SimHydraulics

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

508-647-7000 (Phone)

508-647-7001 (Fax)

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

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

SimHydraulics User's Guide

© COPYRIGHT 2006 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB, Simulink, Stateflow, Handle Graphics, Real-Time Workshop, and xPC TargetBox are registered trademarks, and SimBiology, SimEvents, and SimHydraulics are trademarks of The MathWorks, Inc.

Other product or brand names are trademarks or registered trademarks of their respective holders.

Patents

The MathWorks products are protected by one or more U.S. patents. Please see `www.mathworks.com/patents` for more information.

Revision History

March 2006	Online only	New for Version 1.0 (Release 2006a+)
September 2006	Online only	Revised for Version 1.1 (Release 2006b)

Contents

1 Getting Started

2 Modeling Hydraulic Systems

3 Running Hydraulic Models

Glossary

A Examples

Index

Getting Started

What Is SimHydraulics?

SimHydraulics is a modeling environment for the engineering design and simulation of hydraulic power and control systems within Simulink® and MATLAB®. It is based on the Physical Network approach and contains a comprehensive library of hydraulic blocks, as well as one-dimensional translational and rotational mechanical elements and utility blocks.

SimHydraulics and Physical Modeling

SimHydraulics 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 SimHydraulics User's Guide assumes that you have some experience with modeling hydraulic systems, as well as with building and running models in Simulink.

Getting Online Help for SimHydraulics

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

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.

Related Products

SimHydraulics is the extension of the Physical Modeling family, expanding Simulink capabilities to model and simulate hydraulic and inherently connected to them mechanical one-dimensional translational and rotational elements and devices.

Requirements for SimHydraulics

You must have the following products installed to use SimHydraulics:

- MATLAB
- Simulink

Other Related Products

The related products listed in the SimHydraulics 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 SimHydraulics in various applications.

The Physical Modeling Product Family

In addition to SimHydraulics, the Physical Modeling product family includes:

- SimMechanics, for modeling and simulating general mechanical systems
- SimDriveline, for modeling and simulating powertrain systems
- 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
- The MathWorks Web site at `www.mathworks.com`

Basic Principles of Modeling Physical Networks

SimHydraulics is a set of block libraries and special simulation features for use in the Simulink environment. It is based on the Physical Network approach. You connect SimHydraulics blocks to regular Simulink blocks through special Converter blocks.

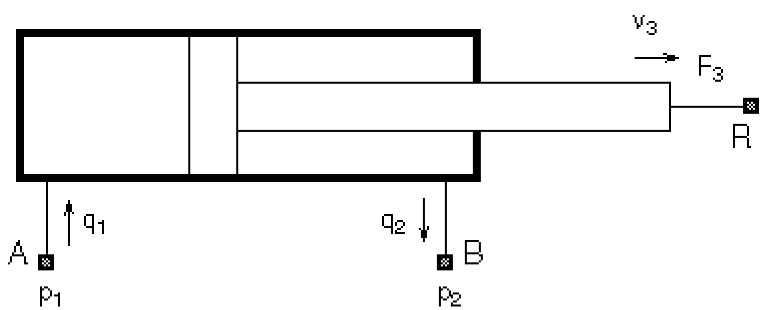
Simulink blocks represent basic mathematical operations. When you connect Simulink blocks together, the resulting diagram is equivalent to the mathematical model, or representation, of the system under design. Unlike Simulink, SimHydraulics uses a network representation of the system under design, based on the Physical Network approach. According to this approach, each system is represented as consisting of functional elements that interact with each other by exchanging energy through their ports.

These connection ports are bidirectional. They mimic physical connections between elements. Connecting SimHydraulics blocks together is analogous to connecting real components, such as pumps, valves, and so on. In other words, SimHydraulics diagrams mimic the physical system layout. If physical components can be connected, their models can be connected, too. You do not have to specify flow directions and information flow when connecting SimHydraulics blocks, just as you do not have to specify this information when you connect real physical components. The Physical Network approach, with its Through and Across variables and bidirectional physical connections, automatically resolves all the traditional issues with variables, directionality, and so on.

The number of connection ports for each element is determined by the number of energy flows it exchanges with other elements in the system, and depends on the level of idealization. For example, a fixed-displacement hydraulic pump in its simplest form can be represented as a two-port element, with one energy flow associated with the inlet (suction) and the other with the outlet. In this representation, the angular velocity of the driving shaft is assumed constant, making it possible to neglect the energy exchange between the pump and the shaft. To account for a variable driving torque, you need a third port associated with the driving shaft.

An energy flow is characterized by its variables. Each energy flow is associated with two variables, one Through and one Across (see “Variable Types” on page 1-5 for more information). Usually, these are the variables whose product is the energy flow in watts. They are called the basic, or conjugate, variables. For example, the basic variables for mechanical translational systems are force and velocity, for mechanical rotational systems—torque and angular velocity, for hydraulic systems—flow rate and pressure, for electrical systems—current and voltage.

The following example illustrates a Physical Network representation of a double-acting hydraulic cylinder.



The element is represented with three energy flows: two flows of hydraulic energy through the inlet and outlet of the cylinder and a flow of mechanical energy associated with the rod motion. It therefore has the following three connector ports:

- A — Hydraulic conserving port associated with pressure p_1 (an Across variable) and flow rate q_1 (a Through variable)
- B — Hydraulic conserving port associated with pressure p_2 (an Across variable) and flow rate q_2 (a Through variable)
- R — Mechanical translational conserving port associated with rod velocity v_3 (an Across variable) and force F_3 (a Through variable)

See "Connector Ports and Connection Lines" on page 1-10 for more information on connector port types.

Variable Types

SimHydraulics supports two types of variables:

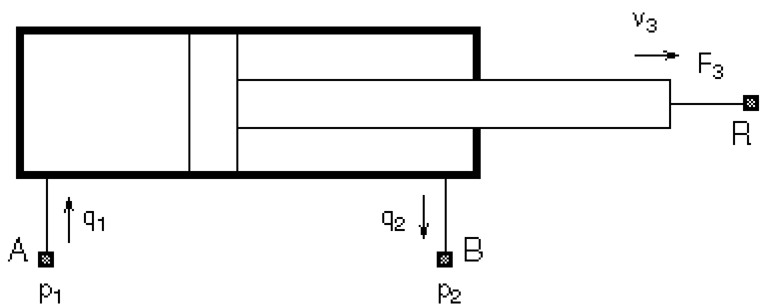
- Through — Variables that are measured with a gauge connected in series to an element.
- Across — Variables that are measured with a gauge connected in parallel to an element.

The following table contains examples of Through and Across variables specific to hydraulic, mechanical, electrical, and thermal systems:

Technology	Across Variables	Through Variables
Mechanical	Position Linear velocity Linear acceleration Angle Angular velocity Angular acceleration	Force Torque
Hydraulic	Pressure	Flow rate Volume
Electrical	Voltage Flux	Current Charge
Thermal	Temperature	Heat flow Enthalpy Entropy

Building the Mathematical Model

Through and Across variables associated with all the energy flows form the basis of the mathematical model of the block.



For example, the model of a double-acting hydraulic cylinder shown in the previous illustration can be described with a simple set of equations:

$$F_3 = p_1 \bullet A_1 - p_2 \bullet A_2$$

$$q_1 = A_1 \bullet v_3$$

$$q_2 = A_2 \bullet v_3$$

where

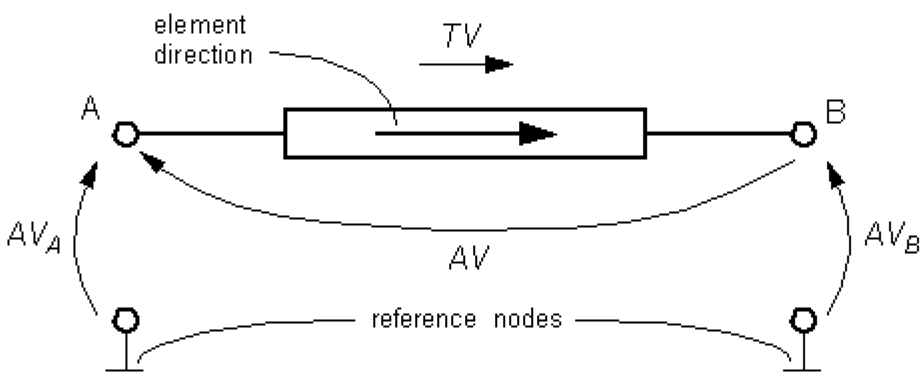
q_1, q_2	Flow rates through ports A and B, respectively (Through variables)
p_1, p_2	Gauge pressures at ports A and B, respectively (Across variables)
A_1, A_2	Piston effective areas
F_3	Rod force (Through variable)
v_3	Rod velocity (Across variable)

The model could be considerably more complex, for example, it could account for friction, fluid compressibility, inertia of the moving parts, and so on. For all these different mathematical models, however, the element configuration (that is, the number and type of ports and the associated Through and Across variables) would remain the same, meaning that the Physical Network approach lets you substitute models of different levels of complexity without introducing any changes to the schematic. For example, you can start developing your system by using the Resistive Tube block from the Foundation library, which accounts only for friction losses. At a later stage in development, you may want to account for fluid compressibility. You can then replace it with a Hydraulic Pipeline block from the SimHydraulics library or, depending on your application, even with a Segmented Pipeline block if you also need to account for fluid inertia. This modeling principle is called incremental modeling.

Direction of Variables

Each variable is characterized by its magnitude and sign. The sign is the result of measurement orientation. The same variable can be positive or negative, depending on the polarity of a measurement gauge. That is why it is very important to apply exactly the same rule to all the variables in the Physical Network.

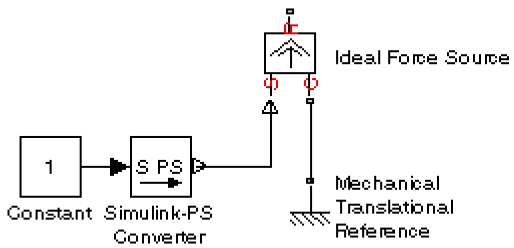
Elements with only two ports are characterized with one pair of variables, a Through variable and an Across variable. Since these variables are closely related, their orientation is defined with one direction. For example, if an element is oriented from port A to port B, it implies that the Through variable (*TV*) is positive if it "flows" from A to B, and the Across variable is determined as $AV = AV_A - AV_B$, where AV_A and AV_B are the element node potentials or, in other words, the values of this Across variable at ports A and B, respectively.



This approach to the direction of variables has the following benefits:

- Provides a simple and consistent way to determine whether an element is active or passive. Energy is one of the most important characteristics to be determined during simulation. If the variables direction, or sign, is determined as described above, their product (that is, the energy) is positive if the element consumes energy, and is negative if it provides energy to a system. This rule is followed throughout SimHydraulics.
- Simplifies the model description. Symbol A $\rightarrow$ B is enough to specify variable polarity for both the Across and the Through variables.
- Lets you apply the oriented graph theory to network analysis and design.

As an example of variables direction rules, let us consider the Ideal Force Source block. In this block, as in many other mechanical blocks, port C is associated with the source reference point (case), and port R is associated with the rod.



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

All the elements in a network are divided into active and passive elements, depending on whether they deliver energy to the system or dissipate (or store) it. Active elements (force and velocity sources, flow rate and pressure sources, etc.) must be oriented strictly in accordance with the line of action or function that they are expected to perform in the system, while passive elements (dampers, resistors, springs, pipelines, etc.) can be oriented either way.

Connector Ports and Connection Lines

SimHydraulics blocks may have the following types of ports:

- Physical Conserving ports — Bidirectional hydraulic or mechanical ports that represent physical connections and relate physical variables based on the Physical Network approach.
- Physical Signal ports — Unidirectional ports transferring signals that use an internal SimHydraulics engine for computations.

Each of these ports and connections between them are described in greater detail below.

Physical Conserving Ports

SimHydraulics blocks have special Conserving ports . You connect Conserving ports with Physical connection lines, distinct from normal Simulink lines. Physical connection lines have no inherent directionality and represent the exchange of energy flows, according to the Physical Network approach.

- You can connect Conserving ports only to other Conserving ports of the same type.
- The Physical connection lines that connect Conserving ports together are bidirectional lines that carry physical variables (Across and Through variables, as described above) rather than signals. You cannot connect Physical lines to Simulink ports or to Physical Signal ports.
- Two directly connected Conserving ports must have the same values for all their Across variables (such as pressure or angular velocity).
- You can branch Physical connection lines. When you do so, components directly connected with one another continue to share the same Across variables. Any Through variable (such as flow rate or torque) transferred along the Physical connection line is divided among the multiple components connected by the branches. How the Through variable is divided is determined by the system dynamics.

For each Through variable, the sum of all its values flowing into a branch point equals the sum of all its values flowing out.

The following table lists the types of Physical Conserving ports used in SimHydraulics, along with the Through and Across variables associated with each type:

Port Type	Across Variable	Through Variable
Hydraulic	Pressure	Flow rate
Mechanical translational	Translational velocity	Force
Mechanical rotational	Angular velocity	Torque

Physical Signal Ports

Physical Signal ports ▷ carry signals between SimHydraulics blocks. You connect them with regular connection lines, similar to Simulink signal connections. Physical Signal ports are used in SimHydraulics block diagrams instead of Simulink input and output ports to increase computation speed and avoid issues with algebraic loops. Unlike Simulink signals, which are essentially unitless, physical signals can have units associated with them. You specify the units along with the parameter values in the block dialogs, and SimHydraulics performs the necessary unit conversion operations when solving a physical network.

SimHydraulics contains a Physical Signals block library. These blocks perform math operations and other functions on physical signals, and allow you to graphically implement equations inside the Physical Network.

