Simscape™ Fluids™

User's Guide

MATLAB®&SIMULINK®

R2023a

MathWorks®

How to Contact MathWorks

Latest news: `www.mathworks.com`

Sales and services: `www.mathworks.com/sales_and_services`

User community: `www.mathworks.com/matlabcentral`

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

Phone: 508-647-7000

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

Simscape™ Fluids™ User's Guide

Trademarks

Patents

Revision History

March 2006	Online only	New for Version 1.0 (Release 2006a+)
September 2006	Online only	Revised for Version 1.1 (Release 2006b)
March 2007	Online only	Revised for Version 1.2 (Release 2007a)
September 2007	Online only	Revised for Version 1.2.1 (Release 2007b)
March 2008	Online only	Revised for Version 1.3 (Release 2008a)
October 2008	Online only	Revised for Version 1.4 (Release 2008b)
March 2009	Online only	Revised for Version 1.5 (Release 2009a)
September 2009	Online only	Revised for Version 1.6 (Release 2009b)
March 2010	Online only	Revised for Version 1.7 (Release 2010a)
September 2010	Online only	Revised for Version 1.8 (Release 2010b)
April 2011	Online only	Revised for Version 1.9 (Release 2011a)
September 2011	Online only	Revised for Version 1.10 (Release 2011b)
March 2012	Online only	Revised for Version 1.10.1 (Release 2012a)
September 2012	Online only	Revised for Version 1.11 (Release 2012b)
March 2013	Online only	Revised for Version 1.12 (Release 2013a)
September 2013	Online only	Revised for Version 1.13 (Release 2013b)
March 2014	Online only	Revised for Version 1.14 (Release 2014a)
October 2014	Online only	Revised for Version 1.15 (Release 2014b)
March 2015	Online only	Revised for Version 1.16 (Release 2015a)
September 2015	Online only	Revised for Version 1.17 (Release 2015b)
March 2016	Online only	Revised for Version 2.0 (Release 2016a) (Renamed from *SimHydraulics® User's Guide*)
September 2016	Online only	Revised for Version 2.1 (Release R2016b)
March 2017	Online only	Revised for Version 2.2 (Release R2017a)
September 2017	Online only	Revised for Version 2.3 (Release R2017b)
March 2018	Online only	Revised for Version 2.4 (Release R2018a)
September 2018	Online only	Revised for Version 2.5 (Release R2018b)
March 2019	Online only	Revised for Version 2.6 (Release R2019a)
September 2019	Online only	Revised for Version 2.7 (Release R2019b)
March 2020	Online only	Revised for Version 3.0 (Release R2020a)
September 2020	Online only	Revised for Version 3.1 (Release R2020b)
March 2021	Online only	Revised for Version 3.2 (Release R2021a)
September 2021	Online only	Revised for Version 3.3 (Release R2021b)
March 2022	Online only	Revised for Version 3.4 (Release R2022a)
September 2022	Online only	Revised for Version 3.5 (Release R2022b)
March 2023	Online only	Revised for Version 3.6 (Release R2023a)

Contents

1 Get Started

2 Modeling Fluid Dynamic Systems

3 Verification and Validation

4 Model Optimization

5 How To

6 Simscape Fluids Examples

Get Started

Simscape Fluids Product Description

Model and simulate fluid systems

Simscape Fluids provides component libraries for modeling and simulating fluid systems. It includes models of hydraulic pumps, valves, actuators, pipelines, and heat exchangers. You can use these components to develop fluid power systems such as front-loader, power steering, and landing gear actuation systems. Simscape Fluids also enables you to develop engine cooling, gearbox lubrication, and fuel supply systems. You can integrate mechanical, electrical, thermal, and other physical systems into your model using components from the Simscape family of products.

Simscape Fluids helps you develop control systems and test system-level performance. You can create custom component models with the MATLAB® based Simscape language, which enables text-based authoring of physical modeling components, domains, and libraries. You can parameterize your models using MATLAB variables and expressions, and design control systems for your hydraulic system in Simulink®. To deploy models to other simulation environments, including hardware-in-the-loop (HIL) systems, Simscape Fluids supports C-code generation.

Key Features

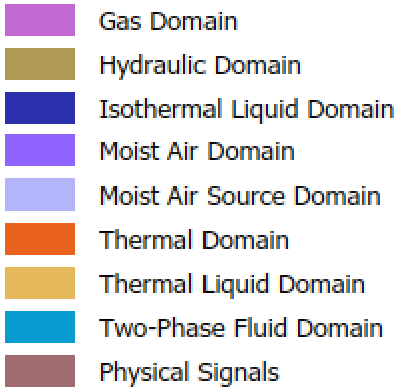
- Pump models, including centrifugal, jet, and axial-piston pumps
- Valve models, including check valves, directional valves, and pressure relief valves
- Rotational and translational actuator models, including optional friction and centrifugal forces
- Heat exchanger models for modeling heating and cooling systems
- Tank and pipe models with elevation effects for modeling fluid transportation systems
- Customizable library of common fluids, with ability to import properties from common databases
- MATLAB based Simscape language for creating custom component models
- Support for C-code generation (with Simulink Coder™)

Select the Working Fluid

The Simscape Fluids software enables you to model a variety of isothermal, thermal, and multi-phase systems. Key application areas include fluid transport, power generation, as well as heating and cooling. Simscape Fluids blocks are organized by domain:

- Isothermal Liquid: Use this domain to model adiabatic liquid systems. This library is based on mass flow rates and uniformly accounts for density as a function of pressure. This domain is best suited for models with little or no variation in temperature, or where changes in fluid viscosity do not significantly influence network dynamics, even if the network is at an elevated or reduced temperature. For more information, see:
  - “Modeling Isothermal Liquid Systems”
  - “Isothermal Liquid Modeling Options”
- Thermal Liquid: Use this domain to model liquid systems with varying temperatures and viscosities, including systems with convective or conductive heat transfer. The fluid properties are functions of pressure and temperature. For more information, see:
  - “Modeling Thermal Liquid Systems”
  - “Thermal Liquid Modeling Framework”
- Gas: Use this domain to model a perfect, semiperfect, or real gas system. This domain is best suited for applications with one predominant network gas. For more information, see:
  - “Modeling Gas Systems”
  - “Simple Gas Model”
  - “Change Flow Boundary Conditions”
  - “Change Flow Direction”
- Two-Phase Fluids: Use this domain to model systems where the working fluid may be both a liquid and a vapor. This domain does not support more than one fluid in the network. The model is based on the average fluid density in each phase; as a result, opalescence and the effect of gravity are not taken into account.
- Moist Air: Use this domain to model air-based systems with water vapor present. This domain is best suited for models where condensation or CO_2 fluctuations contribute to the dynamics of the system. For more information, see “Modeling Moist Air Systems”.
- Hydraulic (Isothermal Liquid): Use this domain to model adiabatic liquid systems. This domain is best suited for models with little or no variation in temperature, or where changes in fluid viscosity do not significantly influence network dynamics, even if the network is at an elevated or reduced temperature.

The connecting lines in each network are colored according to its domain:



Simscape Fluids Color Coding

If there is a functionality in another library that you would like to use in your model, consider using an interface block in the Fluid Network Interfaces library. These blocks allow you to connect two different networks if conditions are constant over the connection. You can also reduce the simulation time of a two-phase fluid model with an interface connection by making use of the gas and thermal liquid domains in regions where the conditions are constant and have a vapor quality of 1 or 0, respectively.

For example, if you would like to model a pipe bend in a thermal liquid network, and if the temperature does not vary in this region of the model, you can use an Interface (TL-IL) block to connect to a Pipe Bend (IL) block, and then another Interface (TL-IL) block to connect back to the thermal liquid network.

You can also model heat exchange and mass transfer within and between domains using blocks from the Fluid Network Interfaces library.

Specify Fluid Properties

A *fluid properties* block is a block that defines the fluid properties of a network. The Simscape Fluids software uses lookup tables to set these fluid properties. If you do not attach a fluid properties block, the domain defaults apply. For the default properties of the different domains, see:

- “Isothermal Liquid Domain”
- “Hydraulic Domain”
- “Thermal Liquid Domain”
- “Gas Domain”
- “Two-Phase Fluid Domain”
- “Moist Air Domain”

To set the fluid properties in an isothermal liquid network using a fluid properties block, you can either:

- Specify your own properties by connecting an Isothermal Liquid Properties (IL) block from the Simscape Foundation library. This block lets you specify whether the liquid bulk modulus is constant or pressure-dependent. You can also specify whether the fluid contains entrained air, and, if present, whether its amount is constant or pressure-dependent. The default working fluid is water, with constant bulk modulus and zero entrained air.

- Connect an Isothermal Liquid Predefined Properties (IL) block from the Simscape Fluids Isothermal Liquid library and select one of the predefined liquids. You can also visualize fluid properties as a function of pressure with this block.

To set the fluid properties in a Hydraulic (Isothermal Liquid) network, you can either:

- Specify your own properties by connecting a Custom Hydraulic Fluid block from the Simscape Foundation library.
- Connect a Hydraulic Fluid block from the Simscape Fluids Hydraulic (Isothermal Liquid) library and select one of the predefined liquids.

To set the fluid properties in a Thermal Liquid network, you can either:

- Specify your own properties by connecting a Thermal Liquid Settings (TL) block at any point in the network. This block is in the Simscape Foundation library.
- Connect a Thermal Liquid Properties (TL) block from the Simscape Fluids Thermal Liquid library and select one of the predefined liquids.

To set the properties in a Two-Phase Fluid network, you can either:

- Specify your own properties by connecting a Two-Phase Fluid Properties (2P) block from the Simscape Foundation library. This block lets you define the properties of liquid and vapor separately, each as a tabulated function of pressure and temperature.
  - You can provide tabulated data for use in the Two-Phase Fluid Properties (2P) block with the `twoPhaseFluidTables` function. This function also allows you to view the properties of the fluid. See “Manually Generate Fluid Property Tables” for more information.
- Connect a Two-Phase Fluid Predefined Properties (2P) block from the Simscape Fluids Two-Phase Liquid library and select one of the predefined liquids. You can also visualize fluid properties as a function of specific internal energy or specific enthalpy and pressure with this block.

To set the properties in a moist air network, you can specify your own properties by connecting a Moist Air Properties (MA) block from the Simscape Foundation library. You can provide tabulated data for dry air and water vapor and specify the fluid temperature and pressure ranges.

To set the fluid properties in a gas network, you can specify your own properties by connecting a Gas Properties (G) block from the Simscape Foundation library. You can use this block to model a perfect, semi-perfect or real gas.

See Also

“Simscape Block Libraries” | “Manually Generate Fluid Property Tables”

More About

- “Modeling Moist Air Systems”
- “Modeling Thermal Liquid Systems”
- “Modeling Gas Systems”
- “Modeling Isothermal Liquid Systems”

Modeling and Simulation of Fluid Networks

In this section...

Simscape Fluids blocks and connections represent one-dimensional flow paths. This means that internal component dynamics, such as the pressure variation along a pump blade, are not modeled in detail.

As with electric circuits, all network components have an associated Across and Through variable. Unlike an electric circuit, some Simscape Fluids domains have more than one Across and Through variable. For example, in the thermal liquid domain, mass flow rate and energy flow rate are block Through variables, and pressure and temperature are block Across variables. Blocks in the gas, moist air, and two-phase domains also have multiple Through and Across variables. For more information on block connections, ports, and variable types, see “Basic Principles of Modeling Physical Networks”.

In Fluids networks, flow is supported in either direction along connection lines. Many Simscape Fluids blocks also support reversed flows. Additionally, pressure and temperature are always absolute values. Regardless of flow direction, the variable values remain positive.

It is essential that your network is grounded. Simscape Fluids blocks with internal volumes, called *dynamic components*, can act as a network ground due to their internally-defined states. Furthermore, dynamic components within the network set the model initial conditions from the specified block parameters.

Due to these internal volumes, fluid variables are calculated at internal nodes as part of the momentum balance over the block. You may observe different values if you choose to measure a variable at an internal node or at a port.

Parameterizing a Block

Some blocks can be parameterized by different methods. You can parameterize these blocks by:

- An analytical formulation
- Data from a data sheet
- Data from measurements

Data tables may be 2-D or 3-D and are composed of independent vectors, such as from a test matrix. For example, in the Variable-Displacement Pump (IL) block, the **Volumetric loss table, q_loss(dp,w,D)** parameter is populated with the volumetric losses at each test point for pressure loss, shaft speed, and displacement volume, over the range of each parameter.

You can plot and check the data in a lookup table with the PS Lookup Table (1D) and PS Lookup Table (2D) blocks. See “Plot Lookup Tables” for more information.

Start and Simulate Your Model

The Simscape Fluids software provides multiple ways to simulate and analyze multidomain fluid networks in the Simulink environment.

- To open the Simscape model template, enter `ssc_new` at the MATLAB command line. You can view the Simscape Fluids block library by entering `SimscapeFluids_lib` on the MATLAB command line. Drag blocks from the Simscape Library Browser to the Simulink Editor to construct your model.
- To view and inspect your network variables before, during, and after simulation, use the “Variable Viewer”. To learn more about initializing models and setting variable priorities, see “Block-Level Variable Initialization”. Initial conditions for Simscape Fluids models are domain-dependent:
  - In the isothermal liquid domain, the initial pressure must be provided.
  - In the thermal liquid domain, the initial pressure and temperature must be provided.
  - In the gas domain, the initial pressure, and either the density or temperature must be provided.
  - In the moist air domain, the initial pressure, temperature, humidity quality, and trace gas amount must be provided.
  - In the two-phase domain, the initial pressure and either the phase quality, temperature, enthalpy, or internal energy must be provided.
- You can view the results of a simulation by attaching a **Scope** block to a sensor, or by enabling data logging and viewing the results in the Results Explorer. Exporting data from the Results Explorer to the Data Inspector allows you to compare data between simulations. For more information, see “About the Simscape Results Explorer” and Simulation Data Inspector.
  - Scope blocks plot Simulink signals. Convert a Simscape physical signal with a **PS-Simulink Converter** block to connect a physical signal to a Scope block. See “Connecting Simscape Diagrams to Simulink Sources and Scopes” for more information.
  - To turn on data logging, from the Simulink toolstrip, select **Modeling** > **Model Settings** > **Model Settings**. In Configuration Parameters, select Simscape. In the Simscape pane, under **Data Logging**, set **Log simulation data** to `All` and select **Open viewer after simulation**.
- To learn more about selecting and configuring the right solver for your system, see “Setting Up Solvers for Physical Models”. The variable-step solvers `ode23t` and `ode15s` are versatile solvers that are suitable for a range of dynamics. To use Simulink Coder to generate code from your model, use the `ode14x` or `ode1be` solvers. For more information about code generation, see “Code Generation”.

See “Essential Steps for Constructing a Physical Model” for a detailed overview of a Simscape modeling workflow.

Modifying Parameters

You can modify parameter values during simulation if they are run-time configurable. To learn more about configurable parameters and settings, see “About Simscape Run-Time Parameters”.

Balancing Accuracy, Speed, and Efficiency in Your Model

To change the accuracy, speed, and efficiency of your simulation, you can adjust one or more of the following parameters in Configuration Parameters:

- **Relative tolerance**
- **Absolute tolerance**
- **Max step size**

You can also adjust the **Constraint Residual Tolerance** in the Solver Configuration block property inspector.

In most cases, the default tolerance values will produce a good balance between the accuracy and speed of your model. However, in some cases you may wish to adjust the **Max step size** from its default setting of `auto`, which is calculated based on the simulation duration and not on the simulation dynamics. If you are concerned that the solver may be missing significant behavior, you can change this value to a smaller value.

For more information about the solver parameters, see:

- Relative tolerance
- Absolute tolerance
- Max step size

See Solver Configuration to learn more about when to adjust the **Constraint Residual Tolerance** parameter value.

See Also

More About

- “Basic Principles of Modeling Physical Networks”
- “Essential Physical Modeling Techniques”
- “Modeling Best Practices”

Networks and Network Connections

Simscape Fluids models can include multiple fluid networks, each representing a single-fluid system. Only use one fluid properties block per network to avoid model compilation errors. Two fluid networks are distinct if they belong to different Simscape domains or if they are not directly connected.

You can use the blocks in the Fluid Network Interfaces library to connect different domains. For example, you can use the Condenser-Evaporator 2P-MA block to model heat exchange between a two-phase and moist air network.

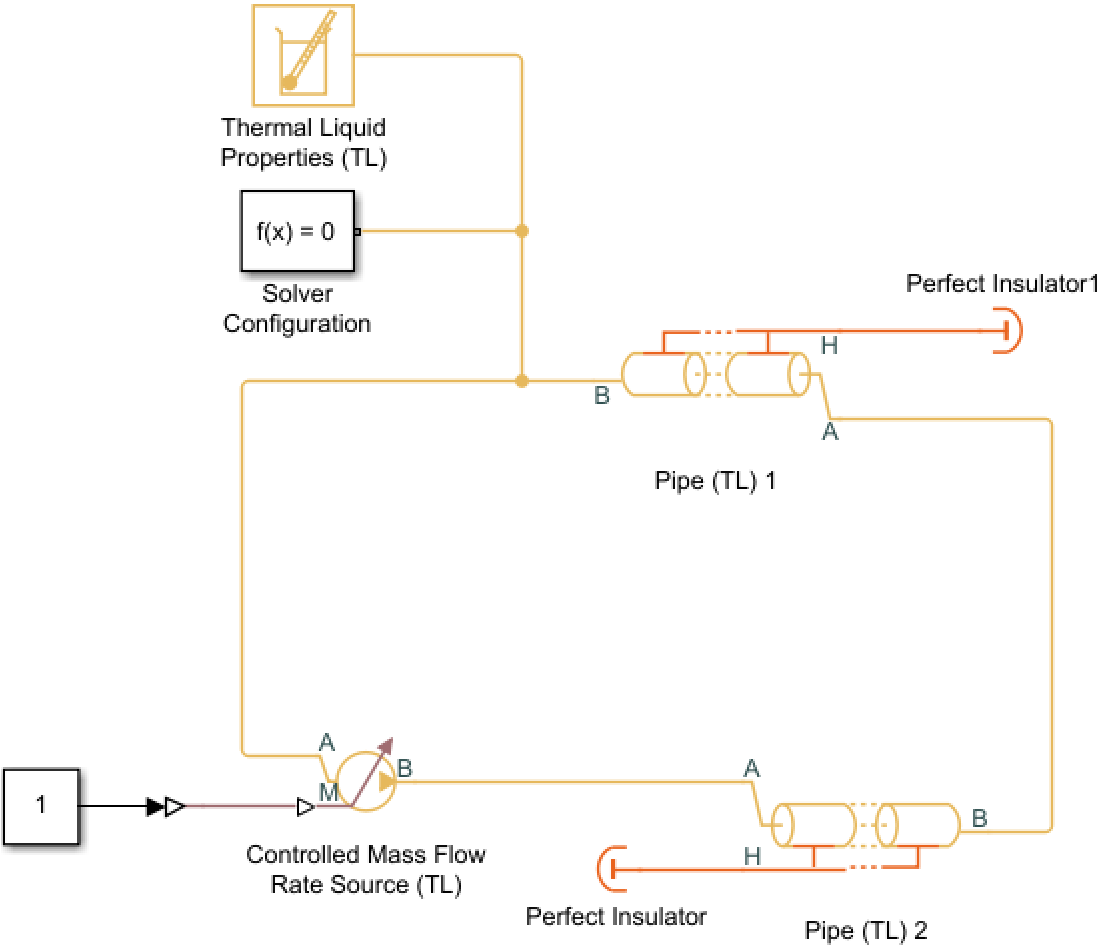
Grounding Rules

The absence of a reference block, such as a Reservoir (TL) block, or dynamic component in a non-isothermal liquid domain may result in variable drift in your model. The internal node of a reference or dynamic component provides a state reference during momentum balance calculations.

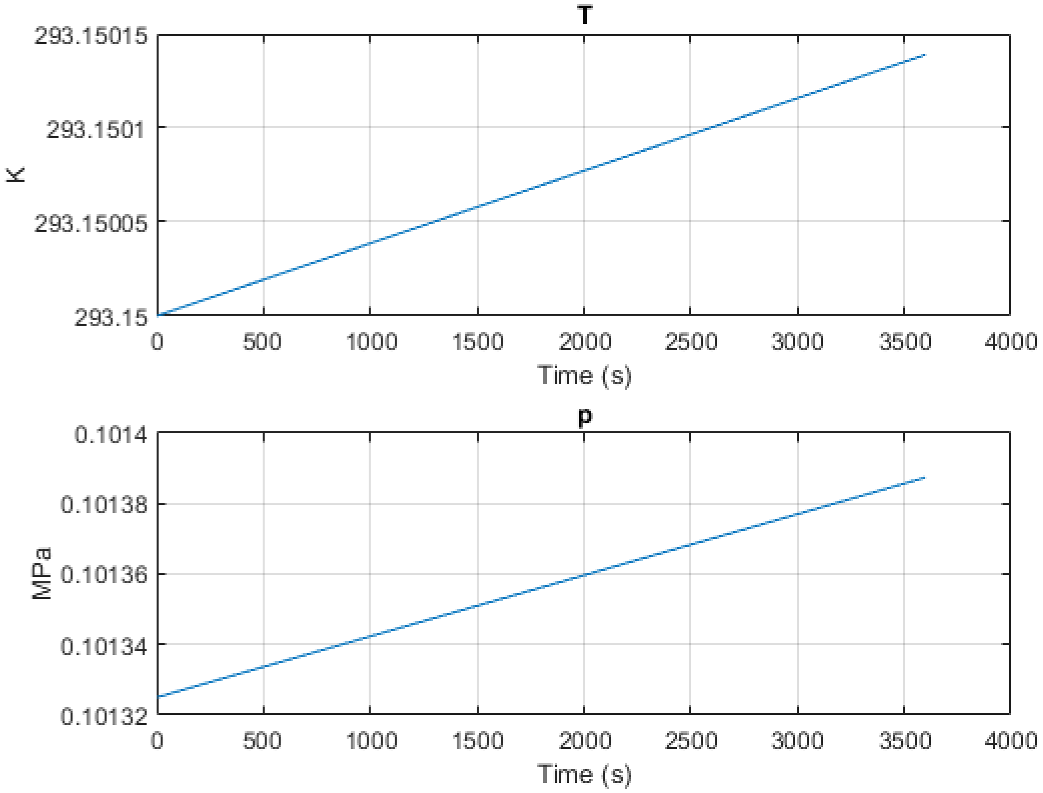
If you observe variable drift during your simulation, add grounding to your model.

Ground a Thermal Liquid Closed Loop

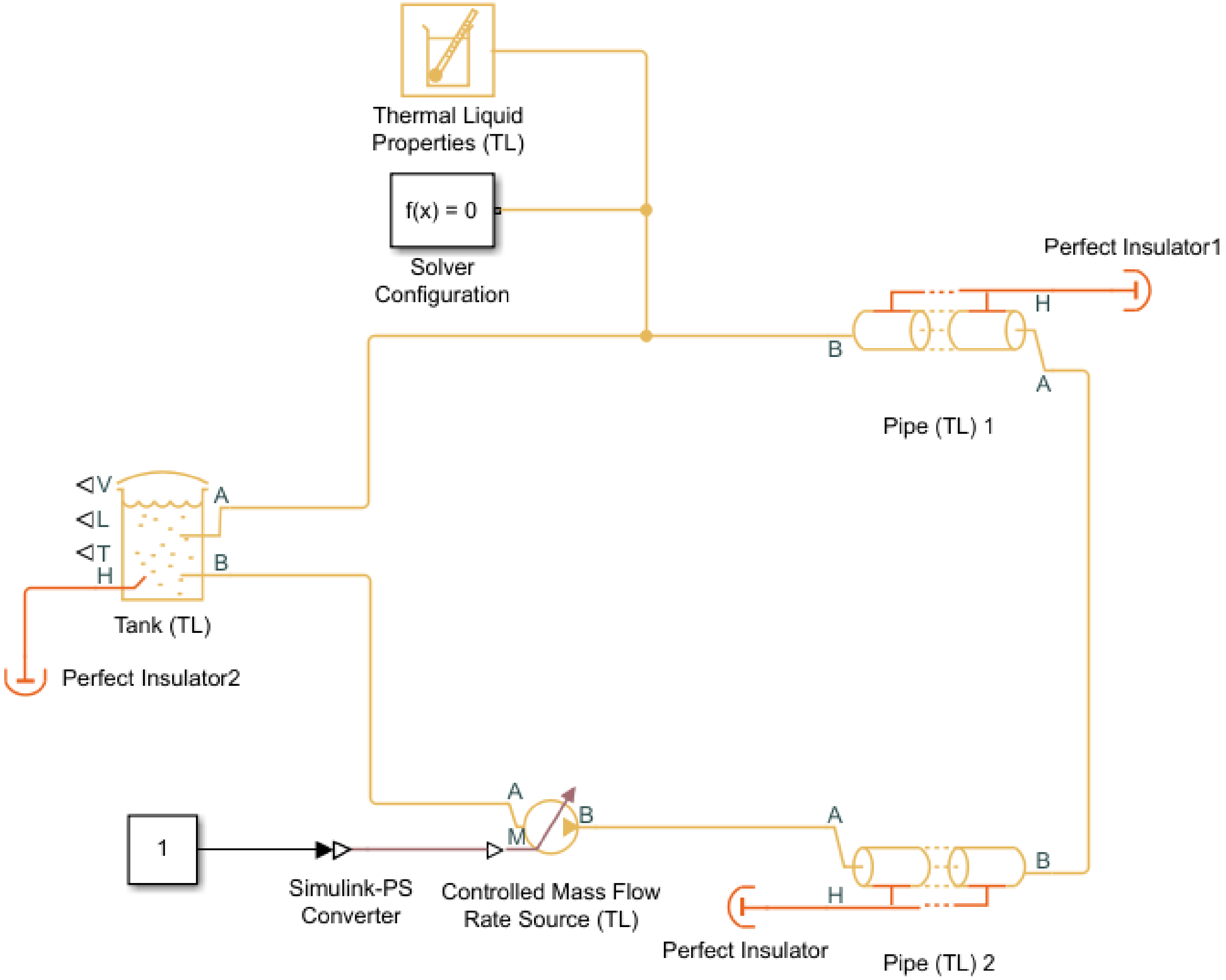
Simulating the example network below, consisting of two pipes connected to a Controlled Mass Flow Rate Source (TL) block, will result in a net increase in pipe temperature and pressure.



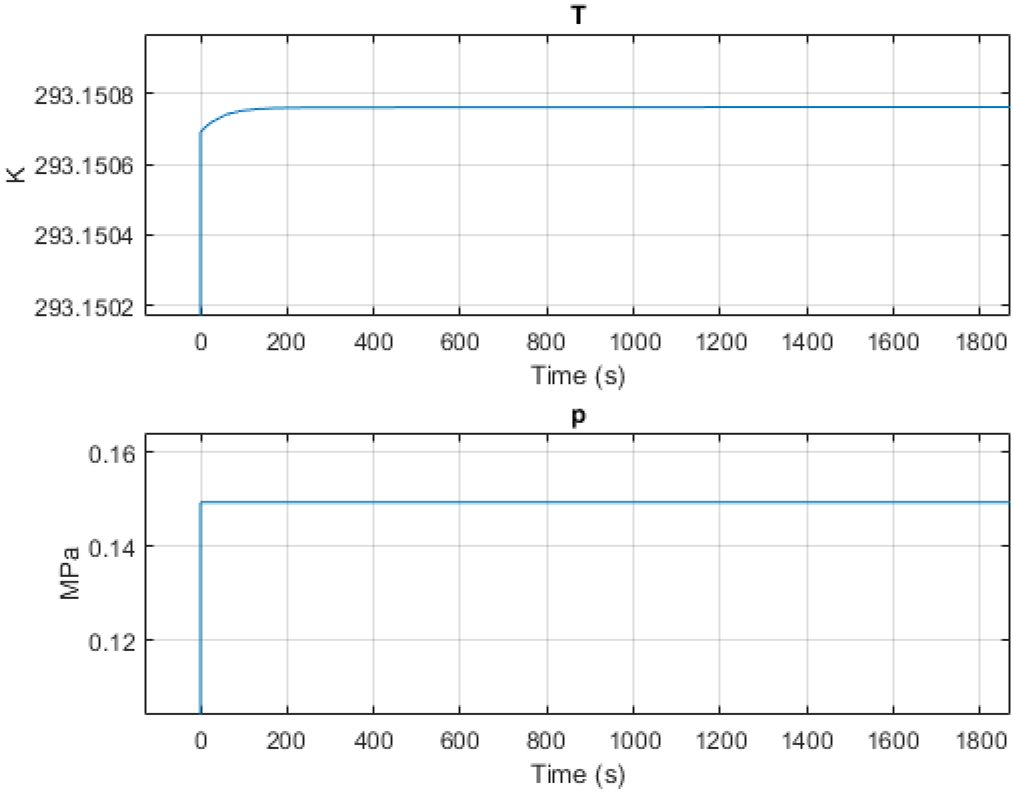
Despite the lack of explicit heat transfer, the absence of thermal grounding in the network results in a net increase of energy due to energy conservation. The increase of energy in the system takes the form of a rise in temperature.



You can mitigate this phenomenon by adding a dynamic component to your network. Add a Tank (TL) block just before the controlled Mass Flow Rate Source (TL). Set the **Number of inlets** parameter of the Tank (TL) to `Two inlets`.



In this scenario, the pipe pressure and temperature reach steady-state. The differential equations associated with the Tank (TL) internal volume reduce the stiffness of the model solution, removing the condition of gradually increasing system temperature and pressure.



See Also

More About

- “Modeling Best Practices”

Troubleshooting Fluids Simulations

Simscape blocks have complex interactions, and incorrectly designing or parameterizing a model can result in simulations that stop before completion with one or more error messages, or fully run but deliver unexpected results. To address these problems, you can troubleshoot the block or the fluids network. For general Simscape troubleshooting techniques, see “Troubleshooting Simulation Errors”.

It is good practice to build and test models incrementally. Start with simulating an idealized, simplified system and verify that it works as expected. Then, change the default settings and add complexity incrementally. You can use subsystems to capture the model hierarchy and test these subsystems separately before testing the entire model.

Baseline Troubleshooting Steps

These steps can help identify basic problems in your model. For more advanced troubleshooting, see “Issues with Specific Model Blocks” on page 1-14 or “Issues in the Fluid Network” on page 1-15.

- Solver Configuration Block — Check that each topologically distinct fluid network has exactly one Solver Configuration block connected to it.
- Units — Be aware of units when converting between Simulink and Simscape signals. Change the **Output signal unit** and **Input signal unit** parameters in PS-Simulink Converter and Simulink-PS Converter blocks from `inherit` and `1`, respectively, to specific values. If you specify the unit, Simscape performs the unit conversion on any input or output signals. For more information, see “How to Work with Physical Units”.
- Network Domain — See “Select the Working Fluid” on page 1-3 to verify that your network fluid is appropriate for your application.
- Fluid Properties Block — Check that each topologically distinct fluid network has no more than one fluid properties block connected to it. For more information, see “Select the Working Fluid” on page 1-3.
- Variables — Use the “Variable Viewer” to inspect variable values, units, and initialization status.
- Absolute Variables — Check that absolute variables, such as pressure or temperature, are positive values. There are no restrictions on the sign of the change in pressure or the change in temperature.
- Affine Units — When specifying temperature, which can be relative or absolute, you may need to apply an affine conversion. For more information about affine conversions, see “How to Apply Affine Conversion”.
- Model Fidelity — Use only as much fidelity as necessary for the model goals.

Issues with Specific Model Blocks

If your model returns an error message that identifies specific model blocks, inspect these blocks for wrong connections or unintended use. For individual blocks, check:

- Valve and Orifice Leakage Area — For blocks with a **Leakage Area** parameter, ensure the parameter is large enough to prevent numerical issues. Parts of the system may become isolated if a small amount of fluid is not present in all components of the network throughout the simulation. Choose a value for the **Leakage Area** parameter that is small enough to be negligible when compared to the maximum valve area, but not as small as possible.

- Dynamic Compressibility — Fluid dynamic compressibility may be relevant for components with internal volumes if the component time constant of the pressure dynamics is slower than inputs to the system. The blocks in the Isothermal Liquid library model dynamic compressibility by default for blocks that support it.
- Scope and Senor Measurements — Sensors and scopes take measurements at the block ports. In components with an internal fluid volume, the results at the ports may not agree with the expected response. You can use the Results Explorer to inspect all logged values of a block, including the properties of the internal fluid volume. For more information, see “About the Simscape Results Explorer”. You can also use a Probe block to probe for the logged values of a block and display the signal on a Scope.
- Block Use — Read the assumptions and limitations in the block documentation for the blocks in your model and ensure your design considers these limitations. For example, an Ideal Pressure Source block can simulate a pump only when the pressure is constant.
- Block Connections — Verify that the model makes sense as a system. For example, look for:

  - Actuators connected against each other, which causes motion in opposite directions.
  - Volume blocks, such as chambers, tanks, or accumulators, that are connected directly together. Place some kind of flow resistance between them, such as a flow resistance, pipe, or orifice block.

  Connections between blocks do not represent physical volumes and values pass along these connections instantaneously.

Issues in the Fluid Network

If your model returns an error message that does not identify specific blocks, you may need to troubleshoot the fluid network.

- Solver Choice — Verify that you are using the best solver for your model. In fluid systems, equations are typically stiff and nonlinear. Using the `ssc_new` function automatically creates a new model with the recommended solver settings:

  - For the solver selection, set **Type** to `Variable-step` and **Solver** to `auto (Automatic solver selection)`.
  - Set **Relative tolerance** and **Absolute tolerance** to `1e-3`.
  - Clear the **Auto scale absolute tolerance** check box.
  - Set **Max step size**, **Min step size**, and **Initial step size** to `auto`.
- Algebraic Loops — Do not use Memory or Unit Delay blocks, which are intended for discrete systems, to break an algebraic loop in a Simscape model. Instead, use a Transfer Fcn block to break algebraic loops in continuous systems. For more information, see Why you should never break a continuous algebraic loop with a Memory block.
- Discrete Controller — If your model uses a discrete controller to drive a continuous Simscape plant, insert a First Order Hold block between the controller and the Simscape model. The First Order Hold block interpolates the discrete controller signal into a continuous signal for the plant. For more information, see A New First Order Hold.
- Dry Nodes — If you have too many quasi-steady components clustered in your network, such as valves, and the calculations occur only at the block ports, your model may be at higher risk for dry nodes, which may slow or stop your simulation. You can improve simulation convergence in these scenarios by adding blocks with internal volumes, which add dynamic components that can

respond to dynamic shifts in the network. You can check for dry nodes with the Model Advisor by running **Check for dry hydraulic nodes** in **By Product** > **Simscape**. This check only applies to blocks in the Hydraulic (Isothermal Liquid) domain.

- Loop Elevation — If you specify elevation change in one component, such as a pipe, you must ensure that the rest of the system is consistent with this elevation change, including the inlet heights in tank blocks. If you have a fluid loop, the net elevation change must sum to zero after a full circle around the loop.
- Grounding Rules — Every fluid network must contain at least one block that models fluid volume, such as a pipe with compressibility, a chamber, or a reservoir. Internal nodes inside these blocks represent the fluid volume, which serves as a reference point for the Across variables in the network. If the loop only contains quasi-steady components, such as valves and pumps, the solver fails.

See Also

"Troubleshooting Simulation Errors"

More About

- "Modeling Best Practices"
- "Essential Steps for Constructing a Physical Model"
- "Basic Principles of Modeling Physical Networks"

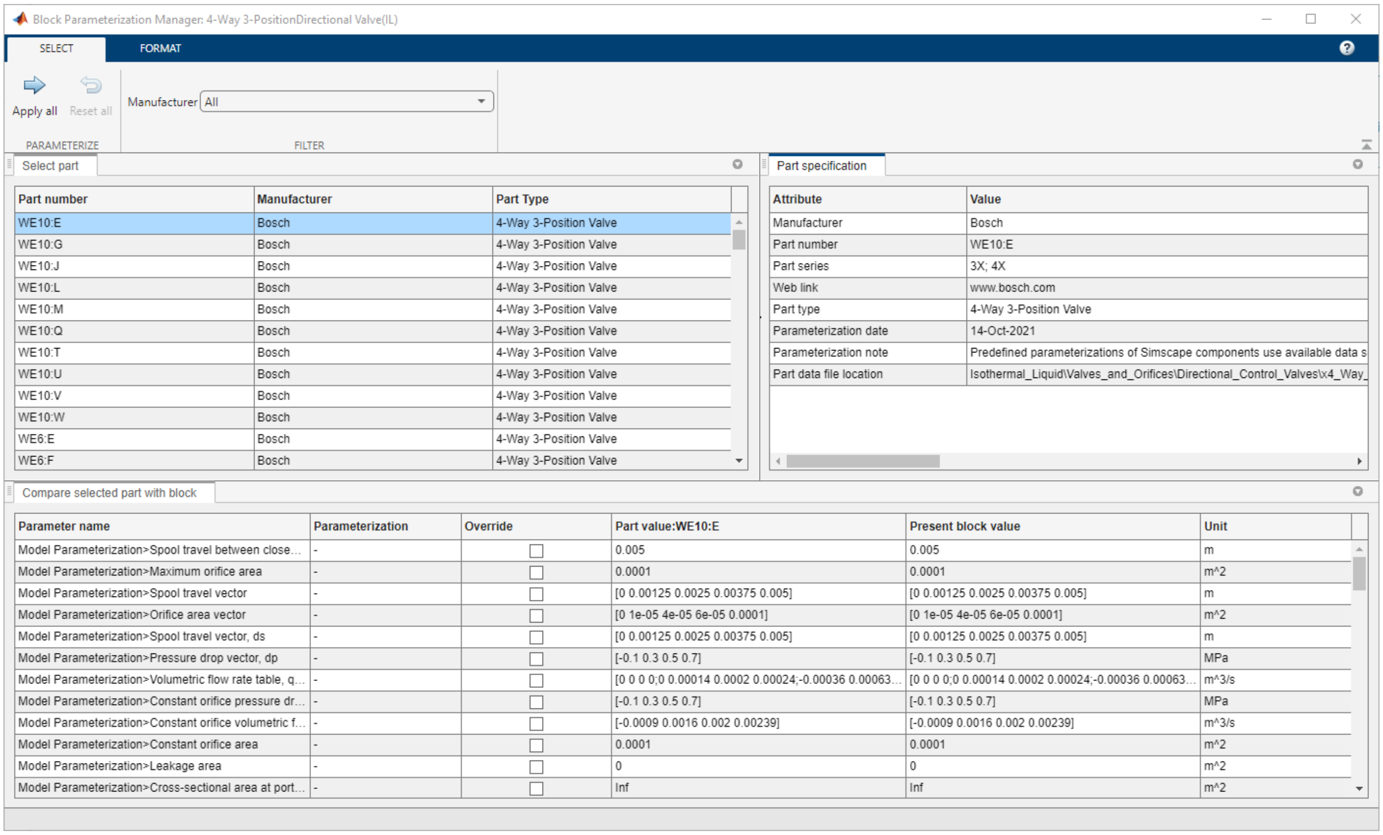
List of Pre-Parameterized Components

Many of the blocks in Simscape Fluids have parameters that correspond to common manufacturer data for their respective component. Some of these blocks also feature predefined parameterizations that give you the ability to automatically parameterize a block according to a stored set of manufacturer data. This is a comprehensive list of the available predefined parameterizations for Simscape Fluids blocks. To load a predefined parameterization, click **Select a predefined parameterization** in the **Description** and select the component from the **Block Parameterization Manager**.

Note The predefined parameterizations of Simscape components use available data sources for the parameter values. Engineering judgement and simplifying assumptions are used to fill in for missing data. As a result, expect deviations between simulated and actual physical behavior. To ensure accuracy, validate the simulated behavior against experimental data and refine component models as necessary.

Load Predefined Parameterizations

To load a predefined parameterization, double-click a block, then click the **<click to select>** hyperlink of the **Selected part** parameter. The Block Parameterization Manager window opens.


Block Parameterization Manager: 4-Way 3-PositionDirectional Valve(IL)
SELECT
FORMAT
Apply all
Reset all
PARAMETERIZE
FILTER
Select part
Part specification
Compare selected part with block

Manufacturer All

Part number	Manufacturer	Part Type
WE10:E	Bosch	4-Way 3-Position Valve
WE10:G	Bosch	4-Way 3-Position Valve
WE10:J	Bosch	4-Way 3-Position Valve
WE10:L	Bosch	4-Way 3-Position Valve
WE10:M	Bosch	4-Way 3-Position Valve
WE10:Q	Bosch	4-Way 3-Position Valve
WE10:T	Bosch	4-Way 3-Position Valve
WE10:U	Bosch	4-Way 3-Position Valve
WE10:V	Bosch	4-Way 3-Position Valve
WE10:W	Bosch	4-Way 3-Position Valve
WE6:E	Bosch	4-Way 3-Position Valve
WE6:F	Bosch	4-Way 3-Position Valve

Attribute	Value
Manufacturer	Bosch
Part number	WE10:E
Part series	3X; 4X
Web link	www.bosch.com
Part type	4-Way 3-Position Valve
Parameterization date	14-Oct-2021
Parameterization note	Predefined parameterizations of Simscape components use available data s
Part data file location	Isothermal_Liquid\Valves_and_Orifices\Directional_Control_Valves\x4_Way_

Parameter name	Parameterization	Override	Part value:WE10:E	Present block value	Unit
Model Parameterization>Spool travel between close...	-	□	0.005	0.005	m
Model Parameterization>Maximum orifice area	-	□	0.0001	0.0001	m^2
Model Parameterization>Spool travel vector	-	□	[0 0.00125 0.0025 0.00375 0.005]	[0 0.00125 0.0025 0.00375 0.005]	m
Model Parameterization>Orifice area vector	-	□	[0 1e-05 4e-05 6e-05 0.0001]	[0 1e-05 4e-05 6e-05 0.0001]	m^2
Model Parameterization>Spool travel vector, ds	-	□	[0 0.00125 0.0025 0.00375 0.005]	[0 0.00125 0.0025 0.00375 0.005]	m
Model Parameterization>Pressure drop vector, dp	-	□	[-0.1 0.3 0.5 0.7]	[-0.1 0.3 0.5 0.7]	MPa
Model Parameterization>Volumetric flow rate table, q...	-	□	[0 0 0 0;0 0.00014 0.0002 0.00024;-0.00036 0.00063...	[0 0 0 0;0 0.00014 0.0002 0.00024;-0.00036 0.00063...	m^3/s
Model Parameterization>Constant orifice pressure dr...	-	□	[-0.1 0.3 0.5 0.7]	[-0.1 0.3 0.5 0.7]	MPa
Model Parameterization>Constant orifice volumetric f...	-	□	[-0.0009 0.0016 0.002 0.00239]	[-0.0009 0.0016 0.002 0.00239]	m^3/s
Model Parameterization>Constant orifice area	-	□	0.0001	0.0001	m^2
Model Parameterization>Leakage area	-	□	0	0	m^2
Model Parameterization>Cross-sectional area at port...	-	□	Inf	Inf	m^2

List of Pre-Parameterized Components in Isothermal Liquid Library

This list shows the available pre-parameterization options in the Isothermal Liquid Library.

Block	Manufacturer	Component
Centrifugal Pump (IL)	Ebara	EVMS(L)1 10/0.55*
		EVMS(L)1 11/0.55*
		EVMS(L)1 12/0.55*
		EVMS(L)1 13/0.55*
		EVMS(L)1 14/0.75*
		EVMS(L)1 16/0.75*
		EVMS(L)1 18/1.1*
		EVMS(L)1 20/1.1*
		EVMS(L)1 22/1.1*
		EVMS(L)1 24/1.1*
		EVMS(L)1 26/1.1*
		EVMS(L)1 27/1.5*
		EVMS(L)1 29/1.5*
		EVMS(L)1 2/0.37*
		EVMS(L)1 32/1.5*
		EVMS(L)1 34/1.5*
		EVMS(L)1 37/2.2*
		EVMS(L)1 39/2.2*
		EVMS(L)1 3/0.37*
		EVMS(L)1 4/0.37*
		EVMS(L)1 5/0.37*
		EVMS(L)1 6/0.37*
		EVMS(L)1 7/0.37*
		EVMS(L)1 8/0.37*
		EVMS(L)1 9/0.55*
		EVMS(L)5 10/2.2*
		EVMS(L)5 11/2.2*
		EVMS(L)5 12/3.0*
		EVMS(L)5 13/3.0*
		EVMS(L)5 14/3.0*
		EVMS(L)5 15/3.0*
		EVMS(L)5 17/4.0*
		EVMS(L)5 19/4.0*
		EVMS(L)5 20/4.0*

Block	Manufacturer	Component
		EVMS(L)5 23/5.5*
		EVMS(L)5 25/5.5*
		EVMS(L)5 27/5.5*
		EVMS(L)5 2/0.37*
		EVMS(L)5 3/0.55*
		EVMS(L)5 4/0.75*
		EVMS(L)5 5/1.1*
		EVMS(L)5 6/1.5*
		EVMS(L)5 7/1.5*
		EVMS(L)5 8/2.2*
		EVMS(L)5 9/2.2*
	Grundfos	SP:10S05-9
		SP:10S07-12
		SP:10S10-15
		SP:10S15-21
		SP:10S20-27
		SP:10S30-34
		SP:10S50-58DS
		SP:16S05-5
		SP:16S07-8
		SP:16S100-75DS
		SP:16S10-10
		SP:16S15-14
		SP:16S20-18
		SP:16S30-24
		SP:16S50-38
		SP:16S75-56DS
		SP:25S05-3
		SP:25S07-5
		SP:25S100-52DS
		SP:25S10-7
		SP:25S15-9
		SP:25S20-11
		SP:25S30-15
		SP:25S50-26
		SP:25S75-39DS

Block	Manufacturer	Component
		SP:40S100-30
		SP:40S10-3
		SP:40S150-44DS
		SP:40S15-5
		SP:40S200-66DS
		SP:40S20-7
		SP:40S30-9
		SP:40S50-15
		SP:40S75-25
		SP:5S05-13
		SP:5S07-18
		SP:5S10-22
		SP:5S15-31
		SP:5S20-39DS
		SP:5S30-48DS
		SP:75S100-16
		SP:75S20-3
		SP:75S30-5
		SP:75S50-8
		SP:75S75-12
		SP:7S05-11
		SP:7S07-15
		SP:7S10-19
		SP:7S15-26
		SP:7S20-32
Counterbalance Valve (IL)	Rexroth Bosch Group	05.41.01-03-03-35
		05.41.01-03-04-35
		05.41.01-10-03-35
		05.41.01-10-04-35
		08.35.20-03-09-35
		08.39.59-03-03-35
		08.39.60-03-03-35
		08.39.62-03-03-35
		08.41.01-03-03-35
		08.41.02-03-*-35
		08.45.85-13-03-35

Block	Manufacturer	Component
		08.47.72-03-03-20
		08.49.68-03-09-35
		08.49.85-03-09-35
	Sun Hydraulics	CBCB***:1
		CBCB***:2
		CBCB***:3
		CBCB***:4
		CBCH***:1
		CBCH***:2
		CBCH***:3
		CBCH***:4
		CBCL***:1
		CBCL***:2
		CBCL***:3
		CBCL***:4
		CBCY***:1
		CBCY***:2
		CBCY***:3
		CBCY***:4
		CBEB***:1
		CBEB***:2
		CBEB***:3
		CBEB***:4
		CBEH***:1
		CBEH***:2
		CBEH***:3
		CBEH***:4
		CBEL***:1
		CBEL***:2
		CBEL***:3
		CBEL***:4
		CBEY***:1
		CBEY***:2
		CBEY***:3
		CBEY***:4
		CBIH***:1

Block	Manufacturer	Component
		CBIH***:2
		CBIH***:3
		CBIH***:4
		CBIL***:1
		CBIL***:2
		CBIL***:3
		CBIL***:4
Fixed-Displacement Pump (IL)	Eaton	GD5:05
		GD5:06
		GD5:09
		GD5:11
		GD5:12
		GD5:16
		GD5:18
		GD5:20
		GD5:24
		PVM:018
		PVM:045
		PVM:057
		PVM:074
		PVM:098
		PVM:131
	Parker	PV:016
		PV:020
		PV:023
		PV:032
		PV:040
		PV:046
		PV:063
		PV:080
		PV:092
		PV:140
		PV:180
		PV:270
		PV:360
Fixed-Displacement Motor (IL)	Eaton	ME100

Block	Manufacturer	Component
		ME1300A
		ME150
		ME175
		ME1900
		ME2600
		ME300B
		ME3100
		ME350B
		ME4100
		ME600B
		ME750B
		ME850B
	INI Hydraulic	INM05_150
		INM1_150
		INM2_300
		INM3_600
		INM4_900
		INM5_1200
		INM6_2500
	Kawasaki	K3X112
		M3X200
		M3X280
		M3X530
		M3X800
4-Way 2-Position Directional Valve (IL)	Rexroth Bosch Group	WE10:C
		WE10:D
		WE10:Y
		WE6:C
		WE6:D
		WE6:Y
		5-.WE:C
		5-.WE:D
		5-.WE:Y
		5-.WE:Y11
	Daikin	JSP-G02:2B
		JSP-G02:2N

Block	Manufacturer	Component
		JSP-G02:33B
		JSP-G02:33N
		JSP-G03:2B
		JSP-G03:2N
		JSP-G03:33B
		KSH-G04:2B
		KSH-G04:2D
		KSH-G04:33B
		KSH-G04:33D
		KSH-G04:3B
		KSH-G04:3D
		KSO-G02:20A
		KSO-G02:20B
		KSO-G02:20D
		KSO-G02:20N
		KSO-G02:2A
		KSO-G02:2B
		KSO-G02:2D
		KSO-G02:2N
		KSO-G02:3A
		KSO-G02:3B
		LS-G02:20A
		LS-G02:20B
		LS-G02:20D
		LS-G02:20N
		LS-G02:2A
		LS-G02:2B
		LS-G02:2D
		LS-G02:2N
	Eaton	S542
		SV11-8-4/4M
		SV17M-12-4
		SV1-8-4/4M
		SV7M-12-4
		SV7-10-4/4P/4S
	HAWE Hydraulik	B

Block	Manufacturer	Component
		W
	Kawasaki	DE10:101
		DE10:103
		DE10:104
		DE10:201
		DE10:202
		DE10:203
		DE10:204
		DE10:225
		DE6:104
		DE6:201
		DE6:203
		DE6:204
	Yuken	DSG-01:2B2
		DSG-01:2B3
		DSG-01:2B8
		DSG-01:2D2
		DSG-03:2B2
		DSG-03:2B3
		DSG-03:2B8
		DSG-03:2D2
		S-DSG-01:2B2
		S-DSG-03:2B2
4-Way 3-Position Directional Valve (IL)	Rexroth Bosch Group	WE10:E
		WE10:G
		WE10:J
		WE10:L
		WE10:M
		WE10:Q
		WE10:T
		WE10:U
		WE10:V
		WE10:W
		WE6:E
		WE6:F
		WE6:G

Block	Manufacturer	Component
		WE6:J
		WE6:L
		WE6:M
		WE6:P
		WE6:Q
		WE6:T
		WE6:U
		WE6:V
		WE6:W
		5-.WE:E
		5-.WE:G
		5-.WE:J
		5-.WE:L
		5-.WE:Q
		5-.WE:R
		5-.WE:U
		5-.WE:W
	Daikin	JSP-G02:2C
		JSP-G02:33C
		JSP-G02:3C
		JSP-G02:44C
		JSP-G02:4C
		JSP-G02:66C
		JSP-G03:2C
		JSP-G03:3C
		JSP-G03:44C
		JSP-G03:4C
		JSP-G03:66C
		KSH-G04:2C
		KSH-G04:2H
		KSH-G04:33C
		KSH-G04:33H
		KSH-G04:3C
		KSH-G04:3H
		KSH-G04:44C
		KSH-G04:44H

Block	Manufacturer	Component
		KSH-G04:4C
		KSH-G04:4H
		KSH-G04:5C
		KSH-G04:5H
		KSH-G04:66C
		KSH-G04:66H
		KSH-G04:6C
		KSH-G04:6H
		KSH-G04:81C
		KSH-G04:81H
		KSH-G04:8C
		KSH-G04:8H
		KSH-G04:91C
		KSH-G04:91H
		KSH-G04:9C
		KSH-G04:9H
		KSO-G02:2C
		KSO-G02:44C
		KSO-G02:4C
		KSO-G02:66C
		KSO-G02:7C
		KSO-G02:81C
		KSO-G02:8C
		KSO-G02:91C
		KSO-G02:9C
		LS-G02:2C
		LS-G02:44C
		LS-G02:4C
		LS-G02:66C
		LS-G02:7C
		LS-G02:8C
		LS-G02:9C
	Eaton	SV9M-10-A
		SV9M-10-E
		SV9-10-A
		SV9-10-E

Block	Manufacturer	Component
		SV9-10-F
		SV9-8-A
		SV9-8-E
	HAWE Hydraulik	D
		G
		L
		X
	Kawasaki	DE10:205
		DE10:207
		DE10:209
		DE10:210
		DE10:212
		DE10:213
		DE10:217
		DE10:221
		DE10:222
		DE6:205
		DE6:207
		DE6:210
		DE6:212
		DE6:213
		DE6:217
		DE6:221
	Yuken	DSG-01:3C10
		DSG-01:3C11
		DSG-01:3C12
		DSG-01:3C2
		DSG-01:3C3
		DSG-01:3C4
		DSG-01:3C40
		DSG-01:3C60
		DSG-01:3C9
		DSG-03:3C10
		DSG-03:3C11
		DSG-03:3C12
		DSG-03:3C2

Block	Manufacturer	Component
		DSG-03:3C3
		DSG-03:3C4
		DSG-03:3C40
		DSG-03:3C5
		DSG-03:3C60
		DSG-03:3C9
		S-DSG-01:3C2
		S-DSG-01:3C4
		S-DSG-03:3C2
		S-DSG-03:3C4
Check Valve (IL)	Rexroth Bosch Group	M-SR 08 KE00-1X/
		M-SR 08 KE02-1X/
		M-SR 08 KE05-1X/
		M-SR 08 KE15-1X/
		M-SR 08 KE30-1X/
		M-SR 08 KE50-1X/
		M-SR 10 KE00-1X/
		M-SR 10 KE02-1X/
		M-SR 10 KE05-1X/
		M-SR 10 KE15-1X/
		M-SR 10 KE30-1X/
		M-SR 10 KE50-1X/
		M-SR 15 KE00-1X/
		M-SR 15 KE02-1X/
		M-SR 15 KE05-1X/
		M-SR 15 KE15-1X/
		M-SR 15 KE30-1X/
		M-SR 15 KE50-1X/
		M-SR 20 KE00-1X/
		M-SR 20 KE02-1X/
		M-SR 20 KE05-1X/
		M-SR 20 KE15-1X/
		M-SR 20 KE30-1X/
		M-SR 20 KE50-1X/
		M-SR 25 KE00-1X/
		M-SR 25 KE02-1X/

Block	Manufacturer	Component
		M-SR 25 KE05-1X/
		M-SR 25 KE15-1X/
		M-SR 25 KE30-1X/
		M-SR 25 KE50-1X/
		M-SR 30 KE00-1X/
		M-SR 30 KE02-1X/
		M-SR 30 KE05-1X/
		M-SR 30 KE15-1X/
		M-SR 30 KE30-1X/
		M-SR 30 KE50-1X/
	Continental Hydraulics	C03MSV-D*-43-*C-C
		C03MSV-D*-70-*C-C
		C03MSV-D*-7-*C-C
		C05MSV-D*-116-*C-C
		C05MSV-D*-7-*C-C
	Daikin	HDIN-F06-05
		HDIN-F10-05
		HDIN-F12-05
		HDIN-F16-05
		HDIN-T03-05
		HDIN-T06-05
		HDIN-T10-05
	Delta Power Company	DE-CVA-**-0010-*
		DE-CVB-**-0005-*
		DE-CVC-**-0003-*
		DE-CVS-**-0010-*
		HB-CVA-**-0005-*
		HE-CVA-**-0005-*
		HT-CVA-**-0010-*
		MA-CVA-**-0005-*
		MA-CVS-**-0005-*
		PB-CVC-**-0005-*
		PB-CVS-**-0005-*
		SJ-CVA-**-0005-*
	Eaton	CV16-10-P-O-5
		CV16-10-P-O-50

Block	Manufacturer	Component
		CV2-20-P-O-100
		CV2-20-P-O-30
		CV2-20-P-O-5
		CV3-4-P-000-05
		CV3-4-P-000-60
		CV(*)3 10*P 000 003 00
		CV(*)3 10*P 000 210 00
		FPR11/4-1
		FPR1-1
		FPR1/2-0.5
		FPR1/2-1
		FPR1/2-5
		FPR1/4-1
		FPR1/4-5
		FPR3/4-0.5
		FPR3/4-1
		FPR3/4-5
		FPR3/8-5
	Hawe Hydraulik	B*-1
		B*-2
		B*-3
		B*-4
		B*-5
		B*-6
		B*-7
		CRB 1
		CRB 2
		CRK 1
		CRK 2
		CRK 3
		EK 01
		ER 01
		ER 1*
		ER 21
		ER 31
		ER 41

Block	Manufacturer	Component
		RB 0
		RB 08
		RB 1
		RB 2
		RB 3
		RB 4
		RC 1
		RC 2
		RC 3
		RK 0
		RK 08
		RK 1
		RK 2
		RK 3
		RK 4
3-Way Directional Valve (IL)	Eaton	S229
		S520(H)
		S521(H)
		S525(H)
		S526
		SV11A/B-10-3
		SV11M-12-3
		SV11-10-3/3M
		SV11-8-3/3M
		SV12M-12-3
		SV14M-12-3
		SV1M-12-3
		SV1-8-3/3M
		SV2M-12-3P
		SV4M-12-3P
		SV4-10-3/3P/3S
		SV4-8-3/3P
	Parker	A
		C
		N04A4
		N04B4

Block	Manufacturer	Component
		N5B125
		N5B300
Pressure Relief Valve (IL)	Daikin	CR-*02*-1-10-*: Set Pressure 4.5 MPa
		CR-*02*-1-10-*: Set Pressure 6 MPa
		CR-*02*-2-10-*: Set Pressure 13 MPa
		CR-*02*-2-10-*: Set Pressure 14.5 MPa
		CR-*02*-3-10-*: Set Pressure 22 MPa
		CR-*02*-3-10-*: Set Pressure 23.5 MPa
		CR-*02*-4-10-*: Set Pressure 28 MPa
		CR-*02*-4-10-*: Set Pressure 29.5 MPa
		HDRIR-*02-1: Set Pressure 5.4 MPa
		HDRIR-*02-3: Set Pressure 19.2 MPa
		JR-*02-1-22-*: Set Pressure 5 MPa
		JR-*02-3-22-*: Set Pressure 19 MPa
		SR-*03-1-**: Set Pressure 2.7 MPa
		SR-*03-1-**: Set Pressure 3 MPa
		SR-*03-1-**: Set Pressure 6 MPa
	Kawasaki Heavy Industries	RD10*-10-*/100
		RD10*-10-*/200
		RD10*-10-*/25
		RD10*-10-*/315
		RD10*-10-*/50
		RD15*-10-*/100
		RD15*-10-*/200
		RD15*-10-*/25
		RD15*-10-*/315
		RD15*-10-*/50

Block	Manufacturer	Component
		RD20*-10-*/100
		RD20*-10-*/200
		RD20*-10-*/25
		RD20*-10-*/315
		RD20*-10-*/50
		RD25*-10-*/100
		RD25*-10-*/200
		RD25*-10-*/25
		RD25*-10-*/315
		RD25*-10-*/50
		RD30*-10-*/100
		RD30*-10-*/200
		RD30*-10-*/25
		RD30*-10-*/315
		RD30*-10-*/50
		RD6*-10-*/100
		RD6*-10-*/200
		RD6*-10-*/25
		RD6*-10-*/315
		RD6*-10-*/50
		RD8*-10-*/100
		RD8*-10-*/200
		RD8*-10-*/25
		RD8*-10-*/315
		RD8*-10-*/50
	Parker	RD102*09*: Set Pressure 34 bar
		RD102*18*: Set Pressure 69 bar
		RD102*30*: Set Pressure 138 bar
		RD102*30*: Set Pressure 207 bar
		RDH042*30*: Set Pressure 138 bar
		RDH042*30*: Set Pressure 207 bar
		RDH042*30*: Set Pressure 69 bar

Block	Manufacturer	Component
		RDH042*50*: Set Pressure 276 bar
		RDH042*50*: Set Pressure 345 bar
		RDH081*10*: Set Pressure 69 bar
		RDH081*30*: Set Pressure 138 bar
		RDH081*30*: Set Pressure 207 bar
		RDH081*50*: Set Pressure 276 bar
		RDH081*50*: Set Pressure 345 bar
		RDH082*15*: Set Pressure 69 bar
		RDH082*30*: Set Pressure 138 bar
		RDH082*30*: Set Pressure 207 bar
		RDH082*50*: Set Pressure 276 bar
		RDH082*50*: Set Pressure 345 bar
		RDH101*10*: Set Pressure 69 bar
		RDH101*30*: Set Pressure 138 bar
		RDH101*30*: Set Pressure 207 bar
		RDH101*50*: Set Pressure 276 bar
		RDH101*50*: Set Pressure 345 bar
	Yuken	DG-02-B-22*: Set Pressure 5.3 MPa
		DG-02-C-22*: Set Pressure 11.7 MPa
		DG-02-H-22*: Set Pressure 17.5 MPa
		DT-02-B-22*: Set Pressure 5.3 MPa

Block	Manufacturer	Component
		DT-02-C-22*: Set Pressure 11.7 MPa
		DT-02-H-22*: Set Pressure 17.5 MPa
Pressure-Reducing Valve (IL)	Hawe Hydraulik	CDK3-2-100
		CDK3-2-150
		CDK3-2-20
		CDK3-2-200
		CDK3-2-50
	Kawasaki Heavy Industries	PRD10P-10-1/150YC
		PRD10P-10-1/210YC
		PRD10P-10-1/25YC
		PRD10P-10-1/75YC
		PRD6P-10-1/150YC
		PRD6P-10-1/210YC
		PRD6P-10-1/25YC
		PRD6P-10-1/75YC
	Rexroth Bosch Group	DR6DP*-5X/150YM
		DR6DP*-5X/210YM
		DR6DP*-5X/25YM
		DR6DP*-5X/315YM
		DR6DP*-5X/75YM
		KRD2F2AB/LV
		KRD2L2AB/LV
		KRD2P2AB/LV
		ZDR6D**-4X/150YM
		ZDR6D**-4X/210YM
		ZDR6D**-4X/25YM
		ZDR6D**-4X/315YM
		ZDR6D**-4X/75YM
	Sun Hydraulics	PRBRL*N: Set pressure 127 psi
		PRBRL*N: Set pressure 2641 psi
		PRBRL*N: Set pressure 564 psi
		PRDRL*N: Set pressure 1102 psi
		PRDRL*N: Set pressure 155 psi
		PRDRL*N: Set pressure 613 psi

Block	Manufacturer	Component
		PRFRL*N: Set pressure 1406 psi
		PRHRL*N: Set pressure 1363 psi
		PRHRL*N: Set pressure 700 psi
		PRJRL*N: Set pressure 1083 psi
		PRJRL*N: Set pressure 2101 psi
		PRJRL*N: Set pressure 440 psi

List of Pre-Parameterized Components in Thermal Liquid Library

This list shows the available pre-parameterization options in the Thermal Liquid Library.

Block	Manufacturer	Component
Pressure Relief Valve (TL)	Daikin	CR-*02*-1-10-*: Set Pressure 4.5 MPa
		CR-*02*-1-10-*: Set Pressure 6 MPa
		CR-*02*-2-10-*: Set Pressure 13 MPa
		CR-*02*-2-10-*: Set Pressure 14.5 MPa
		CR-*02*-3-10-*: Set Pressure 22 MPa
		CR-*02*-3-10-*: Set Pressure 23.5 MPa
		CR-*02*-4-10-*: Set Pressure 28 MPa
		CR-*02*-4-10-*: Set Pressure 29.5 MPa
		HDRIR-*02-1: Set Pressure 5.4 MPa
		HDRIR-*02-3: Set Pressure 19.2 MPa
		JR-*02-1-22-*: Set Pressure 5 MPa
		JR-*02-3-22-*: Set Pressure 19 MPa
		SR-*03-1-**: Set Pressure 2.7 MPa
		SR-*03-1-**: Set Pressure 3 MPa
		SR-*03-1-**: Set Pressure 6 MPa

Block	Manufacturer	Component
	Kawasaki Heavy Industries	RD10*-10-*/100
		RD10*-10-*/200
		RD10*-10-*/25
		RD10*-10-*/315
		RD10*-10-*/50
		RD15*-10-*/100
		RD15*-10-*/200
		RD15*-10-*/25
		RD15*-10-*/315
		RD15*-10-*/50
		RD20*-10-*/100
		RD20*-10-*/200
		RD20*-10-*/25
		RD20*-10-*/315
		RD20*-10-*/50
		RD25*-10-*/100
		RD25*-10-*/200
		RD25*-10-*/25
		RD25*-10-*/315
		RD25*-10-*/50
		RD30*-10-*/100
		RD30*-10-*/200
		RD30*-10-*/25
		RD30*-10-*/315
		RD30*-10-*/50
		RD6*-10-*/100
		RD6*-10-*/200
		RD6*-10-*/25
		RD6*-10-*/315
		RD6*-10-*/50
		RD8*-10-*/100
		RD8*-10-*/200
		RD8*-10-*/25
		RD8*-10-*/315
		RD8*-10-*/50
	Parker	RD102*09*: Set Pressure 34 bar

Block	Manufacturer	Component
		RD102*18*: Set Pressure 69 bar
		RD102*30*: Set Pressure 138 bar
		RD102*30*: Set Pressure 207 bar
		RDH042*30*: Set Pressure 138 bar
		RDH042*30*: Set Pressure 207 bar
		RDH042*30*: Set Pressure 69 bar
		RDH042*50*: Set Pressure 276 bar
		RDH042*50*: Set Pressure 345 bar
		RDH081*10*: Set Pressure 69 bar
		RDH081*30*: Set Pressure 138 bar
		RDH081*30*: Set Pressure 207 bar
		RDH081*50*: Set Pressure 276 bar
		RDH081*50*: Set Pressure 345 bar
		RDH082*15*: Set Pressure 69 bar
		RDH082*30*: Set Pressure 138 bar
		RDH082*30*: Set Pressure 207 bar
		RDH082*50*: Set Pressure 276 bar
		RDH082*50*: Set Pressure 345 bar
		RDH101*10*: Set Pressure 69 bar
		RDH101*30*: Set Pressure 138 bar
		RDH101*30*: Set Pressure 207 bar

Block	Manufacturer	Component
		RDH101*50*: Set Pressure 276 bar
		RDH101*50*: Set Pressure 345 bar
	Yuken	DG-02-B-22*: Set Pressure 5.3 MPa
		DG-02-C-22*: Set Pressure 11.7 MPa
		DG-02-H-22*: Set Pressure 17.5 MPa
		DT-02-B-22*: Set Pressure 5.3 MPa
		DT-02-C-22*: Set Pressure 11.7 MPa
		DT-02-H-22*: Set Pressure 17.5 MPa
4-Way 3-Position Directional Valve (TL)	Daikin	JSP-G02:2C
		JSP-G02:33C
		JSP-G02:3C
		JSP-G02:44C
		JSP-G02:4C
		JSP-G02:66C
		JSP-G03:2C
		JSP-G03:3C
		JSP-G03:44C
		JSP-G03:4C
		JSP-G03:66C
		KSH-G04:2C
		KSH-G04:2H
		KSH-G04:33C
		KSH-G04:33H
		KSH-G04:3C
		KSH-G04:3H
		KSH-G04:44C
		KSH-G04:44H
		KSH-G04:4C
		KSH-G04:4H
		KSH-G04:5C
		KSH-G04:5H

Block	Manufacturer	Component
		KSH-G04:66C
		KSH-G04:66H
		KSH-G04:6C
		KSH-G04:6H
		KSH-G04:81C
		KSH-G04:81H
		KSH-G04:8C
		KSH-G04:8H
		KSH-G04:91C
		KSH-G04:91H
		KSH-G04:9C
		KSH-G04:9H
		KSO-G02:2C
		KSO-G02:44C
		KSO-G02:4C
		KSO-G02:66C
		KSO-G02:7C
		KSO-G02:81C
		KSO-G02:8C
		KSO-G02:91C
		KSO-G02:9C
		LS-G02:2C
		LS-G02:44C
		LS-G02:4C
		LS-G02:66C
		LS-G02:7C
		LS-G02:8C
		LS-G02:9C
	HAWE Hydraulik	D
		G
		L
		X
	Kawasaki	DE10:205
		DE10:207
		DE10:209
		DE10:210

Block	Manufacturer	Component
		DE10:212
		DE10:213
		DE10:217
		DE10:221
		DE10:222
		DE6:205
		DE6:207
		DE6:210
		DE6:212
		DE6:213
		DE6:217
		DE6:221
	Rexroth Bosch Group	WE10:E
		WE10:G
		WE10:J
		WE10:L
		WE10:M
		WE10:Q
		WE10:T
		WE10:U
		WE10:V
		WE10:W
		WE6:E
		WE6:F
		WE6:G
		WE6:J
		WE6:L
		WE6:M
		WE6:P
		WE6:Q
		WE6:T
		WE6:U
		WE6:V
		WE6:W
		5-.WE:E
		5-.WE:G

Block	Manufacturer	Component
		5-.WE:J
		5-.WE:L
		5-.WE:Q
		5-.WE:R
		5-.WE:U
		5-.WE:W
	Yuken	DSG-01:3C10
		DSG-01:3C11
		DSG-01:3C12
		DSG-01:3C2
		DSG-01:3C3
		DSG-01:3C4
		DSG-01:3C40
		DSG-01:3C60
		DSG-01:3C9
		DSG-03:3C10
		DSG-03:3C11
		DSG-03:3C12
		DSG-03:3C2
		DSG-03:3C3
		DSG-03:3C4
		DSG-03:3C40
		DSG-03:3C5
		DSG-03:3C60
		DSG-03:3C9
		S-DSG-01:3C2
		S-DSG-01:3C4
		S-DSG-03:3C2
		S-DSG-03:3C4

Request a Pre-Parameterization

To request a new component pre-parameterization, visit https://www.mathworks.com/products/simscape/model-request.html.

Modeling Fluid Dynamic Systems

Block Equations and the Simscape Numerical Scheme

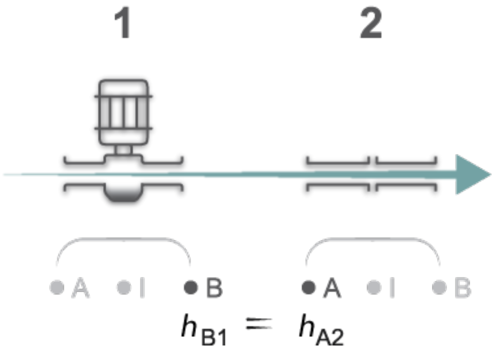
System Equations

Simscape Fluids blocks are a combination of differential equations, which represent the one-dimensional component dynamics, and algebraic equations, which represent the mass and energy continuity over the component. The system of equations, which comprises all blocks over all Simscape domains in the model, are assembled into an exact square matrix that is solved iteratively at each time step until convergence. The number of equations being solved and any zero-crossing events influence the speed and quality of model convergence. For more information on how Simscape software assembles and solves the governing equations of a model, see “How Simscape Simulation Works”.

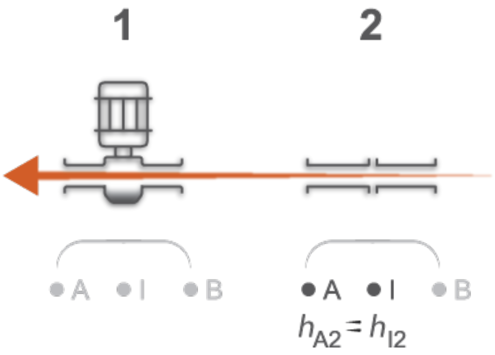
Solving the Network: The Simscape Numerical Scheme

Calculations are computed at each node of your fluid network, such as at block ports, connector intersections, and at internal points in dynamic components. To solve a network, the fluid properties at each node propagate according to the *upwind* numerical scheme. This means that the fluid properties at a given node are calculated based on the fluid properties at that node at the previous solver iteration and the current value of the fluid properties that precede it in space. Flow into an element is considered positive.

The following diagram shows two representations of blocks. Each block has a port A and B and an internal node I. The specific total enthalpy h_{A2} at port A in block 2 is calculated according to the flow direction and is based on the specific total enthalpy, h_{B1}, at port B of block 1:



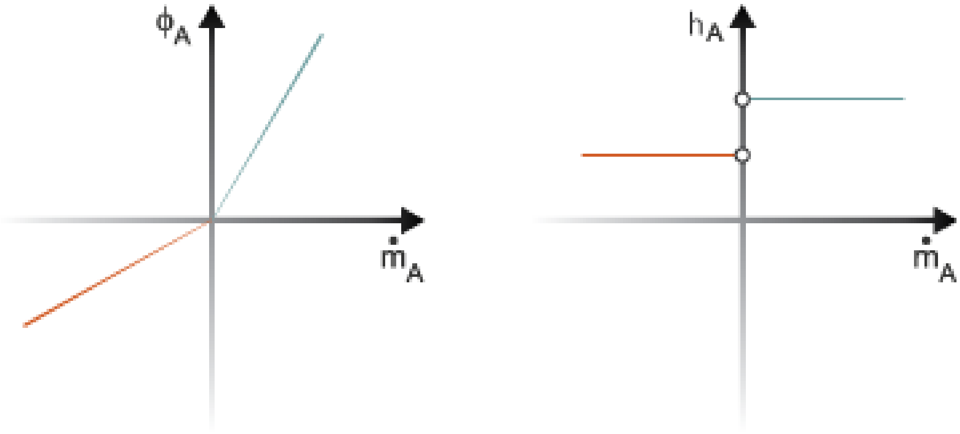
In the same configuration, flow in the reverse direction would mean that the specific total enthalpy at port A of block 2 is calculated based on the specific total enthalpy of the internal node, I, of the same component:



This means that you may observe differences in measured variable values depending on the flow direction and interrogated node.

Smoothing in the Simscape Numerical Scheme

In a flow reversal region, as defined by the total energy flow rate and block thresholds, an energy flow rate without smoothing results in a jump discontinuity of the specific total enthalpy:



Upwind Scheme Without Smoothing

This can result in differences in measured variables between adjacent nodes and may influence the solver performance.

To avoid these discontinuities, smoothing is applied to the calculations and fluid values are calculated in terms of energy flows. For example, in the thermal liquid domain, the smoothed energy flux at A is the sum of the energy flow rate and the specific total enthalpy conduction between port A and internal node I. Conduction contributes to numerical smoothing while contributing a negligible quantity to the energy flow rate under nominal block conditions.

