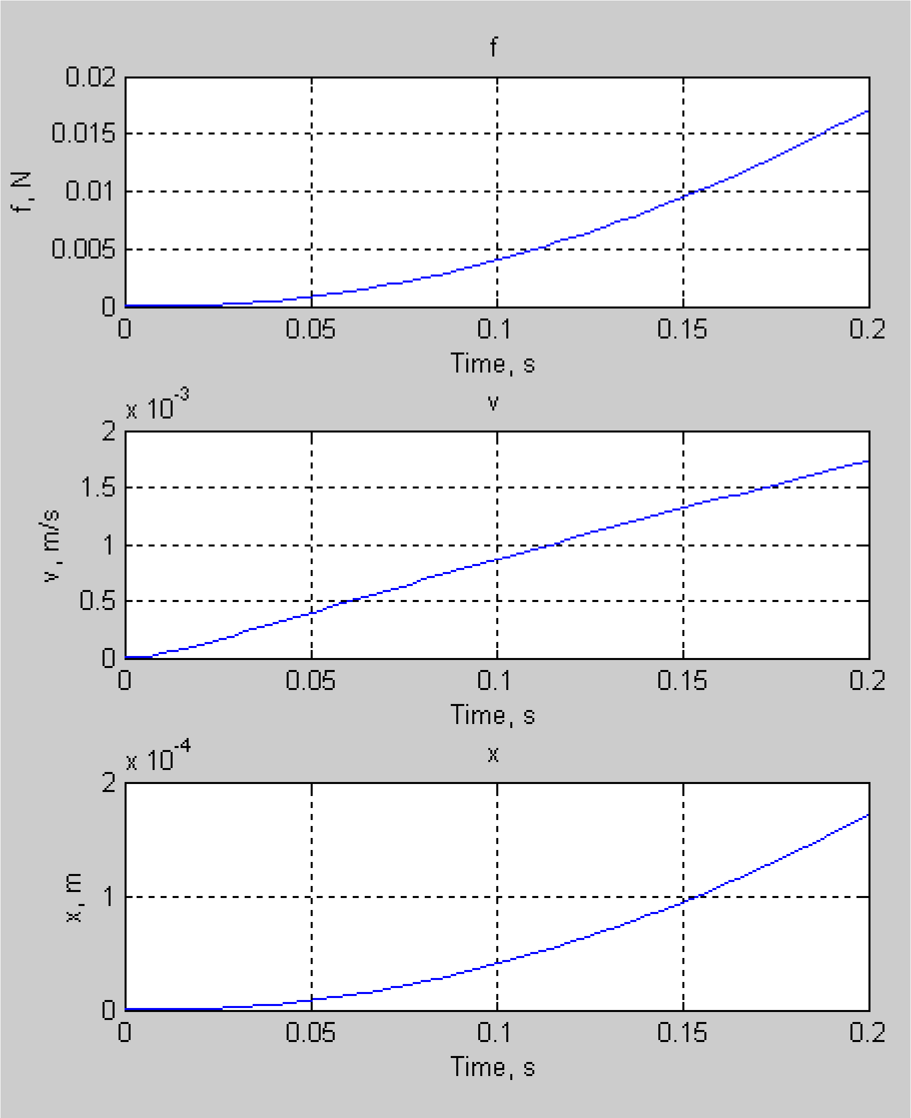
Plot velocities of all the blocks in the model:

```
plot(simlog, 'units', 'm/s', 'depth', 2)
```

This command filters simulation data in two ways. It plots only series that are commensurate with units `m/s` (that is, velocities), based on the `units` argument. And because of the `depth` argument, it plots only those velocity variables that are associated with the block itself. If you refer to the logging tree, only the Ideal Force Source, Translational Damper, and Translational Spring blocks have a velocity (`v`) variable at the second level. Because of the `depth` argument, velocities of the block ports (one level down) do not get plotted.

The next command plots all the variables associated with the Translational Spring block, but not with its ports:

```
plot(simlog.Translational_Spring, 'depth', 1)
```


f
f , N
0.02
0.015
0.01
0.005
0
0
0.05
0.1
0.15
0.2
Time, s
x 10^-3
v
v , m/s
2
1.5
1
0.5
0
0
0.05
0.1
0.15
0.2
Time, s
x 10^-4
x
x , m
2
1
0
0
0.05
0.1
0.15
0.2
Time, s

Alternatives

Use the `simscape.logging.plot` function.

See Also

`simscape.logging.Node` | `simscape.logging.Node.plotxy`

Tutorials

- "Log and Plot Simulation Data"

simscape.logging.Node.plotxy

Purpose

Plot series associated with two node objects against each other

Syntax

```
h = plotxy(x,y,Name,Value)
```

Description

`h = plotxy(x,y,Name,Value)` plots the simulation series values of node `y` along the *y*-axis, with series values of node `x` along the *x*-axis. `h` is a cell array of handles to the resulting figures. Arguments `x` and `y` are objects (`y` can be a cell array of objects) of class `simscape.logging.Node`. Each object must be a simulation variable node (one that has a direct child series). The values of this child series are plotted along the respective axis. All series must have the same time vectors.

Each object name must include a full identifier path to the node, starting with the workspace log variable name. The remaining arguments are optional and provided as name-value pairs.

For more information, including the descriptions of name-value pair arguments, see the `simscape.logging.plotxy` reference page.

Examples

Plot velocities of ports C and R of the Translational Spring block TS against each other, in `mm/s`:

```
plotxy(simlog.TS.C.v, simlog.TS.R.v, 'xunits', 'mm/s', 'yunits', 'mm/s')
```

Alternatives

Use the `simscape.logging.plotxy` function.