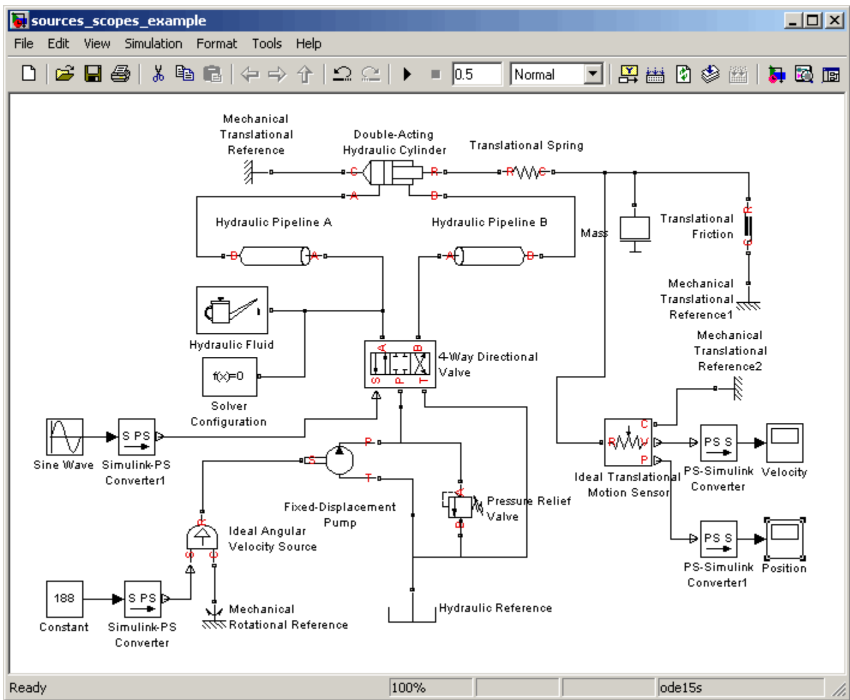
Connecting SimHydraulics Diagrams to Simulink Sources and Scopes

SimHydraulics block diagrams use physical signals instead of regular Simulink signals. Therefore, you need converter blocks to connect SimHydraulics diagrams to Simulink sources and scopes.

Use the Simulink-PS Converter block to connect Simulink sources or other Simulink blocks to the inputs of a Physical Network diagram. You can also use it to specify the input signal units. For more information, see the Simulink-PS Converter block reference page.

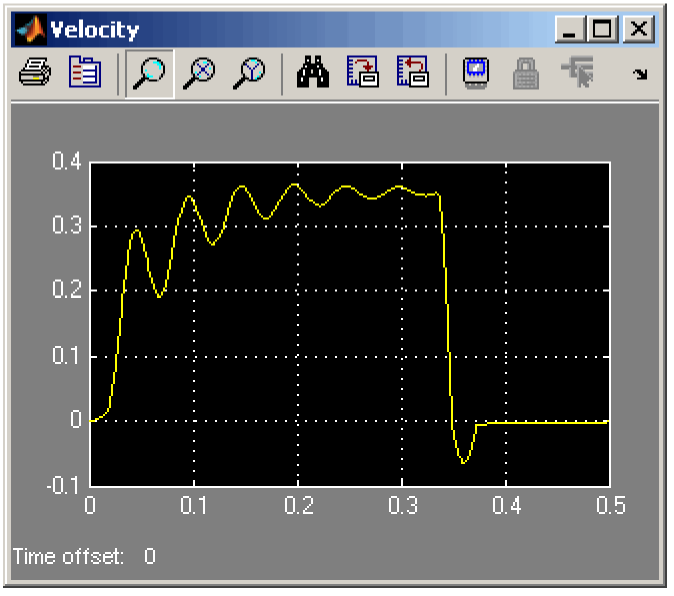
Use the PS-Simulink Converter block to connect outputs of a Physical Network diagram to Simulink scopes or other Simulink blocks. You can also use it to specify the desired output signal units. For more information, see the PS-Simulink Converter block reference page.

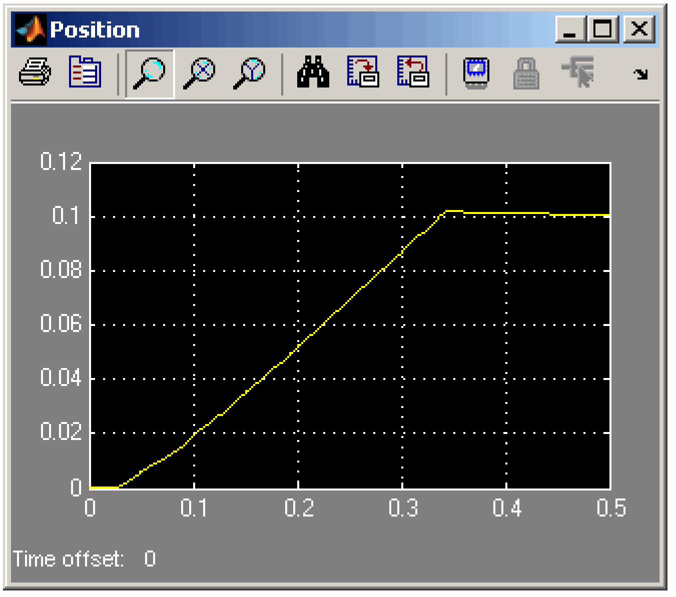
The following illustration shows an example of a SimHydraulics diagram, where Simulink sources provide input signals and Simulink scopes are used to output the simulation results.



The system in the diagram consists of a double-acting hydraulic cylinder controlled by a 4-way directional valve. The valve is driven by a Simulink signal, which is first converted into a Physical Signal with the Simulink-PS Converter block (named `Simulink-PS Converter1` in this diagram) and then fed to the valve control port S. The power unit of the system is built of a fixed-displacement pump, pressure-relief valve, and a prime mover simulated with an Ideal Angular Velocity Source block. The output of the source is controlled by the Simulink Constant source block connected through the Simulink-PS Converter block. The cylinder is connected to the valve through two Hydraulic Pipeline blocks. The mechanical load consists of inertia (simulated by a Mass block), spring (Translational Spring block), and friction (Translational Friction block).

Two variables are set for viewing. They are load velocity (Velocity) and load position (Position). Both of these variables are monitored with the Ideal Translational Motion Sensor block and displayed by Simulink Scope blocks, connected through PS-Simulink Converter blocks.




Position
0.12
0.1
0.08
0.06
0.04
0.02
0
0
0.1
0.2
0.3
0.4
0.5
Time offset: 0

Working with Physical Units

Unlike Simulink signals, which are essentially unitless, physical signals can have units associated with them. You specify the units along with the parameter values in the block dialogs, and SimHydraulics performs the necessary unit conversion operations when solving a physical network. SimHydraulics blocks support standard measurement systems. The default block units are meter-kilogram-second or MKS (SI).

This section covers the following topics:

- "Unit Definitions" on page 1-15
- "Specifying Units in Block Dialogs" on page 1-18

Unit Definitions

SimHydraulics unit names are defined in the `pm_units.m` file, which is shipped with the product. You can open this file to see how the SimHydraulics physical units are defined, and also as an example when adding your own units. This file is located in the directory *matlabroot*`\toolbox\physmod\pm_util\pm_util`.

Default registered units and their abbreviations are listed in the following table. Use the `pm_getunits` command to get an up-to-date list of units currently defined in your unit registry. Use the `pm_adddimension` and `pm_addunit` commands to define additional units.

Physical Unit Abbreviations Defined in SimHydraulics

Quantity	Abbreviation	Unit
Angle	`rad`	Radian
	`deg`	Degree
	`rev`	Revolution
Angular velocity	`rpm`	Revolutions/minute
Flow rate	`lpm`	Liter/minute
	`gpm`	Gallon/minute

Physical Unit Abbreviations Defined in SimHydraulics (Continued)

Quantity	Abbreviation	Unit
Force	N	Newton
	dyn	Dyne
	lbf	Pound-force
Energy	J	Joule
	Btu	British Thermal Unit
Length	m	Meter
	cm	Centimeter
	mm	Millimeter
	km	Kilometer
	in	Inch
	ft	Foot
	mi	Mile
	yd	Yard
Mass	kg	Kilogram
	g	Gram
	mg	Milligram
	lbm	Pound mass
	oz	Ounce
	slug	Slug
Pressure	Pa	Pascal
	bar	Bar
	atm	Atmosphere
	psi	Pound/inch^2

Physical Unit Abbreviations Defined in SimHydraulics (Continued)

Quantity	Abbreviation	Unit
Power	W	Watt
	HP	Horsepower
Temperature	K	Kelvin
Time	s	Second
	min	Minute
	hr	Hour
Velocity	mph	Miles/hour
	fpm	Feet/minute
	fps	Feet/second
Viscosity absolute	Poise	Poise
	cP	Centipoise
	reyn	Reyn
Viscosity kinematic	St	Stokes
	cSt	Centistokes
	Newt	Newt
Volume	l	Liter
	gal	Gallon

Note This table lists the unit abbreviations defined in the product. See the following section, “Specifying Units in Block Dialogs” on page 1-18, for information on how to use the abbreviations above, or mathematical expressions with these abbreviations, to specify units for the parameter values in the block dialogs.

Specifying Units in Block Dialogs

Most of the block dialogs have pull-down menus of units next to a parameter value. For example, in the Pressure Compensator block, the pull-down menu for the **Maximum passage area** parameter contains `in^2`, `ft^2`, `mm^2`, `cm^2`, `m^2`, and `km^2`, and the pull-down menu for the **Valve pressure setting** parameter contains `Pa`, `bar`, `psi`, and `atm`.

The converter blocks (Simulink-PS Converter and PS-Simulink Converter) let you type a unit name, or a mathematical expression with unit names, in the **Unit** field of the block dialog. You can use the abbreviations for the units defined in your registry, or any valid mathematical expressions with these abbreviations. For example, you can specify torque in newton-meters (`N*m`) or pound-feet (`lbf*ft`) . To specify velocity, you can use one of the defined unit abbreviations (`mph`, `fpm`, `fps`), or an expression based on any combination of the defined units of length and time, such as meters/second (`m/s`), millimeters/second (`mm/s`), inch/minute (`in/min`), and so on.

The following operators are supported in the unit mathematical expressions:

`*`	Multiplication
`/`	Division
`^`	Power
`+, -`	Plus, minus — for exponents only
`()`	Brackets to specify evaluation order

Metric unit prefixes, such as *kilo*, *milli*, *micro*, are not supported. For example, if you want to use milliliter as a unit of volume, you have to add it to the unit registry:

```
pm_addunit('ml', 0.001, 'l');
```

Note Currently, physical units are not propagated through the blocks in the Physical Signals library, such as PS Add, PS Gain, and so on. This feature will be added in subsequent releases.

2

Modeling Hydraulic Systems

This chapter introduces you to modeling hydraulic systems with SimHydraulics. After reviewing the essential rules of connecting blocks and working with fluids, it moves you from the simplest model to a complete hydraulic system 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.

Introducing the SimHydraulics Block Libraries

This section shows you how to open and use the block libraries in SimHydraulics:

- "Library Structure Overview" on page 2-2
- "Using the Simulink Library Browser to Access the Block Libraries" on page 2-2
- "Using the Command Prompt to Access the Block Libraries" on page 2-3

Library Structure Overview

SimHydraulics uses the Physical Networks library as its main library. This library organizes its blocks into the following libraries:

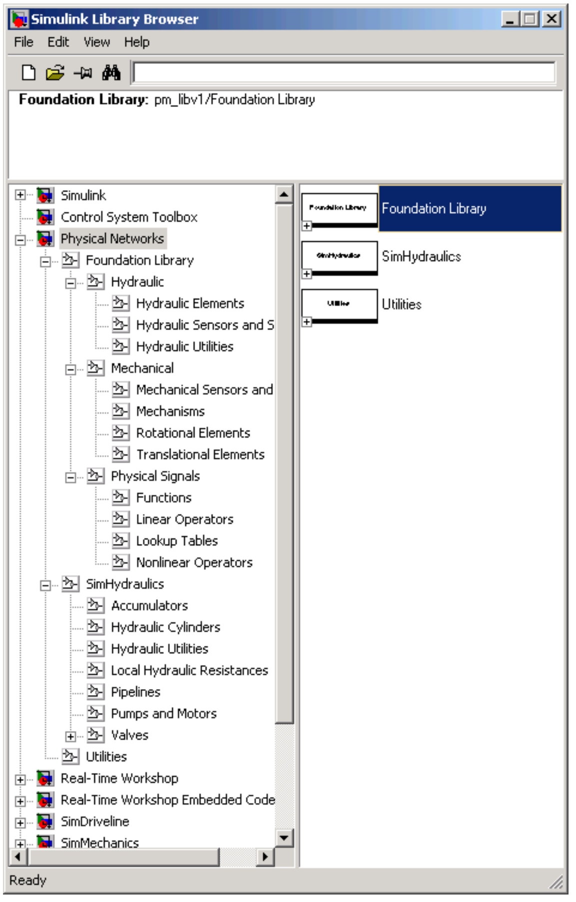
- Foundation library — Contains basic hydraulic, mechanical, and physical signal blocks
- SimHydraulics library — Contains advanced hydraulic diagram blocks, such as valves, cylinders, pipelines, pumps, and accumulators
- Utilities library — Contains essential environment blocks for creating Physical Networks models

You can combine all these blocks in your SimHydraulics diagrams to model hydraulic systems. You can also use the basic Simulink blocks in your diagrams, such as sources or scopes. See "Connecting SimHydraulics Diagrams to Simulink Sources and Scopes" on page 1-11 for more information on how to do this.