Quasi-steady components also account for smoothing to the energy flow rate. Although the block itself does not have an internal volume, the difference between internal enthalpy and enthalpy at the block ports is still calculated. The total energy flow rate at each node is determined by the fluid flow direction:

$$\Phi_A = G(h_A - h_I) + \begin{cases} \dot{m}_A h_A & \dot{m}_A > 0 \\ \dot{m}_A h_I & \dot{m}_A \le 0 \end{cases},$$

where:

- Φ_A is the energy flow rate through port **A** and conduction between port **A** and the internal node **I**.
- G is the thermal conductance coefficient, which for thermal liquid and moist air domains is calculated as:

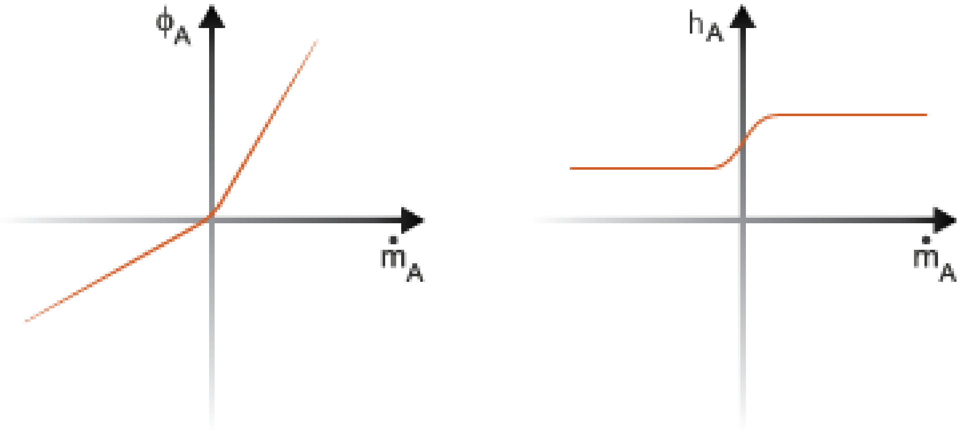
  $$G = \frac{kS}{c_v L},$$

  where:

  - k is the thermal conductivity.
  - S is the port cross-sectional area.
  - c_v is the fluid specific heat.
  - L is the characteristic length between the port and internal node.

For two-phase and gas domains, G is a general tuning parameter and is set by the threshold parameters defined for the block. For example, increasing the value of the **Mach number for flow reversal** parameter in a Two-Phase library block increases the amount of smoothing applied to the energy flow. Similarly, increasing the value of the **Dynamic pressure threshold for flow reversal** parameter in a Gas library block increases the region of smoothing applied.

In a flow reversal region, as defined by the total energy flow rate and block thresholds, the specific total enthalpy changes according to a gradual hyperbolic tangential:



Upwind Energy Scheme with Smoothing

See Also

"How Simscape Simulation Works"

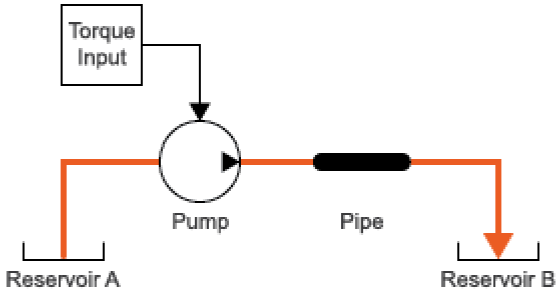
More About

- "Important Concepts and Choices in Physical Simulation"
- "Making Optimal Solver Choices for Physical Simulation"

Model a Thermal Liquid Pipeline

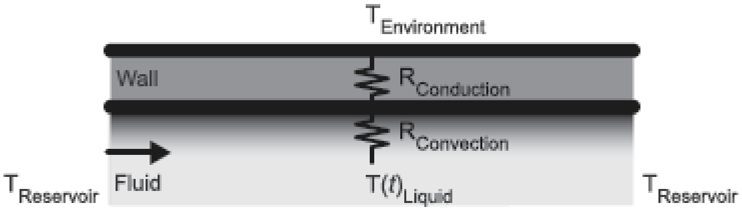
In this section...

This tutorial shows how to model a pump-driven thermal liquid pipeline. The pipeline system comprises a pump, a pipe, and two thermal liquid reservoirs. The pump generates power and the pipe transfers fluid between the reservoirs. The reservoirs act as pressure and temperature boundary conditions for the pipeline system. Note that "thermal" refers to the properties of the solid system, while "thermal liquid" refers to the properties of the fluid system.



Thermal Liquid Pipeline Schematic

The reservoir conditions are fixed at room temperature, 293.15 K, and atmospheric pressure, 0.101325 MPa. The outer pipe surface temperature is fixed at a temperature of 275 K. The pipe fluid volume is initially at room temperature but, through the combined effects of heating due to friction and conductive cooling, gradually shifts toward a new steady-state value. The thermal resistance of the pipe is the combination of these convective and conductive resistances.

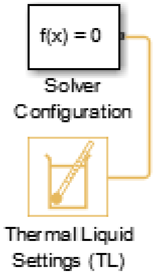


Pipe Thermal Resistances

A Pipe (TL) block models the convective thermal resistance at the inner pipe surface and the heating due to friction in the thermal liquid. A Conductive Heat Transfer block models the conductive thermal resistance between the inner and outer pipe surfaces.

Start a New Model

1. At the MATLAB command prompt, enter `ssc_new`. MATLAB opens the Simscape model template. This template provides a starting point for your Simscape Fluids model, and includes a Solver Configuration block, a Simulink-PS converter block, a PS-Simulink converter block, and a Scope block, which enables you to view the results of your simulation. You will not need the Simulink-PS Converter block in this example.
2. Save the model often as you add new blocks.
3. In the **Simulation** tab, click **Library Browser**. In the pop-up window, select **Simscape** > **Foundation Library** > **Thermal Liquid** > **Utilities** library and drag a Thermal Liquid Settings (TL) block to the model canvas. This block defines the physical properties of the thermal liquid, including its viscosity, thermal conductivity, and bulk modulus.
4. Connect the Solver Configuration and Thermal Liquid Settings (TL) blocks as shown in the figure. The Solver Configuration block provides Simscape solver settings for your model. Each physical network requires a Solver Configuration block.



Model the Pipeline

1. Open the **Library Browser**. Select **Simscape** > **Foundation Library** > **Thermal Liquid** > **Elements** library, drag two Reservoir (TL) blocks to the model canvas. The reservoir blocks set the pressure and temperature boundary conditions for the pipeline model.
2. From the **Simscape** > **Fluids** > **Thermal Liquid** library, insert these blocks.

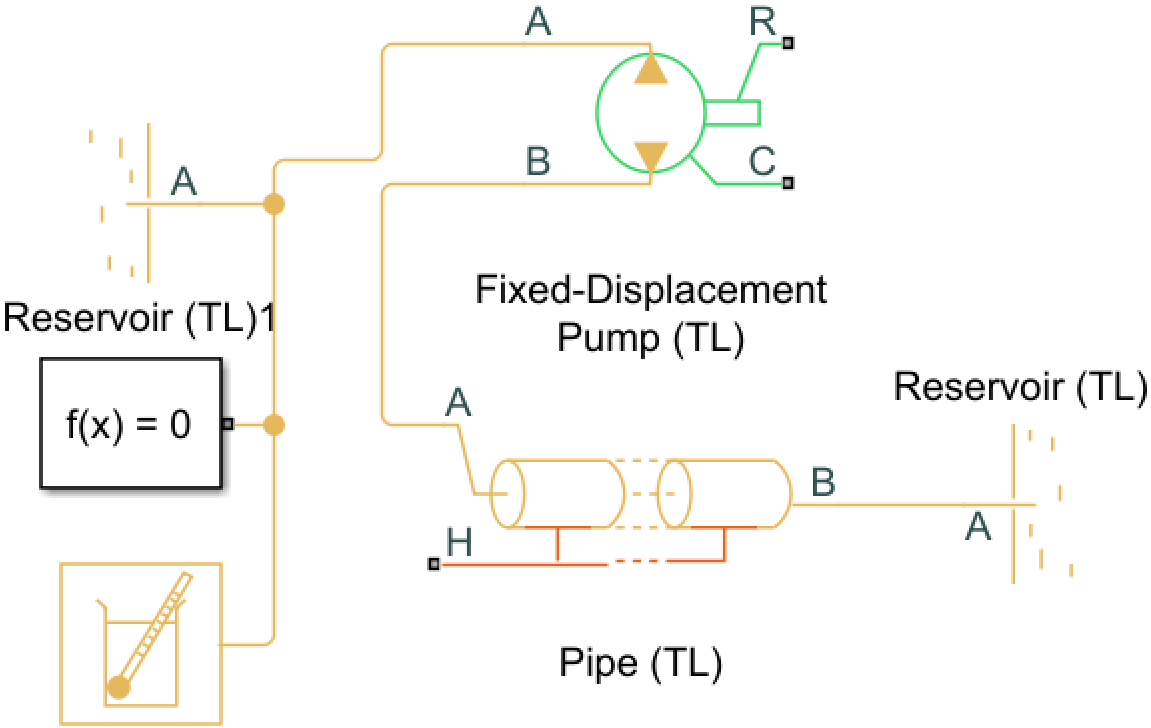
Block	Library	Purpose
Pipe (TL)	**Pipes & Fittings**	Conduit for thermal liquid between reservoirs
Fixed-Displacement Pump (TL)	**Pumps & Motors**	Power generation source

3. Connect the blocks and verify your connections in the figure below. Note that blocks may be rotated.

   a. Port **A** of the Pipe (TL) block connects to port **B** of the Fixed-Displacement Pump (TL) block.

   b. Port **B** of the Pipe (TL) block connects to one Reservoir (TL) block. This serves as a thermal liquid sink.

   c. Port **A** of the Fixed-Displacement Pump (TL) block connects to the second Reservoir (TL) block, the Solver Configuration block, and the Thermal Liquid Settings (TL) block. The Reservoir block serves as a thermal liquid source.



The component colors indicate the physical domains they represent: yellow for Thermal Liquid, red for Thermal, and green for Mechanical Rotational. The pump and pipe blocks are multi-domain blocks that you can use to interface different physical domains.

4 In the Pipe (TL) block, set the **Pipe length** parameter to `1000` m. This length results in observable pressure losses due to dissipation and heat transfer by friction at the pipe walls.

Introduce Thermal Conduction

1 Open the **Library Browser** and insert these blocks from the **Simscape** > **Foundation Library**.

Block	Library	Purpose
Conductive Heat Transfer	**Thermal** > **Thermal Elements**	Models the conductive heat losses through the pipe wall to the surrounding environment
Thermal Reference	**Thermal** > **Thermal Elements**	Provides a temperature reference for the physical signal input
Controlled Temperature Source	**Thermal** > **Thermal Sources**	Acts as an ideal energy source and maintains temperature regardless of heat flow rate
PS Constant	**Physical Signals** > **Sources**	Specifies environmental temperature

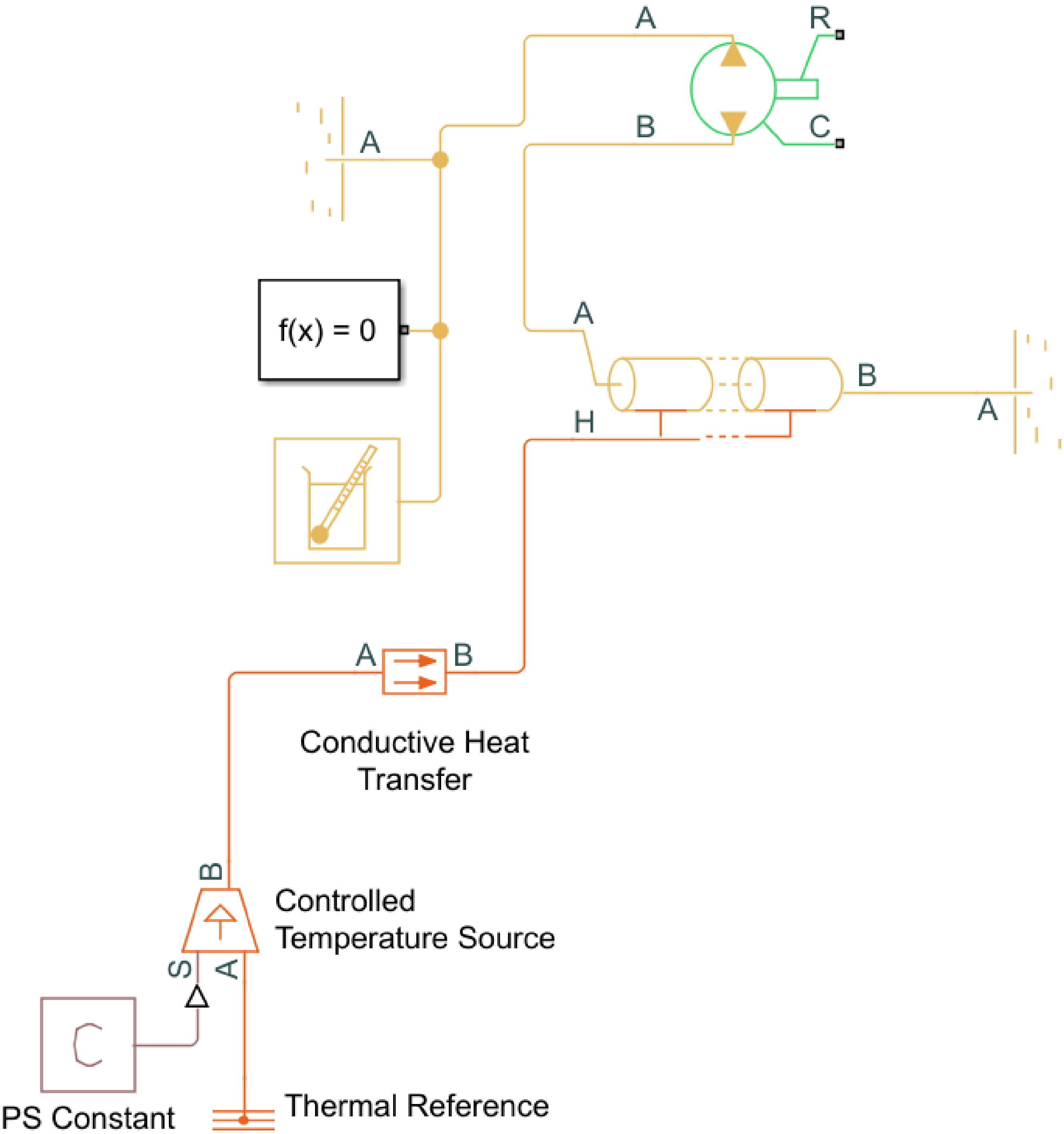
2 Connect the blocks and verify your connections in the figure below. Note that blocks may be rotated.

a Connect port **B** of the Conductive Heat Transfer block to port **H** of the Pipe (TL) block.

b Connect port **B** of the Controlled Temperature Source block to port **A** of the Conductive Heat Transfer block.

c Connect the PS Constant block to port **S** of the Controlled Temperature Source block.

d Connect the Thermal Reference block to port **A** of the Controlled Temperature Source block.



3 In the PS Constant Block Parameters Window, set the **Constant** parameter to `275 K`. This is the environment temperature.

4 In the Conductive Heat Transfer block, set the **Area** parameter to `pi*0.1128*1000` m ^{2}. This is the pipe surface area in terms of the hydraulic diameter, `0.1128` m, and length, `1000` m. The default **Thermal conductivity** value is for copper, `401 W/(mK)` .

Model the Pump Torque

1 Open the **Library Browser** and insert these blocks from **SimscapeFoundation Library**.

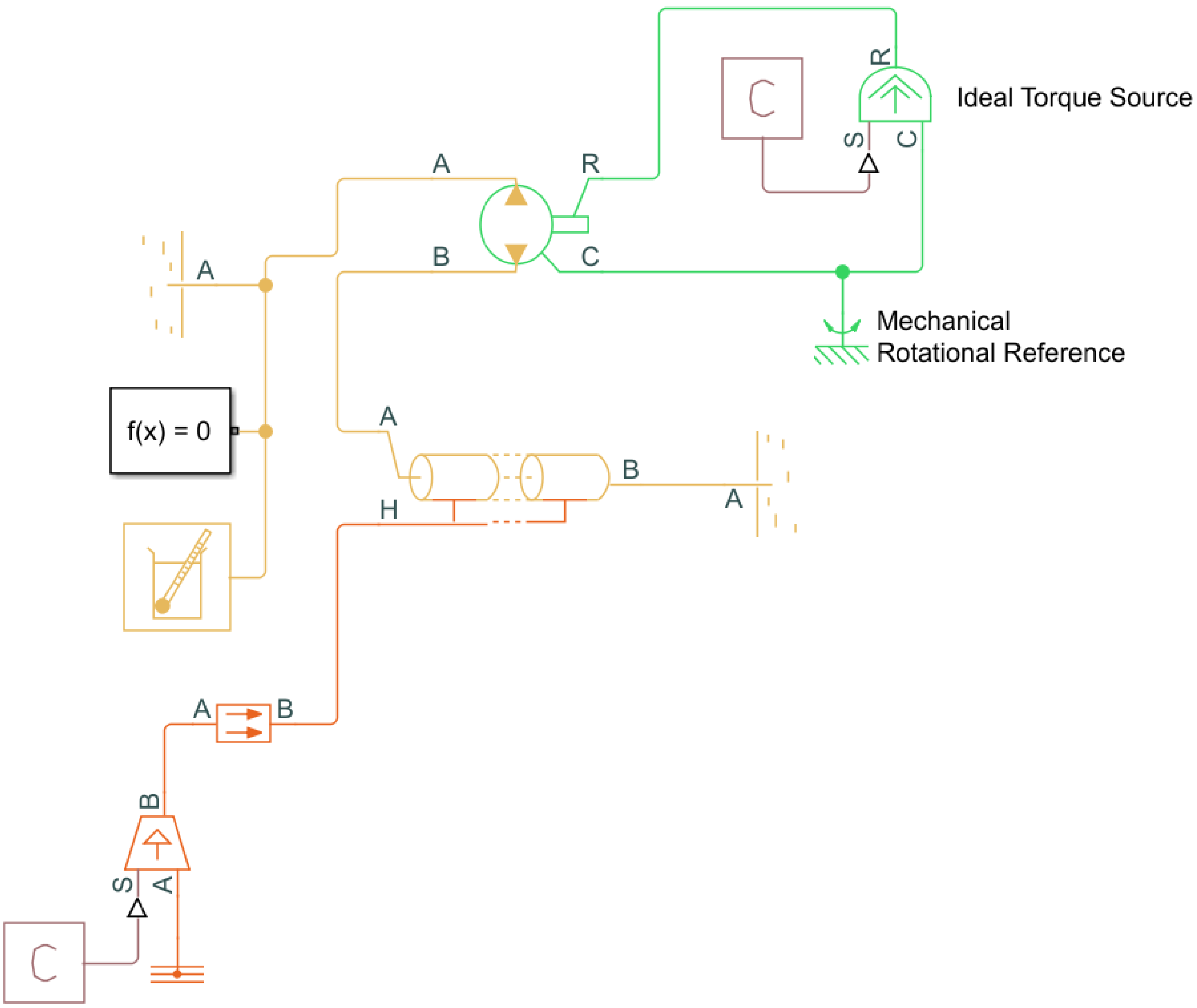
Block	Library	Purpose
Ideal Torque Source	**Mechanical** > **Mechanical Sources**	Sets the pump torque rate as a physical signal
Mechanical Rotational Reference	**Mechanical** > **Rotational Elements**	Acts as a reference for the Ideal Torque Source and maintains a physical signal at a specified value
PS Constant	Double-click on the browser white space, type PS Constant and hit **Enter**	Specifies the pump torque

2 Connect the blocks and verify your connections in the figure. Note that blocks may be rotated.

a Connect port **R** of the Fixed-displacement Pump (TL) block, which represents the rotating pump shaft, to port **R** of the Ideal Torque Source block.

b Connect port **C** of the Ideal Torque Source block, which represents pump casing, to port **C** of the Fixed-Displacement Pump (TL) block and the Mechanical Rotational Reference block.

c Connect port **S** of the Ideal Torque Source block to the PS Constant block.



3 In the PS Constant Block Parameters window, set the **Constant** parameter to `50 N*m`.

Add Thermal Liquid Sensors

1 Open the **Library Browser** and add these blocks to the model.

Block	Library	Purpose
Flow Rate Sensor (TL)	**Simscape > Foundation Library > Thermal Liquid > Sensors**	Measures the Through variables of the Thermal Liquid domain and returns a physical signal
PS-Simulink Converter	**Simscape > Utilities**	Transforms a physical signal into a Simulink signal

Block	Library	Purpose
Scope	**Simulink > Sinks**	Plots Simulink signals

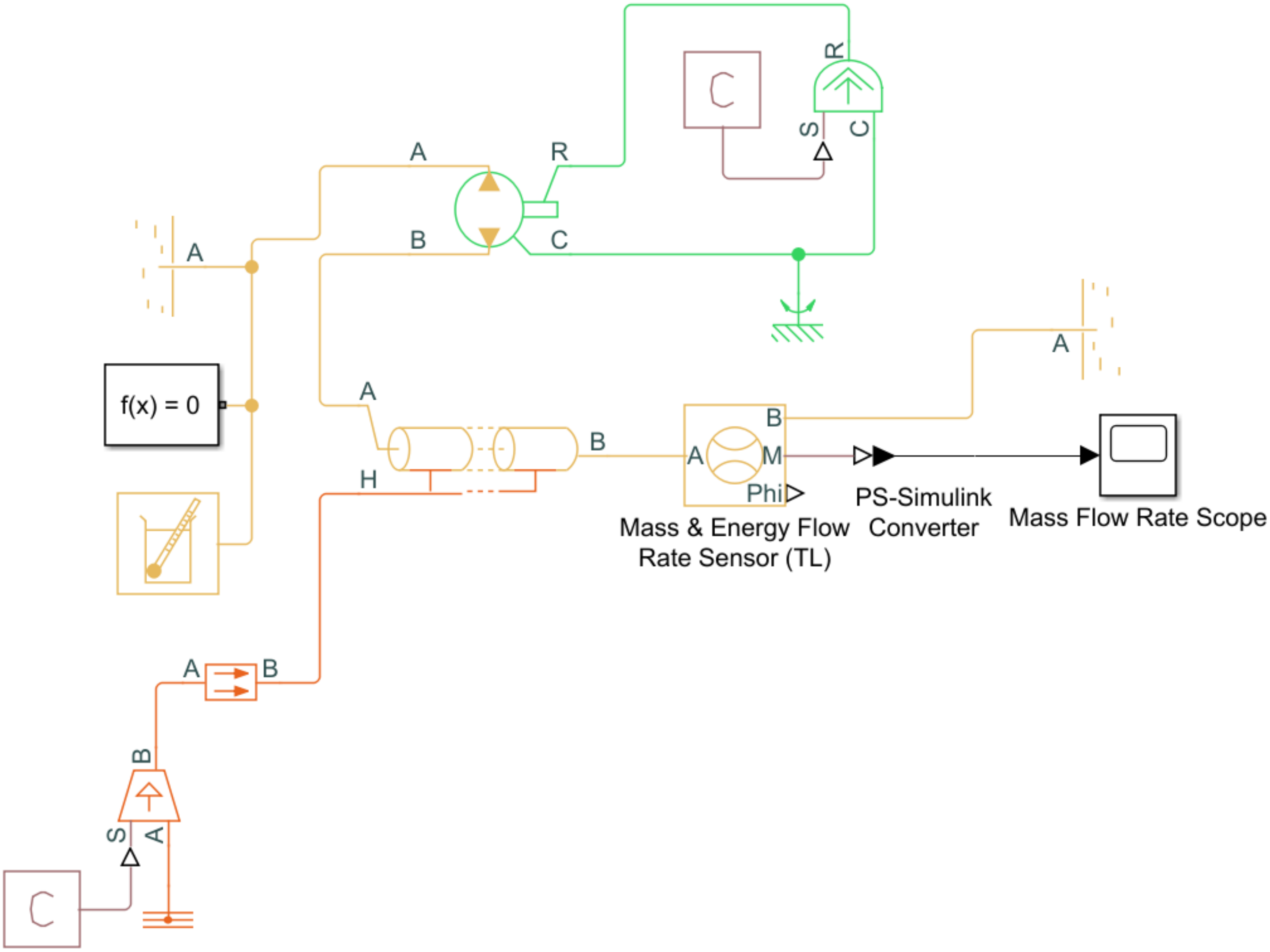
2 Connect the blocks and verify your connections in the figure. Note that blocks may be rotated.

a Your Pipe (TL) block is connected to a Reservoir from Section 1.5.4, which provides pressure and temperature boundary conditions for the system. You will now add a way to measure pipe flow. Disconnect port **B** of the Pipe (TL) block from the Reservoir block and connect it to port **A** of the Mass & Energy Flow Rate Sensor (TL) block.

b Connect port **B** of the Mass & Energy Flow Rate Sensor block to the Reservoir block.

c Connect port **M** of the Mass & Energy flow Rate Sensor (TL) block to the PS-Simulink Converter block.

d Connect the PS-Simulink Converter block to the Scope block. Click the Scope block and name it "Mass Flow Rate Scope."



3 Open the **Library Browser** and add these blocks to the model. You will also use the PS-Simulink Converter and Scope blocks that were included when you opened your new model.

Block	Library	Purpose
Pressure & Temperature Sensor (TL)	**Simscape > Foundation Library > Thermal Liquid > Sensors**	Measures the Across variables of the Thermal Liquid domain and returns a physical signal
Absolute Reference (TL)	**Simscape > Foundation Library > Thermal Liquid > Elements**	Provides a control reference where pressure and temperature are equal to zero

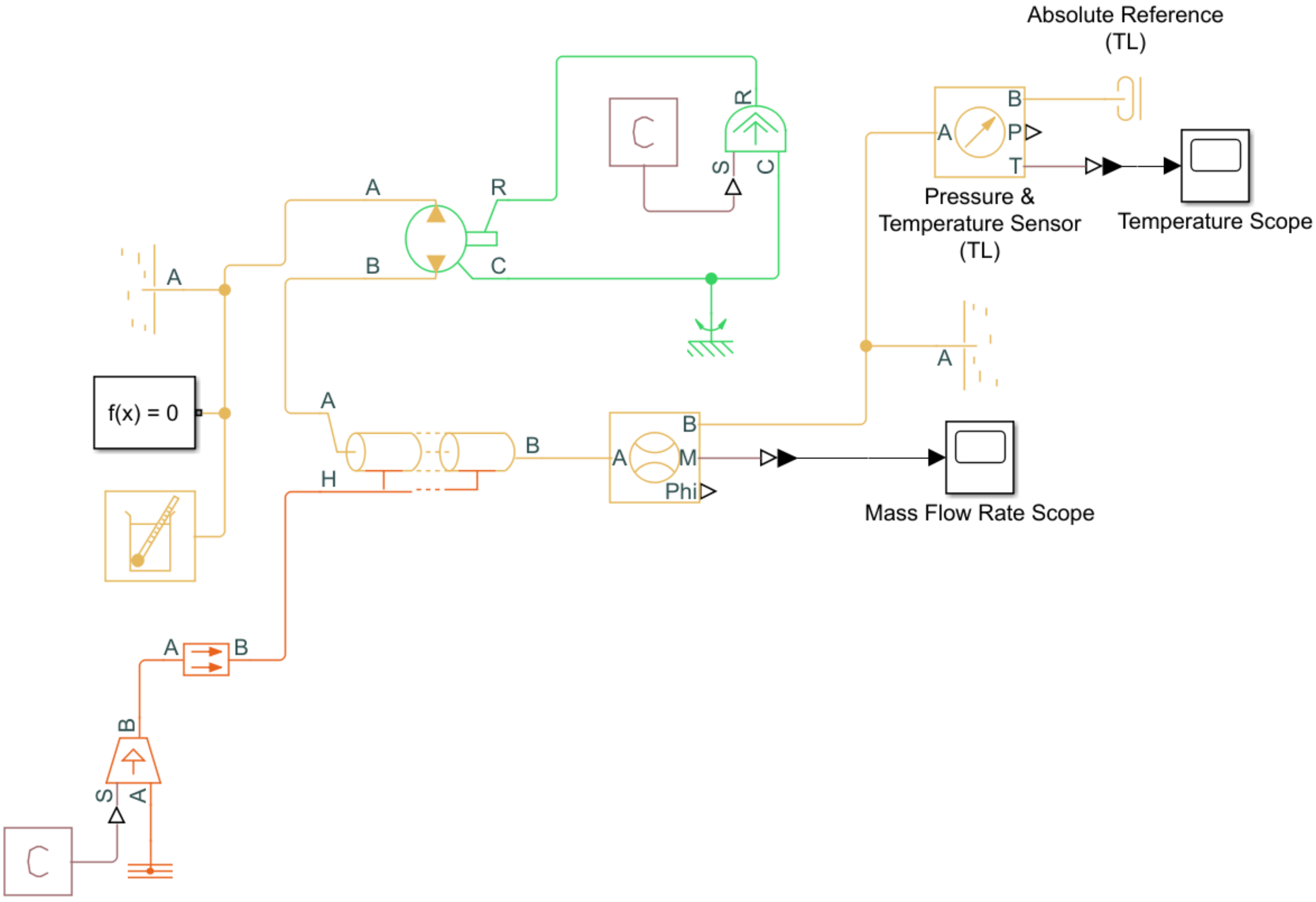
4 Connect the blocks and verify your connections in the figure. Note that blocks may be rotated.

a Create a branch from port **B** of the Mass & Energy Flow Rate Sensor (TL) block and connect it to port **A** of the Pressure & Temperature Sensor (TL) block.

b Connect port **B** of the Pressure & Temperature Sensor(TL) block to the Absolute Reference block.

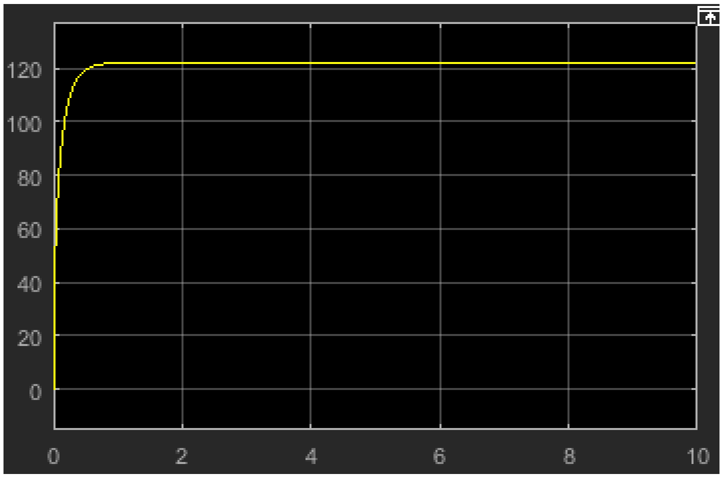
c Connect port **T** of the Pressure & Temperature Sensor (TL) block to the PS-Simulink Converter block.

d If the PS-Simulink Converter is not already connected to the remaining Scope block, connect the two. Click this Scope block and name it "Temperature Scope."

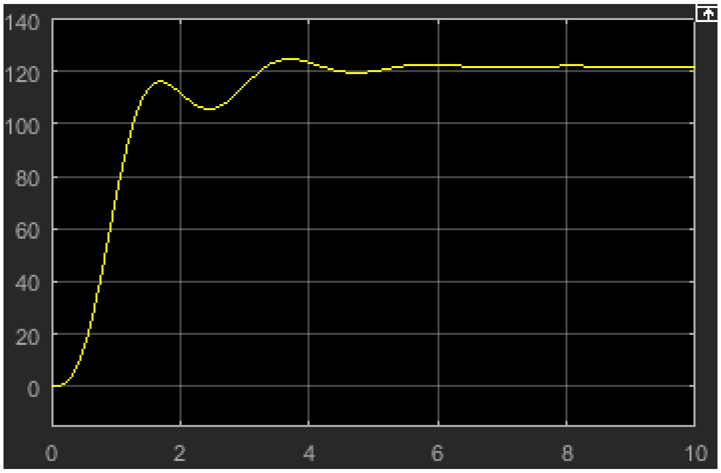


Simulate the Pipeline Model

The Simscape model template specifies suitable solver settings for this model. You can run the simulation by opening the **Simulation** tab and clicking **Run** > **Run**. Once the model has compiled, open the Mass Flow Rate Scope. Your plot should show a quick change in flow rate as the thermal liquid transitions from a state of rest to a new steady-state velocity.



You can experiment with other block settings and input signals. Try setting the Pipe (TL) block **Fluid inertia** parameter to `On`, then run your model. The figure shows the combined effects of fluid dynamic compressibility and fluid inertia on the mass flow rate plot.

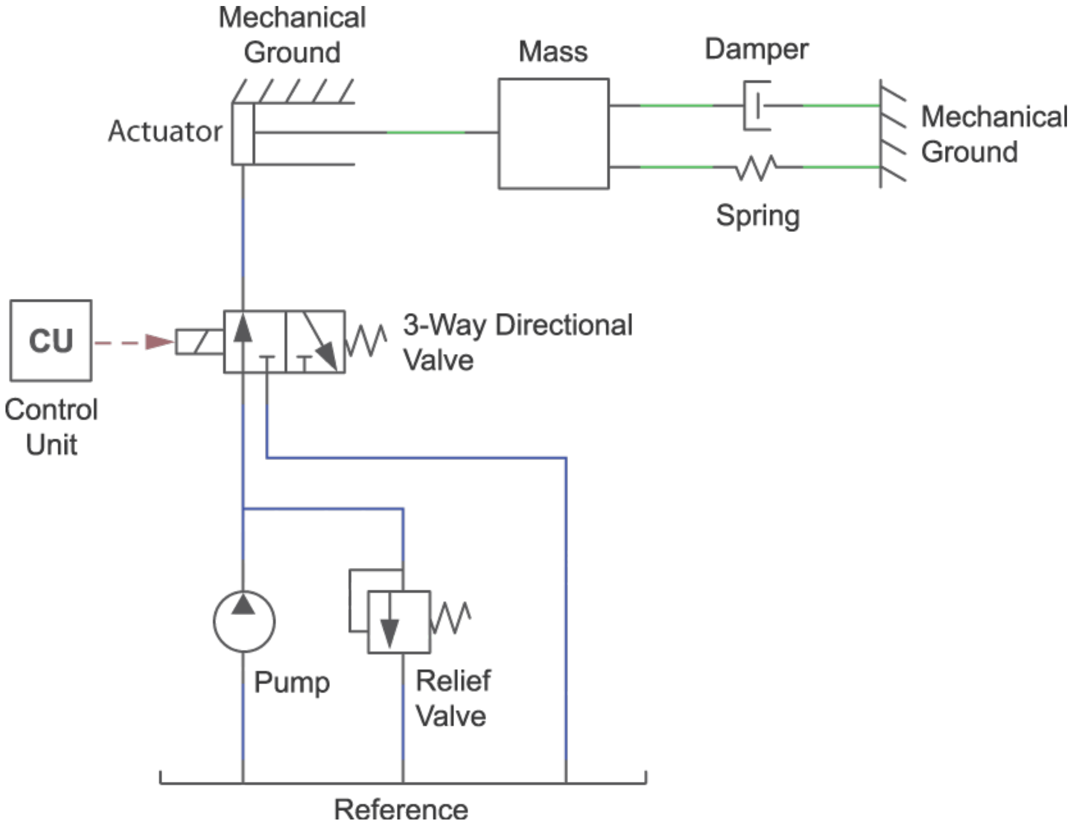


Model an Isothermal Actuation System

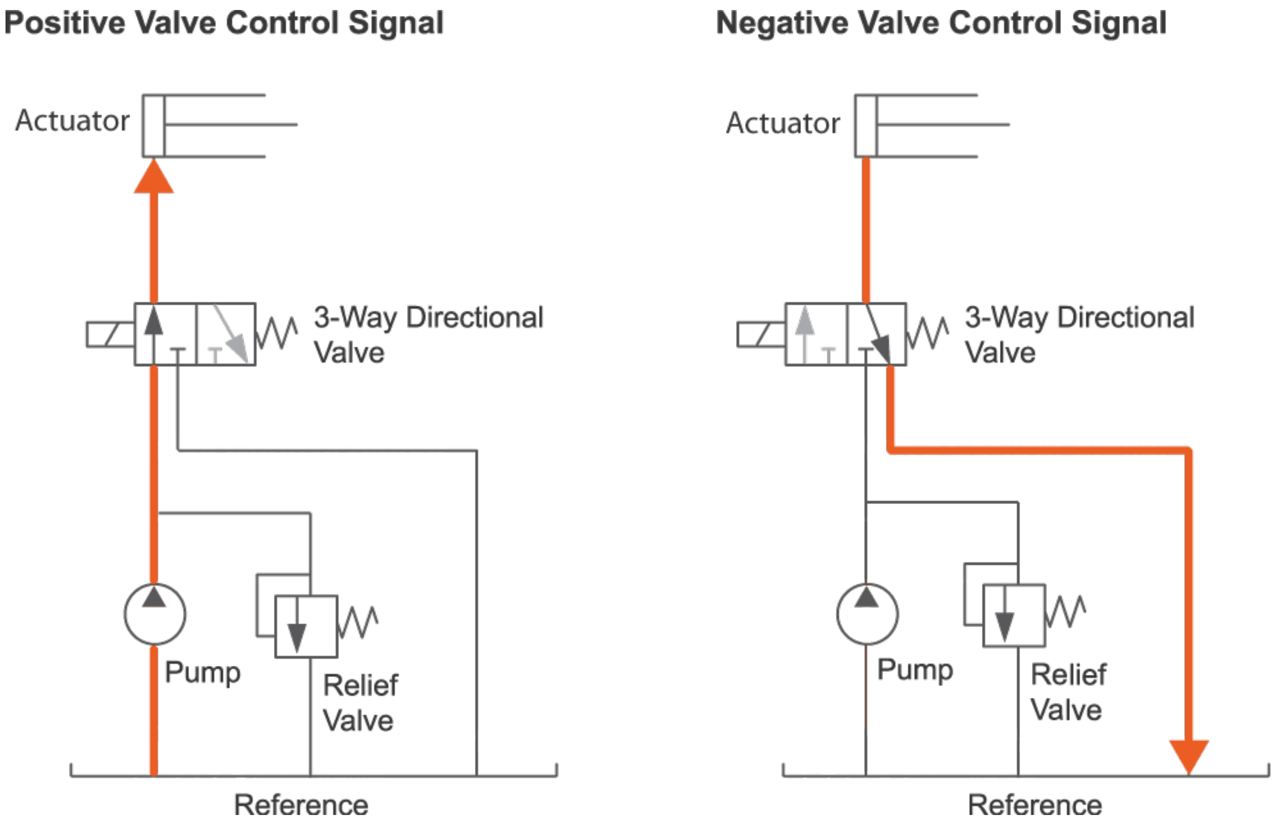
In this section...

Model Schematic

This tutorial shows how to model an isothermal actuation system using Simscape Fluids blocks. The system consists of a pump, which is modeled by an idealized pressure source, a control unit, a flow regulation valve, and a mechanical load.



The control unit represents an incoming signal for actuation. The three-way valve represents flow regulation into the actuator. Depending on the position received from the control unit, the three-way valve allows fluid to enter the actuator chamber or redirects it to the environment ("Reference"). In the 3-Way Directional Valve (IL) block, there is a single spool that adjusts the two valve positions - a flow path between ports **P** and **A** and a flow path between ports **A** and **T**. A change in spool position results in a change in flow path through the valve.

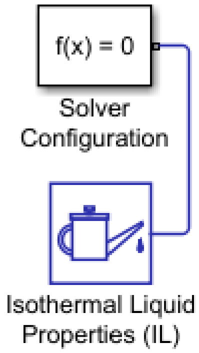


Directional Valve Positions

A relief valve vents to the environment if the pressure between the pump and valve becomes too high.

Start a New Model

1. At the MATLAB command prompt, enter `ssc_new`. MATLAB opens the Simscape model template. This template provides a starting point for your Simscape Fluids model. You will use blocks from both the Simscape Foundation and Simscape Fluids product libraries.
2. From the Simulation tab, click on the **Library Browser**. From the **Simscape** > **Foundation Library** > **Isothermal Liquid** > **Utilities** library, drag an Isothermal Liquid Properties (IL) block to the model canvas. Use this block to define the physical properties of the network fluid, including its density, viscosity, and bulk modulus. Use the Isothermal Liquid Predefined Properties (IL) block from the **Fluids** > **Isothermal Liquid** > **Utilities** library to select a fluid with these properties already defined.
3. Connect the Solver Configuration block to the Isothermal Liquid Properties (IL) block as shown in the figure. The Solver Configuration block provides Simscape solver settings for your model. See “Setting Up Solvers for Physical Models” for more information on solver selection.



Each topologically distinct physical network requires a solver block.

Model the Actuator Subsystem

1 From the **Simscape** > **Foundation Library**, drag these blocks to the model canvas.

Block	Sublibrary
Pressure Source (IL)	**Isothermal Liquid** > **Sources**
Reservoir (IL)	**Isothermal Liquid** > **Elements**

Alternatively, double-click on the model canvas and type the block name. Click on the block name in the drop-down list that appears to add the block to the canvas.

Specify the model configuration in each block dialog by double-clicking on the block icon:

2 A pressure source represents a constant-pressure pump. In the Pressure Source (IL) block dialog, set **Source type** to `Constant`. Input `10e5` Pa for **Pressure differential**.

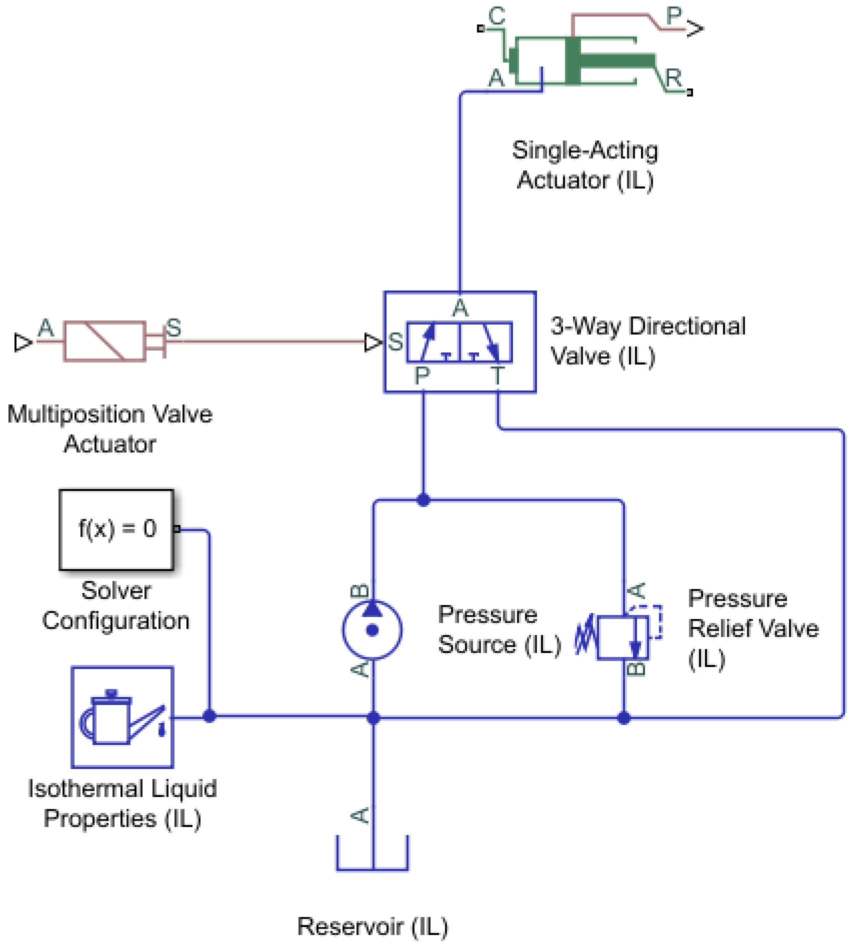
3 A reservoir sets the environmental boundary conditions. In the Reservoir (IL) block, set **Reservoir pressure specification** to `Specified pressure` and **Reservoir pressure** to `0.2` MPa.

4 From the **Simscape** > **Fluids** library, drag these blocks to the model canvas.

Block	Sublibrary
Single-Acting Actuator (IL)	**Isothermal Liquid** > **Actuators**
3-Way Directional Valve (IL)	**Isothermal Liquid** > **Valves & Orifices** > **Directional Control Valves**
Multiposition Valve Actuator	**Isothermal Liquid** > **Valve Actuators**
Pressure Relief Valve (IL)	**Isothermal Liquid** > **Valves & Orifices** > **Pressure Control Valves**

The actuator converts the power generated by the pressure source into a translating force. The directional valve and valve actuator regulate the flow to the actuator.

5 Connect the blocks as shown in the figure. Note that the 3-Way Directional Valve (IL), Pressure Source (IL), and Pressure Relief Valve (IL) blocks are grounded by their connection to the Reservoir (IL) block.


C
P
A
R
Single-Acting Actuator (IL)
A
S
A
Multiposition Valve Actuator
S
A
P
T
3-Way Directional Valve (IL)
f(x) = 0
Solver Configuration
B
A
Pressure Source (IL)
A
B
Pressure Relief Valve (IL)
Isothermal Liquid Properties (IL)
A
Reservoir (IL)

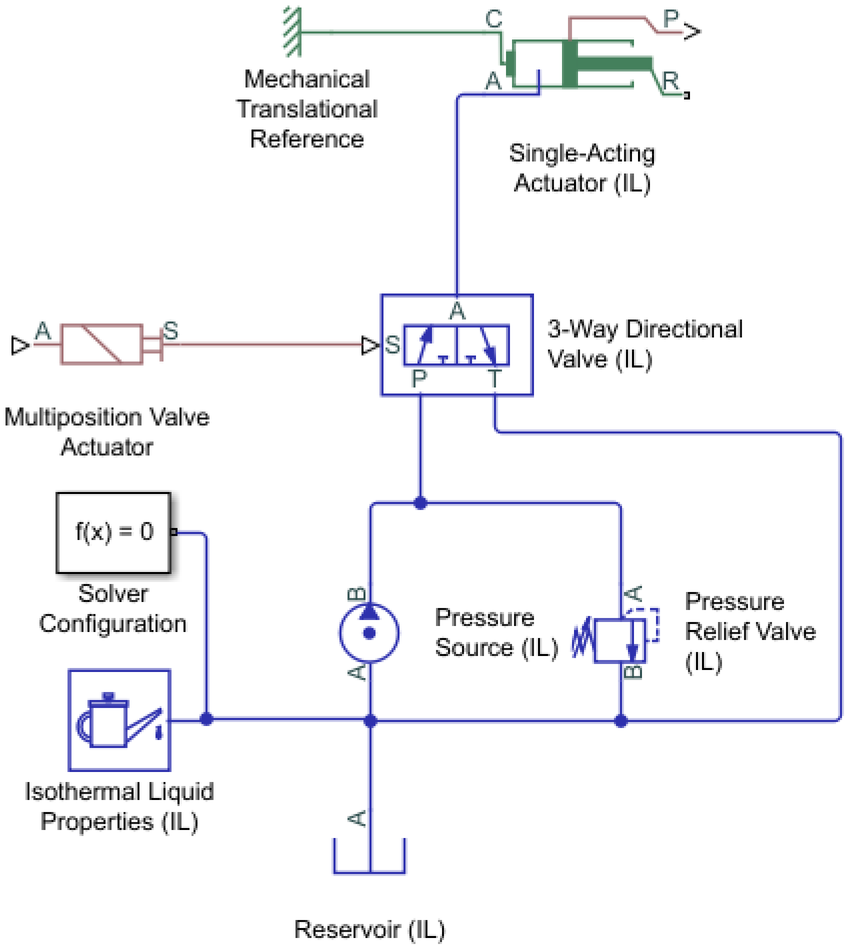
6 Set the operating pressures of the Pressure Relief Valve (IL). In the **Set pressure differential** parameter, enter `50e5` Pa. This is the pressure at which the valve begins to open. In the **Pressure regulation range** parameter, which describes the operational range of the valve, enter `5e5` Pa. The relief valve maximum pressure is the sum of the set pressure and the regulation range.

7 In the 3-Way Directional Valve (IL) block dialog, on the **P-A Orifice** tab, set **Spool position at maximum P-A orifice area** to `0.01` m. This is the position of the valve spool when the P-A flow path is fully open. On the **A-T Orifice** tab, set **Spool position at maximum A-T orifice area** to `0` m.

The **Spool travel between closed and open orifice** parameter on the **Model parameterization** tab sets the distance over which both orifices open and close. At this value, both orifices are closed.

8 In the Single-Acting Actuator (IL) block, set the **Piston cross-sectional area** to `0.001` m^2.

9 From the **Simscape** > **Foundation Library** > **Mechanical** > **Translational Elements** library, add a Mechanical Translational Reference block and connect it as shown in the figure.



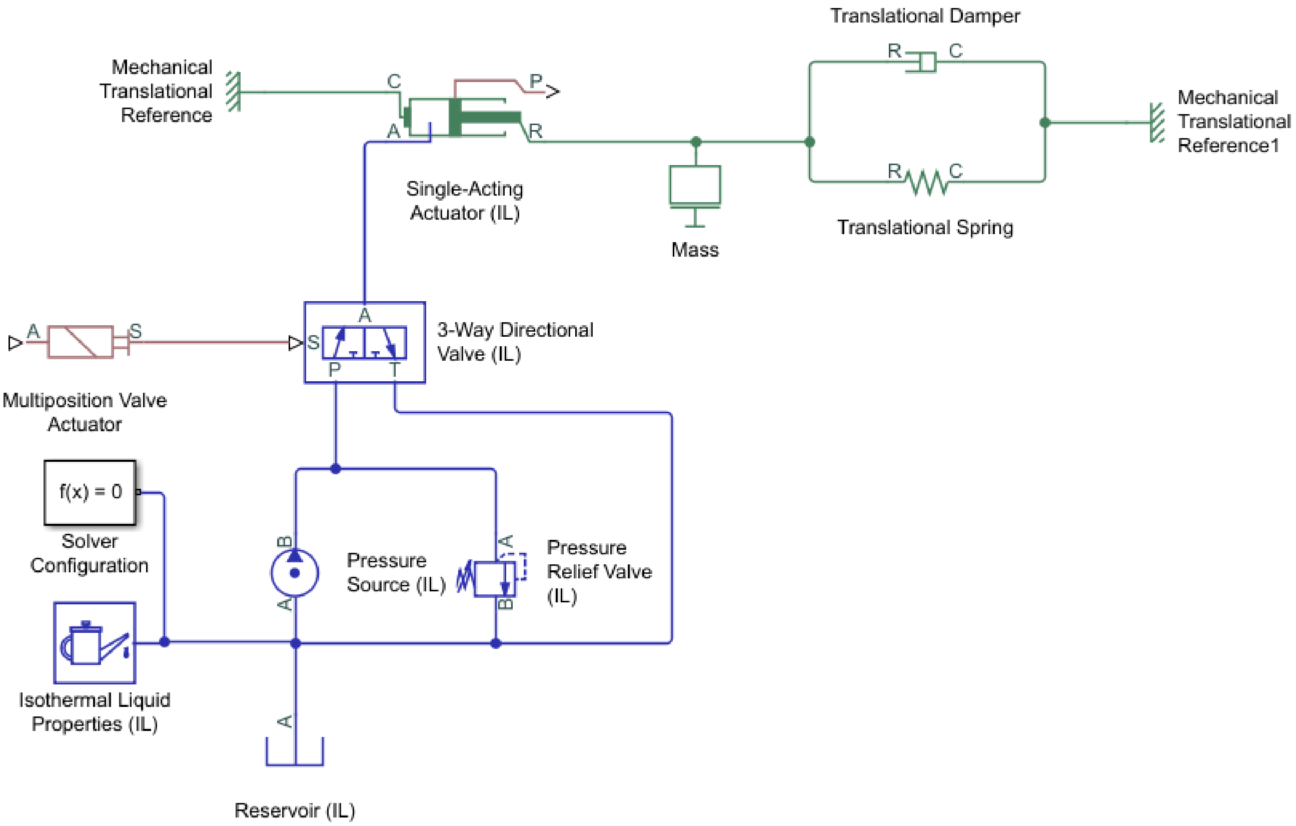
This fixes the mechanical casing of the Single-Acting Actuator (IL), which is represented by port **C**. The actuator shaft, represented by port **R**, moves relative to this reference.

Add a Load Model

1 From the **Simscape** > **Foundation Library** > **Mechanical** > **Translational Elements** library, drag these blocks to the model canvas:

   - Mass
   - Translational Spring
   - Translational Damper
   - Mechanical Translational Reference

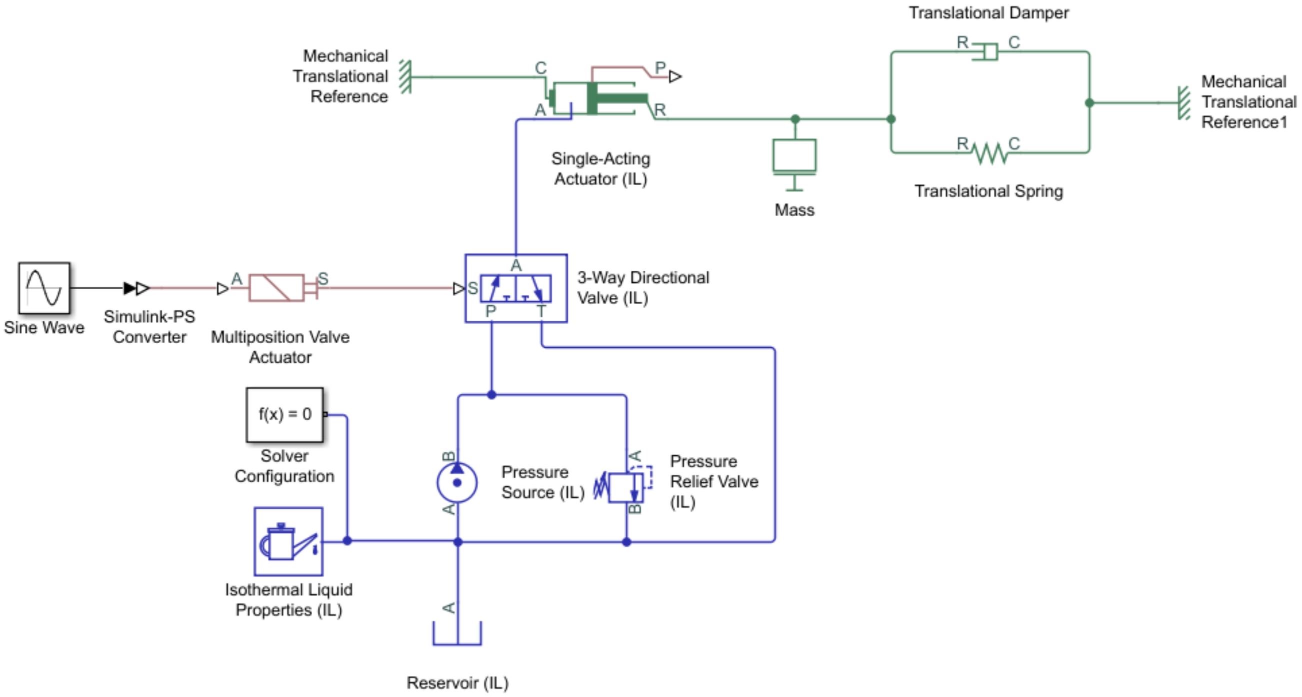
2 Connect the blocks as shown in the figure.


Translational Damper
R
C
Mechanical Translational Reference
C
P
A
R
Mechanical Translational Reference1
R
C
Translational Spring
Single-Acting Actuator (IL)
Mass
A
S
A
S
P
T
3-Way Directional Valve (IL)
Multiposition Valve Actuator
f(x) = 0
Solver Configuration
B
A
Pressure Source (IL)
A
B
Pressure Relief Valve (IL)
Isothermal Liquid Properties (IL)
A
Reservoir (IL)

The mass, spring, and damper elements provide a basic model of a load with forcing, inertia, and damping.

Add a Control System and Monitoring

1. From the **Simulink** > **Sources** library, drag a Sine Wave to the canvas. This represents a varying signal to the control system, which in turn regulates the flow received by the actuator.
2. From the **Simscape** > **Utilities** library, drag one PS-Simulink Converter block. To convert between Simscape physical signals, which are bidirectional, and Simulink signals, which are unidirectional, you will need to use a Simulink-PS Converter or PS-Simulink Converter block.
3. Connect the Sine Wave block, the Simulink-PS Converter block that is already on the canvas, and Multiposition Valve Actuator block in series, as shown in the figure.

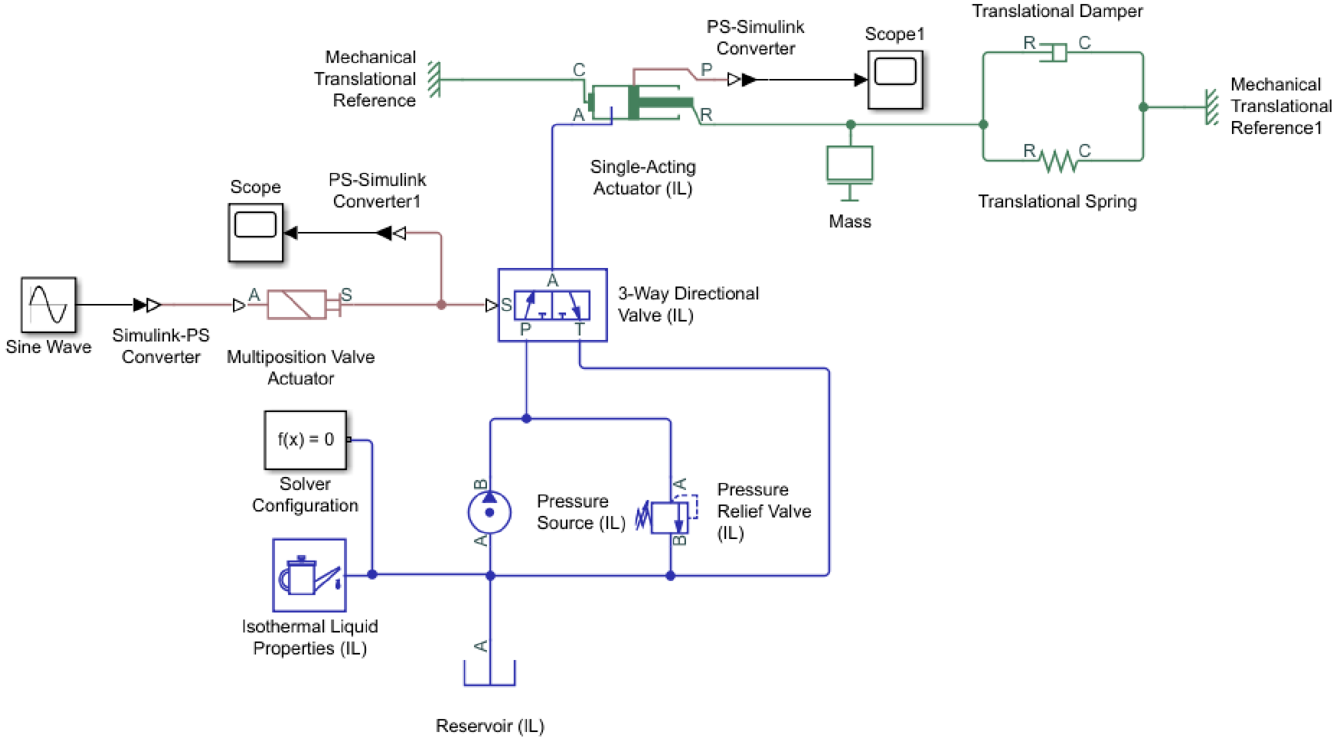

Translational Damper
R
C
Mechanical Translational Reference
C
P
A
R
Mechanical Translational Reference1
Single-Acting Actuator (IL)
R
C
Translational Spring
Mass
Sine Wave
Simulink-PS Converter
A
S
Multiposition Valve Actuator
S
A
P
T
3-Way Directional Valve (IL)
f(x) = 0
Solver Configuration
B
A
Pressure Source (IL)
A
B
Pressure Relief Valve (IL)
Isothermal Liquid Properties (IL)
A
Reservoir (IL)

4 In the Sine Wave block dialog, set the **Amplitude** parameter to 24. This value corresponds to the default nominal signal value of the Multiposition Valve Actuator block.

5 Drag a Scope block from the Simulink Sources library. The Scope block allows you to quickly inspect model results.

6 Connect the Single-Acting Actuator (IL) block, a PS-Simulink Converter block, and a Scope block in series, as shown in the figure.

7 Connect the output side of the Multiposition Valve Actuator block, the other PS-Simulink Converter block, and a Scope block in series, as shown in the figure.



Port **P** of the Single-Acting Actuator (IL) block outputs a physical signal of the actuator shaft position, relative to the **Piston initial displacement**.

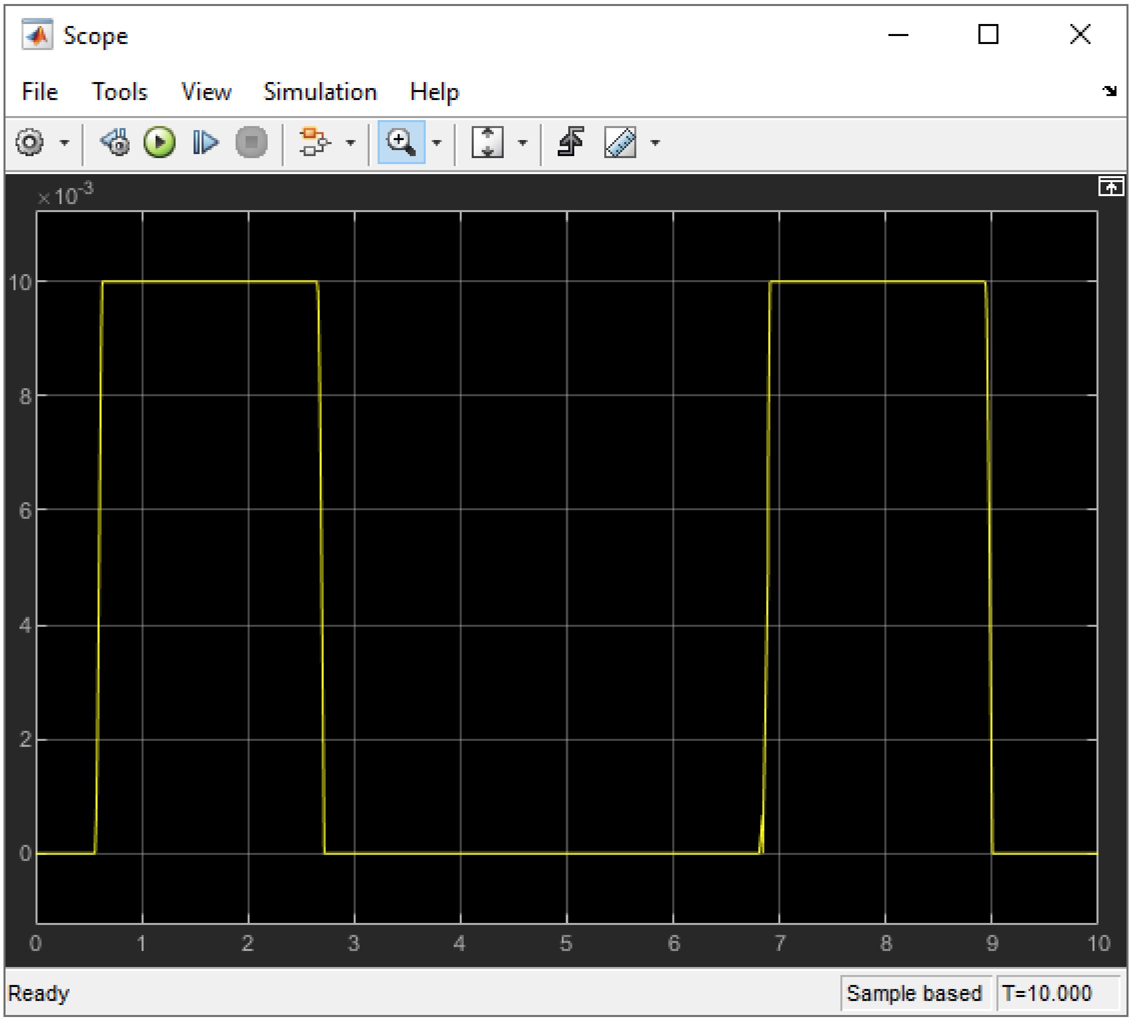
The Multiposition Valve Actuator block outputs a physical signal of the valve opening distance, relative to the **Initial position** and the **Push-pin stroke**.

Run the Simulation

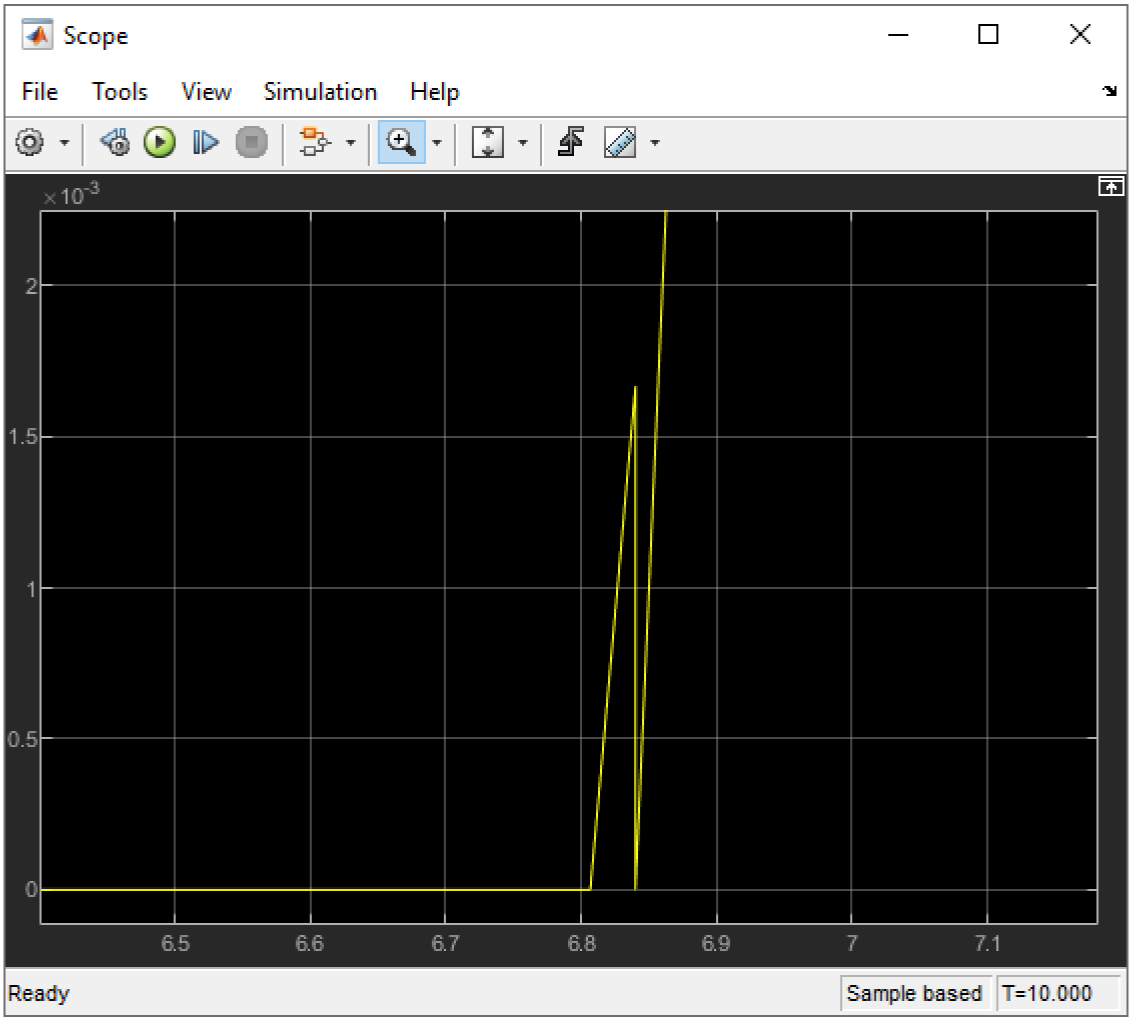
When you run the model, the Simscape software evaluates the model, determines the simulation initial conditions, and simulates the model. To learn more about this process, see “How Simscape Simulation Works”.

1. Click on any block in the model, for example the 3-Way Directional Valve (IL) block. In the **Simscape Block** tab, click **Log Data** and select **Log all Simscape data**. This enables you to explore the simulation variable values in the Results Explorer.
2. Click **Run**.
3. In the Debug tab, select **Simscape** > **Variable Viewer**. The “Variable Viewer” shows the initialization values and priorities of all of the model variables. You can set variable targets and priorities during initialization for blocks with a Variables tab in the block dialog. Verify that the status of all variables is green.
4. You can inspect simulation results in multiple ways. For a quick assessment of the model, use Scope blocks.

   Double-click on the Scope block connected to the Multiposition Valve Actuator. This shows the valve spool position, which is set by the Multiposition Valve Actuator signal.



Zooming in, there is a small spike in the signal at the beginning of the second peak.

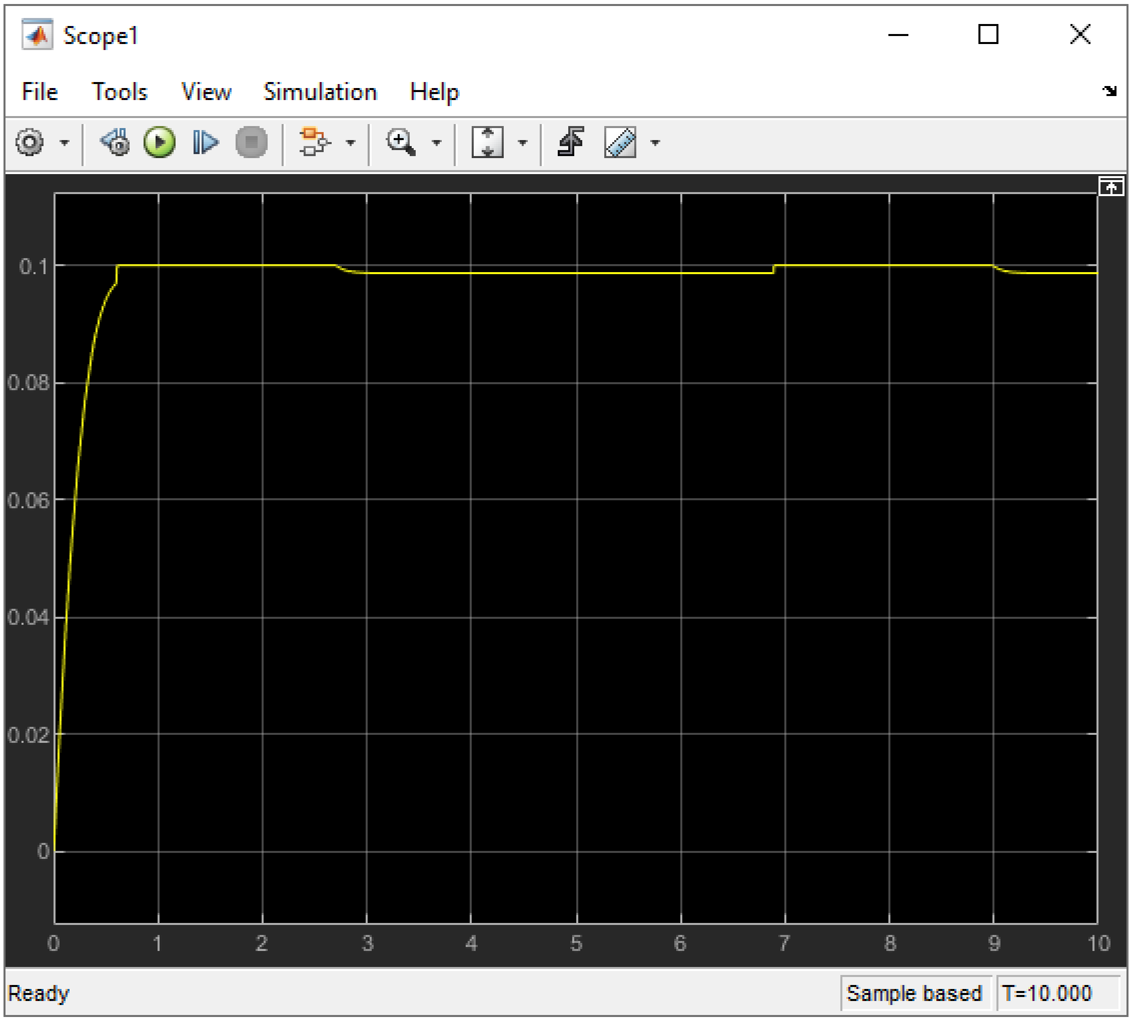


This can be resolved by limiting the size of the solver time step:

- **a** Open Configuration Parameters. On the **Modeling** tab, select **Model Settings** > **Model Settings** and click **Solver** in the left-hand pane.
- **b** Expand the Solver details option.
- **c** Change **Max step size** from `auto` to `0.01`.
- **d** Click **OK**.

This configuration will be used the next time you run the model.

Double-click on the Scope block connected to the Single-Acting Actuator (IL) block. The Scope plot shows the shaft position as a function of the pressure in the actuator chamber.



It appears that the actuator is not fully responding to the signal received from the control unit.

5 The environmental pressure is specified in three locations: in the Isothermal Liquid Properties (IL) block, in the Reservoir (IL) block, and in the Single-Acting Actuator (IL) block. In the model, the Reservoir (IL) block is set to 0.2 MPa and the other two are set to 0.101 MPa. In the Single-Acting Actuator (IL) block, set the **Environment pressure specification** to `Specified pressure` and set **Environment pressure** to `0.2` MPa.

6 Run the model.

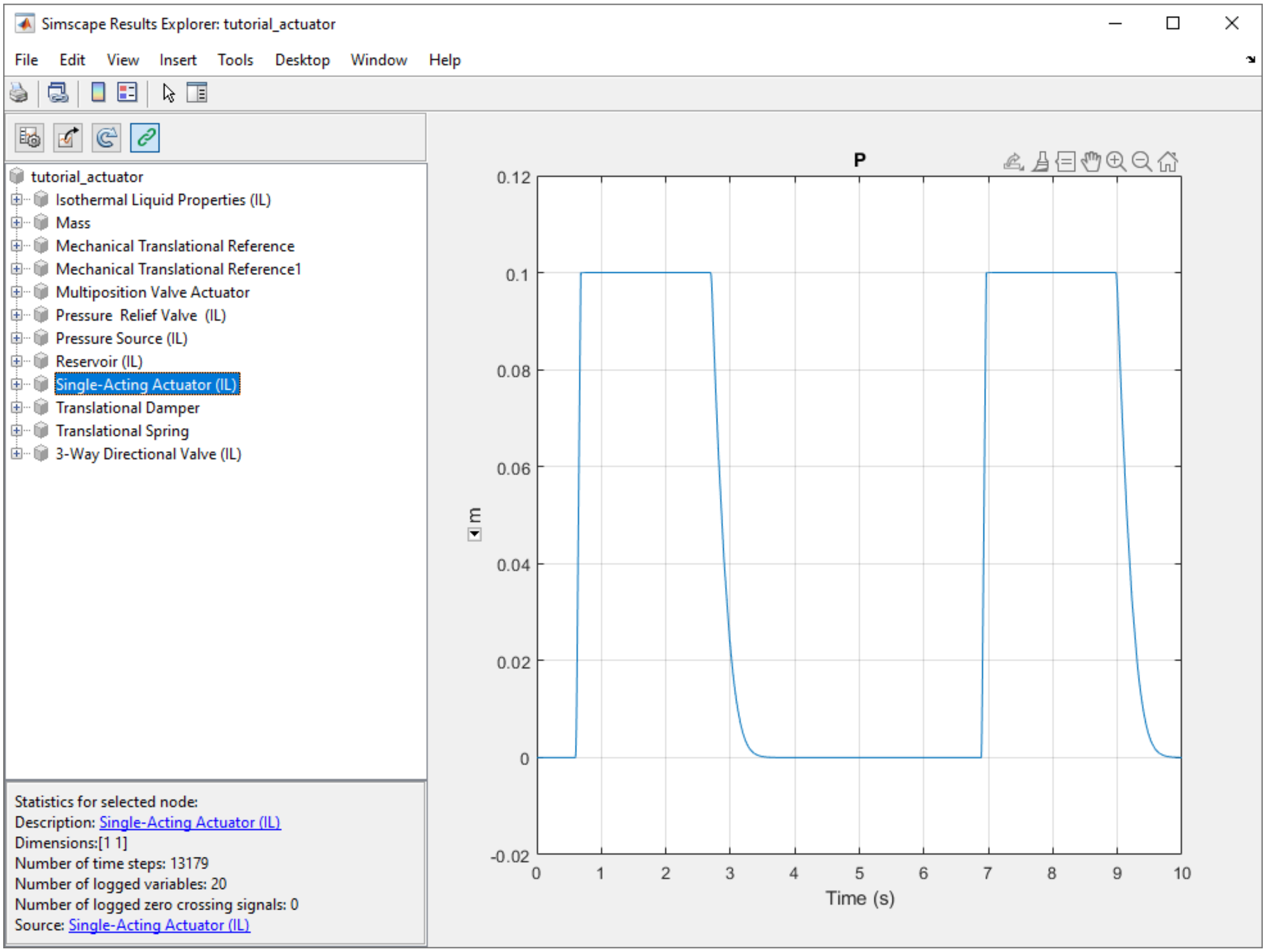
7 Check the results with the Results Explorer by clicking on any block, such as the Multiposition Valve Actuator block, and selecting **Results Explorer** in the **Simscape Block** tab.