See Also

`simscape.logging.Node` | `simscape.logging.Node.plot`

Tutorials

- “Log and Plot Simulation Data”

Purpose

Print complete logging tree of node object

Syntax

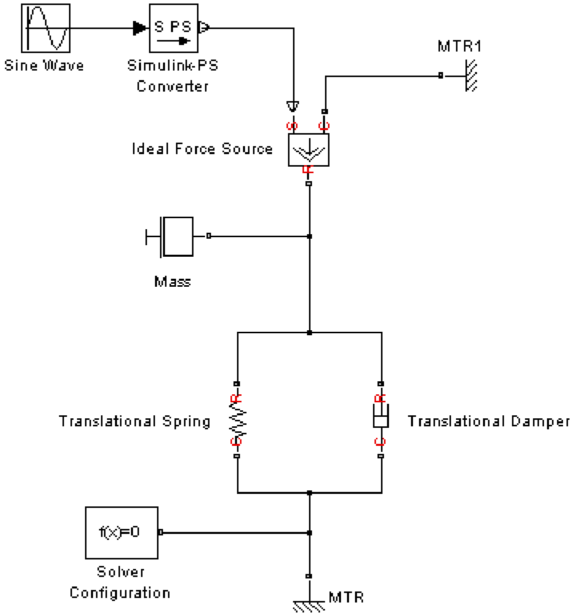
`*path_to_node*.print`

Description

`*path_to_node*.print` prints the complete logging tree starting with the specified node. `*path_to_node*` is a full identifier path to the node, starting with the workspace log variable name.

Examples

Consider the following model. The model name is `simple_mech2`, and data logging is enabled with the default workspace variable name, `simlog`.



Return the complete logging tree for the whole model:

```
simlog.print
    simple_mech2
```

```
+-Ideal_Force_Source
| +-C
| | +-v
| +-R
| | +-v
| +-S
| +-f
| +-v
+-MTR
| +-V
| | +-v
| +-f
+-MTR1
| +-V
| | +-v
| +-f
+-Mass
| +-M
| | +-v
| +-f
+-Simulink_PS_Converter
+-Translational_Damper
| +-C
| | +-v
| +-R
| | +-v
| +-f
| +-v
+-Translational_Spring
  +-C
  | +-v
  +-R
  | +-v
  +-f
  +-v
  +-x
```

Print the logging tree just for the Mass block:

```
simlog.Mass.print
     Mass
     +-M
     | +-v
     +-f
```

See Also

`simscape.logging.Node`

Tutorials

- "Log and Plot Simulation Data"

simscape.logging.Series

Purpose

Represent time-value series for simulation data

Description

This class represents simulation data for a variable in a model. The series is a representation containing time-value pairs for each simulation step. The size of the series is determined by the number of simulation steps. You can also limit the size by specifying the maximum number of logged steps when you set your data logging preferences.

You cannot construct an object of this class. The object is constructed automatically during simulation, as part of the simulation log workspace variable, if you enable data logging for the model.

Properties

points

Size or number of steps in the simulation series.

dimension

Dimension of variable represented by the series.

unit

The default unit associated with the values in the series.

Methods

plot	Plot series values against time
plotxy	Plot two series against each other
time	Extract time vector from simulation series
values	Extract values vector from simulation series

Copy Semantics

Handle. To learn how handle classes affect copy operations, see Copying Objects in the MATLAB Programming Fundamentals documentation.

See Also

`simscape.logging.Node`

Tutorials

- “Log and Plot Simulation Data”

How To

- “How to Log Simulation Data”

simscape.logging.Series.plot

Purpose

Plot series values against time

Syntax

```
h = plot(series,Name,Value)
```

Description

`h = plot(series,Name,Value)` plots the simulation series values along the *y*-axis, with time along the *x*-axis. You can filter data being plotted by using the name-value pair arguments. `h` is a cell array of handles to the resulting figures. `series` is an object of class `simscape.logging.Series`. `series` must include a full identifier path to the series, starting with the workspace log variable name.

For more information, including the descriptions of name-value pair arguments, see the `simscape.logging.plot` reference page.

Examples

Plot velocity of port R of the Translational Spring block, in `mm/s`:

```
plot(simlog.Translational_Spring.R.v.series, 'units', 'mm/s')
```

Alternatives

Use the `simscape.logging.plot` function.

See Also

`simscape.logging.Series` | `simscape.logging.Series.plotxy`

Tutorials

- "Log and Plot Simulation Data"

simscape.logging.Series.plotxy

Purpose

Plot two series against each other

Syntax

```
h = plotxy(x,y,Name,Value)
```

Description

`h = plotxy(x,y,Name,Value)` plots values of the simulation series `y` along the *y*-axis, with values of the simulation series `x` along the *x*-axis. `h` is a cell array of handles to the resulting figures. Arguments `x` and `y` are objects (`y` can be a cell array of objects) of class `simscape.logging.Series`. `Each object name` must include a full identifier path to the series, starting with the workspace log variable name. The series must have the same time vectors. The remaining arguments are optional and provided as name-value pairs.

For more information, including the descriptions of name-value pair arguments, see the `simscape.logging.plotxy` reference page.

Examples

Plot velocities of ports C and R of the Translational Spring block TS against each other, in `mm/s`:

```
plotxy(simlog.TS.C.v.series, simlog.TS.R.v.series, 'xunits', 'mm/s',
```

Alternatives

Use the `simscape.logging.plotxy` function.

See Also

`simscape.logging.Series` | `simscape.logging.Series.plot`

Tutorials

- “Log and Plot Simulation Data”

simscape.logging.Series.time

Purpose

Extract time vector from simulation series

Syntax

```
ta = path_to_var.series.time
```

Description

ta = *path_to_var*.series.time returns a row vector of simulation times contained in the series. *path_to_var* is a full identifier path to the variable node associated with the series.

Examples

Return simulation time data for the deformation of a Translational Spring block, located at the top level of the model diagram:

```
t1 = simlog.Translational_Spring.x.series.time

t1 =

         0
    0.0020
    0.0040
    0.0060
    0.0100
    0.0140
    0.0180
    0.0220
    0.0260
    0.0300
    0.0340
    0.0380
    0.0420
    0.0460
    0.0500
    0.0540
    0.0580
    0.0620
    0.0660
    0.0700
    0.0740
    0.0780
    0.0820
    0.0860
    0.0900
    0.0940
    0.0980
    0.1020
    0.1060
    0.1100
    0.1140
    0.1180
    0.1220
    0.1260
    0.1300
    0.1340
    0.1380
    0.1420
    0.1460
    0.1500
    0.1540
    0.1580
    0.1620
    0.1660
    0.1700
    0.1740
    0.1780
    0.1820
    0.1860
    0.1900
    0.1940
    0.1980
    0.2000
```

See Also

`simscape.logging.Series` | `simscape.logging.Series.values`

Tutorials

- "Log and Plot Simulation Data"

simscape.logging.Series.values

Purpose

Extract values vector from simulation series

Syntax

```
va = path_to_var.series.values
va = path_to_var.series.values('unit')
```

Description

`va = path_to_var.series.values` returns a row vector of variable values contained in the series, in default units. `path_to_var` is a full identifier path to the variable node associated with the series.

