SimDriveline

For Use with Simulink®

- Modeling
- Simulation
- Implementation

User's Guide

Version 1

The MathWorks

How to Contact The MathWorks:

`www.mathworks.com`	Web
`comp.soft-sys.matlab`	Newsgroup
`support@mathworks.com`	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	Mail

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

SimDriveline User's Guide

Printing History:

June 2004	Online only for Version 1.0 (Release 14+) New

Contents

1 Introducing SimDriveline

2 Modeling Drivetrain Systems

3 SimDriveline Block Reference

4 Blocks – Alphabetical List

A Technical Conventions

B Bibliography

Index

1

Introducing SimDriveline

Welcome to SimDriveline

SimDriveline is a block diagram modeling environment for the engineering design and simulation of drivelines, which are idealized powertrain systems. The function of a driveline is to propel a vehicle or craft by transferring its engine torque and power into vehicle momentum and kinetic energy. Drivelines consist of bodies spinning around fixed axes and subject to Newton's laws of motion. The bodies can revolve about one axis, multiple parallel axes, or multiple nonparallel axes. Simple and complex gears constrain the bodies to revolve together and transfer torque up and down the driveline axes. Locking and unlocking clutches switch the driveline from one gear set to another. Gears and clutches make up transmissions.

With SimDriveline, you represent a driveline machine with a connected block diagram, like other Simulink models, and you can group blocks into hierarchical subsystems. You can initiate and maintain rotational motion in a driveline with actuators while measuring, via sensors, the motions of driveline elements and the torques acting on them. You can return sensor signals to the driveline via actuators, forming feedback loops and the basis for controls.

The SimDriveline libraries offer blocks to represent rotating bodies; gear constraints among bodies; special dynamics elements such as spring-damper forces, rotational stops, and clutches; transmissions; and sensors and actuators. SimDriveline also lets you analyze linearized versions of your models and generate code from them.

SimDriveline and Physical Modeling

SimDriveline is part of Simulink Physical Modeling, encompassing the modeling and design of systems according to basic physical principles. Physical Modeling runs within the Simulink environment and interfaces seamlessly with the rest of Simulink and with MATLAB. Unlike other Simulink blocks, which represent mathematical operations or operate on signals, Physical Modeling blocks represent physical components or relationships directly.

Note This SimDriveline User's Guide assumes that you have some experience with building and running models in Simulink.

What's in This Chapter

This chapter introduces you to the major features of SimDriveline and its relationship to other MathWorks products:

- "Related Products" on page 1-4
- "Running a Demo Model" on page 1-5
- "What Can You Do with SimDriveline?" on page 1-17

Getting Online Help for SimDriveline

You can get help online in a number of ways to assist you while you use SimDriveline.

Using the MATLAB Help System for Documentation and Demos

The MATLAB Help browser allows you to access the documentation and demo models for all the MATLAB and Simulink-based products that you have installed. The online Help includes an online index and search system.

Consult the "Help Browser" section of the Getting Started with MATLAB documentation for more about the MATLAB Help system.

Finding Special SimDriveline Help

This User's Guide includes special reference chapters for use with SimDriveline.

- Appendix A, "Technical Conventions" explains special conventions, abbreviations, and units.
- Appendix B, "Bibliography" lists external references on driveline and powertrain modeling and related topics.

Related Products

The MathWorks provides several products that are especially relevant to the kinds of tasks you can perform with SimDriveline.

Requirements for SimDriveline

You must have the following products installed to use SimDriveline:

- MATLAB 7.0
- Simulink 6.5

Other Related Products

The related products listed in the SimDriveline product page at the MathWorks Web site include toolboxes and blocksets that extend the capabilities of MATLAB and Simulink. These products will enhance your use of SimDriveline in various applications.

The Physical Modeling Product Family

In addition to SimDriveline, the Physical Modeling product family includes SimMechanics, for modeling and simulating general mechanical systems, and SimPowerSystems, for modeling and simulating electrical power systems. Use these products together to model physical systems in Simulink.

For Information about MathWorks Products

For more information about any MathWorks software products, see either

- The online documentation for that product if it is installed or if you are reading the documentation from the CD
- The MathWorks Web site at `www.mathworks.com`; see the “Products” section.

Running a Demo Model

The demo model of this section simulates a complete drivetrain. This model will help you understand how to model driveline components with SimDriveline blocks, connect them into a realistic model, use Simulink blocks as well, and simulate and modify a drivetrain model.

The driveline mechanism modeled here is part of a full vehicle, without the engine or engine-drivetrain coupling, and without the final differential and wheel assembly. The model includes an actuating torque, driver and driven shafts, a four-speed transmission, and a braking clutch.

What the Model Illustrates

The demo model is `drive_simpson`. This driveline accepts a driving torque and transfers this torque and the associated angular motion from the input or drive shaft to an output or driven shaft through a transmission. The model includes a Simpson 4-speed transmission subsystem, based on two gears and four clutches. You can set the transmission to four different gear combinations, allowing four different effective torque and angular velocity ratios. A fifth clutch acts as a brake on the driven shaft.

The Simpson subsystem illustrates a critical feature of transmission design, the *clutch schedule*. To be fully engaged, the Simpson 4-speed transmission, with four clutches and two gears, requires two clutches to be locked and the other two unlocked at any time. The choice of which two clutches to lock determines the effective gear ratio across the transmission. The clutch schedule is the table of locked and free clutches corresponding to different gear settings. If all four clutches are unlocked, the transmission is in neutral. If the clutches are completely disengaged, no torque or angular motion at all is transferred across the transmission.

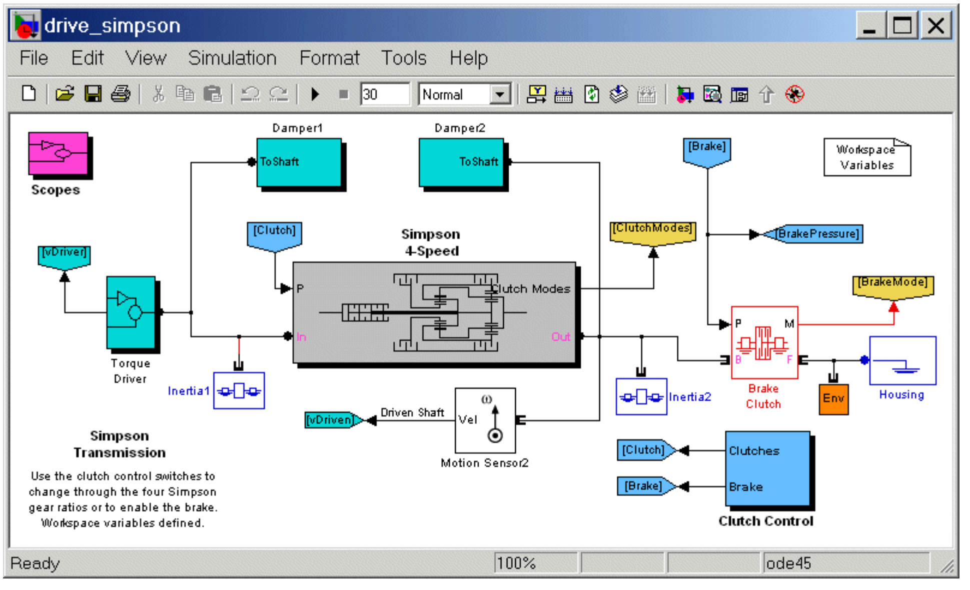
Clutch Schedule for the Simpson 4-Speed Transmission

Gear Setting	Clutch A State	Clutch B State	Clutch C State	Clutch D State
1	L	F	F	L
2	L	F	L	F
3	L	L	F	F
4	F	L	L	F

Opening the Model

You can open the Simpson 4-speed demo in several ways.

- Enter `drive_simpson` at the MATLAB command line.
- Open the MATLAB Help browser from the MATLAB desktop. In the Help navigator pane to the left, click the **Demos** tab. Open the **Simulink** node, then the **SimDriveline** subnode. Locate the Simpson 4-speed transmission demo entry and double-click it.

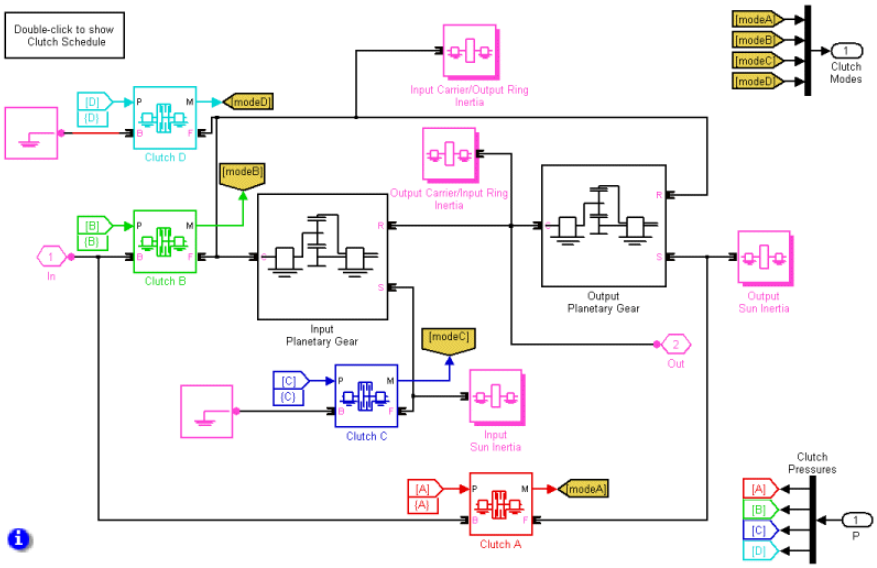


Examine the model and its structure. Open each subsystem in turn.

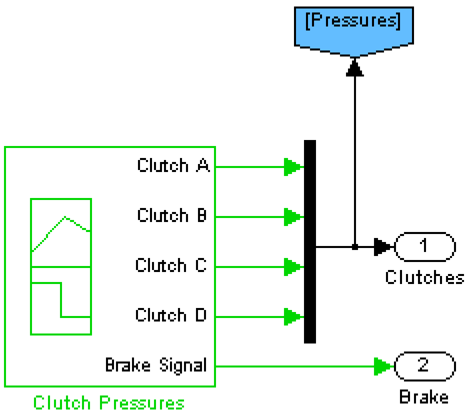
- The main model window contains the Simpson transmission subsystem, the input or driver shaft assembly, and the output or driven shaft assembly. Each assembly consists of a wheel with applied kinetic friction. The driver shaft transmits an externally-specified torque down the driveline.

  The main model also includes a brake clutch. When this clutch is locked, the driven shaft stops turning. This clutch must remain unlocked if the Simpson transmission is engaged.

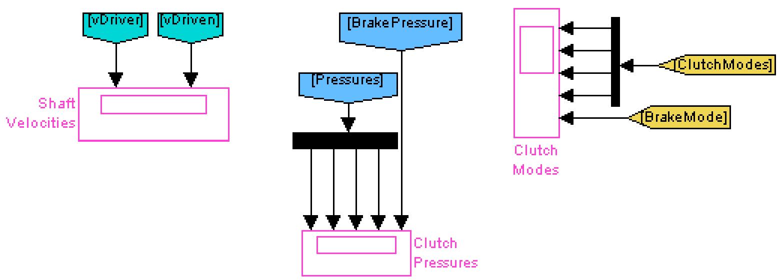
- The Simpson 4-speed transmission subsystem is a set of four clutches, two planetary gears, and four inertias (rotating bodies). Within the transmission subsystem, open the clutch schedule block to see the four possible gear settings for the Simpson 4-speed transmission. Two clutches must be locked at any one time for the transmission to be engaged. Exactly two clutches must be engaged simultaneously in order to avoid conflicting constraints on the gear motions.



- The clutch control subsystem provides the pressures that lock the necessary clutches. The clutch controller is programmed to move the transmission through a fixed sequence of gears, then unlock all the transmission clutches, allowing the driven shaft to “coast” for a time, and then engage and lock the brake clutch to stop the driven shaft.



- The Scopes subsystem provides Scope blocks to observe the clutch pressure, driver and driven shaft velocity, and clutch mode signals.



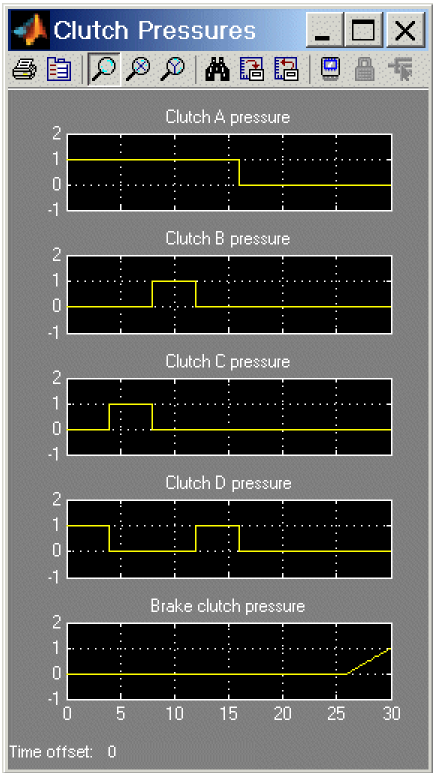
Running the Model

To observe the Simpson driveline model’s behavior,

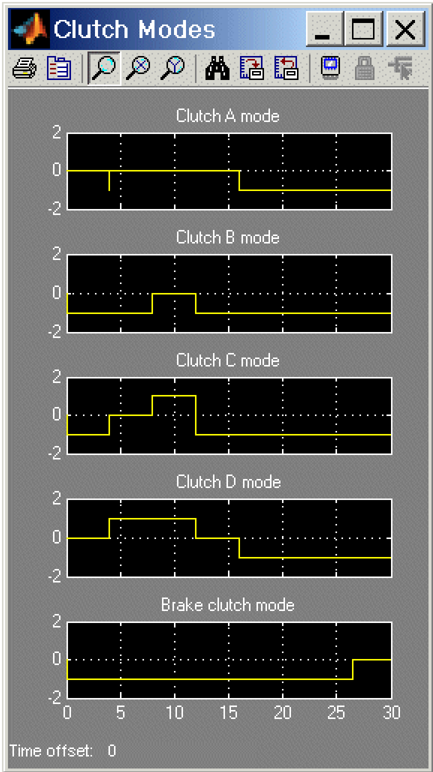
1 Open the Scopes subsystem and then each of the Scope blocks. Close the Scopes subsystem.

2 Click **Start**. The model steps through the sequence of gears and then brakes.

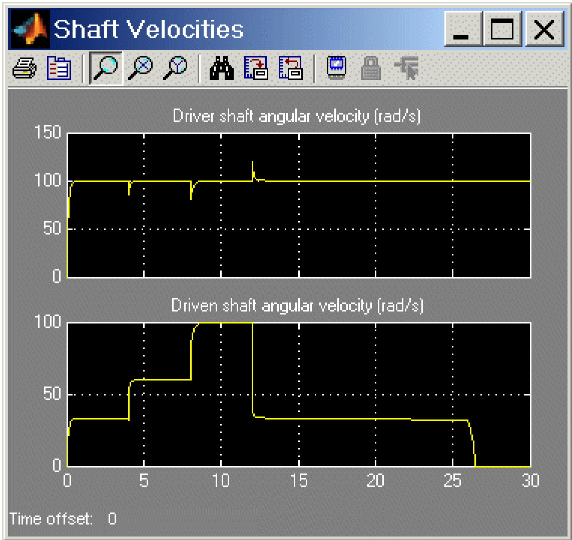
3 Observe how the clutch pressure signals move the transmission into one gear after another, at zero, four, eight, and 12 seconds of simulation time. Compare these clutch pressure signals to the clutch schedule in the Simpson transmission subsystem to determine which gear settings the model is implementing. (In fact, the model steps through gears 1, 2, 3, and then 1 again, before coasting and then braking.)



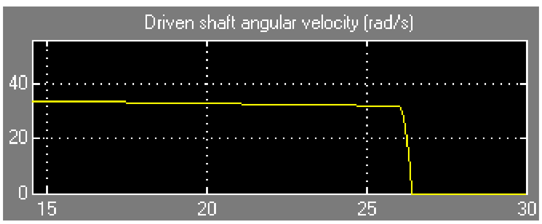
4 Observe the clutch modes at the same time. When a clutch mode is zero, that clutch is locked. The sequence of clutch locking and unlocking matches the sequence from the clutch schedule.



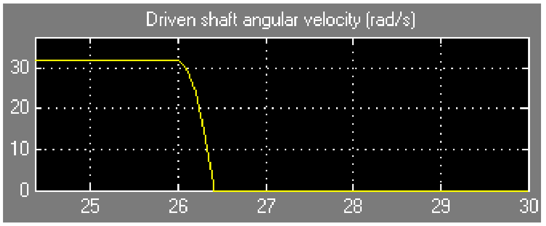
5 Compare the angular velocities of the driven and driver shafts. The effect of the transmission is the result of the two planetary gears coupled in different ways in the different gear settings. The effective *drive ratio* of output to input shafts is the *reciprocal* of the ratio of output to input angular velocities.



6 Observe what happens at 16 seconds. The transmission clutch pressures drop to zero, and the transmission disengages. The transmission ceases to transfer angular motion and torque from the driver to the driven shaft, and the driven shaft continues to spin simply from inertia. A small kinetic friction damping gradually slows down the driven shaft over the next 10 seconds.



7 At 26 seconds of simulation time, the brake clutch pressure begins to rise from zero, and the brake clutch engages. The driven shaft decelerates more drastically now. At about 26.4 seconds, the brake clutch locks, and the driven shaft stops rotating completely.



Modifying the Model

You can modify this demo model to explore other features of SimDriveline. Here you modify and rerun the model to investigate two aspects of its motion.

- Measure the effective drive ratio of the Simpson transmission in each gear setting that it steps through.
- Change the gear sequence.

Measuring the Drive Ratio of the Simpson Transmission States

The gear ratio (output to input) is the ratio of the output gear wheel radius to the input gear radius. Equivalently, the gear ratio is the ratio of the number of teeth on the output gear wheel to the number on the input wheel, or the ratio of the output torque to the input torque. The ratio of the angular velocities of output to input is the reciprocal of this gear ratio.

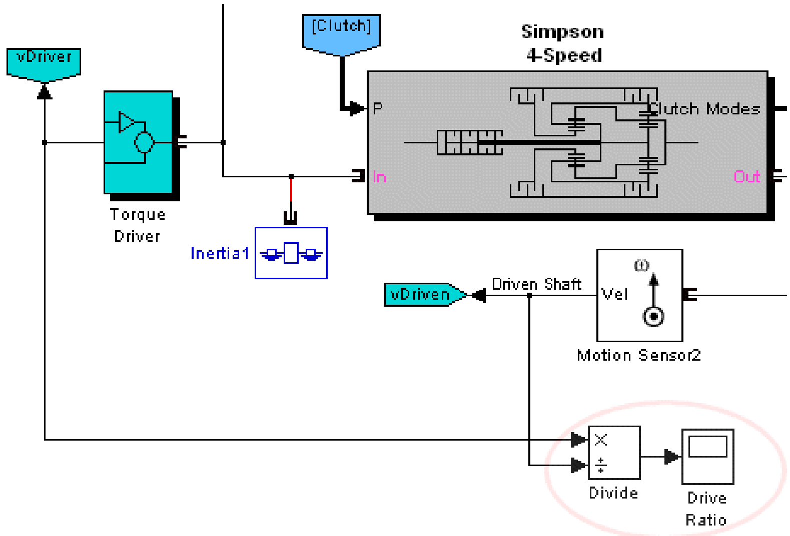
A transmission is a set of coupled gears. But for a particular transmission gear setting, the ratio of driven (output) shaft velocity to the driver (input) is fixed. Its reciprocal, the *drive ratio*, is like a gear ratio of an individual gear coupling, but for the whole transmission.

Add and connect the Simulink blocks necessary to measure the drive ratio of the transmission.

1. From the Simulink Math Operations library, copy a Divide block and, from the Simulink Sinks library, copy a Scope block.
2. From the Torque Driver subsystem outport, branch a signal line from Motion Sensor1's angular velocity output and connect it to the `X` inport on the Divide block. From the velocity output of Motion Sensor2, on the driven (output) shaft, again branch a signal line. Connect it to the ÷ inport on the Divide block.

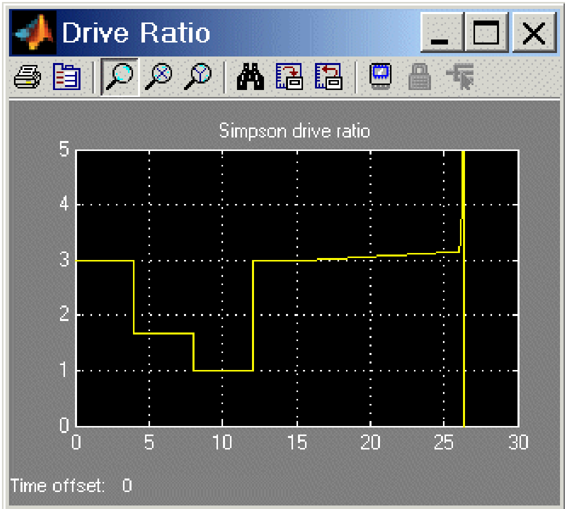
The effective output-to-input drive ratio is the ratio of input to output velocities.

3 Connect the outport of the Divide block to the Scope. Rename Scope to Drive Ratio.



4 Open the Drive Ratio scope and restart the demo. Observe how the drive ratio steps through a sequence of four-second states, in parallel with the clutch pressures and clutch modes, until it reaches 16 seconds. The drive ratio measurement after 16 seconds is not meaningful, because the transmission is uncoupled.

Just after 26 seconds, the driven shaft velocity drops to zero, and your Divide block will result in divide-by-zero warnings at the MATLAB command line.



5 Look inside the Simpson 4-Speed Transmission subsystem for the Clutch Schedule block and open it. Consult the drive ratios for each gear, 1, 2, and 3, in terms of the gear ratios of the transmission's two Planetary Gears. Determine the numerical values for these drive ratios for gear settings 1, 2, and 3, and check them against the values displayed in the Drive Ratio scope.

   The drive ratio sequence should be 3, 5/3, 1, and 3, respectively, for the first, second, third, and fourth intervals of four seconds each.

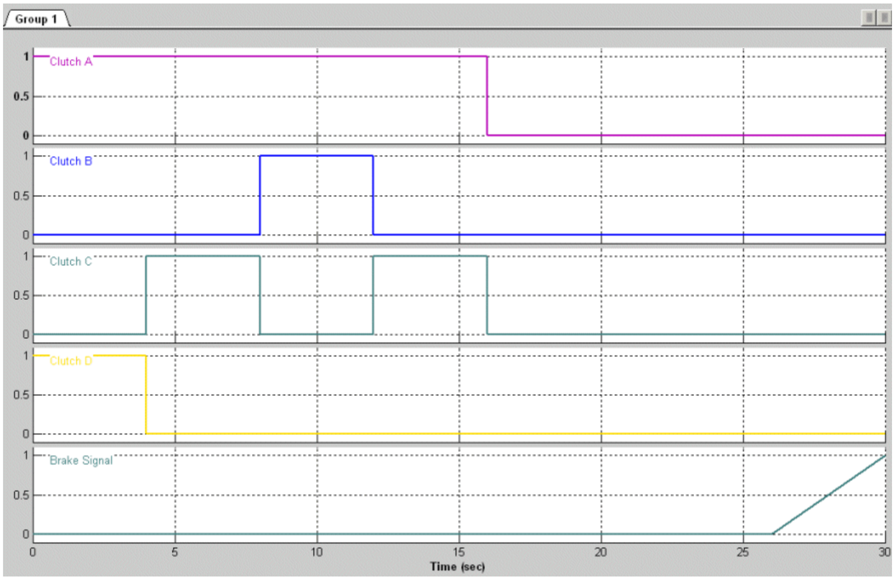
Changing the Transmission Gear Sequence

The `drive_simpson` demo, when you open it, is programmed to step through Simpson gear settings 1, 2, 3, then 1, before disengaging. Modify it to step through settings 1, 2, 3, then 2, then disengage. The first gear requires A, B, C, and D to be locked, free, free, and locked, respectively. You will modify the clutch pressure signal sequence from 12 to 16 seconds so that the transmission is set in second, not first, gear. The second gear requires clutches A, B, C, and D to be locked, free, locked, and free, respectively.

1 Open the Clutch Control and Simpson transmission subsystems. Within the transmission, open the Clutch Schedule block and review the clutch lockings for each gear setting.