8 The Results Explorer shows every variable in the model, including internal variables, which cannot be reported by a Scope block. Click on the **Single-Acting Actuator (IL)** option in the left-hand pane. This shows a plot of the physical signal at port **P**, or the position of the actuator shaft. Click the plus next to the option to see the variables logged for the Single-Acting Actuator (IL) block.

9 The plot shows the shaft position for the last 3.5 seconds of the simulation. To see the shaft position over the entire simulation, turn data log limiting off:

- **a** Open Configuration Parameters. On the **Modeling** tab in the Simulink toolstrip, select **Model Settings** > **Model Settings** and click **Simscape** in the left-hand pane.
- **b** Clear **Limit data points**.
- **c** Run the model.

The actuator shaft position now follows the control signal.



Simscape Results Explorer: tutorial_actuator

tutorial_actuator
- Isothermal Liquid Properties (IL)
- Mass
- Mechanical Translational Reference
- Mechanical Translational Reference1
- Multiposition Valve Actuator
- Pressure Relief Valve (IL)
- Pressure Source (IL)
- Reservoir (IL)
- Single-Acting Actuator (IL)
- Translational Damper
- Translational Spring
- 3-Way Directional Valve (IL)

Statistics for selected node:
Description: Single-Acting Actuator (IL)
Dimensions:[1 1]
Number of time steps: 13179
Number of logged variables: 20
Number of logged zero crossing signals: 0
Source: Single-Acting Actuator (IL)

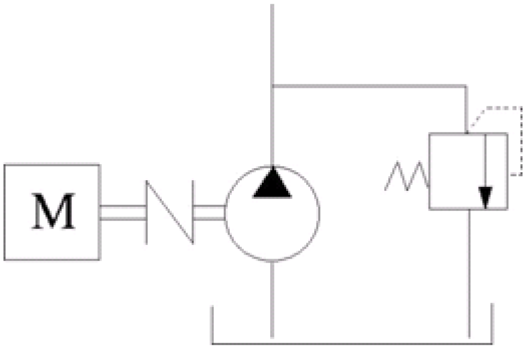
See Also

“Essential Steps for Constructing a Physical Model” | “Essential Physical Modeling Techniques”

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 wide variety of power unit designs varying by the amount and type of pumps, prime movers, valves, tanks, etc. The set of blocks available in the Simscape Fluids libraries 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

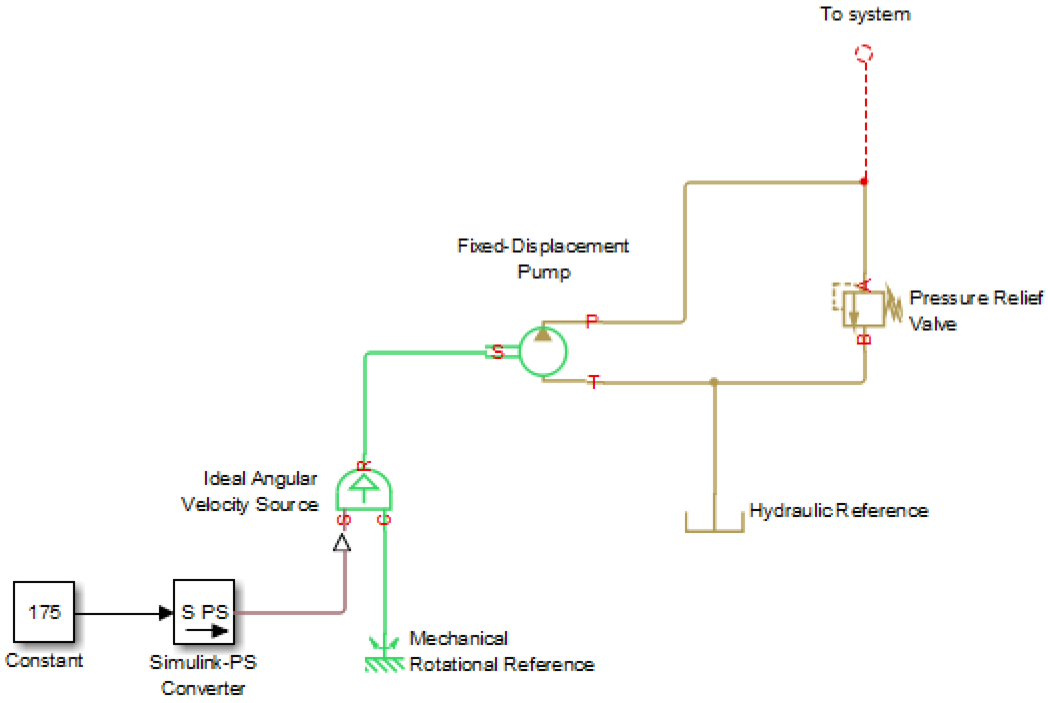
In developing a model of a power unit, you must reach a compromise between the robustness, speed of simulation, and accuracy, meaning that the model should be as simple as possible to provide acceptable accuracy within the working range of variable parameters.

The first option is to simulate a power unit literally, as it is, reproducing all its components. This approach is illustrated in the “Closed-Loop Pump Control with Flexible Drive Shaft” on page 6-450 example. 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.

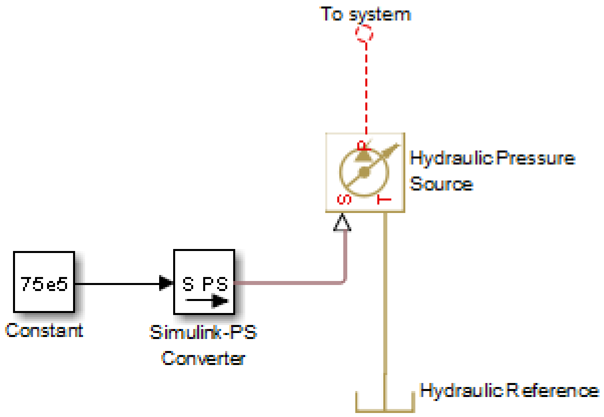
You can implement a considerably more complex model of a prime mover by following the pattern used in the example. For instance, the shaft can be represented with multiple segments and intermediate bearings. The model of a prime mover can be more comprehensive, accounting for its type (DC or AC electric 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 using Simscape Driveline™ software can be combined with the Simscape Fluids model of the 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 pump delivery exceeds the system's fluid requirements at all times, 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 Hydraulic Pressure Source block, as shown in the next illustration.



Using the Hydraulic 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 Valves in Simscape Fluids

In this section...

You can model flow control, pressure control, and directional flow control with Fluids library valve blocks. Flow-controlling valves, such as a gate valve, limit the volumetric flow rate through a valve. Pressure-controlling valves limit valve operational pressures by opening or closing the valve based on pressure, or by venting excess flow.

Directional valves connect different flow paths based on the displacement of a central control member. They can be proportional-area, such as a four-way directional valve, or proportional-pressure, such as a check valve. Directional valves such as a shuttle valve can also control flow switching.

Proportional directional valves are configured based on the flow paths that open and close when the control member displaces, and are classified by:

- The number of external paths or connecting ports
- The number of positions the valve can assume
- Continuous or discrete movement

The directional valves in the Isothermal Liquid, Thermal Liquid, and Gas libraries comprise two-way, three-way, and four-way directional valves with a single, continuous spool control member. The block ports are named to correspond with common connections:

- P — Pump or pressure port
- T — Tank or return port
- A, B — Load ports
- X, Y — Pilot ports

Additionally, you can add mechanical effects to a valve by connecting blocks modeling damping, friction, and a hard stop to the valve actuator signal, such as in the "4-Way Directional Valve with Mechanical Effects" on page 6-544 example.

Valve Control with Valve Actuator Blocks

You can model servo or solenoid valve control with blocks from the Valve Actuator sublibrary, such as the Proportional Valve Actuator or the Pilot Valve Actuator (IL). To model solenoid control, you can also use a Solenoid block from the Simscape Electrical library.

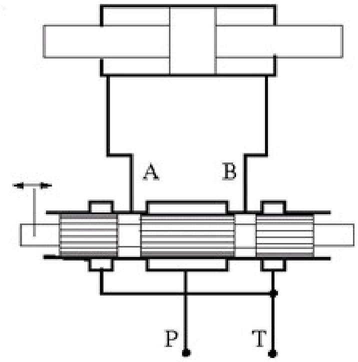
The Multiposition Valve Actuator block can be paired with directional valves to control the valve spool position according to input signals or switching time.

Four-Way Directional Valve Schematic

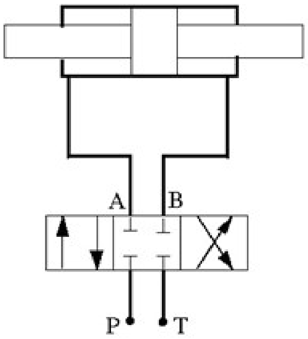
A proportional flow directional valve is characterized by its initial open flow path, the flow paths that open or close with positive or negative control member displacement, and the orifice opening offset.

The relationship between the control member stroke and the opening identifies whether an orifice is overlapped, underlapped, or neutral.

When a four-way, three-position directional valve has all ports closed in the neutral position, the valve, connected to a double-acting actuator, looks like this:

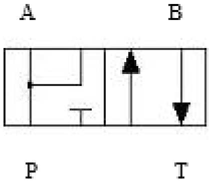


When the control member displaces in the positive direction, to the right in the valve diagram, flow paths P-A and B-T open. When the control member displaces in the negative direction, to the left in the valve diagram, flow paths P-B and A-T open. This configuration for positive and negative displacement is schematically represented as:



where the box on the left represents the positive spool displacement connections, and the box on the right represents the negative spool displacement connections.

In a four-way, two-position valve, additional flow configurations may open only during transition. For example, setting the 4-Way 2-Position Directional Valve (IL) block to this valve configuration



where:

- **Positive spool position open connections** is set to `P-A and P-B`.
- **Negative spool position open connections** is set to `P-A and B-T`.

results in open P-A, P-B, and B-T flow paths when the spool transitions between the positive and negative positions.

Note In the 4-Way 3-Position Directional Valve (IL) and 4-Way 2-Position Directional Valve (IL) blocks, you can:

- Select a pre-parameterization based on manufacturer data sheets. This can be set in the description section of the block dialog.
- Visualize the opening characteristics of each orifice in the valve. Right-click on the block and select **Fluids > Plot Valve Characteristics**.

Parameterizing Four-Way Directional Valve Blocks

In the 4-Way 3-Position Directional Valve (IL) and 4-Way 2-Position Directional Valve (IL) blocks, the *spool position at maximum orifice area* parameter indicates the absolute spool position at which that orifice is fully open. For example, when the valve is configured for a P-A flow path, the P-A orifice is open at the value specified in the **Spool position at maximum P-A orifice area** parameter. Orifices identified in the **Positive spool position open connections** parameter must have a positive spool position at the maximum opening. Orifices identified in the **Negative spool position open connections** parameter must have a negative spool position at the maximum opening. If these valves are not also identified in the **Neutral spool position open connections** parameter, they must be closed at a spool position of 0.

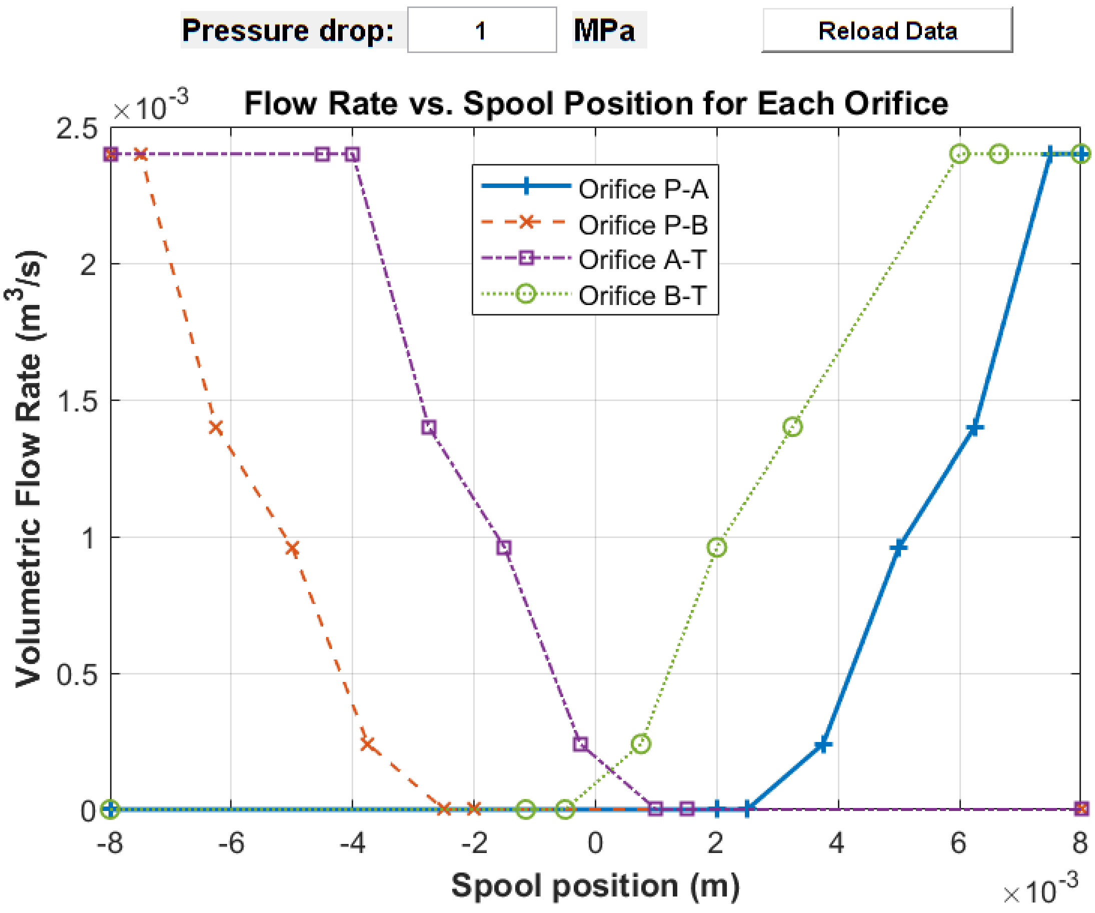
The *spool travel between closed and open orifice* parameter is the stroke that opens the orifice. If **Area characteristics** is set to `Different for each flow path`, the spool travel between the closed and open orifice can be defined differently for each flow path.

To check that the defined spool positions for each orifice opening and closing position comprise a consistent configuration, the **Consistency check for neutral spool position open connections** parameter can notify you if parameterization values are inconsistent when the spool is in the neutral position. A warning or error is generated when you plot or refresh the plot of the valve configuration.

In the 4-Way 3-Position Directional Valve (TL) and 4-Way 3-Position Directional Valve (G) blocks, the *maximum valve opening* parameter is the orifice stroke, which corresponds to the *spool travel between closed and open orifice* parameter in the isothermal liquid blocks. The parameters under **Valve Opening Offsets**, such as **Between ports P and A**, identify the spool displacement from neutral at which the flow path begins to open for each orifice, such as the one between ports **P** and **A**. These parameters correspond to the spool position at the orifice *minimum* opening.

Visualizing a Valve Configuration

In the 4-Way 3-Position Directional Valve (IL) and 4-Way 2-Position Directional Valve (IL) blocks, you can visualize the orifice area for each flow path as a function of spool position:



Pressure drop: 1 MPa

This visualization shows a four-way, three-position valve configured with:

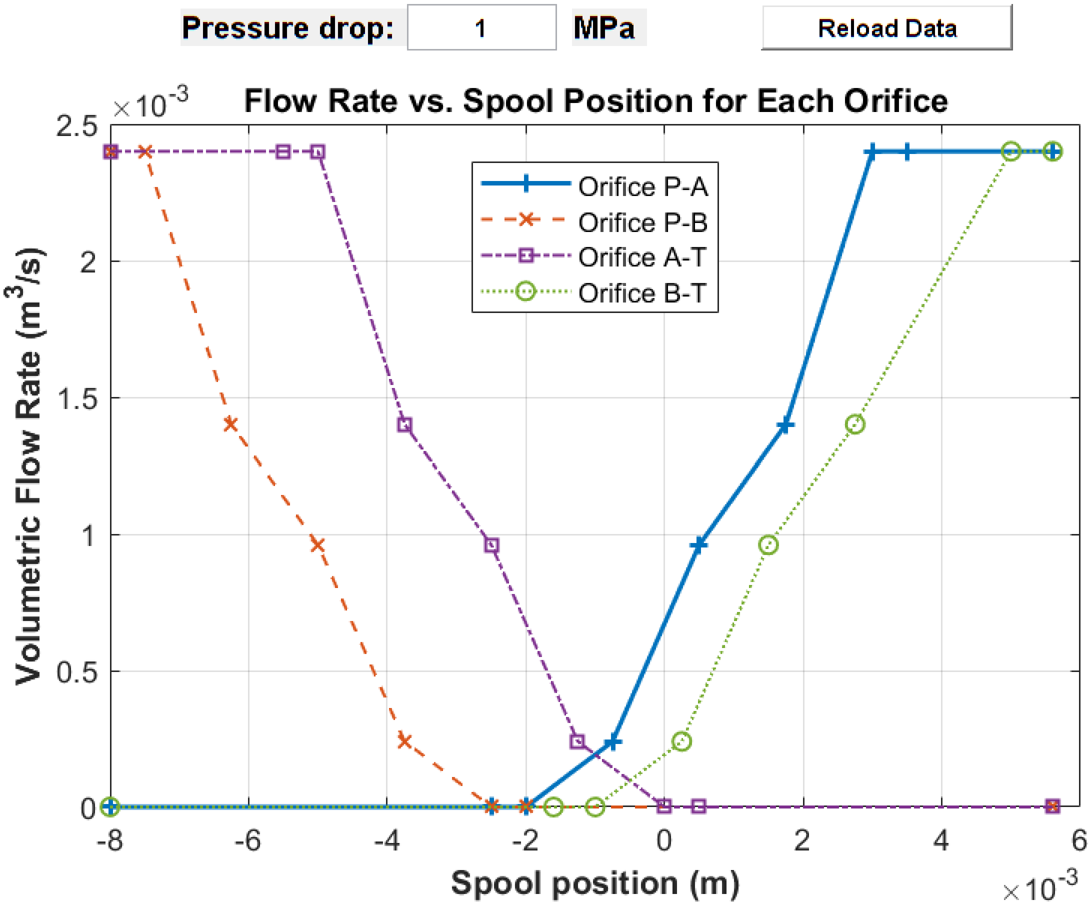
- **Positive spool position open connections** set to `P-A and B-T`
- **Neutral spool position open connections** set to `A-T and B-T`
- **Negative spool position open connections** set to `P-B and A-T`
- **Area characteristics** set to `Different for each flow path`
- **Orifice parameterization** set to `Tabulated data - Volumetric flow rate vs. spool travel and pressure drop`

With the following values changed from the block defaults:

- **Spool position at maximum A-T orifice area** set to `-0.004` m
- **Spool position at maximum B-T orifice area** set to `0.006` m
- **B-T orifice spool travel vector, ds** set to `[0, .00125, .0025, .00375, .0065]` m.

The visualization shows that orifices A-T and B-T have a nonzero flow rate, or are open, when the spool position is 0. Orifice B-T is fully closed when the spool displaces to -0.0005 m; it is fully open at 0.006 m and has an opening length of 0.0065 m. Orifice A-T is fully closed at 0.001 m; it is fully open at -0.004 m and has an opening length of 0.005 m.

Changing **Spool position at maximum P-A orifice area** to `0.003` m results in an open P-A orifice when the spool is in the neutral position, which violates the valve configuration settings. This is because the last element of the **P-A orifice spool travel vector, ds**, is larger than the **Spool position at maximum P-A orifice area**, meaning that the valve must start opening when the spool is still in the negative position:



Parameterization with a Datasheet

Manufacturer data sheets provide flow rate–pressure differential curves for the different flow paths of a directional valve. You can parameterize a directional valve block with data from the manufacturer by setting **Orifice parameterization** to `Tabulated data - Volumetric flow rate vs. spool travel and pressure drop`.

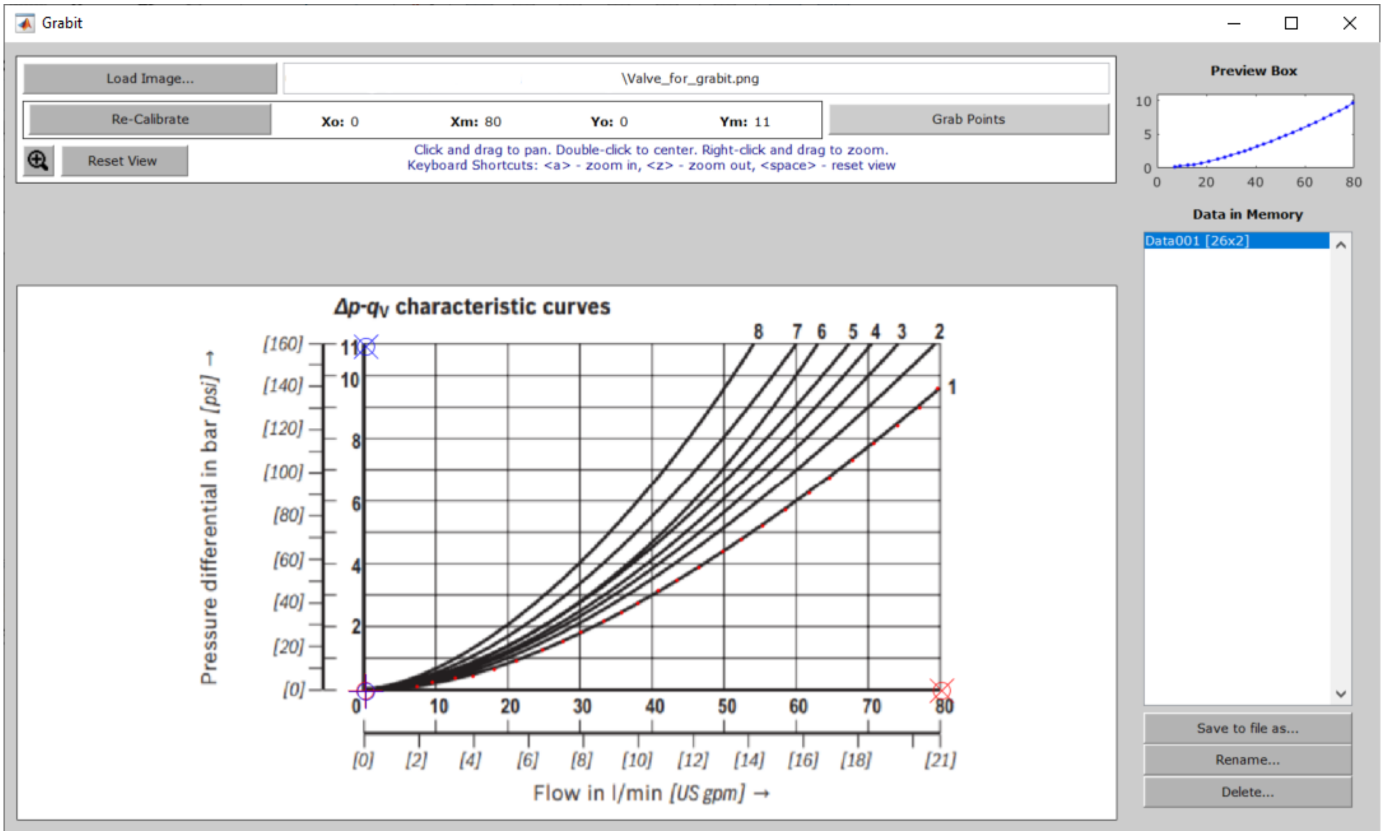
If you do not want to manually input data from a data sheet, you can use the MATLAB File Exchange function `grabit` to extract data points from a plot image:

1 Enter `grabit()` at the MATLAB command line.

2 From the GUI that appears, click **Load Image** to load a valve curve in PNG, TIF, JPG, GIF, or BMP format.

3 Calibrate the image by selecting the axes ranges and entering the range values. Start by clicking on the *x*-axis origin, entering `0` in the box that appears, and pressing **Enter**.

4 Click **Grab Points** to begin collecting data by clicking along the desired curve. To collect another set of data, press **Enter** and click **Grab Points** again.



Xo: 0 **Xm:** 80 **Yo:** 0 **Ym:** 11

Click and drag to pan. Double-click to center. Right-click and drag to zoom.
Keyboard Shortcuts: <a> - zoom in, <z> - zoom out, <space> - reset view

5 Click **Save to file as** to save the dataset as `P_A_pressure_drop.mat` in your working directory.

6 Enter the saved variables as the parameter setting:

See Also

Multiposition Valve Actuator | 4-Way 3-Position Directional Valve (IL) | 4-Way 2-Position Directional Valve (IL) | 4-Way 3-Position Directional Valve (G) | 4-Way 3-Position Directional Valve (TL)

Related Examples

- “Building a Custom Valve” on page 2-36
- “Front-Loader Actuation System” on page 6-39
- “4-Way Directional Valve with Mechanical Effects” on page 6-544
- “4-Way 3-Position Valve Parameterization” on page 6-23

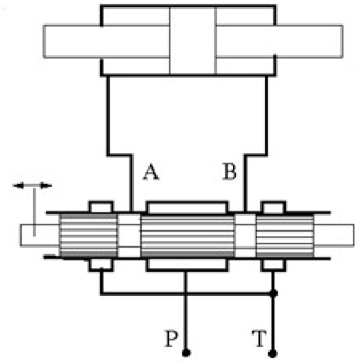
Building a Custom Valve

Valves can be modeled with one or more orifice blocks in series or parallel. For valves that open or close in response to pressure or flow rate, such as a pressure relief valve, you can model your own by connecting a pressure sensor or flow rate sensor to an orifice. Additionally, you can add mechanical effects such as damping and friction, as shown in “4-Way Directional Valve with Mechanical Effects” on page 6-544.

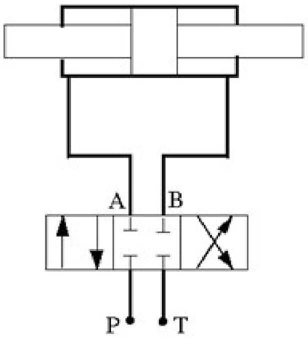
You can use the Solenoid block to model a solenoid-controlled valve. For more information, see “Valve Control with Valve Actuator Blocks” on page 2-30.

Fundamental Directional Valve Composite Structures

When a four-way, three-position directional valve has all ports closed in the neutral position, the valve, connected to a double-acting actuator, looks like this:



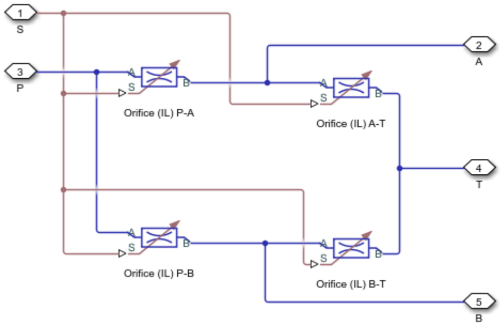
When the control member displaces in the positive direction, flow paths P-B and A-T open. When the control member displaces in the negative direction, flow paths A-P and B-T open. This is schematically represented as:



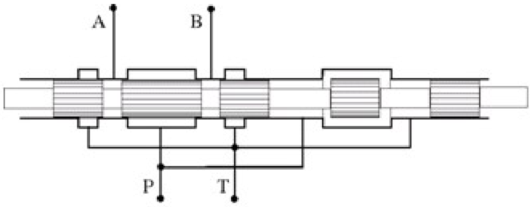
This valve can be modeled as a collection of orifice blocks. In the isothermal liquid domain, this four-way, three-position valve can be recreated with four Orifice (IL) blocks with:

- **Orifice type** set to `Variable`.

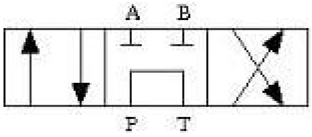
- **Control member position at closed orifice** set to `0` (all valves are closed in the neutral position).
- In Orifice (IL) P-A and Orifice (IL) B-T, **Opening orientation** set to `Positive control member displacement opens orifice`.
- In Orifice (IL) B-P and Orifice (IL) A-T, **Opening orientation** set to `Negative control member displacement opens orifice`.
- For equally proportional flow, set a uniform **Control member travel between closed and open orifice** value for all orifices. For more control over the amount of flow through an orifice, increase the control member travel value with respect to the other orifices.



A different four-way, three-position valve that starts with closed load ports and an open connection between the pump and tank is:

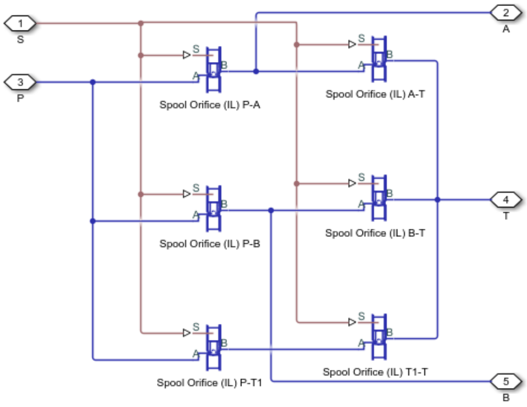


When the control member displaces in the positive direction, flow paths A-T and P-B open. When the control valve displaces in the negative direction, flow paths A-P and B-T open. In the neutral position, flow path P-T is open. This is represented schematically as:

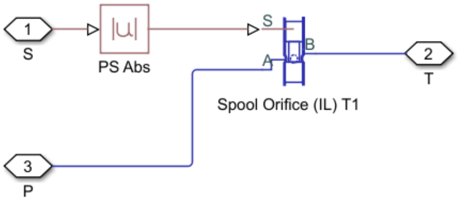


This valve can be built with a collection of spool orifice blocks. One way to model this valve is with two sets of orifices that correspond to an intermediate opening, T1, that represents an opening between **P** and **T** that is only open in the neutral position. One orifice is configured to be open with positive spool displacement, and the other is configured to be open with negative spool displacement.

In addition to the variable orifice block features, the spool orifice block allows you to set the orifice geometry (a series of round holes or one rectangular slot), record the orifice flow force, or connect the flow force physical signal to another block. In the isothermal liquid domain, this directional valve can be represented as:



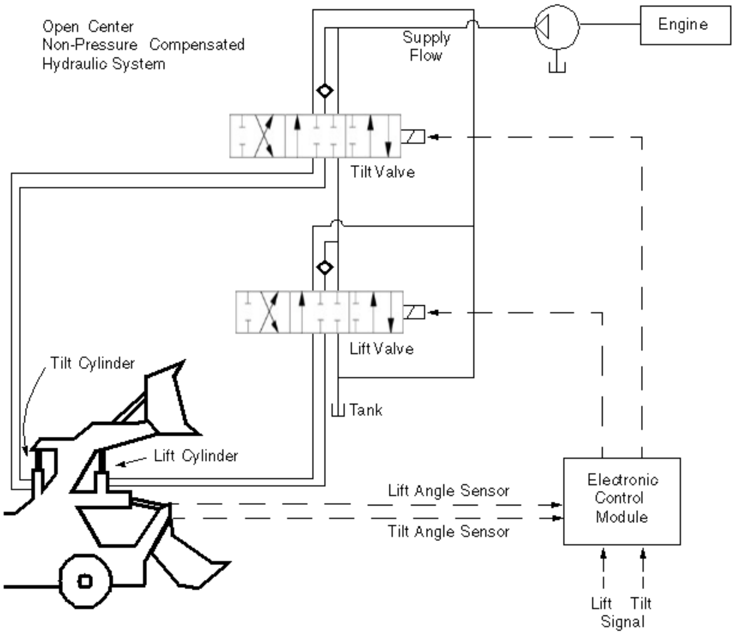
Another way to model an orifice that is always open in the neutral position is to connect one orifice **S** port to the position signal through a PS Abs block and set **Opening orientation** to `Negative spool displacement opens the orifice`:



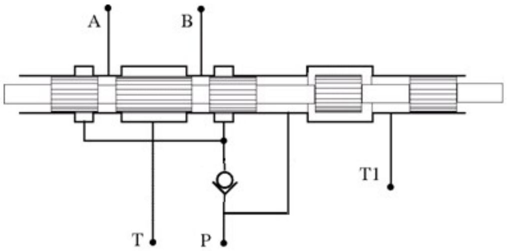
The Spool Orifice (IL) T1 block is configured with **Spool position at closed orifice** set to `0` m.

Custom Directional Valve for a Front Loader

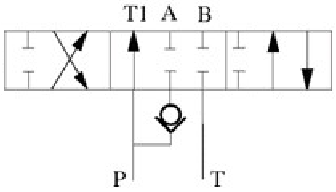
The front loader shown below handles lift and the tilt actuation with two six-way three-position directional valves. If both valves are in the neutral position, the pump port connects to the tank port. The pump is disconnected from the tank if either of the two valves moves from the neutral position.



The valve looks like this:



and is represented schematically as:

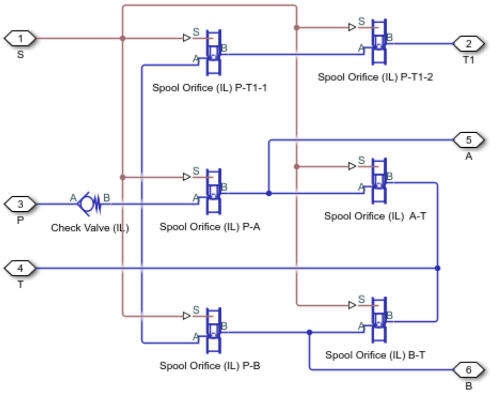


The valve can be modeled as a combination of spool orifice blocks and a check valve block. Orifices P-A, P-B, B-T and A-T are closed when the valve is in the neutral position. Orifices P-T1-1 and P-T1-2 are open at the beginning of the simulation, which allows pump delivery at low pressure to the tank.

In the isothermal liquid domain, in the Spool Orifice (IL) blocks:

- Set **Orifice geometry** to `Round holes`.
- Set **Diameter of round holes** to `0.006`.

- Set **Number of round holes** to `6`.
- In Orifices P-A, P-B, A-T, and B-T, set **Spool position at closed orifice** to `-0.006`.
- In Orifices P-B and A-T, set **Opening orientation** to `Positive spool displacement opens the orifice`.
- In Orifices P-A and B-T, set **Opening orientation** to `Negative spool displacement opens the orifice`.
- In Orifices P1-T1-1 and P1-T1-2, set **Spool position at closed orifice** to `0.008`.



See Also

Multiposition Valve Actuator | 4-Way 3-Position Directional Valve (IL) | 4-Way 2-Position Directional Valve (IL) | Orifice (IL) | Spool Orifice (IL)

Related Examples

- “Modeling Valves in Simscape Fluids” on page 2-30
- “Front-Loader Actuation System” on page 6-39
- “4-Way Directional Valve with Mechanical Effects” on page 6-544
- “4-Way 3-Position Valve Parameterization” on page 6-23

Modeling Hydraulic Directional Valves

In this section...

Classifying Directional Valves

Directional valves connect different flow paths based on the displacement of a central control member. They can be proportional, such as a 4-way directional valve, or control flow switching, such as a shuttle valve. Proportional directional valves are configured based on the flow paths that open and close when the control member displaces, and are classified by:

- The number of external paths or connecting ports.
- The number of positions the valve can assume.
- Being continuous or discrete.

The amount of open orifice at a given time is determined by:

$$h = x_0 + x_{sp}\varepsilon,$$

where:

- h is the current open orifice distance.
- x_0 is the initial orifice offset. This value is positive for underlapped (initially open) valves and negative for overlapped (initially closed) valves.
- x_{sp} is the spool displacement from its original position.
- ε is the orifice orientation. It is +1 is the orifice opens when the spool displaces in the positive direction and -1 if the orifice closes when the spool displaces in the positive direction.

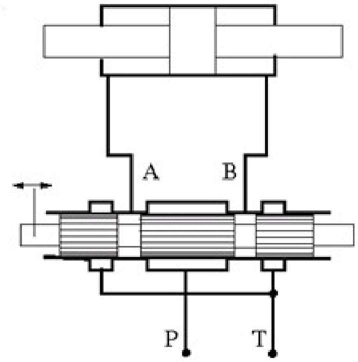
The directional valves in the Simscape Fluids Hydraulics (Isothermal) library comprise two-way, three-way, four-way, and six-way directional valves with a single, continuous spool control member. Block ports are named to correspond with common connections:

- P — Pump or Pressure port
- T — Tank or Return port
- A, B — Load ports
- X, Y — Pilot ports

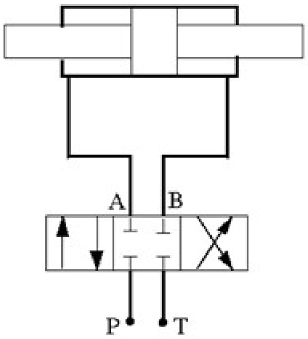
4-Way Directional Valve Schematic

A proportional flow directional valve is characterized by its initial open flow path, the flow paths that open or close with positive or negative control member displacement, and the orifice opening offset. The relationship between the control member stroke and the opening identifies whether an orifice is overlapped, underlapped, or neutral.

One configuration of a 4-way, 3-position directional valve connected to a double-acting cylinder can be connected with all four ports closed in the neutral position. The valve looks like this:



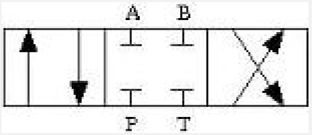
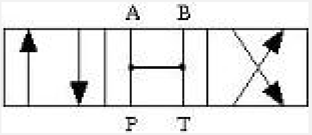
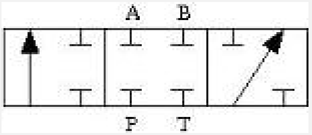
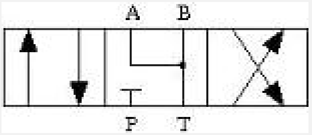
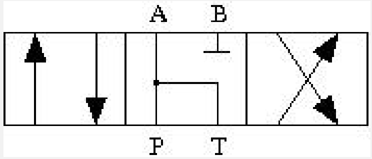
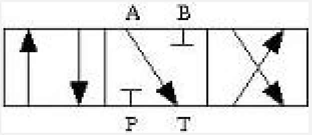
When the control member displaces in the positive direction, to the right in the valve diagram, the flow paths P-A and B-T open. When the control member displaces in the negative direction, to the left in the valve diagram, flow paths B-P and A-T open. This configuration for positive and negative displacement is schematically is represented as:

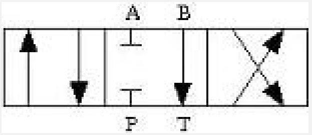
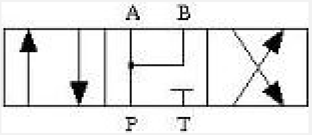
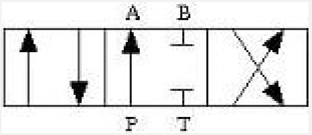
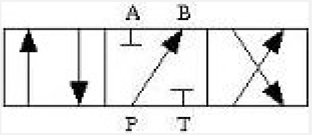
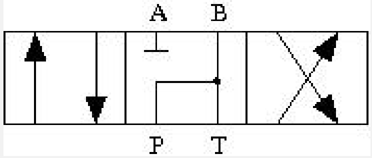


where the box on the right represents the positive spool displacement connections, and the box on the left represents the negative spool displacement connections.

The 4-Way Directional Valve block models this and ten additional configurations, specified in the following table.

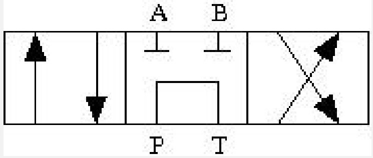
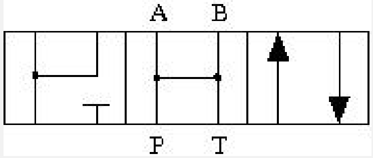
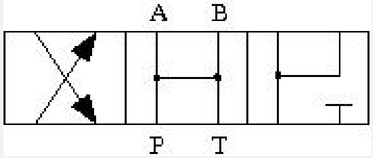
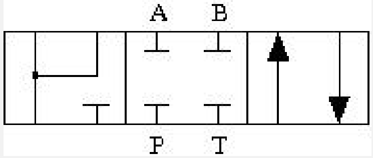
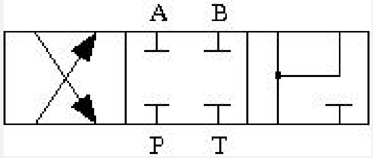
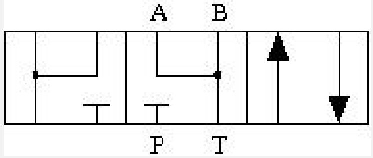
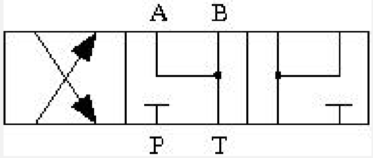
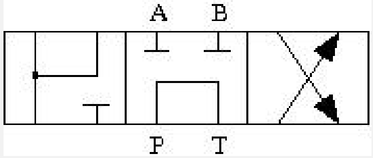
The 4-Way Directional Valve Configurations

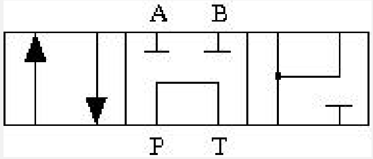
No	Configuration	Initial Openings
1		All four orifices are overlapped in neutral position: • **Orifice P-A initial opening** < 0 • **Orifice P-B initial opening** < 0 • **Orifice A-T initial opening** < 0 • **Orifice B-T initial opening** < 0
2		All four orifices are open (underlapped) in neutral position: • **Orifice P-A initial opening** > 0 • **Orifice P-B initial opening** > 0 • **Orifice A-T initial opening** > 0 • **Orifice B-T initial opening** > 0
3		Orifices P-A and P-B are overlapped. Orifices A-T and B-T are overlapped for more than valve stroke: • **Orifice P-A initial opening** < 0 • **Orifice P-B initial opening** < 0 • **Orifice A-T initial opening** < – *valve_stroke* • **Orifice B-T initial opening** < – *valve_stroke*
4		Orifices P-A and P-B are overlapped, while orifices A-T and B-T are open: • **Orifice P-A initial opening** < 0 • **Orifice P-B initial opening** < 0 • **Orifice A-T initial opening** > 0 • **Orifice B-T initial opening** > 0
5		Orifices P-A and A-T are open in neutral position, while orifices P-B and B-T are overlapped: • **Orifice P-A initial opening** > 0 • **Orifice P-B initial opening** < 0 • **Orifice A-T initial opening** > 0 • **Orifice B-T initial opening** < 0
6		Orifice A-T is initially open, while all three remaining orifices are overlapped: • **Orifice P-A initial opening** < 0 • **Orifice P-B initial opening** < 0 • **Orifice A-T initial opening** > 0 • **Orifice B-T initial opening** < 0

No	Configuration	Initial Openings
7		Orifice B-T is initially open, while all three remaining orifices are overlapped: • **Orifice P-A initial opening** < 0 • **Orifice P-B initial opening** < 0 • **Orifice A-T initial opening** < 0 • **Orifice B-T initial opening** > 0
8		Orifices P-A and P-B are open, while orifices A-T and B-T are overlapped: • **Orifice P-A initial opening** > 0 • **Orifice P-B initial opening** > 0 • **Orifice A-T initial opening** < 0 • **Orifice B-T initial opening** < 0
9		Orifice P-A is initially open, while all three remaining orifices are overlapped: • **Orifice P-A initial opening** > 0 • **Orifice P-B initial opening** < 0 • **Orifice A-T initial opening** < 0 • **Orifice B-T initial opening** < 0
10		Orifice P-B is initially open, while all three remaining orifices are overlapped: • **Orifice P-A initial opening** < 0 • **Orifice P-B initial opening** > 0 • **Orifice A-T initial opening** < 0 • **Orifice B-T initial opening** < 0
11		Orifices P-B and B-T are open, while orifices P-A and A-T are overlapped: • **Orifice P-A initial opening** < 0 • **Orifice P-B initial opening** > 0 • **Orifice A-T initial opening** < 0 • **Orifice B-T initial opening** > 0

Additionally, the Hydraulics (Isothermal) library has nine additional 4-way directional valve blocks, the 4-Way Directional Valves A - K, shown in the following table.

4-Way Directional Valves A - K Configurations

Block Name	Configuration	Description
4-Way Directional Valve A		Contains two additional normally-open, sequentially-located orifices. Valve displacement to the left or to the right closes the path to tank.
4-Way Directional Valve B		Ports P and A are permanently connected through fixed orifice. Orifices P-B and B-T are initially open (underlapped).
4-Way Directional Valve C		Ports P and B are permanently connected through fixed orifice. Orifices P-A, A-T and B-T are initially open (underlapped).
4-Way Directional Valve D		Two orifices are installed in the P-A link. Port A never connects to port T.
4-Way Directional Valve E		Two orifices are installed in the P-B link. Port B never connects to port T.
4-Way Directional Valve F		Two parallel orifices in the P-A arm and two series orifices in the A-T arm.
4-Way Directional Valve G		Two parallel orifices in the P-B arm and two series orifices in the B-T arm.
4-Way Directional Valve H		Two parallel orifices in the P-B arm and two series orifices in the P-T arm.

Block Name	Configuration	Description
4-Way Directional Valve K		Two parallel orifices in the P-A arm and two series orifices in the P-T arm.

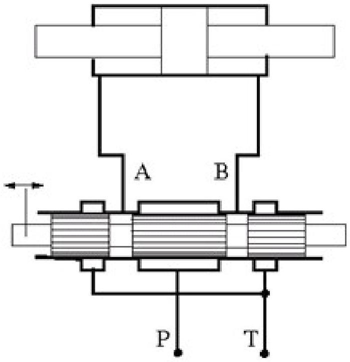
Building a Custom Directional Valve

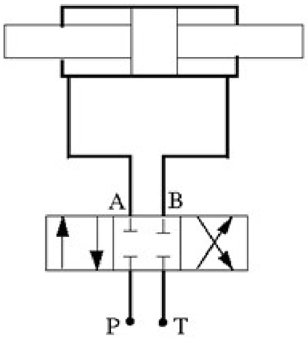
Directional valves can be modeled as a collection of orifices in series and parallel. Available building blocks in the Hydraulics (Isothermal) library for building directional valves are:

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

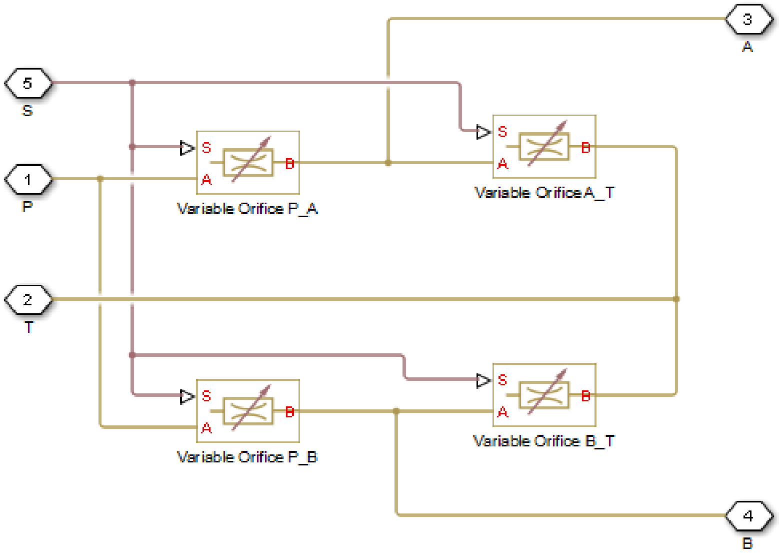
Fundamental Directional Valve Composite Structures

The four-way, three-position valve introduced above:

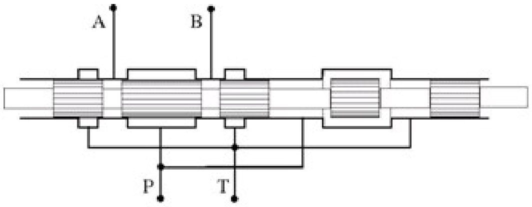




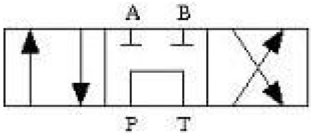
can be modeled as a collection of Variable Orifice blocks:



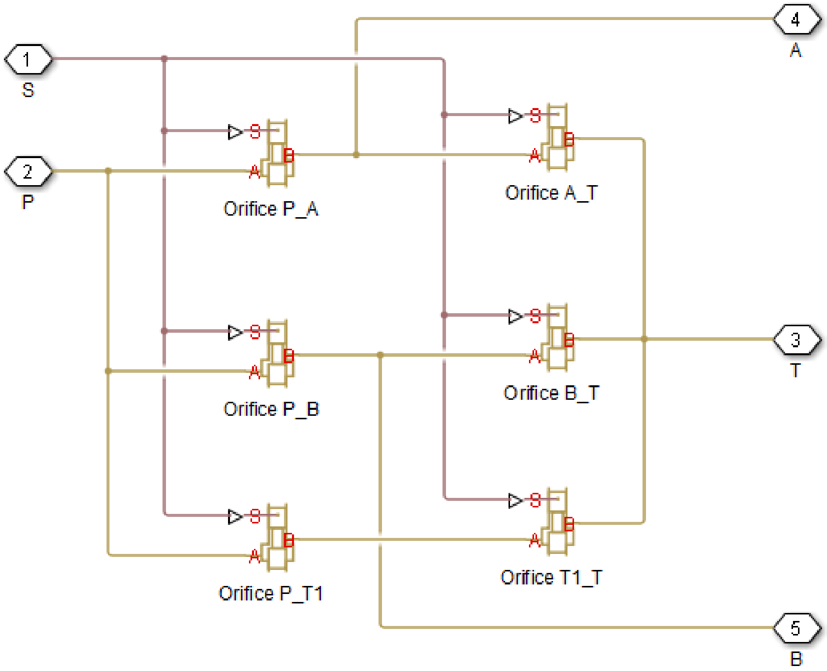
A different valve that starts with closed load ports and an open connection between the pump and tank is:



When the control member displaces in the positive direction, the flow paths P-A and B-T open. When the control valve displaces in the negative direction, flow paths B-P and A-T open. In the neutral position, flow path P-T is open. This is represented schematically as:

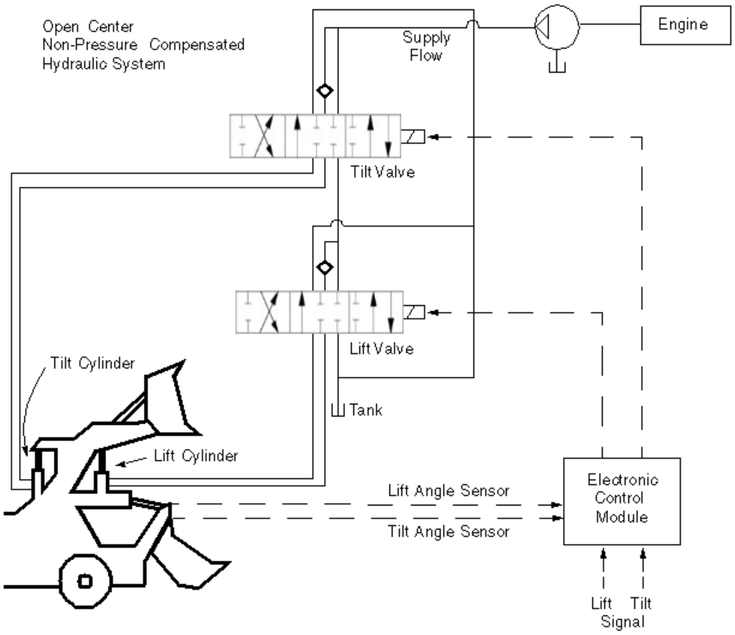


It can be built with a collection of Orifice with Variable Area Round Holes blocks. T1 corresponds to an intermediate point between the pump and tank.

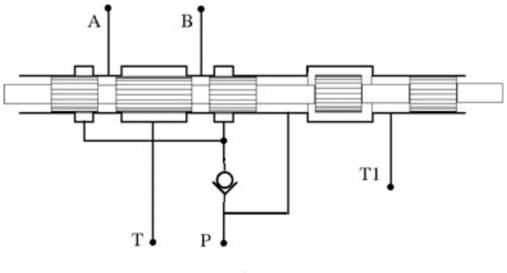


Custom Directional Valve for a Front Loader

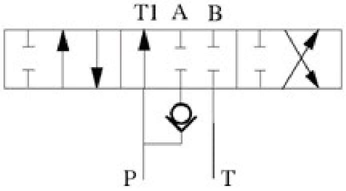
The front loader shown below handles lift and the tilt actuation with two 6-way 3-position directional valves. If both valves are in the neutral position, the pump port connects to the tank port. The pump is disconnected from the tank if either of the two valves moves from the neutral position.



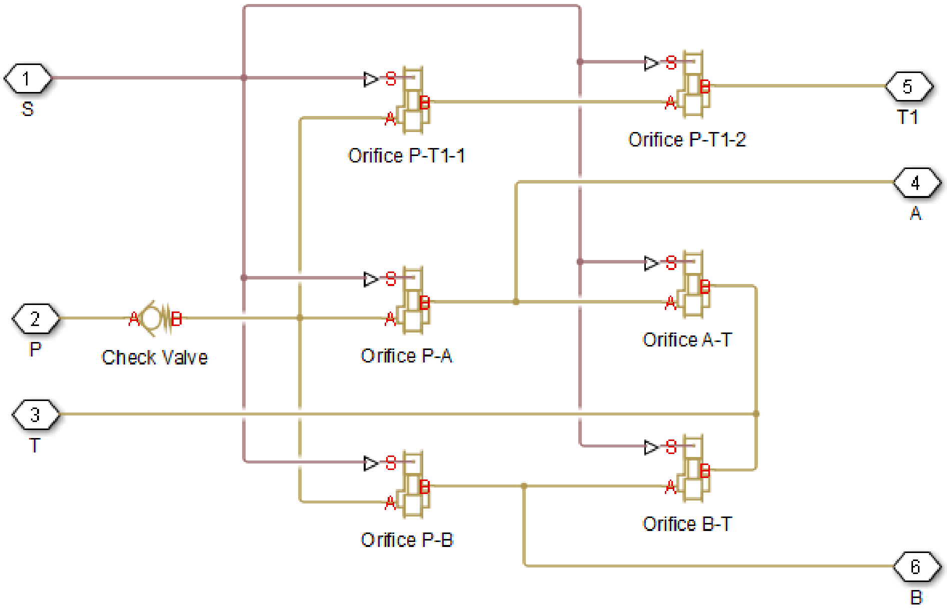
The valve looks like this:



and is represented schematically as:



It can be modeled as a combination of Orifice with Variable Area Round Holes blocks and a Check Valve block.



All of the orifices, except P_T1_1 and P_T1_2, are overlapped when the valve is in the neutral position. Orifices P_T1_1 and P_T1_2 begin the simulation with an opening offset (underlapping) that allows pump delivery at low pressure to the tank.

See Also

4-Way Directional Valve | 3-Way Directional Valve | 2-Way Directional Valve | Shuttle Valve | Check Valve

Related Examples

- "Power-Assisted Steering Mechanism" on page 6-442
- "Hydrostatic Transmission with Shuttle Valve" on page 6-408

Modeling Low-Pressure Fluid Transportation Systems

In this section...

How Fluid Transportation Systems Differ from Power and Control Systems

In hydraulics, the steady uniform flow in a component with one entrance and one exit is characterized by the following energy equation

$$-\frac{\dot{W}_s}{\dot{m}g} = \frac{V_2^2 - V_1^2}{2g} + \frac{p_2 - p_1}{\rho g} + z_2 - z_1 + h_L \qquad (2\text{-}1)$$

where

$\dot{W}_s$	Work rate performed by fluid
$\dot{m}$	Mass flow rate
V_2	Fluid velocity at the exit
V_1	Fluid velocity at the entrance
p_1, p_2	Static pressure at the entrance and the exit, respectively
g	Gravity acceleration
ρ	Fluid density
z_1, z_2	Elevation above a reference plane (datum) at the entrance and the exit, respectively
h_L	Hydraulic loss

Subscripts 1 and 2 refer to the entrance and exit, respectively. All the terms in “Equation 2-1” have dimensions of height and are named kinematic head, piezometric head, geometric head, and loss head, respectively. For a variety of reasons, analysis of hydraulic power and control systems is performed with respect to pressures, rather than to heads, and “Equation 2-1” for a typical passive component is presented in the form

$$\frac{\rho}{2}V_1^2 + p_1 + \rho g z_1 = \frac{\rho}{2}V_2^2 + p_2 + \rho g z_2 + p_L \qquad (2\text{-}2)$$

where

V_1, p_1, z_1	Velocity, static pressure, and elevation at the entrance, respectively
V_2, p_2, z_2	Velocity, static pressure, and elevation at the exit, respectively
p_L	Pressure loss

Term $\frac{\rho}{2}V^2$ is frequently referred to as kinematic, or dynamic, pressure, and ρgz as piezometric pressure. Dynamic pressure terms are usually neglected because they are very small, and "Equation 2-2" takes the form

$$p_1 + \rho g z_1 = p_2 + \rho g z_2 + p_L \tag{2-3}$$

The size of a typical power and control system is usually small and rarely exceeds 1.5 – 2 m. To add to this, these systems operate at pressures in the range 50 – 300 bar. Therefore, ρgz terms are negligibly small compared to static pressures. As a result, Simscape Fluids components (with the exception of the ones designed specifically for low-pressure simulation, described in "Available Blocks and How to Use Them" on page 2-52) have been developed with respect to static pressures, with the following equations

$$\begin{aligned} p &= p_L = f(q) \\ q &= f(p_1, p_2) \end{aligned} \tag{2-4}$$

where

p	Pressure difference between component ports
q	Flow rate through the component

Fluid transportation systems usually operate at low pressures (about 2-4 bar), and the difference in component elevation with respect to reference plane can be very large. Therefore, geometrical head becomes an essential part of the energy balance and must be accounted for. In other words, the low-pressure fluid transportation systems must be simulated with respect to piezometric pressures $p_{pz} = p + \rho gz$, rather than static pressures. This requirement is reflected in the component equations

$$\begin{aligned} p &= p_L = f(q, z_1, z_2) \\ q &= f(p_1, p_2, z_1, z_2) \end{aligned} \tag{2-5}$$

Equations in the form "Equation 2-5" must be applied to describe a hydraulic component with significant difference between port elevations. In hydraulic systems, there is only one type of such components: hydraulic pipes. The models of pipes intended to be used in low pressure systems must account for difference in elevation of their ports. The dimensions of the rest of the components are too small to contribute noticeably to energy balance, and their models can be built with the constant elevation assumption, like all the other Simscape Fluids blocks. To sum it up:

- You can build models of low-pressure systems with difference in elevations of their components using regular Simscape Fluids blocks, with the exception of the pipes. Use low-pressure pipes, described in "Available Blocks and How to Use Them" on page 2-52 .
- When modeling low-pressure systems, you must use low-pressure pipe blocks to connect all nodes with difference in elevation, because these are the only blocks that provide information about the vertical locations of the system parts. Nodes connected with any other blocks, such as valves, orifices, actuators, and so on, will be treated as if they have the same elevation.

Available Blocks and How to Use Them

When modeling low-pressure hydraulic systems, use the pipe blocks from the Low-Pressure Blocks library instead of the regular pipe blocks. These blocks account for the port elevation above reference plane and differ in the extent of idealization, just like their high-pressure counterparts:

- Resistive Pipe LP — Models hydraulic pipe with circular and noncircular cross sections and accounts for friction loss only, similar to the Resistive Tube block, available in the Simscape Foundation library.
- Resistive Pipe LP with Variable Elevation — Models hydraulic pipe with circular and noncircular cross sections and accounts for friction losses and variable port elevations. Use this block for low-pressure system simulation in which the pipe ends change their positions with respect to the reference plane.
- Hydraulic Pipe LP — Models hydraulic pipe with circular and noncircular cross sections and accounts for friction loss along the pipe length and for fluid compressibility, similar to the Hydraulic Pipeline block in the Pipelines library.
- Hydraulic Pipe LP with Variable Elevation — Models hydraulic pipe with circular and noncircular cross sections and accounts for friction loss along the pipe length and for fluid compressibility, as well as variable port elevations. Use this block for low-pressure system simulation in which the pipe ends change their positions with respect to the reference plane.
- Segmented Pipe LP — Models circular hydraulic pipe and accounts for friction loss, fluid compressibility, and fluid inertia, similar to the Segmented Pipe block in the Pipelines library.

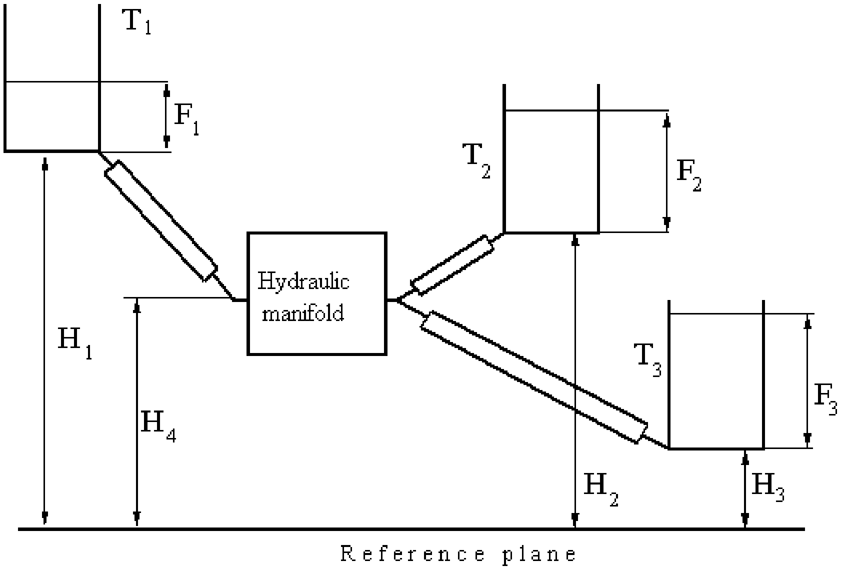
Use these low-pressure pipe blocks to connect all Hydraulic nodes in your model with difference in elevation, because these are the only blocks that provide information about the vertical location of the ports. Nodes connected with any other blocks, such as valves, orifices, actuators, and so on, will be treated as if they have the same elevation.

The additional models of pressurized tanks available for low-pressure system simulation include:

- Constant Head Tank — Represents a pressurized hydraulic reservoir, in which fluid is stored under a specified pressure. The size of the tank is assumed to be large enough to neglect the pressurization and fluid level change due to fluid volume. The block accounts for the fluid level elevation with respect to the tank bottom, as well as for pressure loss in the connecting pipe that can be caused by a filter, fittings, or some other local resistance. The loss is specified with the pressure loss coefficient. The block computes the volume of fluid in the tank and exports it outside through the physical signal port V.
- Variable Head Tank — Represents a pressurized hydraulic reservoir, in which fluid is stored under a specified pressure. The pressurization remains constant regardless of volume change. The block accounts for the fluid level change caused by the volume variation, as well as for pressure loss in the connecting pipe that can be caused by a filter, fittings, or some other local resistance. The loss is specified with the pressure loss coefficient. The block computes the volume of fluid in the tank and exports it outside through the physical signal port V.
- Variable Head Two-Arm Tank — Represents a two-arm pressurized tank, in which fluid is stored under a specified pressure. The pressurization remains constant regardless of volume change. The block accounts for the fluid level change caused by the volume variation, as well as for pressure loss in the connecting pipes that can be caused by a filter, fittings, or some other local resistance. The loss is specified with the pressure loss coefficient at each outlet. The block computes the volume of fluid in the tank and exports it outside through the physical signal port V.
- Variable Head Three-Arm Tank — Represents a three-arm pressurized tank, in which fluid is stored under a specified pressure. The pressurization remains constant regardless of volume change. The block accounts for the fluid level change caused by the volume variation, as well as for pressure loss in the connecting pipes that can be caused by a filter, fittings, or some other local resistance. The loss is specified with the pressure loss coefficient at each outlet. The block computes the volume of fluid in the tank and exports it outside through the physical signal port V.

Low-Pressure Fluid Transportation System

The following illustration shows a simple system consisting of three tanks whose bottom surfaces are located at heights H1, H2, and H3, respectively, from the reference plane. The tanks are connected by pipes to a hydraulic manifold, which may contain any hydraulic elements, such as valves, orifices, pumps, accumulators, other pipes, and so on, but these elements have one feature in common – their elevations are all the same and equal to H4.



The models of tanks account for the fluid level heights F1, F2, and F3, respectively, and represent pressure at their bottoms as

$$p_i = \rho g F_i \text{ for } i = 1, 2, 3$$

The components inside the manifold can be simulated with regular Simscape Fluids blocks, like you would use for hydraulic power and control systems simulation. The pipes must be simulated with one of the low-pressure pipe models: Resistive Pipe LP, Hydraulic Pipe LP, or Segmented Pipe LP, depending on the required extent of idealization. Use the Constant Head Tank or Variable Head Tank blocks to simulate the tanks. For details of implementation, see the “Water Supply System” on page 6-500 and the “Three Constant Head Tanks” on page 6-503 examples.

Model a Refrigeration Cycle

This example explains how to model a closed-loop refrigeration cycle. A refrigeration cycle simulation requires properly tuned component parameters. Imbalances in net energy transfer can cause runaway system pressure and temperature. Because the refrigerant undergoes energy transfer that causes substantial and rapid density changes, start by modeling and verifying the response of individual components. You then integrate the components into an open-loop system. Then, close the loop.

This example walks through developing the model in the “Refrigeration Cycle (Air Conditioning)” on page 6-204 example. In this example you:

1. Use a P-H Diagram (2P) block to define the operating region of your refrigeration system.
2. Build a test harness for the evaporator using the System-Level Condenser Evaporator (2P-MA) block with an ideal flow source and reservoir blocks for boundary conditions.
3. Substitute a Thermostatic Expansion Valve (2P) block into the evaporator test harness.
4. Build a test harness for the condenser using the System-Level Condenser Evaporator (2P-MA) block with a Positive-Displacement Compressor (2P) block, and use two-phase and moist air reservoir blocks for the boundary conditions.
5. Use a Receiver Accumulator (2P) block to connect the harness from step 3 to the harness from step 4.
6. Remove two-phase reservoir blocks and connect the systems before verifying your results.

After each step, test the components at the desired steady-state nominal conditions to make sure the components are stable and the results are what you expect.

After you assemble a stable refrigeration cycle model, you can simulate conditions beyond nominal values by adjusting the reservoir blocks that represent the environment. Keep the nominal parameters in the model component the same, as they represent the nominal sizing of the component.

Step 1: Determine the Pressure-Enthalpy Diagram

You must define parameters for the evaporator, condenser, thermostatic expansion valve, and receiver-accumulator. Before you can define these parameters, you must create a pressure-enthalpy diagram for the cycle you want to simulate.

1. Determine the nominal operating conditions for the system based on your design requirement. This example uses these requirements:

Ambient temperature	30 °C
Desired indoor temperature	22 °C
Area of the house	200 m^2
Cooling capacity	16 kW or 4.5 tons of refrigeration

2. Choose the appropriate refrigerant based on the design requirements. Then, select the four points of the refrigeration cycle on a P-h diagram for that refrigerant. The example model uses R-410a.

   To plot fluid property contours on a pressure-enthalpy diagram, right-click on a Two-Phase Fluid Predefined Properties (2P) block or a Two-Phase Fluid Properties (2P) block. Select **Foundation Library** > **Plot Fluid Properties (Contours)**. Then select **Enthalpy Axis**.

You can use data tips to read pressure, specific enthalpy, and temperature contour values to help you decide on the location of the four points of the refrigeration cycle.

3 Set the condensing temperature, or the saturation temperature in the condenser, to be higher than the outdoor temperature to enable heat transfer from the refrigerant to the outdoor environment.

The example model uses a 45 °C refrigerant temperature to provide a 15 °C temperature difference with ambient.

4 Set the evaporating temperature, or the saturation temperature in the evaporator, to be lower than the desired indoor temperature to enable heat transfer from the indoor air to the refrigerant.

The example model uses a 5 °C evaporator outlet temperature to remove heat from the 22 °C house.

5 To estimate the specific enthalpy end points of the high and low pressure lines in the cycle, set the amount of subcooling at the condenser outlet.

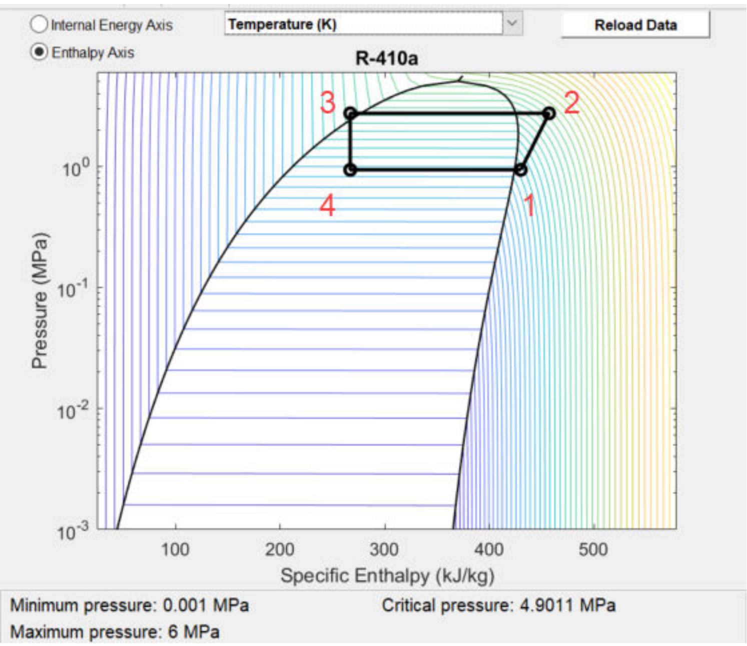
6 Compile the information to find estimates for the four points of the P-h diagram. The example model uses these values:

Location	**Point Number**	**Specific Pressure (p)**	**Specific Enthalpy (h)**	**Notes**
Evaporator outlet	1	0.934 MPa	430 kJ/kg	Corresponds to a superheat of 5 °C
Condenser inlet	2	2.734 MPa	457 kJ/kg	Corresponds to an estimated temperature of 65 °C
Condenser outlet	3	2.734 MPa	267 kJ/kg	Corresponds to a subcooling of 5 °C
Evaporator inlet	4	0.934 MPa	267 kJ/kg	Corresponds to a vapor quality of 0.27

As you develop and refine your model, you can modify these values to be more accurate or precise.

7 Draw the four points onto the refrigerant P-h plot by entering:

```
hold on
plot([430 457 267 267 430], [0.934 2.734 2.734 0.934 0.934], 'k-o', LineWidth = 2)
```



○ Internal Energy Axis

● Enthalpy Axis

Temperature (K)

Minimum pressure: 0.001 MPa

Critical pressure: 4.9011 MPa

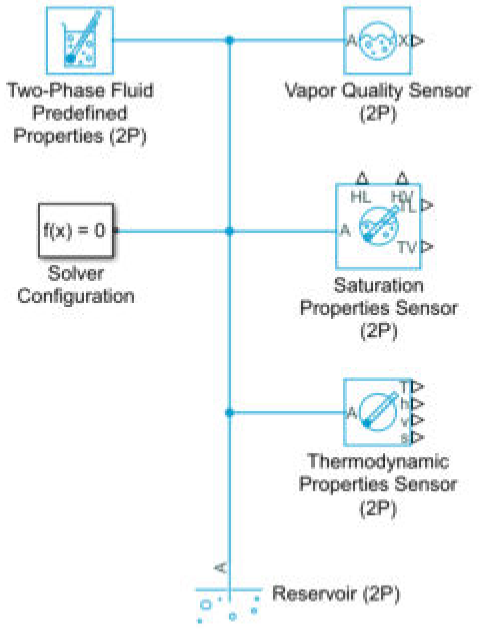
Maximum pressure: 6 MPa

Note Instead of using data tips, you can create a simple model with a reservoir and the desired sensor blocks.

- There is no flow to or from the reservoir, so sensor blocks measure fluid property values in the reservoir.
- Adjust the reservoir conditions as needed to get fluid property values at different conditions from the sensors.

To view the example model created in this step, enter:

```
sscfluids_refrigeration_step1
```



Step 2: Set Up the Evaporator Test Harness

Next, represent the evaporator interaction between the refrigerant and the volume of air in the house. Build the test harness around a System-Level Condenser Evaporator (2P-MA) block. Represent the boundary conditions with the Reservoir (2P) and Reservoir (MA) blocks. Parameterize these blocks using the values you chose in step 1.

1 Add the refrigerant mass flow rate to the harness by using the Mass Flow Rate Source (2P) block, and add the air mass flow rate by using the Mass Flow Rate Source (MA) block. You can approximate values and refine them after closing the loop.

   Set the **Power added** parameter to `None` for both blocks, because these blocks represent boundary conditions that do no work on the flow.

   Note To approximate mass flow rate values for step 2, use the information from step 1. To approximate a mass flow rate value for the:

   - Refrigerant — Divide the cooling capacity by the difference between the evaporator outlet specific enthalpy and the condenser inlet specific enthalpy. This example gives 16 kW / (430 kJ/kg - 267 kJ/kg) = 0.1 kg/s.
   - Air — Divide the cooling capacity by the pressure coefficient of air, and divide that result by the desired temperature drop across the evaporator. This example gives 10 °C, which is

equivalent to a temperature drop of 10 K. Consequently, 16 kW / 1 kg/kJ/K / 10 K = 1.6 kg/s. To calculate the volumetric flow rate, divide the mass flow rate of the air by the density of air, which gives 1.6 kg/s / 1.2 kg/m^3

2 Run the simulation and check your results by using **Simscape Results Explorer**. The simulation should be close to steady-state. Check that the simulation outputs match your estimates from step 1.

3 Adjust the mass flow rate on both the Mass Flow Rate Source (2P) block and the System-Level Condenser Evaporator (2P-MA) block until your model meets the conditions of your P-h diagram.

Note You can use a Condenser Evaporator (2P-MA) block to obtain a higher fidelity model, but you should first validate your model with the System-Level Condenser Evaporator (2P-MA) block.

4 Adjust the mass flow rate on the Mass Flow Rate Source (MA) block and the System-Level Condenser Evaporator (2P-MA) block until your model meets the desired setpoints.