`va = path_to_var.series.values('unit')` returns a row vector of variable values in the specified units. `unit` must be commensurate with the default units of the variable.

For nonscalar variables of size *m*-by-*n*, this method returns a row vector of `m*n*steps` size, where `steps` is the number of steps in the series, and each `m*n` block represents the logged value for the variable in a column major form. For example, if a variable size is 2-by-2, then the first four elements in the row vector are the a_{11}, a_{21}, a_{12}, and a_{22} elements at the first time step.

Examples

Return the deformation values of a Translational Spring block, located at the top level of the model diagram:

```
v1 = simlog.Translational_Spring.x.series.values

v1 =

  1.0e-003 *

         0
    0.0000
    0.0000
    0.0001
    0.0002
    0.0004
    0.0007
    0.0012
    0.0018
    0.0025
    0.0034
    0.0044
    0.0056
    0.0070
    0.0085
    0.0101
    0.0119
    0.0139
    0.0160
    0.0183
    0.0207
    0.0233
    0.0260
    0.0289
    0.0319
    0.0351
    0.0384
    0.0419
    0.0455
    0.0492
    0.0531
    0.0572
    0.0614
    0.0657
    0.0702
    0.0748
    0.0796
    0.0845
    0.0895
    0.0947
    0.1000
    0.1055
    0.1111
    0.1168
    0.1227
    0.1287
    0.1348
    0.1411
    0.1475
    0.1540
    0.1607
    0.1675
    0.1710
```

The previous command returns the deformation values in meters (the default unit of the series). To return the same deformation values in different units, for example, in inches, type:

```
v1 = simlog.Translational_Spring.x.series.values('in')

v1 =

         0
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0001
    0.0001
    0.0001
    0.0002
    0.0002
    0.0003
    0.0003
    0.0004
    0.0005
    0.0005
    0.0006
    0.0007
    0.0008
    0.0009
    0.0010
    0.0011
    0.0013
    0.0014
    0.0015
    0.0016
    0.0018
    0.0019
    0.0021
    0.0023
    0.0024
    0.0026
    0.0028
    0.0029
    0.0031
    0.0033
    0.0035
    0.0037
    0.0039
    0.0042
    0.0044
    0.0046
    0.0048
    0.0051
    0.0053
    0.0056
    0.0058
    0.0061
    0.0063
    0.0066
    0.0067
```

See Also

`simscape.logging.Series` | `simscape.logging.Series.time`

Tutorials

- "Log and Plot Simulation Data"

sl_postprocess

Purpose

Make postprocessing customizations when building custom block library

Syntax

```
sl_postprocess(h)
```

Description

`sl_postprocess(h)` takes a handle to the custom block library, `h`, and allows you to make library postprocessing customizations (for example, add a forwarding table).

If a Simscape file package being built contains a `sl_postprocess.m` file, then `ssc_build` calls `sl_postprocess` once the block library (*`package_name`*`_lib`) is generated but before it is saved to disk. If `sl_postprocess` generates an error, the library does not build.

You can include a `sl_postprocess.m` file at any level in the library package. At the top level, it makes postprocessing changes to the whole custom block library. Similarly, if the `sl_postprocess.m` file resides in a sublibrary in the package, it takes a handle to that sublibrary and makes the corresponding changes.

Examples

Consider the following directory structure:

```
- +MySimscape
|-- +Mechanical
| |-- spring.ssc
| |-- ...
```

For example, you have restructured your custom `Mechanical` library to have two sublibraries, `Rotational` and `Translational`, and moved the `spring` block to the `Rotational` sublibrary. To update old models that reference the block, you need to add a forwarding table.

Instead of manually adding a forwarding table, which will get overwritten every time you rebuild the library, you can include a `sl_postprocess.m` file in the library package, which will add the forwarding table automatically upon rebuilding the library:

```
- +MySimscape
|-- sl_postprocess.m
|-- +Mechanical
| |-- +Rotational
| | |-- spring.ssc
| | |-- ...
| |-- +Translational
| |-- ...
```

The `sl_postprocess.m` file contains a forwarding table:

```
function sl_postprocess(h)
% Forwarding table for the spring block
ft = { {'MySimscape_lib/Mechanical/spring', 'MySimscape_lib/Mechanical/Rotational/spring'} }
set_param(h, 'ForwardingTable', ft);
end
```

Note that if you have customized the library names (using `lib.m` files) or the block name, you have to use these custom names in the forwarding table (for example, `'Rotational Spring'` instead of `'spring'`).

See Also

`ssc_build`

ssc_build

Purpose

Build custom library from collection of Simscape files

Syntax

```
ssc_build package
```

Description

`ssc_build` *`package`* generates a custom Simscape library file, named *`package`*`_lib`, containing all the sublibraries and blocks generated from the Simscape files (either source or protected) located in the package and its subdirectories. Simscape protected files have higher precedence than the source files when you build a library. If both the protected and the source files are present in the package, and the source files are out of date, `ssc_build` will use the protected files to build the library, but you will get a warning.

The argument, *`package`*, must be a top-level package name.

Note The package directory name begins with a leading + character, whereas the argument to `ssc_build` must omit the + character.

The package must be located in a directory on the MATLAB path. The *`package`*`_lib` is automatically placed in the package parent directory. For more information, see "Generate Custom Block Libraries from Simscape Component Files".

If you run the `ssc_build` command from inside the package directory structure, you can omit the argument.

Examples

For example, your top-level package directory, where you store your Simscape files, is named `+SimscapeCustomBlocks`. To generate a custom block library, at the MATLAB Command prompt, type:

```
ssc_build SimscapeCustomBlocks;
```

This command generates a Simulink model file called `SimscapeCustomBlocks_lib` in the parent directory of the top-level package (that is, in the same directory that contains your `+SimscapeCustomBlocks` package).

See Also

sl_postprocess | ssc_clean | ssc_mirror | ssc_protect

ssc_clean

Purpose

Clean all derived files generated by library build process

Syntax

```
ssc_clean package
```

Description

`ssc_clean` *`package`* deletes all derived files generated by `ssc_build` in the package named *`package`*, including the library file.

The argument, *`package`*, must be a top-level package name.

Note The package directory name begins with a leading + character, whereas the argument to `ssc_clean` must omit the + character.