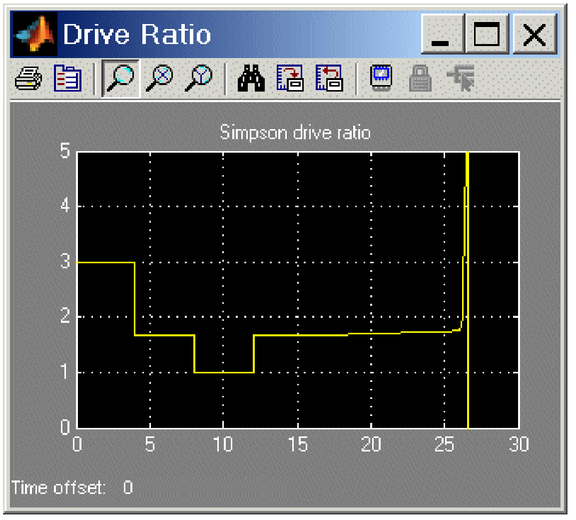
2 Open the Signal Builder block, labeled Clutch Pressures, to view the clutch pressure signals.

Modify clutch pressure signals C and D so that, between 12 and 16 seconds, clutch C is locked (not free) and clutch D is free (not locked). Sufficient pressure will lock the clutches, while zero input pressure leaves a clutch unlocked.



3 Restart the model. Observe that between 12 and 16 seconds of simulation time the clutch pressures, the clutch modes, and the driven shaft velocity are now different from the original version of the model.

Check the effective drive ratio between 12 and 16 seconds to confirm that the Simpson transmission during that time is set in gear 2, not gear 1. This fourth interval of four seconds should exhibit a drive ratio of 5/3 instead of 3.


Drive Ratio
Simpson drive ratio
5
4
3
2
1
0
0
5
10
15
20
25
30
Time offset: 0

What Can You Do with SimDriveline?

SimDriveline is a set of block libraries and special simulation features for use in the Simulink environment. You connect SimDriveline blocks to normal Simulink blocks through special Sensor and Actuator blocks.

The blocks in these libraries are the elements you need to model driveline systems consisting of any number of rotating inertias, rotating about one or more axes, constrained to rotate together by gears, which transfer torque to different parts of the driveline. SimDriveline can represent machines with components organized into hierarchical subsystems, as in normal Simulink models. You can add complex dynamic elements such as clutches and transmissions, actuate bodies with external torques or motions, integrate the Newtonian rotational dynamics, and measure the resulting motions.

Modeling Drivetrains with SimDriveline

SimDriveline extends Simulink with blocks to specify a driveline's components and properties and to solve the machine's equations of motion. The blocks are similar to other Simulink blocksets, with some properties unique to SimDriveline.

These are the major steps you follow, using SimDriveline, to build and run a model representation of a driveline machine:

- Specify rotational inertia for each body and connect the bodies with driveline connection lines representing driveline axes. If needed, ground the driveline to one or more housings fixed in space.
- Constrain the driveline axes to rotate together by connecting them with gears. Gears impose static constraints on driveline motions and transfer torques at fixed ratios.
- As necessary and desired, add dynamic driveline elements that transfer torque and motion among driveline axes in a nonstatic way. These elements include internal torque-generating components such as damped springs, clutches and transmissions, and torque converters. You can also construct and connect your own dynamic elements.
- Set up actuators and sensors to initiate and record body motions, as well as apply external torques to the driveline.

- Connect the SimDriveline motion solver to the machine and configure it. Start the simulation, calling the Simulink solvers to find the motions of the system. Display and analyze the motion.

Connector Ports and Connection Lines

Most SimDriveline blocks have special driveline ports ⊏. You connect driveline ports with driveline connection lines, distinct from normal Simulink lines. Driveline connection lines represent physical rotation axes along which torque is transferred and around which inertias rotate.

- You can connect driveline ports only to other driveline ports.
- The driveline connection lines that connect driveline ports together are not normal Simulink lines, which carry signals or indicate mathematical operations. You cannot connect driveline lines directly to Simulink inports and outports >.
- Two directly connected driveline components must corotate at the same angular velocity.
- You can branch SimDriveline connection lines. When you do so, components directly connected with one another continue to share the same angular velocity. The torque transferred along the driveline axis is divided among the multiple components connected by the branches. How the torque is divided, is determined by the driveline dynamics.

  The sum of all torques flowing into a branch point equals the sum of all torques flowing out.

Inertias and Gears

SimDriveline defines a driveline as a collection of bodies rotating about driveline axes represented by connection lines. The bodies are defined by their rotational inertias. The lines carry the rotational degrees of freedom (DoF) and, unconstrained, rotate freely. Directly connecting one body to another constrains both bodies to rotate at the same angular velocity. A torque applied to one body is effectively applied to both.

You can also ground driveline axes to housings that do not move and that represent infinite effective inertia.

In a real driveline, the bodies can also be connected indirectly by gears that couple driveline axes. The gears constrain the axes to rotate together. These gears can be simple or complex and can couple two or more axes. The gears have two roles:

- They constrain the connected axes to corotate at angular velocities in fixed ratio or ratios.
- They transfer the torques flowing along one or more axes to other axes, also in fixed ratio or ratios.

Complex Driveline Elements

To create more realistic driveline models, you elaborate on simple drivelines consisting of inertias and gears by adding complex mechanical elements that generate torques internally within the driveline, between one axis and another. SimDriveline includes blocks that encapsulate as subsystems entire models of complex driveline elements:

- Clutches that model the locking and unlocking of pairs of driveline axes by applying kinetic and static friction
- Transmission models that incorporate multigear sets and clutches into a single subsystem
- Specialized torque models, such as torque converters, bilateral stops, and damped spring-like torsion

Actuating and Sensing Motion

Sensors and Actuators are the blocks you use to interface between SimDriveline blocks and normal Simulink blocks:

- Actuator blocks impart motion to driveline axes, either at zero time or through the course of a simulation, and impose externally defined torques on the bodies of a driveline.
- Sensor blocks measure the motions of, and the torques transferred along, the axes of a driveline system.

Actuating inputs and sensor outputs are Simulink signals that you can define and use like any other Simulink signal. For example, you can connect a Sensor output to a Simulink Scope block and display the torques in a driveline as functions of time.

Simulating and Analyzing Motion

Once you specify all the rotational inertias of the bodies and interconnect the bodies with gears and other driveline elements, the dynamical problem of finding the system's motion is solvable. To finish a driveline model and prepare it for simulation, you must connect the machine to the SimDriveline environment. The environment defines the solver that integrates the Newtonian dynamics for the system, applying all internal and external torques and constraints to find the motions of the bodies.

Once your model is ready for simulation, you can run it and analyze its motions and internal torques.

Trimming and Linearizing the Motion

In many cases, you do not know the torques necessary to produce a given set of motions. By motion-actuating your driveline and measuring the resulting torques, you can find the torques necessary to produce a specified motion trajectory. This technique inverts the canonical approach to dynamics, which consists of finding motions from torques.

A special case of inverse dynamics is *trimming*. This technique involves searching for steady-state motions of the bodies, when their angular accelerations and the torques they experience vanish. Using Simulink's linearization tools, you can perturb such a steady motion state slightly to find how the system responds to small disturbances. The response indicates the system's stability and suitability for controllers.

Generating Code

SimDriveline is compatible with the Simulink Accelerator (part of Simulink Performance Tools), Real-Time Workshop, and xPC Target. These optional products let you generate C code versions of the models you create originally in Simulink with block diagrams, enhancing simulation speed and model portability.

The presence of clutches in a driveline model creates motion discontinuities and triggers nonphysical algebraic loops in Simulink. These discontinuities and algebraic loops place certain restrictions on code generation.

2

Modeling Drivetrain Systems

This chapter introduces you to modeling drivetrains with SimDriveline. After showing you how you how to access the SimDriveline block library and reviewing the essential rules of connecting blocks and transferring angular motion and torque, it moves you from the simplest gear model to a complete drivetrain simulation in a series of short tutorials. The later examples build cumulatively on the earlier ones but are presented in a way that allows you to learn from one without studying the others.

This user's guide assumes that you are familiar with building models in Simulink. If not, see the Simulink documentation.

Introducing the SimDriveline Block Libraries

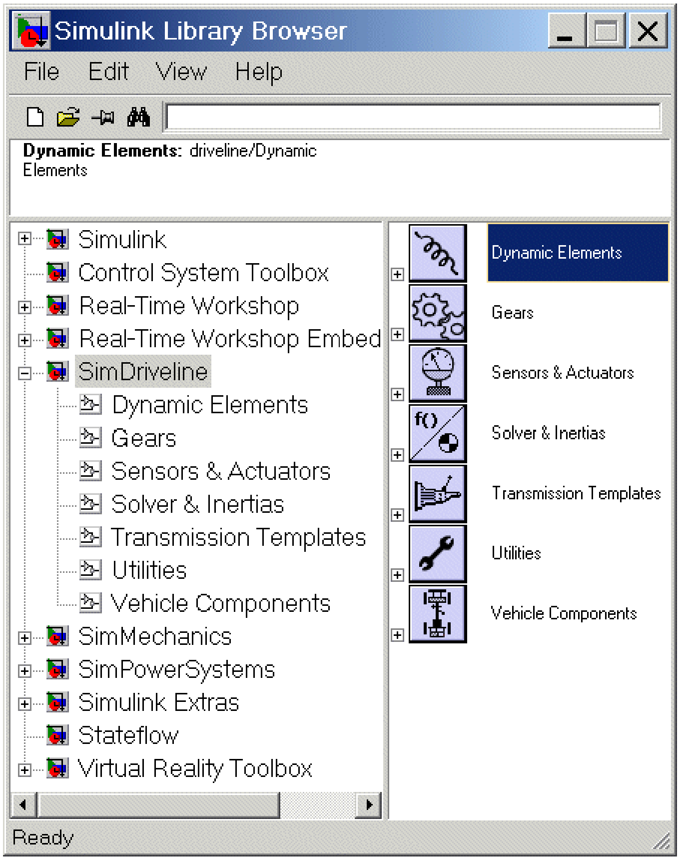
SimDriveline is organized into a set of libraries of closely related blocks. This section shows you how to open these SimDriveline block libraries and explains the nature of each library.

Accesing the SimDriveline Block Library

There are several ways to open the SimDriveline block library on Microsoft Windows and UNIX platforms.

Microsoft Windows Platforms

Microsoft Windows users can access the blocks through the Simulink Library Browser. Expand the SimDriveline entry in the contents tree.



You can also access the blocks directly inside the SimDriveline library in several ways:

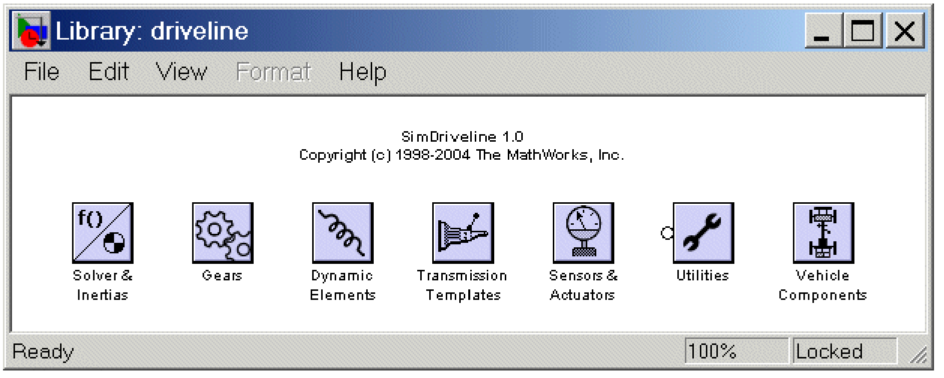
- In the Simulink Library Browser, right-click the SimDriveline entry and select **Open the SimDriveline Library**. The library appears.
- Click the **Start** button in the lower left corner of your MATLAB desktop. In the pop-up menu, select **Simulink**, then **SimDriveline**, then **Block Library**.
- Enter `drivelib` at the MATLAB command line prompt.

UNIX Platforms

UNIX users can click the Simulink icon on the MATLAB menu bar, open the Blocksets & Toolboxes library and then SimDriveline. You can also enter `drivelib` at the command line.

The SimDriveline Library

Once you perform one of these steps, the SimDriveline library opens. This library displays six top-level block groups. You can expand each library by double-clicking its icon.



The next section summarizes the blocks of each library and their use. For explanations of individual blocks, consult the complete SimDriveline block reference chapter.

Using the Libraries

The SimDriveline block library is organized into seven separate libraries, each with a different type of driveline block.

Solver & Inertias

The Solvers & Inertias library provides the Inertia block, which represents a user-defined rotating body specified by its moment of inertia, the fundamental unit of driveline modeling. It also contains the Housing block, which represents an immobile rotational ground.

Finally, the library contains the Driveline Environment block, which configures the driveline settings of a SimDriveline block diagram, and the Shared Environment block, which allows you to connect two driveline block diagrams in a nonphysical way so that they share the same driveline environment settings.

Gears

The Gears library contains blocks that represent simple and complex gears, driveline elements that couple distinct driveline axes and constrain their relative motions. The Gear blocks range from simple two-wheel gear couplings with fixed and variable gear ratios, to complex, multiwheel and multiaxis gears such as planetary and differential gears.

Dynamic Elements

The Dynamic Elements library contains blocks that model critical drivetrain components such as clutches, torque converters, and force elements (damped springs and stops).

Transmission Templates

The Transmission Templates are a set of predesigned transmission examples constructed from gears, clutches, and inertias. You can copy and use these examples in your drivetrain models.

Transmission templates copied into your model are not linked to the block library. You can modify and rebuild these template copies at will.

Sensors & Actuators

The Sensors & Actuators library provides blocks for sensing and initiating the motions of driveline axes and applying and sensing torques along those axes.

Utilities

The Utilities library contains miscellaneous blocks useful in building models.

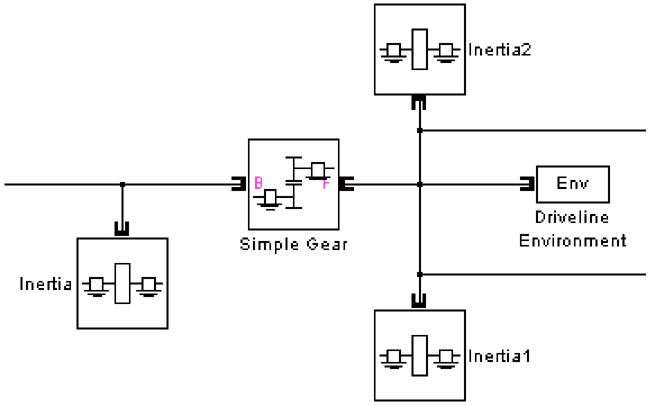
Vehicle Components

The Vehicle Components library contains blocks that represent components of a full vehicle beyond the drivetrain itself. It includes engine models, as well as models of wheeled vehicles and tires in contact with the ground.

Essential Steps to Building a Driveline Model

The demo model of the Chapter 1, “Introducing SimDriveline” chapter illustrates a typical drivetrain system you can model with SimDriveline. It also illustrates the key rules for connecting driveline blocks to each other and the dual roles of driveline connection lines: transferring torque and enforcing angular velocity constraints. You should review these rules before building the tutorial models of this chapter.

- Driveline blocks, in general, feature both driveline connector ports ⊏ and regular Simulink inports and outports >. You connect connector ports to one another and Simulink ports to one another. But you cannot connect a driveline port to a Simulink port.
- The driveline connection lines interconnecting driveline connector ports ⊏ represent driveline axes and enforce physical relationships. Unlike Simulink lines, they do not represent signals or mathematical operations, and they have no inherent directionality.
- A driveline connection line between two ports enforces the constraint that the two driveline components so connected rotate at the same angular velocity. The connection line also transfers any torque applied to a driveline component at one end to the component at the other end. A driveline connection line represents an idealized, massless, and perfectly rigid spinning shaft.
- You can branch driveline connection lines. You must connect the end of any branch of a driveline connection line to a driveline connector port ⊏.
- Branching a driveline connection line modifies the physical constraints that it represents. All driveline components connected to the ends of a set of branched lines rotate at the same angular velocity. The torque transferred along the input driveline axis is split up among the branches. How the torque is split depends on the dynamical details of the system that you are modeling.



Branching Driveline Connection Lines

In this figure, only Inertia1 and Inertia2 are active driveline components. The Driveline Environment block does not use any torque. It does share the angular velocity constraint from the branch point.

Symbolically, the branching conditions on driveline connection lines are:

$\omega = \omega_1 = \omega_2 = \omega_3 \ldots$

$\tau = \tau_1 + \tau_2 + \tau_3 \ldots$

The driveline axes have an implicit directionality. Torque and motion are transferred “down” the driveline from input or drive shafts to output or driven shafts. Certain SimDriveline blocks require explicit directionality and represent it by designating one driveline connector port as the input *base* (B) and the other as the output *follower* (F). Relative motion of driveline axes or shafts, when needed, is measured as follower relative to base.

All motion in SimDriveline models, except when relative motion is explicitly required, is measured in implicit absolute coordinates. An absolute orientation defines zero angle, and an absolute reference frame defines zero angular velocity. The Housing block implements the absolute zero angular velocity and, if connected to a driveline axis, enforces this zero-motion state on that axis.

Coupling Motion and Transferring Torque with Gears

The purpose of a gear set is to transfer rotational motion and torque at a known ratio from one driveline axis to another. The driveline axes represent idealized massless shafts connecting driveline components. This section introduces you to modeling gears and using them to couple bodies rotating on driveline axes.

Coupling Rotational Motion with Gears

A gear set consists of two or more meshed gears corotating at some specified gear ratio(s). The ratio(s) might or might not be constant. The gear ratios determine how angular velocity and torque are transferred from one driveline component to another.

Gear Coupling Rules

Ideal gears mesh and corotate at a point of contact without frictional loss or slippage.

The simplest gear coupling consists of two circular gear wheels of radii r_1 and r_2, spinning with angular velocities $\omega_1$and ω_2, respectively, and lying in the same plane. Their connected shafts are parallel and carry torques τ_1 and τ_2. The *gear ratio* of gear 2 to gear 1 is the ratio of their respective radii: g_{21} = r_2/r_1. The power transferred along either shaft is $\omega \cdot \tau$.

The gear coupling is often specified in terms of the number of gear teeth on each gear, N_1 and N_2. The gear ratio of gear 2 to gear 1 is then $g_{21} = N_2/N_1$ $= r_2/r_1$.

The fundamental conditions on the simple gear coupling of rotational motion are: $\omega_2/\omega_1 = \pm 1/g_{21}$ and $\tau_2/\tau_1 = \pm g_{21}$. That is, the ratio of angular velocities is the reciprocal of the ratio of radii, while the ratio of torques is the ratio of radii. The transferred power, being the product of angular velocity and torque, is the same on either shaft.

The choice of signs indicates that the gears can spin in the same or in opposite directions. If the gears are external to one another (corotating on their respective outside surfaces), they rotate in opposite directions. If the gears are internal to one another (corotating with the outside of the smaller gear meshing with inside of the larger gear), they rotate in the same direction.

Caution Gear ratios should always be strictly positive. If a gear ratio in SimDriveline vanishes or becomes negative, the simulation stops with an error.

Generalized Gear Coupling Rules

You need the general ideal gear coupling conditions if you are coupling gears that are not constant in radii, not lying in the same plane, or not circular.

The general velocity constraint requires that the linear velocities of the gears at the point of contact be the same. This is a vector condition on the angular velocities ω_1 and ω_2 and the radius vectors $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$: $\omega_1 \times \boldsymbol{r}_1 = \omega_2 \times \boldsymbol{r}_2$. The alternative form in terms of the number of gear teeth is equivalent to this linear velocity constraint. In order for the gear teeth to mesh, the number of teeth per unit length of gear circumference must be the same on the two gears.

The general torque condition arises from the force equilibrium at the point of contact. If there is no linear motion of the whole gear assembly, the forces at contact $\boldsymbol{F}$ must be equal and opposite. The ratio of torques is then $|\tau_2|/|\tau_1| = |\boldsymbol{r}_2 \times \boldsymbol{F}|/|\boldsymbol{r}_1 \times \boldsymbol{F}|$.

The power transferred along either shaft is conserved across ideal gear couplings: $\omega_2 \cdot (\boldsymbol{r}_2 \times \boldsymbol{F}) = \omega_1 \cdot (\boldsymbol{r}_1 \times \boldsymbol{F})$.

Coupling Two Spinning Inertias with a Simple Gear

In this example, you couple two spinning inertias, first, along a single shaft (driveline axis), so that they spin with the same angular velocity; then spinning along two shafts and coupled by a gear so that they spin at different velocities; and finally, coupled by a gear and actuated by an external torque, spinning at different rates and experiencing different torques. You use the most basic blocks in SimDriveline, such as `Inertia`, `Simple Gear`, and `Driveline Environment`.

Modeling Two Spinning Inertias

Here you create the first version of the simplest driveline model, two inertias spinning together along the same axis. Open the SimDriveline and Simulink block libraries and a new Simulink model window.

1. Drag and drop two Inertia, two Motion Sensor, and one Initial Condition blocks into the model window.
2. Every topologically distinct driveline block diagram requires exactly one Driveline Environment block, found in the Solver & Inertias library of SimDriveline. Copy one such block into your model.
3. From the Simulink library, drag and drop a Scope and a Mux block. Then connect the blocks as shown.
4. Open the Initial Condition block. In the **Initial angular velocity** field, replace its default `0` entry with `pi` radians/second (rad/s). Click **OK**.

   If you do not connect Initial Condition blocks to a driveline axis, SimDriveline assumes by default that the axis starts the simulation with zero angular velocity. You must ensure that the initial angular velocities of your driveline are consistent with one another. If they are not, the simulation stops with an error.
5. Open the Scope block and start the simulation. The two angular velocities are constant at 3.14 radians/second.

Coupling Two Spinning Inertias with a Simple Gear

Now you modify the model you just created by coupling the two spinning inertias with a simple, ideal gear with a fixed gear ratio.

1. From the SimDriveline block library, drag and drop a Simple Gear block into your model. Open the block. Leave the default follower-base gear ratio value at `2`. Unselect the **Follower and base rotate in opposite directions** checkbox and click **OK**. The simple gear is then two wheels corotating in the same direction, with the smaller wheel inside the larger. Reconnect the blocks as shown.

Leave the initial angular velocities at `pi` in the Initial Condition block. SimDriveline will automatically find the correct initial angular velocity for Inertia1.

2. Open the Scope and start the simulation. The two angular velocities are constant at 3.14 and 1.57 radians/second for Inertia and Inertia1, respectively. The follower-base gear ratio is two, and the angular velocity of Inertia1 is half that of Inertia and with the same sign, because the two bodies are spinning in the same direction.

3. Now select the **Follower and base rotate in opposite directions** checkbox. The simple gear then becomes two wheels corotating in opposite directions, with the two wheels meshed on their respective outer surfaces.

4. Restart the simulation. The two angular velocities are 3.14 and -1.57 radians for Inertia and Inertia1, respectively. The second angular velocity is half the first and with opposite sign, because the two bodies are spinning in opposite directions.

5. Finally, again unselect the **Follower and base rotate in opposite directions** checkbox.

Torque-Actuating Two Coupled, Spinning Inertias

In the final version of the simple gear model, you actuate the inertias with an external torque, instead of starting them with fixed initial angular velocities. The external torque varies sinusoidally. You can find this completed model in the demo `drive_sgear`.

1. From the SimDriveline block library, copy a Torque Actuator and two Torque Sensor blocks. From the Simulink block library, drag and drop a second Scope block, a second Mux block, and a Sine Wave block.

2. Disconnect the Inertia blocks from the Simple Gear and insert the Torque Sensors. Connect the other blocks as shown.

3. Disconnect and delete the Initial Condition block. SimDriveline will now impose zero angular velocities on the two axes.

4. Open both Scope blocks and start the simulation.

The measured torques and angular velocities vary sinusoidally. The angular velocity of Inertia1 is half that of Inertia, as you saw in the previous models.

But the torque in the second shaft is twice that in the first, as required by the laws of gear coupling.

If you select the the **Follower and base rotate in opposite directions** checkbox in Simple Gear and restart the simulation, the same angular velocities and torques result, except that the values associated with Inertia1 and the second shaft are negative, because the second body and second shaft are spinning in opposite directions.

Sensing and Actuating Motion and Torque

The Sensor and Actuator blocks you use in the preceding models illustrate their dual nature: they act as driveline components themselves, but also let you connect driveline blocks with the rest of Simulink.