Using the Simulink Library Browser to Access the Block Libraries

On Microsoft Windows, you can access the blocks through the Simulink Library Browser. To display the Library Browser, click the **Library Browser** button in the toolbar of the MATLAB desktop or Simulink model window:

Alternatively, you can type `simulink` in the MATLAB Command Window. Then expand the Physical Networks entry in the contents tree.



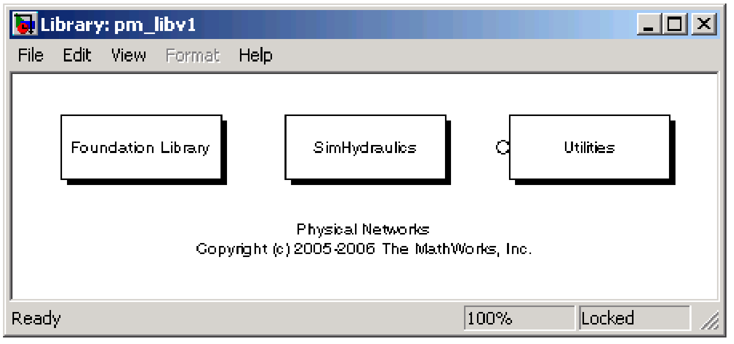
Using the Command Prompt to Access the Block Libraries

On any platform:

- To open the Physical Networks library, type `pn_lib` in the MATLAB Command Window.

- To open the main Simulink library (to access generic Simulink blocks), type `simulink` in the MATLAB Command Window.

The Physical Networks library consists of three top-level libraries, as shown in the following illustration. Some of these libraries contain second-level and third-level sublibraries. You can expand each library by double-clicking its icon. For more details on library hierarchy and descriptions of block categories, see “Blocks — By Category”.



Essential Steps to Building a Hydraulic Model

The rules that you must follow when building a hydraulic model with SimHydraulics are described in “Basic Principles of Modeling Physical Networks” on page 1-4. This section briefly reviews these rules.

- SimHydraulics blocks, in general, feature both Conserving ports and Physical Signal inports and outports.
- There are three types of Physical Conserving ports used in SimHydraulics: hydraulic, mechanical translational, and mechanical rotational. Each type has specific Through and Across variables associated with it.
- You can connect Conserving ports only to other Conserving ports of the same type.
- The Physical connection lines that connect Conserving ports together are bidirectional lines that carry physical variables (Across and Through variables, as described above) rather than signals. You cannot connect Physical lines to Simulink ports or to Physical Signal ports.
- Two directly connected Conserving ports must have the same values for all their Across variables (such as pressure or angular velocity).
- You can branch Physical connection lines. When you do so, components directly connected with one another continue to share the same Across variables. Any Through variable (such as flow rate or torque) transferred along the Physical connection line is divided among the multiple components connected by the branches. How the Through variable is divided is determined by the system dynamics.

  For each Through variable, the sum of all its values flowing into a branch point equals the sum of all its values flowing out.
- You can connect Physical Signal ports to other Physical Signal ports with regular connection lines, similar to Simulink signal connections. These connection lines carry physical signals between SimHydraulics blocks.
- You can connect Physical Signal ports to Simulink ports through special converter blocks. Use the Simulink-PS Converter block to connect Simulink outports to Physical Signal inports. Use the PS-Simulink Converter block to connect Physical Signal outports to Simulink inports.

- Unlike Simulink signals, which are essentially unitless, Physical Signals can have units associated with them. SimHydraulics block dialogs let you specify the units along with the parameter values, where appropriate. Use the converter blocks to associate units with an input signal and to specify the desired output signal units.

For examples of applying these rules when creating an actual hydraulic model, see "Creating a Simple Model" on page 2-8.

The MathWorks recommends that you build, simulate, and test your model incrementally. Start with an idealized, simplified model of your system, simulate it, verify that it works the way you expected. Then incrementally make your model more realistic, factoring in effects such as friction loss, motor shaft compliance, hard stops, and the other things that describe real-world phenomena. Simulate and test your model at every incremental step. Use subsystems to capture the model hierarchy, and simulate and test your subsystems separately before testing the whole model configuration. This approach helps you keep your models well organized and makes it easier to troubleshoot them.

Working with Fluids

A change in the working fluid of your SimHydraulics model affects the global parameters of the system. Global parameters, determined by the type of working fluid, are used in equations for most hydraulic blocks. For example, valves, orifices, and pipelines use fluid density and fluid kinematic viscosity; chambers and cylinders use fluid bulk modulus; and so on. When you change the type of fluid, the appropriate changes to the global parameter values are propagated to all the blocks in the hydraulic circuit.

Each topologically distinct hydraulic circuit in a diagram requires exactly one Hydraulic Fluid block or Custom Hydraulic Fluid block to be connected to it.

- The Custom Hydraulic Fluid block lets you directly specify the fluid properties, such as fluid density, kinematic viscosity, bulk modulus, and the amount of entrapped air, in the block dialog.
- The Hydraulic Fluid block lets you select a type of fluid from a predefined list and specify the amount of entrapped air and fluid temperature.

SimHydraulics determines the fluid properties associated with this type of fluid and these conditions, and displays them in the block dialog.

In both cases, SimHydraulics then applies the fluid properties as global parameters to all the blocks in the hydraulic circuit.

Creating a Simple Model

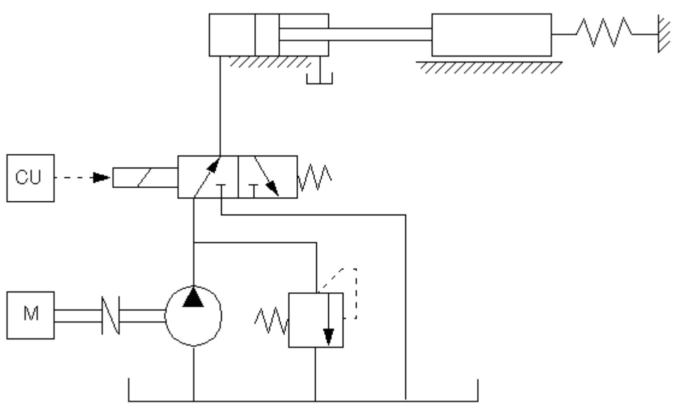
In this example, you are going to model a simple hydraulic system and observe its behavior under various conditions. This tutorial illustrates the essential steps to building a hydraulic model, described in the previous section, and makes you familiar with using the basic SimHydraulics blocks.

The tutorial consists of the following tasks:

- "Building a SimHydraulics Diagram" on page 2-8
- "Modifying Initial Settings" on page 2-16
- "Running the Simulation" on page 2-19
- "Adjusting the Parameters" on page 2-21

Building a SimHydraulics Diagram

The following schematic represents the model you are about to build. It contains a single-acting hydraulic cylinder, which is controlled by an electrically operated 3-way directional valve. The cylinder drives a load consisting of a mass, viscous friction, and preloaded spring.



The power unit consists of a motor, a positive-displacement pump, and a pressure relief valve. Depending on its characteristics, such a power unit can be modeled in a variety of ways, as described in "Modeling Power Units" on page 2-26. In this example, the pump unit is assumed to be powerful enough to maintain constant pressure at the valve inlet. Therefore, we are going to represent it in the diagram by an Ideal Hydraulic Pressure Source block.

To create an equivalent SimHydraulics diagram, follow these steps:

1 Open the Physical Networks and Simulink block libraries, as described in "Introducing the SimHydraulics Block Libraries" on page 2-2.

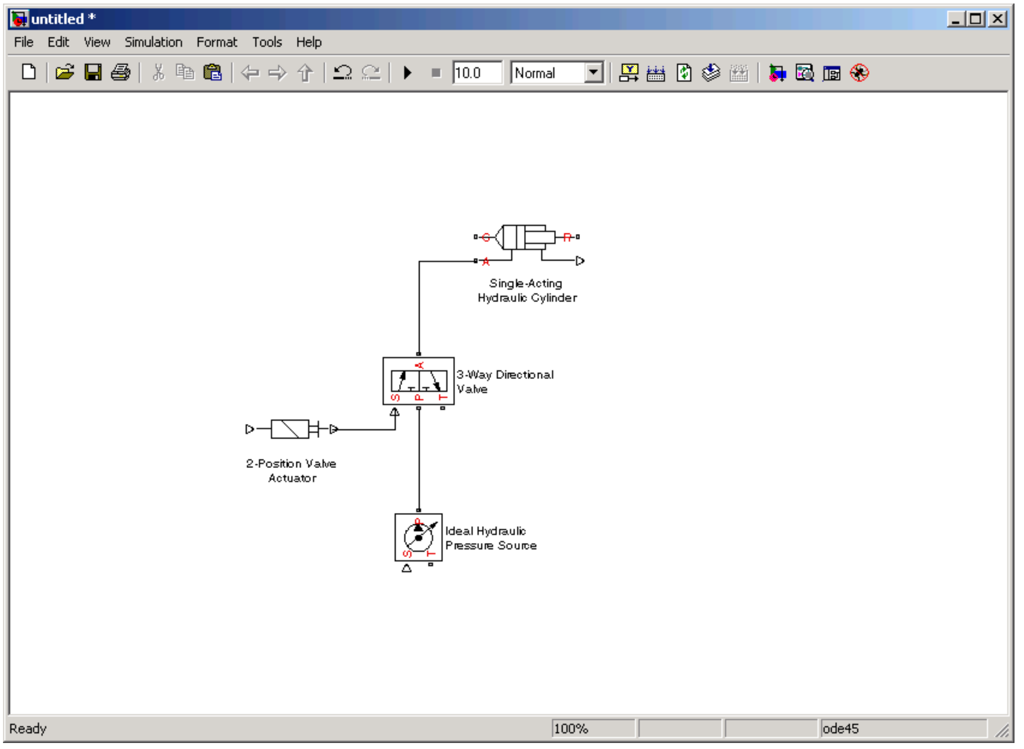
2 Create a new Simulink model. To do this, click the **New** button on the Library Browser's toolbar (Windows only) or choose **New** from the library window's **File** menu and select **Model**. Simulink creates an empty model in memory and displays it in a new model editor window.

3 Open the Physical Networks > Foundation Library > Hydraulic > Hydraulic Sensors and Sources library and drag and drop the Ideal Hydraulic Pressure Source block into the model window.

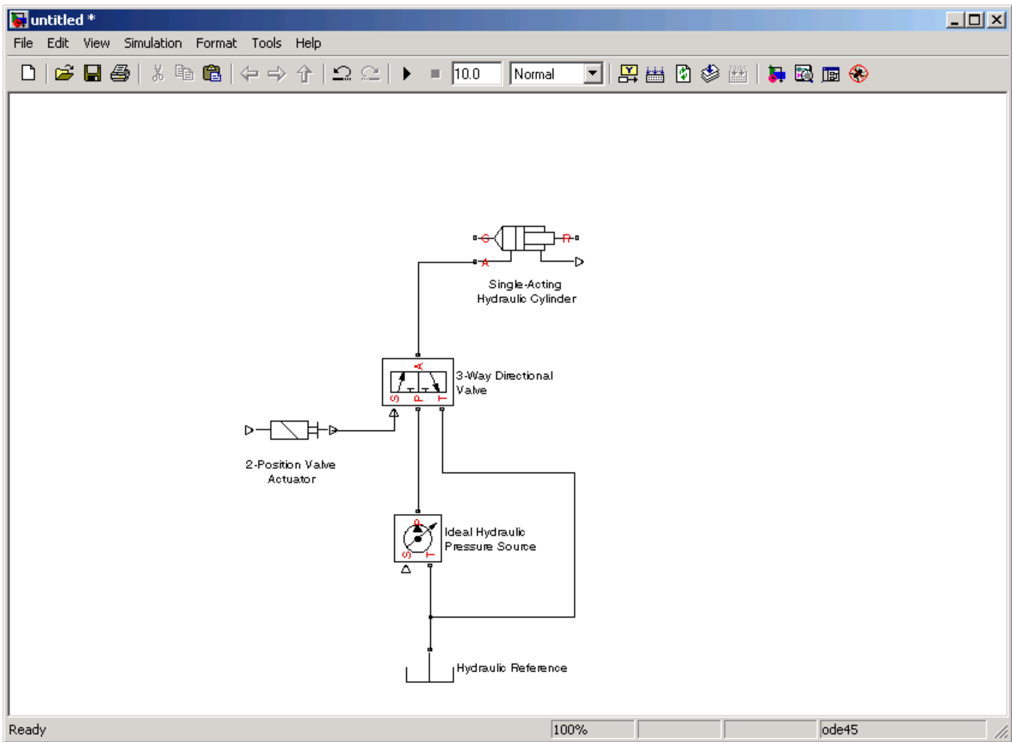
4 Open the Physical Networks > SimHydraulics > Hydraulic Cylinders library and place the Single-Acting Hydraulic Cylinder block into the model window.

5 To model the valve, open the Physical Networks > SimHydraulics > Valves library. Place the 3-Way Directional Valve block, found in the Directional Valves sublibrary, and the 2-Position Valve Actuator block, found in the Valve Actuators sublibrary, into the model window.