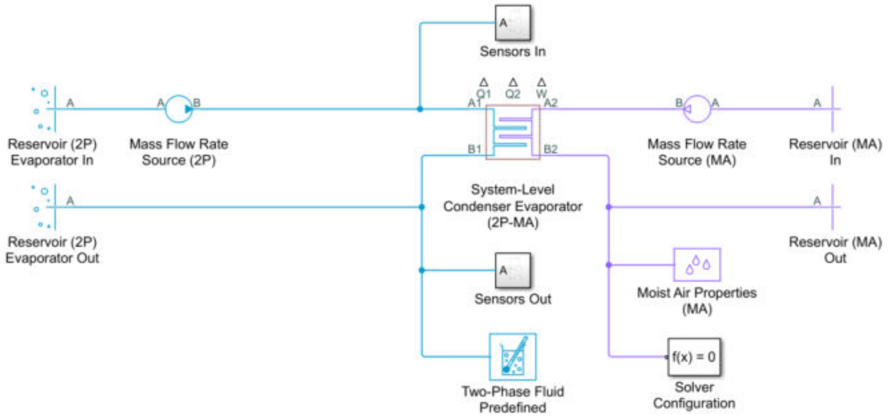
The example model uses a volumetric flow rate of 1.2 m^3/s. This flow rate corresponds to a temperature drop of about 10 °C across the evaporator, where the house temperature is 22 °C and the return air is 12 °C.

5 Choose an appropriate refrigerant tube size based on the refrigerant mass flow rate.

6 Choose an appropriate air duct size based on the air flow rate.

To view the example model created in this step, enter:

```
sscfluids_refrigeration_step2
```



Step 3: Set Up the Thermostatic Expansion Valve Test Harness

Next, model a thermostatic expansion valve to control the performance of the evaporator. A thermostatic expansion valve modulates the flow into the evaporator based on the measured superheat.

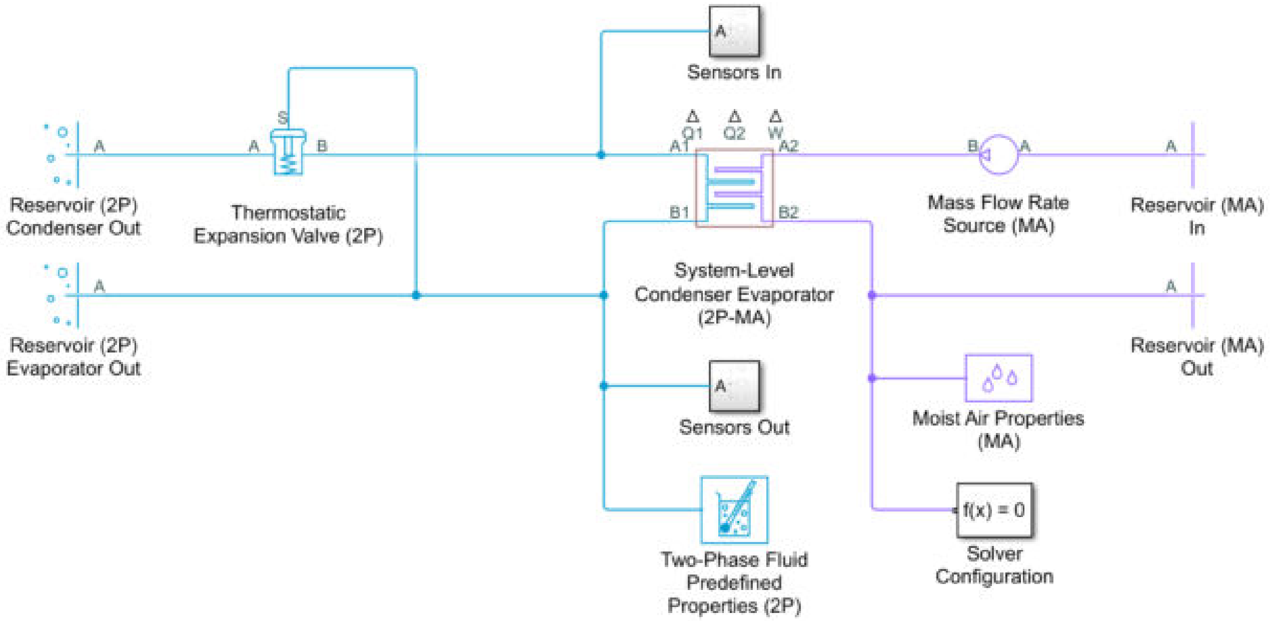
1 Start with your test harness from step 2. Replace the Mass Flow Rate Source (2P) block with the Thermostatic Expansion Valve (2P) block.

a Connect sensing port **S** to the evaporator outlet. This port measures the evaporator superheat.

   b. Use the cycle data from step 1 to change the conditions in the **Reservoir (2P)** block upstream of the valve from the evaporator inlet conditions to the condenser outlet conditions.
2. Set the Thermostatic Expansion Valve (2P) parameters based on the cycle data from steps 1 and 2.
3. Run the model and use the **Simscape Results Explorer** to check the results. The results should be close to the results of step 2. The steady-state value of the **opening_fraction** plot for this model is close to 0.7. Check that the opening fraction for your model meets the design requirements.

To view the example model created in this step, enter:

```
sscfluids_refrigeration_step3
```



Step 4: Set Up the Condenser Test Harness

Build the condenser test harness in the same way that you constructed the evaporator test harness in step 2. Unlike the mass flow rate sources, this compressor must do work on the flow.

1. Model the condenser, flow, and environmental conditions. Use a System-Level Condenser Evaporator (2P-MA) to represent a condenser that rejects heat to the outdoor environment.
2. Connect a Positive-Displacement Compressor (2P) block to drive the refrigerant flow through the condenser.

Note To simplify initial parameterization, you can use a Mass Flow Rate Source (2P) block and set the **Power added** parameter to `Isentropic` to represent the compressor. Switch to the Positive-Displacement Compressor (2P) block before closing the loop in step 6, because a compressor provides more stability to the closed-loop system. This is because the flow rate varies in response to the pressure difference between the high-pressure line and the low-pressure line. In contrast, a mass flow rate source produces an idealized constant mass flow rate regardless of fluctuations in operating conditions.

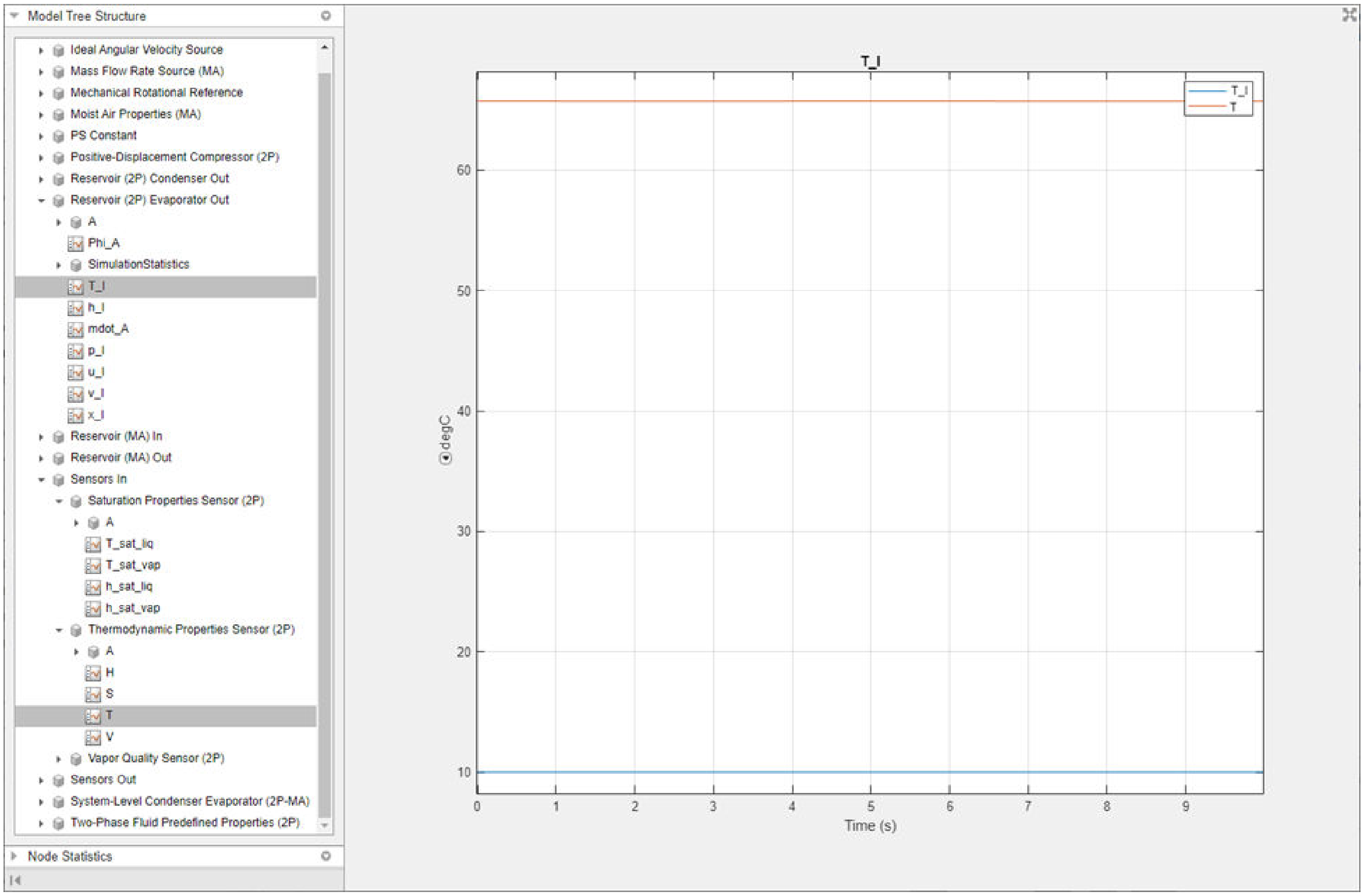
3 Specify the parameters for the nominal operating condition parameters in the condenser block.

4 Use Reservoir (MA) blocks to set up boundary conditions for the external environment. Set the air flow rate with Mass Flow Rate (MA) block.

5 Use the cycle data from step 1 to specify the Positive-Displacement Compressor (2P) block parameters. Set the **Displacement specification** parameter to `Nominal mass flow rate and shaft speed`. Then you use the value that you chose in step 2 for the **Nominal mass flow rate** parameter. The example model uses `0.1 kg/s`.

   In a refrigeration cycle, the compressor drives the refrigerant flow leaving the evaporator and sends it to the condenser. As the compressor does work on the flow, it increases the thermal load on the condenser. Because this portion of the model includes the compressor, you can use the Positive-Displacement Compressor (2P) block instead of the Mass Flow Rate Source (2P) block from step 2.

6 Run the model and use the **Simscape Results Explorer** to check the results. Check the temperatures of the Thermodynamic Properties Sensor (2P) blocks at the condenser inlet and evaporator outlet. Since the compressor does work on the refrigerant, the inlet temperature should be higher than the outlet.


Model Tree Structure
Ideal Angular Velocity Source
Mass Flow Rate Source (MA)
Mechanical Rotational Reference
Moist Air Properties (MA)
PS Constant
Positive-Displacement Compressor (2P)
Reservoir (2P) Condenser Out
Reservoir (2P) Evaporator Out
A
Phi_A
SimulationStatistics
T_I
h_I
mdot_A
p_I
u_I
v_I
x_I
Reservoir (MA) In
Reservoir (MA) Out
Sensors In
Saturation Properties Sensor (2P)
A
T_sat_liq
T_sat_vap
h_sat_liq
h_sat_vap
Thermodynamic Properties Sensor (2P)
A
H
S
T
V
Vapor Quality Sensor (2P)
Sensors Out
System-Level Condenser Evaporator (2P-MA)
Two-Phase Fluid Predefined Properties (2P)
Node Statistics
T_I
T_I
T
degC
10
20
30
40
50
60
0
1
2
3
4
5
6
7
8
9
Time (s)

Adjust the **Nominal inlet temperature** parameter in the System-Level Heat Exchanger (2P) block. Note that the condenser inlet and outlet specific enthalpy match the specific enthalpy end points of the high and low pressure lines in the cycle on the P-h diagram from step 1.

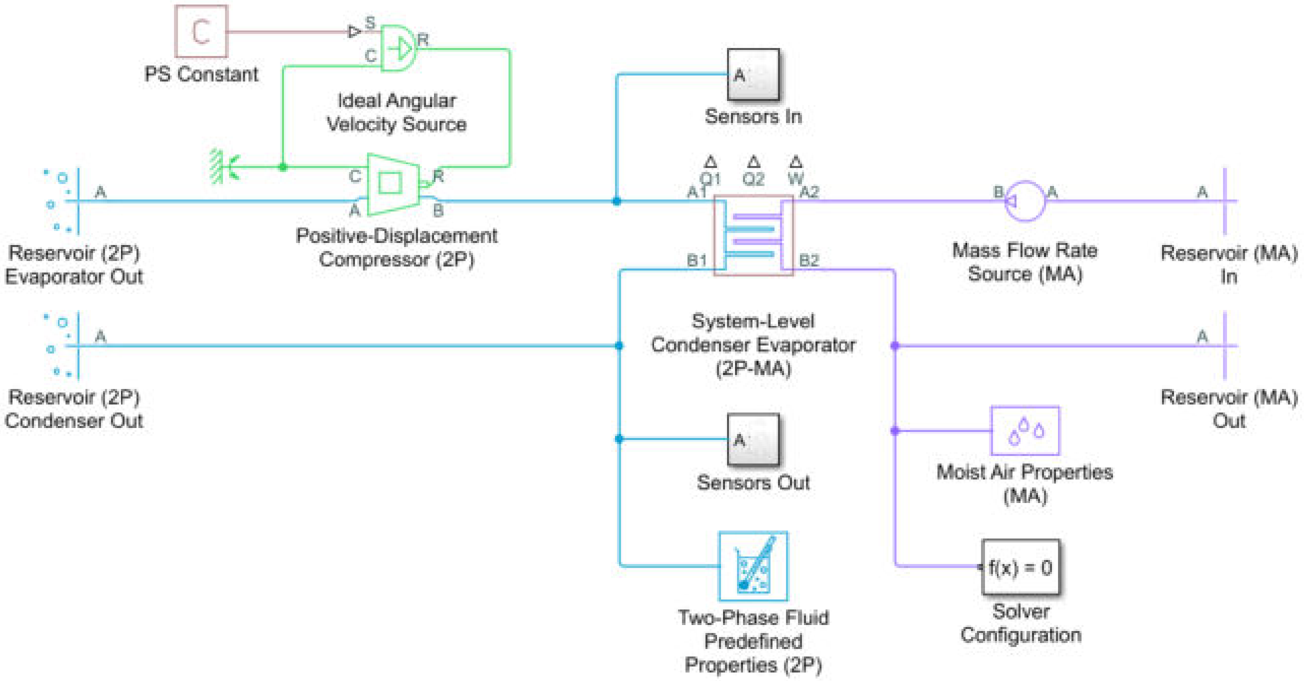
Because the boundary conditions in the reservoir match the nominal operating conditions in the System-Level Heat Exchanger (2P-MA) block and the initial conditions of the System-Level Heat Exchanger (2P-MA) are the same as the nominal operating conditions, the simulation of the test harness model should be close to steady state.

7 Check that the rate of heat transfer in the condenser is approximately equal to the combined rate of heat transfer in the evaporator from step 3 and the fluid power in the compressor. This is important to ensure that the closed-loop system has negligible net energy transfer, which prevents pressure divergence.

8 Adjust the air mass flow rate on both the Mass Flow Rate Source (MA) block and the System-Level Condenser Evaporator (2P-MA) block parameters to safely reject heat from the condenser. The example model uses a volumetric flow rate of 1.5 m^3/s. This flow rate results in an air temperature rise of about 10 °C across the condenser, from 30 °C to 40 °C.

To view the example model created in this step, enter:

```
sscfluids_refrigeration_step4
```


C
PS Constant
S
C
R
Ideal Angular Velocity Source
A
Sensors In
A
Reservoir (2P) Evaporator Out
C
R
A
B
Positive-Displacement Compressor (2P)
Q1
Q2
W
A1
A2
B1
B2
System-Level Condenser Evaporator (2P-MA)
B
A
Mass Flow Rate Source (MA)
A
Reservoir (MA) In
A
Reservoir (2P) Condenser Out
A
Reservoir (MA) Out
A
Sensors Out
Moist Air Properties (MA)
Two-Phase Fluid Predefined Properties (2P)
f(x) = 0
Solver Configuration

Step 5: Create a Model of the Open-Loop System

1 Create a model with all of the components in the system.

- **a** Use a Receiver Accumulator (2P) block to connect the condenser outlet from step 4 to the Thermostatic Expansion Valve (2P) block inlet from step 3.

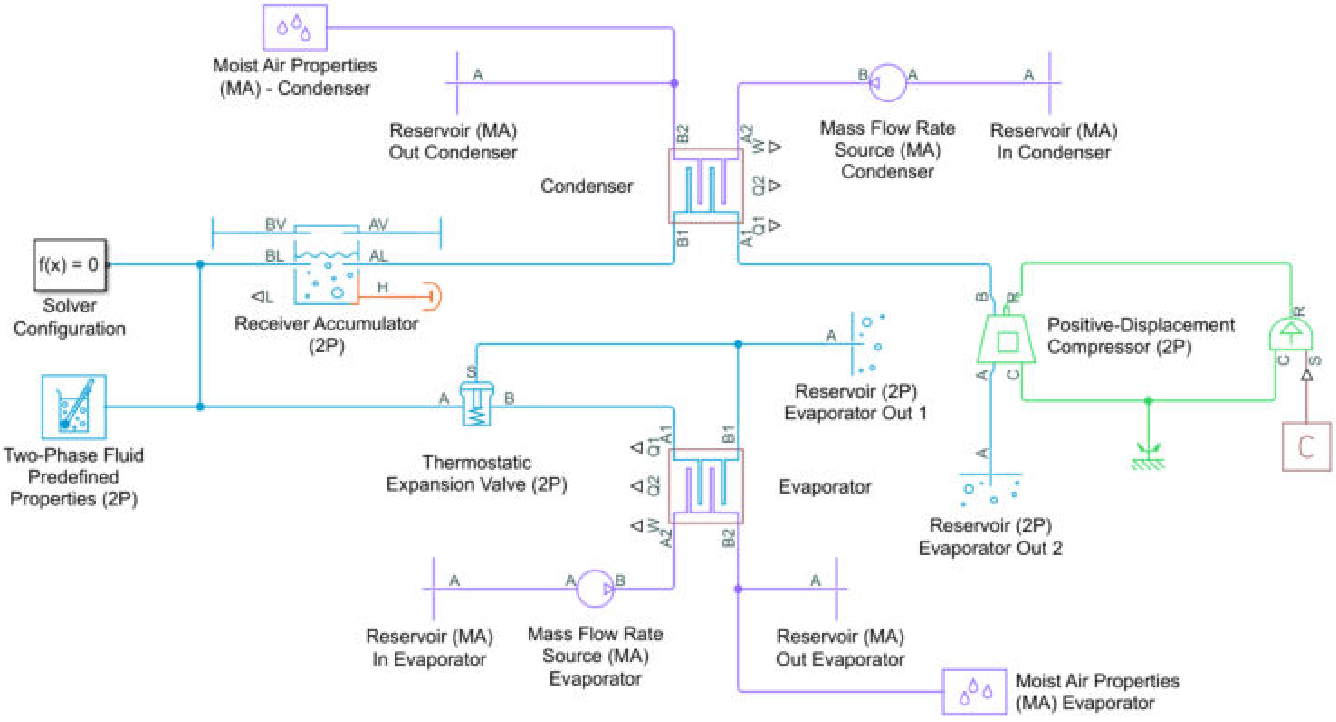
  The Receiver Accumulator (2P) block provides stability to the closed-loop system because it represents a large volume of refrigerant where the liquid level can rise and fall in response to fluctuating operating conditions.

  The volume depends the size of the refrigeration system. Because the model is open-loop for this step, the flow rate through the condenser may not exactly match the flow rate through the evaporator, which causes the liquid level in the Receiver Accumulator (2P) block to rise and fall over time. This is acceptable for the test as long as the level does not change rapidly.
- **b** Keep the evaporator block outlet and the compressor inlet disconnected so that the model remains an open loop.
- **c** Ensure that both Reservoir (2P) blocks match the boundary conditions of the evaporator outlet defined in step 1.

2 Run the model and use **Simscape Results Explorer** to check the results. The results should be close the results from step 3 and step 4.

To view the example model created in this step, enter:

```
sscfluids_refrigeration_step5
```


Moist Air Properties (MA) - Condenser
Reservoir (MA) Out Condenser
A
Condenser
B2
A2
W
Q2
Q1
A1
B1
B
A
Mass Flow Rate Source (MA) Condenser
Reservoir (MA) In Condenser
f(x) = 0
Solver Configuration
BV
AV
BL
AL
L
H
Receiver Accumulator (2P)
Two-Phase Fluid Predefined Properties (2P)
S
Thermostatic Expansion Valve (2P)
Reservoir (2P) Evaporator Out 1
Evaporator
Positive-Displacement Compressor (2P)
R
C
S
Reservoir (2P) Evaporator Out 2
Reservoir (MA) In Evaporator
Mass Flow Rate Source (MA) Evaporator
Reservoir (MA) Out Evaporator
Moist Air Properties (MA) Evaporator

Step 6: Close the Loop

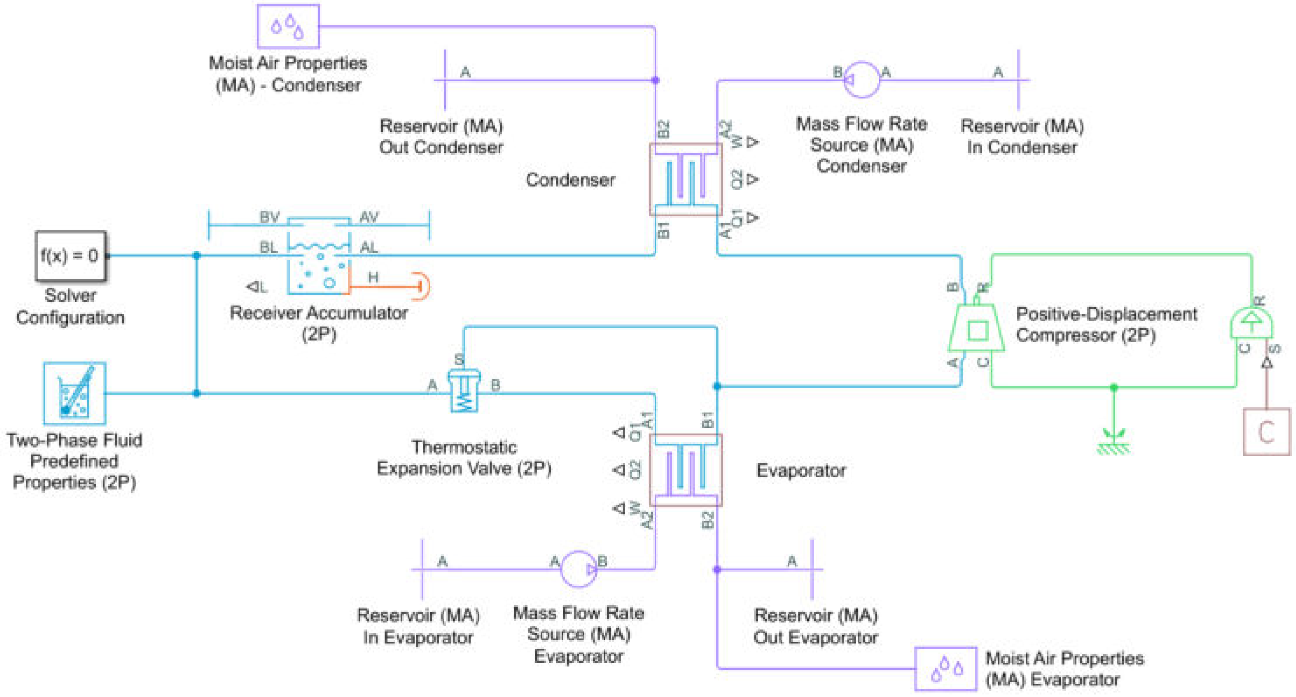
You are now ready to remove the reservoirs and connect the evaporator to the compressor.

1. Remove the reservoirs from step 5 and connect the evaporator outlet to the compressor inlet.
2. Run the model and use the **Simscape Results Explorer** to check your results. They should be close to the results from step 5.

   a. If the pressure keeps increasing or decreasing, then it is likely that the condenser heat transfer does not match the combined evaporator heat transfer and compressor fluid power, which results in a net energy transfer to or from the refrigerant. Return to your harness from step 4.
   b. Check that the liquid level in the Receiver Accumulator block is stable at the steady-state nominal operating condition. Adjust its volume, if necessary.
3. Check that the refrigerant mass flow rate, evaporator pressure, and condenser pressure are stable at the steady-state nominal operating conditions. If the condenser heat transfer does not match the evaporator heat transfer plus the compressor fluid power, the pressure may continue rising or falling during the simulation.
4. Check that the liquid level in the Receiver Accumulator (2P) block is stable at the steady-state nominal operating condition and adjust its volume if it is necessary.

To view the example model created in this step, enter:

```
sscfluids_refrigeration_step6
```


Moist Air Properties (MA) - Condenser
Reservoir (MA) Out Condenser
A
Condenser
B2
A2
W
Q2
Q1
B1
A1
B
A
Mass Flow Rate Source (MA) Condenser
Reservoir (MA) In Condenser
f(x) = 0
Solver Configuration
BV
AV
BL
AL
H
L
Receiver Accumulator (2P)
Positive-Displacement Compressor (2P)
R
C
S
Two-Phase Fluid Predefined Properties (2P)
Thermostatic Expansion Valve (2P)
Evaporator
Reservoir (MA) In Evaporator
Mass Flow Rate Source (MA) Evaporator
Reservoir (MA) Out Evaporator
Moist Air Properties (MA) Evaporator
C

Make Additional Refinements to Fidelity

After you complete your model, you can continue to modify and refine the design to improve the fidelity of your system model. The `sscfluids_refrigeration` model demonstrates these refinements to the model from step 6:

- A house thermal network to represent the moist air network at the evaporator
- A Fan (MA) block in place of the Mass Flow Rate Source (MA) block for greater fidelity
- A controller that turns the system on and off to maintain a given indoor temperature

See Also

Condenser Evaporator (2P-MA) | “Modeling Moist Air Systems”

Related Examples

- “Refrigeration Cycle (Air Conditioning)” on page 6-204

Numerical Smoothing

Numerical Smoothing in Valve Opening Calculations

When a valve block that uses linear parameterization is in a near-open or near-closed position, variables such as valve area or control pressure transition from constant to linear regions. The transitions between the regions are sharp and their slopes discontinuous. These transitions pose a challenge to variable-step solvers and can extend the time needed to finish the simulation run. When you enable numerical smoothing, the block removes the discontinuities.

When the **Smoothing factor** parameter is nonzero, the block enables numerical smoothing. The block applies the smoothing to a normalized variable. For a variable X, the normalized variable, $\widehat{X}$, is

$$\widehat{X} = \frac{X - a}{b - a}$$

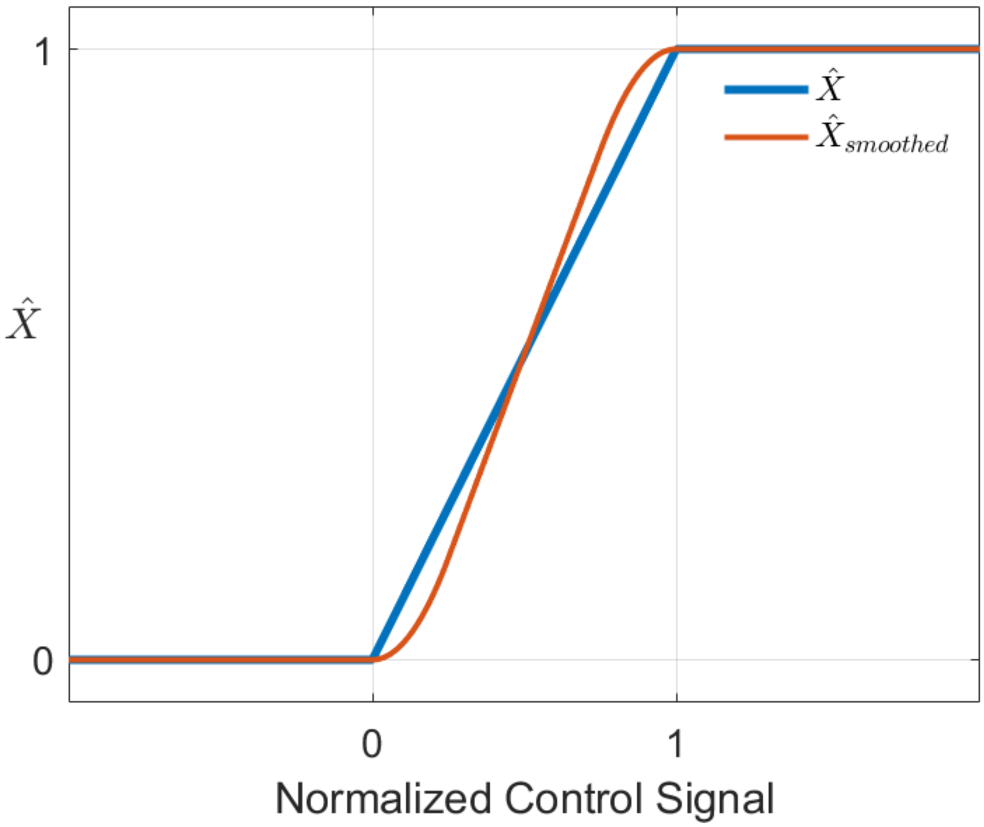
where a and b are the lower and upper limit of the variable, respectively. For most valve blocks, this limit depends on whether the valve is fully open or closed.

When you enable numerical smoothing, the block applies a smoothing factor, f, as

$$\widehat{X}_{smoothed} = \begin{cases} 0, & \widehat{X} < 0 \\ \dfrac{2}{2f - f^2}\widehat{X}^2, & 0 \le \widehat{X} < f/2 \\ \dfrac{2}{2 - f}\widehat{X} + \dfrac{f}{2f - 4}, & f/2 \le \widehat{X} < 1 - f/2 \\ \dfrac{-2}{2f - f^2}\left(1 - \widehat{X}^2\right), & 1 - f/2 \le \widehat{X} < 1 \\ 1, & 1 \le \widehat{X} \end{cases}$$

The smoothing function impacts all calculated valve areas or valve actuator forces, but primarily influences the simulation at the extremes of the smoothed variable. This smoothing function applies to valves and valve actuators in the gas, thermal liquid, two-phase fluid, and isothermal liquid domains.

This image shows a discontinuous normalized variable, $\widehat{X}$, and a corresponding smoothed variable, $\widehat{X}_{smoothed}$, when the smoothing factor, f, is `0.5`.



After smoothing, the block reverts the variable to its original limits, `[`*a*,*b*`]`:

$$X_{smoothed} = \widehat{X}_{smoothed}(b - a) + a$$

A smoothing factor closer to `1` increases the effect of the smoothing function, while a smoothing factor closer to `0` decreases the effect.

See Also

"Modeling Valves in Simscape Fluids" on page 2-30 | Orifice (G) | Orifice (TL) | Orifice (IL)

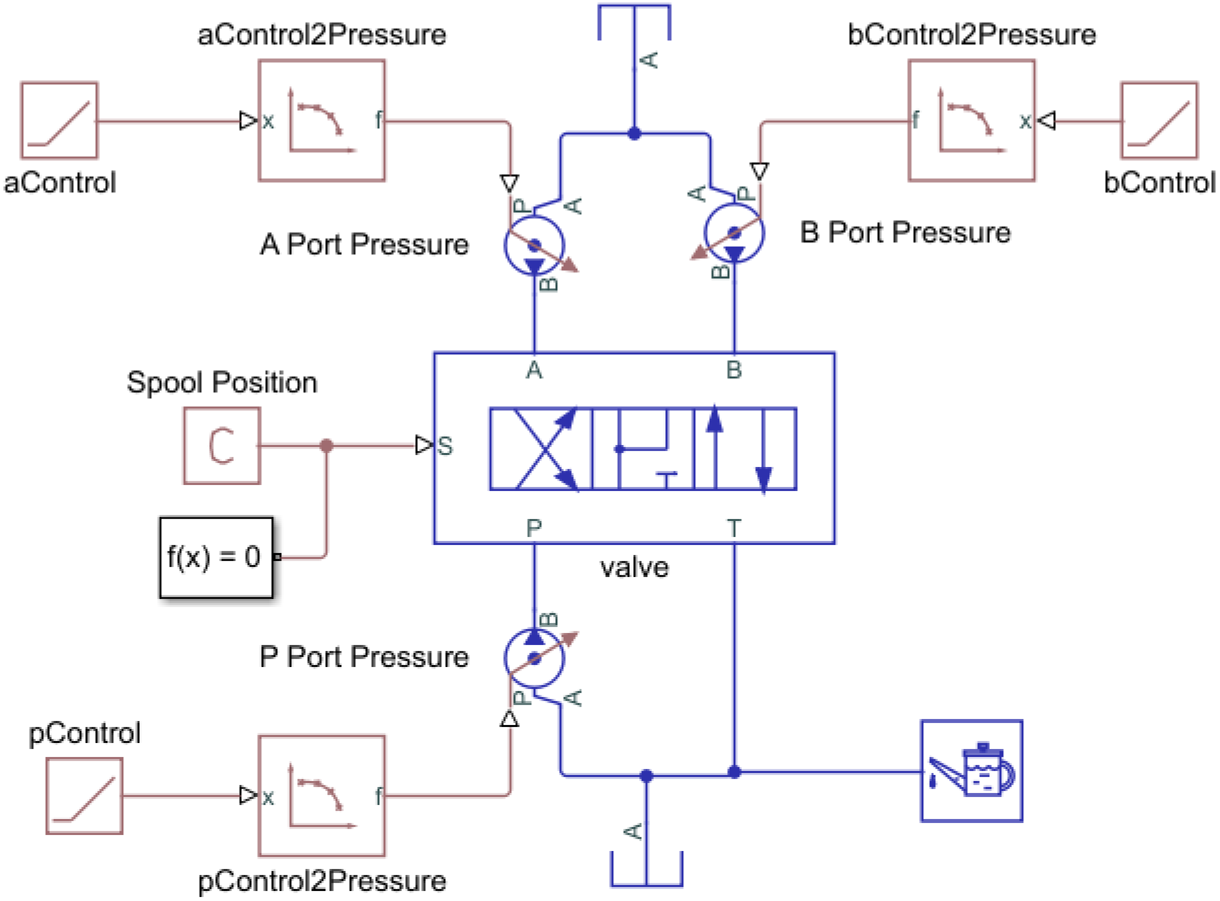
3

Verification and Validation

Validating Flow Response With the Simulation Data Inspector

This example shows the workflow for a unit test assessment of model suitability based on the mass flow rate through a valve by using the Simulation Data Inspector. Performing a successful validation ensures that you are using the most appropriate equations to model a specific application or phenomena. Testing a model's response in a *test harness* helps to assess the response of a model to a controlled range of inputs. You can find other test harness examples in “Test Harnesses”.

This tutorial uses the example `sscfluids_4_way_3_position_valve`, which is composed of a Simulink model and a live script that runs the harness for multiple spool positions:



The `sscfluids_4_way_3_position_valve` example uses a script to run and process the model and plot the simulation data. This tutorial shows an alternative way to validate the model by using the Simulation Data Inspector. Open the model by entering `sscfluids_4_way_3_position_valve` at the command line. Open the live script by clicking the Open live script link at the bottom of the model.

Build a Test Harness

To collect simulation data to validate the four-way valve, construct a unit test that interrogates the valve at a range of pressures. The `sscfluids_4_way_3_position_valve` is an example of a four-way valve in a test harness:

- Use a Ramp block and a PS-Lookup Table (1D) block to set the testing range of the valve. The ramp block specifies the simulation time and the look-up table specifies the validating data range. In `sscfluids_4_way_3_position_valve`:

  - Mass flow rate is the chosen variable to validate against specification sheet data. The test harness is configured to steadily increase pressure from 0 to 5.8 bar.
  - The specification sheet data is for maximum open area, which is when the spool positions are fully extended in the positive and negative positions. The simulation runs two tests: one for the flow paths open at the maximum positive position, and one for the flow paths open at the maximum negative spool position.

- Store your validating data in a Simulation Data Inspector-friendly format. The validating data is specified as volumetric flow rate for pressure differential over the four valve flow paths. However, the data need to be converted to a timeseries in order to compare it in the Data Inspector with the simulation results:

  ```
  %For 11 data points in the look-up table during a ramp from 0 to 100s:
  flowRateData_PB = timeseries(0.0166*[0.58 9.45 20.6 29.9 41.7 55.9 69.3 82.2 94 107 120], lins
  flowRateData_PB.Name = 'Mass Flow Rate PB';

  save validating_data.mat flowRateData_PB
  ```

  The specification sheet data is in lpm. To match the simulation default units, the data set is multiplied by `0.0166` to convert the data to kg/s.

  You can also import timeseries data from an Excel sheet.

- In `sscfluids_4_way_3_position_valve`, the valve performance is modeled by default with tabulated data. Change this to a linear model in the live script at line 26 with the following command:

  ```
  % Setting parameters from Orifice parameterization dropdown
  set_param('sscfluids_4_way_3_position_valve/valve','valve_spec',...
      'fluids.isothermal_liquid.valves_orifices.directional_control_valves.enum.directional_valv
  ```

  This is equivalent to setting the valve **Orifice parameterization** to `Linear - Area vs. spool travel`.

- The test harness is constructed as a unit within a system. To maintain flow through the valve, pumps are connected between the look-up tables and ports.

Collect Simulation Data

The interrogation range is set by the limits of the validating data.

Once the ramp, valve, and 1-D look-up table data and parameters have been set, log the valve data so it can be used in the Simulation Data Inspector. To log the data:

1. In Modeling, select **Model Settings** > **Model Settings**, select Simscape.

2 In the Simscape pane, check **Record data in Simulation Data Inspector**.

3 Check **Open viewer after simulation**.

4 Un-check **Limit data points**.

When the data are recorded in the Simulation Data Inspector, they are saved there under the block name and port. For example, in a test harness saved as sscfluids_4_way_3_position_valve.slx, the mass flow rate in the valve block between ports **P** and **B** is saved as `simlog_sscfluids_4_way_3_position_valve.valve.valve_4_way.orifice_PB.mdot_A.series.values`.

After you enable logging, run the live script.

Assess Model Results

1 Import your validating data.

- Click **Import**. Select **Import From** and navigate to the pressure timeseries data Mass Flow Rate PB. The import pane displays timeseries data currently in your workspace.
- In the **To** section, ensure that **New run** is selected and that the time series data Mass Flow Rate PB is checked. Click **Import**.

2 In the left-hand pane, click **Compare**.

3 Set **Baseline** to **Signals > Mass Flow Rate PB**.

4 Set **Compare to** to **Signals > simlog_sscfluids_4_way_3_position_valve.valve.valve_4_way.orifice_PB.mdot_A (Run 1: 4_way_3_position_valve)**. This run contains the signal data from when the P-B orifice is in open in the positive and neutral positions.

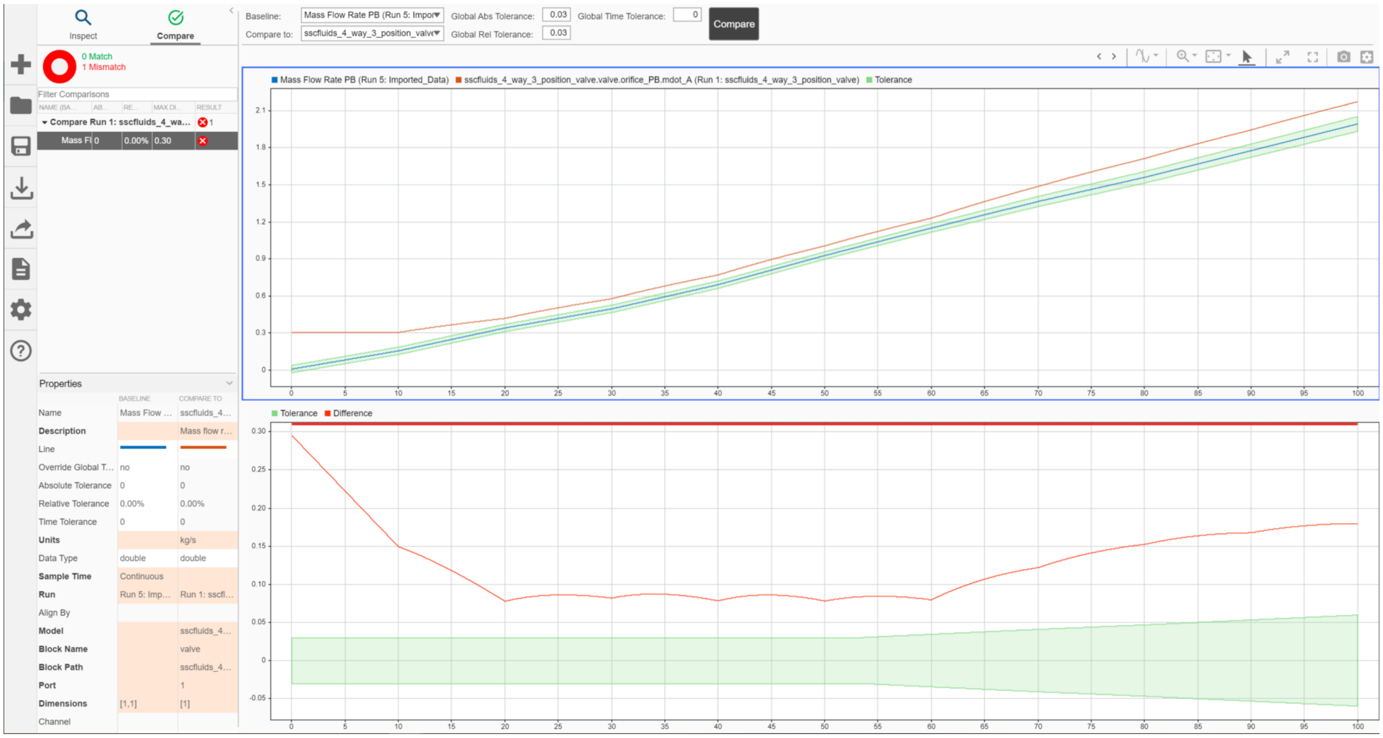
5 The tolerance of your comparison is the precision to which you can validate your model. Depending on the size of a signal, the Simulation Data Inspector uses either the relative or the absolute tolerance for validation:

- *Global absolute tolerance* is the absolute range of acceptable variation between data and a simulation point.
- *Global relative tolerance* signifies the maximum acceptable difference between data and a simulation point relative to the data and simulation values.
- *Global time tolerance* signifies the range in time one value can be compared to another.

For more information on Simulation Data Inspector tolerances, see “How the Simulation Data Inspector Compares Data”.

The absolute and relative tolerances should be equal to or larger than the solver tolerance. In this example, set the absolute and relative tolerances to `0.03` and set the time tolerance to `0`. A nonzero time tolerance value is useful when you expect an impulse response during simulation.

6 Click **Compare**. The simulation and data appear with the tolerance bounding lines:



Baseline: Mass Flow Rate PB (Run 5: Impor▾ Global Abs Tolerance: 0.03 Global Time Tolerance: 0

Compare to: sscfluids_4_way_3_position_valv▾ Global Rel Tolerance: 0.03

0 Match
1 Mismatch

Filter Comparisons

NAME (BA...	AB...	RE...	MAX DI...	RESULT
▾ Compare Run 1: sscfluids_4_wa...				☒ 1
Mass Fl	0	0.00%	0.30	☒

Properties

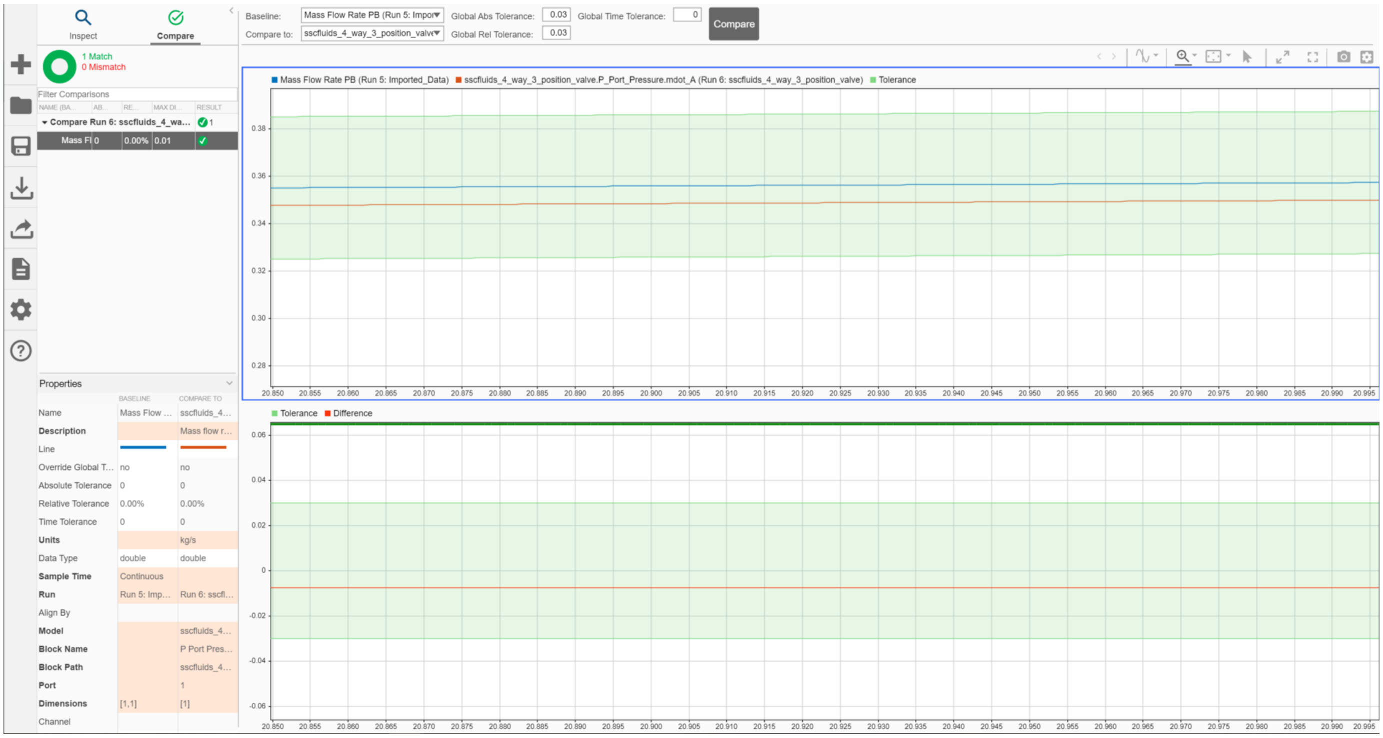
	BASELINE	COMPARE TO
Name	Mass Flow ...	sscfluids_4...
Description		Mass flow r...
Line		
Override Global T...	no	no
Absolute Tolerance	0	0
Relative Tolerance	0.00%	0.00%
Time Tolerance	0	0
Units		kg/s
Data Type	double	double
Sample Time	Continuous	
Run	Run 5: Imp...	Run 1: sscfl...
Align By		
Model		sscfluids_4...
Block Name		valve
Block Path		sscfluids_4...
Port		1
Dimensions	[1,1]	[1]
Channel		

With relative and absolute tolerances of 0.03, the linear model is out of tolerance with the data. When the simulation results are fully within the green band limits, the valve is validated for the tested range of the simulation. The shape of the error suggests that it isn't the most suitable model for the application.

The default parameterization of `sscfluids_4_way_3_position_valve` is the same tabulated data as the specification sheet. To see a model in the Simulation Data Inspector that is fully in tolerance, change the parameterization at line 26 in the live script back to:

```
% Setting parameters from Orifice parameterization dropdown
set_param('sscfluids_4_way_3_position_valve/valve','valve_spec',...
    'fluids.isothermal_liquid.valves_orifices.directional_control_valves.enum.directional_valve_s
```

Run the script again and set **Compare to** to the new run. The error trends shows that this is the correct model for the application, and the results are within tolerance:



Baseline: Mass Flow Rate PB (Run 5: Impor▾ Global Abs Tolerance: 0.03 Global Time Tolerance: 0

Compare to: sscfluids_4_way_3_position_valv▾ Global Rel Tolerance: 0.03

1 Match
0 Mismatch

NAME (BA...	AB...	RE...	MAX DI...	RESULT
▾ Compare Run 6: sscfluids_4_wa...				✓ 1
Mass Fl	0	0.00%	0.01	✓

Properties

	BASELINE	COMPARE TO
Name	Mass Flow ...	sscfluids_4...
Description		Mass flow r...
Line		
Override Global T...	no	no
Absolute Tolerance	0	0
Relative Tolerance	0.00%	0.00%
Time Tolerance	0	0
Units		kg/s
Data Type	double	double
Sample Time	Continuous	
Run	Run 5: Imp...	Run 6: sscfl...
Align By		
Model		sscfluids_4...
Block Name		P Port Pres...
Block Path		sscfluids_4...
Port		1
Dimensions	[1,1]	[1]
Channel		

However, the slight error offset suggests that some tuning would improve the model accuracy.

See Also

More About

- “Creating Custom Components”

4

Model Optimization

Perform Parameter Estimation on a Heat Exchanger

In this section...

If you have a Simulink Design Optimization license, you can use parameter estimation to parameterize a Simscape Fluids block without knowing exact details about the heat exchanger geometry. Parameter estimation uses an objective function and a set of initialized parameters to determine the best values to parameterize a given system.

In this example, you parameterize an air-water heat exchanger with unknown geometry using test data. Follow these steps to fit the parameters to the model:

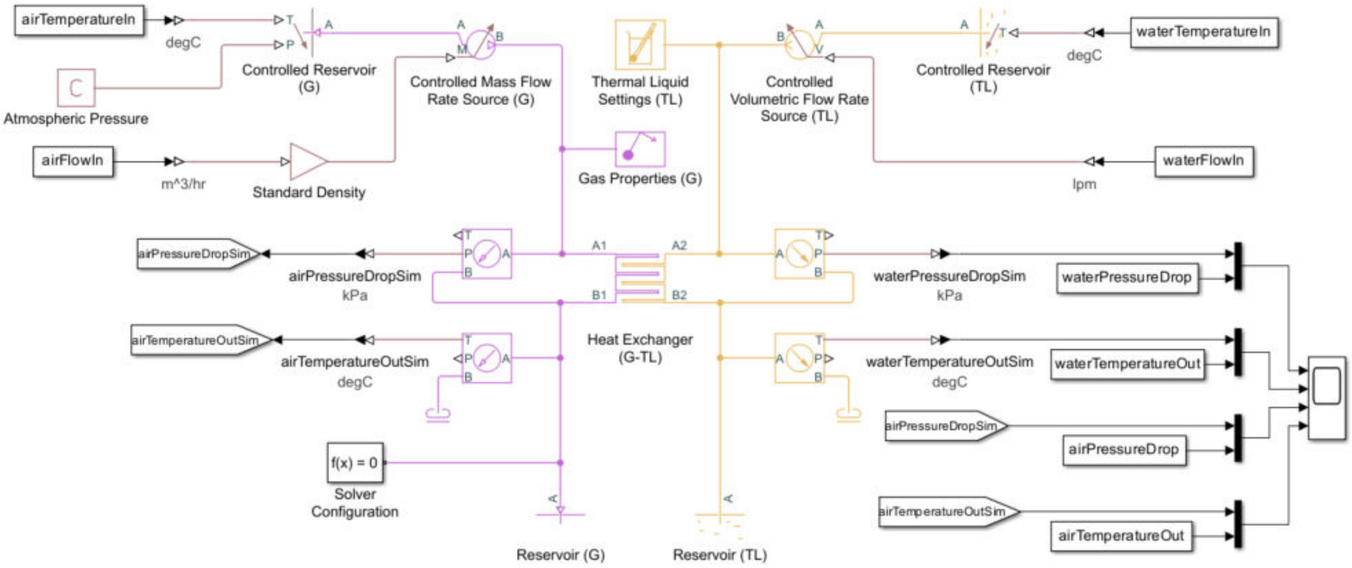
1. “Load the Model and Associated Data” on page 4-2
2. “Design the Parameter Estimation Experiment” on page 4-3
3. “Define the Experimental Parameters” on page 4-4
4. “Configure the Initial Simulation” on page 4-5
5. “Set Up the Objective Function” on page 4-7
6. “Perform Parameter Estimation” on page 4-8

Load the Model and Associated Data

Open the `sscfluids_heat_exchanger_estimation` model. This model contains a test harness that uses a Heat Exchanger (G-TL) block to model an air-water heat exchanger. The only available information about the heat exchanger is the general size and the dimensions of the air and water inlets and outlets along with the operational data. To open the model, enter:

```
exchangerModel = 'sscfluids_heat_exchanger_estimation';
open_system(exchangerModel);
```

The figure shows the default model.



The gas domain represents the air, and the thermal liquid domain represents the water. The heat exchanger takes inlet temperatures and flow rates of the air and the water as inputs. The heat exchanger outputs are the air and water outlet temperatures and flow rates. The Scope block compares the outputs with the test data.

Your system test data must represent steady-state measurements from the physical device and can include signal noise. The test data is most effective when it comprises a range of operating conditions. Your goal is to parameterize the heat exchanger such that the outputs match the test data for the given operating conditions within an acceptable tolerance.

```
load sscfluids_heat_exchanger_estimation_data
```

This data uses these inputs and outputs:

Inputs	Outputs
`airTemperatureIn`	`airTemperatureOut`
`waterTemperatureIn`	`waterTemperatureOut`
`airFlowIn`	`airPressureDrop`
`waterFlowIn`	`waterPressureDrop`

Each variable is a `timeseries` object, which requires sample data stored at discrete time intervals.

Design the Parameter Estimation Experiment

To design the parameter estimation experiment:

- Create an `sdo.Experiment` object.
- Customize `Simulink.SimulationData.Signal` objects for each signal to capture the simulation output signals.

- Associate the Simulink.SimulationData.Signal objects with the sdo.Experiment object.

Use this code to create the sdo.Experiment object:

```
exchangerExperiment = sdo.Experiment(exchangerModel);
```

Create Simulink.SimulationData.Signal objects to store properties and data for each of the four output signals. Set the BlockPath, PortType, PortIndex, Values, and Name properties. Use this code to assign and configure the heat exchanger outputs:

```
airTemperatureOutSignal = Simulink.SimulationData.Signal;
airTemperatureOutSignal.BlockPath = [exchangerModel '/airTemperatureOutSim'];
airTemperatureOutSignal.PortType = 'outport';
airTemperatureOutSignal.PortIndex = 1;
airTemperatureOutSignal.Values = airTemperatureOut;
airTemperatureOutSignal.Name = get(airTemperatureOut, 'Name');

waterTemperatureOutSignal = Simulink.SimulationData.Signal;
waterTemperatureOutSignal.BlockPath = [exchangerModel '/waterTemperatureOutSim'];
waterTemperatureOutSignal.PortType = 'outport';
waterTemperatureOutSignal.PortIndex = 1;
waterTemperatureOutSignal.Values = waterTemperatureOut;
waterTemperatureOutSignal.Name = get(waterTemperatureOut, 'Name');

airPressureDropSignal = Simulink.SimulationData.Signal;
airPressureDropSignal.BlockPath = [exchangerModel '/airPressureDropSim'];
airPressureDropSignal.PortType = 'outport';
airPressureDropSignal.PortIndex = 1;
airPressureDropSignal.Values = airPressureDrop;
airPressureDropSignal.Name = get(airPressureDrop, 'Name');

waterPressureDropSignal = Simulink.SimulationData.Signal;
waterPressureDropSignal.BlockPath = [exchangerModel '/waterPressureDropSim'];
waterPressureDropSignal.PortType = 'outport';
waterPressureDropSignal.PortIndex = 1;
waterPressureDropSignal.Values = waterPressureDrop;
waterPressureDropSignal.Name = get(waterPressureDrop, 'Name');
```

Associate the four output signals with the sdo.Experiment object using this code:

```
exchangerExperiment.OutputData = [
    airTemperatureOutSignal;
    waterTemperatureOutSignal;
    airPressureDropSignal;
    waterPressureDropSignal];
```

Define the Experimental Parameters

To guide the parameter estimation experiment, define which parameters to estimate and supply initial guesses for the block. Focus on the parameters with the greatest potential performance impact. The most significant heat exchanger parameters are the surface area and the pressure loss coefficient. The surface area affects the rate of heat transfer, which affects the outlet temperature. It encompasses the quantity and size of the tubes and fins. The pressure loss coefficient directly affects the pressure loss at the output. For simplicity, you can fix the other parameters to a single value during the experiment.

Use rough values to parameterize the geometry of the heat exchanger. Because the experiment fits the heat exchanger model to the test data, precise values are unnecessary. To roughly parameterize the heat exchanger model, enter:

```
airPipeDiameter = 0.35; % m
waterPipeDiameter = 0.025; % m
exchangerLength = 0.6; % m
exchangerDepth = 0.025; % m
```

These values correspond to the dimensions of the inlet and outlet connections and the general size of the heat exchanger.

Use the `sdo.getParameterFromModel` function to create a `param.Continuous` object for each parameter you want to estimate. Include initial, minimum, and maximum values, then associate the data to the experiment. The experiment tunes the parameters to minimize the error between the model output and the test data. Use engineering judgement when selecting the initial guess and boundaries.

```
airSurfaceAreaInitial = sdo.getParameterFromModel(exchangerModel, 'airSurfaceArea');
airSurfaceAreaInitial.Value = 0.1; % m^2
airSurfaceAreaInitial.Minimum = 0.01;
airSurfaceAreaInitial.Maximum = 10;

waterSurfaceAreaInitial = sdo.getParameterFromModel(exchangerModel, 'waterSurfaceArea');
waterSurfaceAreaInitial.Value = 0.1; % m^2
waterSurfaceAreaInitial.Minimum = 0.01;
waterSurfaceAreaInitial.Maximum = 10;

airLossCoefficientInitial = sdo.getParameterFromModel(exchangerModel, 'airLossCoefficient');
airLossCoefficientInitial.Value = 1;
airLossCoefficientInitial.Minimum = 0.01;
airLossCoefficientInitial.Maximum = 1000;

waterLossCoefficientInitial = sdo.getParameterFromModel(exchangerModel, 'waterLossCoefficient');
waterLossCoefficientInitial.Value = 1;
waterLossCoefficientInitial.Minimum = 0.01;
waterLossCoefficientInitial.Maximum = 1000;
```

Associate the parameters with the estimation experiment.

```
exchangerExperiment.Parameters = [
    airSurfaceAreaInitial;
    waterSurfaceAreaInitial;
    airLossCoefficientInitial;
    waterLossCoefficientInitial];
```

Configure the Initial Simulation

Save time by enabling fast restart mode for the parameters that you want to estimate. This allows Simscape to skip the compilation step after each time Simulink Design Optimization updates a model parameter value. First, open the Heat Exchanger (G-TL) block. Verify that the **Pressure loss coefficient** and **Heat transfer surface area** parameters are run-time enabled for both the gas and thermal liquid domains.

Enable fast restart and create an estimation experiment based on an initial simulation.

```
exchangerSimulator = createSimulator(exchangerExperiment);
exchangerSimulator = fastRestart(exchangerSimulator, 'on');
exchangerSimulator = sim(exchangerSimulator);
```

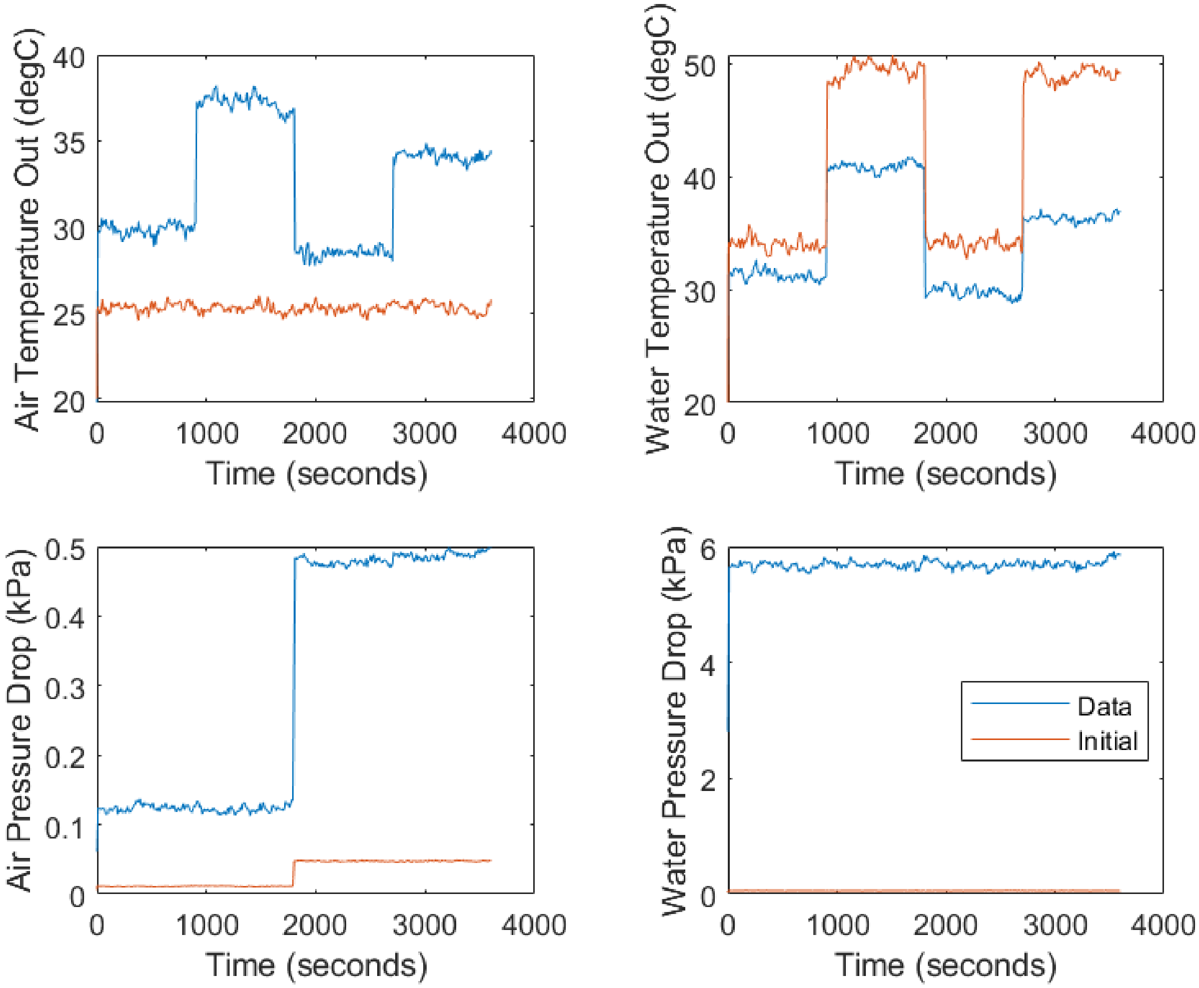
Next, collect the results.

```
logsout = find(exchangerSimulator.LoggedData, get_param(exchangerModel, 'SignalLoggingName'));
airTemperatureOutInitial = find(logsout, 'Air Temperature Out (degC)');
waterTemperatureOutInitial = find(logsout, 'Water Temperature Out (degC)');
airPressureDropInitial = find(logsout, 'Air Pressure Drop (kPa)');
waterPressureDropInitial = find(logsout, 'Water Pressure Drop (kPa)');
```

Use this code to plot the results against the test data.

```
figure
tl = tiledlayout(2,2);
nexttile
plot(airTemperatureOut)
hold on
plot(airTemperatureOutInitial.Values)
nexttile
plot(waterTemperatureOut)
hold on
plot(waterTemperatureOutInitial.Values)
nexttile
plot(airPressureDrop)
hold on
plot(airPressureDropInitial.Values)
nexttile
plot(waterPressureDrop)
hold on
plot(waterPressureDropInitial.Values)
set(get(tl, 'Children'), 'Title', [])
legend('Data', 'Initial', Location = 'best')
```

Observe the significant discrepancy between the simulation results and the test data. These results are acceptable because the simulation outputs are on the same order of magnitude as the heat exchanger data. The parameter estimation experiment iteratively tunes the four experimental parameters until the error between the model output and the test data is smaller than the estimation tolerance. The figure shows the simulation results.



Set Up the Objective Function

To facilitate the experiment, you must set up an objective function that:

1. Runs the simulation.
2. Gets the simulation results from signal logging.
3. Creates a `sdo.requirements.SignalTracking` object that specifies a tracking requirement for the output signals.
4. Uses the `sdo.requirements.SignalTracking` object to compute the error between the model output and the test data.
5. Returns the error.

The experiment runs the objective function using the current parameters as an argument. As the experiment proceeds, the objective function takes the parameter values of the current iteration and returns the error. The experiment iteratively attempts to reduce the error by adjusting the parameters and running the function to check the updated error measurement, or residual.

Create a function called `estimationObjective.m` using this code, and add it to your path.

See Code

```
function residuals = estimationObjective(parameters, exchangerSimulator, exchangerExperiment)
% The parameter estimation objective function simulates the model with the
% current iteration of parameter values and returns the residuals between
% the simulated model outputs and the test data.

% Simulate the model using parameter values for the current iteration.
exchangerExperiment = setEstimatedValues(exchangerExperiment, parameters);
exchangerSimulator = createSimulator(exchangerExperiment, exchangerSimulator);
exchangerSimulator = sim(exchangerSimulator);

% Get the simulation results from logging.
logsout = find(exchangerSimulator.LoggedData, get_param(exchangerSimulator.ModelName, 'SignalLogg
airTemperatureOut = find(logsout, 'Air Temperature Out (degC)');
waterTemperatureOut = find(logsout, 'Water Temperature Out (degC)');
airPressureDrop = find(logsout, 'Air Pressure Drop (kPa)');
waterPressureDrop = find(logsout, 'Water Pressure Drop (kPa)');

% Create a signal tracking object to evaluate the residuals between the
% simulated model outputs and the test data.
signalTracking = sdo.requirements.SignalTracking('Method', 'Residuals');

% Compute the discrepancies between the simulated model outputs and the test data.
airTemperatureOutError = evalRequirement(signalTracking, airTemperatureOut.Values, exchangerExpe
waterTemperatureOutError = evalRequirement(signalTracking, waterTemperatureOut.Values, exchangerE
airPressureDropError = evalRequirement(signalTracking, airPressureDrop.Values, exchangerExperime
waterPressureDropError = evalRequirement(signalTracking, waterPressureDrop.Values, exchangerExpe

% Return the residuals.
residuals.F = [
    airTemperatureOutError(:);
    waterTemperatureOutError(:);
    airPressureDropError(:);
    waterPressureDropError];
end
```

Perform Parameter Estimation

To begin the parameter estimation experiment, create a function handle that returns the residuals from the `estimationObjective.m` function and pass that information to the `sdo.optimize` function.

Create the function handle `objFcn` to store the residuals using this code:

```
objFcn = @(p) estimationObjective(p, exchangerSimulator, exchangerExperiment);
```

Now you can solve the optimization problem. Pass the function handle to `sdo.optimize`, which evaluates the function at each optimization iteration. Specify the configuration of the optimization experiment using the `sdo.OptimizeOptions` function.

```
options = sdo.OptimizeOptions('Method', 'lsqnonlin', 'OptimizedModel', exchangerModel);
parametersEstimated = sdo.optimize(objFcn, exchangerExperiment.Parameters, options);
```

As Simulink Design Optimization works to achieve a solution, the table results update in the MATLAB command window.

```
  Common Properties:
            Name: 'Air Flow Rate (m^3/hr)'
            Time: [361x1 double]
        TimeInfo: [1x1 tsdata.timemetadata]
            Data: [361x1 double]
        DataInfo: [1x1 tsdata.datametadata]

  More properties, Methods


 Optimization started 24-May-2022 17:01:39

                                                  First-order
 Iter F-count        f(x)      Step-size  optimality
    0      9      501.985            1
    1     18      314.079       0.4605      4.2e+03
    2     27      43.9425        1.942     1.07e+03
    3     36      1.17906        1.096         93.7
    4     45     0.696358       0.1276         21.7
    5     54     0.657429      0.05097         8.13
    6     63     0.649538        0.089         8.99
    7     72     0.646351      0.01974         3.49
    8     81     0.643237     0.003086         3.51
    9     90     0.643237      0.07047         3.51
   10     99     0.643237      0.01762         3.51
   11    108     0.643086     0.004404          5.1
   12    117     0.643086     0.001101          5.1
   13    126     0.643086    0.0002753          5.1
Local minimum possible.
lsqnonlin stopped because the size of the current step is less than
the value of the step size tolerance.

<stopping criteria details>
```

Display the results in a table.

```
table(parametersEstimated.Value, 'VariableNames', {parametersEstimated.Name})
```

```
ans =

  1×4 table

    airSurfaceArea    waterSurfaceArea    airLossCoefficient    waterLossCoefficient
    ______________    ________________    __________________    ____________________

        1.3311            0.21375               10.002                 100.34
```

Import the results into the model and run the simulation.

```
exchangerExperiment = setEstimatedValues(exchangerExperiment, parametersEstimated);
exchangerSimulator = createSimulator(exchangerExperiment, exchangerSimulator);
exchangerSimulator = sim(exchangerSimulator);
```

Retrieve the estimated simulation results.

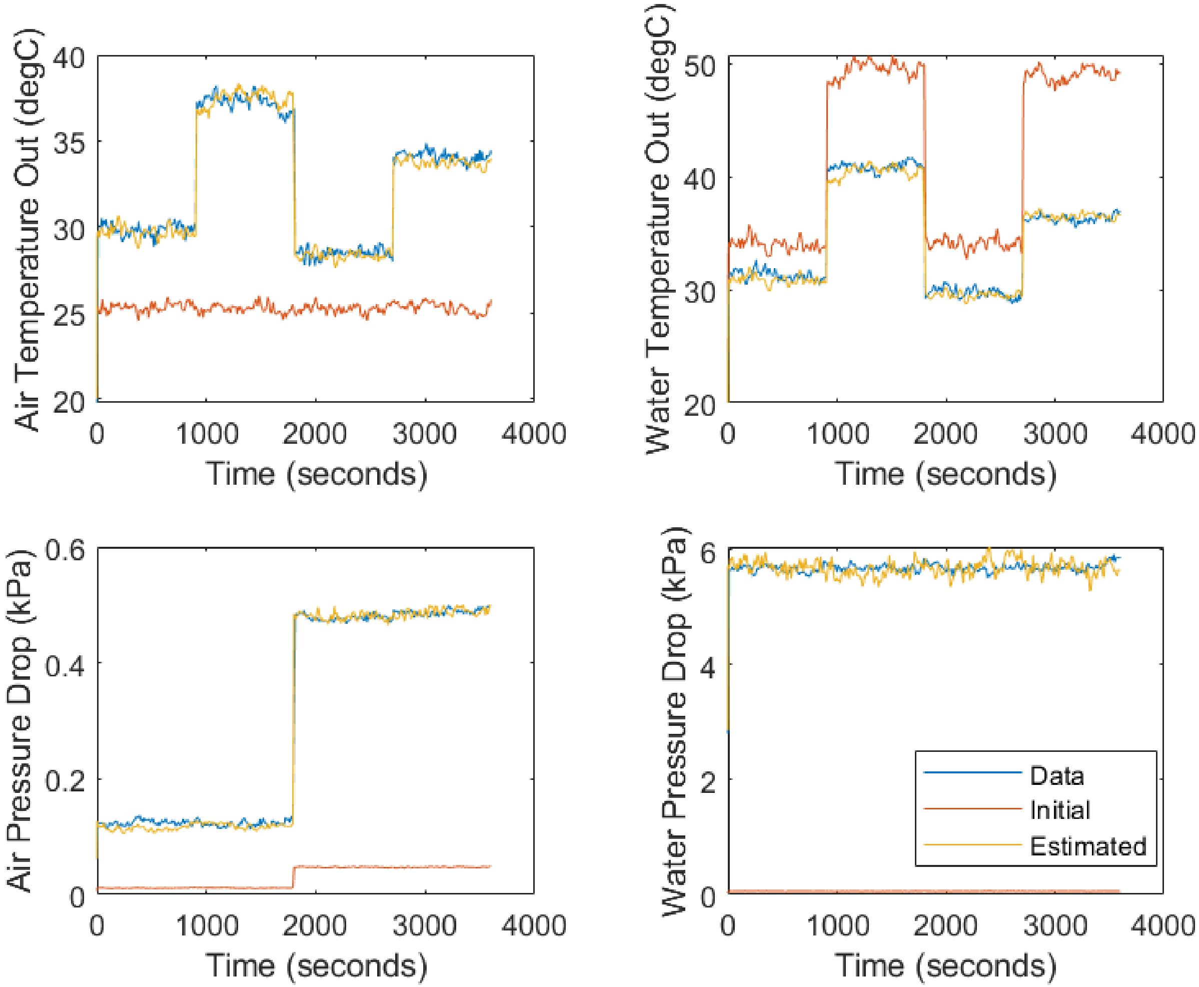
```
logsout = find(exchangerSimulator.LoggedData, get_param(exchangerModel, 'SignalLoggingName'));
airTemperatureOutEstimated = find(logsout, 'Air Temperature Out (degC)');
waterTemperatureOutEstimated = find(logsout, 'Water Temperature Out (degC)');
```

```
airPressureDropEstimated = find(logsout, 'Air Pressure Drop (kPa)');
waterPressureDropEstimated = find(logsout, 'Water Pressure Drop (kPa)');
```

Now you can plot the estimated results against the initial results.

```
nexttile(tl, 1)
plot(airTemperatureOutEstimated.Values)
nexttile(tl, 2)
plot(waterTemperatureOutEstimated.Values)
nexttile(tl, 3)
plot(airPressureDropEstimated.Values)
nexttile(tl, 4)
plot(waterPressureDropEstimated.Values)
legend('Data', 'Initial', 'Estimated', Location = 'best')
```

Observe that the estimated results match the test data closely, accounting for noise. The figure shows the plotted results.



See Also

`sdo.OptimizeOptions` (Simulink Design Optimization) | “About Simscape Run-Time Parameters” | “Improve Parameter-Sweeping Efficiency Using Simscape Run-Time Parameters”

More About

- “How the Software Formulates Parameter Estimation as an Optimization Problem” (Simulink Design Optimization)

How To

Upgrading Hydraulics (Isothermal) Models in Simscape Fluids

Learn how to use the `hydraulicToIsothermalLiquid` tool to upgrade Hydraulics (Isothermal) models. You can convert multiple models at the same time.

The blocks in the Isothermal Liquid library account for density as a function of pressure, and pipe and actuator blocks account for compressibility by default. As a result, expect some differences between a model that uses Hydraulics (Isothermal) library blocks and a converted model when compressibility effects play a significant role due to the treatment of conservation equations in the volumetric-flow-rate-based and mass-flow-rate-based libraries.

These examples show how to run the conversion tool, respond to warning messages, and compare and check converted models with the Simulation Data Inspector.

Automatic Conversion

This example converts the Diesel Engine In-Line Injection System model and requires minimal manual adjustments.

1. Open the Diesel Engine In-Line Injection System model by entering `sh_diesel_injection` at the MATLAB command line.
2. Enable logging by clicking the signal line connected to the Scope block and clicking **Log Signals** on the **Simulation** tab.
3. Save the model as `sh_diesel_injection_hydro.slx` to a location where you have write permissions.
4. Set the MATLAB Current Folder to this location. When you run the conversion, the conversion function saves the converted model and conversion report to this location.
5. Simulate the model.
6. On the **Simulation** tab, click **Data Inspector**.
7. To see one of the logged flow rates, expand **Mux:1** and select **Mux:1(1)** to see the outlet flow rate in the `Injector Pump1` subsystem.
8. Convert the blocks in the model to the isothermal liquid library. At the command prompt, enter `hydraulicToIsothermalLiquid('sh_diesel_injection_hydro')`.
9. The conversion report, `sh_diesel_injection_hydro_converted.html`, opens when the model conversion completes. The report includes three types of warning messages:

   - Predefined fluid has been reparameterized. Behavior change not expected at most temperatures.