Examples

To clean all derived files from the package directory `+MyPackage`, invoke the following from the directory containing the package directory `+MyPackage`:

```
ssc_clean MyPackage;
```

See Also

`ssc_build`

Purpose

Create protected mirror of library of Simscape files

Syntax

```
ssc_mirror package mirrordir buildmirror
```

Description

The `ssc_mirror` command lets you protect and build a whole package of Simscape files in one step.

`ssc_mirror` *`package mirrordir`* `buildmirror` creates a protected mirror of a package of Simscape files in a specified directory *`mirrordir`*, and also optionally builds a custom library from these files.

The first argument, *`package`*, must be a top-level package name.

Note The package directory name begins with a leading + character, whereas the argument to `ssc_mirror` must omit the + character.

The second argument, *`mirrordir`*, is the directory where the protected package is placed. The `ssc_mirror` command creates this directory, if it does not exist, recreates the whole package structure under it, generates the protected files, and places them in the appropriate mirror locations.

If the `buildmirror` flag is set to `true`, the `ssc_mirror` command also builds a custom Simscape library file, named *`package`*`_lib`, containing all the sublibraries and blocks generated from the Simscape files in the mirrored package (similar to the `ssc_build` command), and places the *`package`*`_lib` file in the *`mirrordir`* directory. The `buildmirror` flag is optional and the default is `false`, that is, by default the package is mirrored and protected but the library is not built.

For more information, see "Using Source Protection for Simscape Files".

Examples

For example, your top-level package directory, where you store your Simscape files, is named `+SimscapeCustomBlocks`. To protect, mirror, and generate a custom block library from this package in the directory `C:\Work\deploy`, at the MATLAB Command prompt, type:

```
ssc_mirror SimscapeCustomBlocks C:\Work\deploy true;
```

This command creates a mirror package, equivalent to the +SimscapeCustomBlocks package but consisting of Simscape protected files, in the directory C:\Work\deploy, and generates a file called SimscapeCustomBlocks_lib in the C:\Work\deploy directory.

See Also

sl_postprocess | ssc_build | ssc_clean | ssc_protect

Purpose

Create new Simscape model populated by required and commonly used blocks

Syntax

```
ssc_new
ssc_new('modelname')
ssc_new('modelname','domain')
ssc_new('modelname','domain','solver')
```

Description

`ssc_new` creates a new Simscape model, with required and commonly used blocks already on the model canvas, and opens the Simscape library. By default, it uses the Simulink default new model name `untitled` and the recommended solver `ode15s`.

`ssc_new('modelname')`creates a new Simscape model with the specified name.

`ssc_new('modelname','domain')`creates a new Simscape model with the specified name, and with domain-specific reference block added to the model canvas. Valid domains types are `'electrical'`, `'hydraulic'`, `'magnetic'`, `'pneumatic'`, `'rotational'`, `'translational'`, and `'thermal'`. You can use a cell array of domain types to add more than one type of reference block.

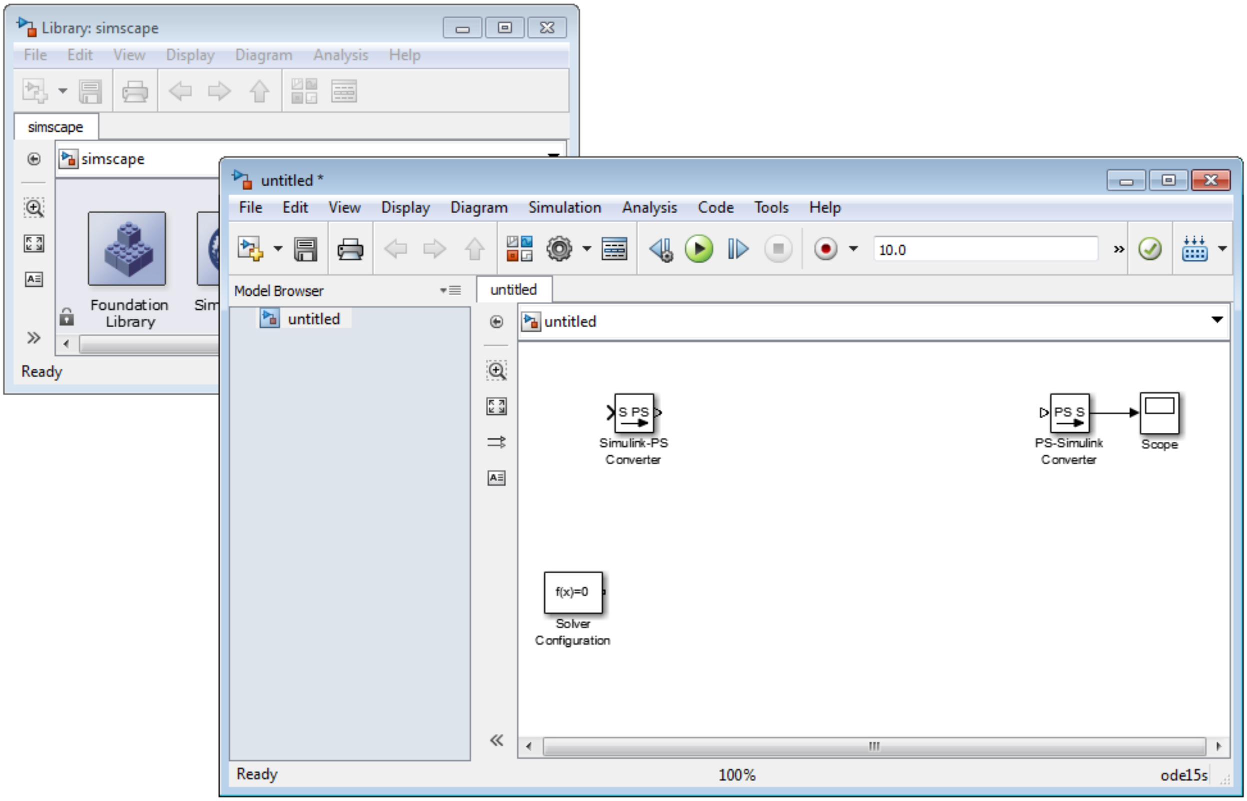
`ssc_new('modelname','domain','solver')` creates a new Simscape model with the specified name and domain type, and with the specified solver type. Recommended solver types for Simscape models are `'ode15s'`, `'ode23t'`, and `'ode14x'`. You can use other Simulink solvers, but, depending on the particular model, they may be less suitable. For more information, see “Setting Up Solvers for Physical Models”.