- Sensor & Actuator blocks have both driveline connector ports ⊏ and normal Simulink ports >. You can extract sensor signal information with a block's Simulink outports. You can actuate motion or apply external torques by feeding in actuation signals with a block's Simulink inports.

  Many other SimDriveline blocks feature Simulink ports for inserting and measuring signals.
- You connect a Torque Sensor *along* a driveline axis, by placing it in series with other driveline components.
- You connect the other Sensor and Actuator blocks *across* a driveline axis, by branching the driveline connection line off to one side and connecting this secondary line to the block; or by connecting the block to the end of a driveline axis.

Coupling Two Spinning Inertias with a Variable Gear

You can modify the simple gear model further by replacing the fixed-ratio gear with a gear whose gear ratio varies in time. You specify the gear ratio variation with a Simulink signal. Start with the simple gear model you built in the preceding section or by opening and editing the `drive_sgear` demo.

1. From the SimDriveline block library, drag and drop a Variable Ratio Gear block and replace the Simple Gear block with it. Open Variable Ratio Gear and unselect the **Follower and base rotate in opposite directions** checkbox.

2. The Variable Ratio Gear block accepts the continuously varying gear ratio as a Simulink signal through the extra inport labelled `r`. For this example, create a Simulink signal for the gear ratio with a Signal Builder block from the Simulink block library. Build a signal that rises with constant slope from one to two over ten seconds. Then connect the Signal Builder block to the `r` port.

3. Leave the other, original settings of the simple gear model unchanged. Open both Scopes and start the simulation.

The ratios of angular velocities and torques start at one, because the initial gear ratio is one. But as the gear ratio increases towards two, the angular velocity of Inertia1 becomes smaller than that of Inertia, while the associated torque in the second shaft becomes larger than that in the first shaft. Because of the changing gear ratio, the motion and the torques are no longer strictly sinusoidal, even though the acutating external torque is.

The `drive_vgear` demo is a full model of this type. To learn more about how to use variable gears, including the Coriolis acceleration, consult the `Variable Ratio Gear` block reference page.

Coupling Three Spinning Inertias with a Planetary Gear

You can further modify the simple gear model and use it as a starting point for studying more complex gear sets. One of the most important is the planetary gear, which has three wheels, the ring, the sun, and the planet, all held in place by a common carrier body. The planetary gear is interesting in its own right, but also important because it is a common component in complex, realistic transmissions.

1. Replace the Simple Gear in your model with a Planetary Gear from the SimDriveline block library. A planetary gear splits input angular motion from the carrier between the ring and sun wheels, each connected to their respective bodies.

2. Copy another Inertia and another Motion Sensor as well. Connect the blocks into the new diagram as shown.

3. Set the **Ring/Sun gear ratio** in Planetary Gear to `2`. Open the Scope and start the simulation to observe the angular velocities of the ring, carrier, and sun, from largest to smallest.

4. To see the ring and sun wheels spinning alone, you can lock the carrier and drive the ring wheel. Copy a Housing block from the SimDriveline block library. Disconnect and delete Inertia, replacing it on the carrier driveline axis with Housing, and reconnect the Driveline Environment block to this connection line.

5. Insert a Motion Actuator and move the Sine Wave block next to it. Connect it to the inport.

6. Open the Scope and start your model. Observe the angular velocities of the ring, carrier, and sun.

The carrier, connect to Housing, does not move. The ring is driven with a sinusoidal torque, and the sun responds by spinning in the opposite direction (ring and sun are external gear wheels to one another) at twice the rate. The ring wheel has twice the radius (or twice the number of teeth) as the sun, so it spins half as fast.

To learn more about modeling planetary gears with SimDriveline, see the `Planetary Gear` block reference page.

Controlling Gear Couplings with Clutches

The most important requirement of a practical drivetrain is the ability to transfer rotational motion and torque among spinning components at different speeds and gear ratios. A single set of gears is usually not sufficient to accomplish this. Clutches are the critical component that allow the drivetrain to selectively transfer motion and torque at different gear ratios under manual or automatic control.

This section explains how to model and use clutches in driveline models.

- "Engaging and Disengaging Gears with Clutches" on page 2-16 explains the essential purpose of a clutch and how to model a simple clutch that couples two shafts with a gear.
- "Modeling Realistic Clutch Systems with Loss" on page 2-19 adds greater realism with non-clutch frictional losses.
- "Braking Motion with Clutches" on page 2-21 shows how to gradually halt spinning motion with a clutch.

Engaging and Disengaging Gears with Clutches

A common problem in drivetrain design is transferring motion and torque at different fixed gear ratios. Drivetrains are typically designed to switch among a set of discrete gear ratios. Implementing the switch from one gear ratio to another requires gradually disenganging one set of driveline couplings and engaging another set. Clutches allow you to gradually engage and disengage driveline shafts from one another. The `Controllable Friction Clutch` block of SimDriveline represents a standard surface friction-based clutch that models this behavior.

You can also model gradual motion-torque transfer with the Torque Converter block, which models fluid viscosity instead of surface friction.

How a Clutch Works

A clutch makes two shafts spinning at different rates spin at a single rate by applying forces that accelerate one shaft and tend to decelerate the other. The most common way for a clutch to accomplish this is with surface friction. Such a clutch can operate in one of three modes of motion:

- Disengaged: the clutch applies no friction at all.
- Engaged but unlocked: the clutch applies kinetic friction, and the two shafts spin, in general, at different rates.
- Engaged and locked: the clutch applies static friction, and the two shafts spin together.

A clutch consists of mated frictional surfaces overlapping one another and connected on either side to a shaft. If the clutch is disengaged, the frictional surfaces have no contact and the shafts spin independently. To engage the clutch, a moderate amount of contact between two surfaces is induced by applying clutch pressure (a force normal to the surfaces). The two surfaces in contact and moving relative to one another experience kinetic friction, which causes them to narrow their relative velocity. The faster surface tends to slow down (unless an external torque is acting) and the slower one speeds up. At some critical combination of reduced relative speed and pressure (normal force), the clutch locks, so that the two shafts are spinning at the same rate. The locking of the shafts is controlled by static friction, which holds the shafts together as long as sufficient normal force is applied and no relative torque is large enough to overcome the locking. If the clutch unlocks but is still engaged, it again applies kinetic rather than static friction.

The transition between the unlocked and locked modes is a discontinuous change of motion. Modeling a clutch locking or unlocking requires searching for the correct combination of pressure and torque acting on the clutch. SimDriveline determines the instant of loacking and unlocking during a simulation by accurate zero-crossing detection and repeated testing or *mode iteration*.

Enganging and Disengaging a Gear with a Clutch

Here you construct a simple model that simulates a gear being engaged, then disengaged, by a clutch. Torque and motion are transferred from one shaft to another over a finite time interval. Start with the simple gear model of the last section or with the `drive_sgear` demo. The completed clutch model is the `drive_sclutch` demo.

From the SimDriveline block library, you need a Controllable Friction Clutch block. Also, copy a Signal Builder and a Constant block from the Simulink block library. Remove the Torque Sensor blocks, insert the Clutch between

Inertia and Simple Gear, then reconnect the connection lines. In the Clutch dialog, select the **Show mode signal port** check box, but leave the other defaults. Rearrange and connect the blocks as shown here.

Use the Constant block as the input torque signal in place of the sinusoidal signal. Reconfigure Mux and the first Scope blocks to accept three signals, the two angular velocities and the clutch pressure. Connect the second Scope to display the Controllable Friction Clutch mode signal.

Signal Builder specifies the clutch pressure signal, which is normalized between zero and one. (The **Peak normal force** field in Controllable Friction Clutch determines the maximum clutch pressure.) Open Signal Builder and construct the following signal.

Time Range (seconds)	Signal Value
0 – 2	0
2 – 4	0 – 0.8 with constant slope
4 – 6	0.8
6 – 7	0.8 – 0 with constant slope
7 – 10	0

Open the Scopes and start the simulation. The normalized clutch pressure signal follows the profile you created in Signal Builder. From zero to two seconds, the velocity of Inertia increases linearly, because it is subject to a constant torque. At two seconds, the clutch begins to engage, and Inertia1 begins to spin. The velocity of Inertia continues to rise, although at a slower rate, because the two inertias now share the external torque. At four seconds, the pressure reaches its maximum, and the clutch locks. Inertia1 continues to speed up at a constant acceleration. At six seconds, the clutch begins to disengage as the pressure drops. Inertia and Inertia1 continue to accelerate with the applied torque. The clutch unlocks at 6.77 seconds and fully disengages at seven seconds. (The clutch unlocks a little before completely disengaging because the pressure, even before vanishing, becomes too small to maintain the lock.) Inertia is still accelerating. But Inertia1, now free of the drive shaft and its torque, no longer accelerates and instead spins at a constant rate without frictional loss.

The Controllable Friction Clutch mode signal indicates the relative motion of its two connected shafts. From zero to four seconds, the two shafts are moving relative to one another. The follower (driven) shaft is slower than the base (drive) shaft, so the mode signal is -1. Once the two shafts lock, their relative velocity is zero, and the mode signal switches to 0. At 6.77 seconds, they unlock, and the drive (base) shaft starts accelerating faster than the driven (follower) shaft. The mode signal switches back to -1.

While the two shafts are locked, between four and 6.77 seconds, Inertia and Inertia1 spin together. But the Simple Gear, with a gear ratio of two between follower and base, transforms Inertia1's velocity to half that of Inertia. To see the two Inertias locked and spinning at the same rate, remove Simple Gear and connect Inertia1 directly to the Controllable Friction Clutch. Change the **Peak normal force** value in Clutch to `2.5` (Newtons). Restart the simulation. Inertia and Inertia1 now spin at the same rate while the clutch is locked.

Modeling Realistic Clutch Systems with Loss

To make your clutch system model more realistic, you should add frictional damping to the spinning shafts. Here you add a kinetic friction torque proportional to the angular velocity to both sides of the clutch. A simple way to do this is to create a friction subsystem that applies such a torque to any driveline axis it is connected to. Then you can copy the subsystem and modify your existing clutch model by connecting the two copies on either side of the clutch.

Creating a Torque Damping Subsystem

The frictional torque is $\tau_{fric} = -\mu\omega$, where μ is the frictional proportionality constant. To apply the frictional torque proportional to the velocity, you need to

1. Measure the angular velocity of the driveline axis
2. Multiply it by $-\mu$, because the frictional torque opposes the motion
3. Apply the resulting torque back to the driveline axis

Copy Motion Sensor and Torque Actuator blocks and, from the Simulink library, a Gain block, into your model window. Connect the angular velocity port `Vel` to the inport of the Gain block and the outport of the Gain block to the torque inport of the Torque Actuator block. Set the **Gain** value in the Gain dialog to `-0.3`, leaving the other defaults.

With your cursor, select the connected Torque Actuator-Gain block pair, and create a subsystem. Call it `Damper`. When you create the subsystem, the single port that appears on its block is a driveline connector port ⊏, not a Simulink port >. Now create an extra copy of `Damper`.

Connecting and Simulating the Damped Clutch System

Connect the two Damper subsystems to the driveline of your previous clutch model as shown.

Change the simulation time to 20 seconds. Then open the Scope blocks and click **Start**. Readjust the horizontal axes of the Scopes with **Autoscale** to see the full plots. The clutch pressure and external torques are applied as before. But the shaft rotations are different now because of the damping.

Inertia1, as before, begins to spin when the clutch starts to engage at two seconds. After the clutch locks at four seconds, the body continues to accelerate, but at a slower rate than it did without damping. At six seconds, the clutch begins to disengage and completely disengages at seven seconds. Unlike the friction-free case, Inertia1, subject to friction, now starts to slow down. Its angular velocity drops exponentially with time once the external torque is removed.

The behavior of Inertia is more complex. It begins to spin up, but at a lower rate than before, because of the damping. Between two and seven seconds, Inertia has to share the external torque with Inertia1 via the Controllable Friction Clutch and the Simple Gear. After seven seconds, the external torque applies to Inertia alone. It continues to accelerate, but at an ever-slowing rate, because of the damping. If you let the simulation run without stopping, Inertia will approach its terminal angular velocity, a state where the frictional torque exactly balances the externally applied torque. The terminal velocity is $\omega_{term} = \tau_{ext}/\mu$ or 1/0.3 = 3.3333 radians/second in this case. The Scope plot shows this terminal value.

Braking Motion with Clutches

A special case of transferring motion occurs when you want to brake the spinning of a driveline component, slowing it down until it stops. The common way to brake the motion is to couple the spinning component to a fixed housing, which effectively has infinite inertia and is represented in SimDriveline by a Housing block. Because the housing cannot move, a driveline axis locked to a housing also cannot move. You can implement the gradual engagement or disengagement of a driveline component with a housing using a clutch, just as you use a clutch to gradually couple or uncouple two spinning shafts.

Braking with a Double-Clutch System

The `drive_clutch_engage` demo model is a elaboration on the preceding models of this chapter and features two clutches, one of which acts as a brake. The model also includes frictional damping for greater realism. The simulation time is set to `inf` (infinity).

This model again uses the basic structure of inertia-clutch-gear-inertia. The first body, Inertia, is still driven by an external torque, and the initial velocities are still zero. There is, however, another clutch for the second body, Inertia1, that can couple Inertia1 to the Housing and bring it to a stop. Another new feature, compared to the preceding models, is the switching assembly made of the Clutch Switch and Flipper blocks. You can flip this switch to apply a constant clutch pressure signal to either Gear Clutch or Brake Clutch. The two Damper subsystems are identical to those you constructed in “Modeling Realistic Clutch Systems with Loss” on page 2-19, except that the frictional constants, the **Gain** values of the Gain blocks, are set to `-0.1`.

Start the model with the Clutch Switch set to one. The clutch pressure is then applied to Gear Clutch, which engages and locks the driver and driven shafts and causes Inertia and Inertia1 to rotate together. The angular velocity of Inertia1 (2.5 radians/second) is half that of Inertia (5 radians/second) because the gear ratio of the Simple Gear block is two, follower to base. In this switch mode, no clutch pressure is applied to Brake Clutch, which remains unengaged. The mode of Brake Clutch is then -1, because Brake Clutch’s follower, the Housing block, is at rest, while the base, Inertia1, is spinning. The mode of Gear Clutch is zero, because its base and follower, the driver and driven shafts, are locked together. After an initial transient, the system settles into a steady state of motion where the external torque is exactly balanced by the frictional losses. The effective frictional constant, with two dampers, is 0.2. With an external torque of 1 Newton-meter, the terminal angular velocity of Inertia is then be $\omega = 1/0.2 = 5$ radians/second.

With the simulation running, now change the Clutch Switch to zero to disengage Gear Clutch and engage Brake Clutch. The system undergoes another transient while Gear Clutch disengages and Brake Clutch engages. The angular velocity of Inertia and the driver shaft settles down to a new steady state of 10 radians/second, twice its old speed. The mode of Gear Clutch is now -1, because the driven shaft (follower) is not moving, while the driver shaft (base) continues to spin. Because Gear Clutch is now disengaged, Inertia is no longer subject to the second frictional damping block, Damper1. The effective frictional constant drops in half, to 0.1, and the terminal velocity doubles. At the same time, Inertia1 is no longer receiving torque through Gear Clutch . But Brake Clutch is engaged and couples Inertia1 to the immobile Housing. Once engaged, the kinetic friction of Brake Clutch and Damper1 bring the driven shaft and Inertia1 to a stop. Because it does not lock, Brake Clutch's mode remains -1.

To see the transient behavior at simulation start and when you switch the clutches,

- Start the simulation and let it run for a short time. Then switch Clutch Switch to the other mode.
- After another short time, stop the simulation. Use the **Autoscale** feature of the Scopes to capture the entire simulation sequence. The transients from the starting behavior and the switching transition will be visible.

For example, in these plots, the model was started with Clutch Switch set to one (Gear Clutch locked, Brake Clutch disengaged, no braking). The velocities quickly climb to their steady-state values. Then Clutch Switch was changed at about 310 seconds of simulation time. Gear Clutch disengaged and Brake Clutch engaged, braking the driven shaft. The driver shaft's angular velocity rises from 5 to 10 radians/second. The driven shaft's angular velocity drops to zero.

Modeling Transmissions

In a real drivetrain, you couple an input or drive shaft to one of many output or driven shafts, or to one driven shaft with a choice of several gear ratios. The drivetrain then requires several clutches to switch between gears. You couple one of the driven shafts or one of the gear sets by engaging one of the clutches. You then switch to another output shaft or another gear ratio by disengaging one clutch and engaging another.

You can also engage more than one clutch at a time in order to use multiple gear sets simultaneously. Realistic transmissions engage multiple gear sets at the same time to produce a single effective gear ratio. Changing gears then requires disengaging one set of clutches and engaging another set. You specify the set of clutches to engage and unengage for each desired gear ratio in a *clutch schedule*. Designing a clutch schedule and shaping and sequencing the clutch pressure signals frequently constitute the most difficult part of transmission design. A realistic transmission model must also include losses due to friction and imperfect gear meshing.

This section explains how to model transmissions, first by creating a transmission model from gears and clutches, then by using the SimDriveline library of predesigned transmission subsystems. One such predesigned transmission, the Simpson 4-speed transmission, is the basis of another example.

- "A Simple Two-Speed Transmission with Braking" on page 2-23
- "Introducing the Transmission Templates Library" on page 2-28
- "A Simpson 4-Speed Transmission Driveline with Braking" on page 2-29

A Simple Two-Speed Transmission with Braking

The demo model `drive_strans` contains a driveline system that makes up a simple but complete transmission.

The model is an elaboration of the `drive_clutch_engage` demo model presented in "Braking Motion with Clutches" on page 2-21. This model also contains two driveline shafts or axes, with an actuating torque applied to the driven shaft. Both the driver and the driven shafts are subject to, respectively, large and very small kinetic damping torques. (The kinetic torque constants μ are 0.1 and 10^{-5} Newton-seconds/radian, respectively, in Damper1 and Damper2.) In the steady state, the driving and damping torques balance one another, and the two shafts spin at constant rates. (If braking is engaged, the driven shaft is stopped, as before.) But there are now two selectable gears to couple the two axes, instead of one.

This transmission model couples the gears in a simple way, with each gear and the brake associated with its own respective clutch. Coupling one gear requires engaging and locking its corresponding clutch, while ensuring that the other two clutches are disengaged. Switching on the brake requires disengaging the two gear clutches and locking the brake clutch.

Setting Up the Gear, Brake, and Clutch Sets

The two gears are Simple Gear blocks with different gear ratios, each connected in series with its corresponding clutch. The two gear-clutch pairs are coupled in parallel, and this parallel assembly then couples the driver shaft to the driven shaft, with their two spinning inertias. One gear is a “low” gear, the other a “high” gear. The “low” and “high” labels, following common usage for automobile gears, refer, not to the gear ratios, but the angular velocity ratios. The ratio of speeds in a gear is the *reciprocal* of the gear ratio. Keep this distinction in mind to prevent confusion.

The “low” gear is the Simple Gear 5:1 block, which can be coupled by engaging its corresponding clutch, modeled by the Lo Gear Clutch block. The gear ratio is 5:1, so that the ratio of output to input (follower to base) angular speeds is 1/5. Hence the name “low” gear. Such a gear, by the same token, has a high torque transfer ratio of five, from base to follower. In an automobile, a “low” gear like this is used to accelerate the vehicle from a stop by transferring a large torque down the drivetrain from the engine.

The “high” gear is the Simple Gear 2:1 block, coupled by engaging its own clutch, represented by the Hi Gear Clutch block. The gear ratio is 2:1, and the angular velocity ratio of follower to base to 1/2, or 5/2 times the ratio in the “low” gear. Hence the name “high” gear. The torque transfer ratio is only two from base to follower. A automotive “high” gear is used for milder acceleration or coasting once a vehicle is moving at a significant speed. The vehicle acceleration generated by this gear is less than that generated by the “low” gear.

While either Gear Clutch is engaged, the Brake Switch is disabled. You can start braking and bring the driven shaft to a stop by engaging Brake Clutch. This clutch, once locked, holds the driven axis fixed relative to the Housing. The driver shaft continues to spin, subject to the competing driving and damping torques. In this transmission, the brake is completely disabled if either gear clutch is engaged. Disengaging the gears puts the transmission into “neutral” and allows you to use the Brake Switch to apply or not apply brake clutch pressure.

Clearly, this simple transmission is based on mapping each transmission state one-to-one with an engaged clutch. You cannot engage more than one clutch at a time without creating conflicts between gear ratios or between the driver shaft and the Housing. In a real transmission, such conflicts generate internal stresses and might destroy the machine. In a SimDriveline, such conflicts cause the simulation to stop with an error.

Controlling the Transmission State with a Clutch Schedule

The requirement to engage a certain clutch or set of clutches and disengage others, in order to both implement transmission functions and to avoid physical conflicts between gears, is the basis for all clutch schedules. Simulink provides a number of ways to implement clutch schedules, depending on the complexity of the transmission and how much realism you require for the clutch pressure signals.

Caution You must check every transmission’s clutch schedule to implement the transmission various states correctly and to avoid physical conflicts among gear sets. You must also check clutch pressure signal profiles to make sure that any transmission’s clutches are engaged, locked, unlocked, and disengaged in a physically realistic and conflict-free fashion. In SimDriveline, unphysical or conflicting clutch schedules and clutch pressure signals lead to simulation errors.

Avoiding such conflicts leads, for the `drive_strans` model, to a unique clutch schedule.

Clutch Schedule for the Simple Two-Speed Transmission

Transmission State	Clutch1 State	Clutch2 State	Clutch3 State
Neutral/Braked	Locked	Disengaged	Disengaged
Low Gear	Disengaged	Locked	Disengaged
High Gear	Disengaged	Disengaged	Locked

The model contains a simple Clutch Control subsystem to implement the clutch schedule and to output (or, in the case of the brake, enable) the clutch pressure signals to lock each clutch as needed.

In this simplified and unrealistic clutch control model, the clutch pressure signals are just constants: one to engage and lock a clutch, and zero to disengage it. (A clutch pressure signal is normalized from zero to one. You specify the absolute or physical frictional torque in each Controllable Friction Clutch dialog.) The brake signal is not a pressure, but only an enabling signal for the Brake Switch in the main model. The table of these constants, for each transmission state, is contained in the Clutch Schedule Table block, customized from the Simulink LookupND Direct block, which is discussed in the Using Simulink documentation. Open the dialog to see this table.

The table is indexed, starting from zero, in both row and column. An input signal of zero causes the block to output the first column of table values; a value of one outputs the second column, and a value of two, the third column. The first column applies zero pressure to the two gear clutches and enables the Brake Switch in the main model. Turning this switch on applies full pressure to the brake clutch. Turning it off releases the brake pressure. The second column applies full pressure to the low gear clutch and zero pressure to the high gear and the brake clutches. The third column applies full pressure to the high gear clutch and zero pressure to the other two.

These different table columns are activated by changing the positions of the two Manual Switch blocks, labeled Gear Switch and Neutral Switch. Putting the transmission into “neutral” and enabling the brake (upper position of Neutral Switch) feeds a zero signal to Clutch Schedule Table and activates the braking schedule. Switching the brake to off (lower position) allows the Gear Switch schedule signal to pass through instead. This signal has value one for the low gear and two for the high gear.

This clutch control subsystem is adequate for a simple model like this one, but not realistic. A full clutch control model requires realistic clutch pressure signals that rise from and fall back to zero in a smooth way. See for more about modeling realistic clutch control pressures.

Running the Model, Switching Gears, and Braking

Start the model. Its initial transmission state is low gear. The driven shaft spins at one-fifth the rate of the driver shaft. Change the Gear Switch from Low to High, and observe how the driven shaft velocity increases in the Shaft Velocities scope. The driven-to-driver ratio is now one-half. (The driver shaft velocity decreases slightly, because it experiences the damping torque on the driven shaft differently depending on which gear is engaged.) Change the Gear Switch back to Low, then observe that the driven shaft again spins more slowly. At the same time, while you switch the gears back and forth, the clutches, as shown in the Clutch Modes scope, switch from Lo Gear Clutch being locked, to the Hi Gear Clutch being locked, and back. When one is locked, the other is unlocked.

Now enable the brake by changing the Neutral Switch to the upper position. The two gear clutches unlock and disengage. The driven shaft, subject to very light damping, starts to gradually slow. The Brake Clutch remains unengaged. By turning the Brake Switch to on, you can switch the Brake Clutch to the locked mode and bring the driven shaft to an immediate and complete stop. The driver shaft continues to spin at 10 radians/second.

Running the Model without Clutch Mode Iteration

In realistic transmissions, the pressure signal applied to one clutch is often determined by the locked/unlocked mode of another clutch. Simulating such a system requires SimDriveline to stop simulation time briefly and enter an algebraic loop to search for a simultaneously self-consistent state of all clutches.