     This message indicates that the conversion process redefined the fluid for the Isothermal Liquid library in the Fuel Properties block in `sh_diesel_injection_hydro_converted` differently, but no specific parameter requires user input.
   - Critical Reynolds number set to 150. Behavior change not expected.

     The conversion process converted the parameter that indicates flow regime to the **Critical Reynolds number** parameter. This parameter has a default value of 150. You do not need to adjust anything in the converted model.
   - Original block had Specific heat ratio of 1.4 Set Air polytropic index to this value in an Isothermal Liquid Properties (IL) block or Isothermal Liquid Predefined Properties (IL) block.

This message indicates that the conversion changed the value of the specific heat ratio parameter in the Fuel Properties block.

10 To address the message in the report, in the `sh_diesel_injection_hydro_converted` model, open the Fuel Properties block. In the **Entrained Air** section, set the **Air polytropic index** parameter to `1.4`

11 Run `sh_diesel_injection_hydro_converted`.

12 Open the Simulation Data Inspector and select **Compare** in the left pane. Set **Baseline** to `sh_diesel_injection_hydro`. Set **Compare to** to `sh_diesel_injection_hydro_converted`.

13 Click **More**. Under **Global Tolerances**, set **Absolute** to `1-e5`. Set **Time** to `1e-4`.

14 Click **Compare**. In the left pane, the four green check marks next to the four Mux signals show that the results of the models agree within the specified tolerances.

Conversion with Manual Adjustments

This example uses a modified dual counterbalance valve model. This conversion requires more manual adjustment steps.

1 Open the `sh_HtoIL_dual_counterbalance_start` model. This model represents a modified dual counterbalance valve. Data logging for the pump angular velocity and cylinder position is already enabled in this model.

2 Save the model as `sh_HtoIL_dual_counterbalance_hydro_start` in a location where you have write permissions.

3 Set the MATLAB Current Folder to this location.

4 Run the model.

5 Convert the blocks in the model to the isothermal liquid library. At the command line, enter `hydraulicToIsothermalLiquid('sh_HtoIL_dual_counterbalance_hydro_start')`.

6 Run the converted model, `sh_HtoIL_dual_counterbalance_hydro_start_converted`.

7 The simulation returns the error: `Invalid use of -. At least one of the operands must be scalar or the operands must be the same size. The units of the operands must be commensurate`. Click the hyperlink below the `Caused by` message to navigate to the PS Subtract block.

   In the Explorer Bar, click **Pipe A** to see the PS Constant blocks that are the inputs to the PS Subtract block.

   The two PS Constant blocks attached to the PS Subtract block have different units. The signal from the `B Elevation` block to **el_B** has units of `1`, while the signal from the `A Elevation` block to **el_A** has units of `m`. In the `B Elevation` block, change the units for **Constant** to `m`.

8 Run the model.

9 In the **Simulation** tab, click **Data Inspector** to compare the pump angular velocity and cylinder position between the two models.

   Select **Compare** in the left pane. Set **Baseline** to `sh_HtoIL_dual_counterbalance_hydro_start`. Set **Compare to** to `sh_HtoIL_dual_counterbalance_hydro_start_converted`.

10 Click **More**. Under **Global Tolerances** set **Absolute** and **Relative** to `0.05`.

11 Click **Compare**. Notice that the Cylinder Position signal agrees, but the Pump Rotational Velocity signal does not. When you convert a model from the Hydraulic to the Isothermal Liquid domain, the model may show some discrepancies in behavior due to how the isothermal liquid blocks model compressibility and behave in off-design conditions. Addressing the messages in the warning report may reduce discrepancies.

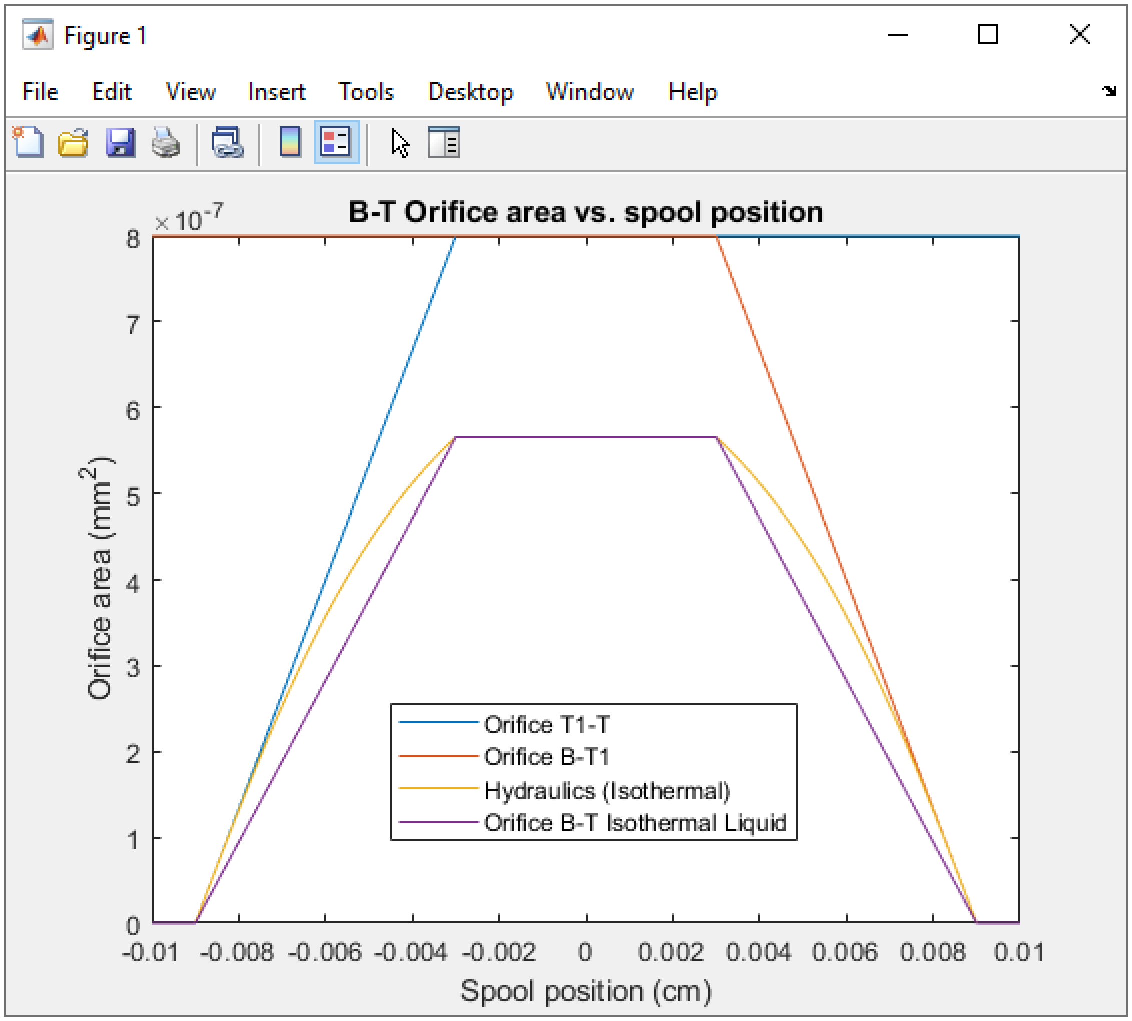
12 Check the conversion report, `sh_HtoIL_dual_counterbalance_hydro_start_converted.html`. Some messages listed in the report do not require any changes in the converted model:

- Critical Reynolds number set to 150. Behavior change not expected.

  The conversion changed the parameter that indicates flow regime to the **Critical Reynolds number** parameter. This parameter has a default value of 150. You do not need to adjust the converted model.

- B-T Orifice reparameterized.

  After conversion, the directional valves in the Isothermal Liquid library that open in the neutral spool position switch from two orifices in series to one orifice between ports. This change results in a modification to the orifice area when approaching the orifice maximum area, as shown below. You do not need to modify the model.



Note that this warning applies to orifices in the 4-Way 3-Position Directional Valve (IL) that open in the neutral spool position. This warning does not affect the P-B Orifice, which opens only in the positive and negative spool positions.

- Pressure losses due to kinetic energy change added. Adjustment of Expansion correction factor and Contraction correction factor may be required.

  If the area change is large, which means that the pressure change due to the second term and power adjustment are within the tolerance of the model, you do not need to adjust the model. If the contraction is small, then you can tune the model by adjusting the smaller or both of the port areas. In the Hydraulics (Isothermal) block, the pressure loss is modeled as

$$\Delta p \approx \frac{K_{PC}\dot{m}^2}{4\rho A_{small}^2}\left(1 - \frac{A_{small}}{A_{large}}\right)^{0.75}.$$

  In the Isothermal Liquid block, the pressure loss is

$$\Delta p \approx \frac{K_{contraction}\dot{m}^2}{4\bar{\rho}A_{small}^2}\left(1 - \frac{A_{small}}{A_{large}}\right) + \frac{\dot{m}^2}{2\bar{\rho}A_{small}^2}\left(1 - \frac{A_{small}}{A_{large}}\right).$$

  In this model, you do not need to adjust these values.

- Contraction loss coefficient reformulated. Adjustment of Contraction correction factor may be required.

  The **Contraction correction factor** parameter is similar to the **Expansion correction factor** parameter. You only need to adjust the parameter value if the area change is large. In this model, you do not need to adjust these values.

13 The remaining messages in `sh_HtoIL_dual_counterbalance_hydro_start_converted.html` require adjustments to the converted model:

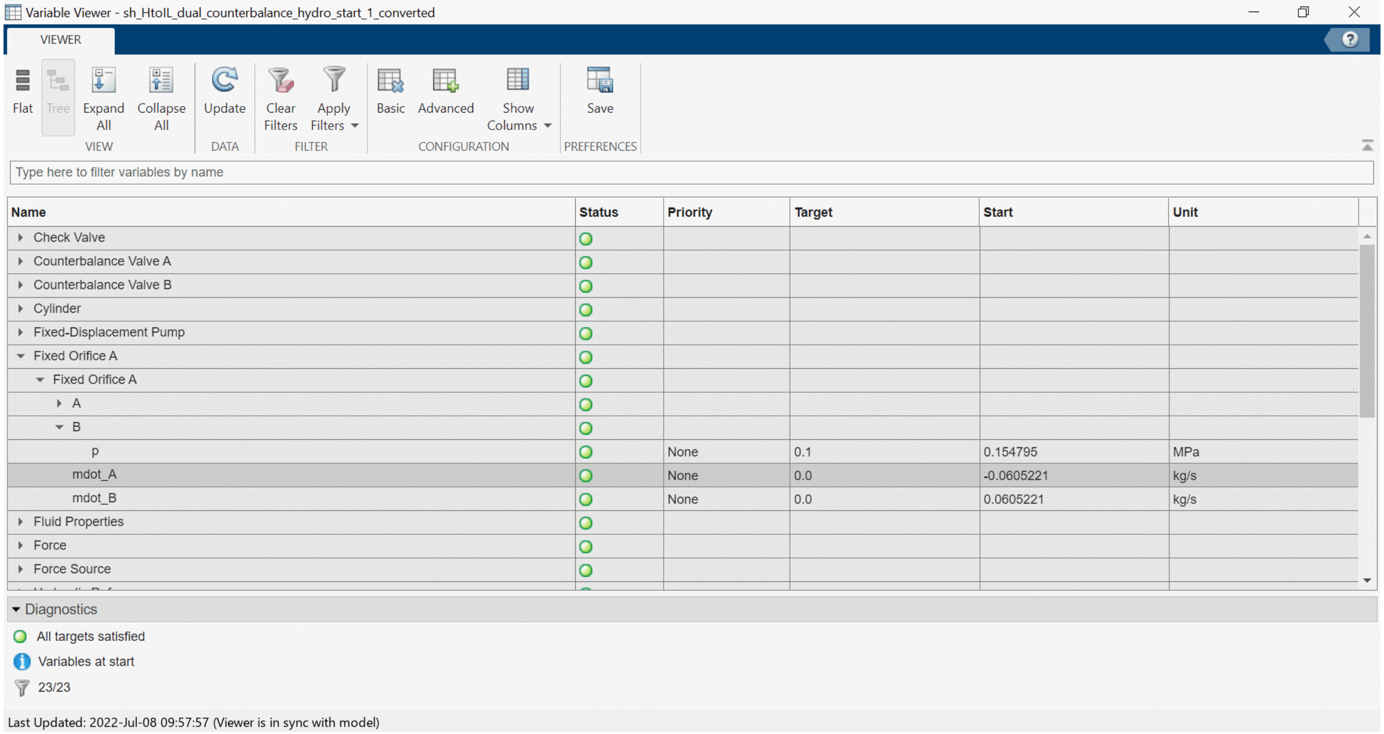
- Original block had Specific heat ratio of 1.4. Set Air polytropic index to this value in an Isothermal Liquid Properties (IL) block or Isothermal Liquid Predefined Properties (IL) block.

  In the Fluid Properties block, in the **Entrained Air** section, set the **Air polytropic index** parameter to `1.4`.

- Beginning value of Flow rate removed. Adjustment of model initial conditions may be required.

  This message indicates that you cannot set the priority and target value of the specified block. Set the priority and target of a variable in an adjacent block.

  - In `sh_HtoIL_dual_counterbalance_hydro_start`, check the initial conditions of the `Fixed Orifice A` block. In the block dialog, under the **Initial targets** section, the flow rate is `0 m^3/s` with **Priority** set to `High`.
  - In `sh_HtoIL_dual_counterbalance_hydro_start_converted`, open the Variable Viewer to check the initial conditions of the `Fixed Orifice A`. The mass flow rate initializes to a nonzero value.



Name	Status	Priority	Target	Start	Unit
Check Valve	○				
Counterbalance Valve A	○				
Counterbalance Valve B	○				
Cylinder	○				
Fixed-Displacement Pump	○				
Fixed Orifice A	○				
Fixed Orifice A	○				
A	○				
B	○				
p	○	None	0.1	0.154795	MPa
mdot_A	○	None	0.0	-0.0605221	kg/s
mdot_B	○	None	0.0	0.0605221	kg/s
Fluid Properties	○				
Force	○				
Force Source	○				

Diagnostics

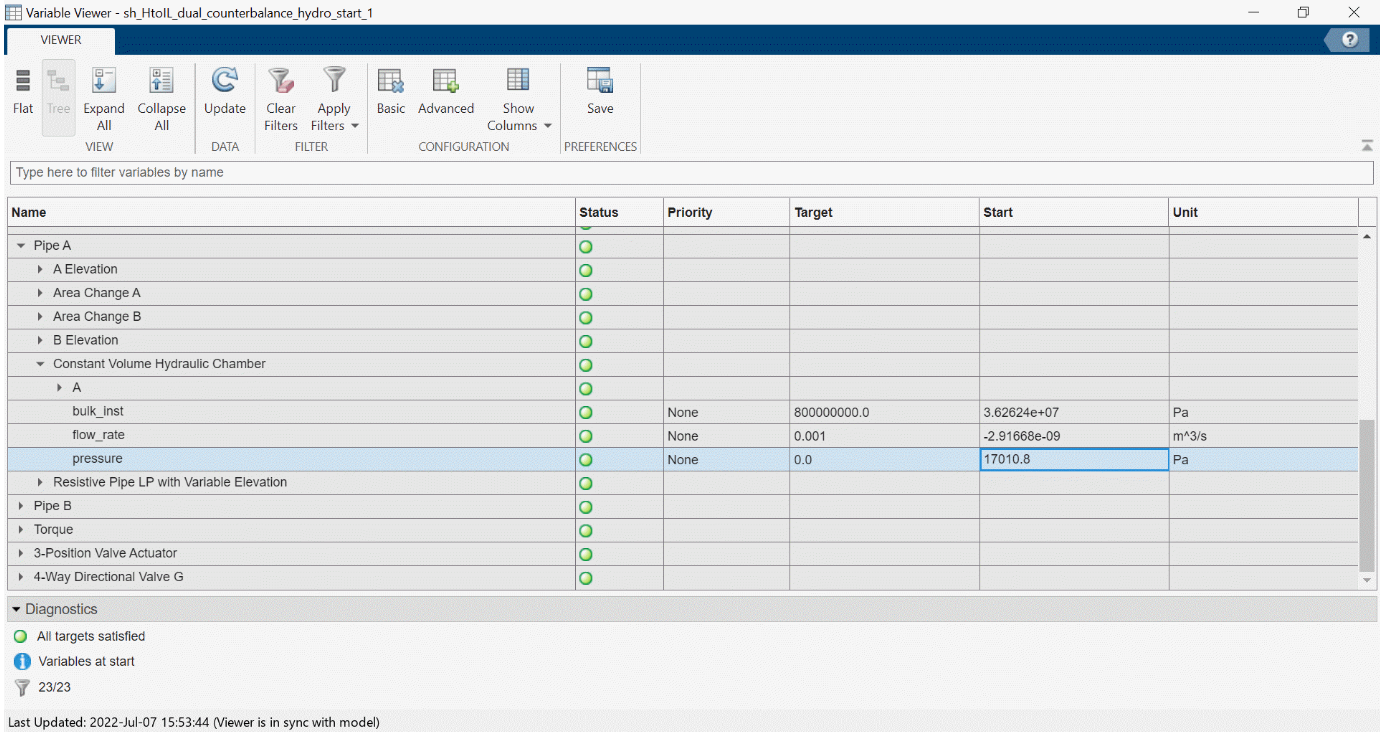
○ All targets satisfied

Variables at start

23/23

Last Updated: 2022-Jul-08 09:57:57 (Viewer is in sync with model)

- In `sh_HtoIL_dual_counterbalance_hydro_start`, open the Variable Viewer. The initial value of the pressure for the Constant Volume Hydraulic Chamber is `17010.8 Pa`.



Name	Status	Priority	Target	Start	Unit
Pipe A	○				
A Elevation	○				
Area Change A	○				
Area Change B	○				
B Elevation	○				
Constant Volume Hydraulic Chamber	○				
A	○				
bulk_inst	○	None	800000000.0	3.62624e+07	Pa
flow_rate	○	None	0.001	-2.91668e-09	m^3/s
pressure	○	None	0.0	17010.8	Pa
Resistive Pipe LP with Variable Elevation	○				
Pipe B	○				
Torque	○				
3-Position Valve Actuator	○				
4-Way Directional Valve G	○				

Diagnostics

○ All targets satisfied

Variables at start

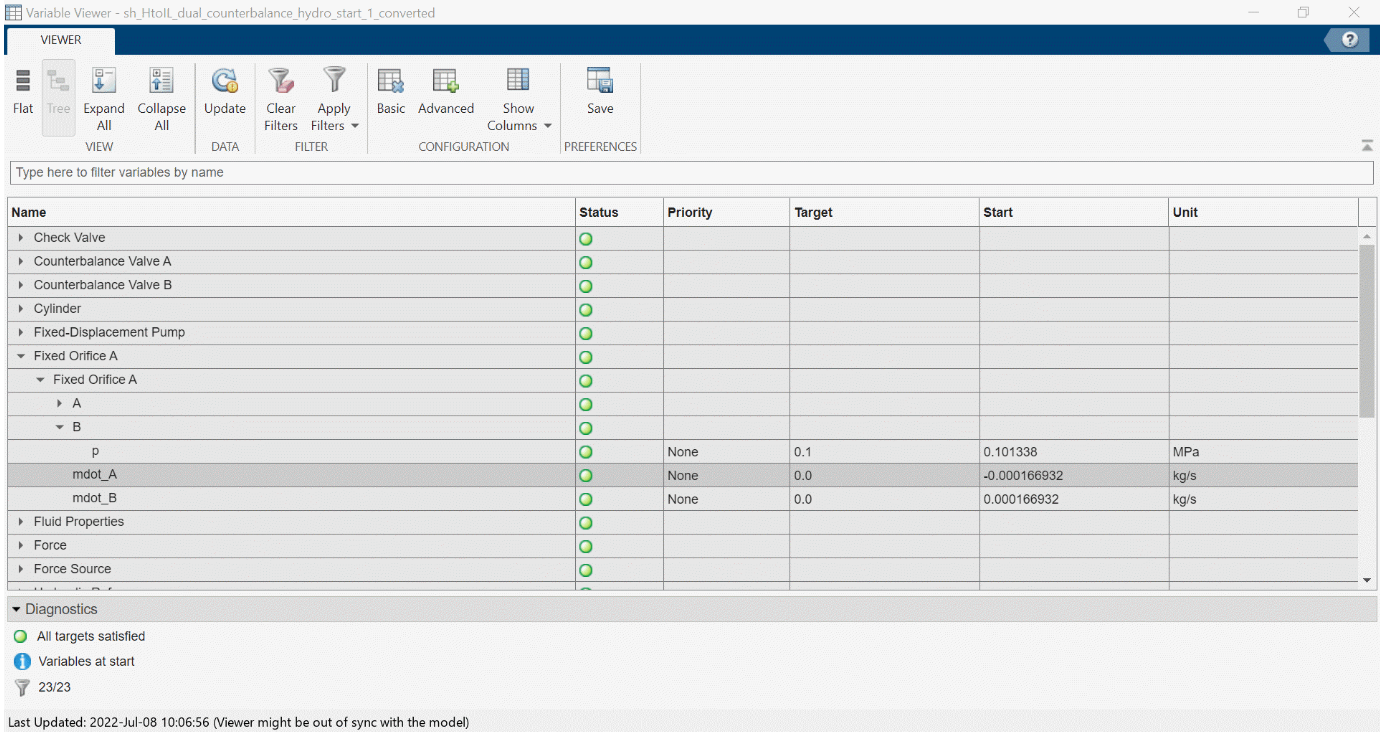
23/23

Last Updated: 2022-Jul-07 15:53:44 (Viewer is in sync with model)

- Set the converted model to match this value. In the `Pipe A` subsystem of `sh_HtoIL_dual_counterbalance_hydro_start_converted`, in the `Constant`

Volume Hydraulic Chamber dialog box, under the **Initial targets** section, set the **Pressure of liquid volume** parameter **Priority** to High and **Value** to 17010.8 + 101325 Pa.

- Run the converted model. The Variable Viewer shows that the mass flow rate in the Fixed Orifice A block starts at a value of -1.6e-4 kg/s. This value is not a perfect match to 0 kg/s, but it is closer than before.



Name	Status	Priority	Target	Start	Unit
Check Valve	○				
Counterbalance Valve A	○				
Counterbalance Valve B	○				
Cylinder	○				
Fixed-Displacement Pump	○				
Fixed Orifice A	○				
Fixed Orifice A	○				
A	○				
B	○				
p	○	None	0.1	0.101338	MPa
mdot_A	○	None	0.0	-0.000166932	kg/s
mdot_B	○	None	0.0	0.000166932	kg/s
Fluid Properties	○				
Force	○				
Force Source	○				

Diagnostics

○ All targets satisfied

ⓘ Variables at start

23/23

Last Updated: 2022-Jul-08 10:06:56 (Viewer might be out of sync with the model)

- Interpolation method changed to Linear. Additional elements in the Reynolds number vector, Contraction loss coefficient vector, and Expansion loss coefficient vector may be required.

  You can add additional elements to the Reynolds number and loss coefficient vectors in the command line by using the interp1 function.

  In the hydraulics model, the workspace variable for the **Reynolds number vector** parameter is defined as Re_vec and the **Loss coefficient vector** parameter is defined as loss_coeff_vec. In the MATLAB command window, enter:

```
Re_vec_smooth = -6000:100:6000;
loss_coeff_vec_smooth = interp1(Re_vec, loss_coeff_vec, Re_vec_smooth, 'makima', 'extrap')
```

  In the Pipe A subsystem, replace Re_vec with Re_vec_smooth and loss_coeff_vec with loss_coeff_vec_smooth in the Area Change B block dialog:

  - Set the **Reynolds number vector** parameter to Re_vec_smooth(Re_vec_smooth>0).
  - Set the **Contraction loss coefficient vector** parameter to [interp1(-fliplr(Re_vec_smooth(Re_vec_smooth<0)), fliplr(loss_coeff_vec_smooth(Re_vec_smooth<0)), Re_vec_smooth(Re_vec_smooth>0), 'linear', loss_coeff_vec_smooth(1))].
  - Set the **Expansion loss coefficient vector** parameter to loss_coeff_vec_smooth(Re_vec_smooth>0).

The command extrapolates the data set linearly to *Re = 6000* and applies a smooth interpolation method the next time the simulation runs.

- Only elements greater than or equal to 0 retained in Reynolds number vector. Expansion loss coefficient values mapped to these Reynolds numbers. Adjustment of Reynolds number vector, Contraction loss coefficient vector, and Expansion loss coefficient vector may be required.

  The **Reynolds number vector** in the original `Area Change B` block has both negative and positive Reynolds numbers, `[-6000 -4000 -1000 -200 -50 -30 -20 -10 -1 1 20 40 100 500 2000 5000]`, but the converted `Area Change B` block only preserves positive Reynolds numbers. The converted `Area Change B` block in the `Pipe A` subsystem symmetrically extends the positive elements of the user-provided Reynolds number vector to negative Reynolds numbers. The converted block also separates and extends the loss coefficient vector to individual expansion and contraction loss coefficient vectors. To match the original model behavior, you need to extend the data set to *Re = 6000* in the base workspace. The commands in the previous step also addressed this warning.
- Nominal fluid density and kinematic viscosity removed. Pump uses network fluid properties. Adjustment of Volumetric efficiency at nominal conditions may be required.

  In the Hydraulics (Isothermal) library, you can define density and viscosity specifically for pump or motor blocks. Isothermal Liquid blocks use network properties to model these values. To account for this change, adjust the **Volumetric efficiency at nominal conditions** parameter in the Fixed-Displacement Pump (IL) block:

  - In the Fixed-Displacement Pump (IL) dialog, set the **Volumetric efficiency at nominal conditions** parameter to `0.957`, which is determined using the formula

$$\eta_{IL,\, equiv} = 1 - \frac{\nu_{H,\, nom}\rho_{H,\, nom}}{\nu_{IL}\rho_{IL}}(1 - \eta_H),$$

    where:

    - $\nu_{H,nom}$ is the value of the **Nominal kinematic viscosity** parameter in the Fixed-Displacement Pump block.
    - $\rho_{H,nom}$ is the value of the **Nominal fluid density** parameter in the Fixed-Displacement Pump block.
    - η_H is the value of the **Volumetric efficiency at nominal conditions** parameter in the Fixed-Displacement Pump block.
    - ν_{IL} is the value of the **Kinematic viscosity at atmospheric pressure** parameter in the Isothermal Liquid Properties (IL) block.
    - ρ_{IL} is the value of the **Density at atmospheric pressure (no entrained air)** parameter in the Isothermal Liquid Properties (IL) block.

14 Save and run `sh_HtoIL_dual_counterbalance_hydro_converted`.

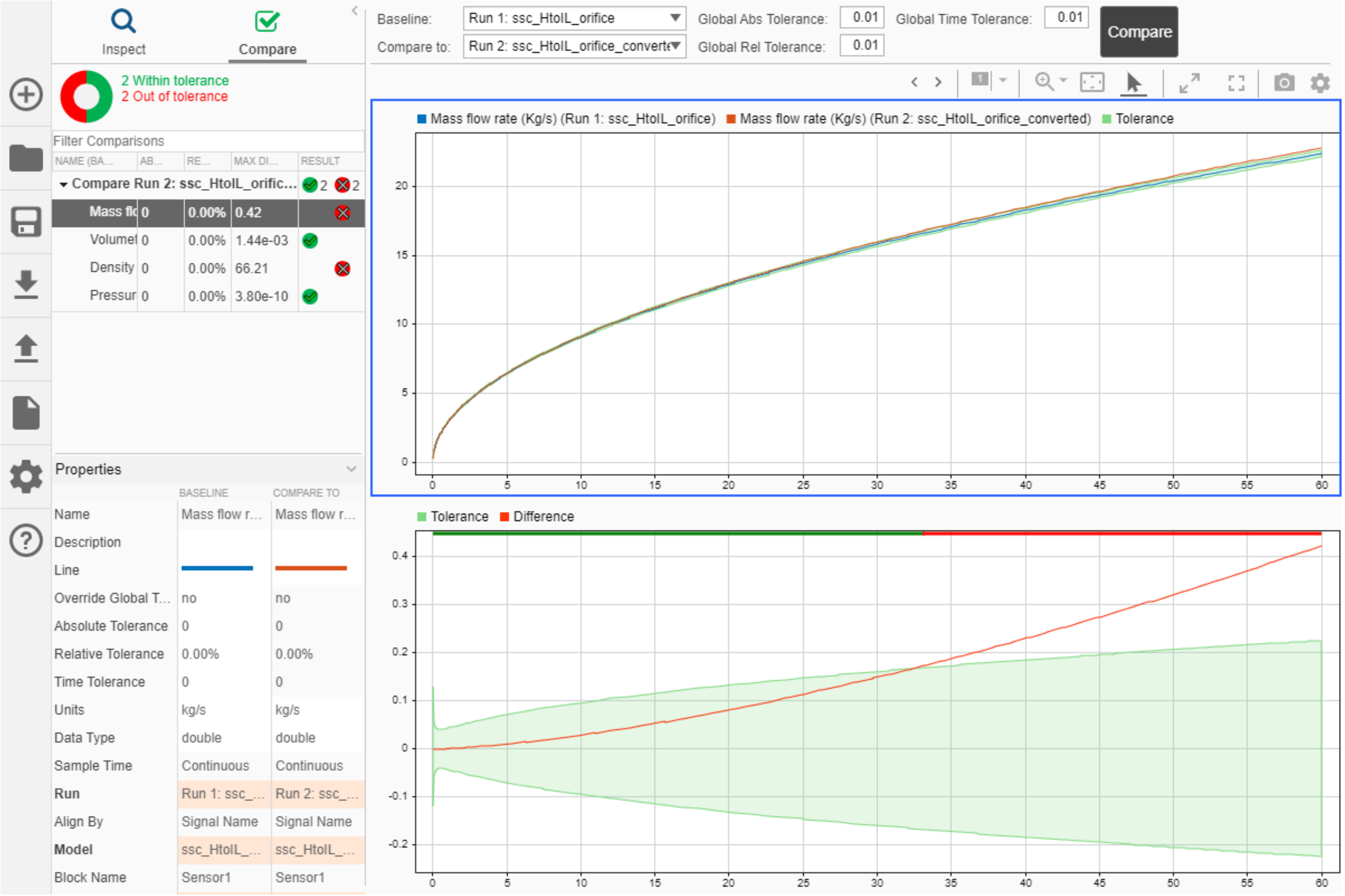
To see a model that contains all of the post-conversion updates, open `sh_HtoIL_dual_counterbalance_end`, which includes all post-conversion updates. Inspect the model signals with the Simulation Data Inspector. The Pump Rotational Velocity signals are now within the tolerance for the majority of the time. There are still some expected discrepancies when fast dynamics occur.

Conversion for Models that Include Fluid Dynamic Compressibility

Unlike the Hydraulics (Isothermal) library, all blocks in the Isothermal Liquid library account for density as a function of pressure. In some converted models, this can cause diverging results, which this example illustrates.

1. Open the `ssc_HtoIL_orifice` model.
2. Save the model as `ssc_HtoIL_hydro.mdl` to a location where you have write permissions.
3. Set the MATLAB Current Folder to this location. The converted model and the conversion report will be saved here.
4. Run the model.
5. Convert the blocks in the model to the isothermal liquid library. At the command prompt, enter `hydraulicToIsothermalLiquid('ssc_HtoIL_hydro')`.
6. Run the converted model.
7. Open the Simulation Data Inspector and select **Compare** in the left pane. Set **Baseline** to `ssc_HtoIL_hydro`. Set **Compare to** to `ssc_HtoIL_hydro_converted`.
8. Click **More**. Under **Global Tolerances** set **Absolute**, **Relative**, and **Time** to `0.01`.
9. Click **Compare**.

   As the applied pressure increases, the mass flow rates in the Hydraulics (Isothermal) and Isothermal Liquid models diverge.


Inspect
Compare
Compare
■ Mass flow rate (Kg/s) (Run 1: ssc_HtolL_orifice) ■ Mass flow rate (Kg/s) (Run 2: ssc_HtolL_orifice_converted) ■ Tolerance
20
15
10
5
0
0
5
10
15
20
25
30
35
40
45
50
55
60
■ Tolerance ■ Difference
0.4
0.3
0.2
0.1
0
-0.1
-0.2
0
5
10
15
20
25
30
35
40
45
50
55
60

Baseline: Run 1: ssc_HtolL_orifice

Compare to: Run 2: ssc_HtolL_orifice_converte

Global Abs Tolerance: 0.01

Global Rel Tolerance: 0.01

Global Time Tolerance: 0.01

2 Within tolerance
2 Out of tolerance

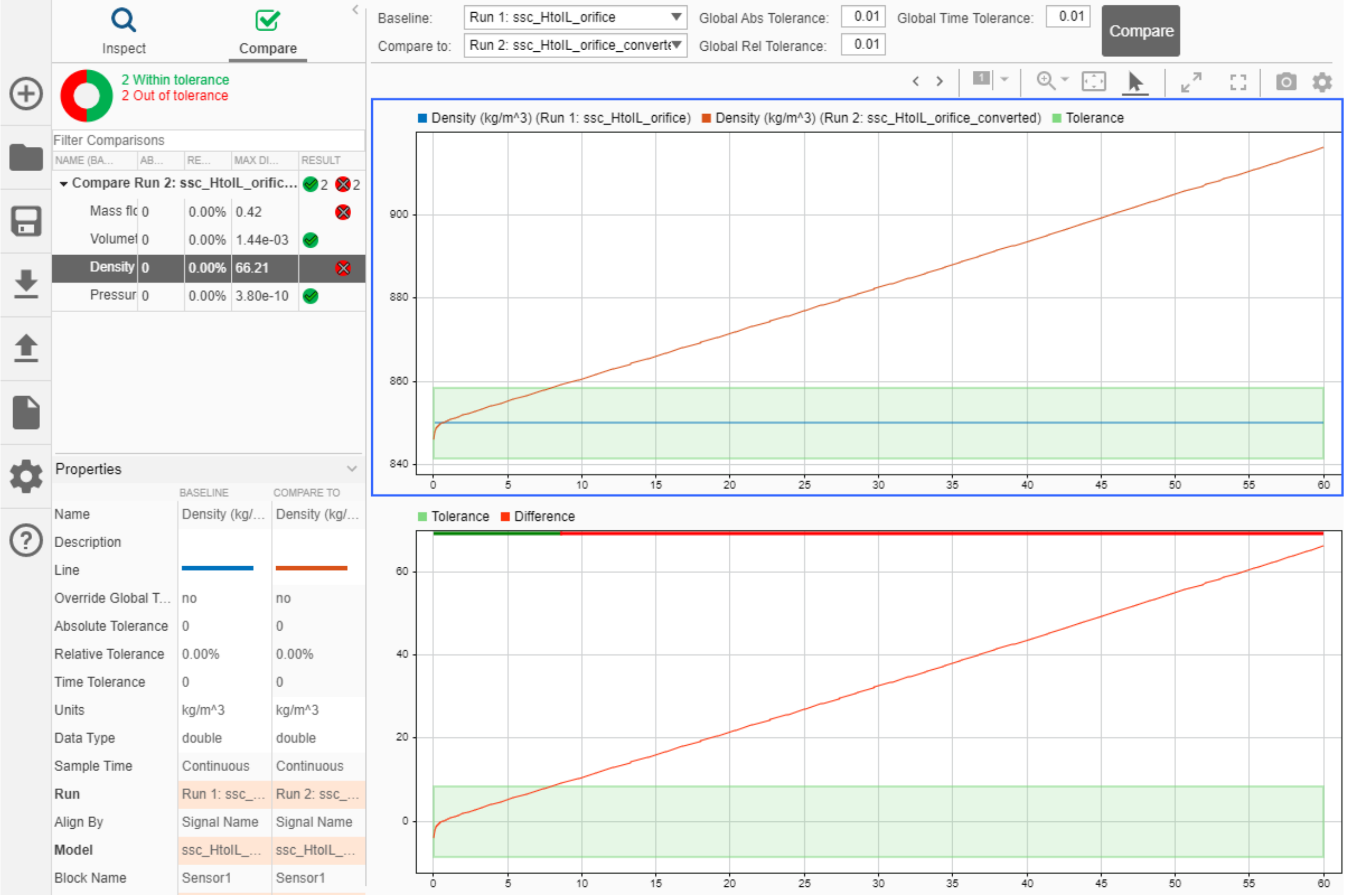
Filter Comparisons

NAME (BA...	AB...	RE...	MAX DI...	RESULT
▾ Compare Run 2: ssc_HtolL_orific...				✓ 2 ✗ 2
Mass flo	0	0.00%	0.42	✗
Volumet	0	0.00%	1.44e-03	✓
Density	0	0.00%	66.21	✗
Pressur	0	0.00%	3.80e-10	✓

Properties

	BASELINE	COMPARE TO
Name	Mass flow r...	Mass flow r...
Description		
Line		
Override Global T...	no	no
Absolute Tolerance	0	0
Relative Tolerance	0.00%	0.00%
Time Tolerance	0	0
Units	kg/s	kg/s
Data Type	double	double
Sample Time	Continuous	Continuous
Run	Run 1: ssc_...	Run 2: ssc_...
Align By	Signal Name	Signal Name
Model	ssc_HtolL_...	ssc_HtolL_...
Block Name	Sensor1	Sensor1

This divergence is due to the fact that density is constant in the Hydraulics (Isothermal) block and varies as a function of pressure in the Isothermal Liquid block.



Baseline: Run 1: ssc_HtoIL_orifice

Compare to: Run 2: ssc_HtoIL_orifice_converte

Global Abs Tolerance: 0.01

Global Rel Tolerance: 0.01

Global Time Tolerance: 0.01

2 Within tolerance
2 Out of tolerance

NAME (BA...	AB...	RE...	MAX DI...	RESULT
Compare Run 2: ssc_HtoIL_orific...				✓2 ☒2
Mass flo	0	0.00%	0.42	☒
Volumet	0	0.00%	1.44e-03	✓
Density	0	0.00%	66.21	☒
Pressur	0	0.00%	3.80e-10	✓

	BASELINE	COMPARE TO
Name	Density (kg/...	Density (kg/...
Description		
Line	🟦	🟥
Override Global T...	no	no
Absolute Tolerance	0	0
Relative Tolerance	0.00%	0.00%
Time Tolerance	0	0
Units	kg/m^3	kg/m^3
Data Type	double	double
Sample Time	Continuous	Continuous
Run	Run 1: ssc_...	Run 2: ssc_...
Align By	Signal Name	Signal Name
Model	ssc_HtoIL_...	ssc_HtoIL_...
Block Name	Sensor1	Sensor1

Depending on your application, your converted model may require tuning to account for density changes due to changes in pressure or temperature.

See Also

`hydraulicToIsothermalLiquid` | `hydraulicToIsothermalLiquidPostProcess` | **Simulation Data Inspector**

More About

- "How to Upgrade Hydraulic Models"
- "Modeling Isothermal Liquid Systems"

Block Substitutions for Simscape Fluids Hydraulics (Isothermal) Blocks

The `hydraulicToIsothermalLiquid` conversion tool replaces hydraulic blocks in your model with the corresponding isothermal liquid blocks. This table lists the Hydraulics (Isothermal) library blocks and their associated Isothermal Liquid library blocks. To match the Hydraulics (Isothermal) block configuration, some converted models may include a subsystem with additional Simscape Fluids or Simscape blocks. Some blocks in the Hydraulics (Isothermal) library do not have an equivalent Isothermal Liquid library block. In this case, you can reconstruct the functionality from a collection of Simscape and Simscape Fluids blocks.

Accumulators

Hydraulics (Isothermal) Block	Isothermal Liquid Block
Gas-Charged Accumulator	Gas-Charged Accumulator (IL)
Spring-Loaded Accumulator	Spring-Loaded Accumulator (IL) The isothermal liquid block does not model hard-stop damping.

Hydraulic Cylinders

Hydraulics (Isothermal) Block	Isothermal Liquid Block
Centrifugal Force in Rotating Cylinder	Rotating Cylinder Force (IL) The **Fluid density** parameter is replaced with port **X**, which senses the network density.
Cylinder Cushion	Cylinder Cushion (IL) The Cylinder Cushion (IL) block is parameterized by plunger diameter, area, and length. The Hydraulics (Isothermal) block is parameterized by tabulated area and displacement. The conversion tool calculates the plunger length, or the orifice control member travel, from the Hydraulics (Isothermal) block as `opening(end) - opening(1)`.
Cylinder Friction	Cylinder Friction (IL) The Cylinder Friction (IL) block does not assign beginning values to variables.
Single-Acting Rotary Actuator	Single-Acting Rotary (IL) The conversion tool converts the Single-Acting Rotary Actuator block to a subsystem. If the Hydraulics (Isothermal) block models leakage, or the **Leak coefficient** is nonzero, port **A** of the Single-Acting Rotary (IL) connects to port **A** of a Laminar Leakage (IL) block, which connects to a Reservoir (IL) block at atmospheric pressure.
Double-Acting Rotary Actuator	Double-Acting Rotary Actuator (IL) The conversion tool converts the Double-Acting Rotary Actuator to a subsystem. If the Hydraulics (Isothermal) block models leakage, or the **Leak coefficient** is nonzero, ports **A** and **B** of the Double-Acting Rotary Actuator (IL) block connect in parallel to ports **A** and **B**, respectively, of a Laminar Leakage (IL) block.
Single-Acting Hydraulic Cylinder	Single-Acting Actuator (IL)
Single-Acting Hydraulic Cylinder (Simple)	Single-Acting Actuator (IL) The conversion tool sets the **Hard stop model** parameter to `Stiffness and damping applied smoothly through transition region, damped rebound`, with default hard stop parameters.
Double-Acting Hydraulic Cylinder	Double-Acting Actuator (IL)

Hydraulics (Isothermal) Block	Isothermal Liquid Block
Double-Acting Hydraulic Cylinder (Simple)	Double-Acting Actuator (IL) The conversion tool sets the **Hard stop model** parameter to `Stiffness and damping applied smoothly through transition region, damped rebound`, with default hard stop parameters.

Utilities

Hydraulics (Isothermal) Block	Isothermal Liquid Block
Hydraulic Fluid	If the **Hydraulic fluid** parameter of the Hydraulic Fluid block is `Water`, `Water-glycol 60/40`, `Diesel fuel`, or `Jet fuel`, the conversion tool converts the Hydraulic Fluid block to an Isothermal Liquid Predefined Properties (IL) block. For all other fluids, the conversion tool converts the Hydraulic Fluid block to an Isothermal Liquid Properties (IL) block. If the conversion tool cannot assess the fluid temperature definition, defines the properties for 20°C.
Reservoir	Reservoir (IL)

Hydraulic Resistances

Hydraulics (Isothermal) Block	Isothermal Liquid Block
Elbow	Elbow (IL)
Gradual Area Change	Area Change (IL) The conversion tool converts the Gradual Area Change block to a subsystem to maintain the original port orientation.
Local Resistance	Local Resistance (IL)
Pipe Bend	Pipe Bend (IL) The **Initial liquid pressure** parameter in the Pipe Bend block is in gauge. In the Pipe Bend (IL) block, this parameter is in absolute pressure. The conversion tool converts the **Critical Reynolds number** parameter into two internally fixed threshold Reynolds numbers in the Pipe Bend (IL), where 2000 indicates a fully laminar flow and 4000 indicates a fully turbulent flow.
Sudden Area Change	Area Change (IL)
T-Junction	The conversion tool converts the T-Junction block to a subsystem with Local Resistance (IL) blocks.

Low Pressure Blocks

Hydraulics (Isothermal) Block	Isothermal Liquid Block
Hydraulic Pipe LP Resistive Pipe LP Resistive Pipe LP with Variable Elevation Segmented Pipe LP	Pipe (IL)
Hydraulic Pipe LP with Variable Elevation	Pipe (IL) Port **EL** is exposed for elevation change as a physical signal input.
Partially Filled Vertical Pipe LP	Partially Filled Pipe (IL) The Partially Filled Pipe (IL) block receives a liquid level instead of a liquid volume at its physical signal port. The conversion tool add a PS Constant block to divide the original fluid volume by a constant tank cross-section within a subsystem. Modify the assumed tank cross-section in the added PS Constant block.
Constant-Head Tank	The conversion tool converts the Constant-Head Tank block to a subsystem that includes Reservoir (IL), Orifice (IL), PS Integrator, and Flow Rate Sensor blocks.
Tank	Tank (IL)

Orifices

Hydraulics (Isothermal) Block	Isothermal Liquid Block
Annular Orifice	Annular Leakage (IL)
Fixed Orifice Fixed Orifice With Fluid Inertia Fixed Orifice Empirical Variable Orifice	Orifice (IL)
Journal Bearing Pressure-Fed	The conversion tool converts the Journal Bearing Pressure-Fed block to a subsystem that includes two Annular Leakage (IL) blocks.
Orifice With Variable Area Round Holes Orifice With Variable Area Slot	Spool Orifice (IL)
Variable Orifice Between Round Holes	Variable Overlapping Orifice (IL)

Pipes

Hydraulics (Isothermal) Block	Isothermal Liquid Block
Rotating Pipe	Rotating Channel (IL)
Hydraulic Pipeline	Pipe (IL)
Segmented Pipeline	Pipe (IL)

Pumps and Motors

Hydraulics (Isothermal) Block	Isothermal Liquid Block
Angle Sensor	The conversion tool converts the Angle Sensor block to a subsystem that contains an Ideal Rotational Motion Sensor block.
Centrifugal Pump	Centrifugal Pump (IL)
Fixed-Displacement Motor	Fixed-Displacement Motor (IL) The Fixed-Displacement Motor (IL) block uses the network fluid properties. You may need to modify the **Volumetric efficiency at nominal conditions** parameter to match the Hydraulics (Isothermal) block functionality.
Fixed-Displacement Pump	Fixed-Displacement Pump (IL) The Fixed-Displacement Pump (IL) block uses the network fluid properties. You may need to modify the **Volumetric efficiency at nominal conditions** parameter to match the Hydraulics (Isothermal) block functionality.
Jet Pump	Jet Pump (IL)
Porting Plate Variable Orifice	Valve Plate Orifice (IL)
Swash Plate	Swash Plate
Variable-Displacement Motor	Variable Displacement Motor (IL) The Variable Displacement Motor (IL) block uses the network fluid properties. You may need to modify the **Volumetric efficiency at nominal conditions** parameter to match the Hydraulics (Isothermal) block functionality.
Variable-Displacement Pressure-Compensated Pump	Pressure-Compensated Pump (IL) The conversion tool converts the Variable-Displacement Pressure-Compensated Pump block to a subsystem. The Pressure-Compensated Pump (IL) block uses the network fluid properties. You may need to modify the **Volumetric efficiency at nominal conditions** parameter to match the Hydraulics (Isothermal) block functionality.
Variable-Displacement Pump	Variable Displacement Pump (IL) The Variable Displacement Pump (IL) block uses the network fluid properties. You may need to modify the **Volumetric efficiency at nominal conditions** parameter to match the Hydraulics (Isothermal) block functionality.

Directional Valves

Hydraulics (Isothermal) Block	Isothermal Liquid Block
2-Way Directional Valve	2-Way Directional Valve (IL) The 2-Way Directional Valve (IL) block behavior and parameterization are based on the spool position at the maximum orifice area and the maximum spool travel distance, while the 2-Way Directional Valve uses initial spool position. Changing the valve parameterization after conversion can lead to inconsistencies. See “Upgrade Considerations When Converting Hydraulic to Isothermal Liquid Models”.
3-Way Directional Valve	3-Way Directional Valve (IL) The 3-Way Directional Valve (IL) block behavior and parameterization are based on the spool position at the maximum orifice area and the maximum spool travel distance, while the 3-Way Directional Valve uses initial spool position. Changing the valve parameterization after conversion can lead to inconsistencies. See “Upgrade Considerations When Converting Hydraulic to Isothermal Liquid Models”.
4-Way Ideal Valve	4-Way 3-Position Directional Valve (IL) The 4-Way 3-Position Directional Valve (IL) block behavior and parameterization are based on the spool position at the maximum orifice area and the maximum spool travel distance, while the 4-Way Ideal Valve uses initial spool position. Changing the valve parameterization after conversion can lead to inconsistencies. See “Upgrade Considerations When Converting Hydraulic to Isothermal Liquid Models”.
4-Way Directional Valve	4-Way 3-Position Directional Valve (IL) The 4-Way 3-Position Directional Valve (IL) block behavior and parameterization are based on the spool position at the maximum orifice area and the maximum spool travel distance, while the 4-Way Directional Valve uses initial spool position. Changing the valve parameterization after conversion can lead to inconsistencies. See “Upgrade Considerations When Converting Hydraulic to Isothermal Liquid Models”.
4-Way Directional Valves A-K	4-Way 3-Position Directional Valve (IL) The 4-Way 3-Position Directional Valve (IL) block behavior and parameterization are based on the spool position at the maximum orifice area and the maximum spool travel distance, while the 4-Way Directional Valves A-K use initial spool position. Changing the valve parameterization after conversion can lead to inconsistencies. See “Upgrade Considerations When Converting Hydraulic to Isothermal Liquid Models”.

Hydraulics (Isothermal) Block	Isothermal Liquid Block
Cartridge Valve Insert Cartridge Valve Insert With Conical Seat	Cartridge Valve Insert (IL)
Check Valve	Check Valve (IL) When you set **Opening dynamics** to `Include valve opening dynamics for the Check Valve block`, the setting affects the valve area. However, when **Opening dynamics** is `On` for the Check Valve (IL) block, the setting affects only the valve pressure.
Hydraulically Operated Remote control Valve	Pressure Compensator Valve (IL)
Pilot-Operated Check Valve	Pilot-Operated Check Valve (IL)
Shuttle Valve	Shuttle Valve (IL)

Flow Control Valves

Hydraulics (Isothermal) Block	Isothermal Liquid Block
Ball Valve	Poppet Valve (IL)
Counterbalance Valve	Counterbalance Valve (IL)
Flow Divider	The conversion tool converts the block to a subsystem.
Flow Divider-Combiner	The conversion tool converts the block to a subsystem.
Gate Valve	Variable Overlapping Orifice (IL)
Needle Valve	Needle Valve (IL)
Poppet Valve	Poppet Valve (IL)
Pressure-Compensated 3-Way Flow Control Valve	Pressure-Compensated 3-Way Flow Control Valve (IL)
Pressure-Compensated Flow Control Valve	Pressure-Compensated Flow Control Valve (IL)

Pressure Control Valves

Hydraulics (Isothermal) Block	Isothermal Liquid Block
Pressure Compensator	Pressure Compensator Valve (IL) When you set **Opening dynamics** to `Include valve opening dynamics for the Check Valve block`, the setting affects the valve area. However, when **Opening dynamics** is `On` for the Check Valve (IL) block, the setting affects only the valve pressure.
Pressure Reducing Valve	Pressure-Reducing Valve (IL) When you set **Opening dynamics** to `Include valve opening dynamics for the Check Valve block`, the setting affects the valve area. However, when **Opening dynamics** is `On` for the Check Valve (IL) block, the setting affects only the valve pressure.
Pressure Relief Valve	Pressure Relief Valve (IL) When you set **Opening dynamics** to `Include valve opening dynamics for the Check Valve block`, the setting affects the valve area. However, when **Opening dynamics** is `On` for the Check Valve (IL) block, the setting affects only the valve pressure.
Pressure Reducing 3-way Valve	Pressure-Reducing 3-Way Valve (IL) When you set **Opening dynamics** to `Include valve opening dynamics for the Check Valve block`, the setting affects the valve area. However, when **Opening dynamics** is `On` for the Check Valve (IL) block, the setting affects only the valve pressure.

Valve Actuators

Hydraulics (Isothermal) Block	Isothermal Liquid Block
2-Position Valve Actuator 3-Position Valve Actuator	Multiposition Valve Actuator
Double-Acting Servo Cylinder	Double-Acting Servo Valve Actuator (IL) The conversion tool sets the **Hard stop model** parameter to `Stiffness and damping applied smoothly through transition region, damped rebound`, with default hard stop parameters.
Hydraulic 4-Port Cartridge Valve Actuator	Cartridge Valve Actuator (IL) The conversion tool converts the block to a subsystem. The Cartridge Valve Actuator (IL) block calculates area at port **B** as the sum of the area at ports **X** and **Y** minus the area at port **A**. The hydraulics block calculates the difference between the areas at port **X** and port **A**.
Hydraulic Cartridge Valve Actuator	Cartridge Valve Actuator (IL)
Hydraulic Single-Acting Valve Actuator Hydraulic Double-Acting Valve Actuator	Pilot Valve Actuator (IL)
Proportional and Servo-Valve Actuator	The conversion tool converts the block into a subsystem that contains physical signal blocks.
Valve Actuator	Proportional Valve Actuator

Valve Forces

Hydraulics (Isothermal) Block	Isothermal Liquid Block
Spool Orifice Hydraulic Force	Spool Orifice Flow Force (IL)
Valve Hydraulic Force	The conversion tool converts the Valve Hydraulic Force block to a subsystem.

Fluid Network Interfaces Library

Interface (TL-IL)	Interface (TL-IL) The conversion tool sets the Interface (TL-IL) parameter **Fluids domain interface** to `Thermal Liquid (TL) - Isothermal Liquid (IL)`. **Note** In R2020a and R2020b, this block is named Interface.
Double-Acting Actuator (H-G)	Double-Acting Actuator (G-IL) The conversion tool converts the Double-Acting Actuator (H-G) block to a subsystem.

SimHydraulics Legacy Library

Variable-Displacement Hydraulic Machine	Variable-Displacement Motor (IL) The conversion tool converts the Variable-Displacement Hydraulic Machine block to a subsystem and sets the Variable-Displacement Motor (IL) **Leakage and friction parameterization** to `Input signal - volumetric and mechanical losses`.
Variable-Displacement Hydraulic Machine (External Efficiencies)	Variable-Displacement Motor (IL) The conversion tool converts the Variable-Displacement Hydraulic Machine (External Efficiencies) block to a subsystem and sets the Variable-Displacement Motor (IL) **Leakage and friction parameterization** parameter to `Input signal - volumetric and mechanical efficiencies`.

See Also

`hydraulicToIsothermalLiquid` | `hydraulicToIsothermalLiquidPostProcess` | Interface (H-IL)

More About

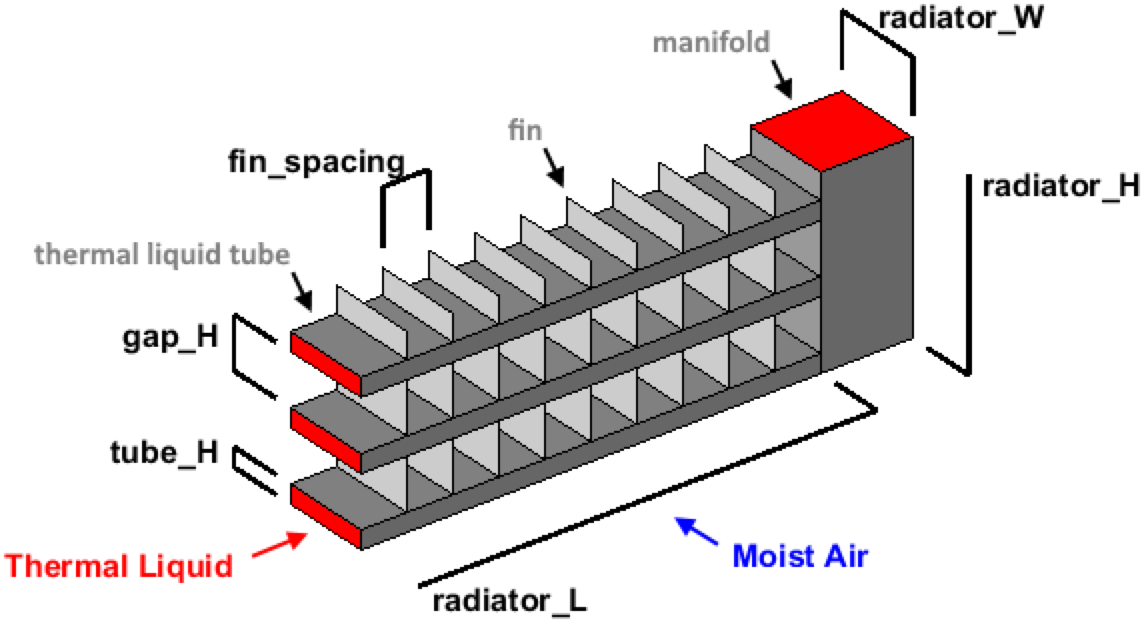
- “How to Upgrade Hydraulic Models”
- “Upgrading Hydraulics (Isothermal) Models in Simscape Fluids” on page 5-2

Model a Radiator as a Heat Exchanger

This example shows how to model a radiator as a heat exchanger by using the physical measurements of the radiator, thermodynamic properties of the two fluids, and the properties of the heat transfer materials used in the radiator. The example explains how to specify the parameters of the heat exchanger, write a script that calculates the variables used in these parameters, and create a model to test the functionality of the block.

In the radiator, the thermal liquid flows through a number of rectangular tubes from left to right, and the moist air flows from front to back. Between the tubes, there are fins that the moist air contacts while flowing through.

This graphic is a corner cross-section of the radiator that shows the measurements and the direction of the fluid flow.



Measurements and Properties Given for the Radiator

Variable Name	Measurement Description	Value	Units
`radiator_L`	Length of the heat exchanger portion of the radiator	`0.6`	m
`radiator_W`	Width of the radiator	`0.015`	m
`radiator_H`	Height of radiator	`0.2`	m
`tubes_N`	Number of radiator fluid tubes	`25`	N/A
`tube_H`	Height of each individual fluid tube	`0.0015`	m
`fin_spacing`	Distance between adjacent fins	`0.002`	m
`wall_thickness`	Wall thickness of the radiator fluid tubes	`0.1`	mm
`wall_conductivity`	Thermal conductivity of the fluid tube wall material	`240`	W/(m*K)
`liquid_pipe_D`	Diameter of the tubes going to and from the radiator	`0.019`	m

Model the Radiator

Because the radiator fluid is not expected to change phase, you can model it with the thermal liquid domain. Because environmental air can contain a non-negligible amount of moisture, you can model it with the moist air domain. Use the Heat Exchanger (TL-MA) block to model the radiator.

On the Simulink Start Page, in the **Simulink** section, click **Blank Model**.

Add a Heat Exchanger (TL-MA) block to this model and open it in order to specify its parameters.

Start the Script

Create a new MATLAB script and specify the measurements and properties of the heat exchanger. Most of these values are shown in the above graphic. `tubes_N` is the total number of thermal liquid tubes in the radiator. `wall_thickness` is the thickness of the material of the thermal liquid tubes. `wall_conductivity` is the thermal conductivity of this material. `liquid_pipe_D` is the diameter of the pipe at the input and output ports of the radiator.

```
radiator_L = 0.6; % [m]
radiator_W = 0.015; % [m]
radiator_H = 0.2; % [m]
tubes_N = 25;
tube_H = 0.0015; % [m]
fin_spacing = 0.002; % [m]
wall_thickness = 1e-4; % [m]
wall_conductivity = 240; % [W/(m*K)]
liquid_pipe_D = 0.019; % [m]
```

Set the Geometry Parameters for the Thermal Liquid

In the **Thermal Liquid 1** settings:

1 Set the **Flow geometry** parameter to `Flow inside one or more tubes`.

2 Set the **Number of tubes** parameter to `tubes_N`.

3 Set the **Total length of each tube** parameter to `radiator_L`. Because the thermal liquid tubes span the entire length of the radiator, this value is the full radiator length.

4 Set the **Tube cross section** parameter to `Rectangular`.

5 Specify the internal width and height of the thermal liquid tubes as `tube_W_internal` and `tube_Hinternal`. In this radiator, the width of each tube is the same as the total width of the radiator. However, the thermal liquid surface area is the inner surface area of these tubes. Subtract twice the wall thickness from both the width and height of the tubes to calculate these values.

```
tube_H_internal = tube_H - 2 * wall_thickness; % [m]
tube_W_internal = radiator_W - 2 * wall_thickness; % [m]
```

Set the **Tube width** and **Tube height** parameters to `tube_W_internal` and `tube_H_internal`, respectively. Set the units for both parameters to `m`.

6 Set the **Total fin surface area** parameter to 0.

Set the Geometry Parameters for the Moist Air

In the **Moist Air 2** settings:

1 Set the **Flow geometry** parameter to `Generic`.

2 Specify the distance between the adjacent thermal liquid tubes as `gap_H` and then the cross-sectional area for the moist air flow through the radiator as `air_area_flow`. `air_area_flow` parameter represents the smallest cross-sectional area that the moist air must flow through inside the heat exchanger. Because there are no particular constrictions to the moist air flow, use the total cross sectional flow area. The air flows through the gaps between the thermal liquid tubes.

```
gap_H = (radiator_H - tubes_N * tube_H) / (tubes_N - 1); % [m]
air_area_flow = (tubes_N - 1) * radiator_L * gap_H; % [m^2]
```

Set the **Minimum free-flow area** parameter to `air_area_flow`.

3 Specify the external surface area of the thermal liquid tubes as `air_area_primary`. This parameter is the primary heat transfer surface area between the moist air and the heat exchanger. The value of this parameter is equal to the external surface area of the tubes that the thermal liquid is flowing through. Do not subtract the wall thickness because the moist air contacts the outside of these tubes.

```
air_area_primary = tubes_N * 2 * (radiator_W + tube_H) * radiator_L; % [m^2]
```

Set the **Heat transfer surface area without fins** parameter to `air_area_primary`.

4 Set the **Moist air volume inside heat exchanger** parameter to `air_area_flow*radiator_W`.

Because the flow has a constant cross-section, this value is the cross-sectional flow area multiplied by the heat exchanger width. This value is the volume of fluid that is inside the heat exchanger at any point in time.

5 Specify the number of fins in the radiator as `fins_N` and the total surface area of all the fins as `air_area_fins`. For the moist air, the fin surface area is the majority of the heat transfer surface area. Calculate the number of fins based on the fin spacing and the number of gaps in which the fins are found. The total fin surface area is the sum of all of the individual fin surface areas, including both sides of every fin.

```
fins_N = (tubes_N - 1) * radiator_L / fin_spacing;
air_area_fins = 2 * fins_N * radiator_W * gap_H; % [m^2]
```

Set the **Total fin surface area** parameter to `air_area_fins`.

6 Set the **Fin efficiency** parameter to `0.7`.

Because heat must move along the fins, the fin surface area is not as efficient at increasing the amount of heat transfer as primary surface area. The efficiency depends on many factors, including the fin shape, length, thickness, and material. The value `0.7` is an estimation for the radiator.

Set the Parameters in the Configuration Section

In the **Configuration** settings:

1 Set the **Flow arrangement** parameter to `Cross flow`. In a radiator, the moist air flows from front to back, while the thermal liquid flows left or right. Because the two flows are perpendicular to each other, the radiator has it a cross-flow arrangement.

2 Set the **Cross flow arrangement** parameter to `Thermal Liquid 1 mixed & Moist Air 2 unmixed`. This parameter describes if either of the two fluids are mixed in the heat exchanger. Because there is no physical structure that prevents the movement of the fluid within these tubes from front to back, the thermal liquid is mixed. The fins block the moist air from moving left and right while traveling through the radiator, so the moist air is unmixed.

3 Specify the thermal resistance through the walls of the thermal liquid tubes as `thermal_resistance_primary`. This value represents the total thermal resistance between the two primary heat transfer surfaces. Calculate this using the wall thickness, the heat conductivity, and the surface area. Do not include any change in thermal resistance due to wall fouling, fluid convective heat transfer, or fins in either fluid because the block accounts for these affects. Because the primary surface areas of the thermal liquid and moist air are very close, this calculation can use the primary heat transfer surface area of only the moist air.

```
thermal_resistance_primary = wall_thickness / air_area_primary / wall_conductivity; % [K/kW]
```

Set the **Thermal resistance through heat transfer surface** parameter to `thermal_resistance_primary`.

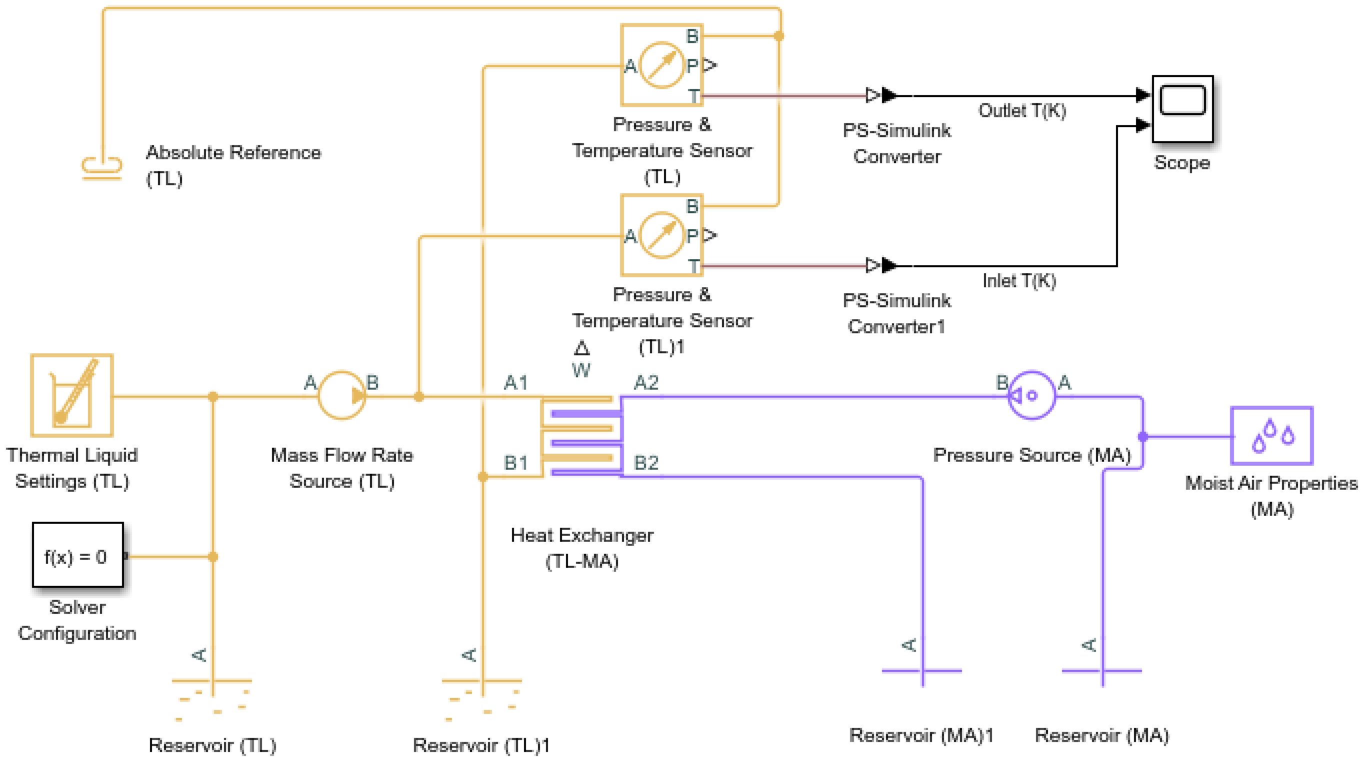
4 Specify the internal cross-sectional area for the thermal liquid pipes going to and from the radiator as `liquid_pipe_A`.

```
liquid_pipe_A = pi * liquid_pipe_D^2 / 4; % [m^2]
```

Set **Cross-sectional area at port A1** and **Cross-sectional area at port B1** to `liquid_pipe_A`. Set **Cross-sectional area at port A2** and **Cross-sectional area at port B2** to `air_area_flow`.

Test the Radiator

Create this model to test the radiator. This model simulates a thermal liquid at a constant temperature pumped at a constant flow rate through the radiator and a moist air at a constant temperature forced through the radiator at a constant pressure. The model measures the temperature difference between the inlet and outlet thermal fluids.



Create the above Simscape model by following these steps:

1 Add blocks shown in the diagram, and connect the blocks to the specified Heat Exchanger (TL-MA)

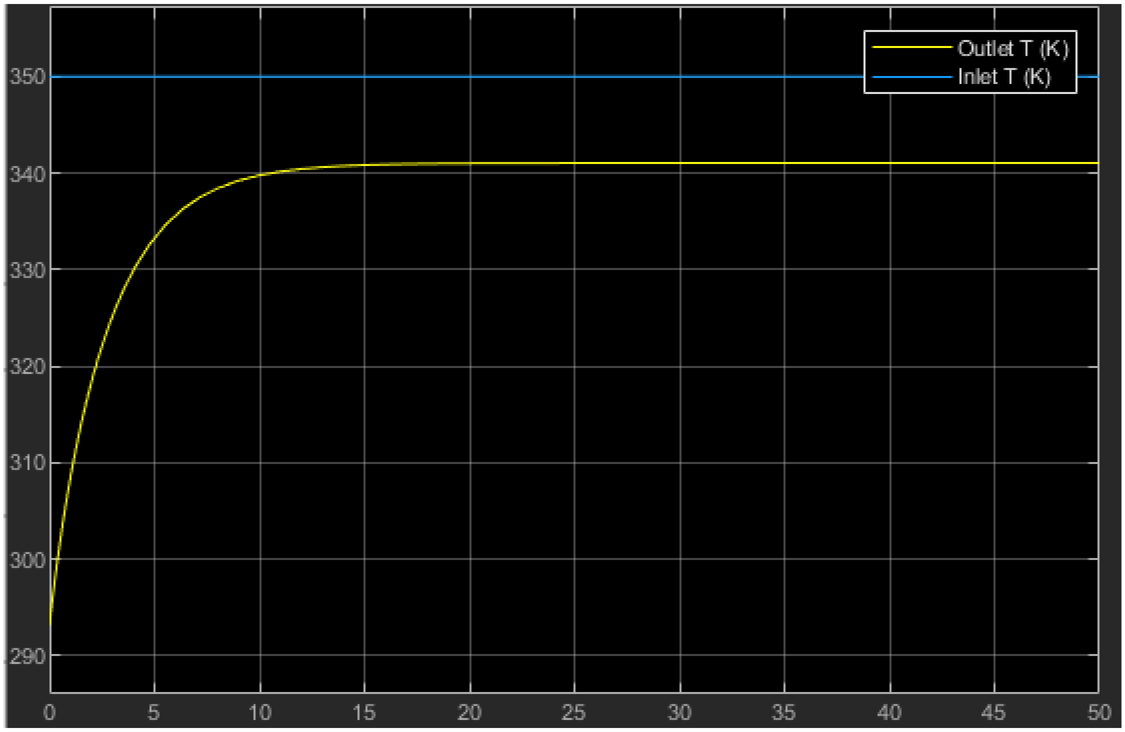
2 Specify the block parameters for these blocks:

Block	Parameter	Value	Units
Reservoir (TL)	**Reservoir temperature**	350	K
	Cross-sectional area at port A	liquid_pipe_A	m^2
Reservoir (TL)1	**Reservoir temperature**	293.15	K
	Cross-sectional area at port A	liquid_pipe_A	m^2
Mass Flow Rate Source (TL)	**Mass flow rate**	0.1	kg/s
	Cross-sectional area at ports A and B	liquid_pipe_A	m^2
Reservoir (MA)	**Cross-sectional area at port A**	air_area_flow	m^2

Block	Parameter	Value	Units
Reservoir (MA)1	**Cross-sectional area at port A**	`air_area_flow`	`m^2`
Pressure Source (MA)	**Pressure differential**	`500`	`Pa`
	Cross-sectional area at port A	`air_area_flow`	`m^2`
	Cross-sectional area at port B	`air_area_flow`	`m^2`

3 On the **Simulation** tab, in the **Simulate** section, set **Stop Time** to `50`.

On the **Simulation** tab, in the **Simulate** section, click **Run**. When the model finishes running, open the Scope block.



The blue and yellow curves indicate the temperatures of the inlet and outlet thermal fluids. The steady state temperature of the outlet thermal fluid is 342 `k`, which indicates the radiator cools the thermal liquid by 8 `k`. Note that the initial temperature of the outlet thermal fluid equals room temperature, which is 293.15 `k`.

See Also

Related Examples

- “Condenser and Evaporator Heat Transfer” on page 6-576

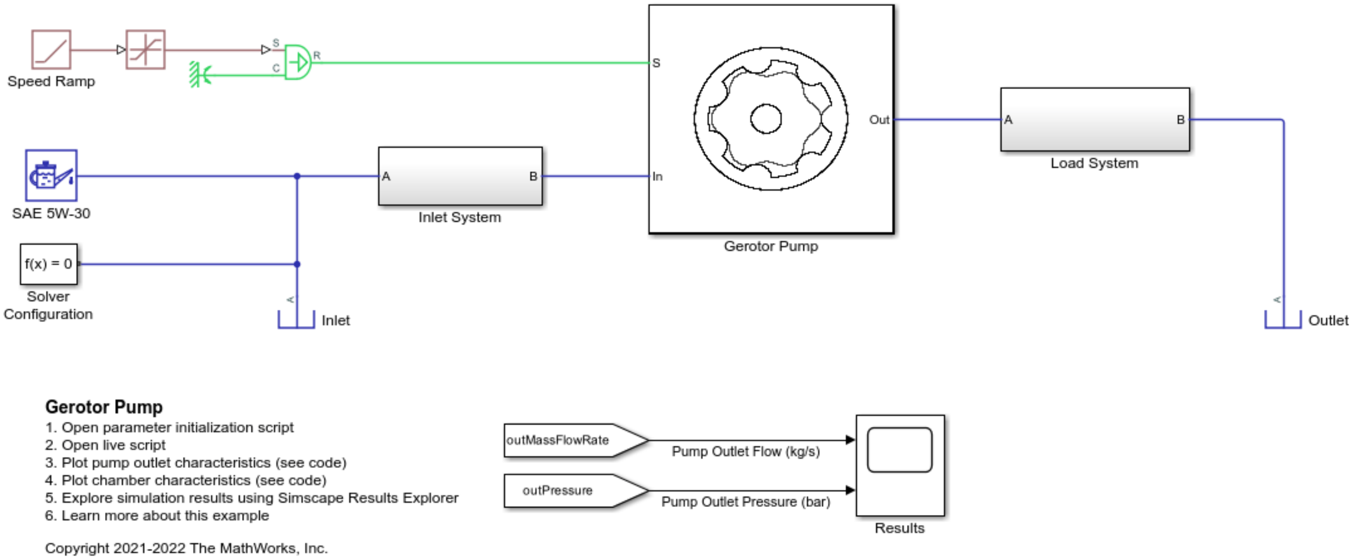
Simscape Fluids Examples

Gerotor Pump

This example shows how to model, parameterize, and test a gerotor pump.

Model

The following figure shows a model of the gerotor pump with a lubrication system. The flow circuit consists of an inlet system upstream of the pump, the gerotor pump and a load system downstream of the pump.