6 Connect the blocks as shown in the following illustration.



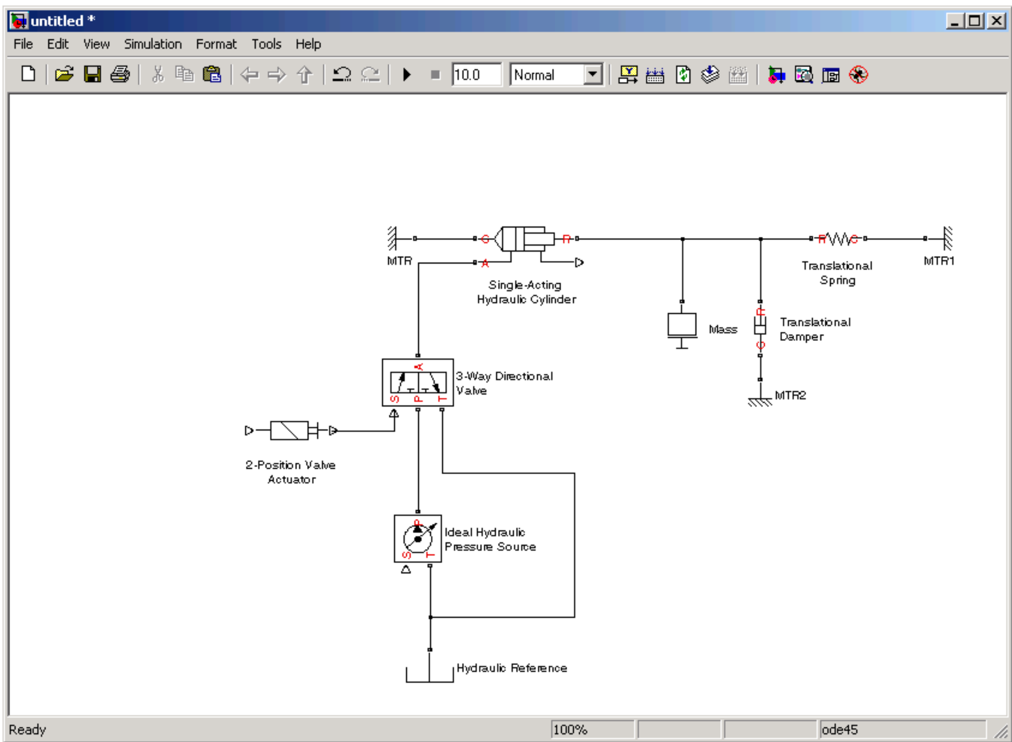
7 Ports T of the Ideal Hydraulic Pressure Source and 3-Way Directional Valve blocks have to be connected to the tank, at atmospheric pressure. To model this connection, open the Physical Networks > Foundation Library > Hydraulic > Hydraulic Elements library and add the Hydraulic Reference block to your diagram, as shown below. To do this, connect the only port of the Hydraulic Reference block to port T of the Ideal Hydraulic Pressure Source block, then right-click this connection line to create a branching point, and connect this point to port T of the 3-Way Directional Valve block.



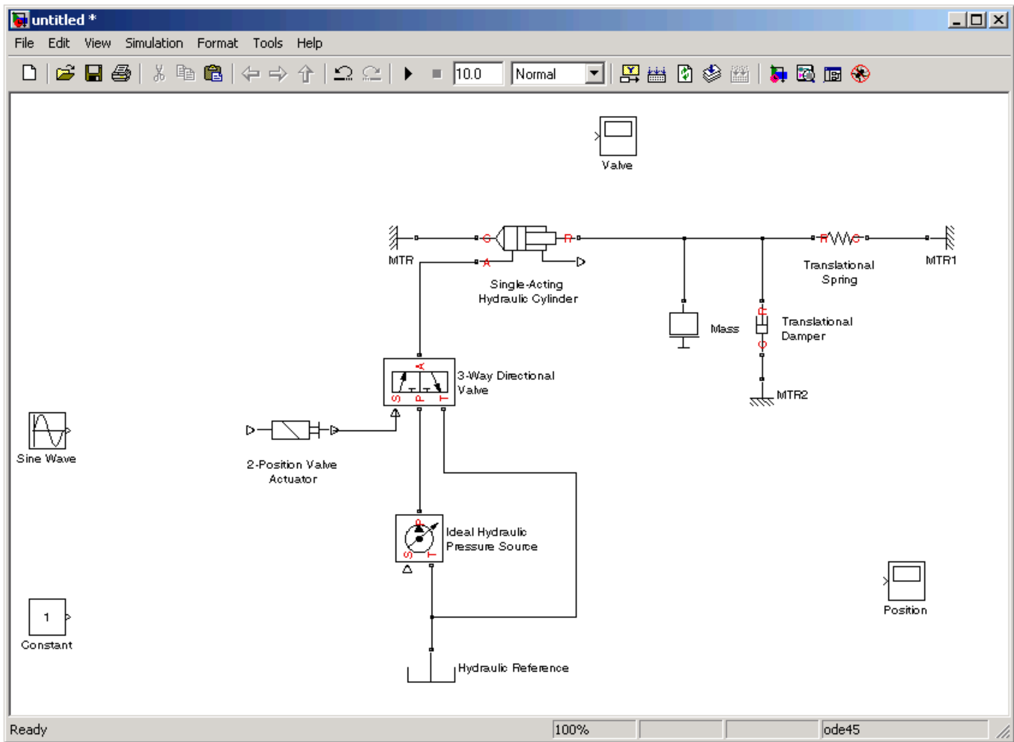
8 Model the mechanical load for the cylinder. Open the Physical Networks > Foundation Library > Mechanical > Translational Elements library and add the Mass, Translational Spring, Translational Damper, and three Mechanical Translational Reference blocks to your diagram.

To indicate that the cylinder case is fixed, connect port C of the Single-Acting Hydraulic Cylinder block to one of the Mechanical Translational Reference blocks. To rotate the Mechanical Translational Reference block, select it and press **Ctrl+R**. You can also shorten the block name to MTR to make the diagram easier to read.

Connect the other blocks to port R of the Single-Acting Hydraulic Cylinder block, as shown below.

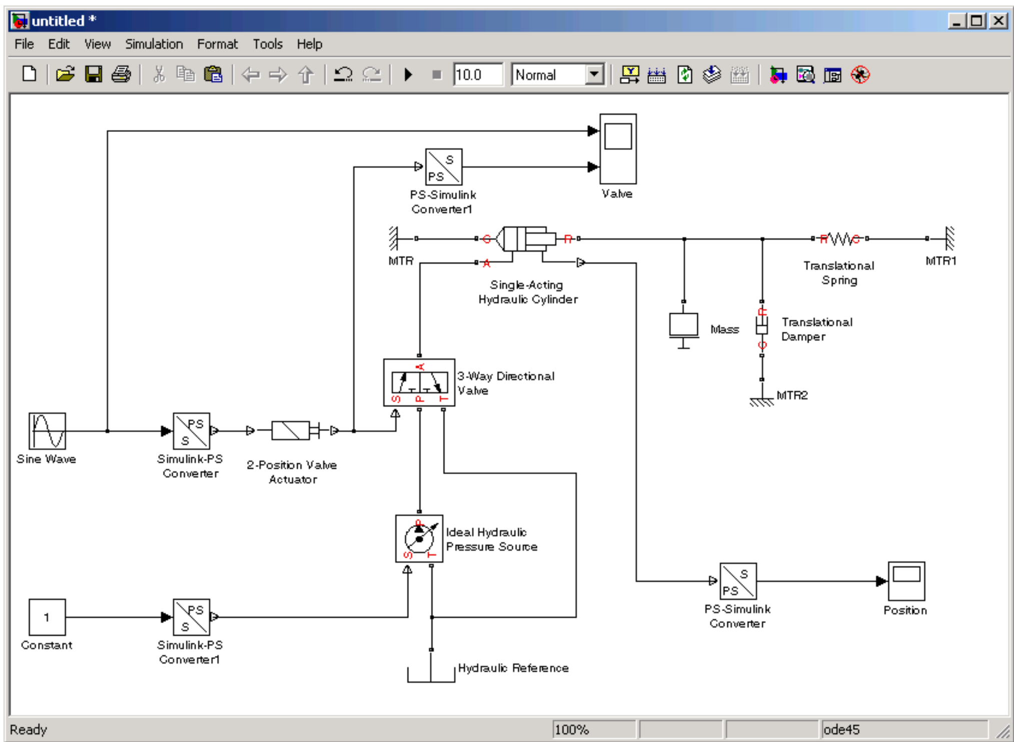


9 Now you need to add the sources and scopes. They are found in the regular Simulink libraries. Open the Simulink > Sources library and copy the Constant block and the Sine Wave block into the model. Then open the Simulink > Sinks library and copy two Scope blocks. Rename one of the Scope blocks to `Valve`. It will monitor the valve opening based on the input signal variation. The other Scope block will monitor the position of the cylinder rod; rename it to `Position`.



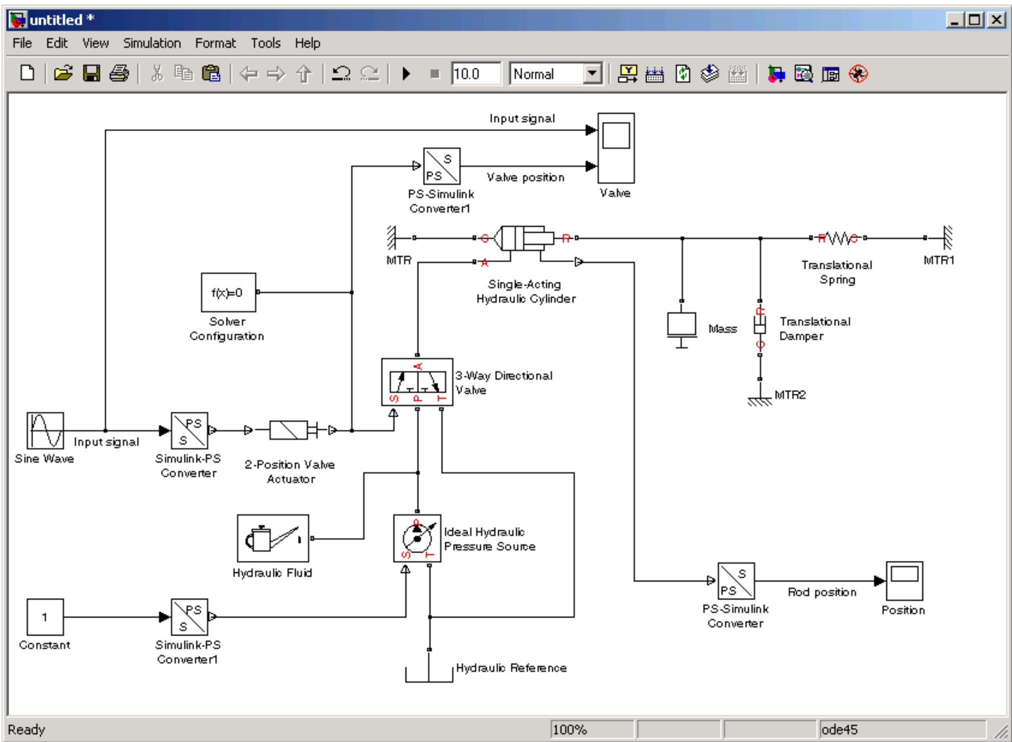
10 Double-click the `Valve` scope to open it. In the scope window, click  to access the scope parameters, change **Number of axes** to `2`, and click **OK**. The scope window now displays two sets of axes, and the `Valve` scope in the diagram has two input ports.

11 Every time you connect a Simulink source or scope to a SimHydraulics diagram, you have to use an appropriate converter block, to convert Simulink signals into physical signals and vice versa. Open the Physical Networks > Utilities library and copy two Simulink-PS Converter blocks and two PS-Simulink Converter blocks into the model. Connect the blocks as shown below.



12 To specify the fluid properties, add the Hydraulic Fluid block, found in the Physical Networks > SimHydraulics > Hydraulic Utilities library, to your diagram. You can add this block anywhere on the hydraulic circuit by creating a branching point and connecting it to the only port of the Hydraulic Fluid block.

13 Each topologically distinct physical network in a diagram requires exactly one Solver Configuration block, found in the Physical Networks > Utilities library. Copy this block into your model and connect it to the circuit, similar to the Hydraulic Fluid block. Your diagram now should look like this.



14 Your block diagram is now complete. Save it as `simple.mdl`.

Modifying Initial Settings

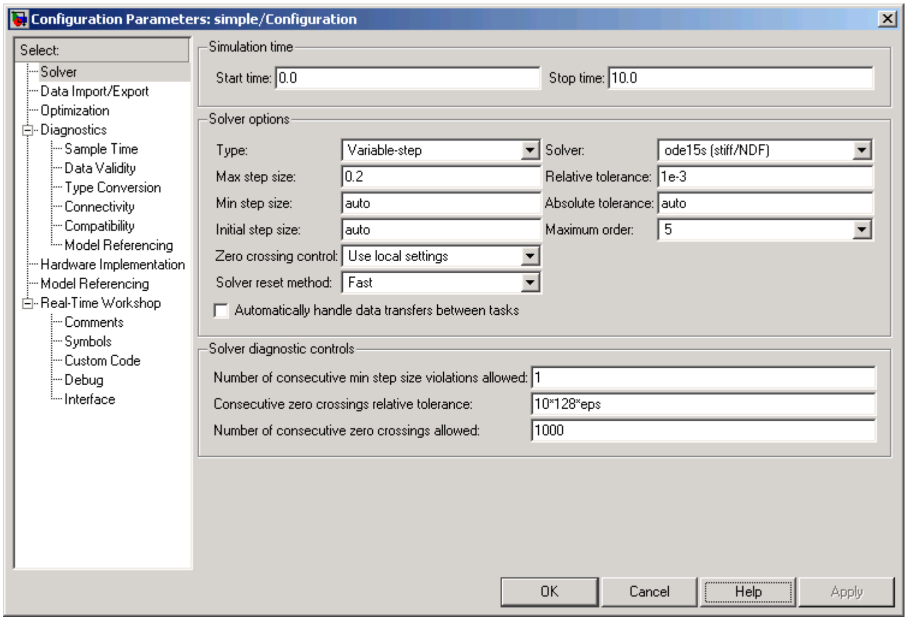
After you have put together a block diagram of your model, as described in the previous section, you need to select a solver and provide the correct values for block parameters. All the blocks have default parameter values that allow them to run "out of the box," but you may need to change some of them to suit your particular application.