This transmission is simple and non-self-referential, insofar as each clutch is controlled by external signals only. No clutch is controlled by the mode of another clutch. In such a case, you do not need mode iteration for the clutches, because SimDriveline does not have to search for a collective self-consistent state of all clutches. The externally-imposed clutch schedule does that automatically.

To turn off clutch mode iteration, open the model's Driveline Environment block (the block with the "Env" icon) and select the **Disable mode iteration for clutch locking** check box. When you start the model again, it will run faster and without algebraic loops.

Disabling clutch mode iteration to avoid algebraic loops is sometimes necessary when you are using code generation-based simulation options in Simulink and Real-Time Workshop. See the `Controllable Friction Clutch` and `Driveline Environment` block reference pages for more details.

Introducing the Transmission Templates Library

The Transmission Templates library of SimDriveline provides examples of complete, working multiclutch transmission subsystems. The blocks in this library are unmasked. When you double-click one, the subsystem opens directly, allowing you to inspect the component blocks.

The Transmission blocks are not library-linked. Once you make a copy from the library to your model, you are free to modify your copy.

Each type of transmission block has its own clutch schedule, which you can view by opening the subsystem, then opening the clutch schedule block inside. (The corresponding block reference pages also list the clutch schedules for each Transmission block.) Properly engaging a transmission in a particular gear setting requires engaging a certain number of clutches, no more and no fewer. Locking too few or too many clutches, or engaging the wrong clutches, will lead to conflicting gear meshings and simulation errors. You can disengage a transmission by turning all clutch pressure signals to zero.

Customizing and Using Transmission Blocks

With no extra steps to unlink a transmission block from its library, you can easily modify the transmission block copies in your models. You will typically need change gear ratios, clutch pressures, and gear shaft inertias in any case. If you open the Transmission block to view the underlying subsystem, you can proceed with modifying blocks at will. Observe certain cautions when modifying the transmission subsystem component blocks.

- Do not remove any of the gear shaft inertia blocks or set their inertia values to zero. These inertias are needed for realistic simulation and preventing acceleration singularities when torques are applied.
- The clutch schedule for any transmission type specifies those clutches that must be engaged and those that must be free at any instant for the transmission to be properly in gear. Make sure that your clutch pressures reflect this requirement. Set all clutch pressures to zero only if you want to disengage the transmission completely (place in neutral). Do not engage any more or fewer clutches than needed, at any time during simulation.
- If you want to redesign the transmission, by adding or removing gears, you must consider whether you need as well to add or remove clutches and redesign the clutch schedule. You also might need to add or remove gear shaft inertias.

The next section presents a driveline model based on the Simpson 4-speed transmission model of the Transmissions library.

A Simpson 4-Speed Transmission Driveline with Braking

The `drive_simpson` demo model builds on the previous clutch and transmission models with a more realistic transmission. It uses the Simpson 4-Speed transmission block from the Transmissions library to transfer motion and torque from one shaft and inertia to another. The model is otherwise similar to `drive_strans`.

A Torque Driver subsystem feeds a constant driving torque to the driver shaft (Inertia1). Two damping subsystems apply heavy and light kinetic friction to the driver and driven shafts, respectively. The three Scopes measure the shaft velocities, clutch pressures, and clutch modes, respectively. The model pre-load function defines essential parameters in the workspace. You can view these by opening the Workspace Variables block or opening the **Callbacks** tab of the **File : Model Properties** dialog. The Simpson 4-Speed transmission subsystem couples the driver to the driven shaft (Inertia2). The Clutch Control subsystem lets you engage or disengage the clutch and switch among various gear settings. A brake clutch and fixed housing allow you to brake the driven shaft if the transmission is disengaged.

For clarity, the model's major signal buses have been bundled as vectors and directed using Goto and From blocks. The Scopes are collected in the Scopes subsystem for convenience.

Configuring the Simpson Transmission Subsystem

The Simpson 4-Speed transmission block used in `drive_simpson` has default settings for its component Gears, Clutches, and Inertias, with some exceptions. Certain parameters are changed for greater realism and are referenced to variables defined in the workspace when the model opens. These variables are used in the four Simpson Clutch blocks A, B, C, and D.

Simpson 4-Speed Transmission Clutch Variables

Workspace Variable	Meaning
`num_fric_surf`	Number of frictional surfaces in each clutch
`eff_tor_rad`	Effective torque radius in each clutch (m)
`peak_normal`	Peak normal force on clutch surfaces (N)
`fric_coeff` (matrix)	Kinetic friction coefficient as a function of the relative angular velocity of the clutch shafts

Within the Simpson transmission subsystem, open the Clutch Schedule block to see the table of gear settings, clutch lockings, and gear ratios. The `Simpson 4-Speed` block reference page also discusses the clutch schedule. There are four distinct gear settings, each with a different effective gear ratio. For the transmission to be properly engaged and transmit torque and motion, exactly two clutches must be locked at any instant. Unlocking all the clutches simultaneously puts the transmission into neutral (no motion or torque transfer).

Close the transmission subsystem and return to the main model window. The main model's Damping subsystems use these variables for frictional damping of the driving (engine) and driven shafts coupled across the transmission.

Drive Shaft Damping Coefficients

Workspace Variable	Meaning
eng_damping	Driver (engine) shaft kinetic friction coefficient (N·m·s/rad)
driven_damping	Driven shaft kinetic friction coefficient (N·m·s/rad)

Programming the Clutch Schedule Logic

From the main model, open the Clutch Control subsystem. The Clutch Schedule Logic block embodies the Simpson 4-Speed clutch schedule as a truth table. Each row represents a different gear setting. You select a particular row for output by inputting a set of 1's and 0's that specify the row value as a binary number.

Simpson 4-Speed Clutch Schedule Logic

Gear Setting	Truth Table Row	Truth Table Value
1	$00_2 = 0$	1 0 0 1
2	$01_2 = 1$	1 0 1 0
3	$10_2 = 2$	1 1 0 0
4	$11_2 = 3$	0 1 1 0

In the order of the Simpson Clutches — A, B, C, and D, respectively — the sequence of 1's and 0's indicates which clutchs are locked (1) and which are free (0). These boolean values are then converted into normalized clutch pressure signals.

Running the Simpson Transmission Model — Changing Gears

Open the Scopes subsystem, then the individual Scope blocks. Close the Scopes subsystem. With the Scopes, you can observe the angular velocities of the driving and driven shafts, and the pressures and modes of the four clutches. Then start the model.

Open the Clutch Control subsystem. You can change the gear settings by flipping the Bit 0 and Bit1 Switch blocks and moving through truth table entries corresponding to each setting. (See the Simpson 4-Speed Clutch Schedule Logic on page 2-31 table preceding.) Switching from one gear setting to another unlocks some clutches and locks others, but always leaves two clutches locked. As you flip between gear settings, the transmission transfers motion and torque at different ratios.

You can disengage the Simpson transmission completely by flipping on the Neutral Switch. This step also enables the Brake Switch in the main model. (If the transmission is engaged, not in neutral, the Brake Switch is disabled.) If the transmission is disengaged but without braking, the driven shaft velocity slowly decreases under the influence of frictional damping. If you brake, however, by switching on the Brake Switch, the driven shaft velocity immediately drops to zero. The braking here works the same way as in the previous examples.

Shaping Realistic Clutch Pressure Signals

The `drive_simpson` model allows you to switch gear settings without placing the Simpson transmission in neutral. Of course, controlling a real manual transmission requires moving the transmission out of gear and into neutral, picking a new gear setting, then putting the transmission into the new gear. You can mimic these steps by flipping the Neutral Switch on, changing the gear setting, then slipping the Neutral Switch off.

The most critical addition you can make to this model for greater realism is to change the clutch pressure signals from step functions (zero to one, or one to zero) to signals with a smooth rise and fall. But the price of this greater realism is a large increase in model complexity. It is critical for Simulink to determine transmission motion for exactly two clutches to always remain locked, or for all four to be unlocked, at any instant. Changing the Simpson gear settings while maintaining this requirement is an example of the central problem of transmission design.

3

SimDriveline Block Reference

Blocks — Categorical List

This section lists all SimDriveline blocks arranged by library. The subsequent pages contain reference information for all blocks in SimDriveline, arranged in alphabetical order by block name.

Use the Simulink Library Browser or the SimDriveline library to access the blocks directly, guided by this hierarchical library list.

Creating a Model

Driveline Environment	Represent the SimDriveline environment
Housing	Rotationally lock the connected driveline axis and prevent it from turning
Inertia	Represent a body with rotational inertia
Shared Environment	Connect two driveline components so that they share the same driveline environment

Modeling Gears

Differential	Represent a differential gear with specified differential gear ratio
Dual-Ratio Planetary	Represent a set of carrier, sun, planet, and ring gear wheels with specified ring-planet and planet-sun gear ratios
Planet-Planet	Represent a set of carrier, inner planet, and outer planet gear wheels with specified planet-planet gear ratio

Planetary Gear	Represent a set of carrier, sun, planet, and ring gear wheels with specified ring-sun gear ratio
Ravigneaux	Represent a Ravigneaux planetary set of carrier, sun, planet, and ring gear wheels with specified ring-sun gear ratios
Ring-Planet	Represent a set of carrier, planet, and ring gear wheels with specified ring-planet gear ratio
Simple Gear	Represent a gear with fixed gear ratio
Variable Ratio Gear	Represent a gear with controllable, variable gear ratio

Modeling Dynamic Elements

Controllable Friction Clutch	Represent a frictional clutch with kinetic and static friction and controlled by a pressure signal
Hard Stop	Model the restriction on the relative angular motion of two driveline axes to a free gap with elastic upper and lower limits
Torque Converter	Transfer torque between two driveline axes as a function of their relative angular velocity
Torsional Spring-Damper	Represent a damped torsional spring torque, with a free play gap, acting between two rotating axes

Modeling Transmissions

Lepelletier 6-Speed	Model a six-speed Lepelletier transmission based on a planetary gear and a Ravigneaux gear
Lepelletier 7-Speed	Model a seven-speed Lepelletier transmission based on a planetary gear and a Ravigneaux gear
Ravigneaux 4-Speed	Model a Ravigneaux four-speed transmission based on a Ravigneaux gear
Simpson 4-Speed	Model a Simpson four-speed transmission based on two planetary gear sets

Modeling Vehicle Components

Diesel Engine	Model a diesel fuel engine with throttle control and driveline output
Gasoline Engine	Model a gasoline fuel engine with throttle control and driveline output
Longitudinal Vehicle Dynamics	Model longitudinal and vertical dynamics and motion of a two-axle, four-wheel vehicle
Tire	Model tire dynamics and motion at the end of a driveline axis

Sensing and Actuating Motion

Initial Condition	Set the initial angular velocity of a driveline axis to a nonzero value
Motion Actuator	Actuate a driveline axis with specified motions

Motion Sensor	Measure the motion of a driveline axis
Torque Actuator	Actuate a driveline axis with a specified torque
Torque Sensor	Measure the torque transferred along a driveline axis

Additional Useful Blocks

Connection Port	Create a Physical Modeling connector port for a subsystem

4

Blocks – Alphabetical List

Connection Port

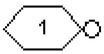
Purpose

Create a Physical Modeling connector port for a subsystem

Library

Utilities

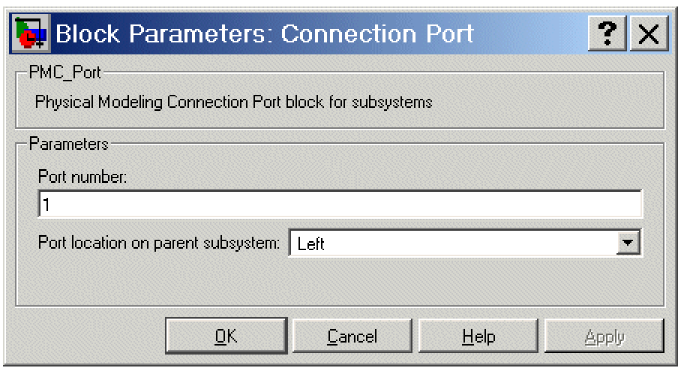
Description



The Connection Port block, placed inside a subsystem composed of SimDriveline blocks, creates an open SimDriveline round connector port ○ on the boundary of the subsystem. Once connected to a connection line, the port becomes a driveline connector port ⊏.

You connect individual SimDriveline blocks and subsystems made of SimDriveline blocks to one another with SimDriveline connection lines instead of normal Simulink signal lines. These are anchored at the open square driveline connector ports ⊏. Subsystems constructed from SimDriveline blocks automatically have such driveline connector ports. You can add additional connector ports by adding Connection Port blocks to your subsystem.

Dialog Box and Parameters



Port number

Labels the subsystem connector port created by this block. 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 on which side of the parent subsystem boundary the port is placed. The choices are `Left` or `Right`. The default choice is `Left`.

See Also

In Simulink, see Creating Subsystems.

Controllable Friction Clutch

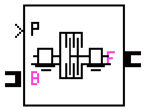
Purpose

Represent a frictional clutch with kinetic and static friction and controlled by a pressure signal

Library

Dynamic Elements

Description



A friction clutch transfers angular velocity and torque between two driveline axes by coupling them with friction. The Controllable Friction Clutch block models a standard friction clutch with kinetic and static locking friction acting on the two axes. The motion is measured as follower (F) axis relative to base (B) axis, $\omega = \omega_F - \omega_B$.

The friction clutch can be bidirectional, allowing the follower to rotate relative to the base in either direction; or unidirectional, allowing the follower to rotate relative to the base in the forward direction only ($\omega \geq 0$).

The clutch requires a dimensionless input pressure signal P that modulates the applied friction. This signal must be greater positive or zero.

Clutch Friction and Locking

A friction clutch can be in one of three states: *unengaged*, when it applies no friction at all; *engaged* (but not locked), when it applies kinetic friction; and *locked*, when it applies static friction. There is also a fourth, virtual state between locked and unlocked called the *wait state* (see “Friction Clutch Model” on page 4-9).

The kinetic friction torque requires that you specify six factors:

- Number of friction disks
- Effective torque radius
- Peak normal force
- Normalized pressure signal P and pressure threshold P_{th}
- Kinetic friction coefficient μ expressed as a discrete tabulated function of ω

If it locks, a Controllable Friction Clutch block imposes a constraint on your driveline system by requiring that two otherwise independent angular velocities be equal. A locked clutch thus reduces the number of independent degrees of freedom by one. Locking requires that the relative speed $|\omega|$ be smaller than a velocity threshold ω_{Tol}.

The static friction torque controls the unlocking of a friction clutch. (You can optionally lock the clutch at the start of the simulation as well.) When the clutch is locked, it remains locked unless the friction constraint torque across the clutch exceeds a static friction limit. By default, SimDriveline decides when to unlock a clutch after repeatedly suspending the simulation in time, entering a nonphysical algebraic loop, and testing a set of unlocking conditions (see "Friction Clutch Model" on page 4-9). This virtual testing is called *mode iteration*.

Enabling and Disabling Clutch Mode Iteration

You can turn off mode iteration for a whole driveline machine from that machine's Driveline Environment block. In that case, SimDriveline tests the friction clutches of your model without entering an algebraic loop. Instead, the unlocking tests are applied over multiple time steps, improving your simulation performance, but possibly decreasing its accuracy.

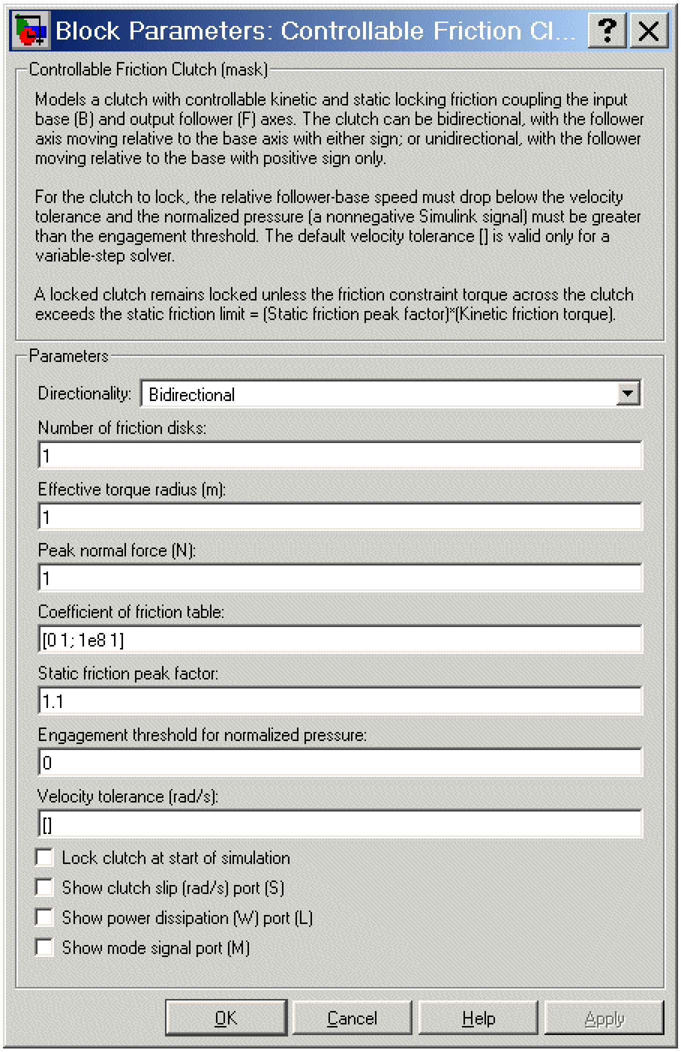
Caution If you convert your model to a real-time C code simulation using Real-Time Workshop, first check the Clutch mode iteration compatibility with the code generation option you choose.

You might have to adjust clutch mode iteration if you are generating code from your SimDriveline model. The code generation-based options are consistent with mode iteration in some cases, but not in others.

Code Generation Option	Clutch Mode Iteration
Simulink Accelerator	Can be enabled or disabled

Code Generation Option	Clutch Mode Iteration
Real-Time Workshop: RSIM Target	Can be enabled or disabled
Real-Time Workshop: targets other than RSIM	Must be disabled

Dialog Box and Parameters



Directionality

Select `Bidirectional` or `Unidirectional` to determine how the follower axis can turn relative to the base, in both directions or only in the forward direction, respectively. The default is `Bidirectional`.

Number of friction disks
Number of friction-generating contact surfaces inside the clutch. The default is `1`.

Effective torque radius
The effective moment arm radius that determines the kinetic friction torque inside the clutch. The default is `1 m` (meter).

Peak normal force
The maximum force normal to the frictional surfaces in the clutch. This value normalizes the Simulink input signal *P* for clutch pressure and determines the maximum kinetic friction torque. The default is `1 N` (Newton).

Coefficient of friction table
Dimensionless kinetic friction factor μ as a function of the angular velocity in tabular form. The table is a matrix of *N* vectors of length 2, separated by semicolons. Each two-vector specifies a pair of values, an angular velocity and a corresponding μ value, in that order. The simulation automatically interpolates a cubic spline from these values.

The default matrix is `[0 1; 1e8 1]`, which is a constant μ of value one. The angular velocity values are in units of radians/second.

Static friction peak factor
Ratio of the static friction limit for clutch locking to the kinetic friction. The default is `1.1`.

Engagement threshold for normalized pressure
Minimum normalized pressure P_{th} that activates clutch friction. If the normalized pressure input signal *P* is less than this threshold, the clutch applies no friction. The default is `0`.

Velocity tolerance
Sets the maximum angular velocity allowed during modal iteration to maintain clutch locking. The units are radians/second.

The default [] value is valid only for a variable-step solver and allows the simulation to automatically find a velocity tolerance based on the Simulink ODE solver settings.

For a fixed-step solver, you must enter a numerical value or expression.

Lock clutch at start of simulation
Select to start the simulation with the clutch already locked. The default is unselected.

Show mode signal port (M)
Select to make available the Simulink port for the discrete clutch mode signal. The default is unselected.

The signal value is the sign ±1 of the angular velocity ω of the follower relative to the base. If the follower and base are locked and rotate together, the signal value is 0.

Show clutch slip port (S)
Select to make available the Simulink outport for the clutch slippage signal. The default is unselected.

The clutch slippage is the relative angular velocity ω of the two coupled driveline axes. The signal measures the clutch slippage in radians/second.

Show power dissipation port (L)
Select to make available the Simulink outport for the power dissipation signal. The default is unselected.

The signal measures the power in Watts being dissipated by friction torques applied by the clutch to the driveline axis.

Friction Clutch Model

When a pressure signal above threshold ($P \geq P_{th}$) is applied, the Controllable Friction Clutch block applies two kinds of friction to the driveline motion, kinetic and static. The kinetic friction torque is applied only when one driveline axis is spinning relative to the other driveline axis. The static friction torque is applied when the two driveline axes are locked and spin together. SimDriveline iterates through a multistep conditional testing of motion to decide when to unlock the clutch.

Kinetic Friction — The Unlocked State

The kinetic friction torque is a product of five factors:

> μ*(Number of friction disks)*(effective torque radius)*(peak normal force)*(normalized pressure – pressure threshold)

The effective torque radius is the effective radius, measured from the driveline axis, at which the frictional forces are applied at the frictional surfaces. You specify the kinetic friction coefficient μ as a discrete tabulated function of relative angular velocity ω_{rel}. This function is assumed to be symmetric for positive and negative values of the relative angular velocity, so that you need to specify μ only for positive values of ω_{rel}. The peak normal force is the largest normal force applied to the frictional surfaces.

The normalized pressure input signal must be nonnegative. If P is less than P_{th}, the clutch applies no friction at all.

Static Friction — The Locked State

Once a friction clutch is locked, it remains locked until the torque across the clutch exceeds the static friction limit. The static friction limit is a product of two factors:

> (Static friction peak factor)*(Kinetic friction torque for $\omega \to 0$)

The kinetic friction torque used here is computed with the kinetic friction coefficient μ interpolated to zero relative angular velocity with a cubic spline.

The static friction peak factor measures how much larger the static friction is compared to the kinetic friction, at the instant of unlocking, when $\omega = 0$.

How a Friction Clutch Locks and Unlocks — The Wait State

If the pressure signal $P \geq P_{th}$, the friction clutch locks the two driveline axes together when $|\omega| \leq \omega_{Tol}$. You can also lock a clutch before the simulation starts with the **Lock clutch at start of simulation** option in the dialog. In either case, SimDriveline treats a locked clutch as an extra constraint on the angular motion of the system. If the absolute value of the friction constraint torque applied across the two driveline axes exceeds the static friction limit, the clutch enters the wait state and might unlock.

The unlocking of a friction clutch is a conditional, multistep process implemented internally by SimDriveline.

- If you turn off mode iteration for your driveline model (in the Driveline Environment block dialog), the unlocking process takes place over multiple simulation time steps.
- If you leave it on, the unlocking process suspends the simulation in time and starts an unphysical algebraic loop.

The wait state encompasses the multiple conditional steps that test the entire machine for unlocking.

1. SimDriveline first checks the relative acceleration $\alpha = d\omega/dt$ of the two driveline axes, given the torques present when the clutch enters the wait state.

   If the whole machine requires the axes to turn in the relative forward direction, but α is negative; or if the whole machine requires the axes to turn in the relative reverse direction, but α is positive, the clutch returns from the wait state to the locked state.

2. If the clutch remains in the wait state instead of returning to locked, SimDriveline integrates the relative acceleration in time to obtain the absolute value of the virtual angular velocity and checks this result against angular velocity tolerance ω_{Tol}. If the result is less than ω_{Tol}, the clutch returns to the start of the wait state and the relative acceleration check. If the result exceeds ω_{Tol}, the clutch unlocks.

3 In the unlocked state, the clutch begins applying kinetic friction again. SimDriveline also begins again to check for the locking condition ($|\omega| = \omega_{Tol}$ with $P \geq P_{th}$).

If a clutch unlocks, the unlocking removes a constraint from your driveline system and restores an independent degree of freedom.