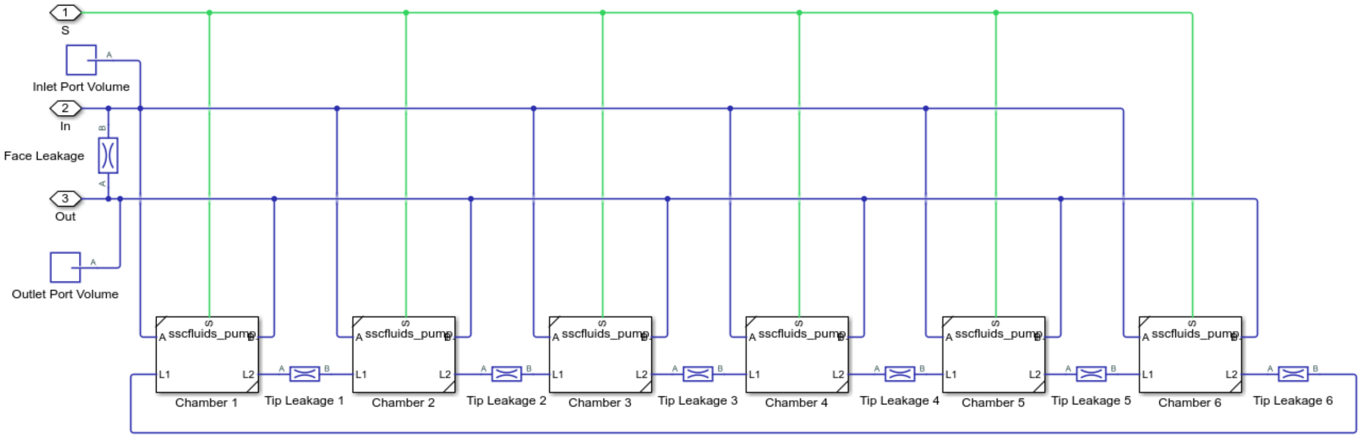
Gerotor Pump

1. Open parameter initialization script
2. Open live script
3. Plot pump outlet characteristics (see code)
4. Plot chamber characteristics (see code)
5. Explore simulation results using Simscape Results Explorer
6. Learn more about this example

Copyright 2021-2022 The MathWorks, Inc.

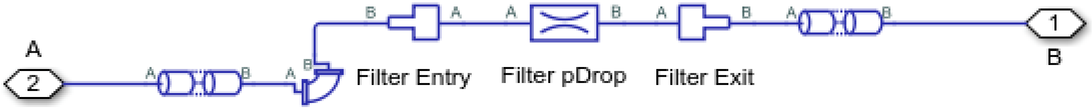
Gerotor Pump Subsystem

This subsystem shows how the gerotor pump is modeled. The gerotor pump comprises of an outer gear (driven gear) and an inner gear (driving gear) mounted on a rotating shaft, modeled as an angular velocity source. For the pump modeled here, the inner gear teeth profile is based on a trochoid curve geometry, and the outer gear teeth profile is a conjugate of the inner gear teeth profile. Gerotor pump is a positive displacement type pump, as the volumetric flow is effectively produced by the continual suction and delivery actions performed by the cyclical variation of the volume between the teeth of the mating gear set. This liquid volume between the mating gears can be divided into a number of sub-volumes or chambers which are connected with the adjacent ones through a gap due to teeth (or tip) clearance leading to tip leakage between chambers. The inlet and outlet port volumes are connected through a leakage flow called face leakage resulting from the clearance between the gear faces and housing inner faces. The volumes of the chambers change as the pump rotates and they are connected to the inlet port volume and outlet port volume through port area profiles. The number of variable volume liquid chambers equal the number of outer gear teeth.



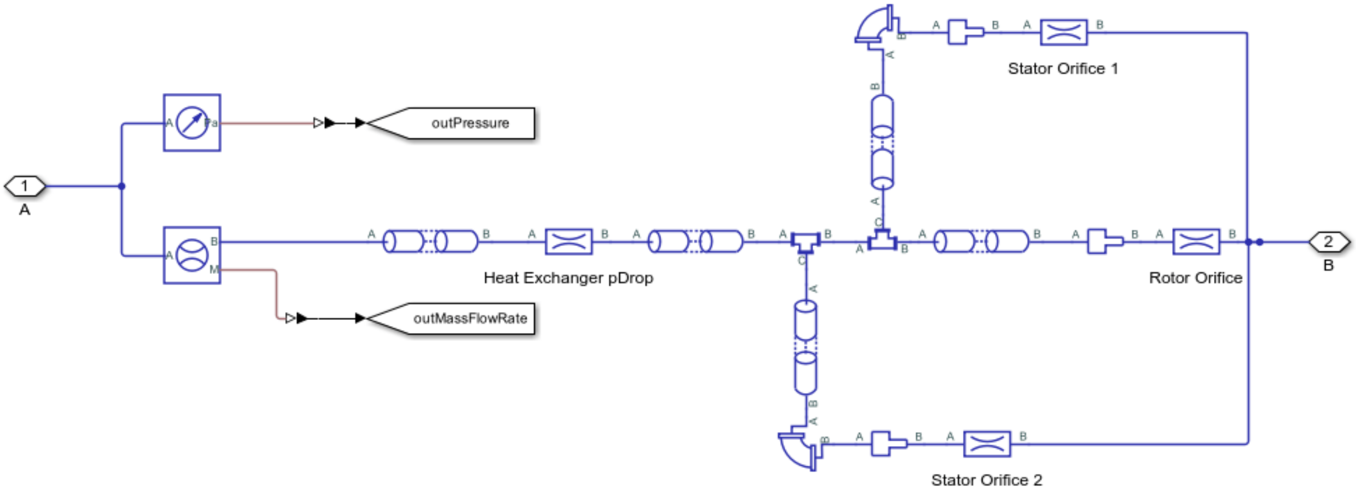
Inlet System Subsystem

This subsystem shows a representation of a inlet system or suction system upstream of a gerotor pump. The inlet system is modeled using a combination of pipes, bends, sudden area changes and an orifice representing a pressure drop across a filter. These components contribute to pressure drop upstream of the pump.



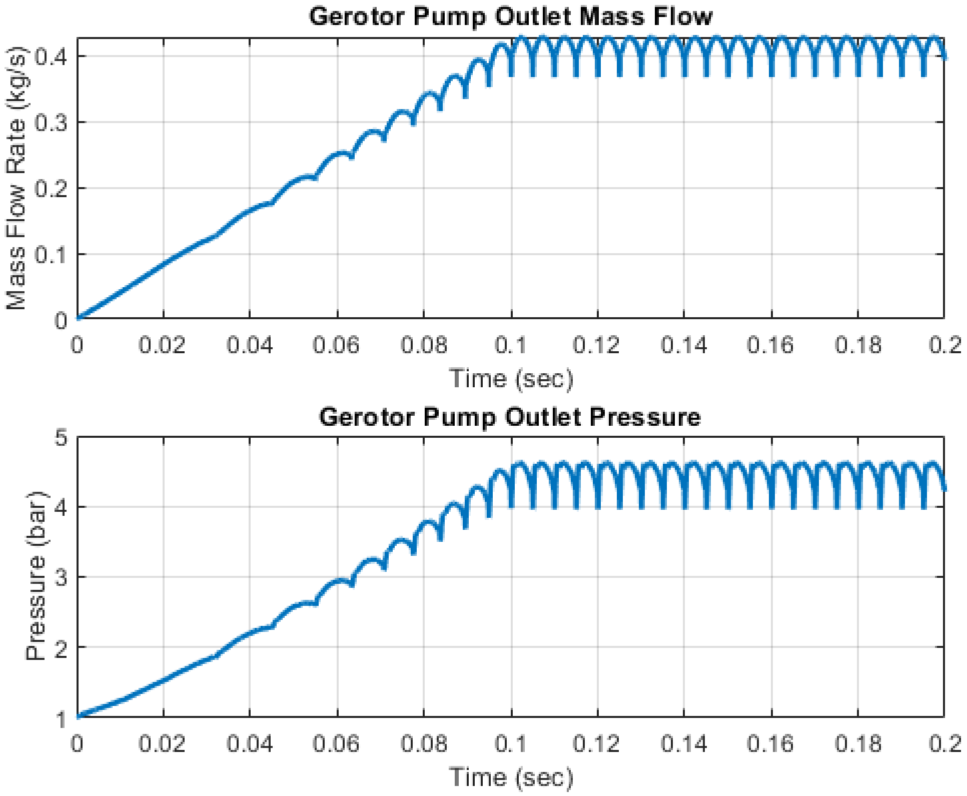
Load System Subsystem

This subsystem shows a representation of a lubrication and cooling system in an automotive where a motor is lubricated and cooled by the delivery flow of a gerotor pump. The load system is modeled using a combination of pipes, bends, sudden area changes, and orifices representing the pressure drop across a heat exchanger and orifices at entry to stator and rotor. All components contribute to load system pressure drop. The thermal effects are not considered in this model as it is purely isothermal.



Simulation Results from Simscape™ Logging

This model generates a plot of the mass flow rate and pressure at the outlet of the gerotor pump versus time.

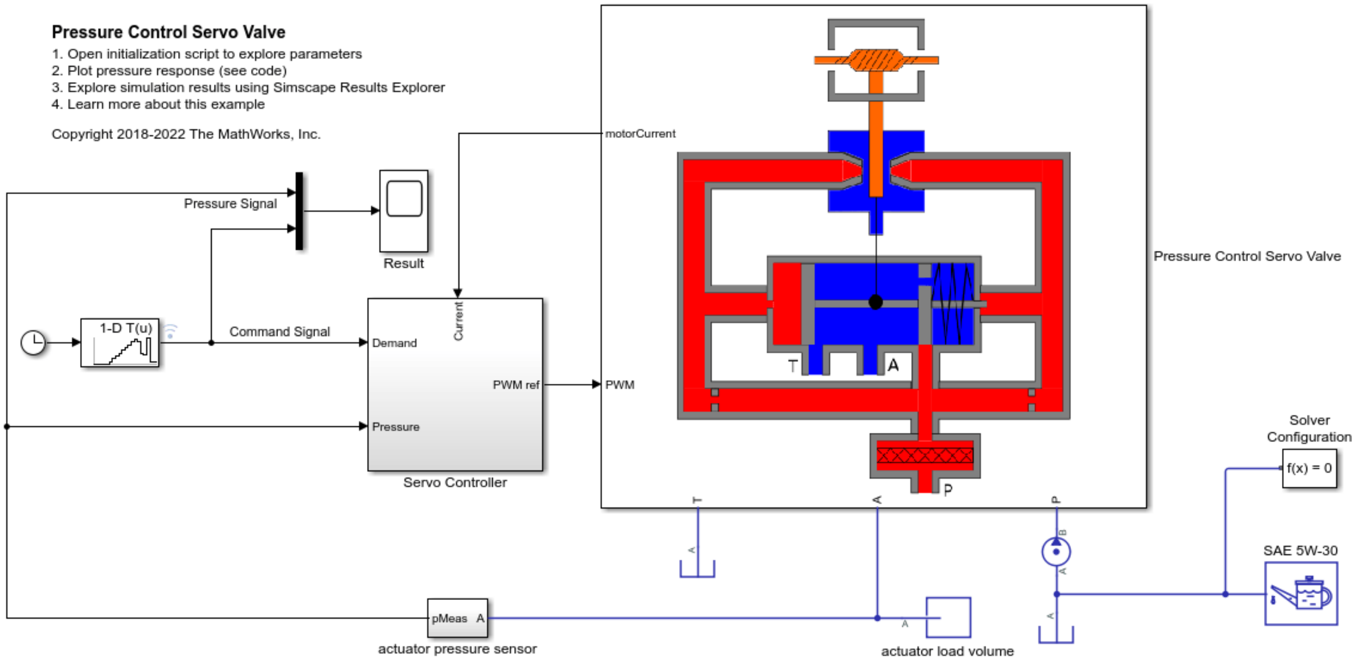


Pressure Control Servo Valve

This example shows how to model, parameterize, and test a pressure control servo valve with a closed loop control. This example also generates a comparison plot between the commanded and the achieved pressure in the actuator with respect to the time. This valve provides a precise and fast control of the pressure in the actuator with a very small electrical signal to the torque motor. These valves are used in the aerospace industry and the automotive industry for the safety critical applications.

Model

The following figure shows the model of a pressure control servo valve. Here P represents the pressure source port, T represents the reservoir, and A represents the actuator port.



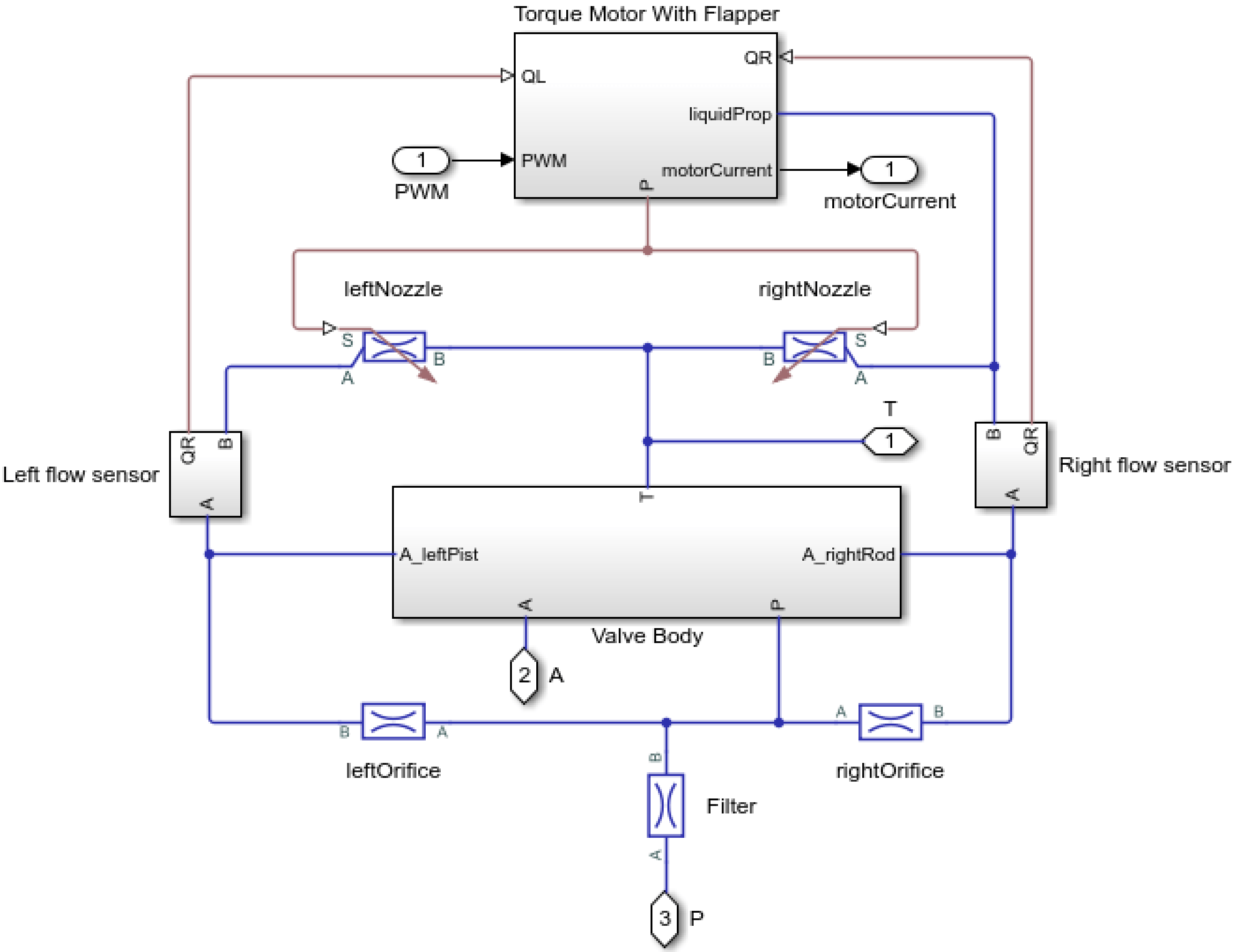
Pressure Control Servo Valve

1. Open initialization script to explore parameters
2. Plot pressure response (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2018-2022 The MathWorks, Inc.

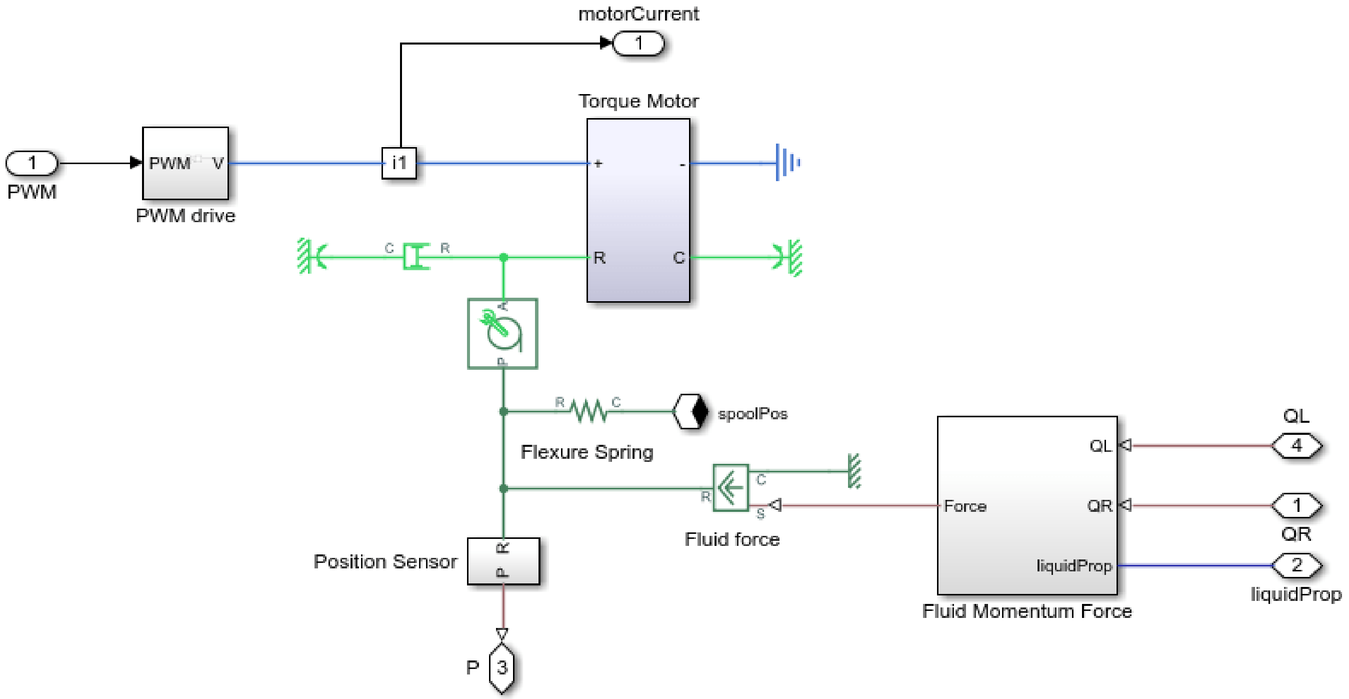
Pressure Control Servo Valve Subsystem

This subsystem shows how the pressure control servo valve is modeled. This subsystem includes modeling of the flapper nozzle amplifier circuit , the torque motor, and the valve body. The flapper nozzle amplifier circuit includes modeling of the left and the right side nozzle around the flapper, the filter, and the left and right side fixed orifices. The flapper nozzle amplifier circuit controls the pressure on the spool's left surface and the spool's right side extended rod.



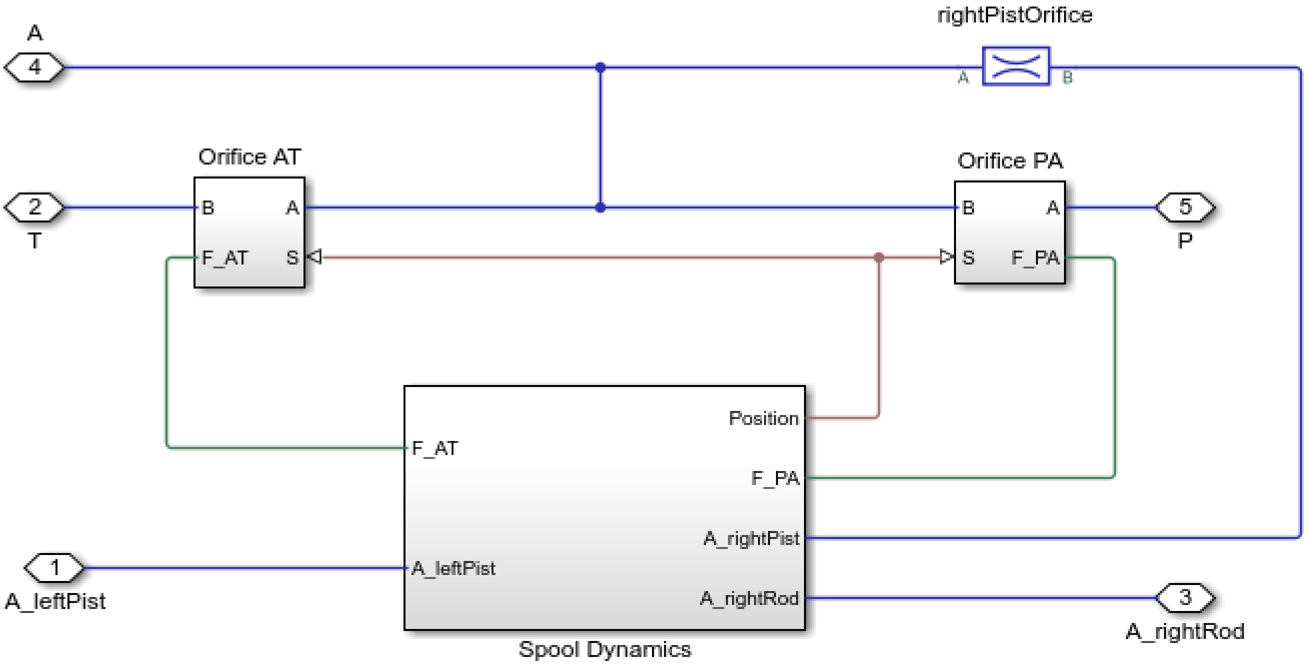
Torque Motor With Flapper Subsystem

This subsystem shows how the torque motor with the flapper is modeled. The flapper transfers the load torque to the torque motor. The load torque is generated due to the forces on the flapper from the flexure tube compression and the nozzle jets.



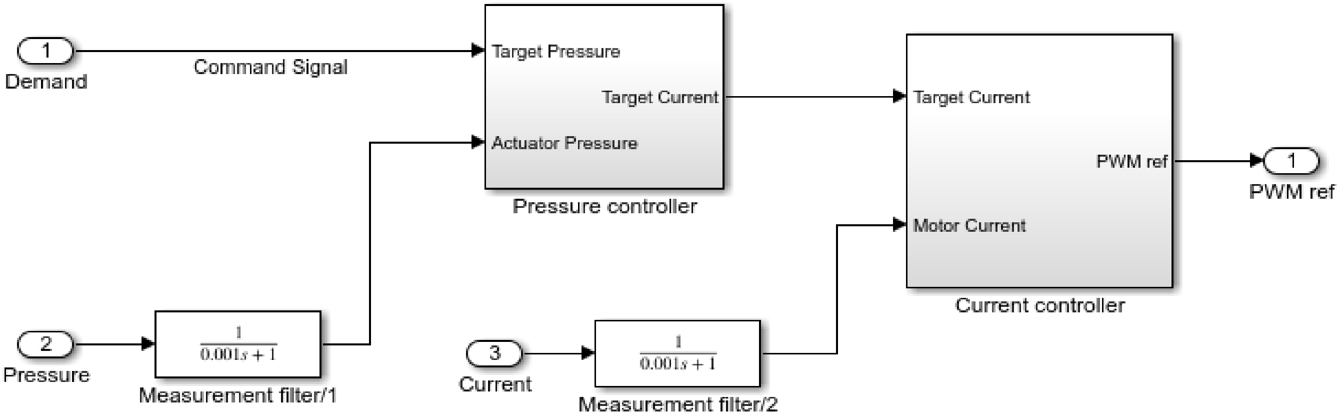
Valve Body Subsystem

This subsystem shows how the valve body is modeled. This subsystem includes modeling of the spool dynamics, the orifices opened and closed between the P to A and A to T ports due to the spool movement, and the drilled orifice in the spool's right surface. The spool dynamics includes modeling of the forces on the spool due to pressure, flexure tube compression, change in the direction of the fluid flow, friction, damping, spool spring compression, and the hard stop.



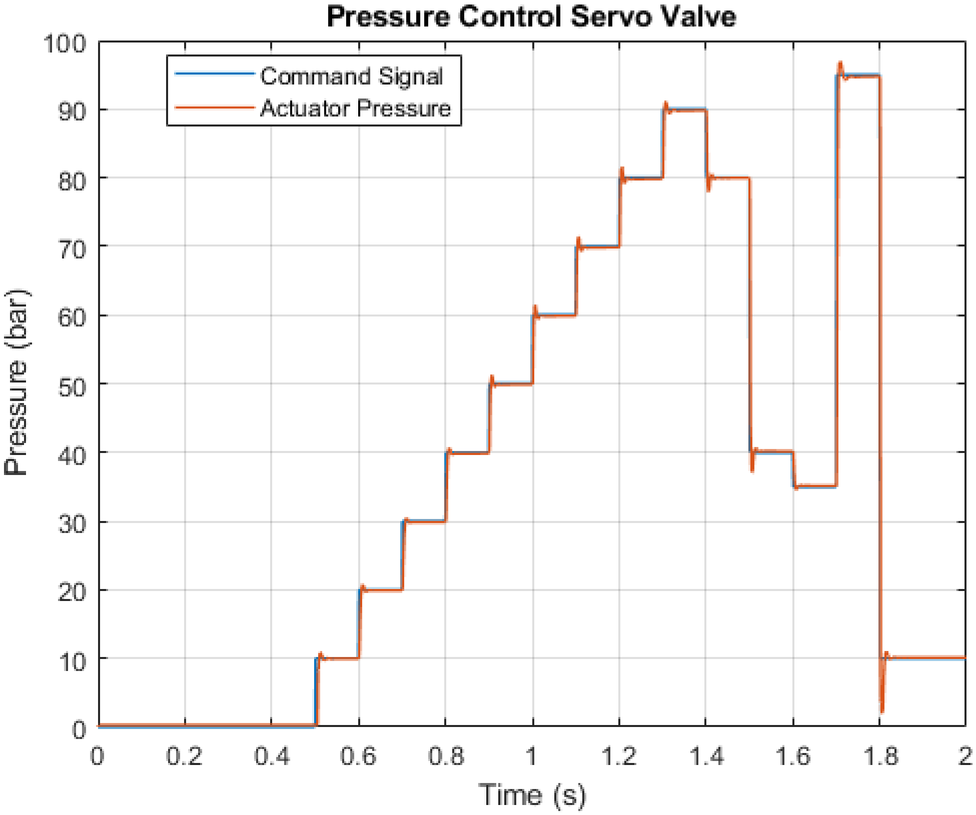
Servo Controller Subsystem

This subsystem shows how the servo controller is modeled. There are two control loops. The inner loop controls the motor current and the outer loop controls the pressure in the actuator.



Simulation Results from Simscape™ Logging

This model generates a comparison plot between the commanded and the achieved pressure in the actutator with respect to the time.

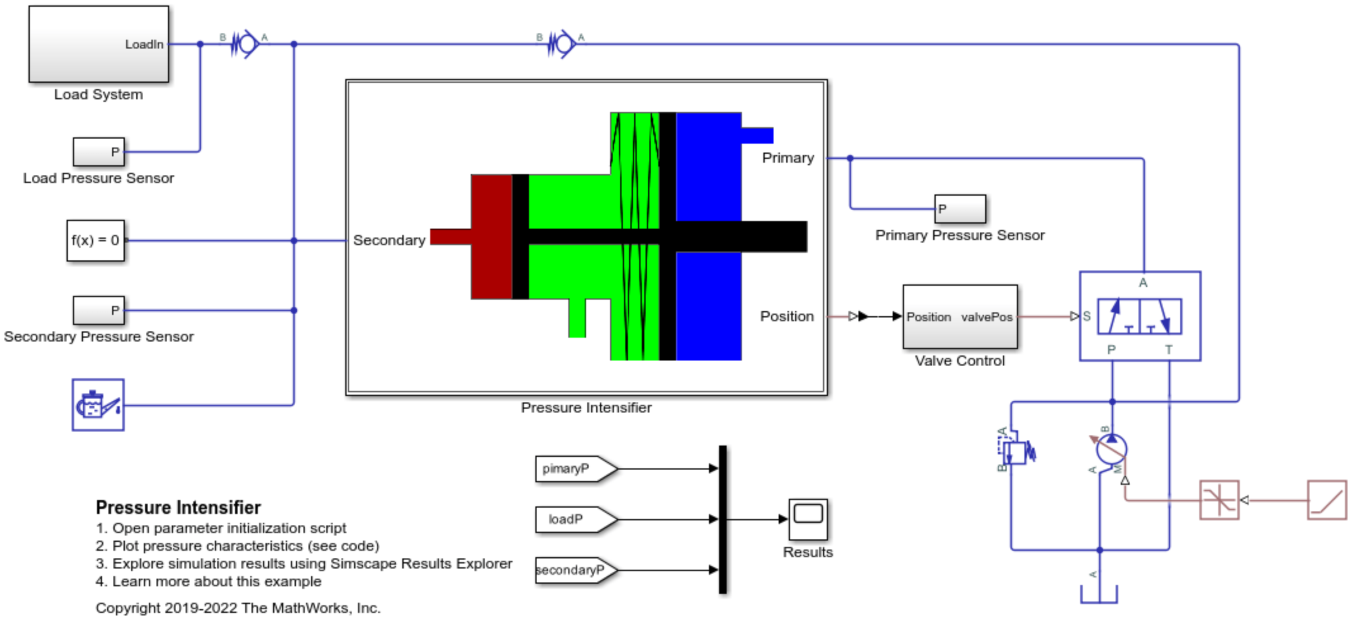

Pressure Control Servo Valve
Command Signal
Actuator Pressure
Pressure (bar)
100
90
80
70
60
50
40
30
20
10
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time (s)

Pressure Intensifier

This example shows how to model, parameterize, and test a pressure intensifier.

Model

The following figure shows a model of the pressure intensifier with a test harness. A fixed displacement pump with a relief valve is used as a power source to the system. The 3-way 2-position valve connects the primary side of the pressure intensifier with the power source or the reservoir as per command signal from the valve control. The secondary side of the pressure intensifier is connected with the load system via a check valve.

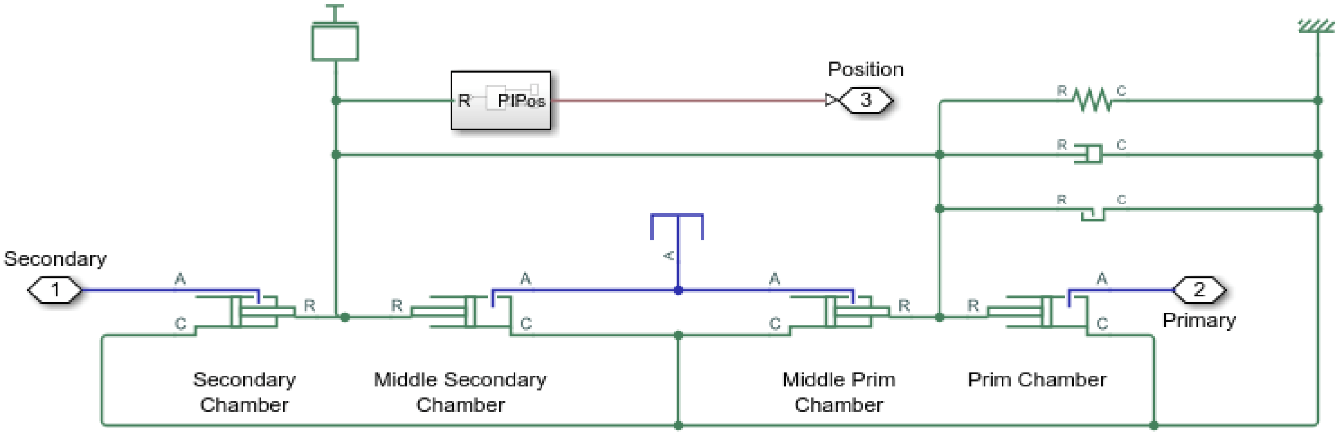


Pressure Intensifier
1. Open parameter initialization script
2. Plot pressure characteristics (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2019-2022 The MathWorks, Inc.

Pressure Intensifier Subsystem

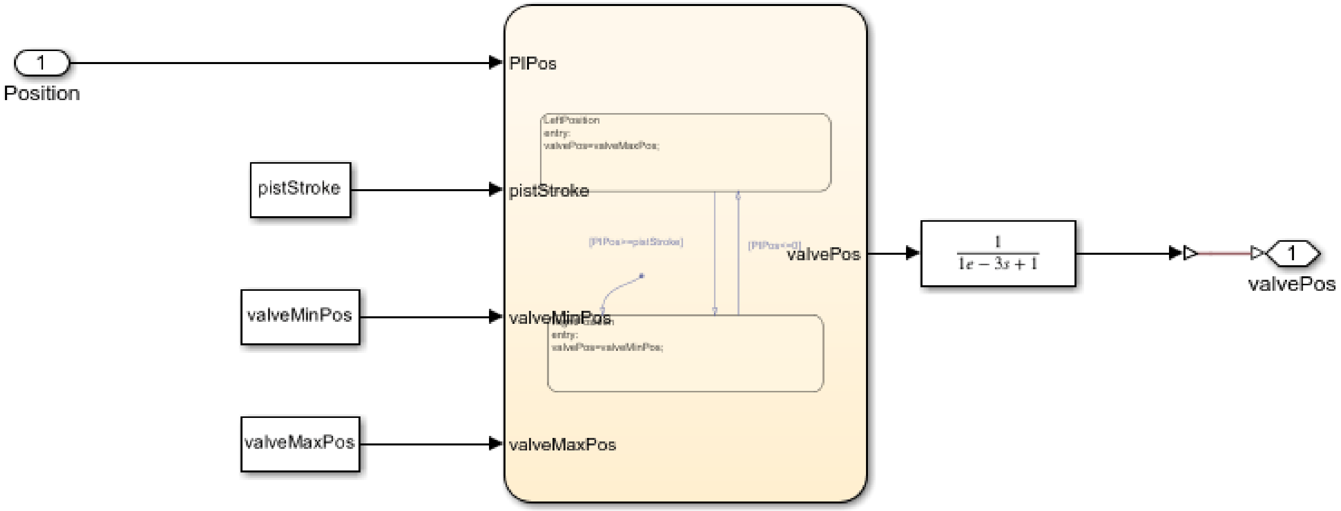
This subsystem shows how the pressure intensifier is modelled. The pressure intensifier boosts the pressure of the primary side to the pressure of the secondary side by the area ratio between the primary side and the secondary side pistons.



Valve Control Subsystem

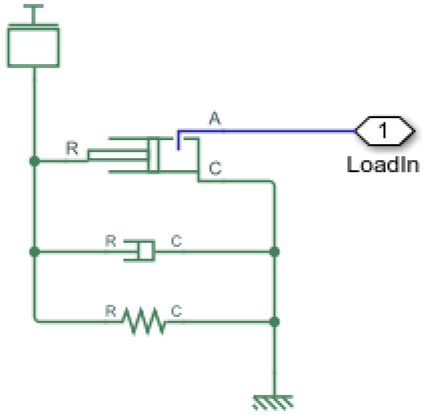
This subsystem shows an abstraction of a control system for the solenoid operated 3-way 2-position directional control valve. The control system senses the position of the pressure intensifier spool and signals the directional control valve to move to the A-T open or the P-A open position.

When the directional control valve is signaled to the A-T position, the primary side's fluid goes to the tank and the secondary side is connected to the pump. And, when the directional valve is signaled to the P-A position, the primary side is connected to the pump and the secondary side starts filling the load.



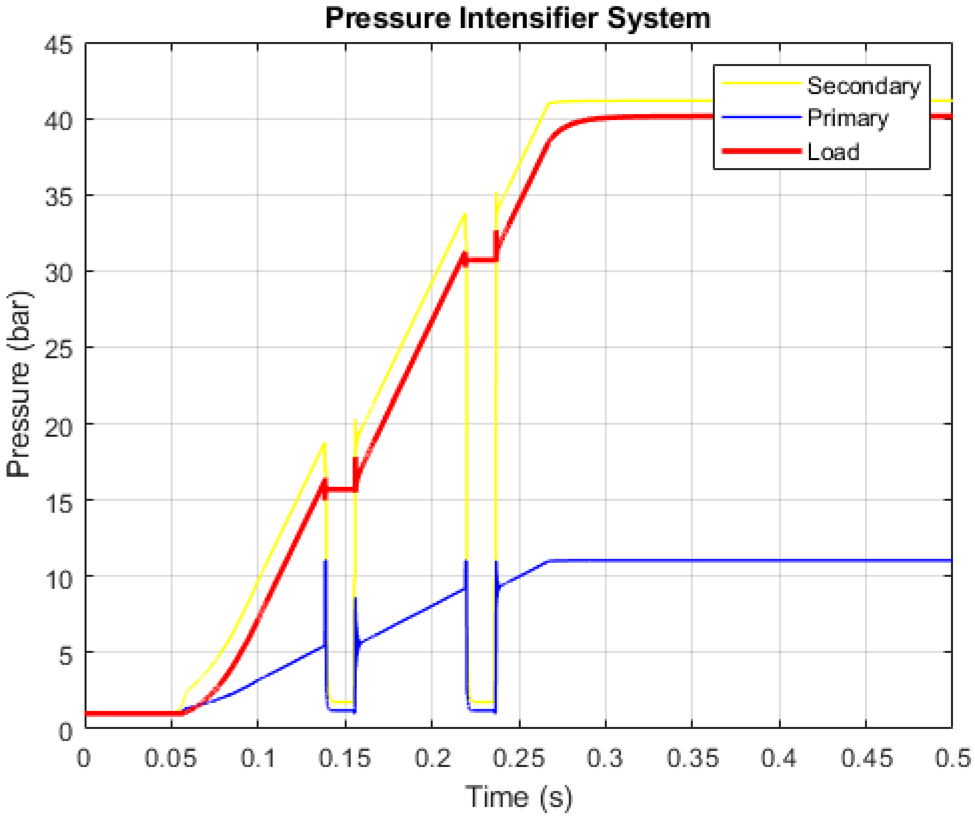
Load System Subsystem

This subsystem shows how the load system is modelled.



Simulation Results from Simscape™ Logging

This model generates a plot between the time and the generated pressures at the primary side, the secondary side, and at the load.



Water Hammer Effect

This demo shows how the Isothermal Liquid library can be used to model water hammer in a long pipe. After opening a valve to slowly establish steady flow in the pipe, the valve is quickly shut. If the Valve is shut quickly enough, it triggers a water hammer effect. A water hammer arrestor suppresses the pressure spikes.

The valve is modeled using a Variable Local Restriction (IL) block, and the pipe is divided into 20 segments using two Pipe (IL) blocks. Breaking the pipe into more segments increases fidelity at the expense of simulation performance. The water hammer effect requires modeling both dynamic compressibility and inertia in the pipes. The water hammer arrestor is modeled using a Gas-Charged Accumulator (IL) block. A Variant Connector block enables and disables the water hammer arrestor.

Model

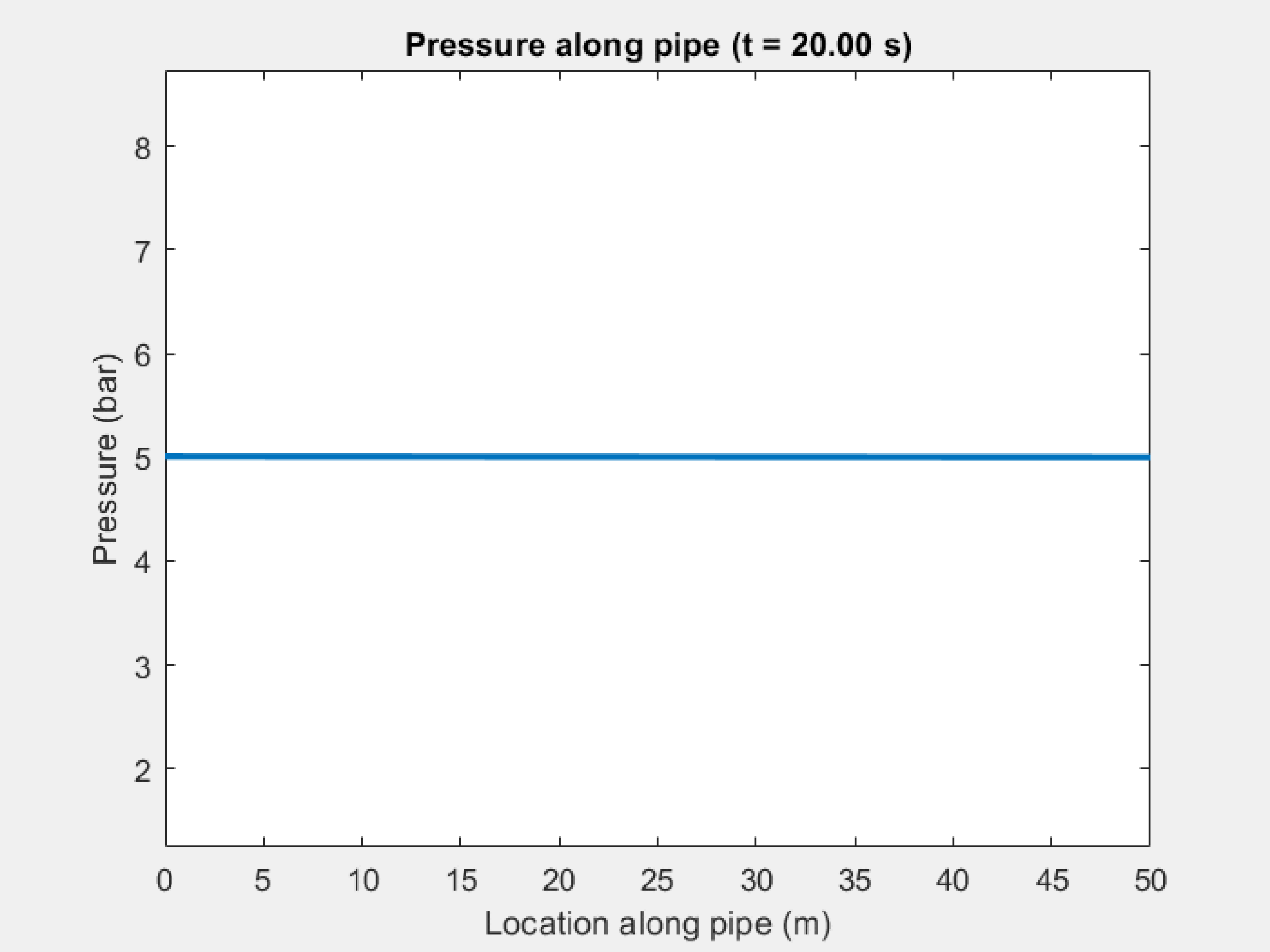
Simulation Results from Scopes

These figures show the pressure shock wave and resulting oscillations due to a sudden valve closure when the water hammer arrestor is disabled.

Power Spectral Density

Animation of Simscape Logging Results

This animation shows the pressure shock wave and resulting oscillations along the pipeline due to a sudden valve closure when the water hammer arrestor is disabled.



Simulation Results from Simscape Logging

This figure compares the pressure and flow rate at the pipe midpoint with the water hammer arrestor disabled and enabled.

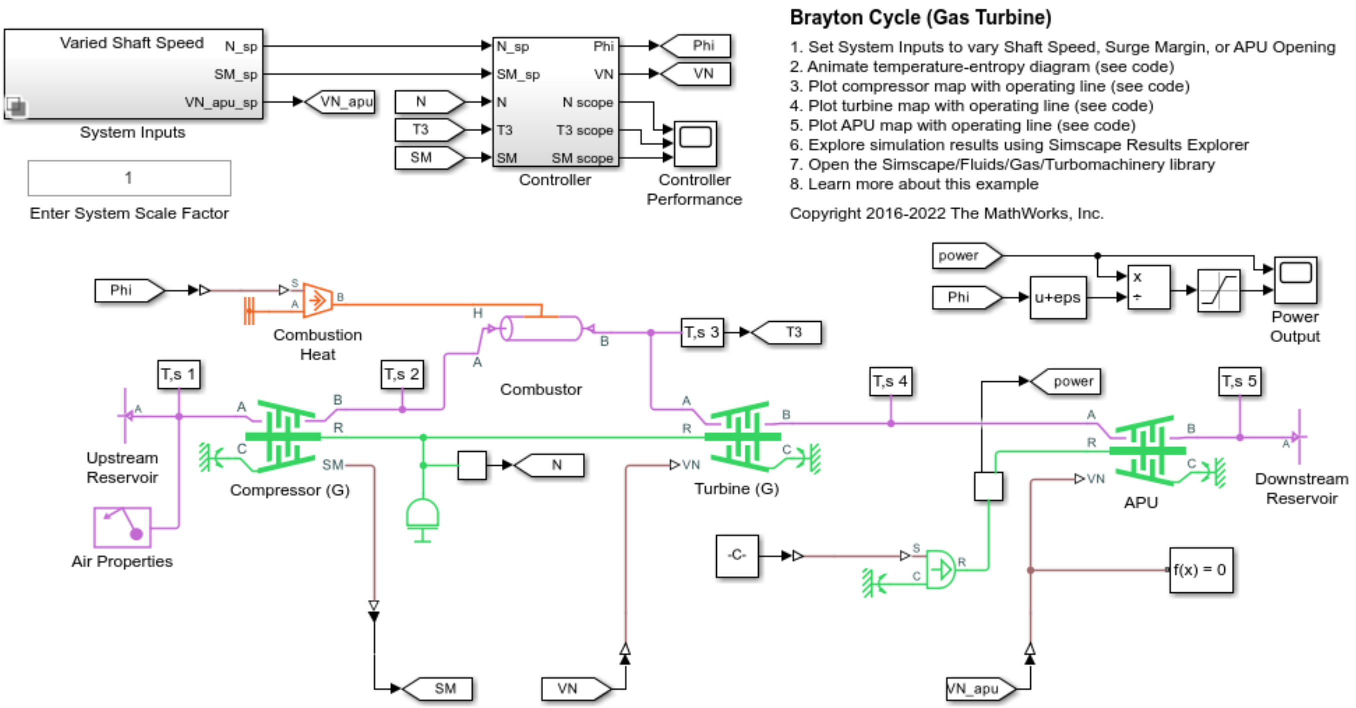
Brayton Cycle (Gas Turbine)

This example models a gas turbine auxiliary power unit (APU) based on the Brayton Cycle. The power input to the system is represented by heat injection into the combustor; actual combustion chemistry is not modeled. A single shaft connects the compressor and the turbine so that the power from the turbine drives the compressor. The APU is a free turbine that further expands the exhaust stream to produce output power.

Three PID controllers regulate the shaft speed, the turbine inlet temperature, and the compressor surge margin. System inputs are defined for three scenarios: varying shaft speed, varying surge margin, and varying APU vane opening. Running the first scenario produces the typical operating line on the compressor map. Running the second and third scenarios show where the maximum power output and maximum global efficiency occurs.

The entire system can be scaled up or down by changing the scaling factor which is applied to all area and flow rate related parameters.

Model

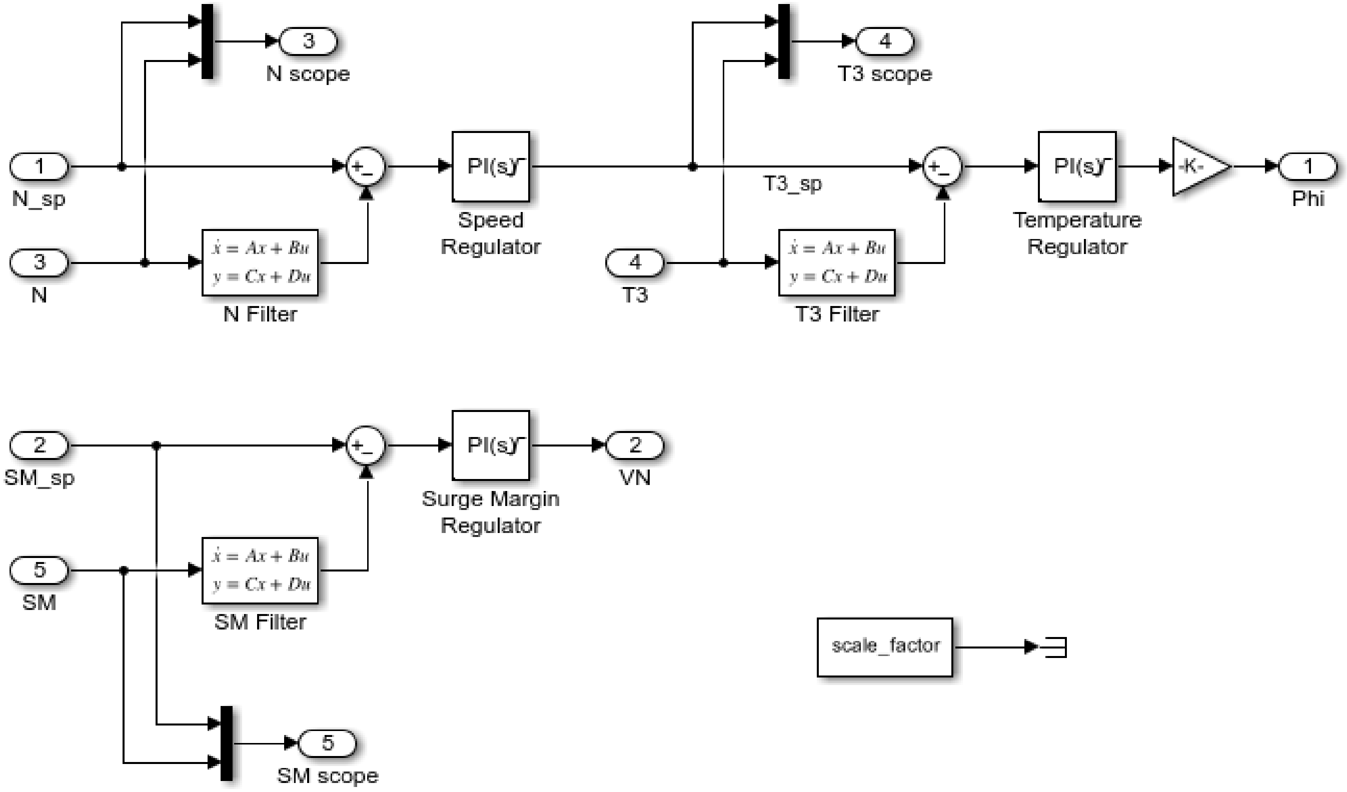


Brayton Cycle (Gas Turbine)

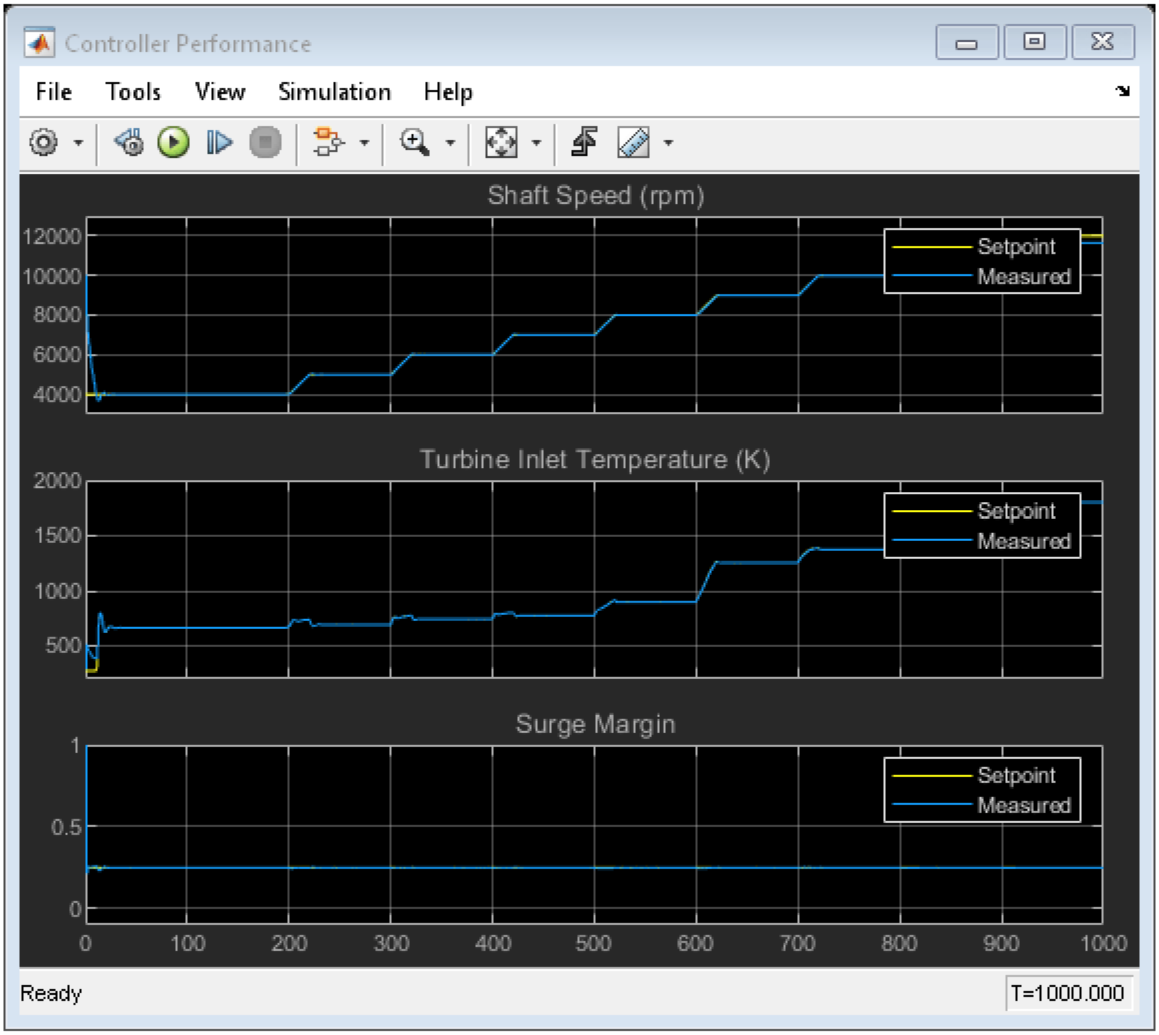
1. Set System Inputs to vary Shaft Speed, Surge Margin, or APU Opening
2. Animate temperature-entropy diagram (see code)
3. Plot compressor map with operating line (see code)
4. Plot turbine map with operating line (see code)
5. Plot APU map with operating line (see code)
6. Explore simulation results using Simscape Results Explorer
7. Open the Simscape/Fluids/Gas/Turbomachinery library
8. Learn more about this example

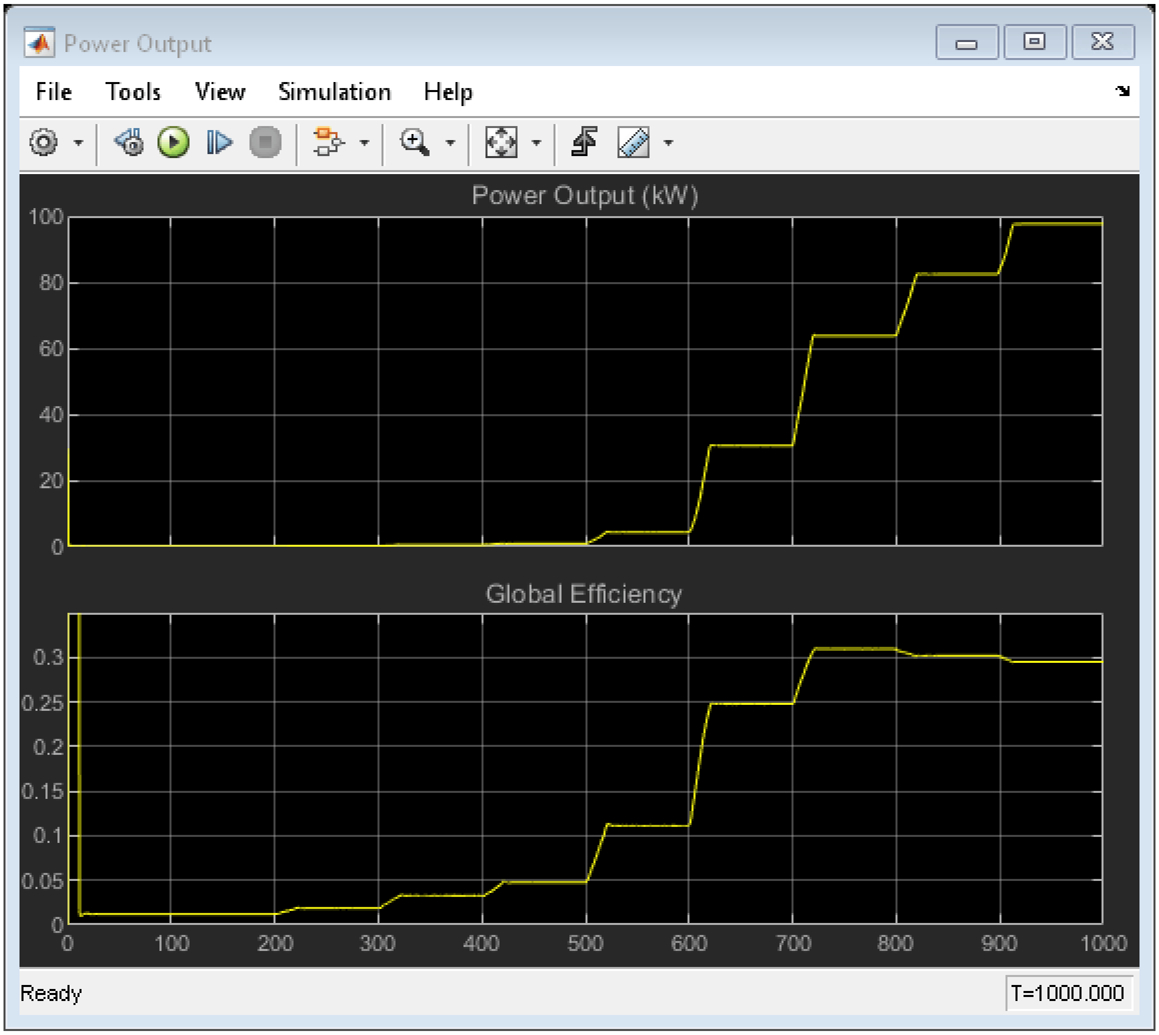
Copyright 2016-2022 The MathWorks, Inc.

Controller Subsystem



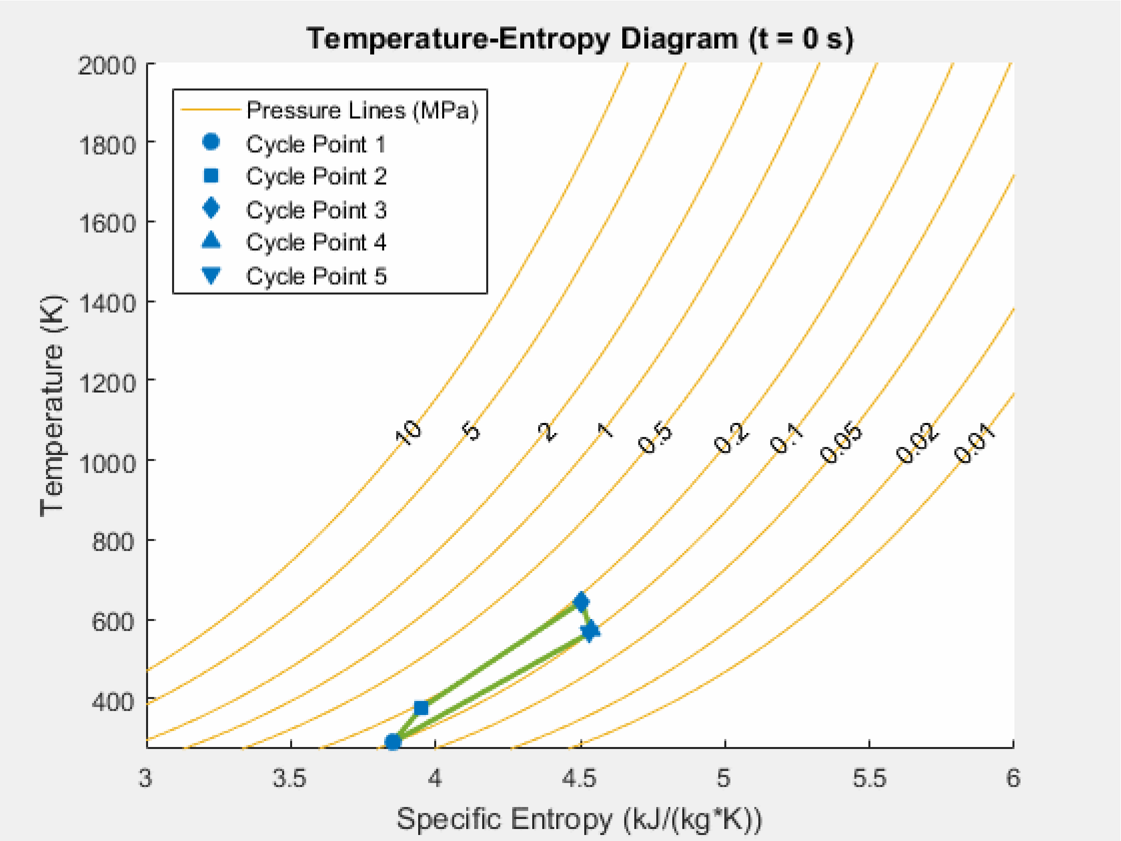
Simulation Results from Scopes



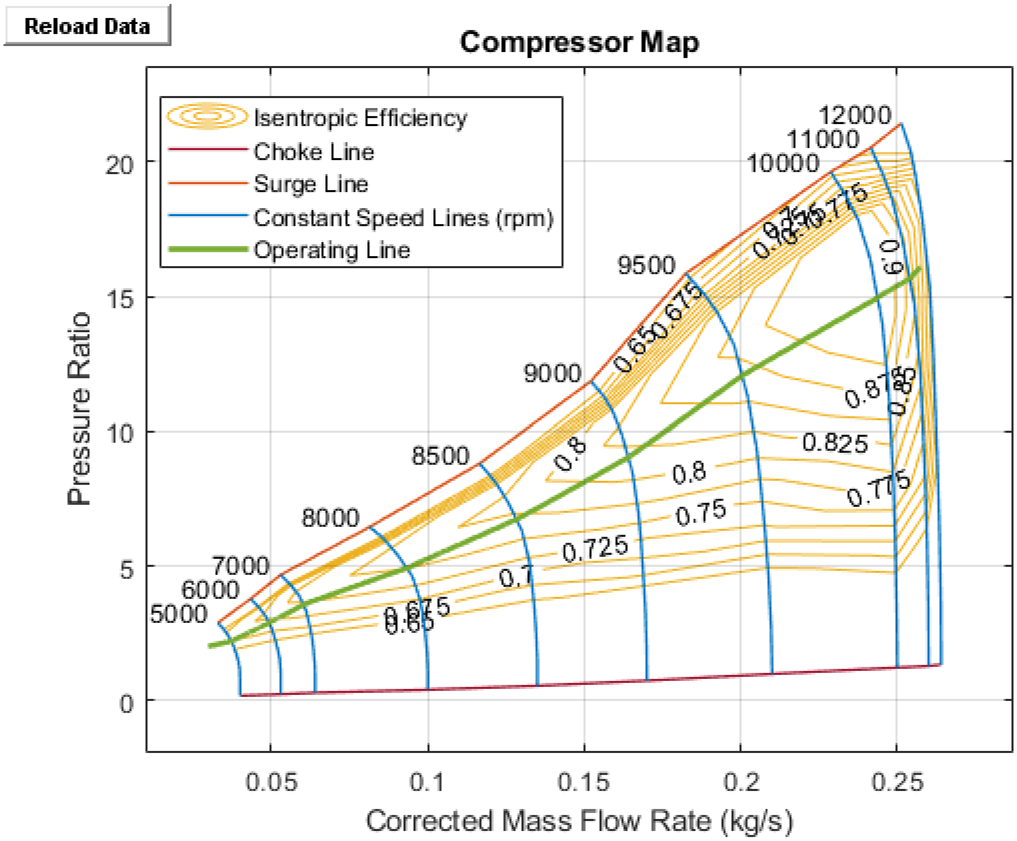


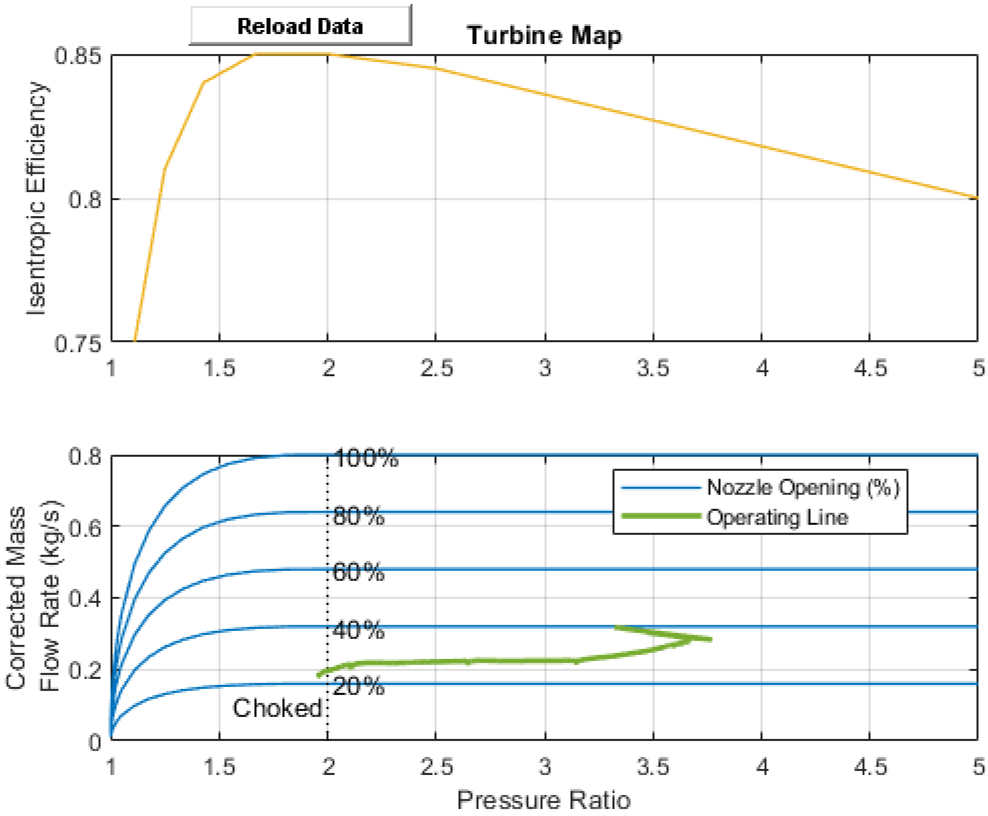
Animation of Simscape Logging Results

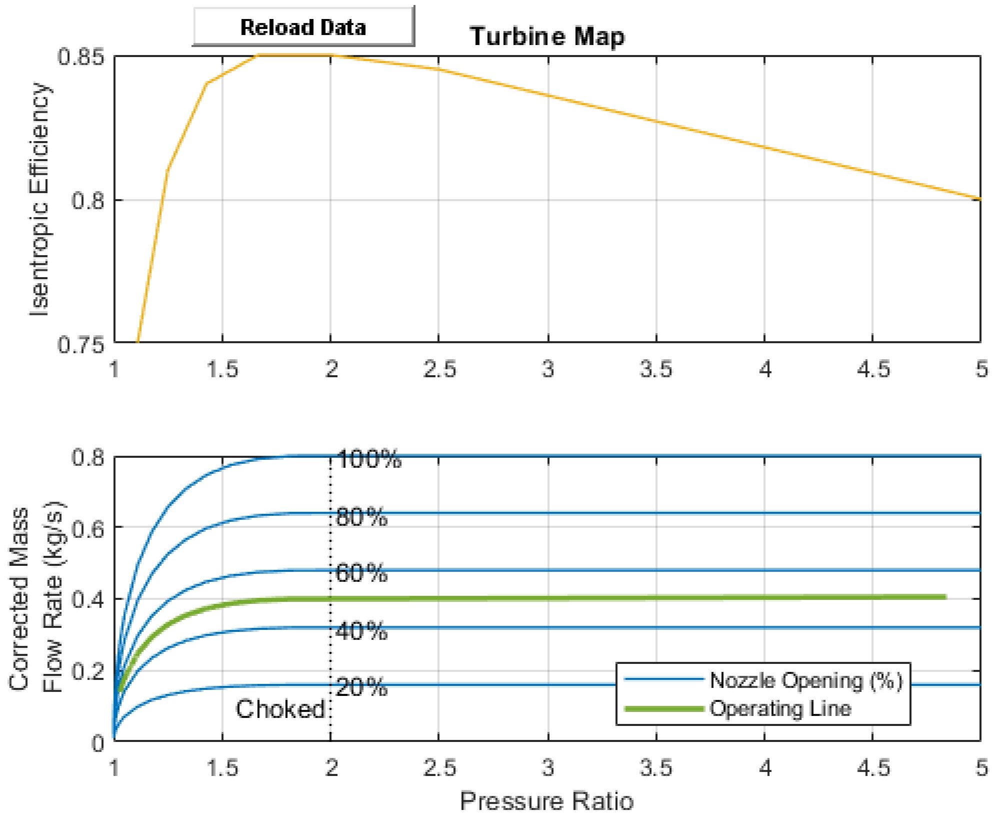
This figure shows an animation of the Brayton Cycle on a temperature-entropy diagram over time. The five cycle points on the figure correspond to the sensor measurements labeled "T,s 1" to "T,s 5", respectively.


Temperature-Entropy Diagram (t = 0 s)
Pressure Lines (MPa)
● Cycle Point 1
■ Cycle Point 2
◆ Cycle Point 3
▲ Cycle Point 4
▼ Cycle Point 5
10
5
2
1
0.5
0.2
0.1
0.05
0.02
0.01
Temperature (K)
2000
1800
1600
1400
1200
1000
800
600
400
Specific Entropy (kJ/(kg*K))
3
3.5
4
4.5
5
5.5
6

Simulation Results from Simscape Logging




Reload Data
Turbine Map
Isentropic Efficiency
0.85
0.8
0.75
1
1.5
2
2.5
3
3.5
4
4.5
5
Corrected Mass
Flow Rate (kg/s)
0.8
0.6
0.4
0.2
0
100%
80%
60%
40%
20%
Choked
Nozzle Opening (%)
Operating Line
1
1.5
2
2.5
3
3.5
4
4.5
5
Pressure Ratio

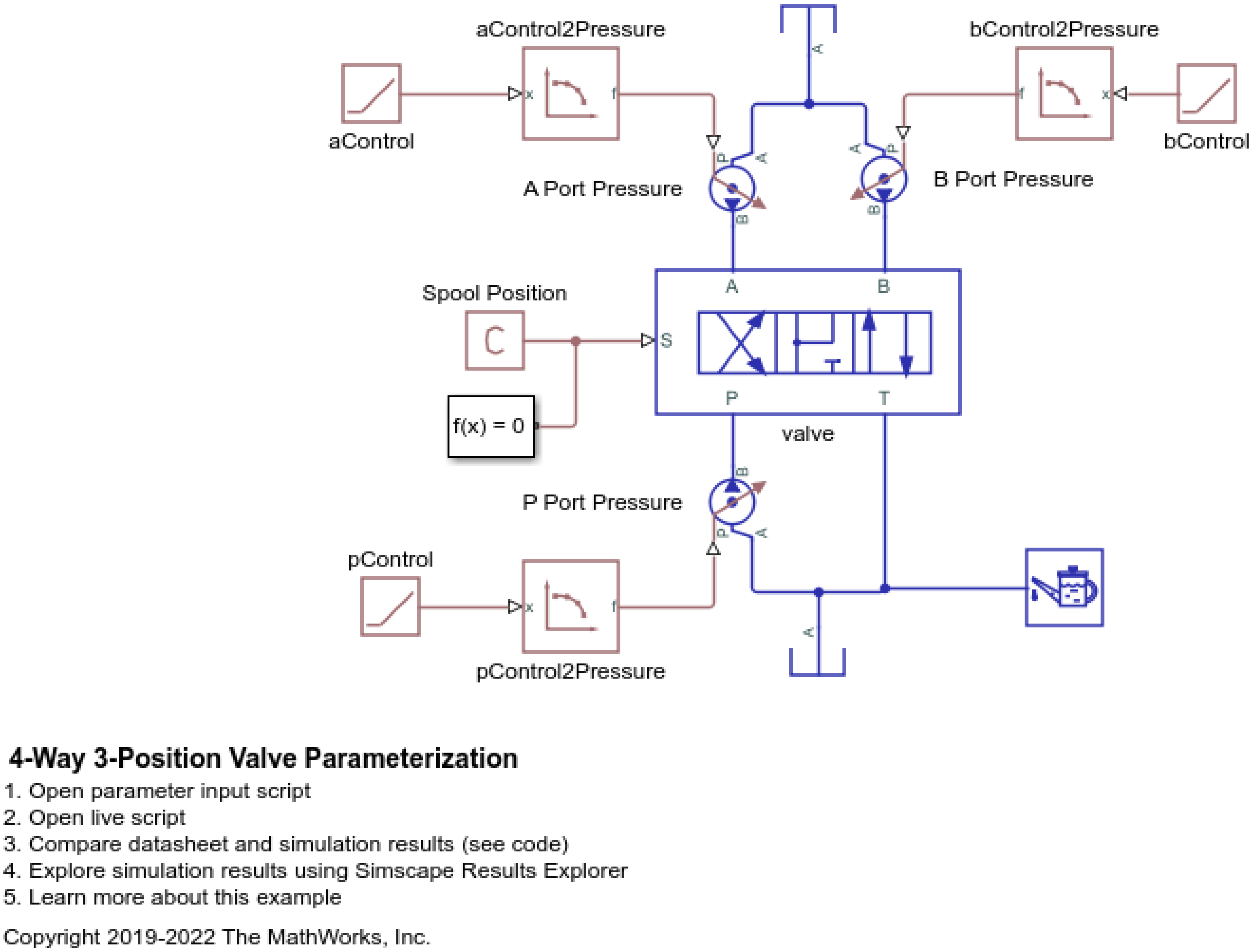

Reload Data
Turbine Map
Isentropic Efficiency
0.85
0.8
0.75
1
1.5
2
2.5
3
3.5
4
4.5
5
Corrected Mass
Flow Rate (kg/s)
0.8
0.6
0.4
0.2
0
100%
80%
60%
40%
20%
Choked
Nozzle Opening (%)
Operating Line
Pressure Ratio

4-Way 3-Position Valve Parameterization

This example shows how to parameterize and test a 4-way 3-position valve with a test harness. A plot script is provided with the example for comparing output flow between the block and data to verify the test harness. A live script is also provided with this example to explain the parameterization and the test harness workflow in detail.

Model

The following figure shows a 4-way 3-position valve test harness. The Pressure Source (IL) blocks; with Ramp blocks are used to create pressure boundaries for the directional control valve. The spool displacement is provided using a Constant block.