To prepare for simulating the model, follow these steps:

1 Select a Simulink solver. On the top menu bar of the model window, select **Simulation > Configuration Parameters**. The Configuration Parameters dialog box opens, showing the **Solver** node.

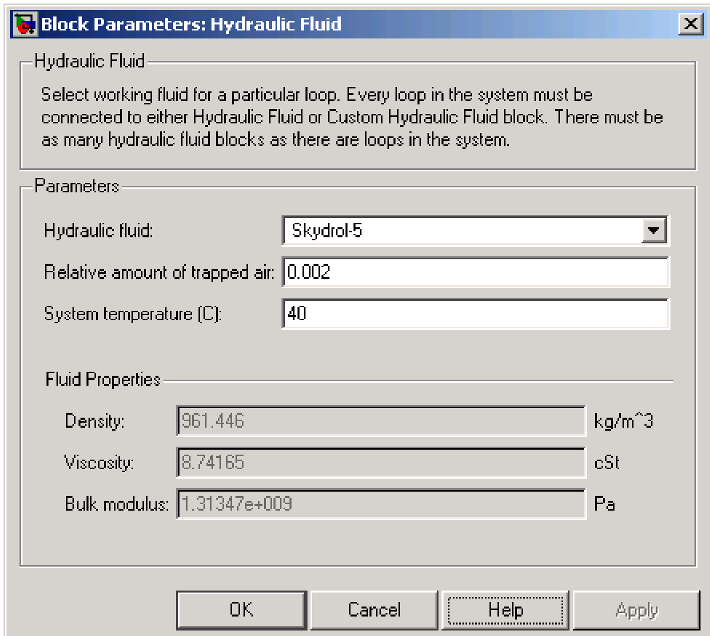
   Under **Solver options**, set **Solver** to `ode15s (Stiff/NDF)` and **Max step size** to `0.2`.

   Also note that **Simulation time** is specified to be between 0 and 10 seconds. You can adjust this setting later, if needed.



   Click **OK** to close the Configuration Parameters dialog box.

2 Select a fluid. Double-click the Hydraulic Fluid block. In the Block Parameters dialog box, set **Hydraulic Fluid** to `Skydrol 5` and set the other block parameters as shown below.



Click **OK** to close the Block Parameters dialog box.

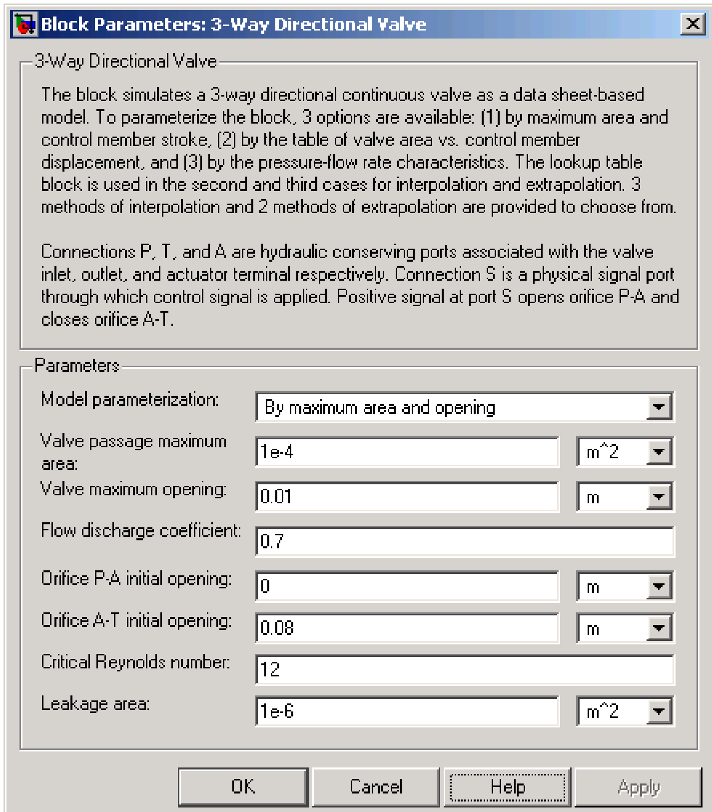
3 Specify the units for the pressure input signal. Simulink signals are unitless. When you convert them to physical signals, you can supply units by using the converter blocks. Double-click the Simulink-PS Converter1 block, enter `Pa` in the **Unit** text box, and click **OK**. When SimHydraulics parses the model, it matches the input signal units with the block input ports and provides error messages if there is a discrepancy. For more information, see “Model Validation” on page 3-3.

4 Specify a realistic value for the pressure input signal. Double-click the Constant block, enter `10e5` in the **Constant value** text box, and click **OK**.

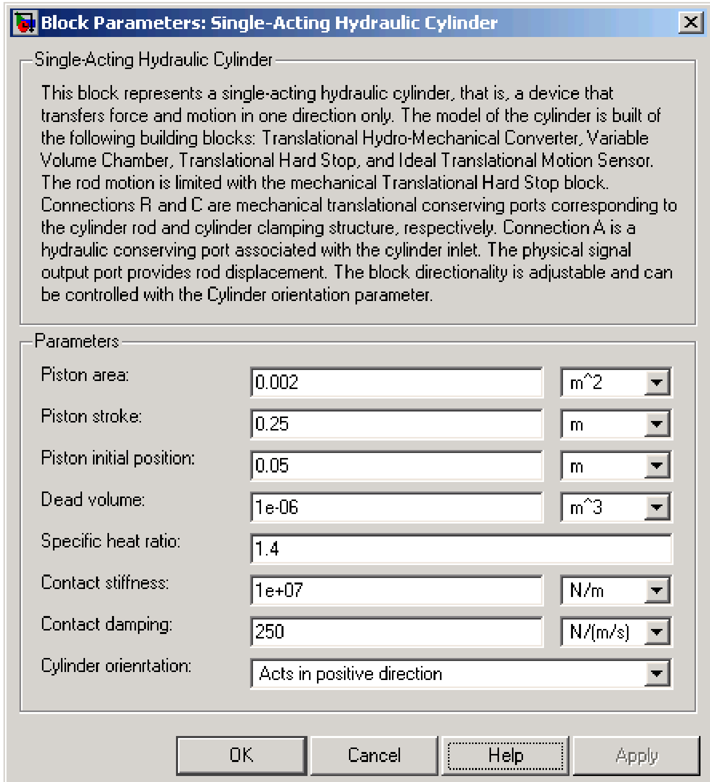
5 Open the 2-Position Valve Actuator block and note that its **Nominal Signal Value** parameter is set to `24`.

6 Double-click the Sine Wave block and change its **Amplitude** to a value greater than 50% of the nominal signal value for the 2-Position Valve Actuator block, for example, to 20.

7 Adjust the 3-Way Directional Valve block parameters as shown below.



8 Adjust the Single-Acting Hydraulic Cylinder block parameters as shown below.



9 Double-click the Mass block and change its **Mass** to `4.5 kg`.

10 Double-click the Translational Damper block, which models the viscous friction, and change its **Damping coefficient** to `250 s*N/m`.

11 Double-click the Translational Spring block. Set its **Spring rate** to `6e3 N/m` and **Initial deformation** to `0.02 m`.

12 Save the model.

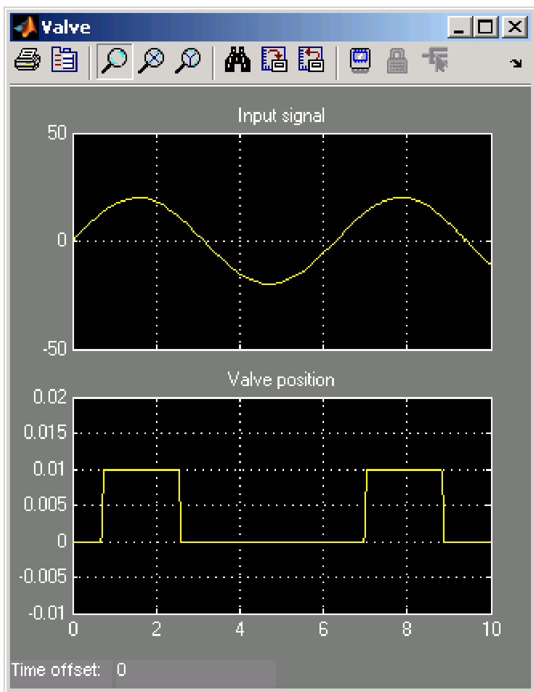
Running the Simulation

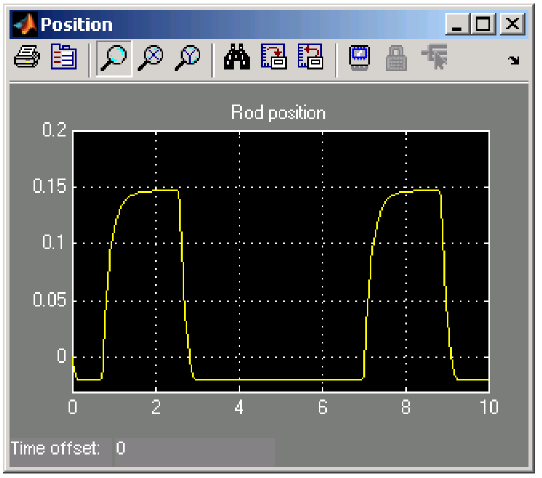
After you've put together a block diagram and specified the initial settings for your model, you can run the simulation.

1 The input signal for the valve opening is provided by the Sine Wave block. The Valve scope reflects both the input signal and the valve opening as functions of time. The Position scope outputs the cylinder rod displacement as a function of time. Double-click both scopes to open them.

2 To run the simulation, click  in the model window toolbar. SimHydraulics solver evaluates the model, calculates the initial conditions, and runs the simulation. For a detailed description of this process, see “How SimHydraulics Works” on page 3-2. Completion of this step may take a few seconds. The message in the bottom-left corner of the model window provides the status update.

3 Once the simulation starts running, the Valve and Position scope windows display the simulation results, as shown in the next illustration.





In the beginning, the valve is closed. Then, as the input signal reaches 50% of the actuator's nominal signal, the valve gradually opens to its maximum value and moves the cylinder rod in the positive direction. When the input signal goes below 50% of the nominal signal, the actuator closes the valve. The spring returns the cylinder rod to its initial position.

You can now adjust various inputs and block parameters and see their effect on the valve opening profile and the cylinder rod displacement.

Adjusting the Parameters

After running the initial simulation, you can experiment with adjusting various inputs and block parameters.

Try the following adjustments:

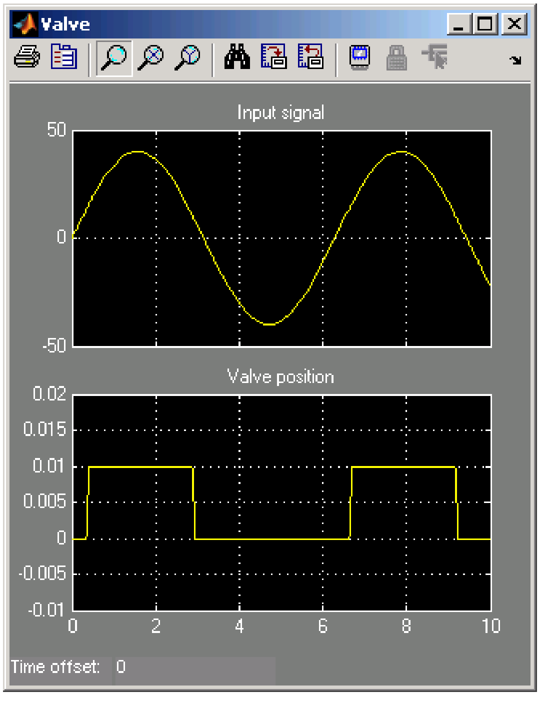
1. Change the input signal for valve opening.
2. Change the cylinder load parameters.
3. Change the rod position output units.

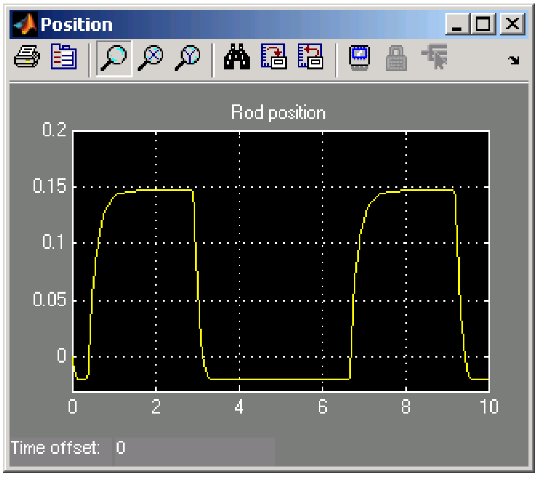
Changing the Valve Input Signal

This example shows how a change in the input signal affects the opening of the valve, and therefore the cylinder rod displacement.

1 Double-click the Sine Wave block, enter `40` in the **Amplitude** text box, and click **OK**.

2 Run the simulation. The simulation results are shown in the following illustration. With the increase in the input signal amplitude, it reaches 50% of the actuator's nominal signal sooner, and the valve stays open longer, which in turn affects the cylinder rod position.