Examples

These SimDriveline demo models contain working examples of clutches used to change gear couplings:

- `drive_sclutch`
- `drive_clutch_engage`
- `drive_strans`
- `drive_simpson`
- `drive_full_car`

See Also

`Differential`, `Driveline Environment`, `Torque Converter`

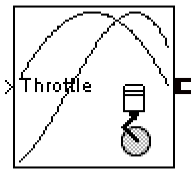
Purpose

Model a diesel fuel engine with throttle control and driveline output

Library

Vehicle Components

Description



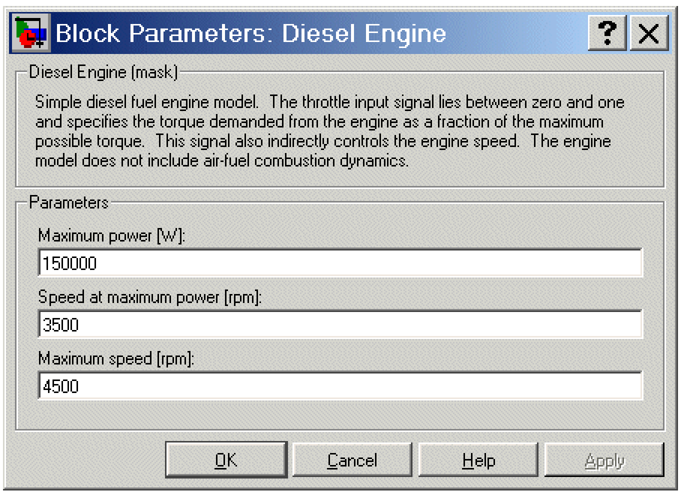
The Diesel Engine block models a diesel-fuel, compression-ignition engine. The engine runs at a variable speed that you can control with a Simulink throttle signal. The throttle signal directly controls the output torque that the engine generates and indirectly controls the speed at which the engine runs. The model does not include the air-fuel dynamics of combustion.

The block accepts the throttle signal through a Simulink inport. This signal must lie between 0 and 1.

Using Vehicle Component Blocks

Use the blocks of the Vehicle Components library as a starting point for vehicle modeling. The blocks of this library serve as suggestions for developing variant or entirely new models to simulate the same components.

Dialog Box and Parameters



Maximum power
: Maximum power that the engine can output, in Watts (W). The default is 150000.

Speed at maximum power
: Engine speed, in revolutions per minute (rpm), when the engine is running at maximum power. The default is 3500.

Maximum speed
: Maximum speed, in revolutions per minute (rpm), at which the engine can turn. The default is 4500.

Engine Model

The engine model uses a programmed relationship between torque and speed, modulated by the throttle signal. An actual diesel engine does not have a throttle.

Engine Speed, Torque, and Throttle

The engine model is specified by an *engine torque demand* function $g(\Omega)$ built into the block. It provides the maximum torque available for a given engine speed Ω. The block dialog entries (maximum power, speed at maximum power, and maximum speed) normalize this function to physical maximum torque and speed values.

The throttle input signal T specifies the actual engine torque delivered as a fraction of the maximum torque possible in a steady state at a fixed engine speed. It modulates the actual torque delivered τ from the engine: $\tau = T \cdot g(\Omega)$. The actual engine drive shaft speed Ω is fed back to the engine input.

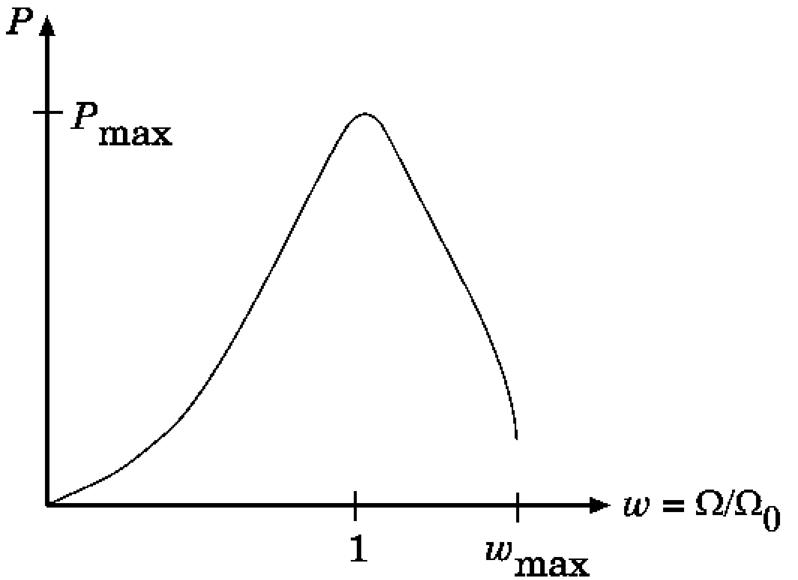
Engine Power and Torque Demand

The demand function $g(\Omega)$ is specified in terms of the steady-state engine power $P(\Omega)$.

The engine speed is limited to a maximum: $0 \le \Omega \le \Omega_{max}$. The absolute maximum engine power P_{max} defines Ω_0 such that $P_{max} = P(\Omega_0)$. Define $w = \Omega/\Omega_0$ and $P(\Omega) = P_{max} \cdot p(w)$. Then $p(1) = 1$ and $dp(1)/dw = 0$. Power is the product of torque and angular velocity. The torque demand function is thus

$$\tau_{max} = g(w) = (P_{max} / \Omega_0) \cdot [p(w) / w]$$

You can derive forms for *p(w)* from engine data and models.



Typical Engine Power Demand Function

See Also

`Controllable Friction Clutch`, `Gasoline Engine`, `Torque Converter`

Differential

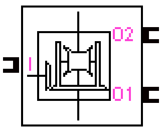
Purpose

Represent a differential gear with specified differential gear ratio

Library

Gears

Description



The Differential block represents a differential gear that couples rotational motion about the longitudinal axis to rotational motion about two lateral axes.

Any one axis can be the input. But in normal use, the longitudinal shaft is the input, and motion, torque, and power flow out through the lateral shafts. The output axes in general have different angular velocities. The longitudinal motion is divided by the drive gear ratio that you specify and then split between the two lateral shafts.

The torques along the lateral axes, τ_{O1} and τ_{O2}, are constrained to the longitudinal torque τ_I in such a way that the power input equals the sum of the power outputs:

$$\omega_I \tau_I = g_D(\omega_{O1} + \omega_{O2})\tau_I = \omega_{O1}\tau_{O1} + \omega_{O2}\tau_{O2} ,$$

$$g_D\tau_I = (\omega_{O1}\tau_{O1} + \omega_{O2}\tau_{O2})/(\omega_{O1} + \omega_{O2})$$

Differentials in drivelines often have a controllable clutch connecting the two output shafts. You can add this clutch control by appropriately connecting a Controllable Friction Clutch block to the Differential block.

Axis Motions and Constraint

The three rotational degrees of freedom, the longitudinal ω_I and the lateral ω_{O1} and ω_{O2}, are subject to one gear constraint and reduce to two independent degrees of freedom. In terms of the drive gear ratio g_D, the longitudinal motion is related to the sum of the lateral motions:

$$\omega_I = g_D(\omega_{O1} + \omega_{O2})$$

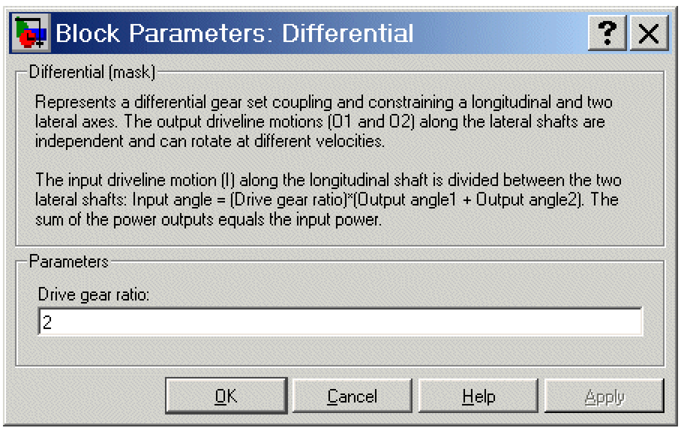
Thus the *sum* of the lateral motions depends on the longitudinal motion, once the longitudinal axis is connected. The difference of lateral motions ω_{O1} - ω_{O2} is independent of the longitudinal motion. These two independent degrees of freedom have this physical significance:

- One degree of freedom is equivalent to the two lateral shafts rotating at the same angular velocity ($\omega_{O1} = \omega_{O2}$) and at a fixed ratio with respect to the longitudinal shaft.
- The other degree of freedom is equivalent to keeping the longitudinal shaft locked ($\omega_I = 0$) while the lateral shafts rotate with respect to each other in opposite directions ($\omega_{O1} = -\omega_{O2}$).

The general motion of the lateral shafts is a superposition of these two motions.

Caution All gear ratios must be strictly positive. If any gear ratio equals 0 or becomes negative at any time during a simulation, SimDriveline stops with an error.

Dialog Box and Parameters



Drive gear ratio

Ratio g_D of the input angular motion to the sum of the two output angular motions. This ratio must be strictly positive. The default is 2.

Example

The demo model `drive_4wd_dynamics` combines two differentials with four tire-wheel assemblies to model the contact of tires with the road and the longitudinal vehicle motion.

See Also

`Controllable Friction Clutch`, `Longitudinal Vehicle Dynamics`, `Tire`

Purpose

Represent the SimDriveline environment

Library

Solver & Inertias

Description

Each driveline machine represented by a connected SimDriveline block diagram requires global environment information for simulation. The Driveline Environment block specifies this global information and connects the solver that your model needs before you can begin simulation.

Each topologically distinct driveline block diagram requires exactly one Driveline Environment block to be connected to it.

Setting the Simulation Mode

With the Driveline Environment block, you specify the simulation mode as either dynamics or linearization.

- *Dynamics* mode solves for the motion of your driveline system in time, starting with the initial conditions and applying the necessary torques, constraints, and motion constants.
- *Linearization* mode also solves for the driveline system motion forward in time, but simplifies the actions of certain blocks.

Blocks with Behavior Modified by Linearization Mode

The Variable Ratio Gear accepts the gear ratio as a function of time as a Simulink input signal. When you simulate a driveline model in linearization mode, all Variable Ratio Gears hold their changing gear ratios at their initial values.

Clutch Mode Changes and Clutch Mode Iteration

The Controllable Friction Clutch block does not change its dialog if you switch to linearization mode. However, during linearization, to avoid simulation errors or faulty results, you must ensure that the Controllable Friction Clutch blocks in your model do not change their state discontinuously by locking or unlocking.

The Driveline Environment block also controls how the Controllable Friction Clutches of your model unlock during dynamics simulation. Once a clutch locks and switches from kinetic to static friction, it can unlock only under certain conditions that are tested by *mode iteration*.

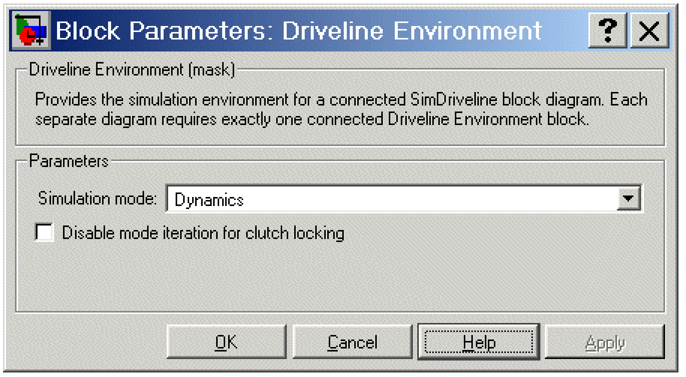
- With mode iteration turned on, SimDriveline suspends the simulation in time and enters a nonphysical algebraic loop to check and recheck the conditions for unlocking until it reaches global consistency across the model.
- With mode iteration turned off, SimDriveline checks for locking and unlocking while it continues to simulate in time. Turning off mode iteration improves your simulation performance, but might degrade its accuracy.

Clutch Mode Iteration and Code Generation

You might have to adjust clutch mode iteration if you are generating code from your SimDriveline model. The code generation-based options are consistent with mode iteration in some cases, but not in others.

Code Generation Option	Clutch Mode Iteration
Simulink Accelerator	Can be enabled or disabled
Real-Time Workshop: RSIM Target	Can be enabled or disabled
Real-Time Workshop: Targets other than RSIM	Must be disabled

Dialog Box and Parameters



Simulation mode

In this pull-down menu, select the mode that SimDriveline uses to analyze the motion of your driveline model, either `Dynamics` or `Linearization`. The default is `Dynamics`.

Disable mode iteration for clutch locking

Controls the mode iteration for locking and unlocking of Controllable Friction Clutches in your driveline model. Select the check box to disable mode iteration. The default is unselected.

See Also

`Controllable Friction Clutch`, `Shared Environment`, `Variable Ratio Gear`

Dual-Ratio Planetary

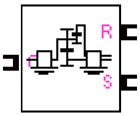
Purpose

Represent a set of carrier, sun, planet, and ring gear wheels with specified ring-planet and planet-sun gear ratios

Library

Gears

Description



The Dual-Ratio Planetary block represents a set of carrier, sun, planet, and ring gear wheels. The planet is a single gear wheel with two different radii meshing with the ring and the sun, respectively. The ring and planet corotate with one fixed gear ratio. The planet and sun corotate with another fixed gear ratio.

To model the planet's rotational inertia, connect an Inertia block to the optional planet connector port.

Axis Motions and Constraints

The Dual-Ratio Planetary block imposes two kinematic and two geometric constraints on the three connected axes and the fourth, internal wheel (planet):

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

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

In terms of the ring-to-planet gear ratio $g_{RP} = r_R/r_{P2}$ and the planet-to-sun gear ratio $g_{PS} = r_{P1}/r_S$, the key kinematic constraint is

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

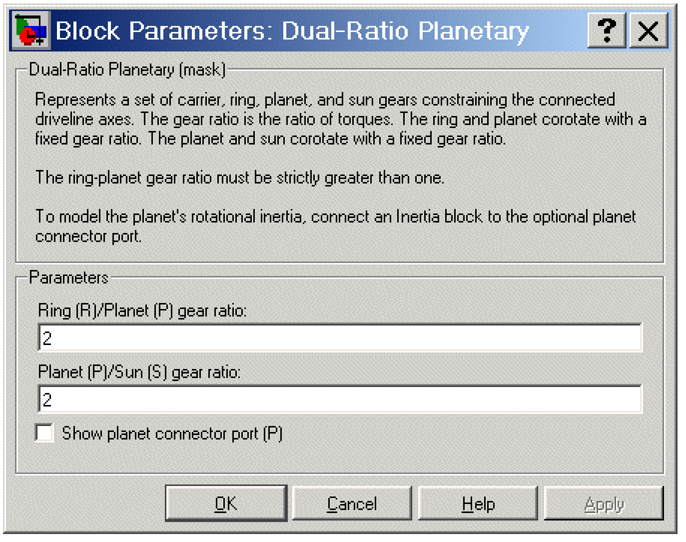
The four degrees of freedom are reduced to two independent degrees of freedom.

The gear ratios are also the ratios of the number of teeth on each gear and the ratios of the torques in each axis, $g_{RP} = N_R/N_{P2} = \tau_R/\tau_{P2}$ and $g_{PS} = N_{P1}/N_S = \tau_{P1}/\tau_S$.

Caution All gear ratios must be strictly positive. If any gear ratio equals 0 or becomes negative at any time during a simulation, SimDriveline stops with an error.

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

Dialog Box and Parameters



Ring (R)/Planet (P) gear ratio

Ratio g_{RP} of the ring gear wheel radius to the planet gear wheel radius. This ratio must be strictly greater than 1. The default is 2.

Planet (P)/Sun (S) gear ratio

Ratio g_{PS} of the planet gear wheel radius to the sun gear wheel radius. This ratio must be strictly positive. The default is 2.

Show planet connector port (P)

Selecting this check box makes the connector port for the planet gear visible and available for connection to other driveline blocks.

Use this connector port to connect an Inertia block if you want to model the planet gear's inertia. The default is unselected, with the planet gear's inertia neglected in the dynamics.

See Also

`Planet-Planet`, `Planetary Gear`, `Ring-Planet`

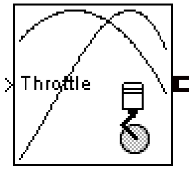
Purpose

Model a gasoline fuel engine with throttle control and driveline output

Library

Vehicle Components

Description



The Gasoline Engine block models a gasoline-fuel, spark-ignition engine. The engine runs at a variable speed that you can control with a Simulink throttle signal. The throttle signal directly controls the output torque that the engine generates and indirectly controls the speed at which the engine runs. The model does not include the air-fuel dynamics of combustion.

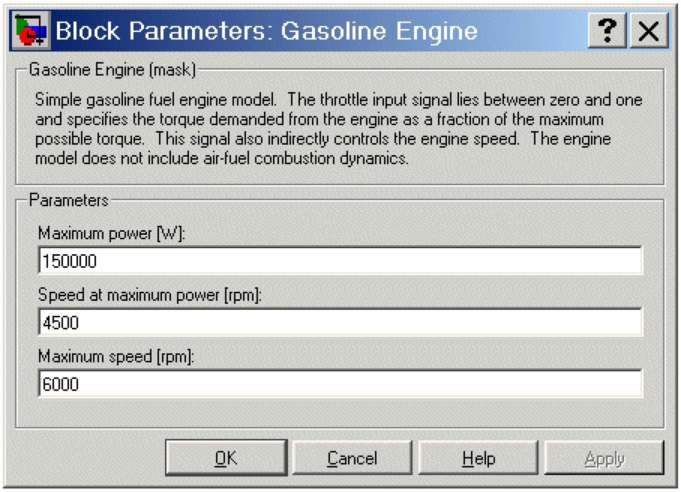
The block accepts the throttle signal through a Simulink inport. This signal must lie between 0 and 1. It specifies the engine torque as a fraction of the maximum torque possible in a steady state at a fixed engine speed.

Using Vehicle Component Blocks

Use the blocks of the Vehicle Components library as a starting point for vehicle modeling. The blocks of this library serve as suggestions for developing variant or entirely new models to simulate the same components.

Gasoline Engine

Dialog Box and Parameters



Maximum power

Maximum power that the engine can output, in Watts (W). The default is 150000.

Speed at maximum power

Engine speed, in revolutions per minute (rpm), when the engine is running at maximum power. The default is 4500.

Maximum speed

Maximum speed, in revolutions per minute (rpm), at which the engine can turn. The default is 6000.

Engine Model

The engine model uses a programmed relationship between torque and speed, modulated by the throttle signal.

Engine Speed, Torque, and Throttle

The engine model is specified by an *engine torque demand* function $g(\Omega)$ built into the block. It provides the maximum torque available for a given engine speed Ω. The block dialog entries (maximum power, speed at maximum power, and maximum speed) normalize this function to physical maximum torque and speed values.

The throttle input signal T specifies the actual engine torque delivered as a fraction of the maximum torque possible in a steady state at a fixed engine speed. It modulates the actual torque delivered τ from the engine: $\tau = T \cdot g(\Omega)$. The actual engine drive shaft speed Ω is fed back to the engine input.

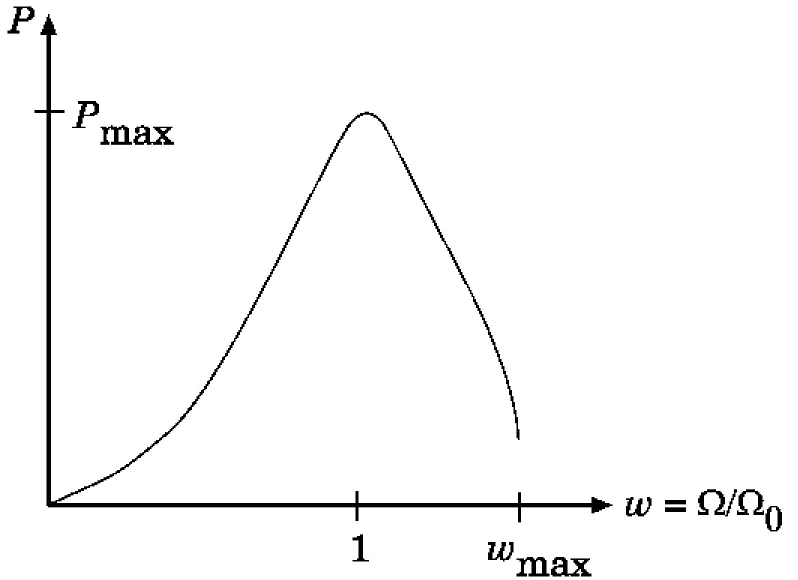
Engine Power and Torque Demand

The demand function $g(\Omega)$ is specified in terms of the steady-state engine power $P(\Omega)$.

The engine speed is limited to a maximum: $0 \leq \Omega \leq \Omega_{max}$. The absolute maximum engine power P_{max} defines Ω_0 such that $P_{max} = P(\Omega_0)$. Define $w = \Omega/\Omega_0$ and $P(\Omega) = P_{max} \cdot p(w)$. Then $p(1) = 1$ and $dp(1)/dw = 0$. Power is the product of torque and angular velocity. The torque demand function is thus

$$\tau_{max} = g(w) = (P_{max} / \Omega_0) \cdot [p(w) / w]$$

You can derive forms for $p(w)$ from engine data and models.



Typical Engine Power Demand Function

See Also `Controllable Friction Clutch`, `Diesel Engine`, `Torque Converter`

Purpose

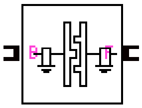
Model the restriction on the relative angular motion of two driveline axes to a free gap with elastic upper and lower limits

Library

Dynamic Elements

Description

The Hard Stop block simulates a two-position rotational stop that restricts the relative angular displacement θ of the two connected driveline axes.



- If the relative displacement falls in the gap between the stop's upper and lower limits, the stop applies no torque.
- If the relative displacement becomes greater than the upper limit θ_+ or smaller than the lower limit θ_-, the stop applies a torque.

At each limit, the stop imposes a one-sided damped, linear torsional torque limiting the motion of θ.

The relative angular displacement is the difference of the follower and base driveline axis angles, $\theta = \theta_F - \theta_B$, and the relative angular velocity $\omega = d\theta/dt = \omega_F - \omega_B$. If the angular displacement reaches beyond one of the stop limits, the torque applied is a sum of restoring and damping terms,

$$\tau = -k\cdot(\theta - \theta_\pm) - b\omega$$

where k is the contact stiffness and b the contact damping. Both constants must be nonnegative. The restoring torque depends only on the deformation angle $\theta - \theta_\pm$, measuring how far the displacement has penetrated beyond the upper or lower limit.

Relationship to Restitution

A restitution description of impact specifies the ratio of postimpact and preimpact velocities.

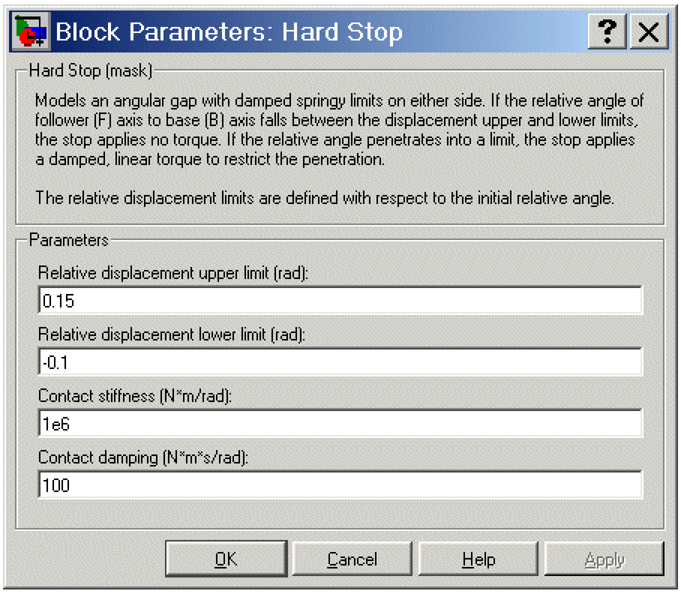
The effective inertias attached to the base and follower axes are I_B and I_F, respectively. The reduced inertia for the relative motion is

$$I = I_B I_F / (I_B + I_F)$$

Let the relative motion begin penetration of a one-sided stop limit at time t_i and complete its bounce at time t_f. The damped, linear springy torque reduces the final angular velocity, compared to the initial, by the ratio

$$\left|\frac{\dot{\theta}(t_f)}{\dot{\theta}(t_i)}\right| = \sqrt{1 - \frac{2b}{I} \cdot \int_i^f dt \cdot \left[\frac{\dot{\theta}(t)}{\dot{\theta}(t_i)}\right]^2}$$

Dialog Box and Parameters



Relative displacement upper limit

The largest relative displacement angle, in radians (rad), for which the stop does not apply a torque, measured relative to the initial relative angle. Must be larger than the relative displacement lower limit. The default is `0.15 rad`.

Relative displacement lower limit
The smallest relative displacement angle, in radians (rad), for which the stop does not apply a torque, measured relative to the initial relative angle. Must be smaller than the relative displacement upper limit. The default is `-0.1 rad`.