Examples

To create a generic Simscape model, type:

```
ssc_new
```

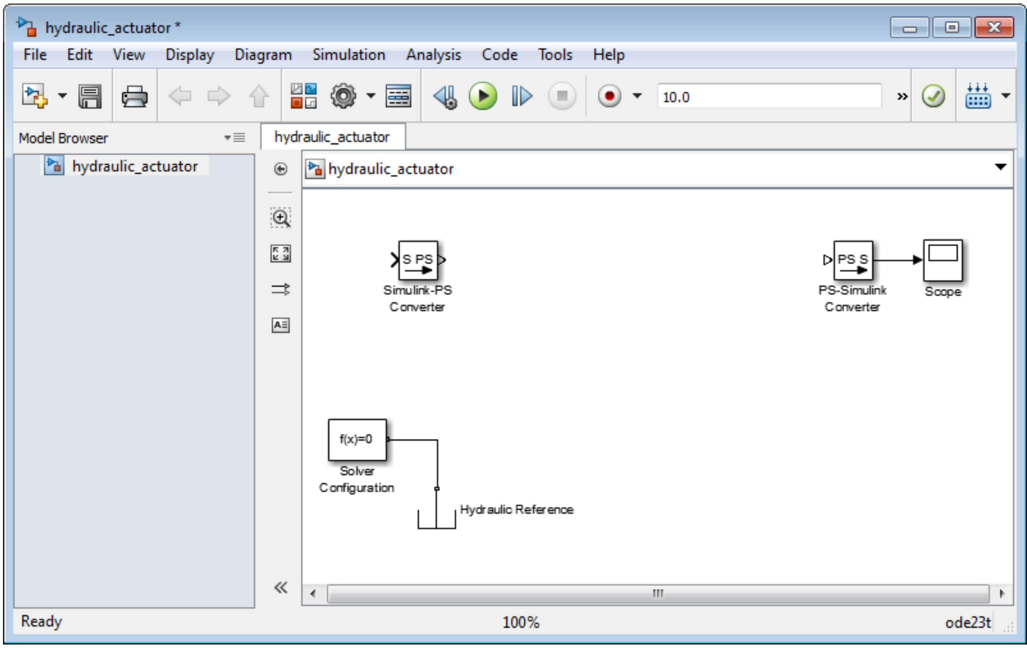
The software opens the main Simscape library and creates a new untitled model, which contains a Solver Configuration block with the default solver set to `ode15s`, a Simulink-PS Converter block, and a PS-Simulink Converter block connected to a Scope block.



To create a hydraulic model, called `hydraulic_actuator` and using the `ode23t` solver, type:

```
ssc_new('hydraulic_actuator','hydraulic','ode23t')
```

The software opens the main Simscape library and creates the following model.



After using `ssc_new`, continue developing your model by copying the blocks, as needed, and adding other blocks from the Simscape libraries.

How To

- "Creating a New Simscape Model"

Purpose

Generate Simscape protected files from source files

Syntax

```
ssc_protect filename
ssc_protect filename -inplace
ssc_protect dirname
ssc_protect dirname -inplace
```

Description

The `ssc_protect` command creates content-obscured files (Simscape protected files) from Simscape source files, to enable model sharing without disclosing the component or domain source. While Simscape source files have the extension `.ssc`, Simscape protected files have the extension `.sscp`.

`ssc_protect` *`filename`* generates a Simscape protected file, named *`filename`*`.sscp`, from the Simscape source file named *`filename`*`.ssc`, and places the protected file in your current working directory. *`filename`* can include absolute path to the file, or relative path if the file is in a subfolder of the current working directory. If this path includes package directories, the package structure will be recreated under the current working directory (unless it already exists) and the protected file placed in the package (see examples). The extension `.ssc` in *`filename`* is optional.

`ssc_protect` *`filename`* `-inplace` generates a Simscape protected file, named *`filename`*`.sscp`, from the Simscape source file named *`filename`*`.ssc`, and places the protected file in the same directory as the source file.

`ssc_protect` *`dirname`* generates Simscape protected files from all the Simscape source files in the directory named *`dirname`*, and places the protected files under your current working directory. If the path to *`dirname`* includes package directories, the package structure will be recreated under the current working directory (unless it already exists) and the protected files placed in the package, similar to when protecting a single file.

`ssc_protect` *`dirname`* `-inplace` generates Simscape protected files from all the Simscape source files in the directory named *`dirname`*, and places the protected files in the same directory as the source files.

Note Existing Simscape protected files are overwritten without warning.

For more information, see "Using Source Protection for Simscape Files".

Simscape protected files have higher precedence than the source files when you build a library. If the protected and the source files are in the same directory, and protected files are out of date, `ssc_build` will use the protected files to build the library, but you will get a warning.

Examples

To protect a single file, with the protected file placed under your current working directory, at the MATLAB Command prompt, type:

```
ssc_protect C:\Work\libraries\source\+SimscapeLibrary\+MechanicalElements\my_spring.ssc
```

This command creates a folder called `+SimscapeLibrary` and a subfolder called `+MechanicalElements` in your current working directory (unless these folders already exist) and generates a file called `my_spring.sscp` in the `+MechanicalElements` folder.

To protect a single file, with the protected file placed in the same directory as the source file, type:

```
ssc_protect C:\Work\libraries\source\+SimscapeLibrary\+MechanicalElements\my_spring.ssc -inplace
```

This command generates a file called `my_spring.sscp` in the `C:\Work\libraries\source\+SimscapeLibrary\+MechanicalElements` folder.

To protect all files in a directory, with the protected files placed under your current working directory, type:

```
ssc_protect C:\Work\libraries\source\+SimscapeLibrary\+MechanicalElements
```

This command generates protected files for each source file in the `C:\Work\libraries\source\+SimscapeLibrary\+MechanicalElements` folder, and places the protected files in a folder called

+SimscapeLibrary\+MechanicalElements in your current working directory (creating this folder structure, if it does not exist).

To protect all files in a directory, with the protected files placed in the same directory as the source files, type:

```
ssc_protect C:\Work\libraries\source\+SimscapeLibrary\+MechanicalElements -inplace
```

This command generates protected files for each source file in the C:\Work\libraries\source\+SimscapeLibrary\+MechanicalElements folder, and places the protected files in the same folder.