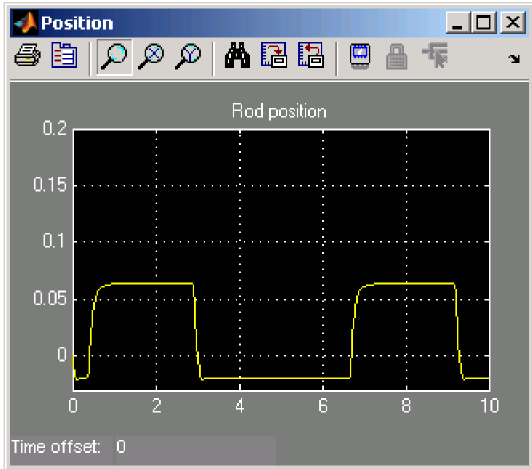


Changing the Cylinder Load Parameters

In our model, the cylinder drives a load consisting of a mass, viscous friction, and preloaded spring. This example shows how a change in the spring stiffness affects the cylinder rod displacement.

1 Double-click the Translational Spring block. Set its **Spring rate** to `12e3 N/m`.

2 Run the simulation. The valve opening profile is not affected, but increase in spring stiffness results in smaller amplitude of cylinder rod displacement, as shown in the following illustration.

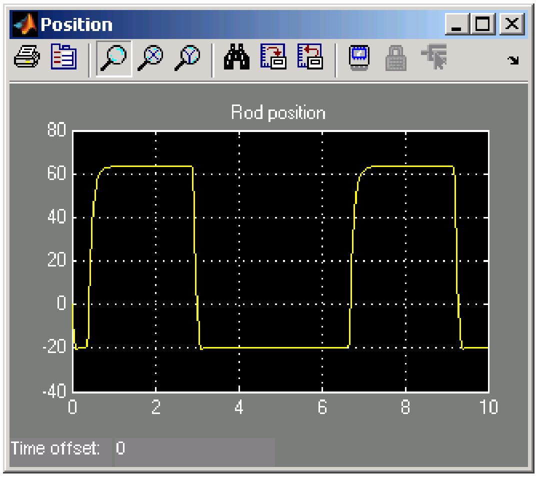


Changing the Rod Position Output Units

In our model, we have used the PS-Simulink Converter block in its default parameter configuration, which does not specify units. Therefore, the `Position` scope outputs the cylinder rod displacement in the units specified for the parameters of the Single-Acting Hydraulic Cylinder block; in this case, in meters. This example shows how to change the output units for the cylinder rod displacement to millimeters.

1. Double-click the PS-Simulink Converter block. Type `mm` in the **Unit** text box and click **OK**.

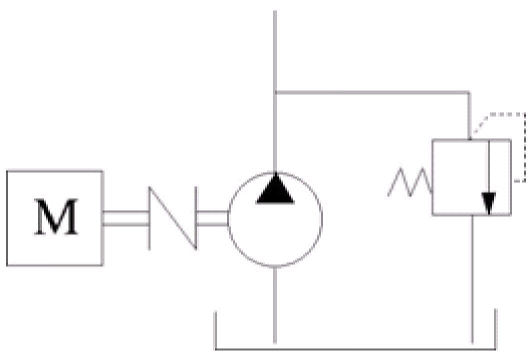
2. Run the simulation. In the `Position` scope window, click  to autoscale the scope axes. The cylinder rod displacement is now output in millimeters, as shown in the following illustration.


Position
Rod position
80
60
40
20
0
-20
-40
0
2
4
6
8
10
Time offset: 0

Modeling Power Units

The power unit is perhaps the most prevalent unit in hydraulic systems. Its main function is to supply the required amount of fluid under specified pressure. There is a huge variety of power unit designs varying by the amount and type of pumps, prime movers, valves, tanks, etc. The set of blocks available with SimHydraulics allows you to simulate practically any of these configurations. This section considers basic approaches in simulating power units and examples of typical schematics.

A typical power unit of a hydraulic system, as shown in the following illustration, consists of a fixed-displacement or variable-displacement pump, reservoir, pressure-relief valve, and a prime mover that drives the hydraulic pump.



Typical Hydraulic Power Unit

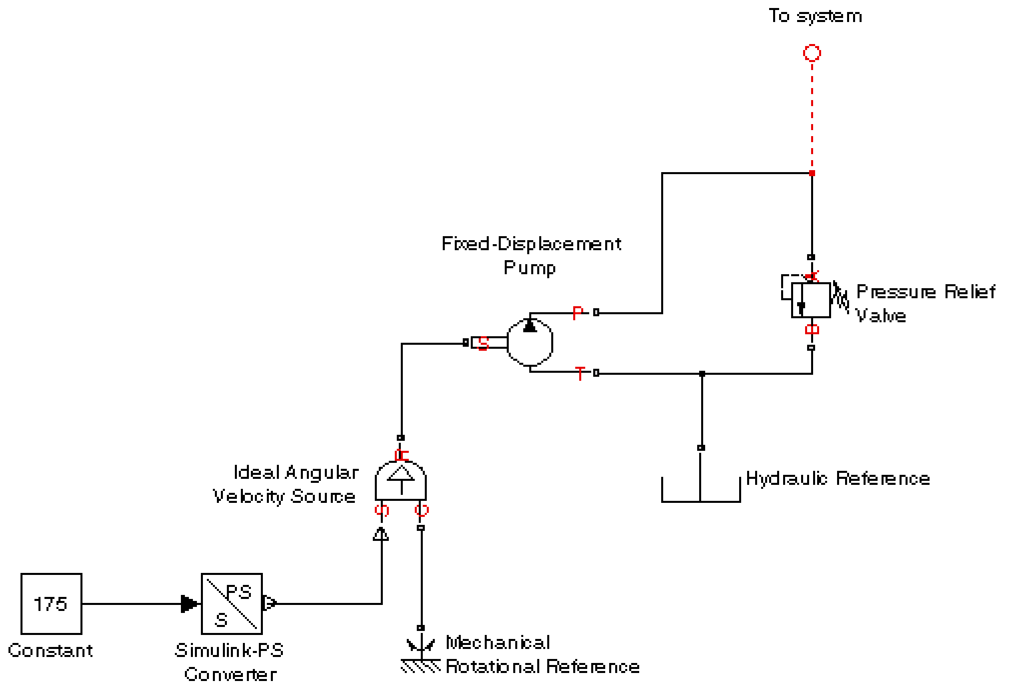
In developing a model of a power unit, you must reach a compromise between the robustness, speed of simulation, and accuracy, meaning that the model should be as simple as possible to provide acceptable accuracy within working range of variable parameters.

The first option is to simulate a power unit literally, as it is, reproducing all its components. This approach is illustrated in the Power Unit with Fixed-Displacement Pump demo (`sh_power_unit_fxd_dspl_pump`). The power unit consists of a fixed-displacement pump, which is driven by a motor through a compliant transmission, a pressure-relief valve, and a variable orifice, which simulates system fluid consumption. The motor model is represented as a source of angular velocity rotating shaft at 188 rad/s at zero torque. The load on the shaft decreases the velocity with a slip coefficient of 1.2 (rad/s)/Nm. The load on the driving shaft is measured with the torque sensor. The shaft between the motor and the pump is assumed to be compliant and simulated with rotational spring and damper.

The simulation starts with the variable orifice opened, which results in a low system pressure and the maximum flow rate going to the system. The orifice starts closing at 0.5 s, and is closed completely at 3 s. The output pressure builds up until it reaches the pressure setting of the relief valve (75e5 Pa) and is maintained at this level by the valve. At 3 s, the variable orifice starts opening, thus returning system to its initial state.

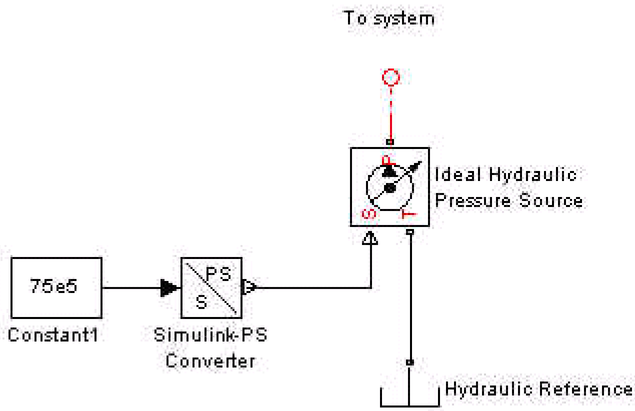
You can implement a considerably more complex model of a prime mover by following the pattern used in the demo. For instance, the shaft can be represented with multiple segments and intermediate bearings. The model of a prime mover can be more comprehensive, accounting for its type (DC or AC electric motor, diesel or gasoline engine), characteristics, control type, and so on. In addition, a complex mechanical transmission driven by a diesel or gasoline internal combustion engine modeled in SimDriveline can be combined with the SimHydraulics model of hydraulic portion of a power unit.

Depending on your particular application, you may be able to simplify the model of a power unit practically without a loss in accuracy. The main factors to be considered in this process are the driving shaft angular velocity variation magnitude and the system pressure variation range. If the prime mover angular velocity remains practically constant during simulated time or varies insignificantly with respect to its steady-state value, the entire driving shaft subsystem can be replaced with the Ideal Angular Velocity Source block, whose output is set to the steady-state value, as it is shown in the following illustration.



Using the Ideal Angular Velocity Source Block in Modeling Power Units

Furthermore, if a pump delivery exceeds system fluid requirement all the time, the pump output pressure remains practically constant and close to the pressure setting of the pressure-relief valve. If this assumption is true and acceptable, the entire power unit can be reduced to an Ideal Pressure Source block, as shown in the next illustration.



Using the Ideal Pressure Source Block in Modeling Power Units

The two previous examples demonstrate that the use of ideal sources is a powerful means of reducing the complexity of models. However, you should exercise extreme caution every time you use an ideal source instead of a real pump. The substitution is possible only if there is an assurance that the controlled parameter (angular velocity in the first example, and pressure in the second example) remains constant. If this is not the case, the power unit represented with an ideal source will generate considerably more power than its simulated physical counterpart, thus making the simulation results incorrect.

Modeling Directional Valves

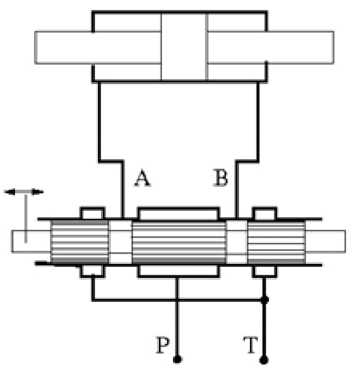
This section describes the models of directional valves available in SimHydraulics, as well as the rules of building a custom model of a directional valve and the building blocks available for this purpose. It covers the following topics:

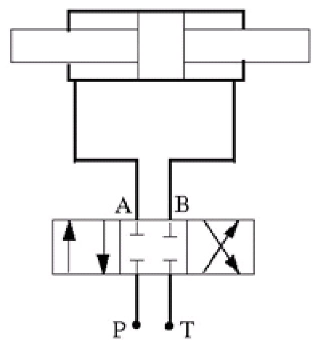
Types of Directional Valves

The main function of directional valves in hydraulic systems is to direct and distribute flow between consumers. As far as the valve modeling is concerned, the valves are classified by the following main characteristics:

- Number of external paths (connecting ports) — One-way, two-way, three-way, four-way, multiple-way
- Number of positions a control member of the valve can assume — Two-position, three-position, multiple-position, continuous (can assume any position within working range)
- Control member type — Spool, poppet, sliding flat spool, and so on

As an example, the following illustration shows a portion of a hydraulic system with a 4-way, 3-position directional valve controlling a double-acting cylinder, next to its schematic diagram.





Throughout SimHydraulics libraries, hydraulic ports are identified with the following symbols:

- P — Pressure port
- T — Return (tank) port
- A, B — Actuator ports
- X, Y — Pilot or control ports

Available Blocks and How to Use Them

SimHydraulics offers five models of directional valves:

- 2-Way Directional Valve
- 3-Way Directional Valve
- 4-Way Directional Valve
- Check Valve
- Pilot-Operated Check Valve

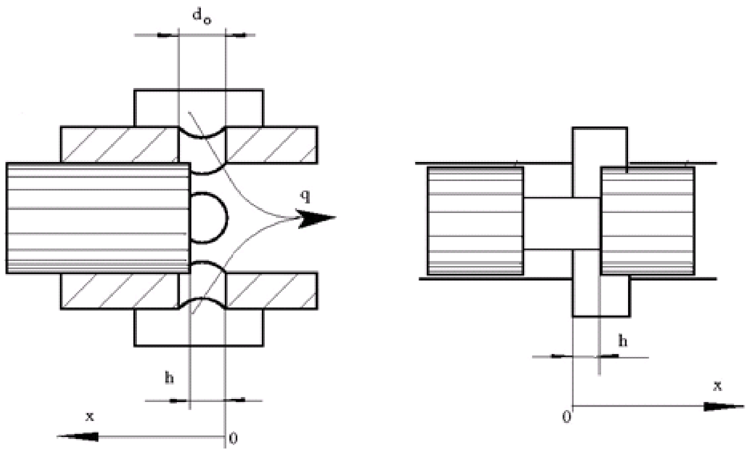
As indicated in their descriptions, all of them are symmetrical, continuous valves. In other words, the control member in 2-way, 3-way, and 4-way valves can assume any position, controlled by the physical signal port S. The valves are symmetrical in that all the orifices the valve is built of are of the same type and size. The only possible difference between orifices is the orifice initial opening.