Contact stiffness
The linear contact stiffness constant *k,* in Newton-meters/radian (N·m/rad). Must be nonnegative. The default is `1e6 N·m·s/rad`.

Contact damping
The linear damping torque constant *b,* in Newton-meter-seconds/radian (N·m·s/rad). Must be nonnegative. The default is `100 N·m·s/rad`.

Example

The demo model `drive_hard_stop` simulates angular motion limited by a hard stop.

See Also

`Torsional Spring-Damper`

Housing

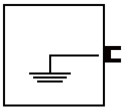
Purpose

Rotationally lock the connected driveline axis and prevent it from turning

Library

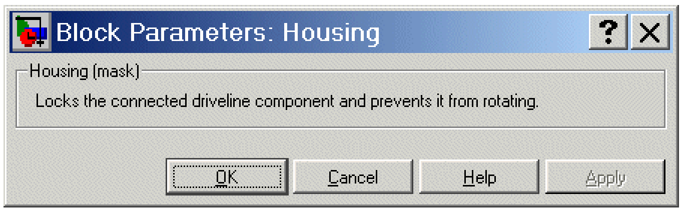
Solver & Inertias

Description



The Housing block prevents any driveline component connected to it from rotating about its driveline axis by locking its motion to zero angular velocity.

Dialog Box and Parameters



This block has no parameters.

See Also

`Inertia`, `Shared Environment`

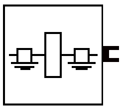
Purpose

Represent a body with rotational inertia

Library

Solver & Inertias

Description



The Inertia block represents a rigid rotating body. It rotates about the connected driveline axis, which carries the degree of freedom of motion. The body carries a rotational moment of inertia about that axis, which you specify in the block dialog.

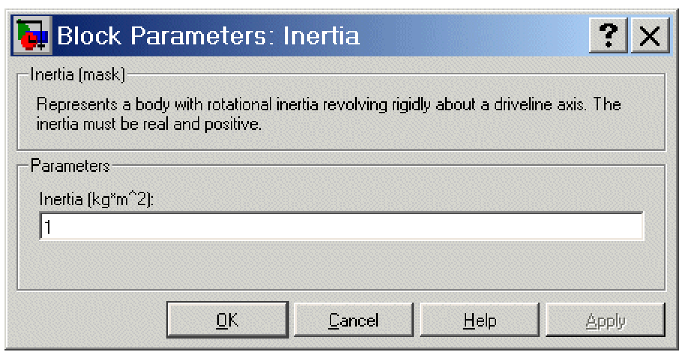
The Inertia block has one port. You can connect it to a driveline axis

- By connecting the port to the end of the axis
- By branching a connection line off the main line and connecting it to the port

---

Caution Normally, a rigid body in SimDriveline has a positive inertia. You can also enter zero inertia for particular driveline bodies. But if you do so, you must ensure that the effective inertia of your entire driveline is positive before actuating it with torques.

Dialog Box and Parameters



Inertia

Rotational moment of inertia of the body represented by the block. Must be a real, nonnegative number or MATLAB expression. The default is `1 kg*m`^{2} (kilogram-meters^2).

See Also

`Housing`

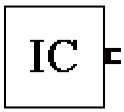
Purpose

Set the initial angular velocity of a driveline axis to a nonzero value

Library

Sensors & Actuators

Description

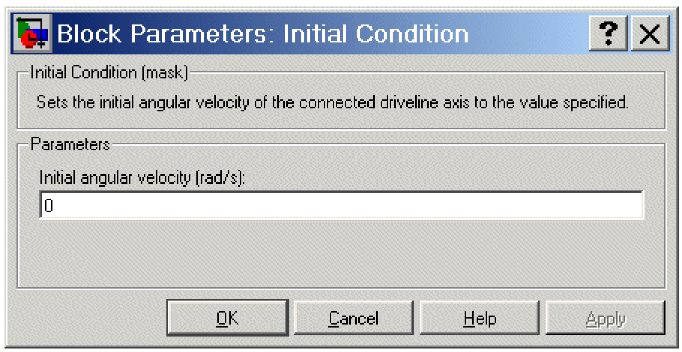


The Initial Condition block connects to a driveline axis and specifies a value for the initial angular velocity of that axis. You specify the initial angular velocity, in radians/second, in the block dialog.

The Initial Condition block has one port. You can connect it to a driveline axis

- By connecting the port to the end of the axis
- By branching a connection line off the main line and connecting it to the port

Dialog Box and Parameters



Initial angular velocity

The initial angular velocity of the driveline axis connected to this block. Units are radians/second (rad/s). The default is `0 rad/s`.

See Also

`Housing`, `Inertia`, `Motion Actuator`

Lepelletier 6-Speed

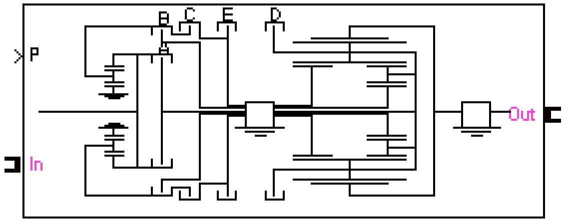
Purpose

Model a six-speed Lepelletier transmission based on a planetary gear and a Ravigneaux gear

Library

Transmission Templates

Description



The Lepelletier 6-Speed transmission block is a subsystem that models a standard automotive transmission having six selectable forward gear ratios and a single reverse gear ratio. The Lepelletier gearbox is constructed by connecting a planetary gear to a Ravigneaux gear. The sun of the planetary gear is connected to the housing and cannot rotate. The carrier of the planetary gear is connected, by clutches (A and C), to the large and small sun wheels of the Ravigneaux gear, respectively. The input, or driver, shaft is always connected to the ring of the planetary gear and can simultaneously be connected to the carrier of the Ravigneaux gear using a separate clutch (B). The output, or driven, shaft is connected to the ring of the Ravigneaux gear. The five transmission clutches A, B, C, D, and E are modeled with Controllable Friction Clutch blocks. You connect the transmission along a driveline axis, with the In and Out connector ports representing the input and output shafts, respectively.

This transmission subsystem has two independent internal degrees of freedom and therefore requires that two clutches be locked at any instant in order to achieve a unique drive ratio from the input shaft to the output shaft. The clutch schedule and the corresponding drive ratios are provided in the block subsystem's clutch schedule table. You disengage this transmission by unlocking all its clutches simultaneously.

Using Transmission Template Subsystems

A Transmission Template block is not library-linked. Once you make a copy in your model, you can use it as is. You can also open and customize it as a subsystem by reconfiguring the properties of the individual Gear, Controllable Friction Clutch, and Inertia blocks.

You must provide a five-component Simulink vector signal of the normalized pressures applied to each clutch. The order of the pressure signals is ABCDE.

Default Clutch and Inertia Settings

All the Controllable Friction Clutch blocks in this Transmission subsystem have their default settings, except for the **Number of friction disks** field, which is set to 6.

To prevent dynamical singularities, some of the gear wheels have attached Inertia blocks with small default inertias of 10^{-4} kg*m^2 (kilogram-meters^2).

Subsystem Parameters

The gear ratio is the ratio of gear wheel radii, gear wheel teeth, or torque transferred. The gear ratio is the reciprocal of the ratio of the angular velocities transferred. The drive ratio is the effective gear ratio, output to input, of the entire transmission.

The basic Lepelletier 6-speed transmission gear ratios are

g_{RS} = Planetary ring/sun gear ratio = $r_{pR}/r_{pS} = N_{pR}/N_{pS}$

g_{RSl} = Ravigneaux ring/large sun gear ratio = $r_R/r_{Sl} = N_R/N_{Sl}$

g_{RSs} = Ravigneaux ring/small sun gear ratio = $r_R/r_{Ss} = N_R/N_{Ss}$

This table specifies the locked (*L*) and free (F) clutches A, B, C, D, and E for each gear setting. A free clutch is completely disengaged.

Lepelletier 6-Speed Clutch Schedule

Gear Setting	Drive Ratio	A	B	C	D	E
Reverse	$-g_{RSl}\cdot(1+g_{RS})/g_{RS}$	*L*	F	F	*L*	F
1	$g_{RSs}\cdot(1+g_{RS})/g_{RS}$	F	F	*L*	*L*	F
2	$(1+g_{RS})(g_{RSs}+g_{RSl})/[g_{RS}\cdot(1+g_{RSl})]$	F	F	*L*	F	*L*
3	$(1 + g_{RS})/g_{RS}$	*L*	F	*L*	F	F
4	$g_{RSs}\cdot(1 + g_{RS})/[g_{RSs}\cdot(1 + g_{RS}) - 1]$	F	*L*	*L*	F	F
5	$g_{RSl}/[g_{RSl} + 1/(1 + g_{RS})]$	*L*	*L*	F	F	F
6	$g_{RSl}/(1 + g_{RSl})$	F	*L*	F	F	*L*

See Also

`Controllable Friction Clutch`, `Inertia`, `Lepelletier 7-Speed`, `Planetary Gear`, `Ravigneaux`, `Ravigneaux 4-Speed`, `Simpson 4-Speed`

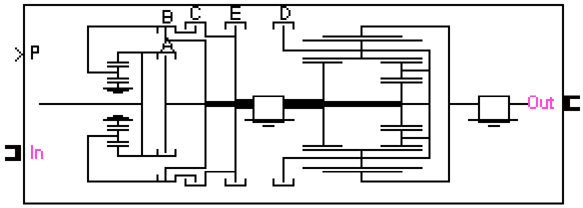
Purpose

Model a seven-speed Lepelletier transmission based on a planetary gear and a Ravigneaux gear

Library

Transmission Templates

Description



The Lepelletier 7-Speed transmission block is a subsystem that models a standard automotive transmission having seven selectable forward gear ratios and a single reverse gear ratio. The Lepelletier gearbox is constructed by connecting a planetary gear to a Ravigneaux gear. The sun of the planetary gear is connected by a clutch (F) to the housing and can be braked. The carrier of the planetary gear is connected, by another clutch (C), to the small sun wheel of the Ravigneaux gear. The input, or driver, shaft is always connected to the ring of the planetary gear and can simultaneously be connected to the carrier and large sun of the Ravigneaux gear using separate clutches (B and A). The output, or driven, shaft is connected to the ring of the Ravigneaux gear. The six transmission clutches A, B, C, D, E, and F are modeled with Controllable Friction Clutch blocks. You connect the transmission along a driveline axis, with the In and Out connector ports representing the input and output shafts, respectively.

This transmission subsystem has three independent internal degrees of freedom and therefore requires that three clutches be locked at any instant in order to achieve a unique drive ratio from the input shaft to the output shaft. The clutch schedule and the corresponding drive ratios are provided in the block subsystem’s clutch schedule table. You disengage this transmission by unlocking all its clutches simultaneously.

Using Transmission Template Subsystems

A Transmission Template block is not library-linked. Once you make a copy in your model, you can use it as is. You can also open and customize it as a subsystem by reconfiguring the properties of the individual Gear, Controllable Friction Clutch, and Inertia blocks.

You must provide a six-component Simulink vector signal of the normalized pressures applied to each clutch. The order of the pressure signals is ABCDEF.

Default Clutch and Inertia Settings

All the Controllable Friction Clutch blocks in this Transmission subsystem have their default settings, except for the **Number of friction disks** field, which is set to 6.

To prevent dynamical singularities, some of the gear wheels have attached Inertia blocks with small default inertias of 10^{-4} kg*m^2 (kilogram-meters^2).

Subsystem Parameters

The gear ratio is the ratio of gear wheel radii, gear wheel teeth, or torque transferred. The gear ratio is the reciprocal of the ratio of the angular velocities transferred. The drive ratio is the effective gear ratio, output to input, of the entire transmission.

The basic Lepelletier 7-speed transmission gear ratios are

g_{RS} = Planetary ring/sun gear ratio = $r_{pR}/r_{pS} = N_{pR}/N_{pS}$

g_{RSl} = Ravigneaux ring/large sun gear ratio = $r_R/r_{Sl} = N_R/N_{Sl}$

g_{RSs} = Ravigneaux ring/small sun gear ratio = $r_R/r_{Ss} = N_R/N_{Ss}$

This table specifies the locked (*L*) and free (F) clutches A, B, C, D, E, and F for each gear setting. A free clutch is completely disengaged.

Lepelletier 7-Speed Clutch Schedule

Gear Setting	Drive Ratio	A	B	C	D	E	F
Reverse	$-g_{RSl}\cdot(1+g_{RS})/g_{RS}$	*L*	F	F	*L*	F	*L*
1	$g_{RSs}\cdot(1+g_{RS})/g_{RS}$	F	F	*L*	*L*	F	*L*
2	$(g_{RSl} + g_{RSs})\cdot(1 + g_{RS})/[g_{RS}\cdot(1 + g_{RSl})]$	F	F	*L*	F	*L*	*L*
3	$(1 + g_{RS})/g_{RS}$	*L*	F	*L*	F	F	*L*
4	$g_{RSs}\cdot(1 + g_{RS})/[g_{RSs}\cdot(1 + g_{RS}) - 1]$	F	*L*	*L*	F	F	*L*
5	1	*L*	*L*	*L*	F	F	F
6	$g_{RSl}\cdot(1 + g_{RS})/[g_{RSl}\cdot(1 + g_{RS}) + 1]$	*L*	*L*	F	F	F	*L*
7	$g_{RSl}/(1 + g_{RSl})$	F	*L*	F	F	*L*	*L*

See Also

`Controllable Friction Clutch`, `Inertia`, `Lepelletier 6-Speed`, `Planetary Gear`, `Ravigneaux`, `Ravigneaux 4-Speed`, `Simpson 4-Speed`

Longitudinal Vehicle Dynamics

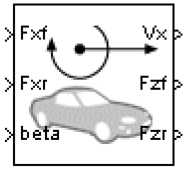
Purpose

Model longitudinal and vertical dynamics and motion of a two-axle, four-wheel vehicle

Library

Vehicle Components

Description



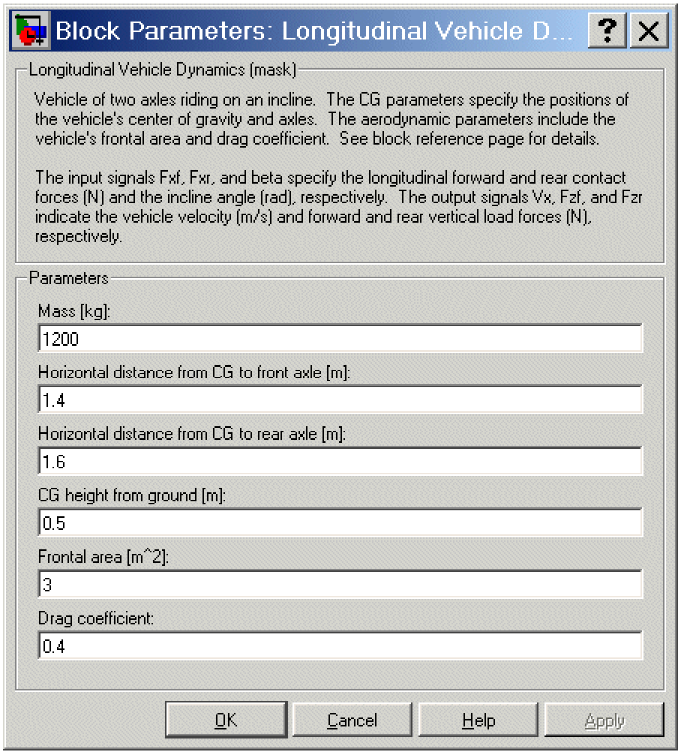
The Longitudinal Vehicle Dynamics block models a two-axle vehicle, with four equally sized wheels, moving forward or backward along its longitudinal axis. You specify forward and rear longitudinal forces F_{xf}, F_{xr} applied at the forward and rear wheel contact points, as well as the incline angle β, as a set of Simulink input signals. The block computes the vehicle velocity V_x and the forward and rear vertical load forces F_{zf}, F_{zr} on the vehicle as a set of Simulink output signals. All signals are specified in MKS units.

You must specify the vehicle mass and certain geometric details: position of the vehicle's center of gravity (CG) relative to the front and rear axles and to the ground, as well as an effective frontal cross sectional area and an aerodynamic drag coefficient. See "Vehicle Model" on page 4-44 for the details of the vehicle dynamics.

Using Vehicle Component Blocks

Use the blocks of the Vehicle Components library as a starting point for vehicle modeling. The blocks of this library serve as suggestions for developing variant or entirely new models to simulate the same components.

Dialog Box and Parameters



Mass

Mass m of the vehicle in kilograms (kg). The default is `1200 kg`.

Horizontal distance from CG to front axle

Horizontal distance d_f, in meters (m), from the vehicle's center of gravity to the vehicle's front wheel axle. The default is `1.4 m`.

Horizontal distance from CG to rear axle

Horizontal distance d_r, in meters (m), from the vehicle's center of gravity to the vehicle's rear wheel axle. The default is `1.6 m`.

CG height from ground

Height h, in meters (m), of the vehicle's center of gravity from the ground. The default is `0.5 m`.

Frontal area

Effective cross-sectional area *A*, in meters squared (m^2), presented by the vehicle in longitudinal motion, for the purpose of computing the aerodynamic drag force on the vehicle. The default is `3 m`^{2}.

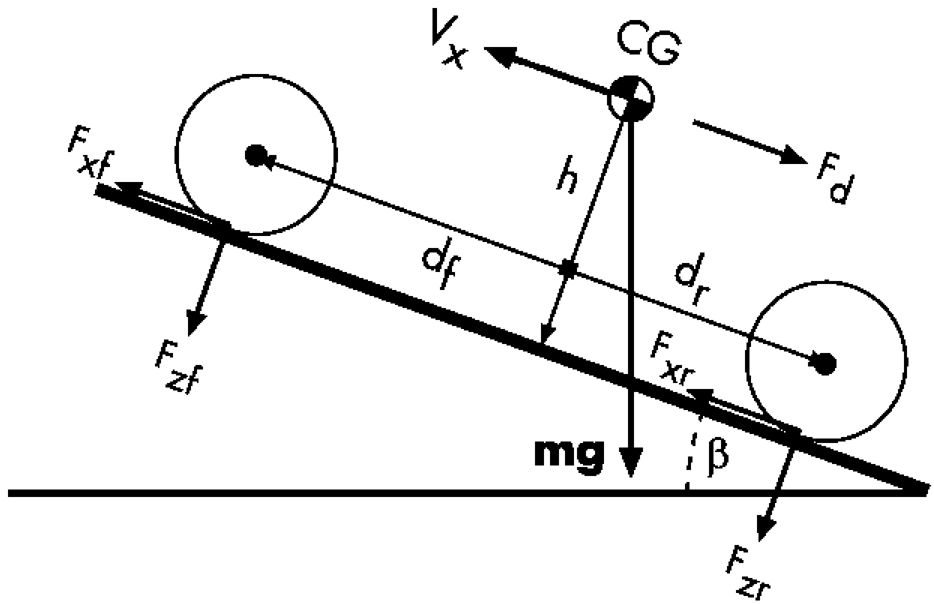
Drag coefficient

The dimensionless aerodynamic drag coefficient C_d, for the purpose of computing the aerodynamic drag force on the vehicle. The default is `0.4`.

Vehicle Model

The vehicle axles are parallel and lie in a plane parallel to the ground. The longitudinal x direction lies in this plane and perpendicular to the axles. If the vehicle is traveling on an incline slope β, the vertical z direction is not parallel to gravity but is always perpendicular to the axle-ground plane.

This figure and table define the vehicle motion model variables.



Vehicle Dynamics and Motion

Vehicle Model Variables and Constants

Symbol	Meaning and Units
g = -9.81 m/s^2	Gravitational acceleration (m/s^2)
β	Incline angle (rad)
m	Vehicle mass (kg)
A	Effective frontal vehicle cross-sectional area (m^2)
h	Height of vehicle CG above the ground (m)
d_f , d_r	Distance of front and rear axles from the vertical projection point of vehicle CG onto the axle-ground plane (m)
V_x	Longitudinal vehicle velocity (m/s)

Symbol	Meaning and Units
F_{xf} , F_{xr}	Longitudinal forces on the vehicle at the front and rear wheel ground contact points, respectively (N)
F_{zf} , F_{zr}	Vertical load forces on the vehicle at the forward and rear ground contact points, respectively (N)
C_d	Aerodynamic drag coefficient (N·s^2/kg·m)
ρ = 1.2 kg/m^3	Mass density of air (kg/m^3)
$\lvert F_d \rvert = \frac{1}{2} C_d \rho A V_x^2$	Aerodynamic drag force (N)

Vehicle Dynamics and Motion

The vehicle motion is determined by the net effect of all the forces and torques acting on it. The longitudinal tire forces push the vehicle forward or backward. The weight *mg* of the vehicle acts through its center of gravity (CG) and either pulls it back or forward, depending on the incline angle. Whether it travels forward or backward, aerodynamic drag slows the vehicle down. For simplicity, the drag is assumed to act through the CG.

$$m\dot{V}_x = F_x + F_d - mg \cdot \sin\beta,$$
$$F_x = F_{xf} + F_{xr} ,$$
$$F_d = -\frac{1}{2} C_d \rho A V_x^{\,2} \cdot \mathrm{sgn}(V_x)$$

Torque balance of all forces and zero vertical acceleration require

$$F_{zf} = \frac{d_r mg \cdot \cos\beta - hF_d}{d_f + d_r}$$
$$F_{zr} = \frac{d_f mg \cdot \cos\beta + hF_d}{d_f + d_r}$$

Example

The demo model `drive_4wd_dynamics` combines two differentials with four tire-wheel assemblies to model the contact of tires with the road and the longitudinal vehicle motion.

Reference

H. B. Pacejka, *Tire and Vehicle Dynamics,* Society of Automotive Engineers/Butterworth-Heinemann, Oxford, 2002.

See Also

`Differential`, `Tire`

Motion Actuator

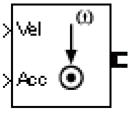
Purpose

Actuate a driveline axis with specified motions

Library

Sensors & Actuators

Description

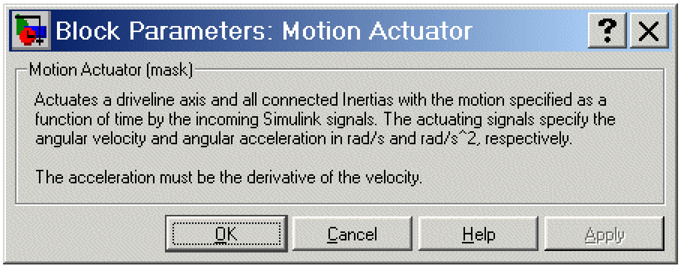


The Motion Actuator block actuates a driveline axis with specified motions. You specify the motion as velocity and acceleration with a set of two Simulink input signals in radians/second and radians/second^2, respectively. The motion specified is absolute.

The Motion Actuator block has one driveline port. You can connect it to a driveline axis

- By connecting the port to the end of the axis
- By branching a connection line off the main line and connecting it to the port

Dialog Box and Parameters



This block has no active parameters.

See Also

`Initial Condition`, `Motion Sensor`, `Torque Actuator`, `Torque Sensor`

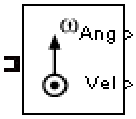
Purpose

Measure the motion of a driveline axis

Library

Sensors & Actuators

Description

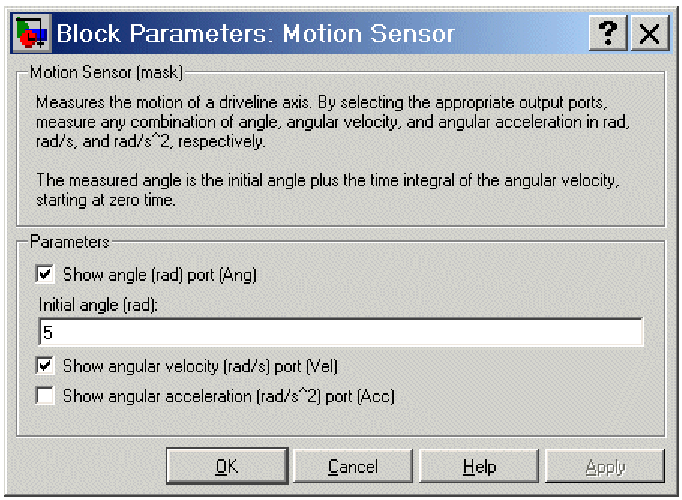


The Motion Sensor block senses the motion of a driveline axis. The block can output the motions as a set of three Simulink signals for the angle, angular velocity, and angular acceleration, in radians, radians/second, and radians/second^2, respectively. You can select any combination or all of these output signals. The motion measured is absolute.