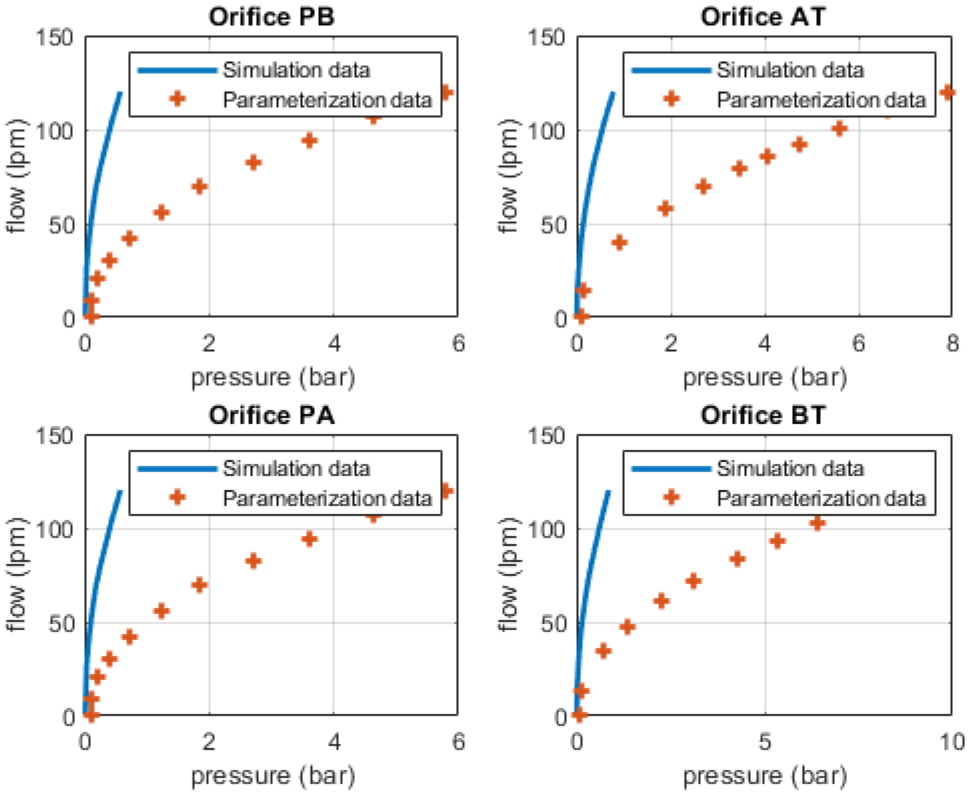
4-Way 3-Position Valve Parameterization

1. Open parameter input script
2. Open live script
3. Compare datasheet and simulation results (see code)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example

Copyright 2019-2022 The MathWorks, Inc.

Simulation results from Simscape™ logging

The model test harness is parameterized for the positive and the negative spool positions and the simulated output is compared with the parameterization setting.

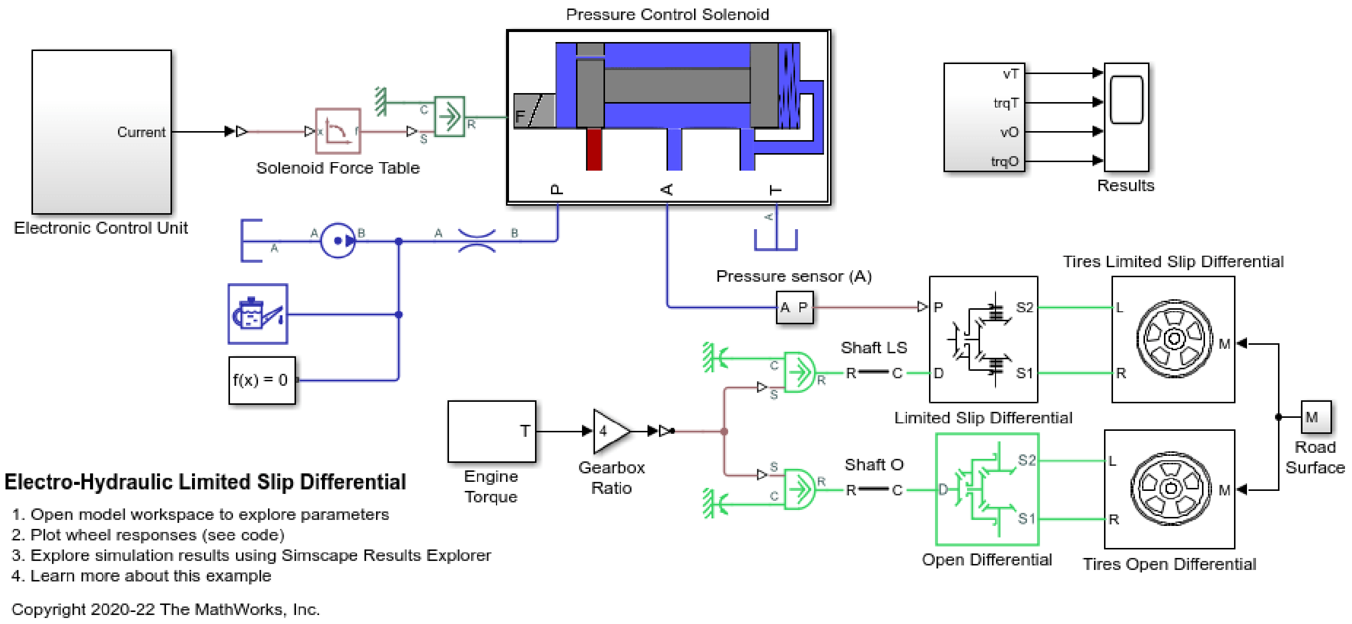

Orifice PB
Simulation data
Parameterization data
flow (lpm)
pressure (bar)
0
50
100
150
0
2
4
6
Orifice AT
Simulation data
Parameterization data
flow (lpm)
pressure (bar)
0
50
100
150
0
2
4
6
8
Orifice PA
Simulation data
Parameterization data
flow (lpm)
pressure (bar)
0
50
100
150
0
2
4
6
Orifice BT
Simulation data
Parameterization data
flow (lpm)
pressure (bar)
0
50
100
150
0
5
10

Electro-Hydraulic Limited Slip Differential

This example shows a simple way of modeling an electro-hydraulic limited slip differential system. The model shows the velocity and the torque profile responses achieved for the left and the right wheels.

Model

The following figure shows the model of an electro-hydraulic limited slip differential system with a test harness. The electronic control unit sends a command to the pressure control solenoid valve based on the relative slip between the right and the left wheels. The pressure control solenoid valve applies a pressure on the actuating mechanism which in turn applies the limited slip control.



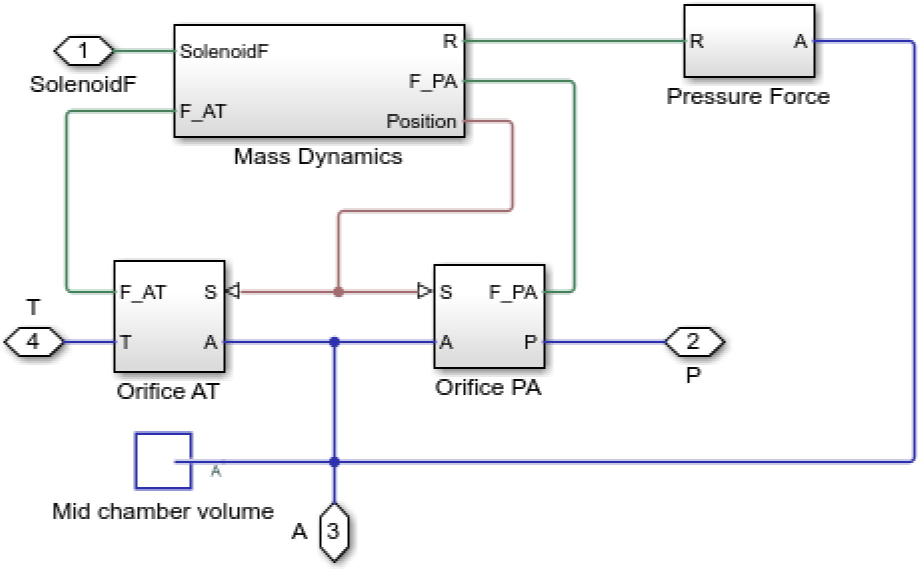
Electro-Hydraulic Limited Slip Differential

1. Open model workspace to explore parameters
2. Plot wheel responses (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2020-22 The MathWorks, Inc.

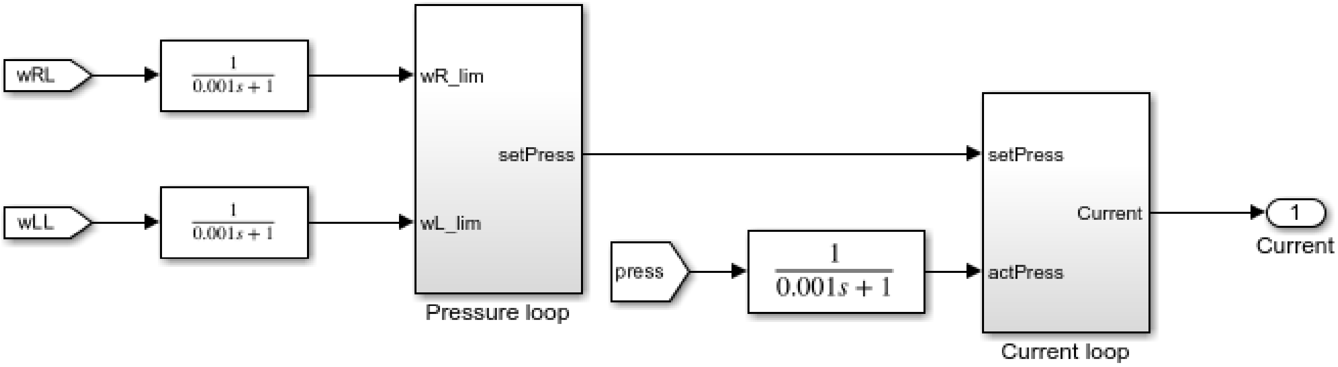
Pressure Control Solenoid Subsystem

This subsystem shows the modeling of the pressure control solenoid valve.



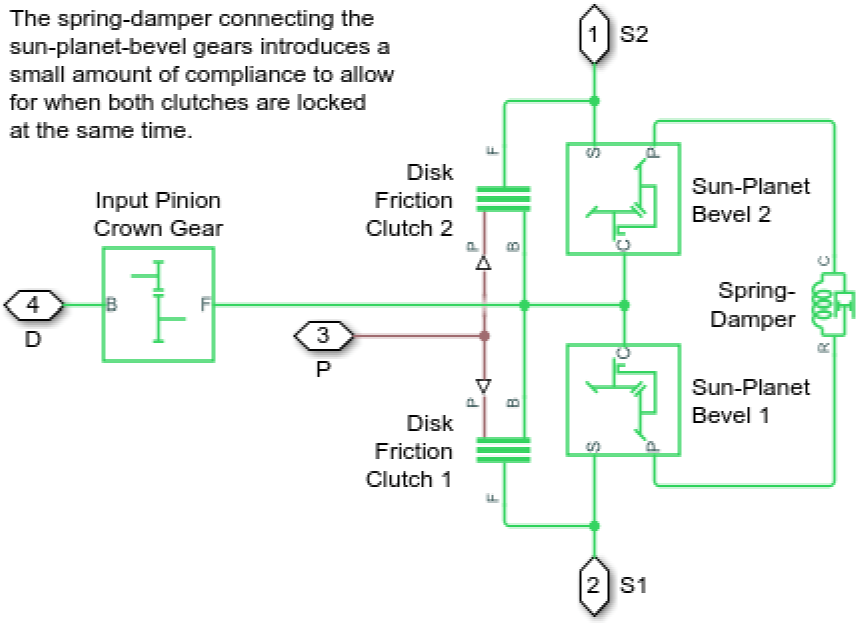
Electronic Control Unit Subsystem

This subsystem shows the modeling of the control of the pressure control solenoid.



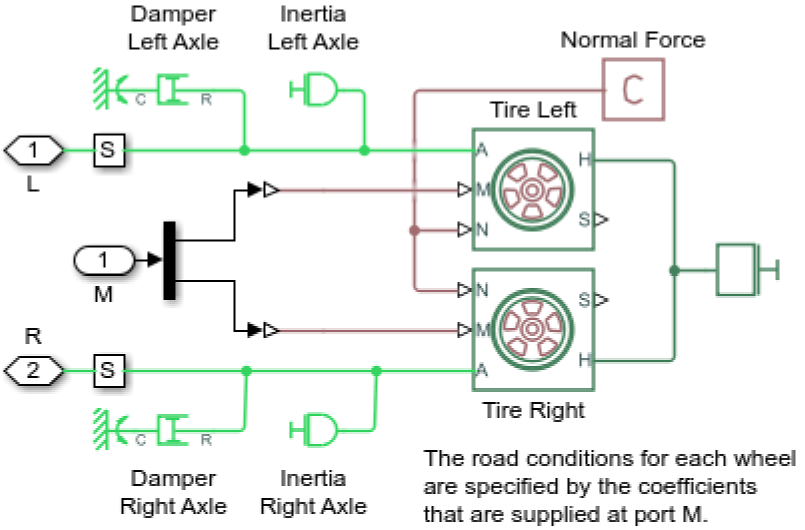
Limited Slip Differential Subsystem

This subsystem shows the modeling of the limited slip differential.


The spring-damper connecting the sun-planet-bevel gears introduces a small amount of compliance to allow for when both clutches are locked at the same time.
1
S2
Input Pinion Crown Gear
Disk Friction Clutch 2
Sun-Planet Bevel 2
4
D
B
F
3
P
Spring-Damper
C
R
Sun-Planet Bevel 1
Disk Friction Clutch 1
2
S1

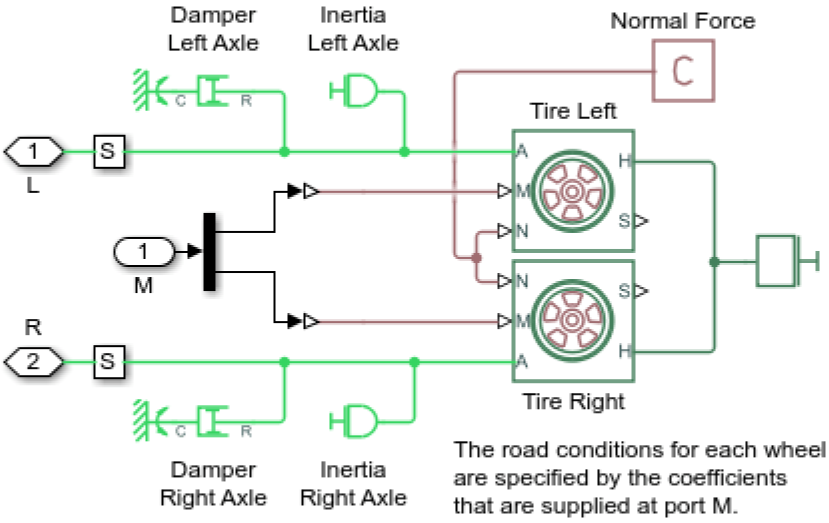
Tires Limited Slip Differential Subsystem

This subsystem shows the modeling of the tires for limited slip differential.



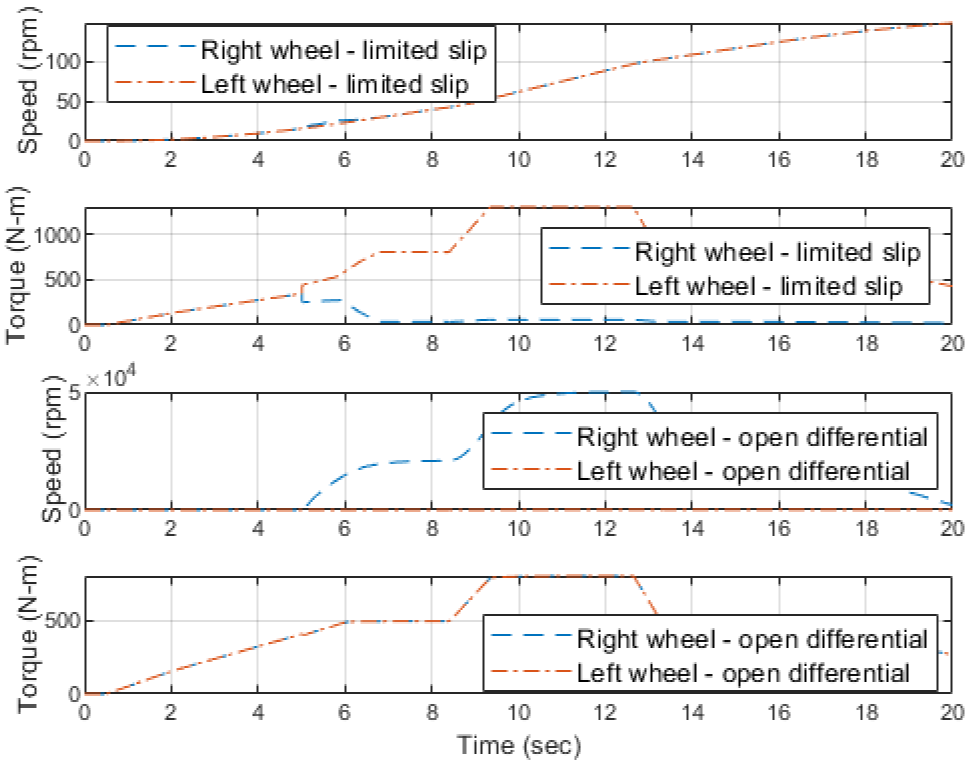
Tires Open Differential Subsystem

This subsystem shows the modeling of the tires for open differential.



Simulation results from Simscape™ logging

Based on the given inputs, the model generates a plot of the velocity and the torque profile achieved for the left and the right wheels.

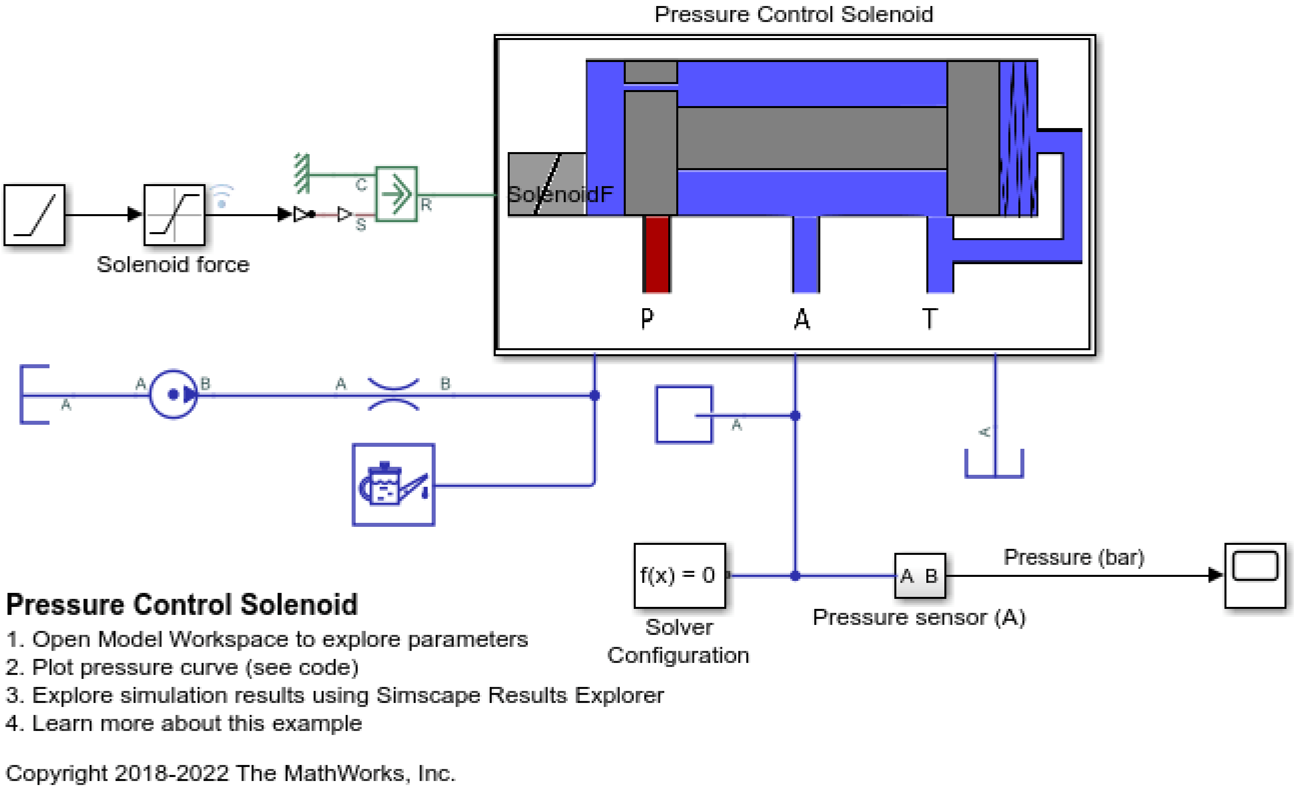


Pressure Control Solenoid

This example shows how to model, parameterize, and test a pressure control solenoid valve. This example also generates a plot of the relationship between applied solenoid force and the resulting actuator port pressure.

Model

The following figure shows the model of a pressure control solenoid valve. Here P represents the pressure source port,T represents the reservoir, and A represents the actuator port.



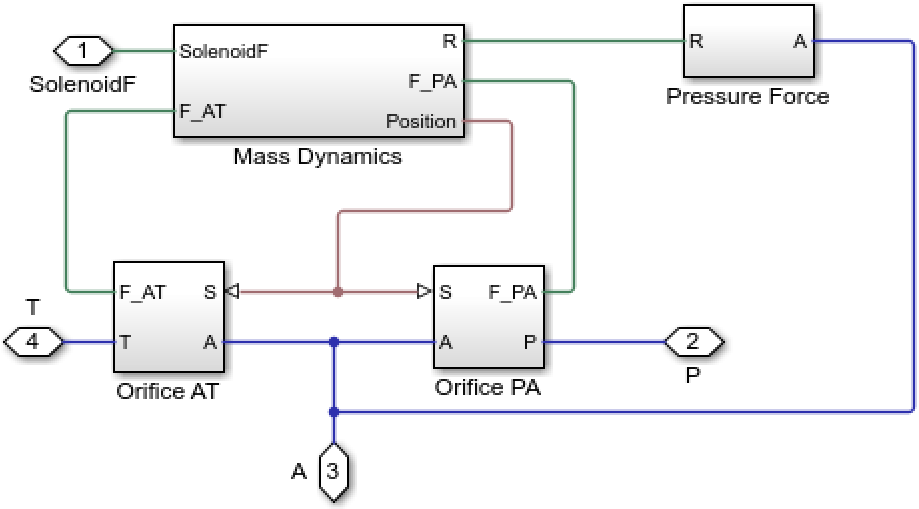
Pressure Control Solenoid

1. Open Model Workspace to explore parameters
2. Plot pressure curve (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2018-2022 The MathWorks, Inc.

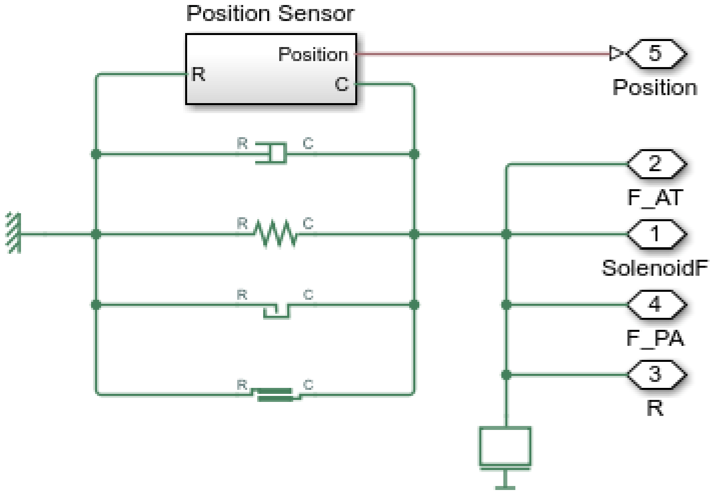
Pressure Control Solenoid Subsystem

This subsystem shows how the pressure control solenoid valve is modelled.



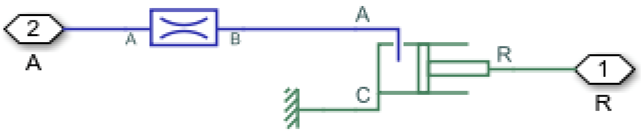
Mass Dynamics Subsystem

This subsystem shows how the spool mass dynamics are modelled, including the forces on the spool due to spool inertia; the forces on the spool due to the spring, damper, and hard stop; friction between the spool and the valve body; the change in fluid momentum at the valve entry and exit; and the pressure acting on the spool surface.



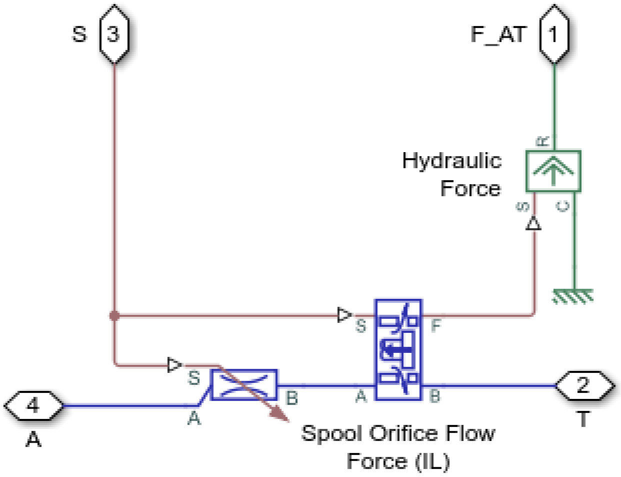
Pressure Force Subsystem

This subsystem shows how the fluid pressure at port A is modelled as a force on the spool left face.



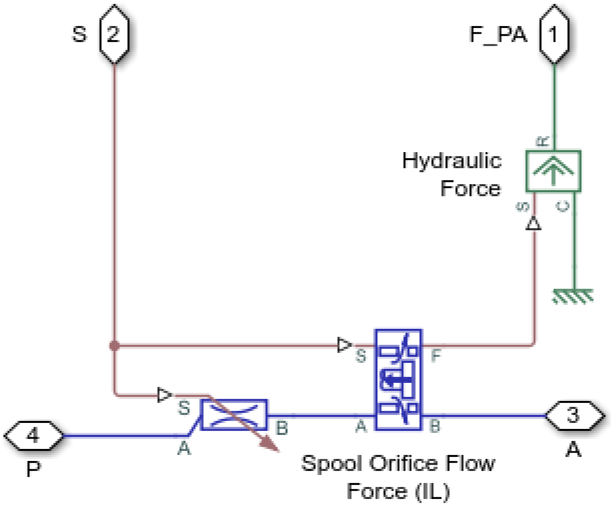
Orifice AT Subsystem

This subsystem shows how the orifice between ports A and T is modelled.



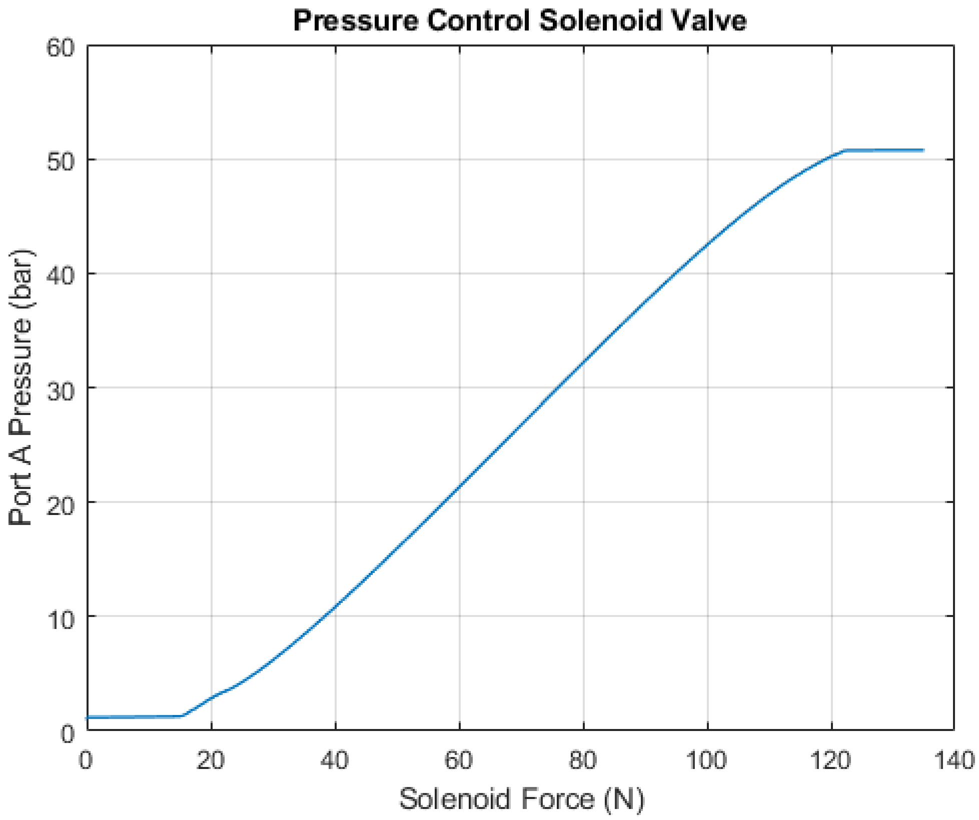
Orifice PA Subsystem

This subsystem shows how the orifice between ports A and P is modelled.



Simulation Results from Simscape™ Logging

This model generates a plot between the applied solenoid force and the generated actuator port pressure.

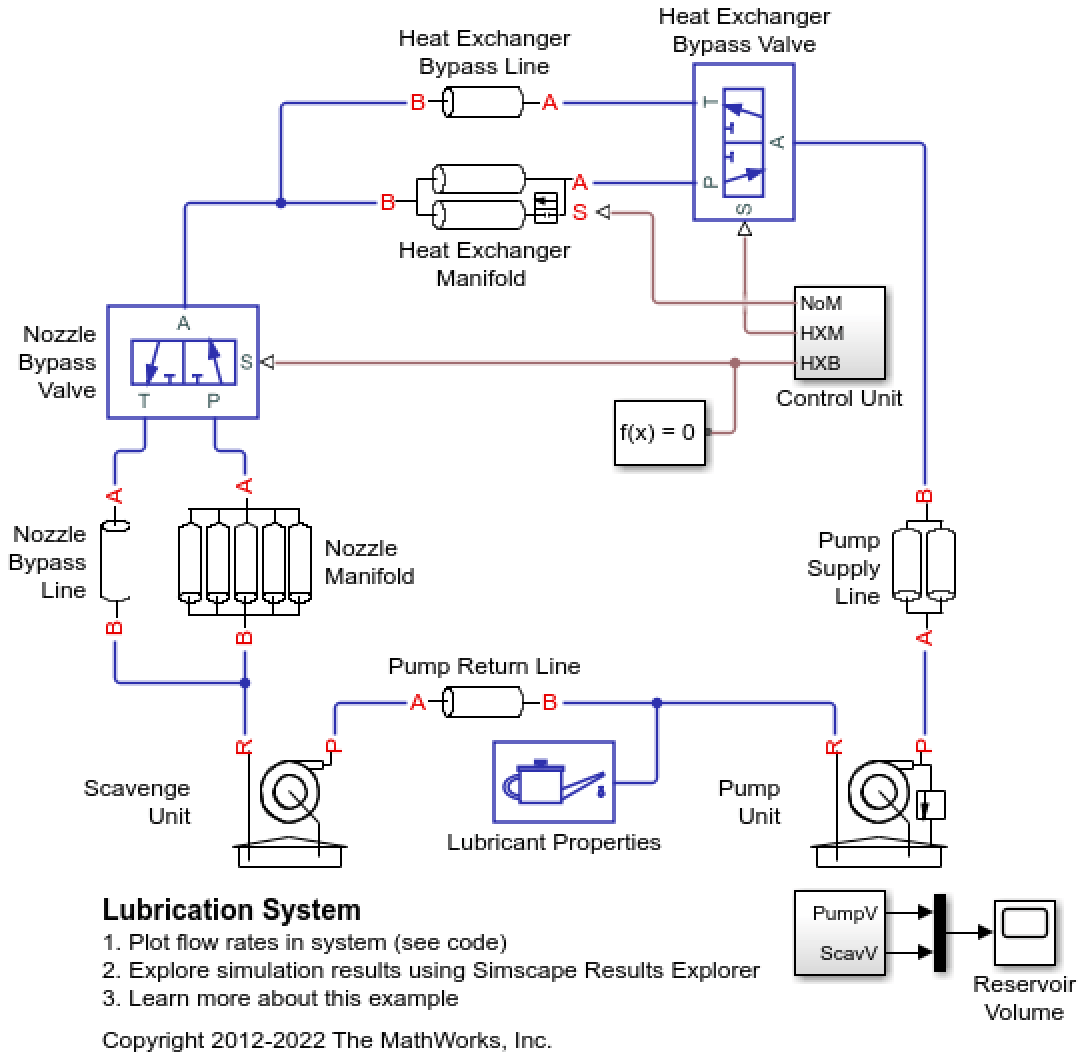


Lubrication System

This example shows a simplified version of a lubrication system fed with the centrifugal pump. The system consists of five major units: Pump Unit, Scavenge Unit, Heat Exchanger Manifold, Nozzle Manifold, and Control Unit. Both the pump and the scavenge unit are built around the centrifugal pump. The Scavenge Unit collects fluid discharged by nozzles and pumps it back into the reservoir of the Pump Unit. The Control Unit generates commands to bypass either the heat exchanger, represented as a local resistance, or the nozzles block. In a real system, these commands are generated by temperature sensors installed in lubrication cavities.

The cycle simulated in the example allows you to observe pressures and flow rates at various points of the system. The example also shows the usage of local resistance blocks such as the Elbow, T-junction, and Sudden Area Change.

Model

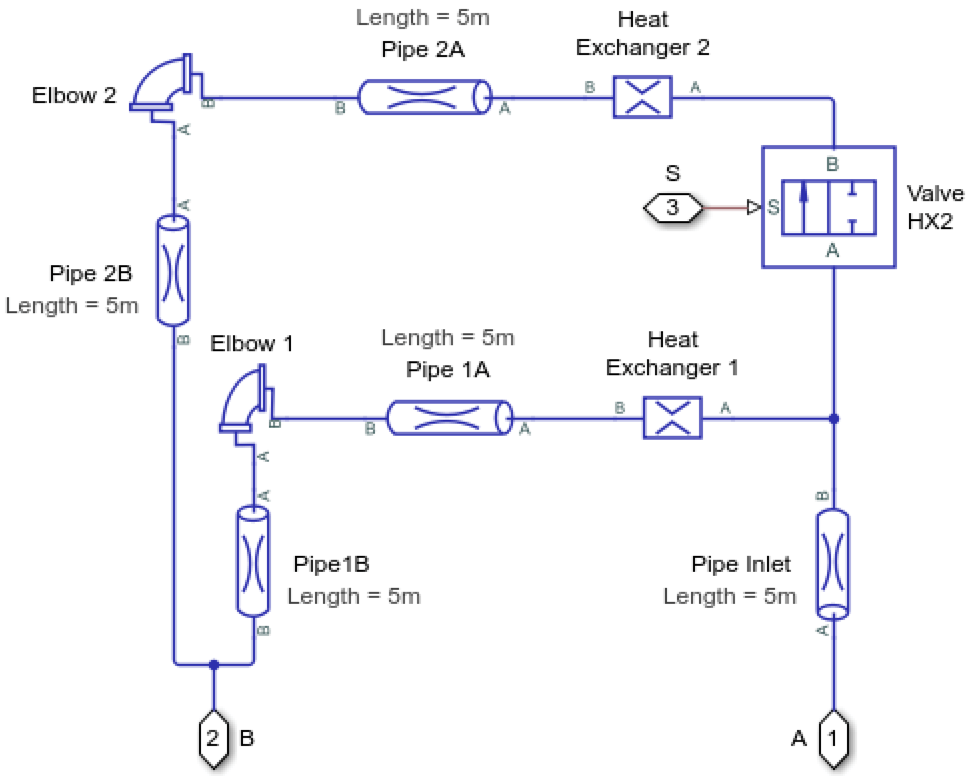


Lubrication System

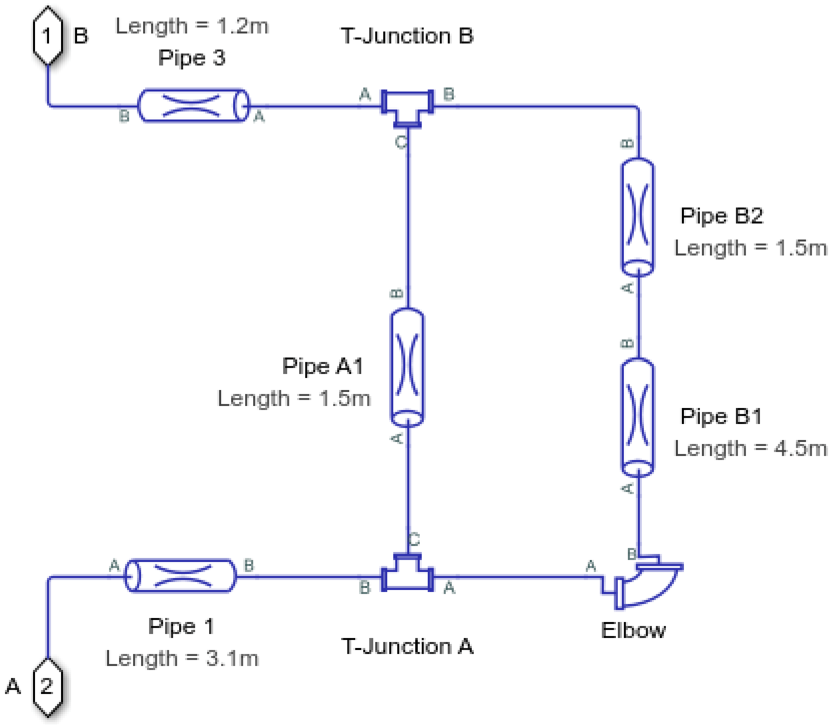
1. Plot flow rates in system (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2012-2022 The MathWorks, Inc.

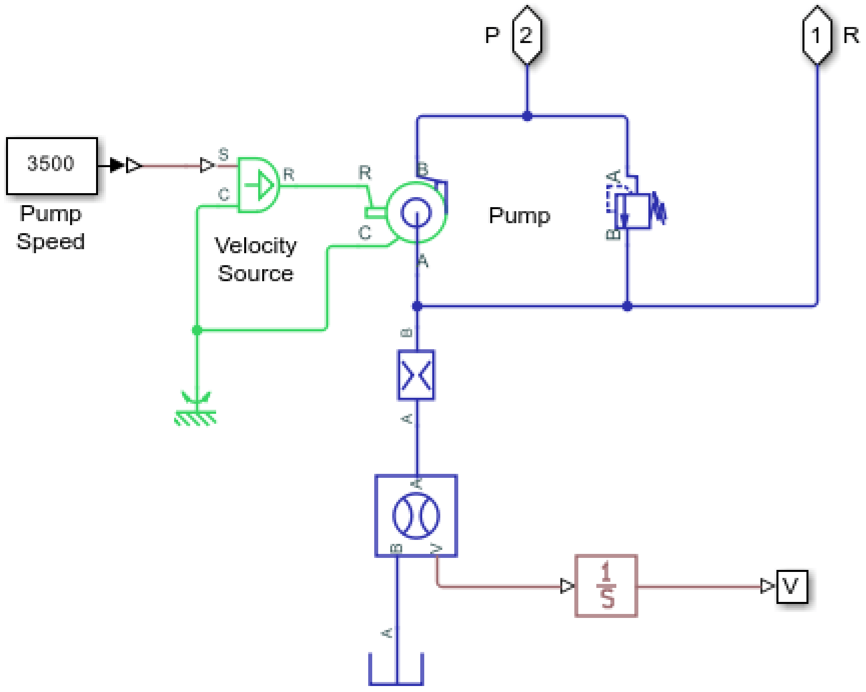
Heat Exchanger Manifold Subsystem



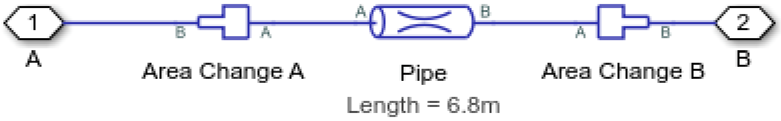
Pump Supply Line Subsystem



Pump Unit Subsystem

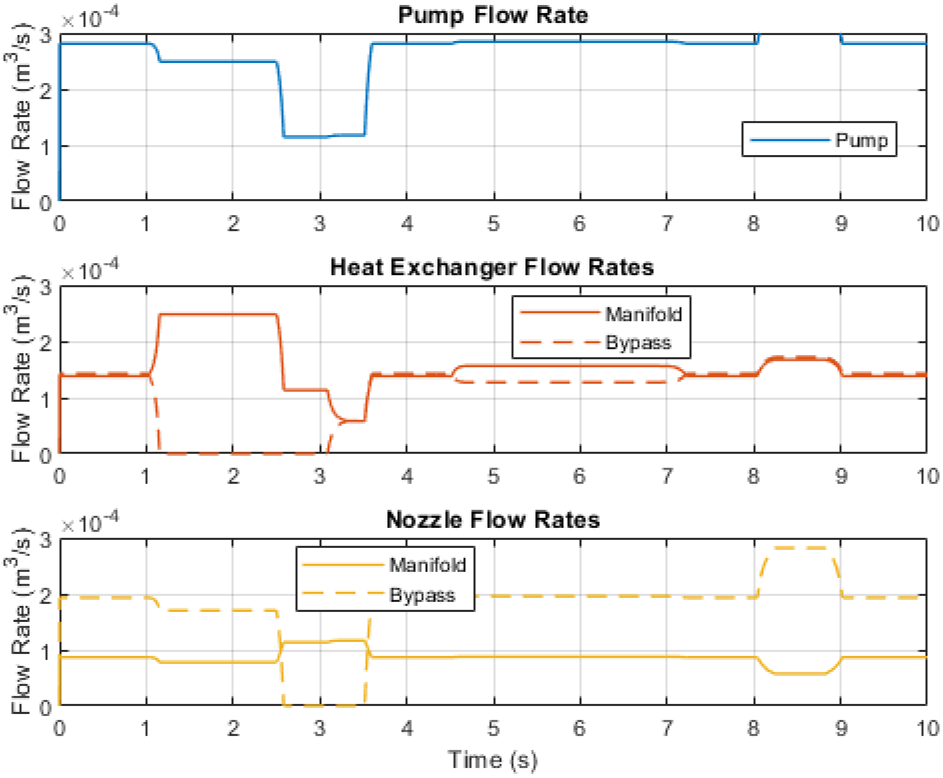


Pump Return Line Subsystem



Simulation Results from Simscape Logging

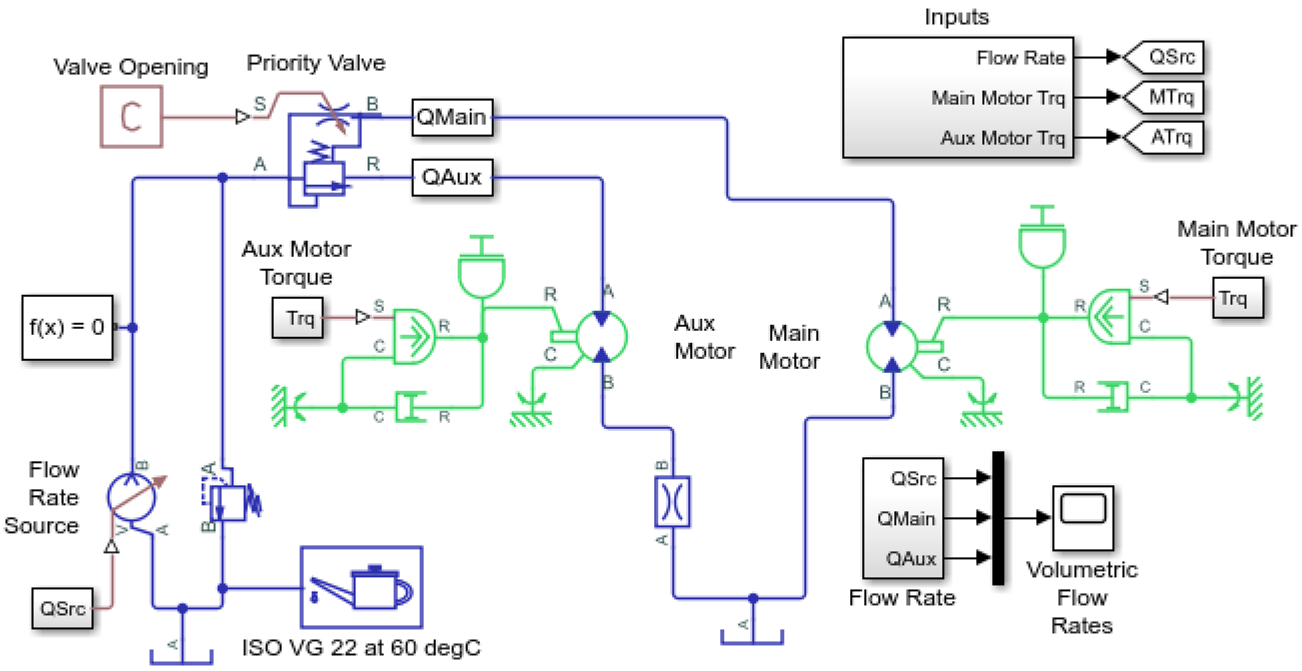
The plots below show the flow rates through different parts of the lubrication system. As the control member within the different valves are adjusted, flow rates change within the system.



Priority Valve Controlling Two Hydraulic Motors

This example shows a pressure-compensated 3-way flow control valve. This valve maintains constant flow rate through the main hydraulic motor, which is connected to the pressure-compensated outlet of the flow control valve. It acts as a priority valve, diverting the excess flow to the auxiliary hydraulic motor if the main hydraulic motor receives enough fluid to maintain a preset angular velocity. The auxiliary motor is shut off completely if there is insufficient flow to power the main hydraulic motor.

Model

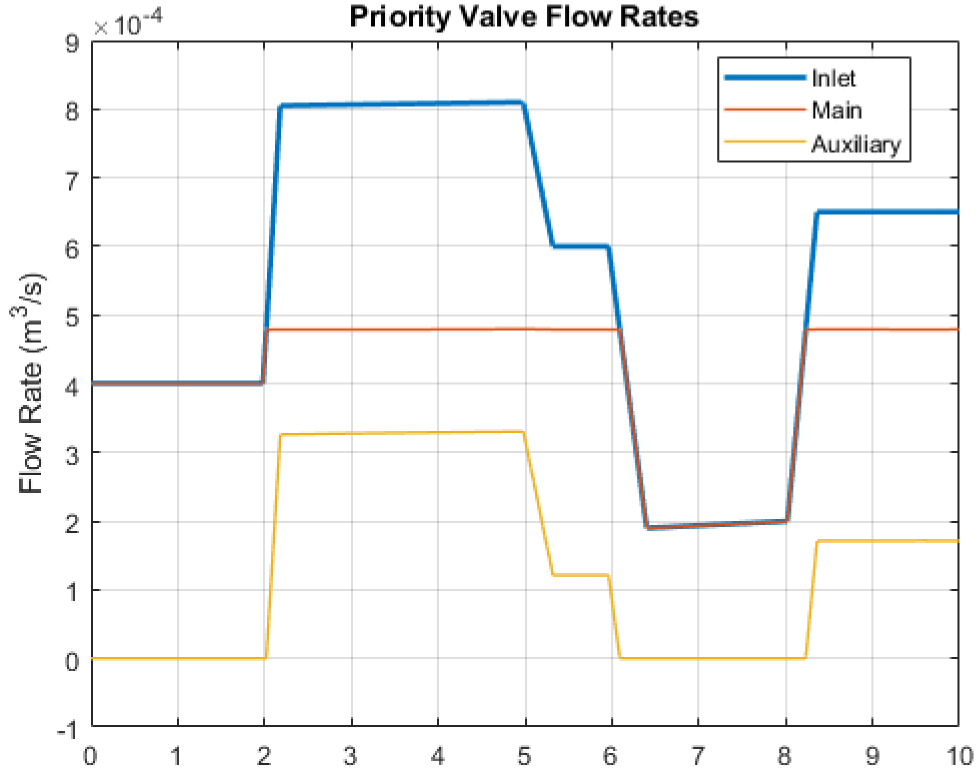


Priority Valve Controlling Two Hydraulic Motors

1. Plot flow rates through priority valve (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Simulation Results from Simscape Logging

The plot below shows the flow rates through the priority valve. As the flow rate at the inlet varies, the main motor is driven at the desired speed and excess flow is directed to the auxiliary hydraulic motor. When the inlet flow drops below the threshold to maintain the speed for the main hydraulic motor, the main motor slows down as well.


Priority Valve Flow Rates
× 10^-4
Flow Rate (m^3/s)
9
8
7
6
5
4
3
2
1
0
-1
0
1
2
3
4
5
6
7
8
9
10
Inlet
Main
Auxiliary

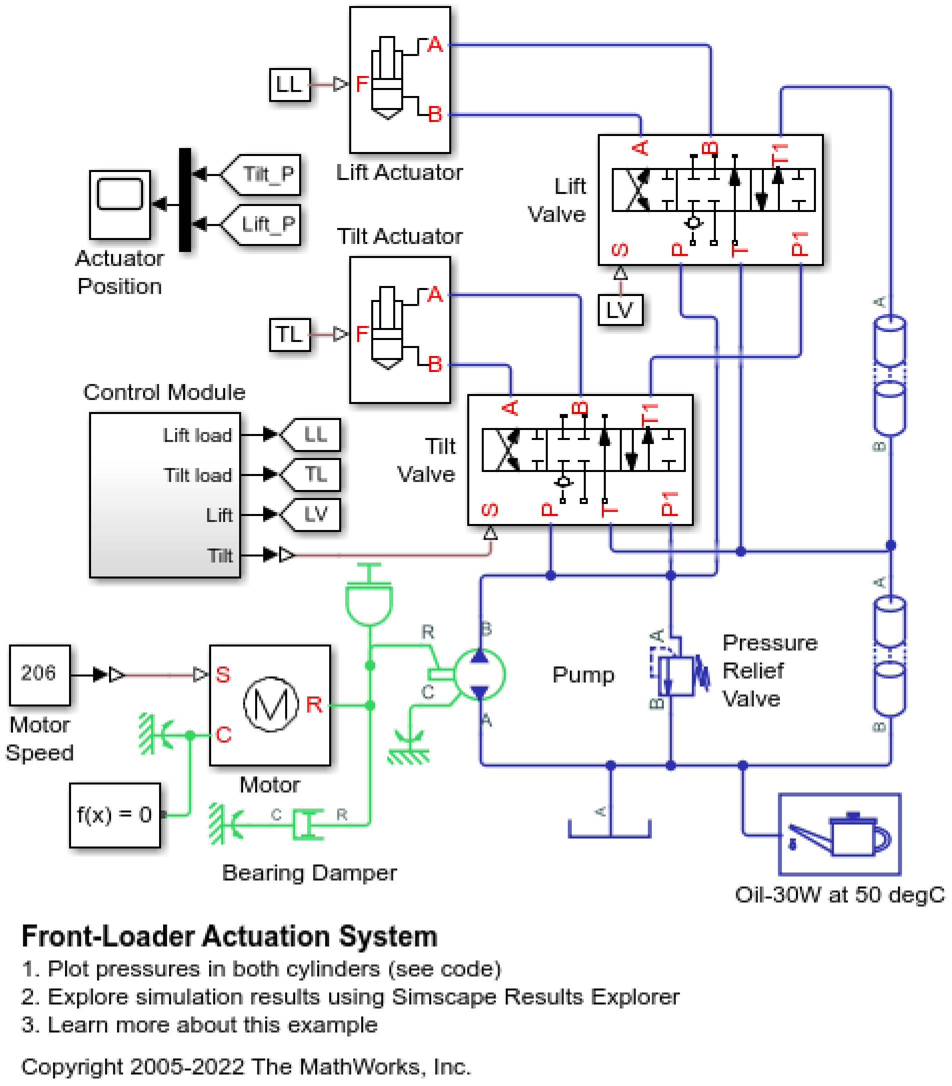
Front-Loader Actuation System

This example shows a simplified version of an actuation system consisting of the lift and tilt cylinders. Each cylinder is controlled by an open center, 6-way, 3-position directional valve. The valves are connected in series through their unloading branch such that the system pump is unloaded when both command levers are in neutral position. If either tilt or lift command is applied, the unloading path is closed.

The pump is driven by a diesel engine simulated with a model that accounts for velocity change caused by the load on the output shaft. The directional valve is a custom model composed of Orifice with Variable Area Round Holes blocks and a Check Valve block.

A typical cycle of a front-loader actuation system is simulated with all the motion and load commands generated as a function of time.

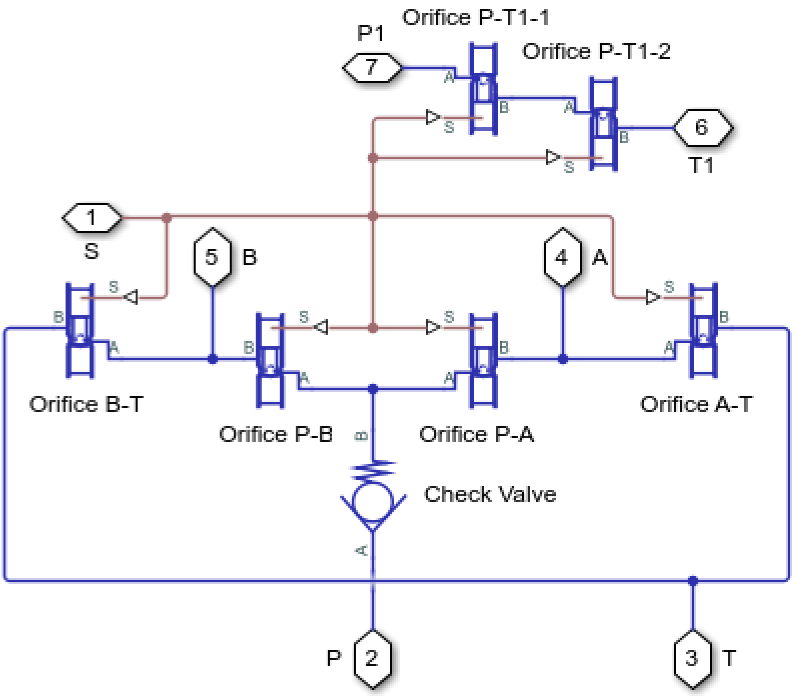
Model



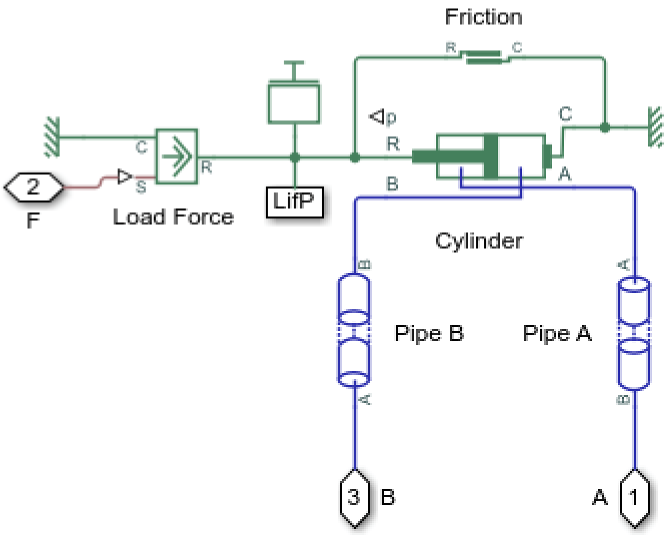
Front-Loader Actuation System

1. Plot pressures in both cylinders (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Lift Valve Subsystem

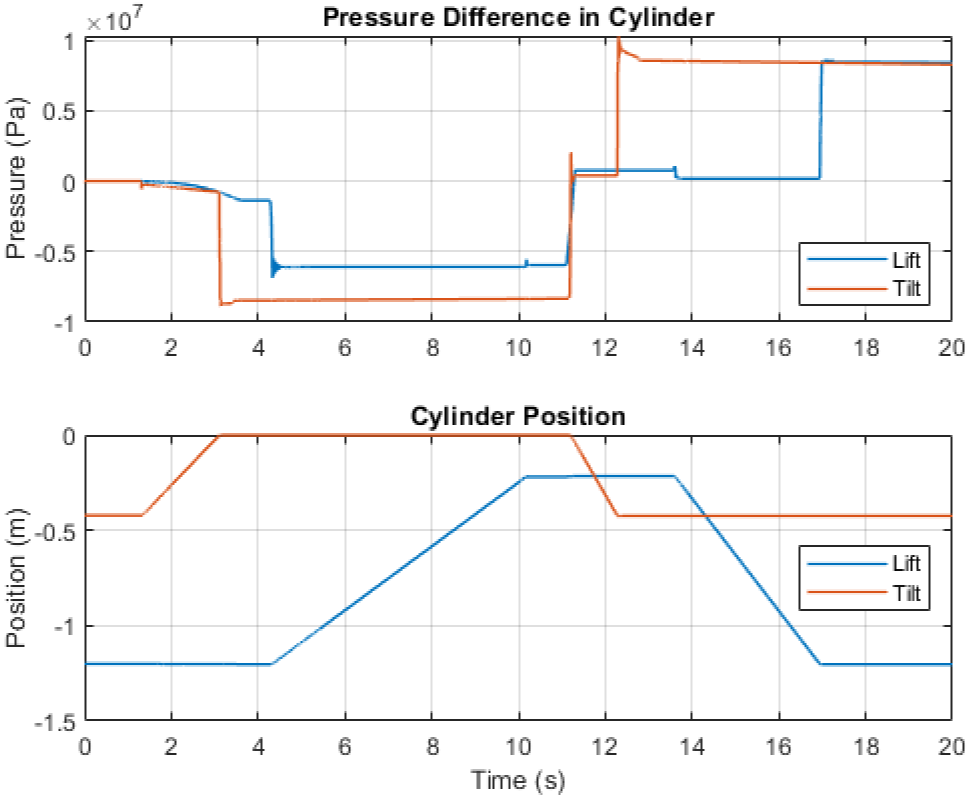


Lift Actuator Subsystem



Simulation Results from Simscape Logging

The plots below show the pressure difference in the lift and tilt cylinders and the position of the in each cylinder.


Pressure Difference in Cylinder
×10^7
Pressure (Pa)
1
0.5
0
-0.5
-1
0 2 4 6 8 10 12 14 16 18 20
Lift
Tilt
Cylinder Position
Position (m)
0
-0.5
-1
-1.5
0 2 4 6 8 10 12 14 16 18 20
Time (s)
Lift
Tilt

Drill-Ream Actuator

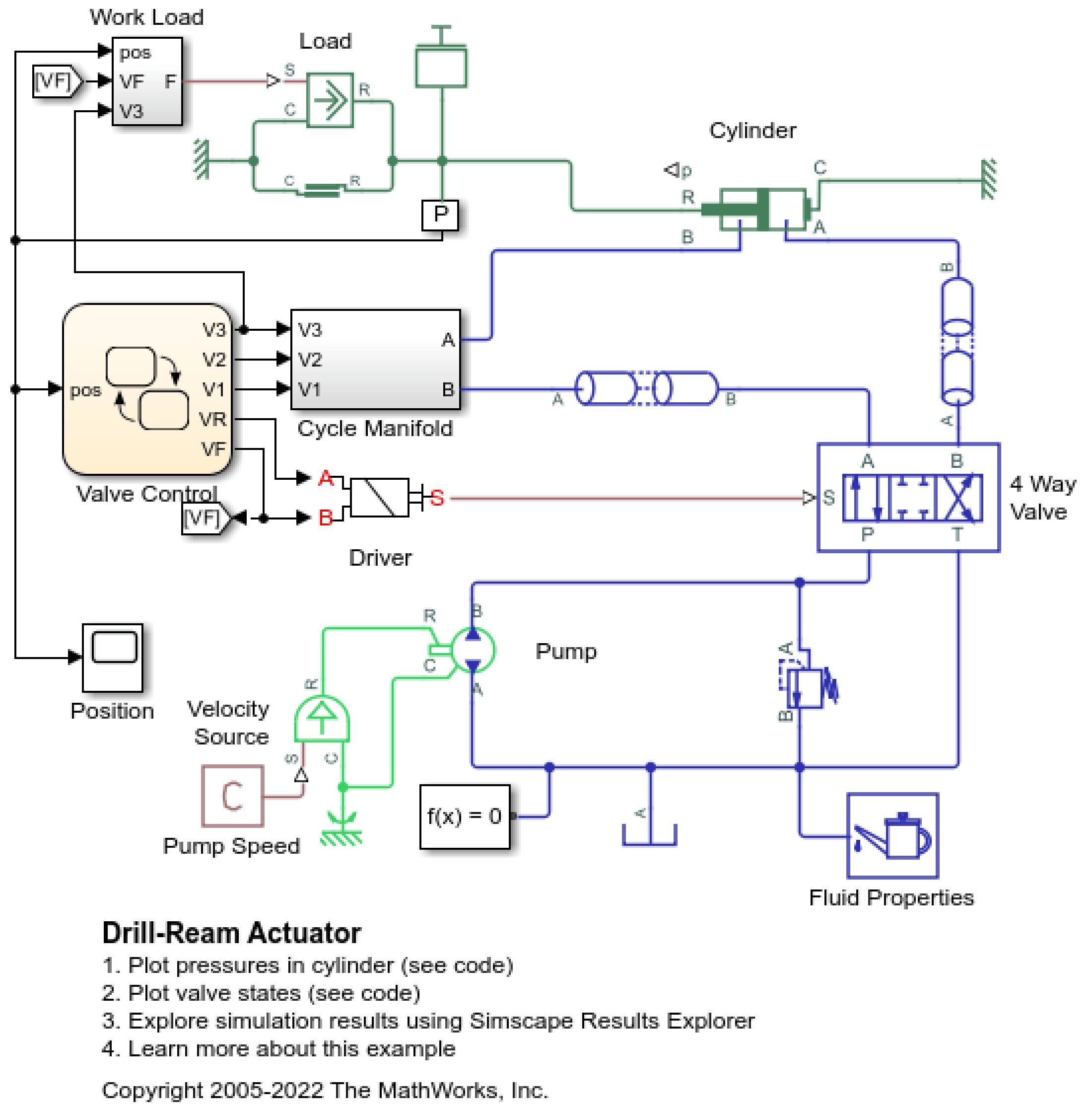
This example shows an actuator that drives a machine tool working unit performing a sequence of three technological operations: coarse drilling, fine drilling, and reaming. The actuator speed is controlled by one of three pressure-compensated flow control valves metering out return flow from the cylinder. The selection of an appropriate flow control is performed by directional valves that are activated by a control unit.

From the Valve Control, signals VF and VR are the signals that activate forward and return motions, respectively, by shifting the main directional valve. Signal V1 activates normally- closed valve Valve1 Retract. When opened, Valve1 Retract bypasses all flow controls and sets the fast traverse speed. V2 controls valve Valve2 Ream. When deenergized, this valve directs return flow through Flow Control Reaming, which is set to reaming speed. Signal V3 directs flow either through Flow Control Fine Drilling or Flow Control Coarse Drilling.

The load is simulated as a position-dependent force using a table lookup block.

Flow control valves are arranged as a subsystem comprised of a fixed orifice and a pressure compensator.

Model

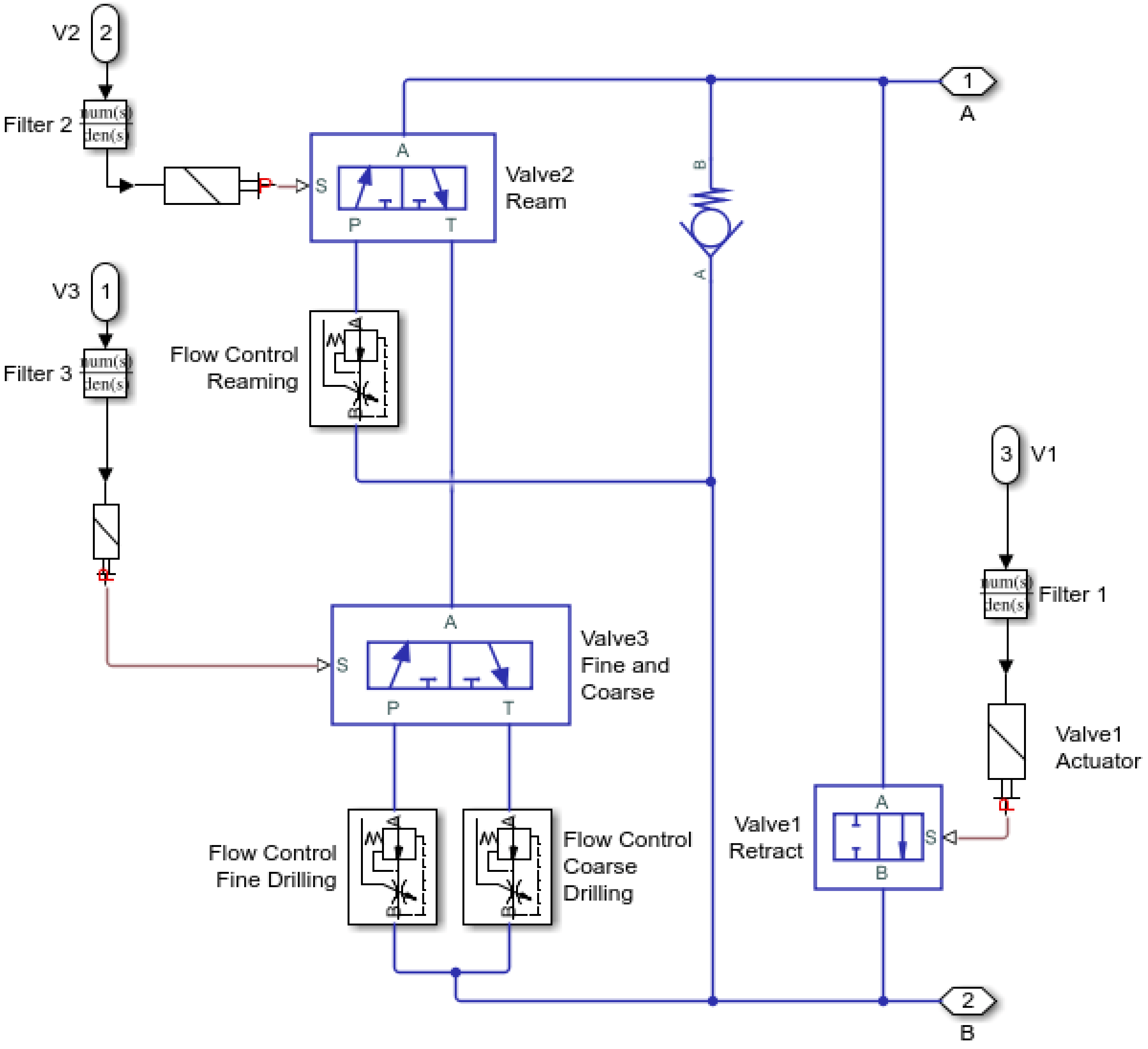


Drill-Ream Actuator

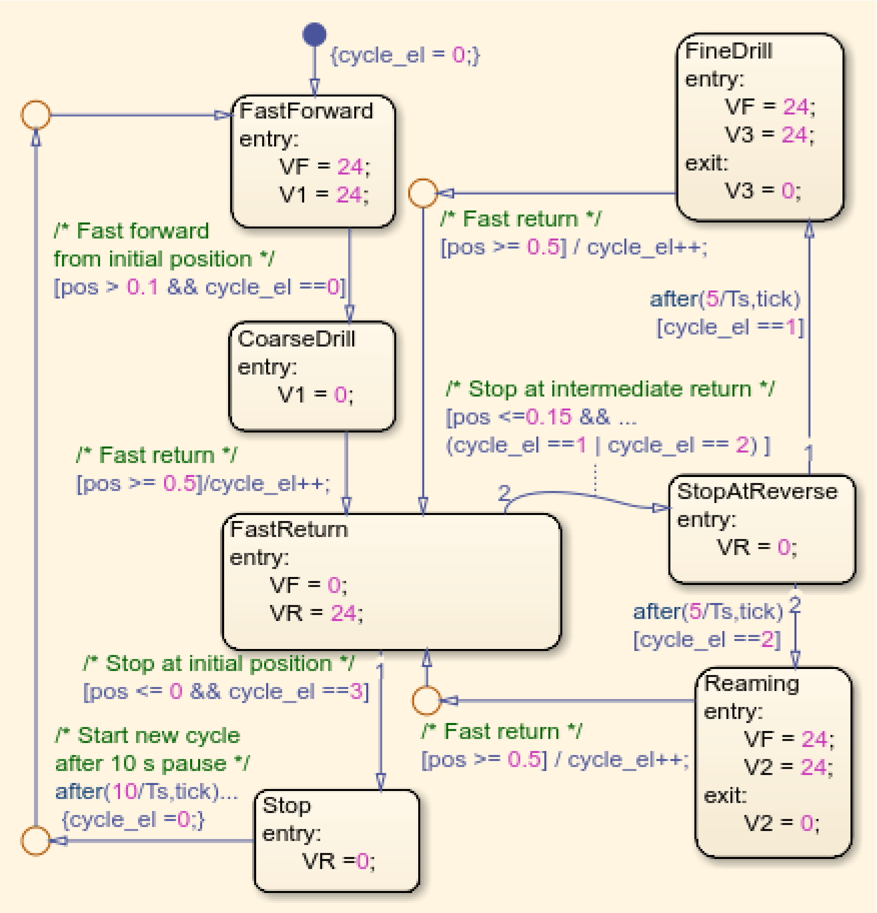
1. Plot pressures in cylinder (see code)
2. Plot valve states (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2005-2022 The MathWorks, Inc.

Cycle Manifold Subsystem

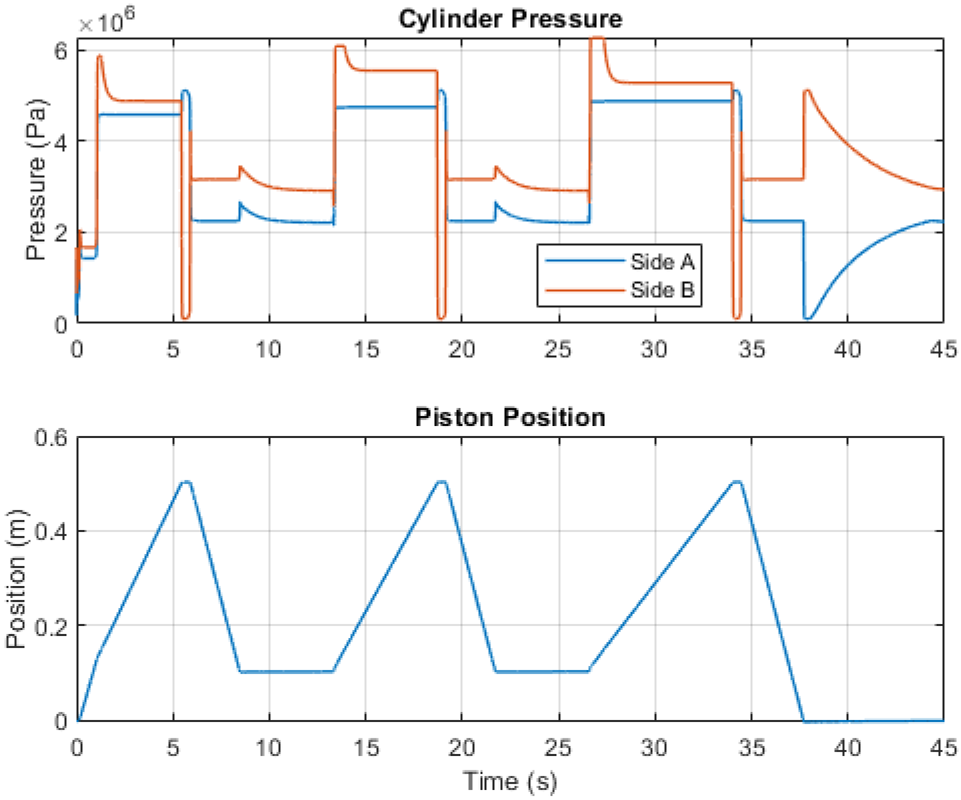


Valve Control Subsystem

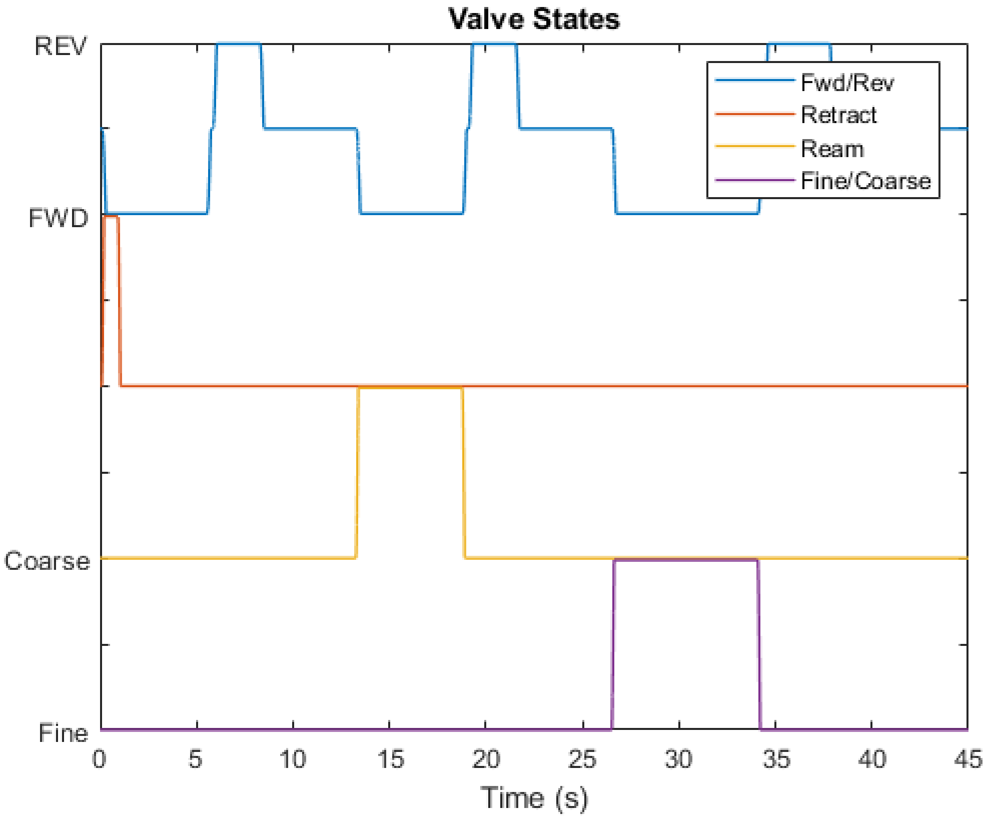


Simulation Results from Simscape Logging

The plots below show the pressures in the cylinder and the position of the piston. Three operating speeds are shown in the plots.



The plots below show the states of the different valves. Forward/Reverse has 3 states (Forward/Locked/Reverse), while the others have two states. Retract and Ream have states of on/off, and Fine/Coarse valve selects between two different flow restrictions.


Valve States
REV
FWD
Coarse
Fine
0
5
10
15
20
25
30
35
40
45
Time (s)
Fwd/Rev
Retract
Ream
Fine/Coarse

Electric Vehicle Thermal Management

This example models the thermal management system of a battery electric vehicle. The system consists of two coolant loops, a refrigeration loop, and a cabin HVAC loop. The thermal load are the batteries, powertrain, and cabin.