See Also

ssc_build | ssc_clean | ssc_mirror

Purpose

List reserved words

Syntax

```
ssc_reserved
words = ssc_reserved
```

Description

`ssc_reserved` returns a list of reserved Simscape language words. Simscape language has certain words, in addition to its keywords, that you cannot use as model or member names. This list may change from release to release, as limitations are removed. Use the `ssc_reserved` command to see the current list of reserved words.

`words = ssc_reserved` returns a list of reserved words in `words` as a cell array of strings.

`ssc_reserved` does not list the Simscape language keywords.

Examples

List the currently reserved words:

```
ssc_reserved

ans =

    'across_variable'
    'build'
    'description'
    'descriptor'
    'element'
    'input'
    'interface_input'
    'interface_node'
    'interface_output'
    'item_type'
    'local_variable'
    'name'
    'node'
    'output'
    'parameter'
    'setup'
```

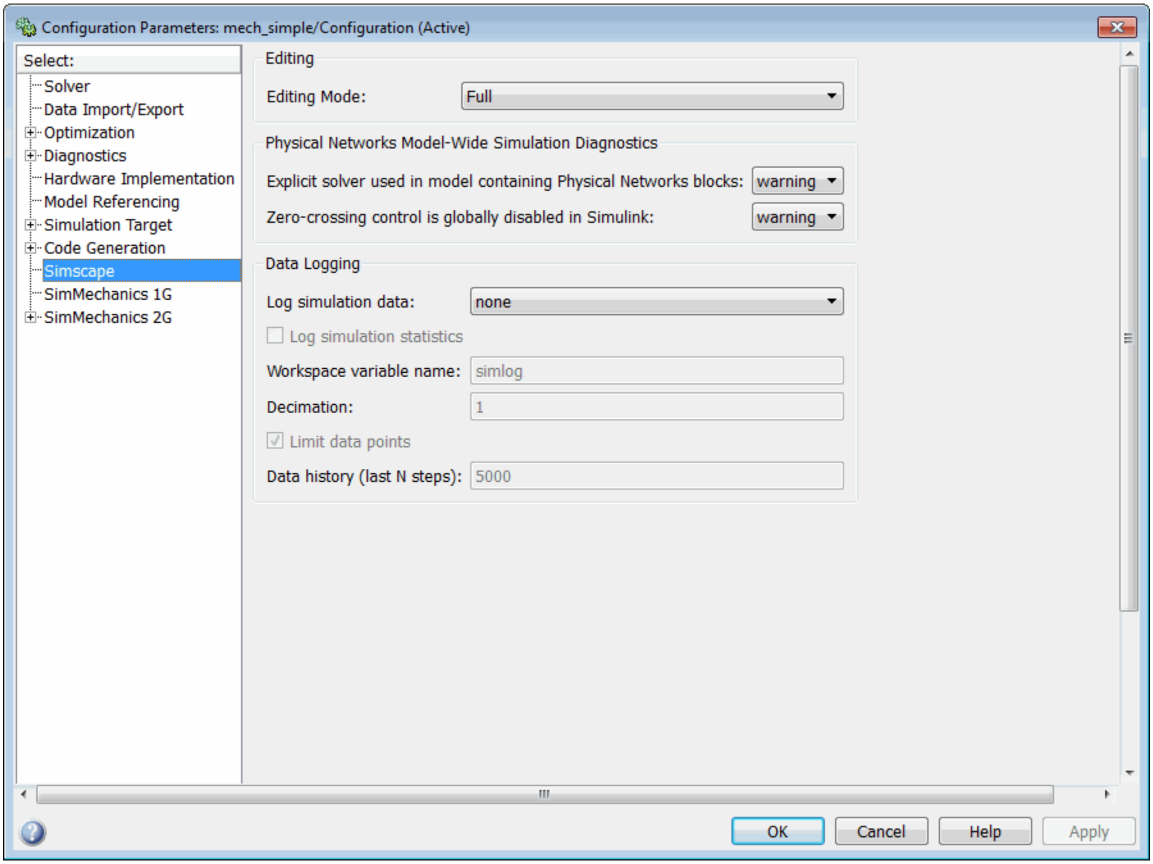
```
'signal'
'source'
'terminal'
'through_variable'
'variable'
```

You cannot use any of these words as model names (domain or component) or member names (parameter, variable, and so on). In addition to these reserved words, you cannot use any of the Simscape language keywords as model or member names.

3

Configuration Parameters

Simscape Pane: General



Configuration Parameters: mech_simple/Configuration (Active)

Editing

Editing Mode: Full

Physical Networks Model-Wide Simulation Diagnostics

Explicit solver used in model containing Physical Networks blocks: warning

Zero-crossing control is globally disabled in Simulink: warning

Data Logging

Log simulation data: none

□ Log simulation statistics

Workspace variable name: simlog

Decimation: 1

☑ Limit data points

Data history (last N steps): 5000

In this section...

In this section...

Simscape Pane Overview

The **Editing Mode** parameter controls the Simscape Editing Mode functionality, which allows you to open, simulate, and save models that contain blocks from add-on products in Restricted mode, without checking out add-on product licenses, as long as the products are installed on your machine. Simscape add-on products include SimDriveline™, SimElectronics®, SimHydraulics, and SimMechanics. Use this functionality to perform multidomain physical modeling and simulation while minimizing the number of required licenses.

Note Unless your organization uses concurrent licenses, see the Simscape product page on the MathWorks Web site for specific information on how to install add-on products on your machine, to be able to work in Restricted mode.

The parameters in the **Physical Networks Model-Wide Simulation Diagnostics** section let you configure your preferences for solver-related warnings when you simulate models containing blocks from Simscape libraries.

The parameters in the **Data Logging** section let you log simulation data to workspace.

Configuration

This pane appears only if your model contains a block from the Simscape libraries (including Simscape add-on products).

See Also

- About the Simscape Editing Mode
- Working with Restricted and Full Modes
- Harmonizing Simulink and Simscape Solvers
- About Simulation Data Logging

Editing Mode

Set the editing mode of the model to either Full or Restricted.

Settings

Default: `Full`

`Full`

Sets the editing mode of the model to Full. In this mode, you can make any modifications to the model.

When you open a model in Full mode, the license manager checks out all the add-on product licenses for the blocks present in the model.

When you switch from Restricted to Full mode, the license manager checks whether the required add-on product licenses are available and checks them out. If some of the add-on product licenses are not available, the license manager issues an error and the model stays in Restricted mode.