These configurations cover a substantial portion of real valves, but the directional valves family is so diverse as to make it practically impossible to have a library model for every member. Instead, SimHydraulics offers a set of building blocks that is comprehensive enough to build a model for any real configuration. This section describes the rules of building a custom model of a directional valve.

All directional valve models are built of variable orifices. In SimHydraulics, the following variable orifice models are available:

- Orifice with Variable Area Round Holes
- Orifice with Variable Area Slot
- Variable Orifice
- Ball Valve
- Needle Valve
- Poppet Valve

To simplify the way variable orifices are combined in a model, their instantaneous opening is computed in the same way for all types of orifices. The orifice opening is always computed in the direction the spool, or any other control member, opens the orifice. In other words, positive value of the opening corresponds to open orifice, while negative value denotes overlapped, or closed, orifice. The origin always corresponds to zero-lap position, when the edge of the control member coincides with the edge of the orifice. In the illustration below, origins are marked with `0` for the orifice with variable area round holes (schematic on the left) and for the orifice with variable slot (schematic on the right). The `x` arrow denotes the direction in which orifice opening is measured in both cases.



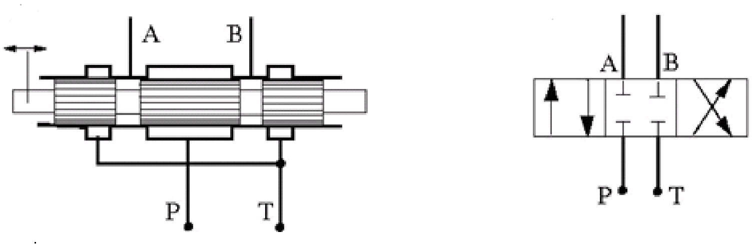
The instantaneous value of the orifice opening is determined as

$$h = x_0 + x_{sp} \cdot or$$

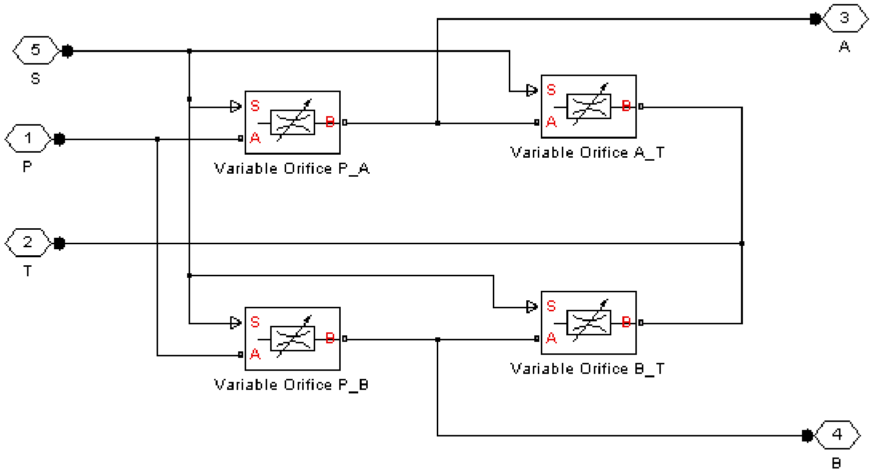
where

h	Instantaneous orifice opening.
x_0	Initial opening. The initial opening value is positive for initially open (underlapped) orifices and negative for overlapped orifices.
x_{sp}	Spool (or other control member) displacement from initial position, which controls the orifice.
or	Orifice orientation indicator. The variable assumes +1 value if a spool displacement in the globally assigned positive direction opens the orifice, and -1 if positive motion decreases the opening.

The number of variable orifices and the way they are connected are determined by the valve design. Usually, the model of a valve mimics the physical layout of a real valve. The illustration below shows an example of a 4-way valve, its symbol, and an equivalent circuit of its SimHydraulics model.

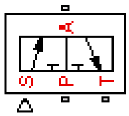


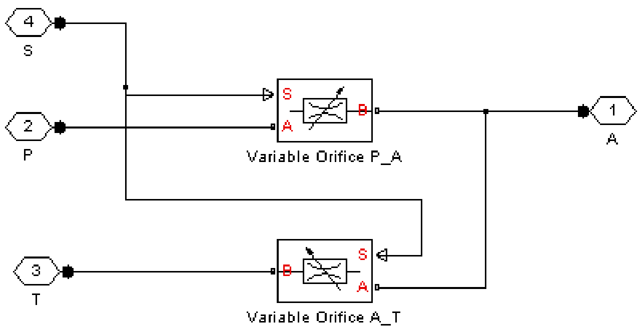
Modeling a 4-Way Directional Valve



The 4-way valve in its simplest form is built of four variable orifices. In the equivalent circuit, they are named P_A, P_B, A_T, and B_T. The Variable Orifice block, which is the most generic model of a variable orifice in SimHydraulics, is used in this particular example. You can use any other variable orifice blocks if the real valve design employs a configuration backed by a stock model, such as an orifice with round holes or rectangular slots, poppet, ball, or needle. In general, all orifices in the model can be simulated with different blocks or with the same block, but with different way of parameterization. For instance, two orifices can be represented by their pressure-flow characteristics, while two others can be simulated with the table-specified area variation option (for details, see the Variable Orifice block reference page).

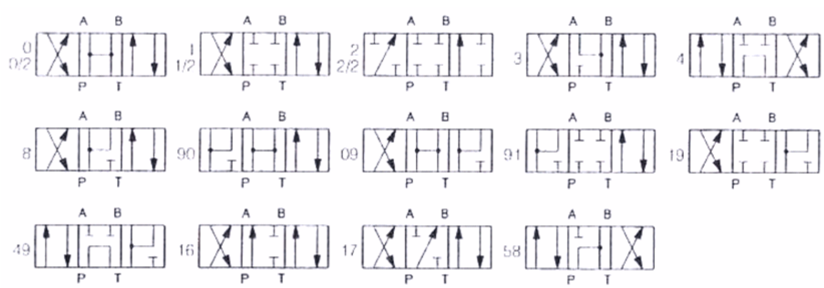
The next example shows the SimHydraulics symbol and the equivalent circuit and of the 3-Way Directional Valve block, the model of which is built of two variable orifices representing P_A and A_T paths.



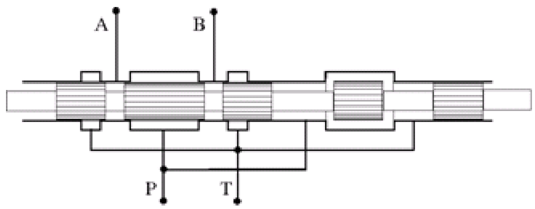


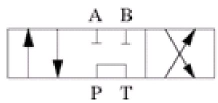
Possible Configurations of 4-Way Directional Valves

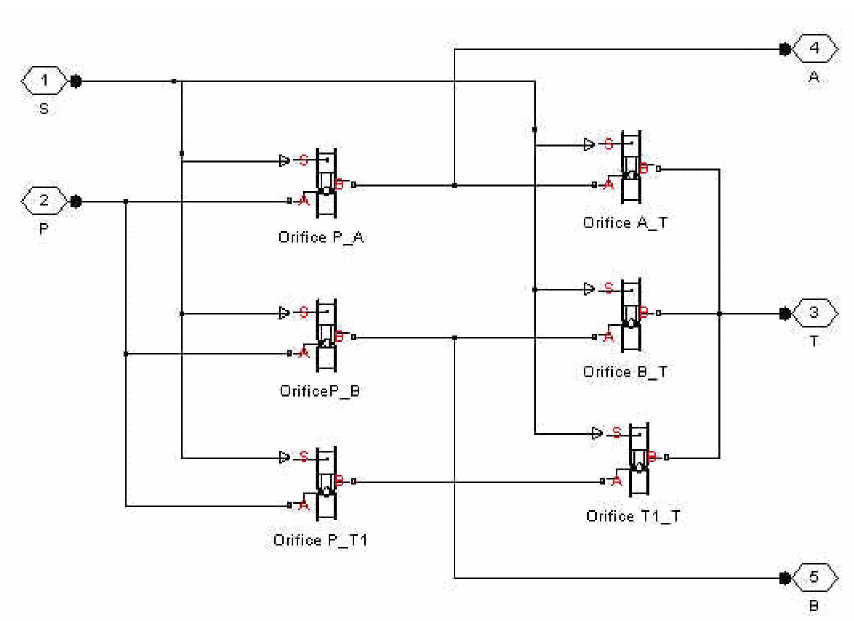
Directional valves also vary by the pattern their ports are connected in fixed or intermediate positions. The following illustration shows the examples of these patterns for 4-way valves.



Most of these patterns can be implemented by selecting initial orifice openings in the four-orifice valve configuration shown in the previous section, in the figure named Modeling a 4-Way Directional Valve on page 2-34 (for example, patterns 0, 1, and 3). However, some patterns require a model with six or more variable orifices. For instance, pattern 4 can be implemented with the following configuration, which requires six variable orifice blocks.



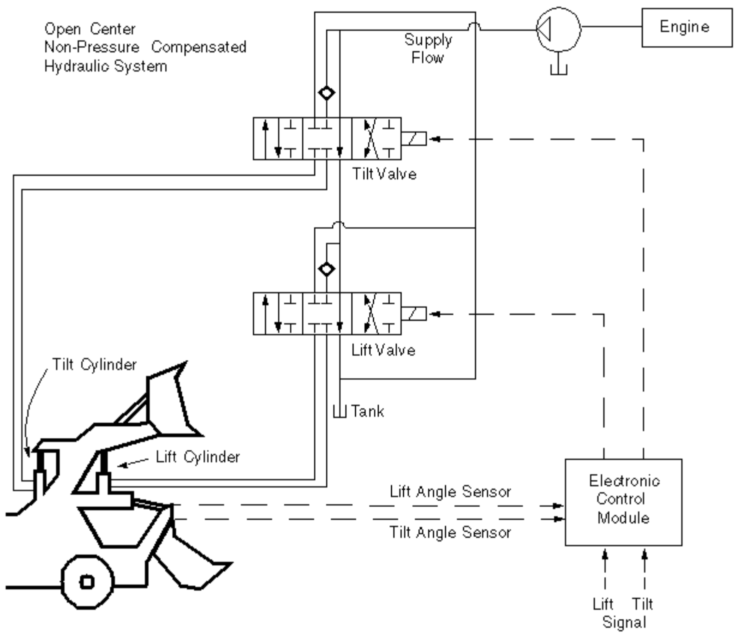




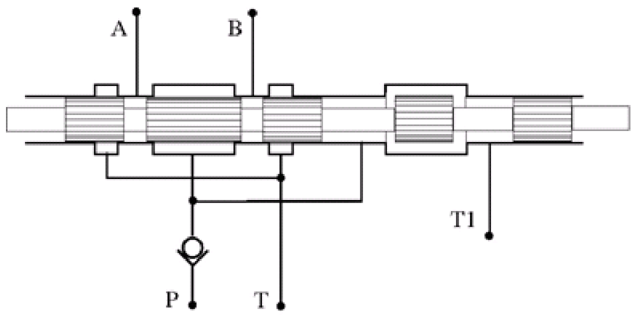
The Orifice with Variable Area Round Holes blocks have been used as a variable orifice in this model. Port T1 corresponds to an intermediate point between ports P and T.

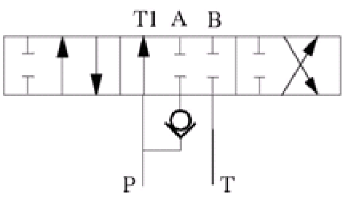
Example of Building a Custom Directional Valve

Finally, let us consider a more complex directional valve example. The figure below shows basic elements of a front loader hydraulic system. Both the lift and the tilt cylinders are controlled by custom 3-position, 5-way valves, developed for this particular application. The valves are designed in such a way that the pump delivery is diverted to tank (unloaded) if both cylinders are commanded to be in neutral position. The pump is disconnected from the tank if either of the two control valves is shifted from neutral position.

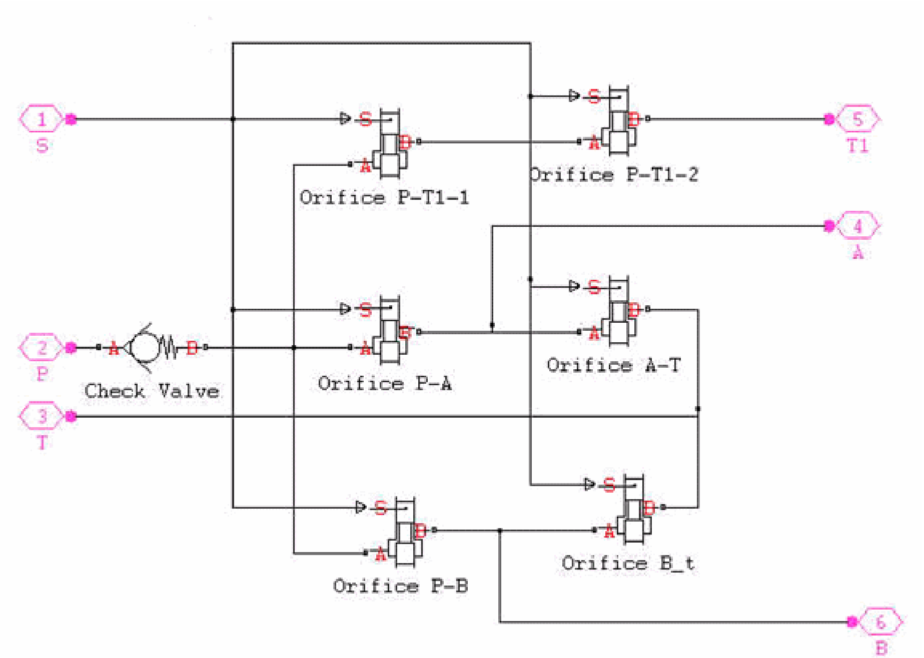


To develop a model, the physical version of the valve must be created first. The following illustration shows one of the possible configurations of the valve.