The Motion Sensor block has one driveline port. You can connect it to a driveline axis

- By connecting the port to the end of the axis
- By branching a connection line off the main line and connecting it to the port

Dialog Box and Parameters



Show angle port (Ang)

Select this check box to create a Simulink outport on the block for the angular motion signal. The default is unselected.

Initial angle
Enter a value for the initial angle of the axis motion, in radians (rad).

The measured angle is this value plus the time integral of the angular velocity, starting at zero time.

Show angular velocity port (Vel)
Select this check box to create a Simulink outport on the block for the angular velocity signal. The default is selected.

Show angular acceleration port (Acc)
Select this check box to create a Simulink outport on the block for the angular acceleration signal. The default is selected.

See Also

`Motion Actuator`, `Torque Actuator`, `Torque Sensor`

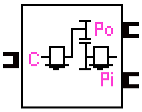
Purpose

Represent a set of carrier, inner planet, and outer planet gear wheels with specified planet-planet gear ratio

Library

Gears

Description



The Planet-Planet gear block represents a set of carrier, inner planet, and outer planet gear wheels. The outer planet is connected to and rotates with respect to the carrier. The inner planet rotates independently. The planets corotate with a fixed gear ratio that you specify and in opposite directions with respect to the carrier. A planet-planet gear is, along with a ring-planet gear, a basic element of a planetary gear set.

Axis Motions and Constraints

The Planet-Planet imposes one kinematic and one geometric constraint on the three connected axes:

$$r_C\omega_C = r_{Po}\omega_{Po} + r_{Pi}\omega_{Pi}\ ,\ r_C = r_{Po} + r_{Pi}$$

In terms of the outer planet-to-inner planet gear ratio $g_{oi} = r_{Po}/r_{Pi}$, the effective kinematic constraint is

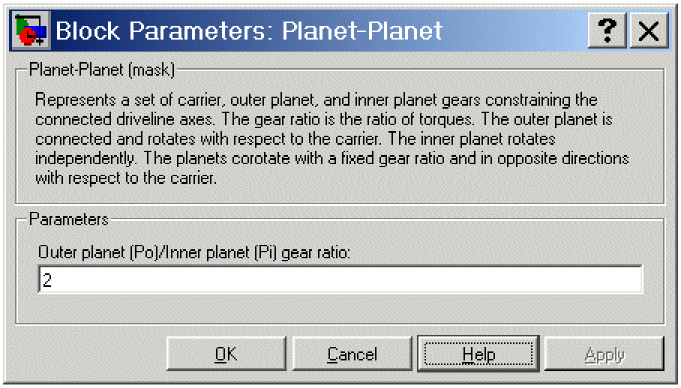
$$(1 + g_{oi})\omega_C = \omega_{Pi} + g_{oi}\cdot\omega_{Po}\ ,$$

reducing the three axes to two independent degrees of freedom.

The gear ratio is also the ratio of the number of teeth on each gear and the ratio of the torques in each axis, $g_{oi} = N_{Po}/N_{Pi} = \tau_{Po}/\tau_{Pi}$.

Caution All gear ratios must be strictly positive. If any gear ratio equals 0 or becomes negative at any time during a simulation, SimDriveline stops with an error.

Dialog Box and Parameters



Outer planet (Po)/Inner planet (Pi) gear ratio

Ratio g_{oi} of the outer planet gear radius wheel to the inner planet gear wheel radius. This gear ratio must be strictly positive. The default is `2`.

See Also

`Planetary Gear`, `Ring-Planet`

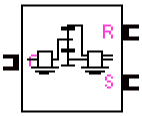
Purpose

Represent a set of carrier, sun, planet, and ring gear wheels with specified ring-sun gear ratio

Library

Gears

Description



The Planetary Gear block represents a set of carrier, ring, planet, and sun gear wheels. A planetary gear set can be constructed from planet-planet and ring-planet gears. The ring and sun corotate with a fixed gear ratio and in opposite directions with respect to the carrier.

To model the planet's rotational inertia, connect an Inertia block to the optional planet connector port.

Axis Motions and Constraints

The Planetary Gear imposes two kinematic and two geometric constraints on the three connected axes and the fourth, internal wheel (planet):

$$r_C\omega_C = r_S\omega_S + r_P\omega_P \ , \ r_C = r_S + r_P$$

$$r_R\omega_R = r_C\omega_C + r_P\omega_P \ , \ r_R = r_C + r_P$$

In terms of the ring-to-sun gear ratio $g_{RS} = r_R/r_S$, the key effective kinematic constraint is

$$(1 + g_{RS})\omega_C = \omega_S + g_{RS}\cdot\omega_R$$

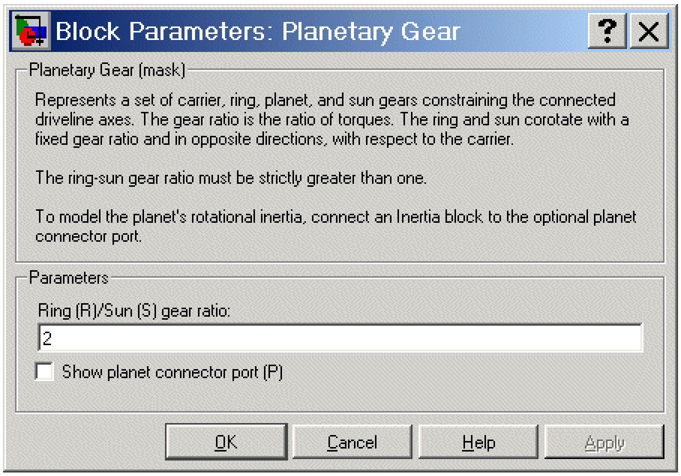
The four degrees of freedom are reduced to two independent degrees of freedom.

The gear ratio is also the ratio of the number of teeth on each gear and the ratio of the torques in each axis, $g_{RS} = N_R/N_S = \tau_R/\tau_S$.

Caution All gear ratios must be strictly positive. If any gear ratio equals 0 or becomes negative at any time during a simulation, SimDriveline stops with an error.

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

Dialog Box and Parameters



Ring (R)/Sun (S) gear ratio

Ratio g_{RS} of the ring gear wheel radius to the sun gear wheel radius. This gear ratio must be strictly greater than 1. The default is 2.

Show planet connector port (P)

Selecting this check box makes the connector port for the planet gear visible and available for connection to other driveline blocks.

Use this connector port to connect an Inertia block if you want to model the planet gear's inertia. The default is unselected, with the planet gear's inertia neglected in the dynamics.

See Also

`Dual-Ratio Planetary`, `Planet-Planet`, `Ring-Planet`

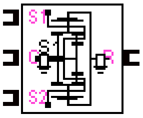
Purpose

Represent a Ravigneaux planetary set of carrier, sun, planet, and ring gear wheels with specified ring-sun gear ratios

Library

Gears

Description



The Ravigneaux block represents a double planetary gear set commonly used in automatic transmissions. This planetary gear set is constructed from two gear pairs, ring-planet and planet-planet. The Ravigneaux set has two sun gear wheels, a large sun and a small sun, and a single carrier gear with two independent planetary gear wheels connected to it, an inner planet and an outer planet. The carrier is one wheel but has two radii to couple with the inner and outer planets, respectively. The two planet gears rotate independently of the carrier but corotate with a fixed gear ratio with respect to each other. The inner planet couples with the small sun gear and corotates at a fixed gear ratio with respect to it. The outer planet couples with the large sun gear and corotates with a fixed gear ratio with respect to it. Finally, the ring gear also couples and corotates with the outer planet in a fixed gear ratio with respect to it.

To model the planets' rotational inertia, connect an Inertia block to the optional planet connector port.

Axis Motions and Constraints

The Ravigneaux block imposes four kinematic and four geometric constraints on the four connected axes and the two internal wheels (inner and outer planets):

$$r_{Ci}\omega_C = r_{Ss}\omega_{Ss} + r_{Pi}\omega_{Pi}\ ,\ r_{Ci} = r_{Ss} + r_{Pi}$$

$$r_{Co}\omega_C = r_{Sl}\omega_{Sl} + r_{Po}\omega_{Po}\ ,\ r_{Co} = r_{Sl} + r_{Po}$$

$$(r_{Co} - r_{Ci})\omega_C = r_{Pi}\omega_{Pi} + r_{Po}\omega_{Po}\ ,\ r_{Co} - r_{Ci} = r_{Po} + r_{Pi}$$

$$r_R\omega_R = r_{Co}\omega_C + r_{Po}\omega_{Po}\ ,\ r_R = r_{Co} + r_{Po}$$

In terms of the ring-to-small sun gear ratio $g_{RSs} = r_R/r_{Ss}$ and the ring-to-large sun gear ratio $g_{RSl} = r_R/r_{Sl}$, the key kinematic constraints are

$$(g_{RSs} - 1)\omega_C = g_{RSs} \cdot \omega_R - \omega_{Ss}$$

$$(g_{RSl} + 1)\omega_C = g_{RSl} \cdot \omega_R + \omega_{Sl}$$

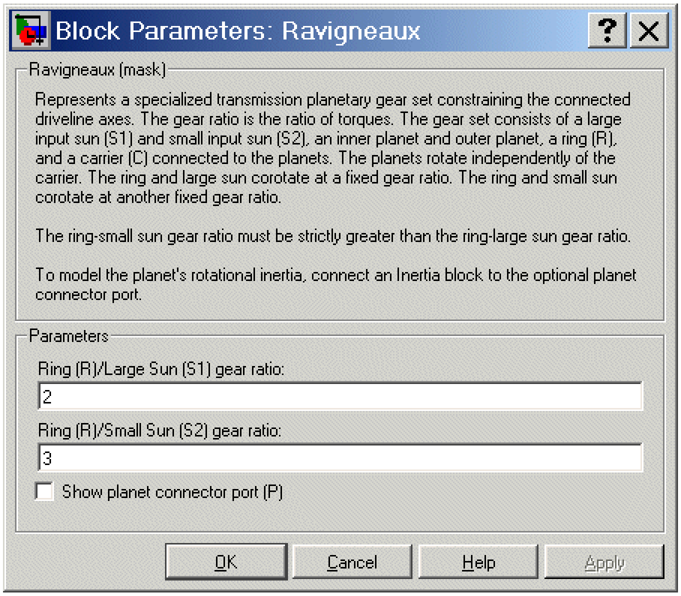
The six degrees of freedom are reduced to two independent degrees of freedom.

The gear ratios are also the ratios of the number of teeth on each gear and the ratios of torques in each axis, $g_{RSl} = N_R/N_{Sl} = \tau_R/\tau_{Sl}$ and $g_{RSs} = N_R/N_{Ss} = \tau_R/\tau_{Ss}$.

Caution All gear ratios must be strictly positive. If any gear ratio equals 0 or becomes negative at any time during a simulation, SimDriveline stops with an error.

The gear ratio g_{RSs} must be strictly greater than the gear ratio g_{RSl}.

Dialog Box and Parameters



Ring (R)/Large Sun (S1) gear ratio

Ratio g_{RSl} of the ring gear wheel radius to the large sun gear wheel radius. This gear ratio must be strictly smaller than the ring-small sun gear ratio. The default is 2.

Ring (R)/Small Sun (S2) gear ratio

Ratio g_{RSs} of the ring gear wheel radius to the small sun gear wheel radius. This gear ratio must be strictly greater than the ring-large sun gear ratio. The default is 3.

Show planet connector port (P)

Selecting this check box makes the connector port for the planet gears visible and available for connection to other driveline blocks.

Use this connector port to connect an Inertia block if you want to model the planet gears' inertia as a single body. The default is unselected, with the planet gears' inertia neglected in the dynamics.

See Also

`Dual-Ratio Planetary`, `Planet-Planet`, `Planetary Gear`, `Ring-Planet`, `Ravigneaux 4-Speed`

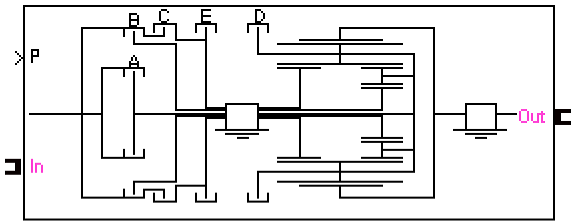
Purpose

Model a Ravigneaux four-speed transmission based on a Ravigneaux gear

Library

Transmission Templates

Description



The Ravigneaux 4-Speed transmission block is a subsystem that models a standard automotive transmission having four selectable forward gear ratios and a single reverse gear ratio. The main component of the transmission is a Ravigneaux gear. The input, or driver, shaft is connected, through a combination of the five clutches, to the small sun, the large sun, and the carrier of the Ravigneaux gear. The output, or driven, shaft is connected to the ring of the Ravigneaux gear. The five transmission clutches A, B, C, D, and E are modeled with Controllable Friction Clutch blocks. You connect the transmission along a driveline axis, with the In and Out connector ports representing the input and output shafts, respectively.

This transmission subsystem has two independent degrees of freedom and therefore requires two clutches be locked at any instant in order to achieve a unique drive ratio from the input shaft to the output shaft. The clutch schedule and the corresponding drive ratios are provided in the block subsystem's clutch schedule table. You disengage this transmission by unlocking all its clutches simultaneously.

Using Transmission Template Subsystems

A Transmission Template block is not library-linked. Once you make a copy in your model, you can use it as is. You can also open and customize it as a subsystem by reconfiguring the properties of the individual Gear, Controllable Friction Clutch, and Inertia blocks.

You must provide a five-component Simulink vector signal of the normalized pressures applied to each clutch. The order of the pressure signals is ABCDE.

Default Clutch and Inertia Settings

All the Controllable Friction Clutch blocks in this Transmission subsystem have their default settings, except for the **Number of friction disks** field, which is set to 6.

To prevent dynamical singularities, some of the gear wheels have attached Inertia blocks with small default inertias of 10^{-4} kg*m^2 (kilogram-meters^2).

Subsystem Parameters

The gear ratio is the ratio of gear wheel radii, gear wheel teeth, or torque transferred. The gear ratio is the reciprocal of the ratio of the angular velocities transferred. The drive ratio is the effective gear ratio, output to input, of the entire transmission.

The basic Ravigneaux 4-speed transmission gear ratios are

g_{RSl} = Ring/large sun gear ratio = $r_R/r_{Sl} = N_R/N_{Sl}$

g_{RSs} = Ring/small sun gear ratio = $r_R/r_{Ss} = N_R/N_{Ss}$

This table specifies the locked (*L*) and free (F) clutches A, B, C, D, and E for each gear setting. A free clutch is completely disengaged.

Ravigneaux 4-Speed Clutch Schedule

Gear Setting	Drive Ratio	A	B	C	D	E
Reverse	$-g_{RSl}$	F	F	*L*	*L*	F
1	g_{RSs}	F	*L*	F	*L*	F
2	$(g_{RSl} + g_{RSs})/(1 + g_{RSl})$	F	*L*	F	F	*L*

Gear Setting	Drive Ratio	A	B	C	D	E
3	1	*L*	*L*	F	F	F
4	$g_{RSl}/(1 + g_{RSl})$	*L*	F	F	F	*L*

See Also

`Controllable Friction Clutch`, `Inertia`, `Lepelletier 6-Speed`, `Lepelletier 7-Speed`, `Ravigneaux`, `Simpson 4-Speed`

Ring-Planet

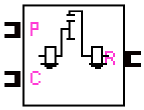
Purpose

Represent a set of carrier, planet, and ring gear wheels with specified ring-planet gear ratio

Library

Gears

Description



The Ring-Planet gear block represents a set of carrier, planet, and ring gear wheels. The planet is connected to and rotates with respect to the carrier. The planet and ring corotate with a fixed gear ratio that you specify and in the same direction with respect to the carrier. A ring-planet gear is, along with a planet-planet gear, a basic element of a planetary gear set.

Axis Motions and Constraints

The Ring-Planet imposes one kinematic and one geometric constraint on the three connected axes:

$$r_R\omega_R = r_C\omega_C + r_P\omega_P \text{ , } r_R = r_C + r_P$$

In terms of the ring-to-planet gear ratio $g_{RP} = r_R/r_P$, the effective kinematic constraint is

$$g_{RP}\cdot\omega_R = \omega_P + (g_{RP} - 1)\omega_C \text{ ,}$$

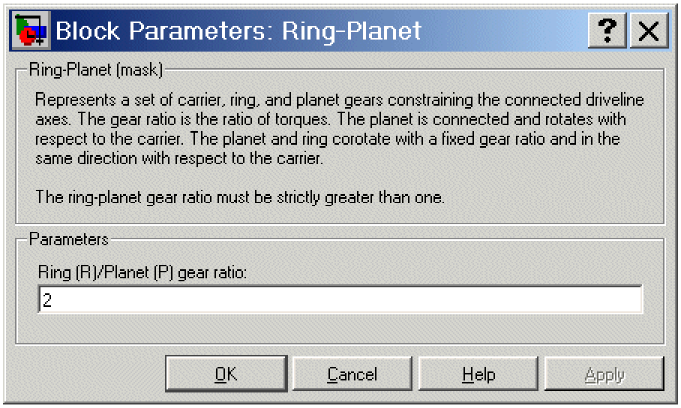
reducing the three axes to two independent degrees of freedom.

The gear ratio is also the ratio of the number of teeth on each gear and the ratio of the torques in each axis, $g_{RP} = N_R/N_P = \tau_R/\tau_P$.

Caution All gear ratios must be strictly positive. If any gear ratio equals 0 or becomes negative at any time during a simulation, SimDriveline stops with an error.

The ring-planet gear ratio g_{RP} must be strictly greater than one.

Dialog Box and Parameters



Ring (R)/Planet (P) gear ratio

Ratio g_{RP} of the ring gear wheel radius to the planet gear wheel radius. This gear ratio must be strictly greater than 1. The default value is `2`.

See Also

`Planet-Planet`, `Planetary Gear`

Shared Environment

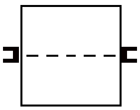
Purpose

Connect two driveline components so that they share the same driveline environment

Library

Solver & Inertias

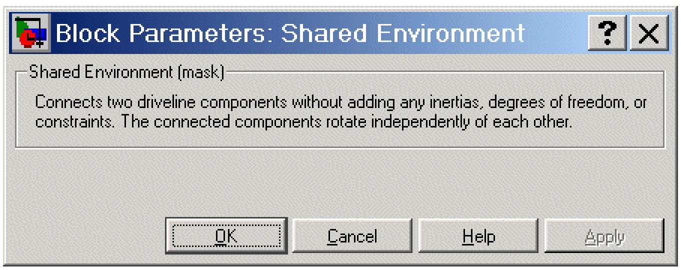
Description



The Shared Environment block provides a nonphysical connection between two independent driveline block diagrams. The block carries no inertia, adds no degrees of freedom, imposes no constraints, and transfers no motion or torque between the SimDriveline blocks to which it is connected.

You can use this block to connect two independent machines into one machine, so that the two machines, as one, then share the same driveline environment. Making this connection does not change the mechanical structure or dynamics of either machine.

Dialog Box and Parameters



This block has no parameters.

See Also

`Driveline Environment`, `Housing`

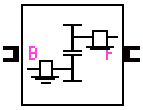
Purpose

Represent a gear with fixed gear ratio

Library

Gears

Description



The Simple Gear block represents a gear box that constrains the two connected driveline axes, base (B) and follower (F), to corotate with a fixed ratio that you specify. You can choose whether the follower axis rotates in the same or opposite direction as the base axis. If they rotate in the same direction, ω_F and ω_B have the same sign. If they rotate in opposite directions, ω_F and ω_B have opposite signs.

Axis Motion and Constraint

The Simple Gear imposes a single constraint, specified by the fixed gear ratio g_{FB}, on the motions and torques of the two axes:

$$\pm g_{FB} = \omega_B/\omega_F = \tau_F/\tau_B$$

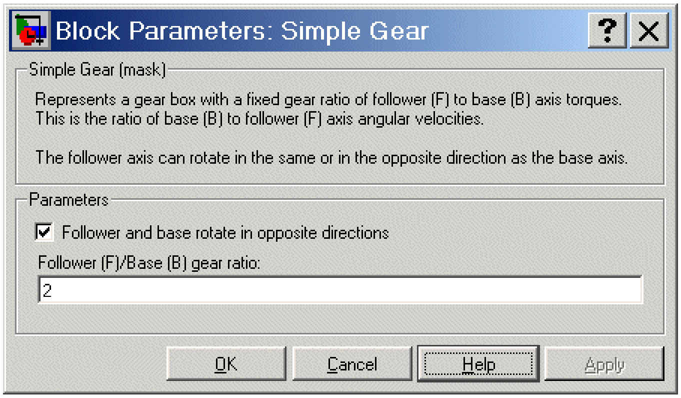
The plus and minus signs refer to the axes corotating in the same or opposite directions.

If the Simple Gear represents two coupled gear wheels, the gear ratio is related to the ratio of the radii r of the gear wheels and the ratio of the number N of teeth on each gear wheel:

$$g_{FB} = r_F/r_B = N_F/N_B$$

Caution The gear ratio g_{FB} must be strictly positive. If any gear ratio equals 0 or becomes negative at any time during a simulation, SimDriveline stops with an error.

Dialog Box and Parameters



Follower and base rotate in opposite directions
Select to make the follower and base axes corotate in opposite directions. The default is selected.

Follower (F)/Base (B) gear ratio
Fixed ratio g_{FB} of the follower axis to the base axis. The gear ratio must be strictly positive. The default is `2`.

See Also

`Variable Ratio Gear`

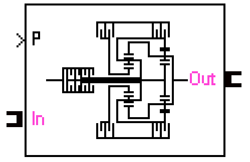
Purpose

Model a Simpson four-speed transmission based on two planetary gear sets

Library

Transmission Templates

Description



The Simpson 4-Speed transmission block is a subsystem model based on a simple planetary gearbox consisting of two standard planetary gear sets, an input set and an output set, connected together. The ring of the input planetary gear set is connected to the carrier of the output planetary set. Similarly, the ring of the output planetary gear set is connected to the carrier of the input planetary gear set. The input, or driver, shaft is connected either to the carrier of the input planetary gear set or to the sun of the output planetary gear set. The output, or driven, shaft is connected to the ring of the output planetary gear set and to the carrier of the input planetary gear set. The four transmission clutches A, B, C, and D are modeled with Controllable Friction Clutch blocks. You connect the transmission along a driveline axis, with the In and Out connector ports representing the input and output shafts, respectively.

This transmission subsystem has two independent internal degrees of freedom and therefore requires that two clutches be locked at any instant in order to achieve a unique drive ratio from the input shaft to the output shaft. The clutch schedule and the corresponding drive ratios are provided in the block subsystem's clutch schedule table. You disengage this transmission by unlocking all its clutches simultaneously.

Using Transmission Template Subsystems

A Transmission Template block is not library-linked. Once you make a copy in your model, you can use it as is. You can also open and customize it as a subsystem by reconfiguring the properties of the individual Gear, Controllable Friction Clutch, and Inertia blocks.

You must provide a four-component Simulink vector signal of the normalized pressures applied to each clutch. The order of the pressure signals is ABCD.

Default Clutch and Inertia Settings

All the Controllable Friction Clutch blocks in this Transmission subsystem have their default settings, except for the **Number of friction disks** field, which is set to 6.

To prevent dynamical singularities, some of the gear wheels have attached Inertia blocks with small default inertias of 10^{-4} kg*m^2 (kilogram-meters^2).

Subsystem Parameters

The gear ratio is the ratio of gear wheel radii, gear wheel teeth, or torque transferred. The gear ratio is the reciprocal of the ratio of the angular velocities transferred. The drive ratio is the effective gear ratio, output to input, of the entire transmission.

The basic Simpson 4-speed transmission gear ratios are

g_i = Input planetary ring/input planetary sun gear ratio = r_{Ri}/r_{Si} = N_{Ri}/N_{Si}

g_o = Output planetary ring/output planetary sun gear ratio = r_{Ro}/r_{So} = N_{Ro}/N_{So}

This table specifies the locked (L) and free (F) clutches A, B, C, and D for each gear setting. A free clutch is completely disengaged.