`Restricted`

Sets the editing mode of the model to Restricted. In this mode, you can simulate the model, generate code, and make limited modifications.

When you open a model in Restricted mode, the license manager does not check out the add-on product licenses.

When you switch from Full to Restricted mode, all the add-on product licenses for the blocks present in the model remain checked out until the end of the MATLAB session.

Command-Line Information

Parameter: `EditingMode`
Type: string
Value: `'Full' | 'Restricted'`
Default: `'Full'`

See Also

- Saving a Model in Restricted Mode
- Switching from Restricted to Full Mode

Explicit solver used in model containing Physical Networks blocks

Specify whether or not the system will issue a warning or error upon simulation if the model uses an explicit solver.

Settings

Default: `warning`

`warning`

Makes the system issue a warning upon simulation if the model uses an explicit solver.

It is possible to choose any variable-step or fixed-step solver for models containing Simscape blocks. When you first create a model, the default Simulink solver is ode45. However, implicit solvers, such as ode14x, ode23t, and ode15s, are a better choice for a typical model. In particular, for stiff systems, implicit solvers typically take many fewer timesteps than explicit solvers, such as ode45, ode113, and ode1. To alert you to a potential issue, the system issues a warning when you use an explicit solver in a model containing Simscape blocks.

`error`

Makes the system issue an error upon simulation if the model uses an explicit solver.

If your model is stiff, and the use of explicit solvers undesirable, you may choose to select this option to avoid troubleshooting errors in the future.

`none`

Turns off issuing a warning or error upon simulation with explicit solver.

For models that are not stiff, explicit solvers can be effective, often taking fewer timesteps than implicit solvers. If you work with such models and use explicit solvers, select this option to turn off the warning upon simulation.

Command-Line Information

Parameter: `ExplicitSolverDiagnosticOptions`
Type: string
Value: `'warning'` | `'error'` | `'none'`
Default: `'warning'`

See Also

Switching from the Default Explicit Solver to Other Simulink Solvers

Zero-crossing control is globally disabled in Simulink

Specify whether or not the system will issue a warning or error upon simulation if the **Zero-crossing control** parameter in the **Solver** pane is set to `Disable all`, which means that zero-crossing control is globally disabled.

Settings

Default: `warning`

`warning`
: Makes the system issue a warning upon simulation if zero-crossing control is globally disabled.

`error`
: Makes the system issue an error upon simulation if zero-crossing control is globally disabled.

Command-Line Information

Parameter: `GlobalZcOffDiagnosticOptions`
Type: string
Value: `'warning' | 'error'`
Default: `'warning'`

See Also

Enabling or Disabling Simulink Zero Crossing Detection

Log simulation data

Specify whether or not the system logs simulation data to workspace.

Settings

Default: `none`

`none`

Performs no data logging upon simulation.

`all`

Upon simulating the model, logs all simulation data from Simscape blocks to a workspace variable specified by the **Workspace variable name** parameter.

Command-Line Information

Parameter: `SimscapeLogType`
Type: string
Value: `'none' | 'all'`
Default: `'none'`

See Also

How to Log Simulation Data

Log simulation statistics

Specify whether to log simulation statistics as part of simulation data.

Settings

Default: off

☑ On

Logs simulation statistics.

□ Off

Does not log simulation statistics.

Command-Line Information

Parameter: `SimscapeLogSimulationStatistics`
Type: string
Value: `'on' | 'off'`
Default: `'off'`

See Also

How to Log Simulation Data

Workspace variable name

Specify the name of the workspace variable for simulation data logging.

Settings

Default: `simlog`

- The default value logs all the simulation data to a workspace variable named `simlog`.
- You can specify any other valid string as the workspace variable name.

Command-Line Information

Parameter: `SimscapeLogName`
Type: string
Value: any valid value
Default: `'simlog'`

See Also

How to Log Simulation Data

Decimation

Lets you limit data points being logged, by skipping time steps. Logs data points for the first time step and every *n*th time step thereafter, where *n* is the decimation factor.

Settings

Default: `1`

- The default value logs simulation data for each step.
- You can specify any other positive integer number. For example, specifying `2` logs data points for every other time step, while specifying `10` logs data points for just one in ten steps.

Tips

- Saving data to workspace can slow down the simulation and consume memory. Use this parameter to limit the number of data points saved.
- Another way to limit the number of data points saved is using the **Limit data points** check box in conjunction with the **Data history (last N steps)** parameter. The two methods work independently from each other and can be used separately or together.

Command-Line Information

Parameter: `SimscapeLogDecimation`
Type: numeric
Value: any positive integer value
Default: `1`

See Also

How to Log Simulation Data

Limit data points

Specify that the number of data points logged to workspace is limited to the value corresponding to the number of simulation steps specified by the **Data history (last N steps)** parameter.

Settings

Default: on

☑ On

Limits the number of data points exported to workspace to those for the number of steps specified by the **Data history (last N steps)** parameter.

□ Off

Does not limit the number of data points.

Tips

- Saving data to workspace can slow down the simulation and consume memory. Use this parameter to limit the number of data points saved.
- Another way to limit the number of data points saved is using the **Decimation** parameter. The two methods work independently from each other and can be used separately or together.
- You must select the **Limit data points** check box before specifying the number of steps in the **Data history (last N steps)** parameter.

Command-Line Information

Parameter: `SimscapeLogLimitData`
Type: string
Value: `'on' | 'off'`
Default: `'on'`

See Also

How to Log Simulation Data

Data history (last N steps)

Specify the number of simulation steps to limit the number of data points output to workspace. The workspace variable defined by the **Workspace variable name** parameter contains the data points corresponding to the last N steps of the simulation, where N is the value you specify for the **Data history (last N steps)** parameter. If the simulation contains fewer steps than the number specified, the workspace variable contains the data points for the whole simulation.

Settings

Default: `5000`

- The default value logs simulation data for the last 5000 steps.
- You can specify any other positive integer number.

Tips

- Saving data to workspace can slow down the simulation and consume memory. Use this parameter to limit the number of data points saved.
- You must select the **Limit data points** check box before specifying the number of steps in the **Data history (last N steps)** parameter.

Command-Line Information

Parameter: `SimscapeLogDataHistory`
Type: numeric
Value: any positive integer value
Default: `5000`

See Also

How to Log Simulation Data

Model Advisor Checks

Simscape Model Advisor Checks

In this section...