The SimHydraulics model, shown below, is an exact copy of the physical valve configuration.



All the orifices in the model are closed (overlapped) in valve neutral position, except orifices P_T1_1 and P_T1_2. These two orifices should be set open to an extent that allows pump delivery to be discharged at low pressure.

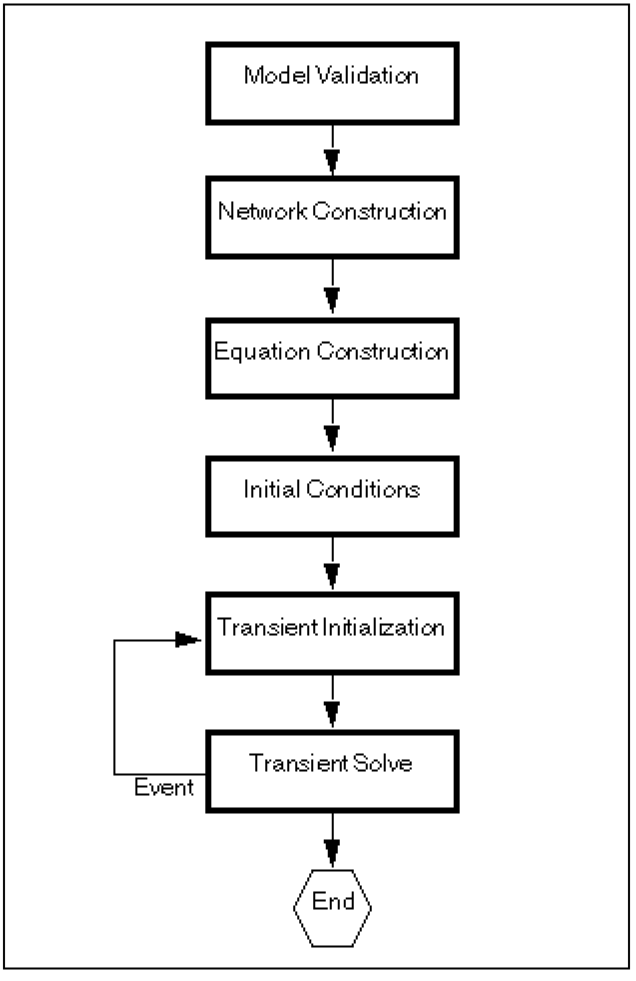
Running Hydraulic Models

SimHydraulics gives you multiple ways to simulate and analyze hydraulic power and control systems in the Simulink environment. Running a hydraulic simulation is similar to running a simulation of any other Simulink model. It entails setting various simulation options, starting the simulation, and viewing the simulation results. See the Using Simulink documentation for a general discussion of these topics. This chapter focuses on aspects of simulation specific to SimHydraulics models.

How SimHydraulics Works

You might find this brief overview of how SimHydraulics works helpful for constructing models and understanding errors.

Hydraulic simulation sequence is shown in the following flow chart.



Hydraulic simulation consists of the following major phases:

1 "Model Validation" on page 3-3

2 "Network Construction" on page 3-3

3 "Equation Construction" on page 3-4

4 "Computing Initial Conditions" on page 3-4

5 "Performing Transient Initialization" on page 3-5

6 "Transient Solve" on page 3-5

Model Validation

SimHydraulics first validates the model configuration and checks your data entries from the block dialogs. In particular:

- Each topologically distinct physical network in a diagram requires exactly one Solver Configuration block.
- Each topologically distinct hydraulic circuit in a diagram requires exactly one Hydraulic Fluid block or Custom Hydraulic Fluid block to be connected to it. Therefore, there must be as many Hydraulic Fluid blocks (or Custom Hydraulic Fluid blocks) as there are hydraulic circuits in the system. These blocks define the fluid properties that act as global parameters for all the blocks connected to the hydraulic circuit.
- Signal units specified in a Simulink-PS Converter block must match the input type expected by the SimHydraulics block connected to it. For example, when you provide the input signal for an Ideal Hydraulic Pressure Source block, specify pressure units, such as `Pa` or `bar`, in the Simulink-PS Converter block, or leave it unitless. Similarly, units specified in a PS-Simulink Converter block must match the type of physical signal provided by the SimHydraulics block outport.

Network Construction

After validating the model, SimHydraulics constructs the physical network based on the following principles:

- Two directly connected Conserving ports have the same values for all their Across variables (such as pressure or angular velocity).
- Any Through variable (such as flow rate or torque) transferred along the Physical connection line is divided among the multiple components connected by the branches. For each Through variable, the sum of all its values flowing into a branch point equals the sum of all its values flowing out.

Equation Construction

Based on the network configuration, the parameter values provided in the block dialogs, and the global parameters defined by the fluid properties, SimHydraulics constructs the system of equations for the model.

These equations contain variables of the following types:

- *Dynamic* — Time derivative of this variable appears in equations. Dynamic variables can be either *independent* or *dependent* (on other dynamic variables).
- *Algebraic* — Time derivative of this variable does not appear in equations. Algebraic variables are always dependent.

Computing Initial Conditions

SimHydraulics computes the initial conditions only once, in the beginning of simulation (t=0).

Initial conditions are computed by setting all dynamic variables to 0, except those corresponding to blocks that have an initial condition field in their block dialogs, and solving for all the system variables. The blocks with initial conditions have their dynamic variables set according to the user-provided value in the block dialog. For example, the Translational Spring block has the **Initial deformation** parameter, so the corresponding spring position state is set to the initial offset specified in the block dialog. Refer to the block reference documentation to find which blocks have initial conditions specified through their dialogs.

This concludes the initial conditions computation when the **Start simulation from steady state** check box in the Solver block dialog box is not selected (this is the default) .

When this box is selected, the solver attempts to find the steady state that would result if the inputs to the system were held constant for a sufficiently large time, starting from the initial state obtained from the initial conditions computation, described previously. Although the solver tries to find the particular steady state resulting from the given initial conditions, it is not guaranteed to do so. All that is guaranteed is that if the steady-state solve succeeds, the state found is a steady state (within tolerance). Steady state means that the system variables are no longer changing with time. Simulation then starts from this steady state.

Note If the simulation fails at or near the start time when you use the **Start simulation from steady state** option, consider clearing the check box and simulating with the plain initial conditions computation only.

Performing Transient Initialization

After computing the initial conditions, or after a subsequent event (such as a zero-crossing or discontinuity), SimHydraulics performs transient initialization. This is done by fixing all dynamic variables and solving for algebraic variables and derivatives of dynamic variables. The goal of transient initialization is to provide a consistent set of initial conditions for the next transient solve step. At this phase, the initial conditions at the beginning of simulation are no longer enforced, but the dynamic state of the system is preserved (or, in other words, the energy in the system is preserved).

Transient Solve

Finally, SimHydraulics performs transient solve of the system of equations. In transient solve, continuous differential equations are integrated in time to compute all the variables as a function of time.

SimHydraulics continues to perform the simulation according to the results of the transient solve until it encounters an event, such as a zero crossing or discontinuity. The event may be within the physical network or elsewhere in the Simulink model. If an event is encountered, SimHydraulics returns to the phase of transient initialization, and then back to transient solve. This cycle continues until the end of simulation.

Linearizing SimHydraulics Models

The Simulink `linmod` and `dlinmod` commands create continuous- or discrete-time linear time-invariant (LTI) state-space models from Simulink models. You can use these commands to generate an LTI state-space model from a model containing SimHydraulics components.

There are two basic ways `linmod` and `dlinmod` can be used, and the behavior of linearization differs depending on which method is chosen. If `linmod(mdl)` is called (that is, the arguments for time, state, and input are not provided), then consistent initial conditions are solved for in the same way as on the first step of a simulation. In particular, any initial conditions, such as initial offset from equilibrium for a spring, are set just as if the simulation were starting from the initial time.

However, if `linmod(mdl, t, x, u)` is called, it does not solve for the start-of-simulation initial conditions. An important difference of SimHydraulics from Simulink is that the states in SimHydraulics are not generally independent (some states have algebraic dependencies on other states). Thus, it is possible, though erroneous, to provide `linmod` with an inconsistent state to linearize about. If the state is not consistent (within a suitably chosen tolerance based on the solver tolerance), then SimHydraulics issues a warning at the command line when linearization is attempted. SimHydraulics will then attempt to make the provided `x` consistent by changing some of its components, possibly by large amounts. Thus it is important, when providing a state to `linmod` and `dlinmod`, to ensure that the state provided is reasonably consistent. This is most easily done by taking the state from some time in a simulation run, for example, by using the **States** check box on the **Data Import/Export** pane of the Configuration Parameters dialog box, accessed by selecting **Simulation > Configuration Parameters** from the top menu bar.

Note If the **Enable automatic function scaling** check box in the Solver block dialog box is selected (or cleared) when you obtain a state through simulation to provide it to `linmod` or `dlinmod`, this check box must remain selected (or cleared) when `linmod` or `dlinmod` is called.

The technique used to obtain linear models involves obtaining an independent subset of the system states as the states for linearization. These independent state variables are a subset of the dynamic variables of the system. Thus the A matrix, of size n_states by n_states, is all zeros except for a submatrix of size n_ind by n_ind, where n_ind is the number of independent states. The B matrix, of size n_states by n_inputs, is all zeros except for a submatrix of size n_ind by n_inputs. The C matrix, of size n_outputs by n_states, is all zeros except for a submatrix of size n_outputs by n_ind. The D matrix, of size n_outputs by n_inputs, may have nonzeros everywhere. To obtain a model where the number of states equals the number of independent states, use the function sminreal in the Control System Toolbox.

Limitations

Certain Simulink tools and features are not supported in SimHydraulics.

Solver Support

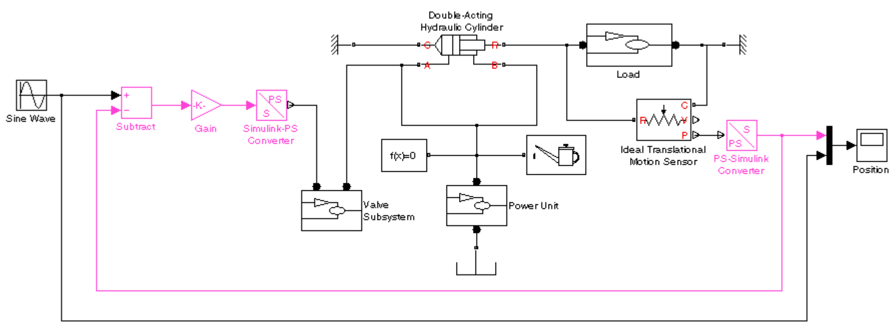
Explicit solvers are not currently supported in SimHydraulics. Only the following solvers are supported:

- `ode15s`
- `ode23t`
- `ode14x`

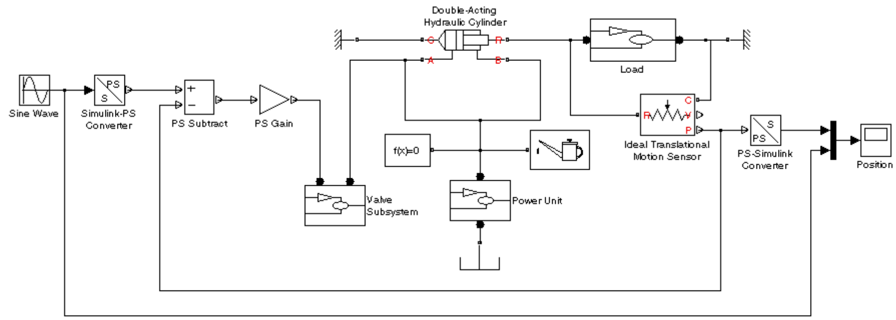
Algebraic Loops

A SimHydraulics physical network should not exist within a Simulink algebraic loop. This means that you should not directly connect an output of a PS-Simulink Converter block to an input of a Simulink-PS Converter block of the same physical network.

For example, the following model contains a direct feedthrough between the PS-Simulink Converter block and the Simulink-PS Converter block (highlighted in magenta). To avoid the algebraic loop, you can insert a Transfer Function block anywhere along the highlighted loop.



A better way to avoid an algebraic loop without introducing additional dynamics is shown in the modified model below.



Code Generation

Code generation is not currently supported in SimHydraulics.

Glossary

across variables
Variables that are measured with a gauge connected in parallel to an element.

behavioral block implementation model
A block that is implemented based on its physical behavior, described by a system of mathematical equations. An example of a behavioral block implementation is Variable Orifice block.

conserving ports
Bidirectional hydraulic or mechanical ports that represent physical connections and relate physical variables based on the Physical Network approach.

constructional block implementation model
A block that is constructed out of other blocks, connected in a certain way. An example of a constructional block implementation is the 4-Way Directional Valve block, which is constructed based on four Variable Orifice blocks.

data-sheet-based model
A block with a set of parameters determined by data that is usually listed in the manufacturer's catalogs or data sheets.

globally assigned positive direction
Direction considered positive for a model diagram.

physical signal ports
Unidirectional ports (inports and outports) transferring signals that use an internal SimHydraulics engine for computations.

through variables
Variables that are measured with a gauge connected in series to an element.

Examples

Use this list to find examples in the documentation.

Getting Started

"Creating a Simple Model" on page 2-8

Modeling Directional Valves

"Example of Building a Custom Directional Valve" on page 2-37

Index