Simpson 4–Speed Clutch Schedule

Gear Setting	Drive Ratio	A	B	C	D
1	$1 + g_o$	*L*	F	F	*L*
2	$1 + g_o/(1 + g_i)$	*L*	F	*L*	F
3	1	*L*	*L*	F	F
4	$g_i/(1 + g_i)$	F	*L*	*L*	F

See Also

`Controllable Friction Clutch`, `Inertia`, `Lepelletier 6-Speed`, `Lepelletier 7-Speed`, `Planetary Gear`, `Ravigneaux 4-Speed`

Tire

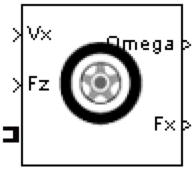
Purpose

Model tire dynamics and motion at the end of a driveline axis

Library

Vehicle Components

Description



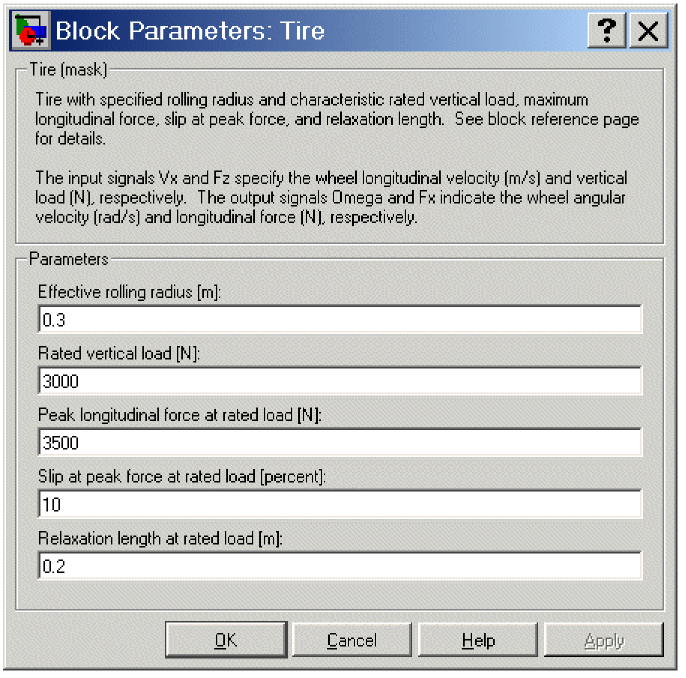
The Tire block models a vehicle tire in contact with the road. The torque on the wheel axis is transferred through the driveline port ⊏. You must specify the vertical load F_z and required longitudinal velocity V_x as Simulink input signals. The model provides the tire angular velocity Ω and the longitudinal force F_x on the vehicle as Simulink output signals. All signals have MKS units.

The longitudinal direction lies along the forward-backward axis of the vehicle. See "Tire Model" on page 4-72 for model details.

Using Vehicle Component Blocks

Use the blocks of the Vehicle Components library as a starting point for vehicle modeling. The blocks of this library serve as suggestions for developing variant or entirely new models to simulate the same components.

Dialog Box and Parameters



Effective rolling radius

Effective radius r_e, in meters (m), at which the longitudinal force is transferred to the wheel as a torque. The default is `0.3 m`.

Rated vertical load

Normalization of the vertical load force F_z, in Newtons (N). This entry also normalizes the longitudinal force F_x, which is generated by friction from the vertical load at the contact point. The default is `3000 N`.

Peak longitudinal force at rated load

Maximum longitudinal force F_x, in Newtons (N), the tire exerts on the wheel when the vertical load equals its rated value. The default is `3500 N`.

Slip at peak force at rated load

The value of the contact slip, in percent (%), when F_x equals its maximum value and F_z equals its rated value. (The contact slip in percent is the value of κ' times 100.) The default is `10%`.

Relaxation length at rated load

Tire relaxation length σ_κ, in meters (m), that determines how fast the tire flexes at the specified rated load. The default is `0.2 m`.

Tire Model

The tire is a flexible body in contact with the road surface and subject to slip. When a torque is applied to the wheel axle, the tire deforms, pushes on the ground (while subject to contact friction), and transfers the resulting reaction as a force back on the wheel, pushing the wheel forward or backward.

The Tire block models the tire as a rigid wheel-flexible body combination in point contact with the road. The model includes only longitudinal motion and no camber, turning, or lateral motion. The flexible body is represented by a deformable, circumferential spring. This figure and table define the tire model variables.

Tire Dynamics and Motion

Tire Model Variables and Constants

Symbol	Meaning and Units
r_e	Effective rolling radius (m)
I_w	Wheel-tire assembly inertia (kg·m^2)
τ_{drive}	Torque applied by the axle to the wheel (N·m)
V_x	Wheel center longitudinal velocity (m/s)
Ω	Wheel angular velocity (rad/s)
Ω'	Contact point angular velocity (rad/s)
$V_{sx} = V_x - r_e\Omega$	Wheel slip velocity (m/s)
$V'_{sx} = V_x - r_e\Omega'$	Contact point slip velocity (m/s)
$\kappa = -V_{sx}/\lvert V_x \rvert$	Wheel slip
$\kappa' = -V'_{sx}/\lvert V_x \rvert$	Contact patch slip

Symbol	Meaning and Units
u	Tire carcass compliance or deformation (m)
F_z	Vertical load on tire (N)
$F_x = f(u, \kappa', F_z)$	Longitudinal force exerted by the tire on the wheel at the contact point (N). Also a characteristic function f of the tire.
$C_{Fx} = (\partial F_x / \partial u)_0$	Tire carcass stiffness (N/m)
$\sigma_\kappa = (\partial F_x / \partial \kappa')_0 / (C_{Fx})$	Tire relaxation length (m)

Tire Deformation and Characteristic Function

If the tire were rigid and did not slip, it would roll and translate as $V_x = r_e\Omega$. But even a rigid tire slips, and the wheel slip velocity $V_{sx} = V_x - r_e\Omega \neq 0$. The *wheel slip* $\kappa = -V_{sx}/|V_x|$ is more convenient. For a locked, sliding tire, $\kappa = -1$.

The tire is also flexible. Because it deforms, the contact point turns at a different angular velocity Ω' from the wheel. The *contact point slip* $\kappa' = -V'_{sx}/|V_x|$, where $V'_{sx} = V_x - r_e\Omega'$.

The *tire deformation* u directly measures the difference of wheel and contact point slip and satisfies

$$\frac{du}{dt} = -(V_{sx} - V'_{sx})$$

A tire model must provide the longitudinal force F_x the tire exerts on the wheel once the deformation u, the contact slip κ', and the vertical load F_z are given. The *tire characteristic function* specifies this relationship in the steady state: $F_x = f(u, \kappa', F_z)$. The longitudinal force is almost linearly proportional to the vertical load, because F_x is generated by contact friction and the normal force F_z. (The relationship is somewhat nonlinear because of tire deformation and slip.) The dependence of F_x on κ' and u is more complex.

Tire Dynamics

In addition, the contact slip κ' and deformation u are not constant, because the rolling, stressed tire is not in a steady state. Before they can be used in the characteristic function, their time evolution must be accounted for. In this model, u and κ' are moderate to small. The relationships of F_x to u and u to κ' are then linear:

$$F_x = C_{Fx} \cdot u \ , \ C_{Fx} = \left(\frac{\partial F_x}{\partial u}\right)_{u=0}$$

$$u = \sigma_\kappa \cdot \kappa' \ , \ \sigma_\kappa = \left(\frac{\partial F_x}{\partial \kappa'}\right)_{\kappa'=0} \bigg/ \left(\frac{\partial F_x}{\partial u}\right)_{u=0}$$

so that the *tire stiffness* $C_{Fx} = C_{Fx}(\kappa', F_z)$ and *tire relaxation length* $\sigma_\kappa = \sigma_\kappa(\kappa', u, F_z)$. These properties are taken from empirical tire data.

For slowly-varying vertical loads F_z, the contact patch slip κ' evolves according to

$$\sigma_\kappa \cdot \frac{d\kappa'}{dt} + |V_x|\kappa' = r_e\Omega - V_x - \frac{1}{C_{Fx}}\left(\frac{\partial F_x}{\partial F_z}\right)\frac{dF_z}{dt}$$

The deformation u follows from σ_κ.

Finding the Wheel Motion

With the tire characteristic function $f(u, \kappa', F_z)$, the vertical load F_z, and the evolved u and κ', you can find the longitudinal force F_x and wheel velocity Ω. From these, the equations of motion determine the wheel angular motion (the angular velocity Ω) and longitudinal motion (the wheel center velocity V_x):

$$I_w \frac{d\Omega}{dt} = \tau_{\text{drive}} - r_e F_x$$

$$m\frac{dV_x}{dt} = F_x - mg \cdot \sin\beta$$

where β is the slope of the incline upon which the vehicle is traveling (positive for uphill), and m and g are the wheel load mass and the gravitational acceleration, respectively. τ_{drive} is the driveshaft torque applied to the wheel axis.

Relationship to Block Parameters

The effective rolling radius is r_e. The rated load normalizes the tire characteristic function $f(u, \kappa', F_z)$, and the peak force, slip at peak force, and relaxation length fields determine the peak and slope of $f(u, \kappa', F_z)$ and thus C_{Fx} and σ_κ.

Example

The demo model `drive_4wd_dynamics` combines two differentials with four tire-wheel assemblies to model the contact of tires with the road and the longitudinal vehicle motion.

Reference

H. B. Pacejka, *Tire and Vehicle Dynamics,* Society of Automotive Engineers/Butterworth-Heinemann, Oxford, 2002.

See Also

`Differential`, `Longitudinal Vehicle Dynamics`

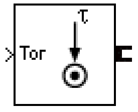
Purpose

Actuate a driveline axis with a specified torque

Library

Sensors & Actuators

Description

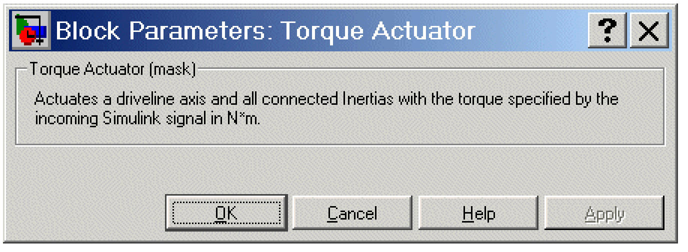


The Torque Actuator block actuates the connected driveline axis with a torque. You specify this torque as a Simulink input signal in Newton-meters.

The Torque Actuator block has one driveline port. You can connect it to a driveline axis

- By connecting the port to the end of the axis
- By branching a connection line off the main line and connecting it to the port

Dialog Box and Parameters



This block has no active parameters.

See Also

`Motion Actuator`, `Motion Sensor`, `Torque Sensor`

Torque Converter

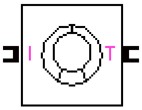
Purpose

Transfer torque between two driveline axes as a function of their relative angular velocity

Library

Dynamic Elements

Description

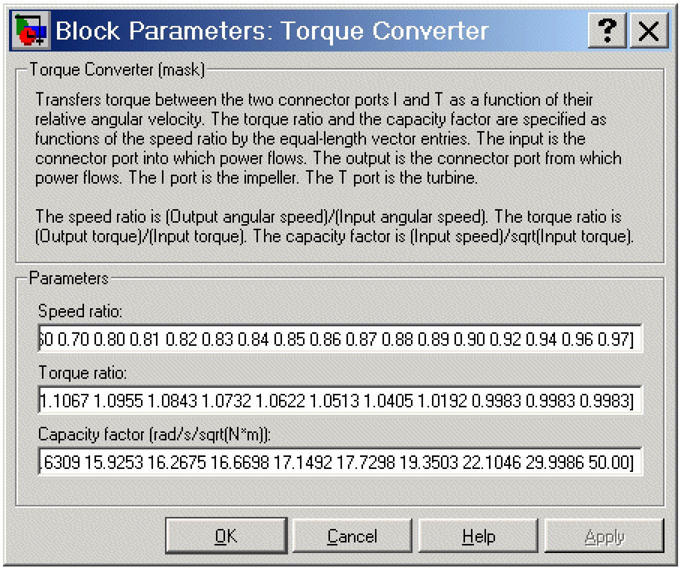


A torque converter couples two driveline axes, transferring torque and angular motion by the hydrodynamic action of a viscous fluid. Unlike a friction clutch, it cannot lock. The Torque Converter block models a torque converter acting between the two connector ports I and T as a function of the relative angular velocity of the two connected driveline axes. The input is the connector port into which power flows into the block. The output is the connector port from which power flows out of the block. The I port represents the impeller or pump. The T port represents the turbine. Forward power flow means power flowing from I to T. Reverse power flow means power flowing from T to I.

You specify the torque ratio and the capacity factor of the torque converter as discrete functions of the speed ratio with tabular vector entries. The speed ratio is the output angular speed divided by the input angular speed. You specify a range of speed ratio values from 0 up to, but not including, 1. The torque ratio is the output torque divided by the input torque. The capacity factor is the input speed divided by the square root of the input torque.

The three vectors of the independent and two dependent variable values must have the same length.

Dialog Box and Parameters



Speed ratio
: Vector of values of the block function's independent variable, the dimensionless speed ratio. These values must be greater than or equal to 0 and strictly less than 1.

Torque ratio
: Vector of values of the block function's first dependent variable, the dimensionless torque ratio. Each torque ratio value corresponds to a speed ratio value.

Capacity factor
: Vector of values of the block function's second dependent variable, the torque conversion capacity factor. Each capacity factor value corresponds to a speed ratio value. The units are radians/second/√(Newton-meters).

Example

The demo model `drive_torque_convert` simulates a torque converter.

Reference

Society of Automotive Engineers, *Hydrodynamic Drive Test Code (Surface Vehicle Recommended Practice),* SAE J643, May 2000.

See Also

`Controllable Friction Clutch`, `Diesel Engine`, `Gasoline Engine`

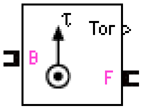
Purpose

Measure the torque transferred along a driveline axis

Library

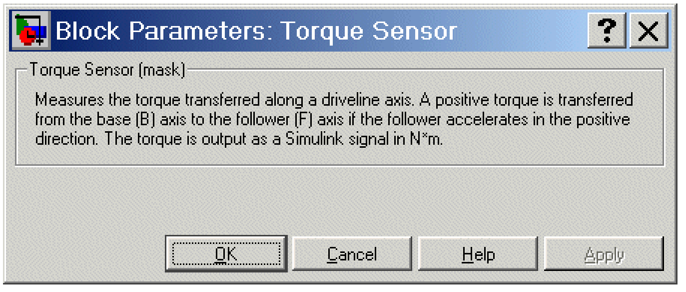
Sensors & Actuators

Description



The Torque Sensor block measures the torque transferred along a driveline axis at the point where the Torque Sensor is inserted. A positive torque is transferred from the base (B) axis to the follower (F) axis at that point if the follower axis accelerates positively with respect to the base and if no other torques are applied to the follower-connected inertias. The torque is output as a Simulink signal in Newton-meters.

Dialog Box and Parameters



This block has no active parameters.

See Also

`Motion Actuator`, `Motion Sensor`, `Torque Actuator`

Torsional Spring-Damper

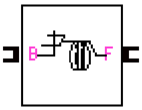
Purpose

Represent a damped torsional spring torque, with a free play gap, acting between two rotating axes

Library

Dynamic Elements

Description



The Torsional Spring-Damper block models a damped torsional spring-like torque acting between two rotating axes, the base (B) and the follower (F). This torque is a function of the relative displacement angle $\theta = \theta_F - \theta_B$ and relative angular velocity $\omega = d\theta/dt = \omega_F - \omega_B$.

$$\tau = -k(\theta - \theta_{back}) - b\omega, \text{ if } \theta > +\theta_{back}$$

$$\tau = -k(\theta + \theta_{back}) - b\omega, \text{ if } \theta < -\theta_{back}$$

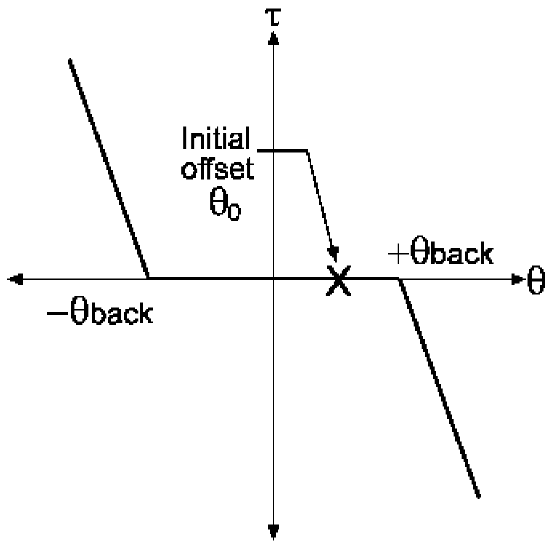
$$\tau = -b\omega, \text{ if } -\theta_{back} < \theta < +\theta_{back}$$

You specify the restoring torque spring constant or spring rate k as the stiffness and the kinetic friction torque constant b as the damping. Both constants must be nonnegative.

Backlash and Initial Offset

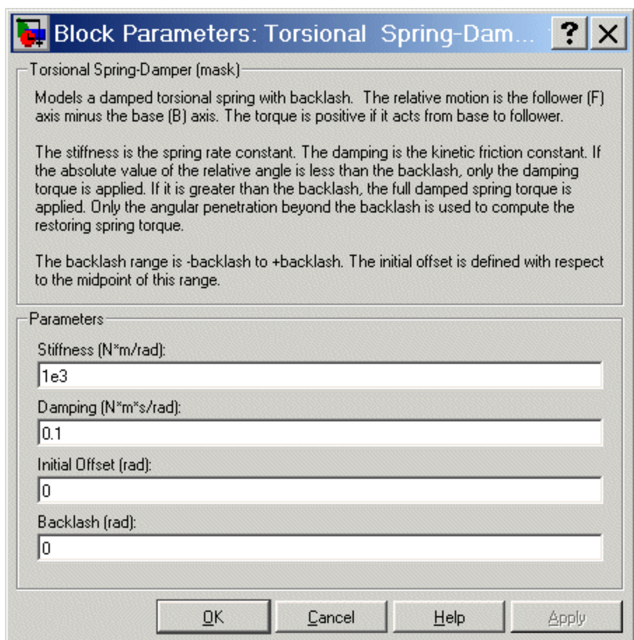
The backlash interval is a free play gap of size $2\theta_{back}$ across. If θ lies in this range, the spring torque is not applied, but the damping torque is. Above and below the backlash range, both the spring and damping torques are active.

You specify the initial relative displacement θ_0 as the initial offset, relative to the midway point between the backlash interval endpoints.



Torsional Spring-Damper Torque Law (Spring Only)

Dialog Box and Parameters



Stiffness

The spring constant or spring rate k for the restoring torque imposed by the spring. Must be nonnegative. The default is `1e3 N*m/rad` (Newton-meters/radian).

Damping

The damping constant b for the kinetic frictional torque imposed by the spring. Must be nonnegative. The default is `0.1 N*m*s/rad` (Newton-meters-seconds/radian).

Initial Offset

The initial angular offset θ_0 of the relative displacement θ, in radians. The default is `0 rad` (radians).

Backlash

The angular free play θ_{back} allowed in the torsional spring. Must be nonnegative. The default is `0 rad` (radians).

Example

The demo model `drive_spring` illustrates a simple torsional spring-damper system.

See Also

`Hard Stop`

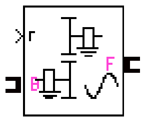
Purpose

Represent a gear with controllable, variable gear ratio

Library

Gears

Description



The Variable Ratio Gear block represents a gear box that constrains the two connected driveline axes, base (B) and follower (F), to corotate with a variable ratio that you can control. You can choose whether the follower axis rotates in the same or opposite direction as the base axis. If they rotate in the same direction, ω_F and ω_B have the same sign. If they rotate in opposite directions, ω_F and ω_B have opposite signs.

You specify the variable gear ratio as a function of time with the Simulink input signal r.

Axis Motion and Constraint

The Variable Ratio Gear imposes a single constraint, specified by the variable gear ratio $g(t)_{FB}$, on the motions and torques of the two axes:

$$\pm g(t)_{FB} = \omega_B/\omega_F = \tau_F/\tau_B$$

Caution The gear ratio g_{FB} must be strictly positive. If any gear ratio equals 0 or becomes negative at any time during a simulation, SimDriveline stops with an error.

The Effect of Coriolis Acceleration

With the Variable Ratio Gear block, you can choose to include or not include the effect of Coriolis acceleration on the gear motion. If you choose to include it, you must supply the first derivative, dg_{FB}/dt, of the gear ratio as another Simulink input signal r.

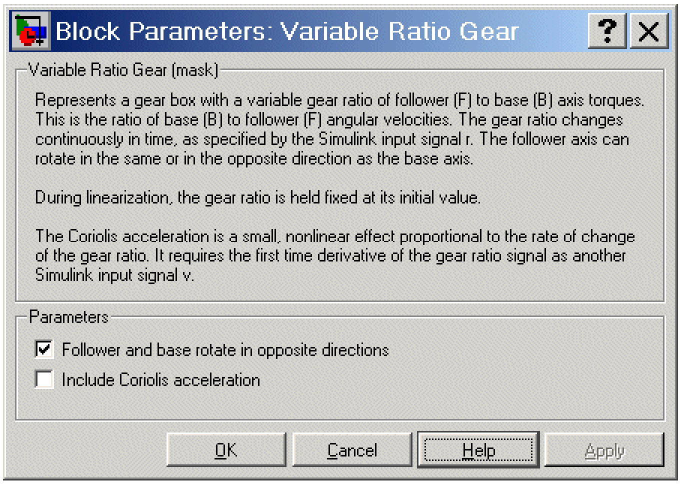
The Coriolis acceleration is a small, nonlinear effect proportional to the angular velocity and the first derivative of the gear ratio:

$$d\omega_B/dt = g_{FB} \cdot d\omega_F/dt + \omega_F \cdot dg_{FB}/dt$$

The Effect of Linearization

If you simulate your model in linearization mode, SimDriveline holds the variable gear ratio $g(t)_{FB}$ fixed at its initial value, $g(0)_{FB}$. You choose the **Simulation mode** in the Driveline Environment dialog box.

Dialog Box and Parameters



Follower and base rotate in opposite directions
Select to make the follower and base axes corotate in opposite directions. The default is selected.

Include Coriolis acceleration
Select to include the small nonlinear effect of the nonzero first derivative of the variable gear ratio in the driveline dynamics. The default is unselected.

See Also

`Driveline Environment`, `Simple Gear`

A

Technical Conventions

Driveline Abbreviations and Conventions

An important abbreviation is DoF, which means *degree of freedom* and refers to one coordinate of angular motion.

Standard symbols for angular motion analysis include the following:

Symbol	Meaning (Units)
r	Gear radius (meters)
N	Number of gear teeth
θ	Angle (radians)
ω	Angular velocity (radians/second)
τ	Torque (Newton-meters)

Gear Ratios

For a pair of coupled, coplanar gear wheels, the gear ratio g_{21} of gear 2 to gear 1 is defined as the ratio of the second gear wheel radius to the first. This definition is equivalent to the ratio of the number of teeth on the second gear wheel to the number of teeth on the first.

$$g_{21} \equiv r_2/r_1 = N_2/N_1$$

The gear ratio is the ratio of torques and the reciprocal of the angular velocity ratio.

$$g_{21} = \tau_2/\tau_1 = \omega_1/\omega_2$$

For gear boxes made of more than two gear wheels, the gear ratio is defined to be the ratio of torques or the reciprocal of the ratio of angular velocities, between the output and input shafts.

If the gear is reversing, the output ω and τ have opposite signs from the input ω and τ.

Driveline Units

SimDriveline accepts MKS (SI) units only.

Quantity	Unit
Length	meter (m)
Angle	radian (rad)
Time	second (s)
Angular Velocity	radians/second (rad/s)
Angular Acceleration	radians/second^2 (rad/s^2)
Mass	kilogram (kg)
Force	Newton (N)
Inertia	kilogram-meter^2 (kg-m^2)
Torque	Newton-meter (N-m)

Bibliography

Goodman, L.E., and W.H. Warner, *Statics* (1964) and *Dynamics* (1965), New York, Dover Publications, 2001.

Jurgen, R.K., *Electronic Transmission Controls,* Troy, Michigan, Society of Automotive Engineers, 2000.

Juvinall, R.C., *Fundamentals of Machine Component Design,* New York, John Wiley & Sons, 1983.

The MathWorks, Inc., `www.mathworks.com/products/industry/auto/`, Products: Industry: Automotive.

Meriam, J.L., and L.G. Kraige, *Dynamics*, Volume 2 of *Engineering Mechanics,* New York, John Wiley & Sons, 1987.

Nwagboso, C.O., *Automotive Sensory Systems,* London, Chapman & Hall, 1993.

Society of Automotive Engineers, `www.sae.org`, see “SAE Store”.

Wong, J.Y., *Theory of Ground Vehicles,* 3rd Ed., New York, Interscience, 2001.

Index