Simscape Checks Overview

Use Simscape Model Advisor checks to identify Simscape blocks with ambiguous setting of parameter units, or outdated Simscape blocks in your model.

See Also

- Consulting Model Advisor

Modeling Physical Systems Checks Overview

Use the Modeling Physical Systems Model Advisor checks to identify Simscape blocks with ambiguous setting of parameter units.

See Also

- Consulting Model Advisor

Check consistency of block parameter units

Check model for Simscape blocks with ambiguous setting of parameter units.

Description

This check identifies blocks in your model that have an ambiguous setting of parameter units. This situation most often applies to frequency and angular velocity units.

For example, a parameter expected in Hz (1/s) may be specified in the block dialog with unit of rad/s. These units are commensurate, but not directly convertible, and using one instead of the other may result in unexpected conversion factors applied to the numerical value by the block equations. The purpose of the check is to verify that the specified unit matches your design intent.

Available with Simscape.

Results and Recommended Actions

Condition	Recommended Action
This model contains blocks where parameter units are not directly convertible to those expected by the block. After running the check, you get a table of results in the right pane of the Model Advisor window. Each cell in the first column of the table contains a link to the problematic block, and the corresponding cell in the second column contains the name of parameter in question, the expected unit, and the specified unit. Clicking on a link highlights the corresponding block in the model.	Double-click the highlighted block, verify the parameter unit setting and correct it, if necessary. Then save and reload the model.

See Also

- "Units for Angular Velocity and Frequency"

Check for outdated Simscape blocks

Check model for Simscape blocks that should be updated to the current version of the product.

Description

This check identifies blocks in your model that do not match the latest version of the block in the Simscape block libraries.

Blocks from previous versions may be missing parameters available in the latest version. In this case, simulating the model may produce warnings or unexpected results.

Available with Simscape.

Results and Recommended Actions

Condition	Recommended Action
This model contains outdated Simscape blocks. After running the check, you get a list of links to the outdated blocks in the right pane of the Model Advisor window. Clicking on a link highlights the corresponding block in the model.	To update the blocks, scroll down the right pane of the Model Advisor window and click the **Update** button. • If the automatic update is successful, the Results box displays a message that all blocks have been updated to the current Simscape version. • If the message says that some of the blocks could not be updated automatically, rerun the check and manually replace the outdated blocks with the latest version from the block library. Alternately, you can run `slupdate(`*`modelname`*`)` at MATLAB command prompt to migrate the model to the latest version of Simscape software.

See Also

- `slupdate`

Bibliography

[1] Andersen, B. W. *The Analysis and Design of Pneumatic Systems.* New York: John Wiley & Sons, 1967.

[2] Armstrong, B., and C. C. de Wit. “Friction Modeling and Compensation.” *The Control Handbook.* Boca Raton, Florida: CRC Press, 1995.

[3] Beater, P. *Pneumatic Drives. System Design, Modeling and Control.* New York: Springer, 2007.

[4] Brauer J.R. *Magnetic Actuators and Sensors.* Piscataway, NJ: Wiley-IEEE Press, 2006.

[5] Fitzgerald A.E., Kingsley C., Jr., and Umans S.D. *Electric Machinery.* Sixth edition. New Delhi: Tata McGraw Hill, 2002.

[6] Holcke, Jan. “Frequency Response of Hydraulic Hoses.” Licentiate Thesis. Royal Institute of Technology, KTH, Stockholm, 2002.

[7] Kahaner, D., Cleve Moler, and Stephen Nash. *Numerical Methods and Software.* Englewood Cliffs, NJ: Prentice Hall, 1988.

[8] Lorenz R.D. and Haines L.P. *Understanding Modern Power Conversion.* Third edition. Madison: University of Wisconsin, 2000.

[9] Martin, H. *The Design of Hydraulic Components and Systems.* New York: Ellis Horwood, 1995.

[10] Meritt, H. E. *Hydraulic Control Systems.* New York: John Wiley & Sons, 1967.

[11] Moran M.J. and Shapiro H.N. *Fundamentals of Engineering Thermodynamics.* Second edition. New York: John Wiley & Sons, 1992.

[12] Pêcheux, F., B. Allard, C. Lallement, A. Vachoux, and H. Morel. “Modeling and Simulation of Multi-Discipline Systems using Bond Graphs and VHDL-AMS.” International Conference on Bond Graph Modeling and Simulation (ICBGM). New Orleans, USA, 23–27 Jan. 2005.

[13] Press, W. H., B. P. Flannery, S. A. Teulkolsky, and W. T. Wetterling. *Numerical Recipes in C: The Art of Scientific Computing*. New York: Cambridge University Press, 1992.

[14] Sanville, F. E. “A New Method of Specifying the Flow Capacity of Pneumatic Fluid Power Valves.” Paper D3, p.37-47. BHRA. Second International Fluid Power Symposium, Guildford, England, 1971.

[15] Shapiro, A. H. *The Dynamics and Thermodynamics of Compressible Fluid Flow*. Vol. 1. New York: John Wiley & Sons, 1953.

[16] White, F. M. *Viscous Fluid Flow*. New York: McGraw-Hill, 1991.

[17] Yeapple, F. *Fluid Power Design Handbook*, Third edition. New York: Marcel Dekker, 1996.

Glossary

across variables
Variables that are measured with a gauge connected in parallel to an element.

add-on products
Products in the Physical Modeling family that use Simscape platform and, as a result, share common functionality such as physical units management, editing modes, and so on.

conserving ports
Bidirectional hydraulic or mechanical ports that represent physical connections and relate physical variables based on the Physical Network approach.

globally assigned positive direction
Direction considered positive for a model diagram.

nonrestricted parameters
Parameters that are available for modification when you open a model in Restricted mode. Usually, these are the block parameters with plain numerical values, such as **Chamber volume** or **Wheel radius**. Information on restricted and nonrestricted parameters is listed in block reference pages.

physical connections
Bidirectional connections between the blocks that mimic physical connections between elements.

physical signal ports
Unidirectional ports (inports and outports) transferring signals that use an internal Simscape engine for computations.

restricted parameters
Parameters that are not available for modification when you open a model in Restricted mode. You have to be in Full mode to modify them. Usually, these are the block parameterization options, such as **Chamber specification** or **Mechanism orientation**. Information

on restricted and nonrestricted parameters is listed in block reference pages.

through variables

Variables that are measured with a gauge connected in series to an element.

Index