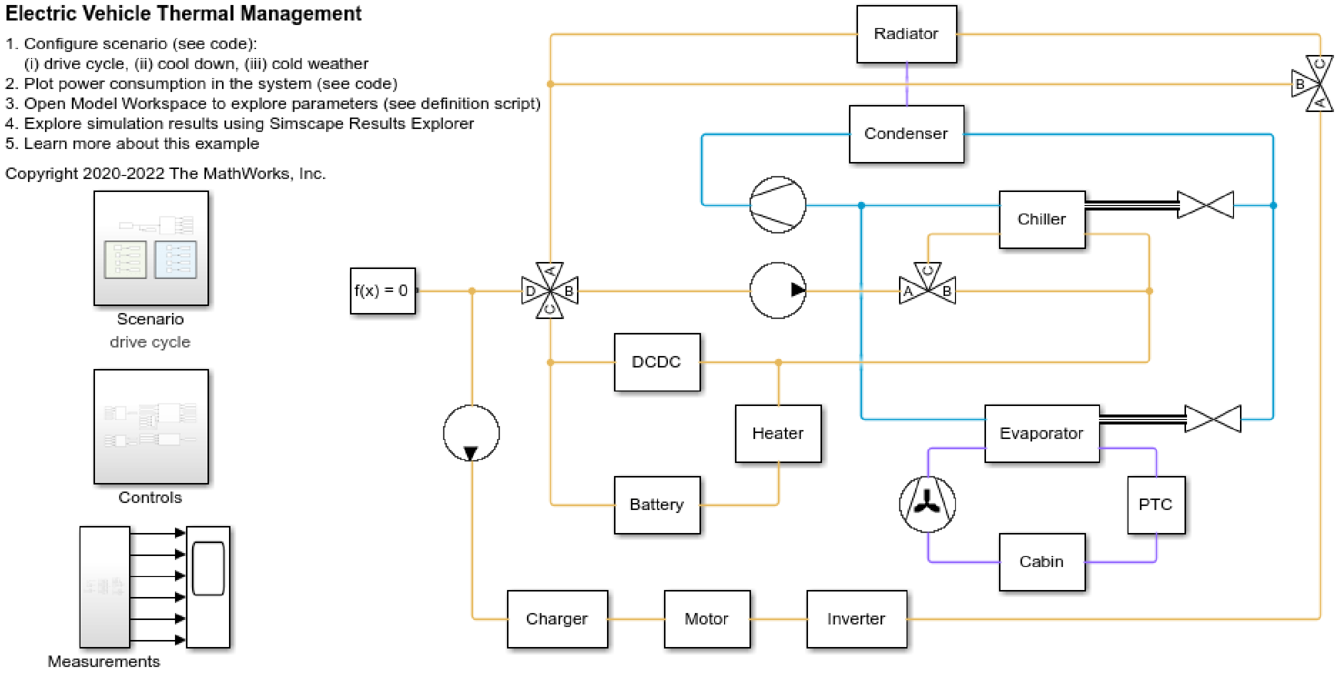
The two coolant loops can be joined together in serial mode or kept separate in parallel mode using the 4-way valve. In cold weather, the coolant loops are in serial mode so that heat from the motor warms the batteries. If necessary, a heater can provide additional heat. In warm weather, the coolant loops remain in serial mode and both the batteries and the powertrain are cooled by the radiator. In hot weather, the coolant loop switches to parallel mode and separates. One loop cools the powertrain using the radiator. The other cools the batteries using the chiller in the refrigeration loop.

The refrigeration loop consists of a compressor, a condenser, a liquid receiver, two expansion valves, a chiller, and an evaporator. The chiller is used to cool the coolant in hot weather when the radiator alone is insufficient. The evaporator is used to cool the vehicle cabin when air conditioning is turned on. The compressor is controlled such that the condenser can dissipate the heat absorbed by either or both the chiller and the evaporator.

The HVAC loop consists of a blower, an evaporator, a PTC heater, and the vehicle cabin. The PTC heater provides heating in cold weather; the evaporator provides air conditioning in hot weather. The blower is controlled to maintain the specified cabin temperature setpoint.

This model has three scenarios set up. The drive cycle scenario simulates driving conditions in 30 degC weather with air conditioning on. The vehicle speed is based on the NEDC followed by 30 min of high speed to push the battery heat load. The cool down scenario simulates a stationary vehicle in 40 degC weather with air conditioning on. Finally, the cold weather scenario simulates driving conditions in -10 degC weather, which requires the battery heater and PTC heater to warm up the batteries and cabin, respectively.

Model



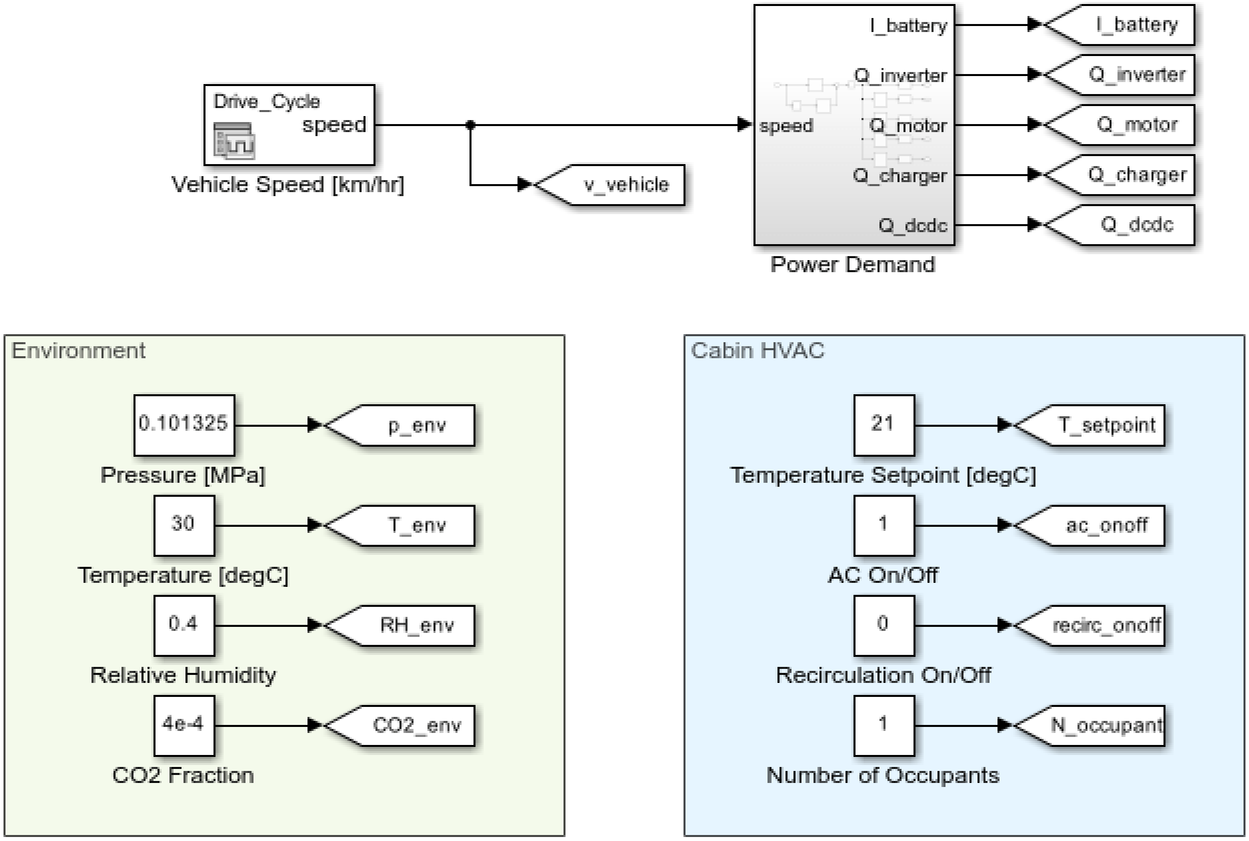
Electric Vehicle Thermal Management

1. Configure scenario (see code):
   (i) drive cycle, (ii) cool down, (iii) cold weather
2. Plot power consumption in the system (see code)
3. Open Model Workspace to explore parameters (see definition script)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example

Copyright 2020-2022 The MathWorks, Inc.

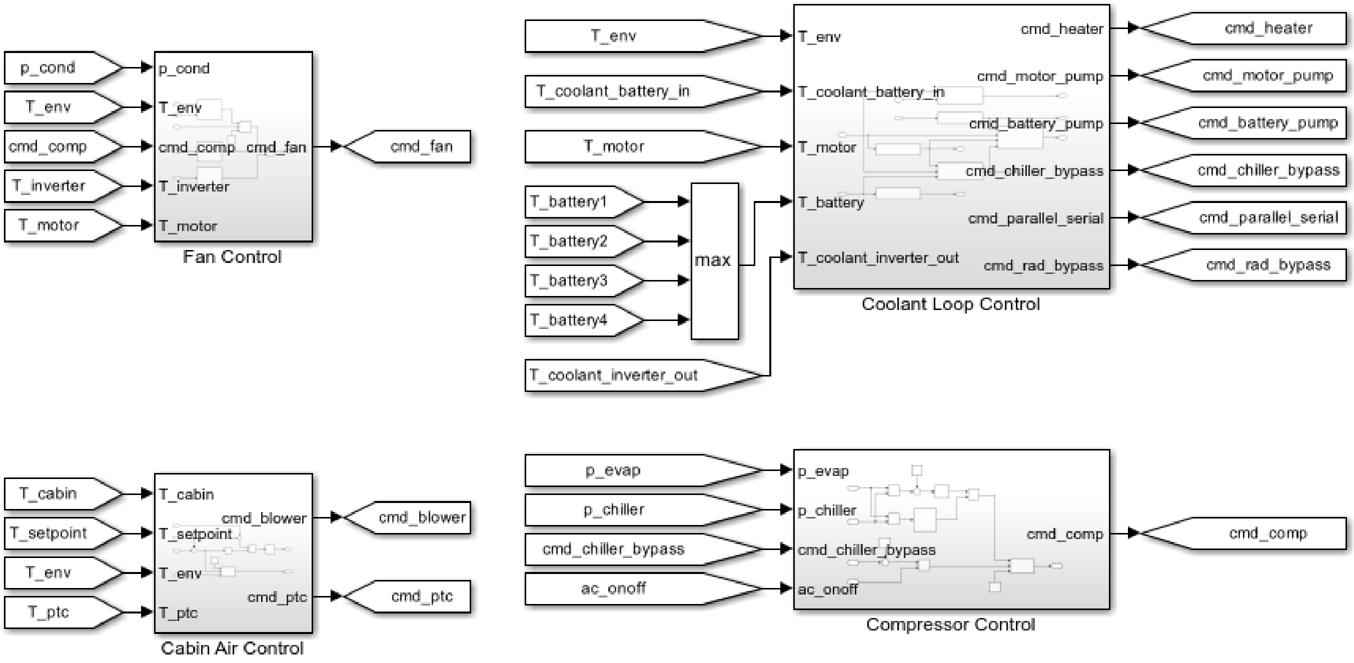
Scenario Subsystem

This subsystem sets up the environment conditions and inputs to the system for the selected scenario. The battery current demand and powertrain heat load are a function of the vehicle speed based on tabulated data.



Controls Subsystem

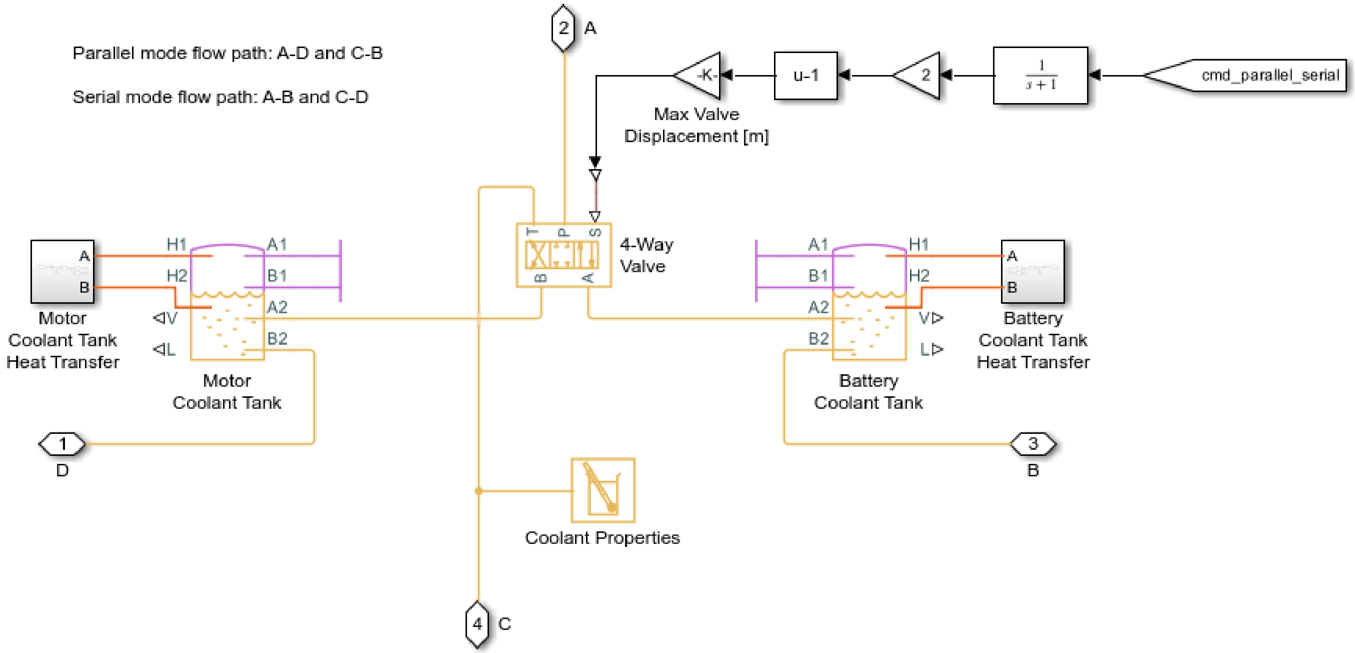
This subsystem consists of all of the controllers for the pumps, compressor, fan, blower, and valves in the thermal management system.



Parallel-Serial Mode Valve Subsystem

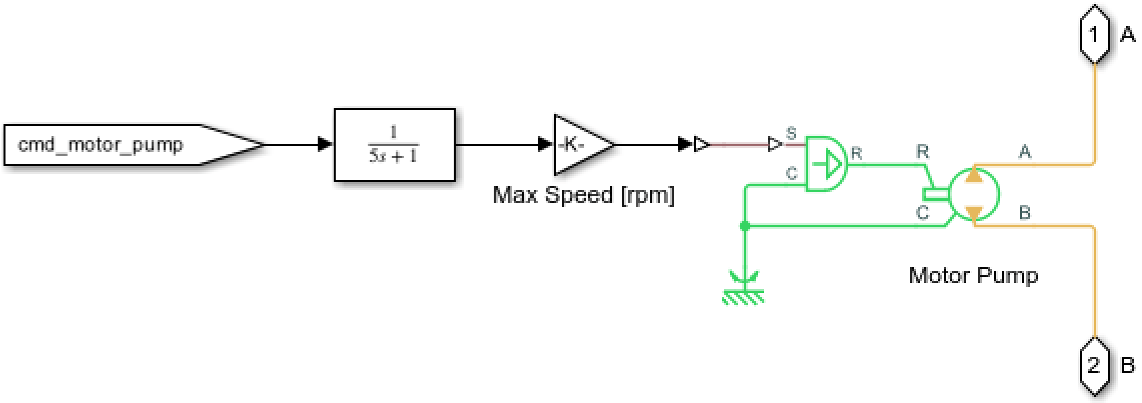
The 4-way valve in this subsystem controls whether the coolant loop operates in parallel or serial mode. When ports A and D are connected and ports C and B are connected, it is in parallel mode. The two coolant loops are separated with their own coolant tanks and pumps.

When ports A and B are connected and ports C and D are connected, it is in serial mode. The two coolant loops are merged and the two pumps are synchronized to provide the same flow rate.


Parallel mode flow path: A-D and C-B
Serial mode flow path: A-B and C-D
2 A
-K-
Max Valve Displacement [m]
u-1
2
$\frac{1}{s+1}$
cmd_parallel_serial
4-Way Valve
T P S B A
A
B
Motor Coolant Tank Heat Transfer
H1
H2
A1
B1
A2
B2
V
L
Motor Coolant Tank
1
D
A1
B1
A2
B2
H1
H2
V
L
Battery Coolant Tank
A
B
Battery Coolant Tank Heat Transfer
3
B
Coolant Properties
4 C

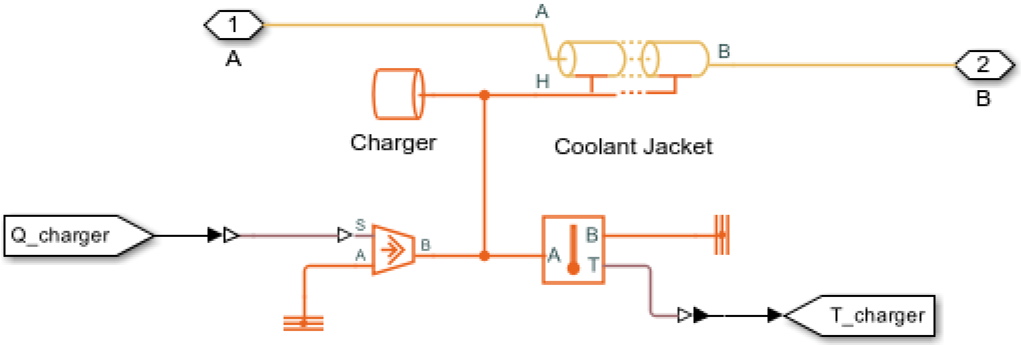
Motor Pump Subsystem

This pump drives the coolant loop that cools the charger, motor, and inverter.



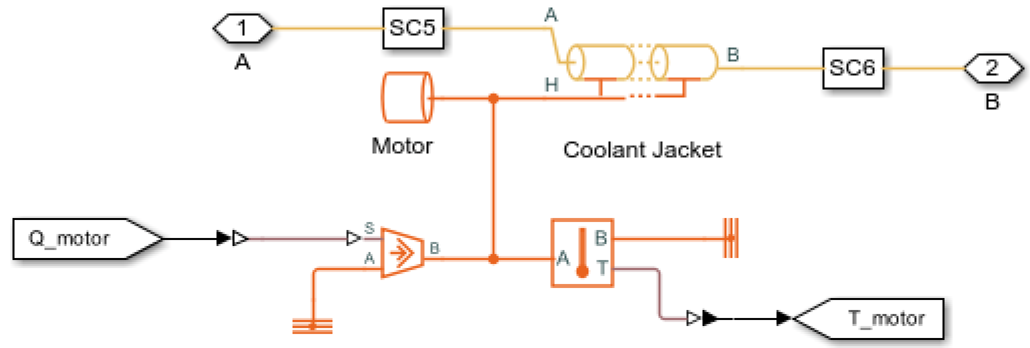
Charger Subsystem

This subsystem models a coolant jacket around the charger, which is represented by a heat flow rate source and a thermal mass.



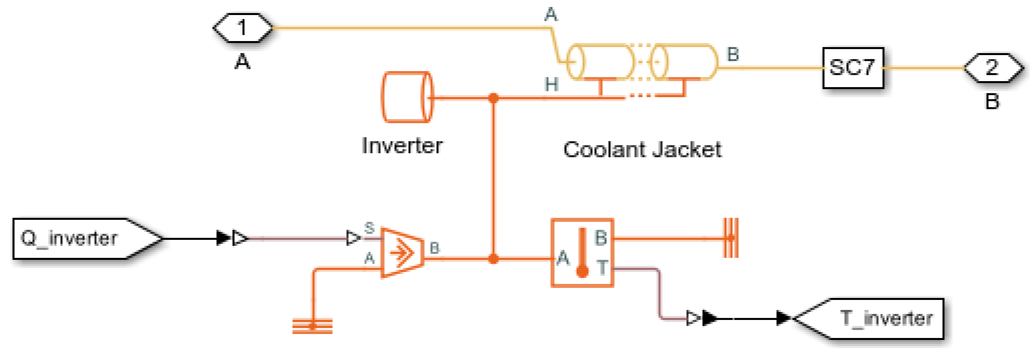
Motor Subsystem

This subsystem models a coolant jacket around the motor, which is represented by a heat flow rate source and a thermal mass.



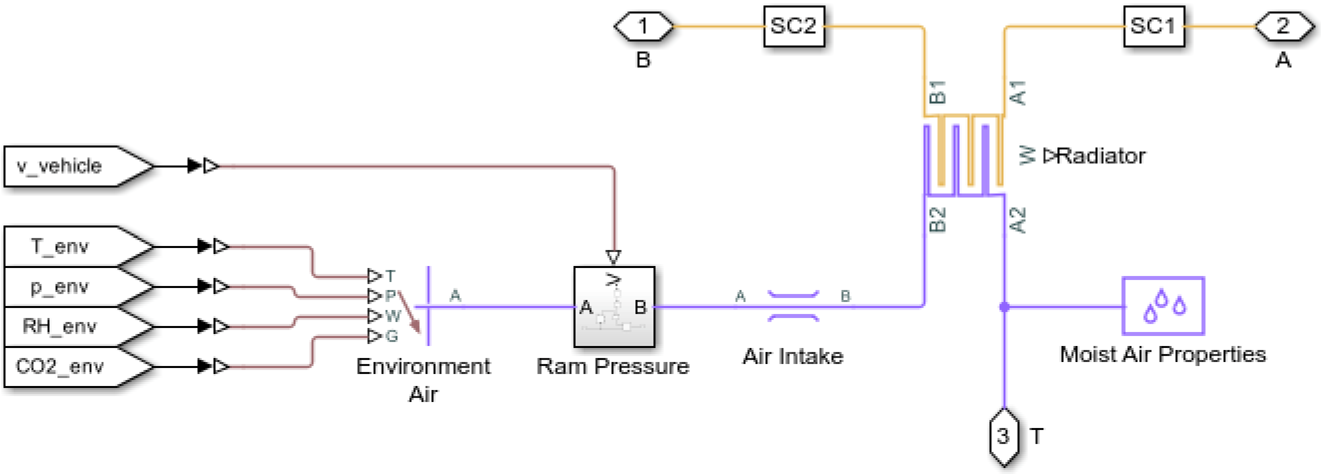
Inverter Subsystem

This subsystem models a coolant jacket around the inverter, which is represented by a heat flow rate source and a thermal mass.



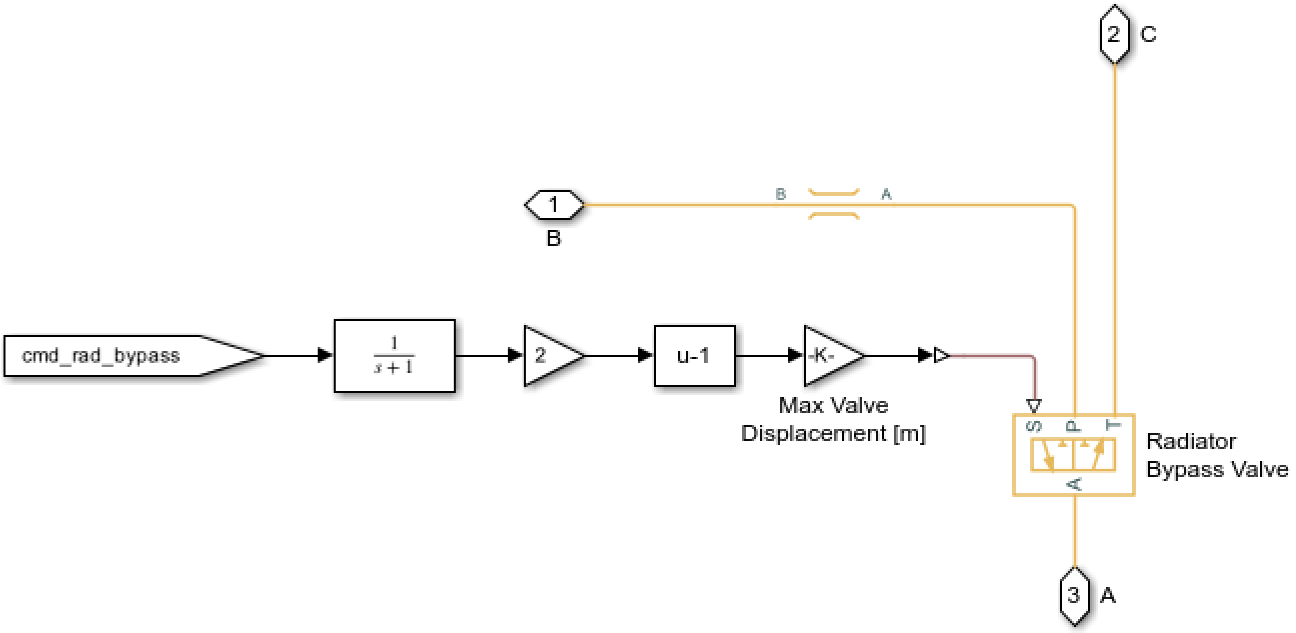
Radiator Subsystem

The radiator is a rectangular tube-and-fin type heat exchanger that dissipates coolant heat to the air. The air flow is driven by the vehicle speed and the fan located behind the condenser.



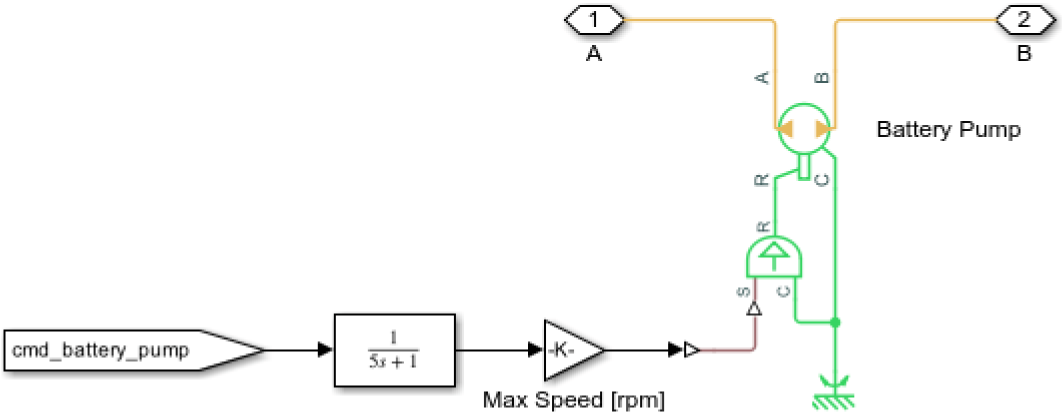
Radiator Bypass Valve Subsystem

In cold weather, the radiator is bypassed so that heat from the powertrain can be used to warm up the batteries. This is controlled by the 3-way valve that either sends coolant to the radiator or bypasses the radiator.



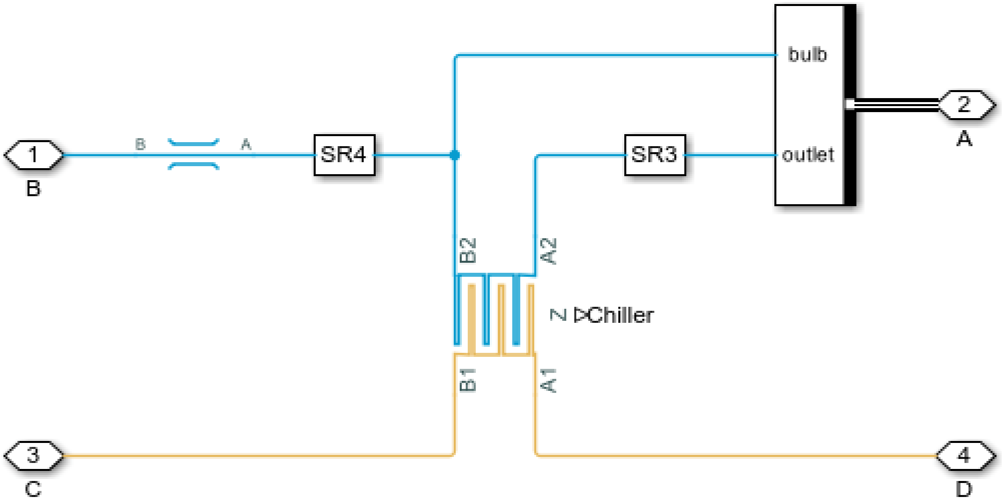
Battery Pump Subsystem

This pump drives the coolant loop that cools the batteries and the DC-DC converter.



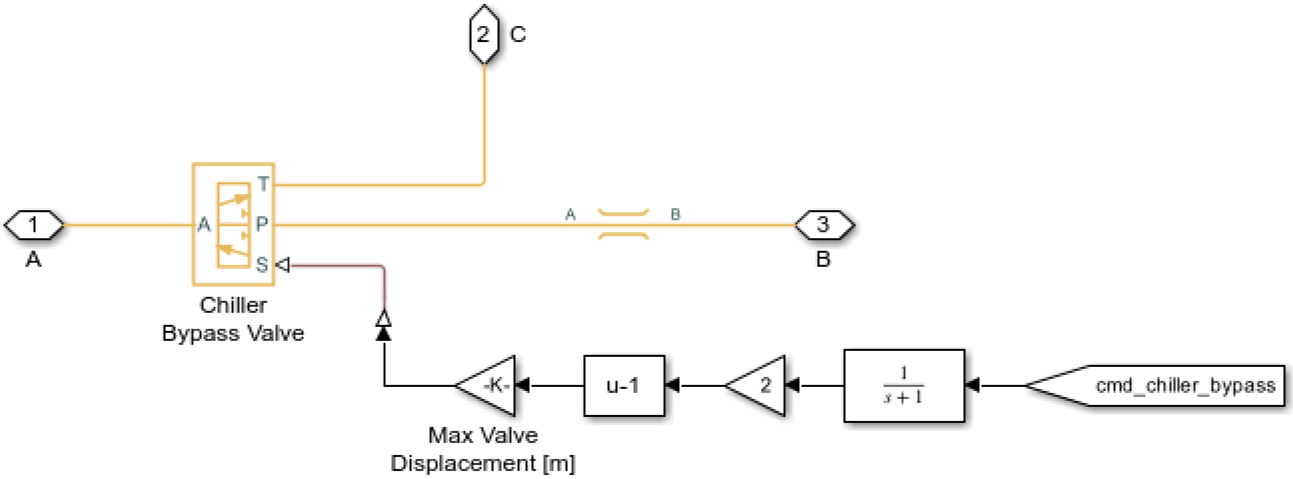
Chiller Subsystem

The chiller is assumed to be a shell-and-tube type heat exchanger that lets the refrigerant absorb heat from the coolant.



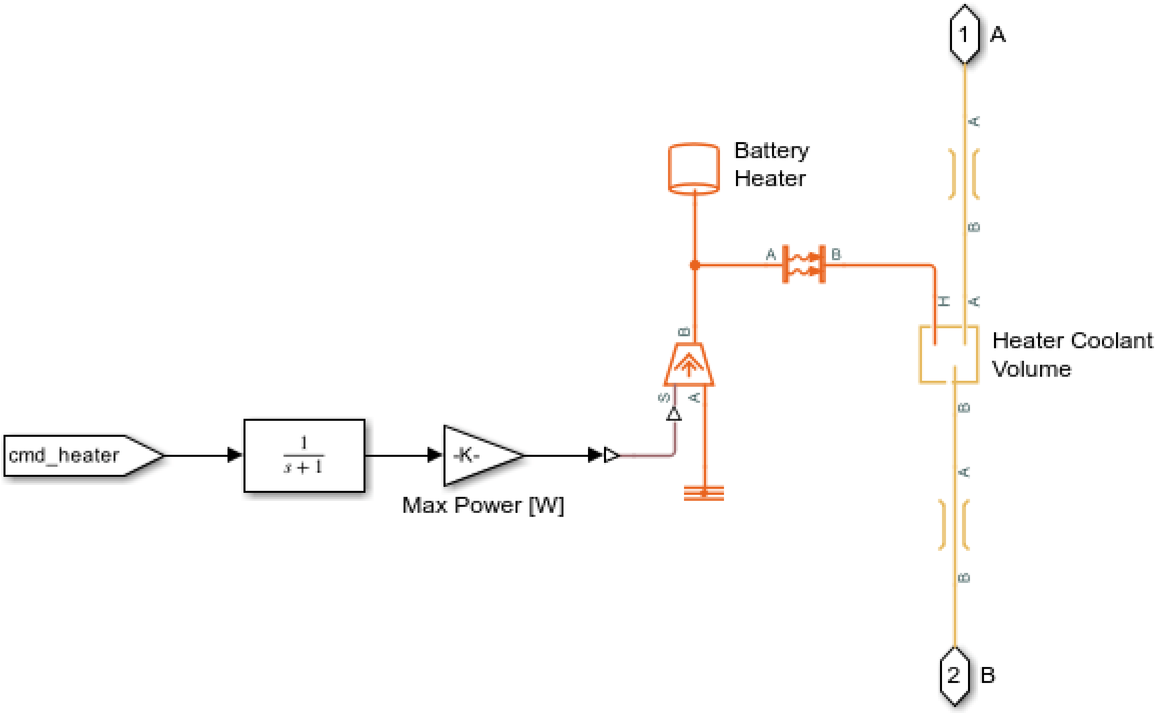
Chiller Bypass Valve Subsystem

The chiller operates in an on-off manner depending on the battery temperature. This is controlled by the 3-way valve that either sends coolant to the chiller or bypasses the chiller.



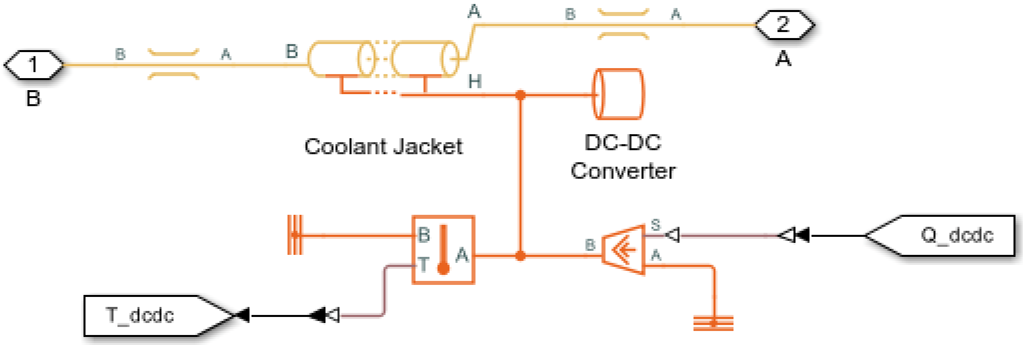
Heater Subsystem

The battery heater is modeled as a heat flow rate source and a thermal mass. It is turned on in cold weather to bring the battery temperature above 5 degC.



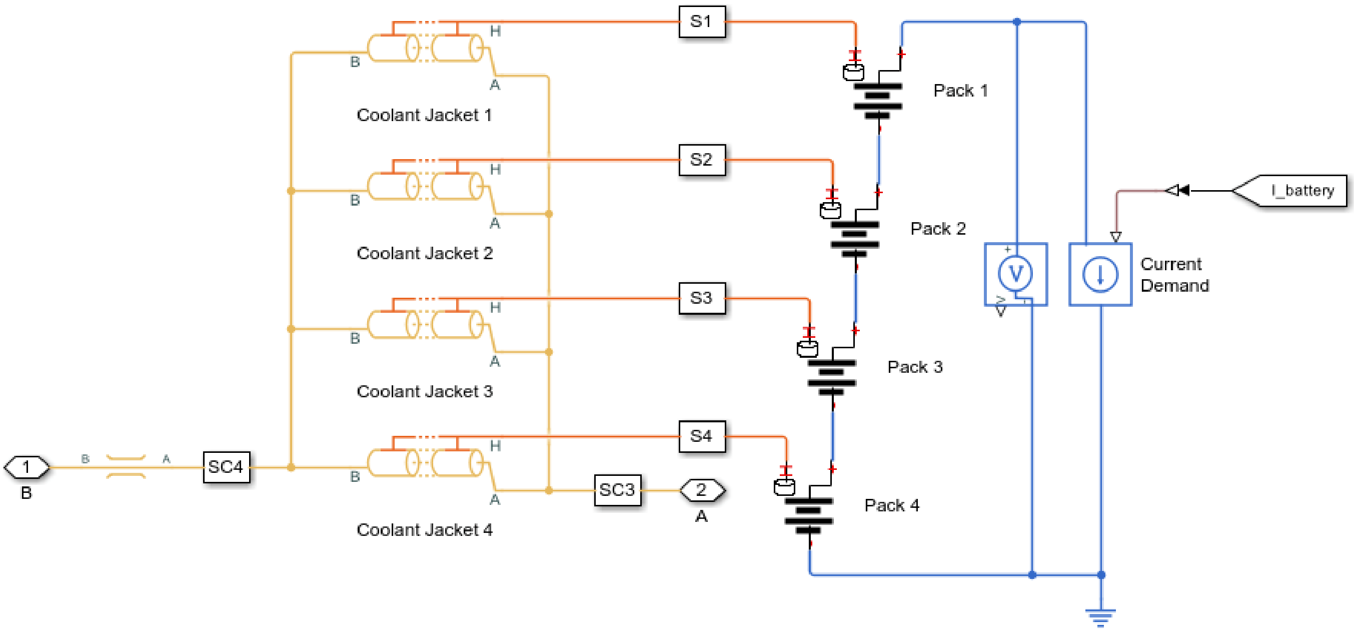
DCDC Subsystem

This subsystem models a coolant jacket around the DC-DC converter, which is represented by a heat flow rate source and a thermal mass.



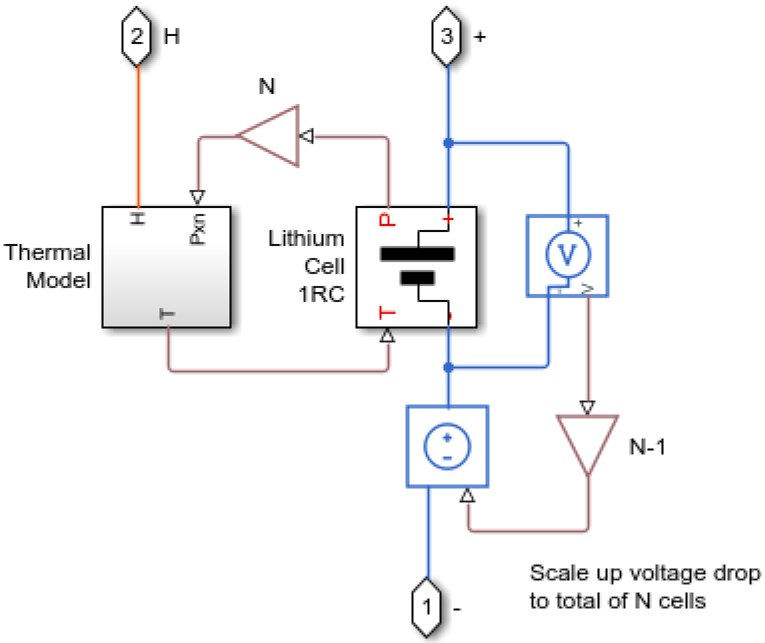
Battery Subsystem

The batteries are modeled as four separate packs surrounded by a coolant jacket. The battery packs generate voltage and heat based on the current demand. The coolant is assumed to flow in narrow channels around the battery packs.



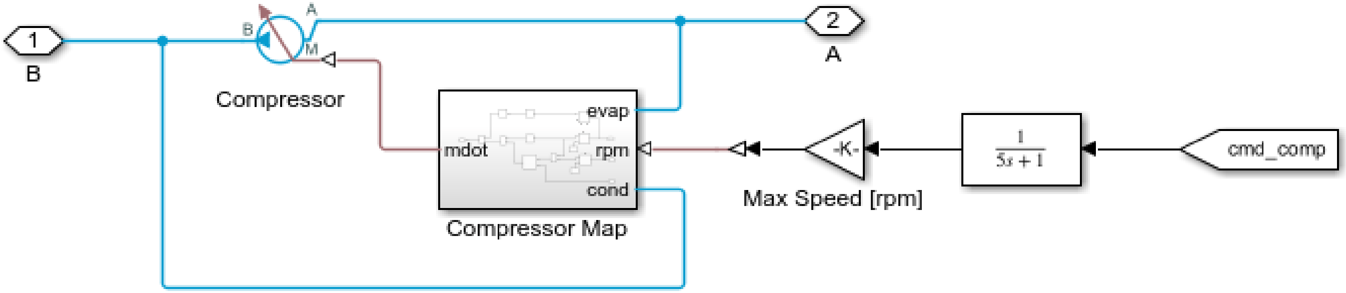
Pack 1 Subsystem

Each battery pack is modeled as a stack of lithium-ion cells coupled with a thermal model. Heat is generated based on the power losses in the cells.



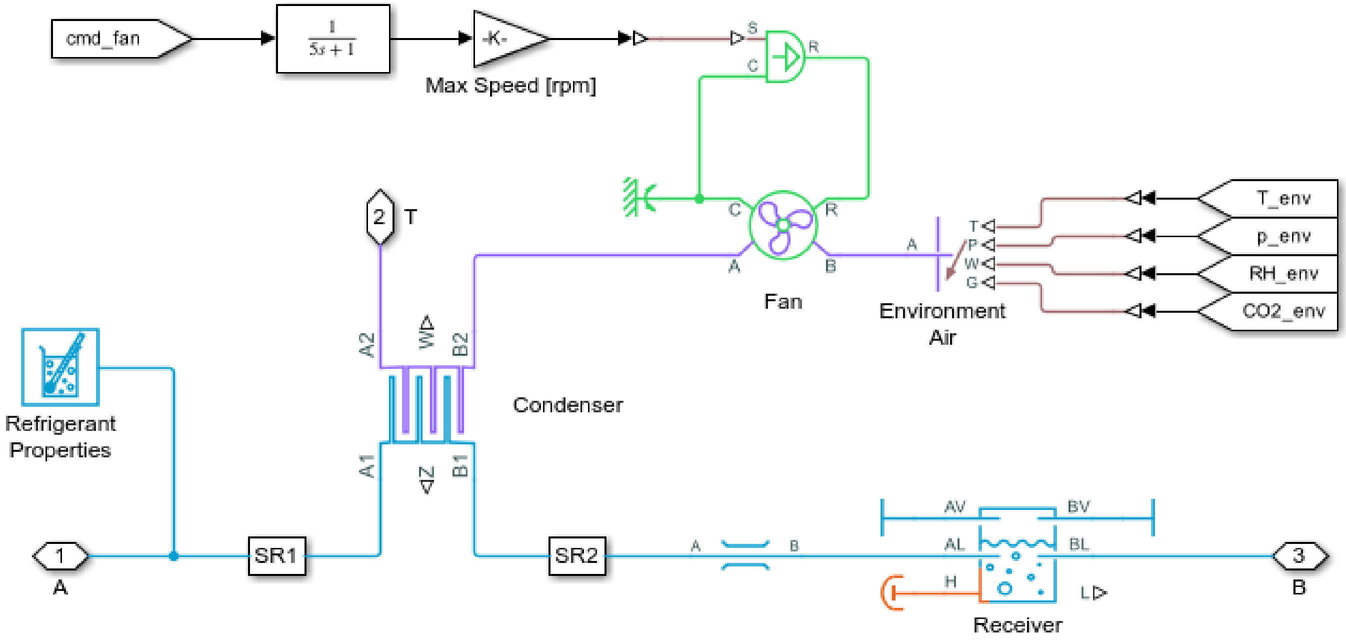
Compressor Subsystem

The compressor drives the flow in the refrigerant loop. It is controlled to maintain a pressure of 0.3 MPa in the chiller and the evaporator, which corresponds to a saturation temperature of around 1 degC.



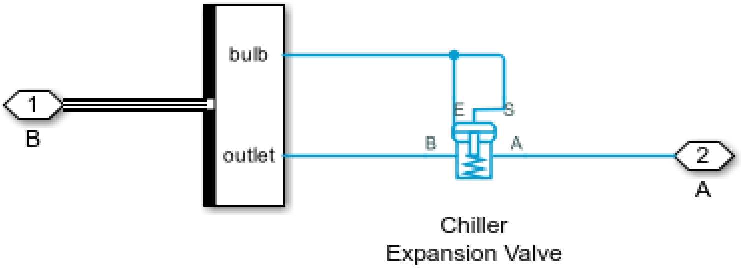
Condenser Subsystem

The condenser is a rectangular tube-and-fin type heat exchanger that dissipates refrigerant heat to the air. The air flow is driven by the vehicle speed and the fan. The liquid receiver provides storage for the refrigerant and permits only subcooled liquid to flow into the expansion valves.



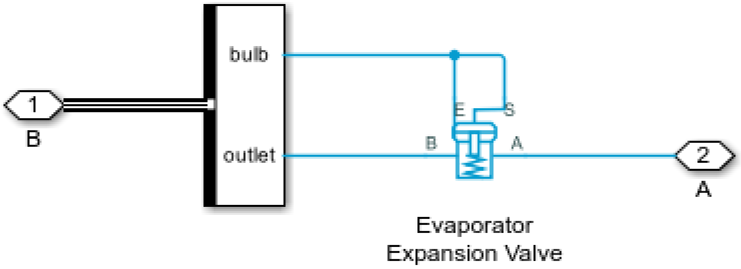
Chiller Expansion Valve Subsystem

This expansion valve meters refrigerant flow to the chiller to maintain a nominal superheat.



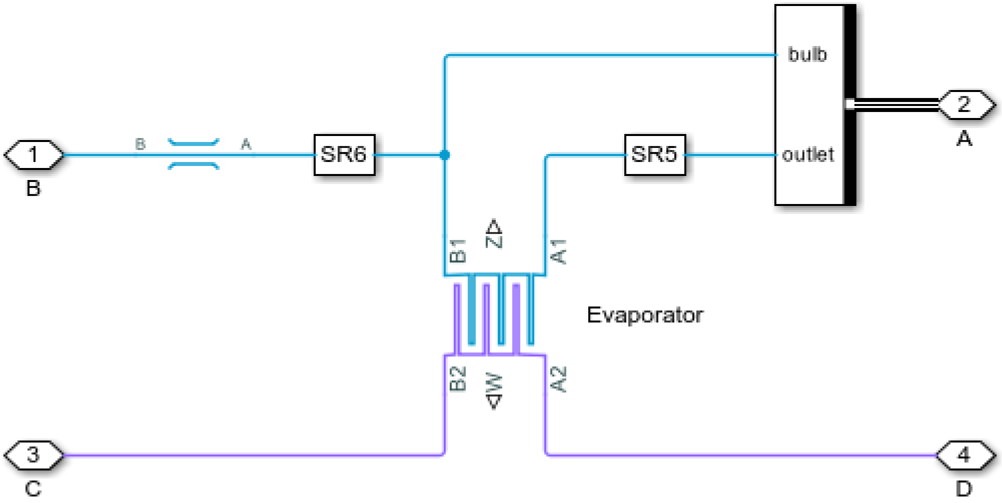
Evaporator Expansion Valve Subsystem

This expansion valve meters refrigerant flow to the evaporator to maintain a nominal superheat.



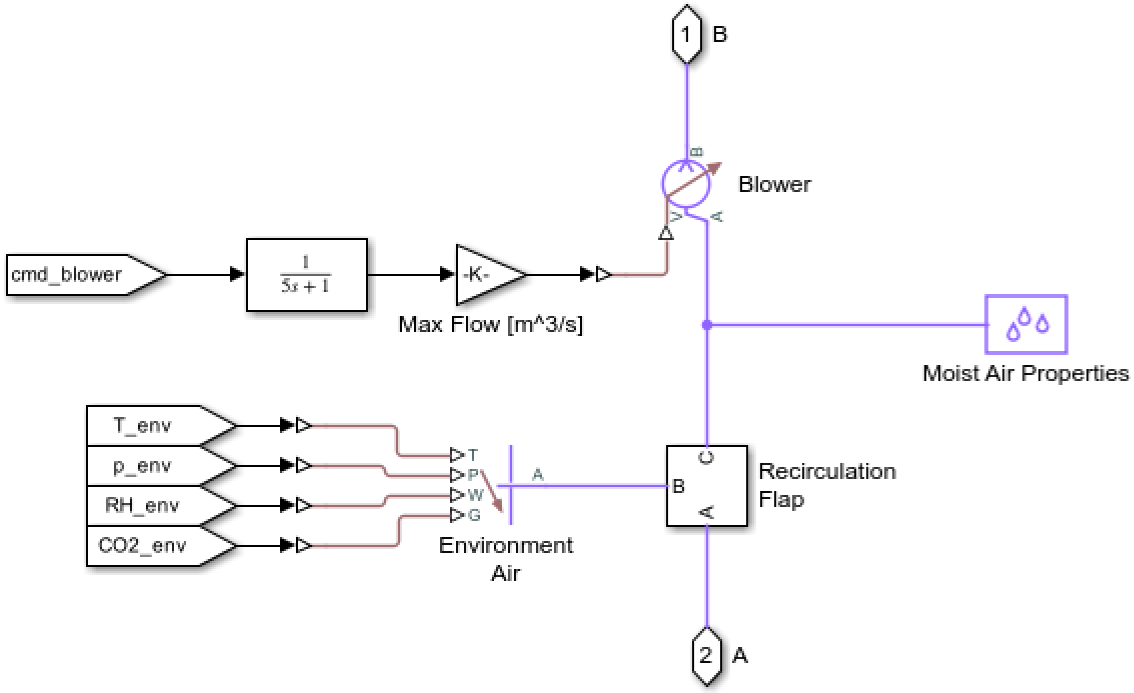
Evaporator Subsystem

The evaporator is a rectangular tube-and-fin type heat exchanger that lets the refrigerant absorb heat from the air. It also dehumidifies the air when the air is humid.



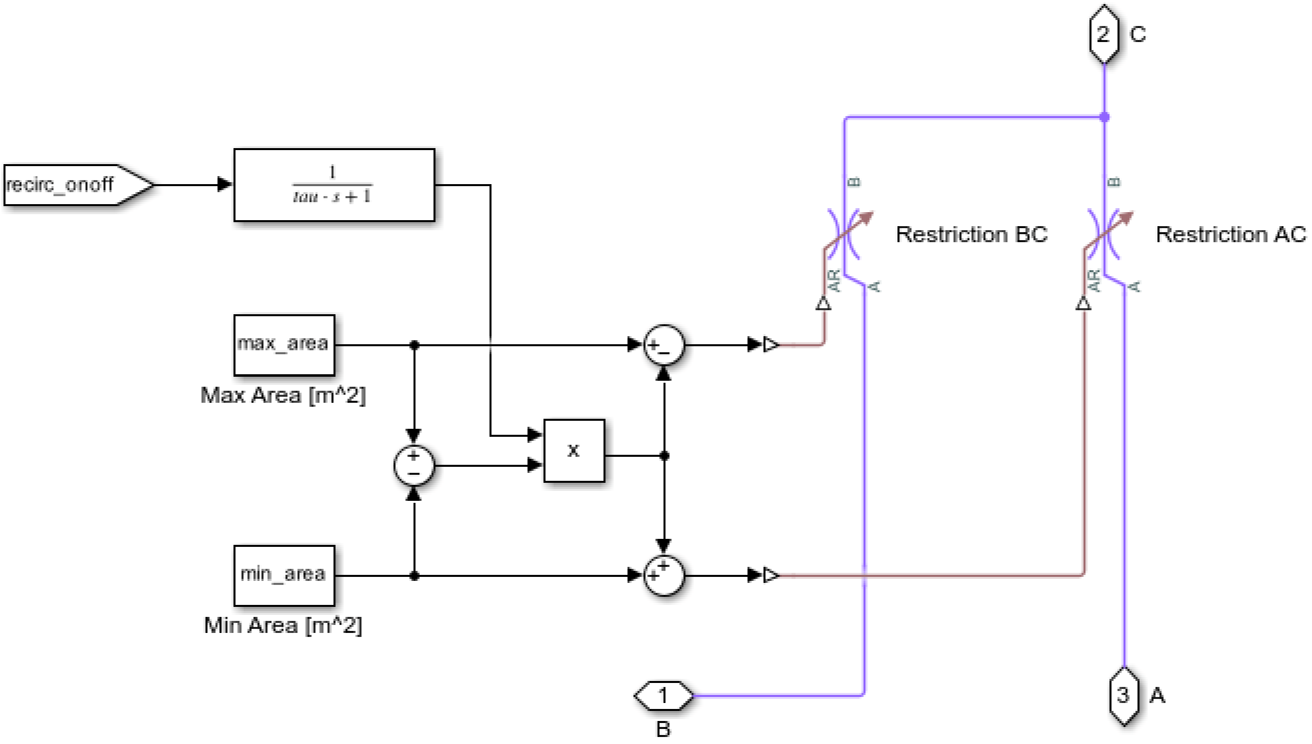
Blower Subsystem

The blower drives the air flow in the HVAC loop. It is controlled to maintain the cabin temperature setpoint. The source of air can come from the environment or from recirculated cabin air.



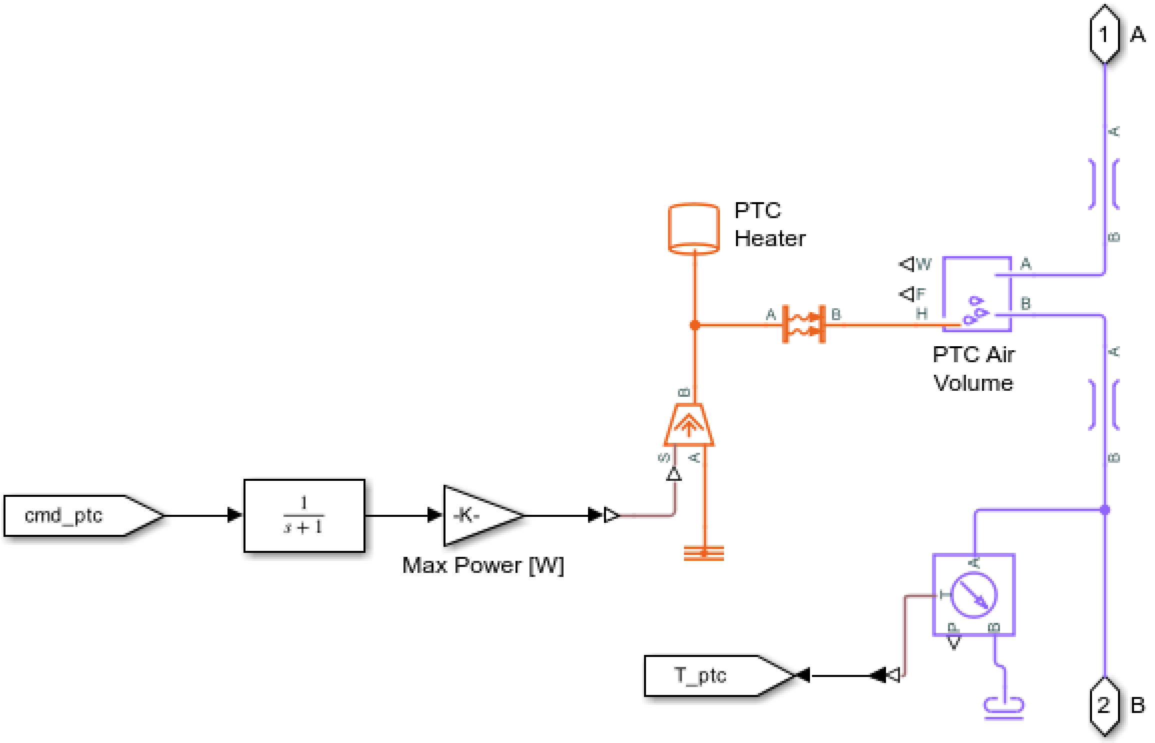
Recirculation Flap Subsystem

The recirculation flap is modeled as two restrictions operating in the opposite manner to let either environment air or cabin air to the blower.



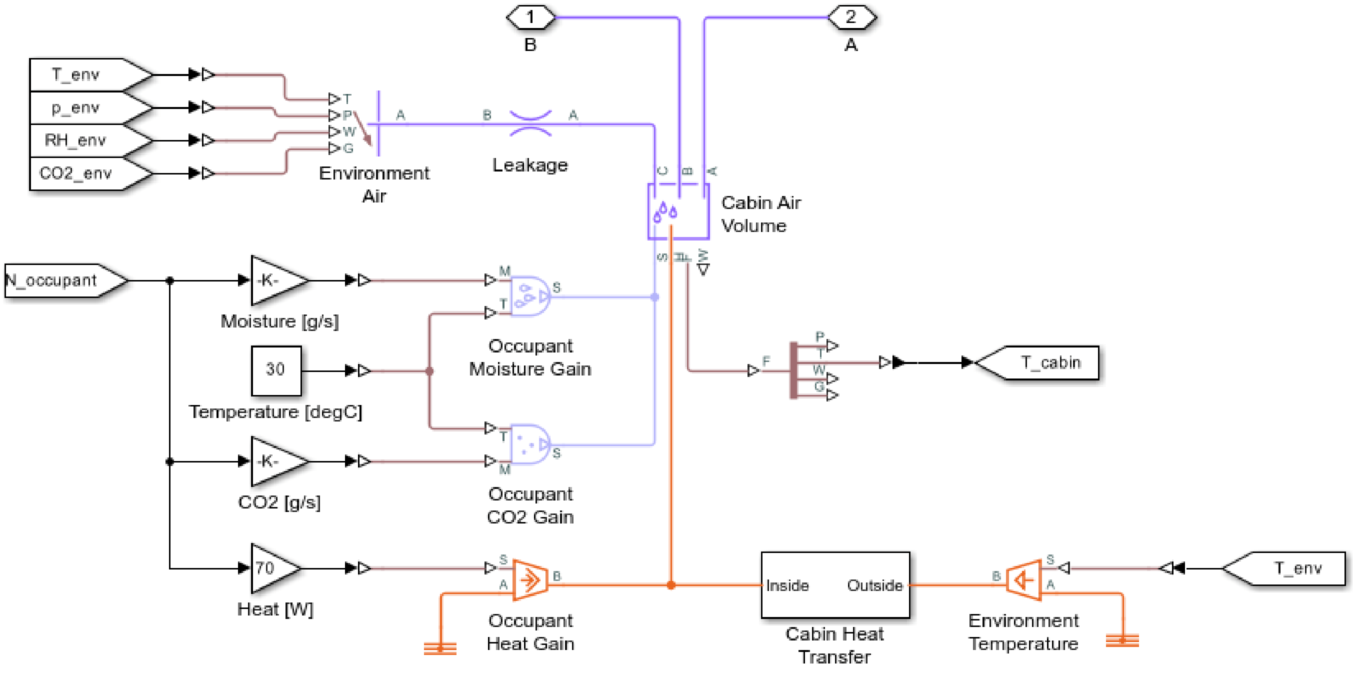
PTC Subsystem

The PTC heater is modeled as a heat flow rate source and a thermal mass. It is turned on in cold weather to provide heating to the vehicle cabin.



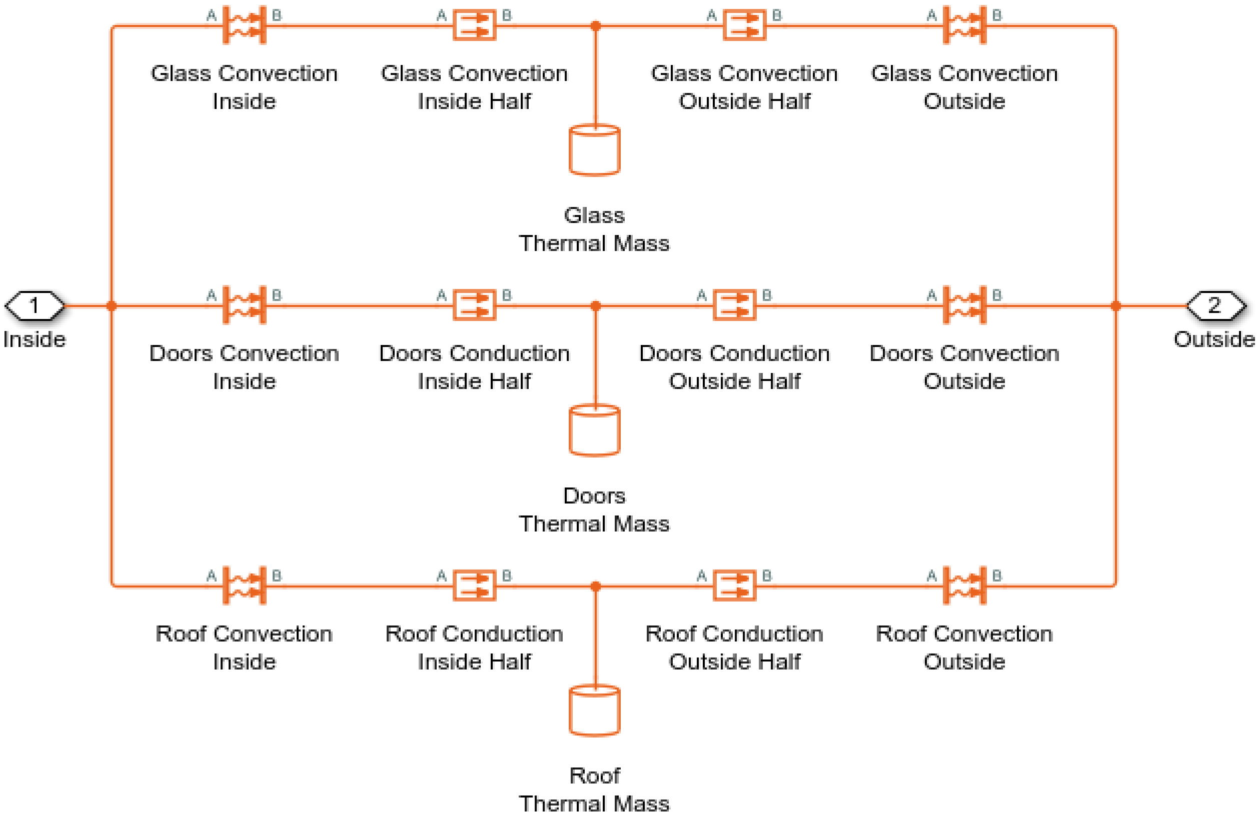
Cabin Subsystem

The vehicle cabin is modeled as a large volume of moist air. Each occupant in the vehicle is a source of heat, moisture, and CO2.



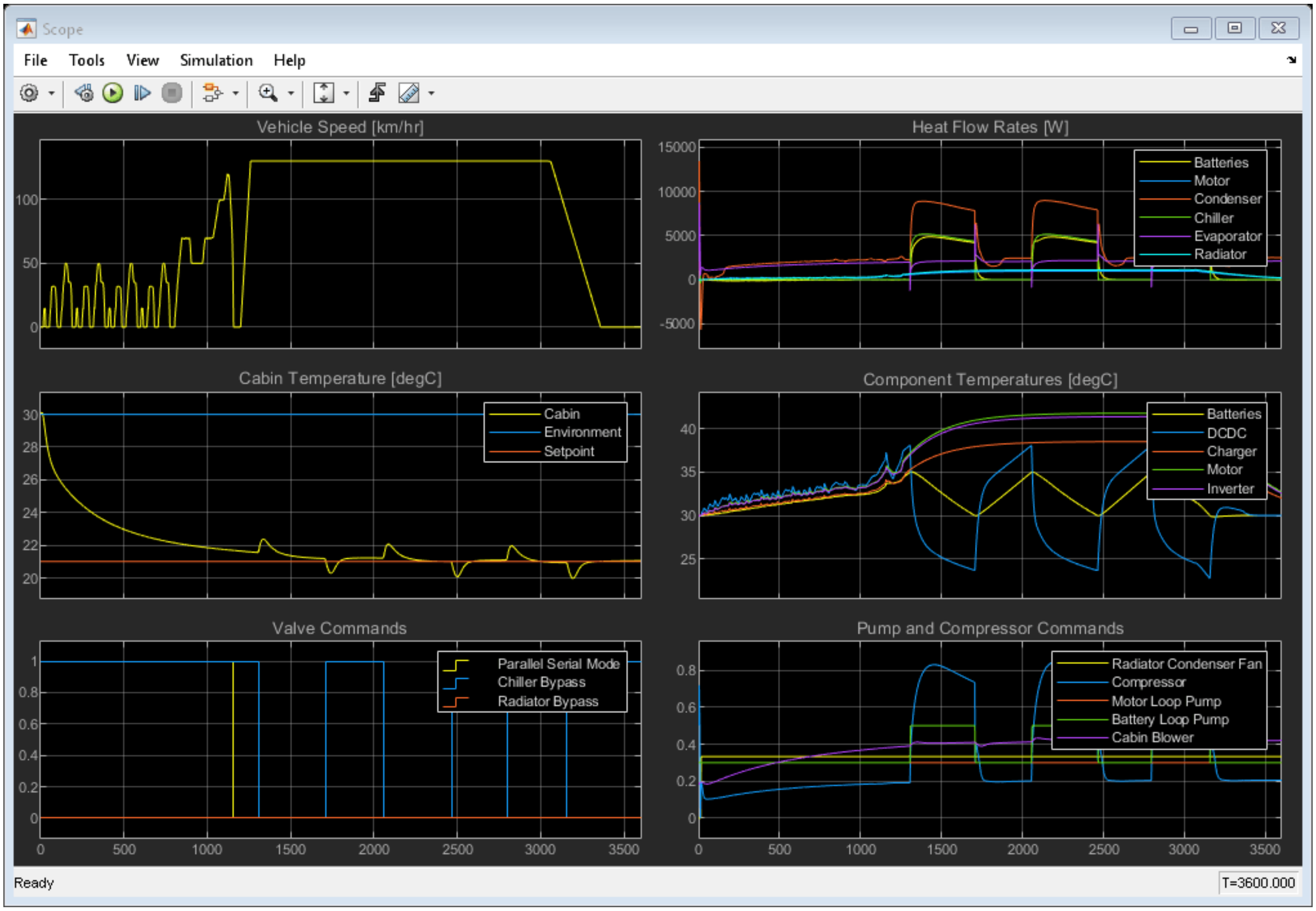
Cabin Heat Transfer Subsystem

This subsystem models the thermal resistances between the cabin interior and the external environment.



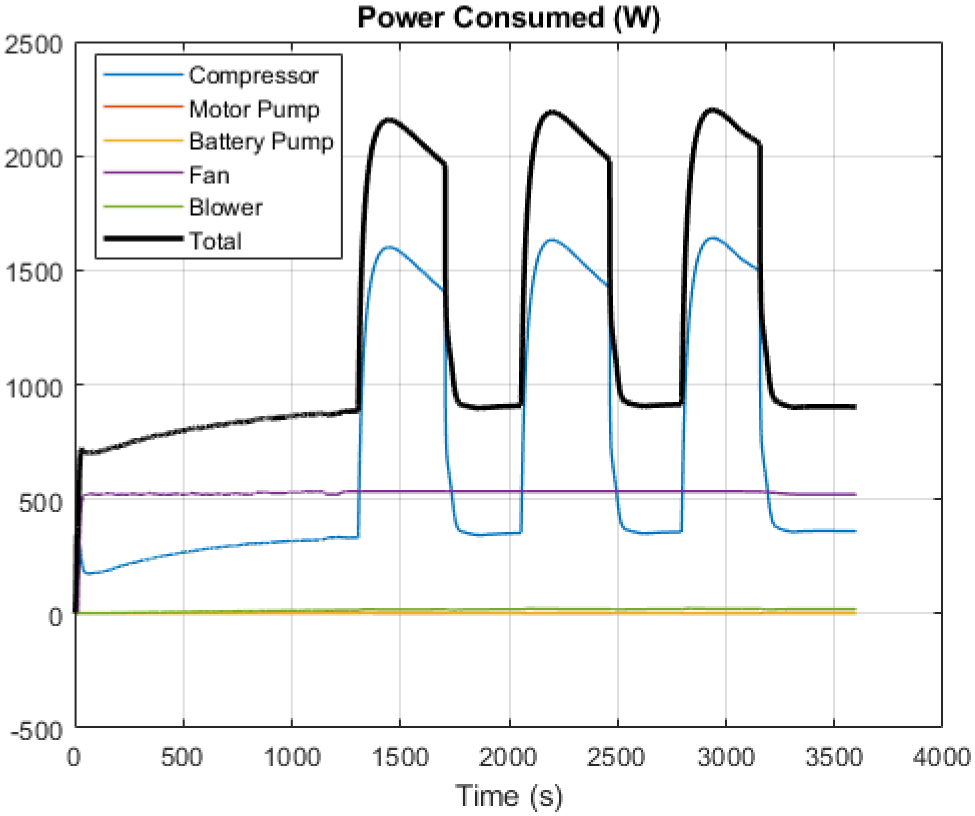
Simulation Results from Scopes

The following scope shows the vehicle speed, heat dissipation, cabin temperature, component temperatures, and control commands for the drive cycle scenario. At the beginning, the coolant loop is in serial mode. After about 1100 s, it switches to parallel mode and the chiller is used to keep the batteries below 35 degC.



Simulation Results from Simscape Logging

This plot shows the power consumed by the thermal management system to cool the vehicle components and cabin. The largest power consumption occurs in the refrigerant compressor when the chiller is used to cool the batteries.


Power Consumed (W)
Compressor
Motor Pump
Battery Pump
Fan
Blower
Total
2500
2000
1500
1000
500
0
-500
0
500
1000
1500
2000
2500
3000
3500
4000
Time (s)

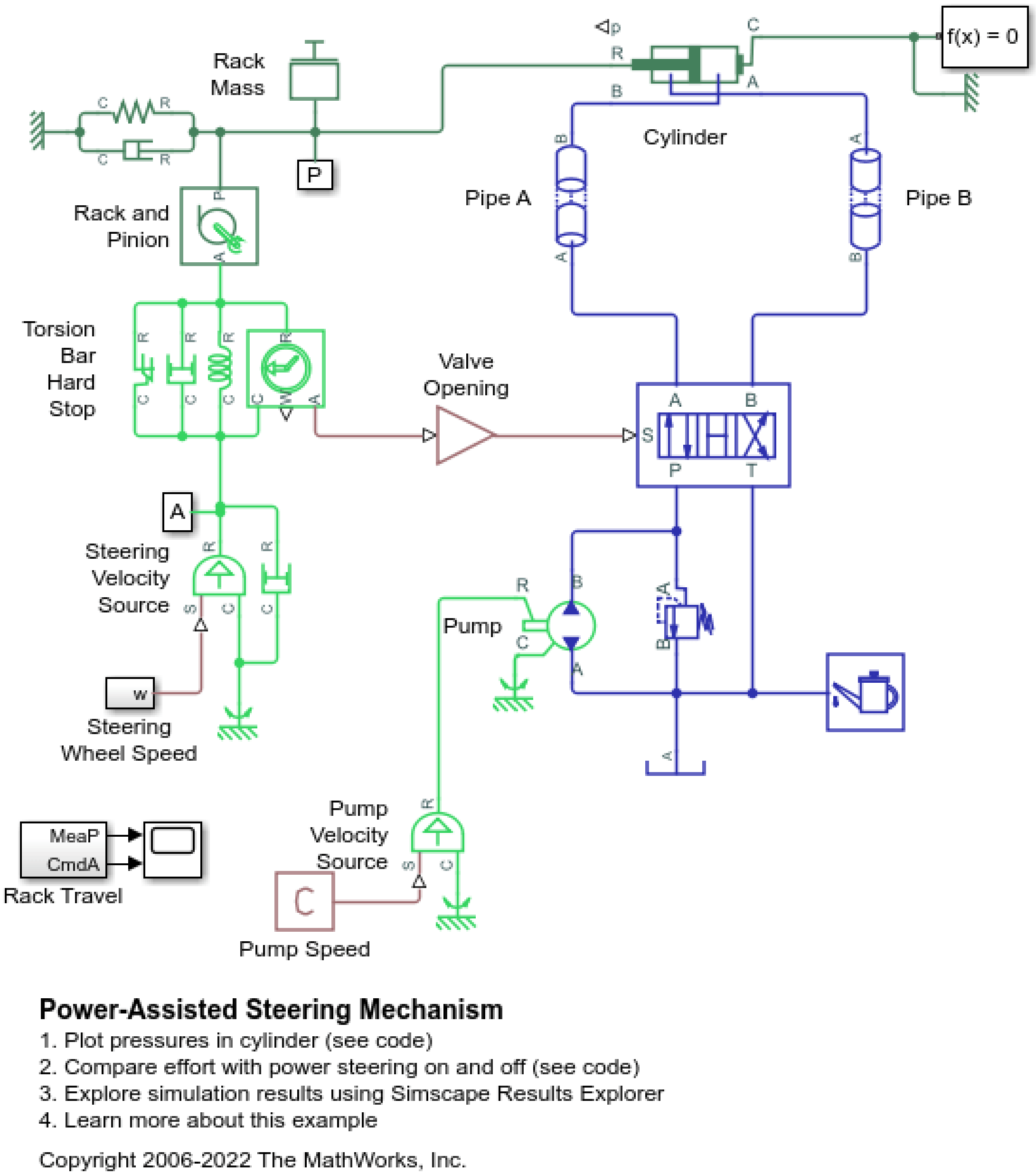
Power-Assisted Steering Mechanism

This example shows a simplified version of a power-assisted steering mechanism. The hydraulic actuation system includes a double-acting hydraulic cylinder, 4-way valve, fixed-displacement pump, and a pressure-relief valve. The steering rack acts against a load modeled by a spring and damper.

The valve opens based on the amount of twist of a torsion bar installed between the steering wheel and the pinion of a rack-and-pinion mechanism. The rotation of the steering wheel causes the torsion bar to twist relative to the pinion. The deformation of the bar is transformed into opening of the 4-way valve, which connects the chambers of the cylinder to pressure or return lines depending upon the direction of rotation. The PS Gain block Valve Opening converts torsion bar twist to the amount of opening for the 4-way valve. Setting the Gain parameter of this block to 0 disables the power steering.

If the torsion bar deformation exceeds 9 degrees the wheel is connected directly to the pinion through the hard stops installed in parallel with the torsion bar. The cylinder moves the steering rack which twists the torsion rod in the opposite direction until the valve is in the neutral position.

Model



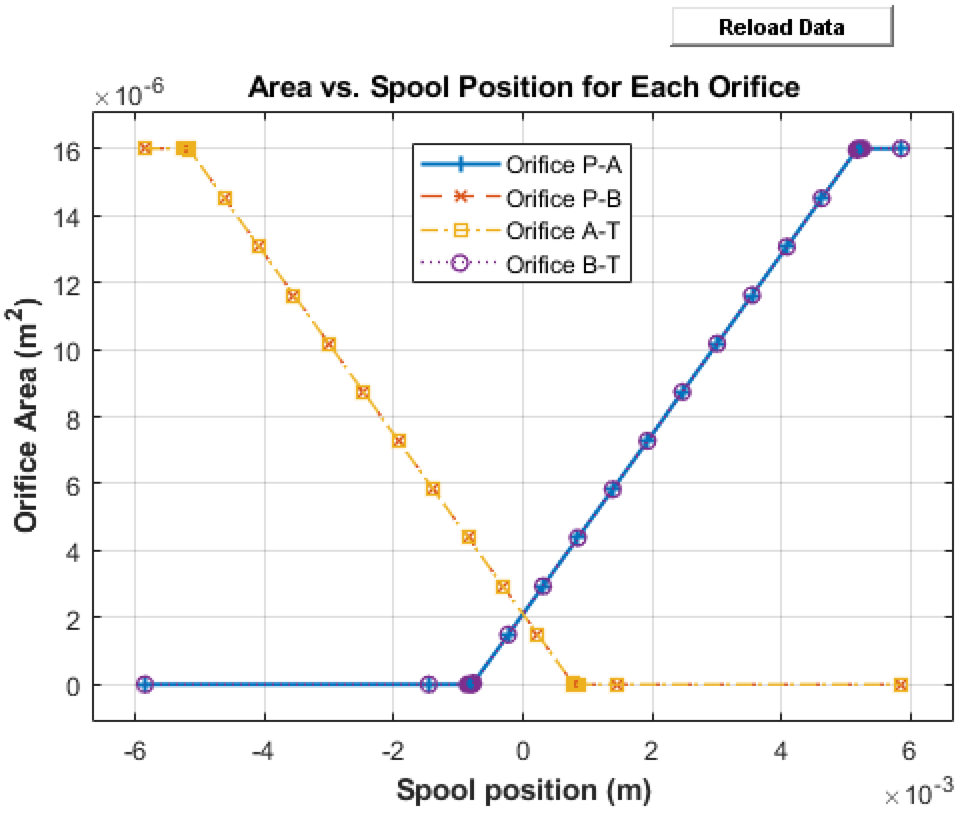
Power-Assisted Steering Mechanism

1. Plot pressures in cylinder (see code)
2. Compare effort with power steering on and off (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

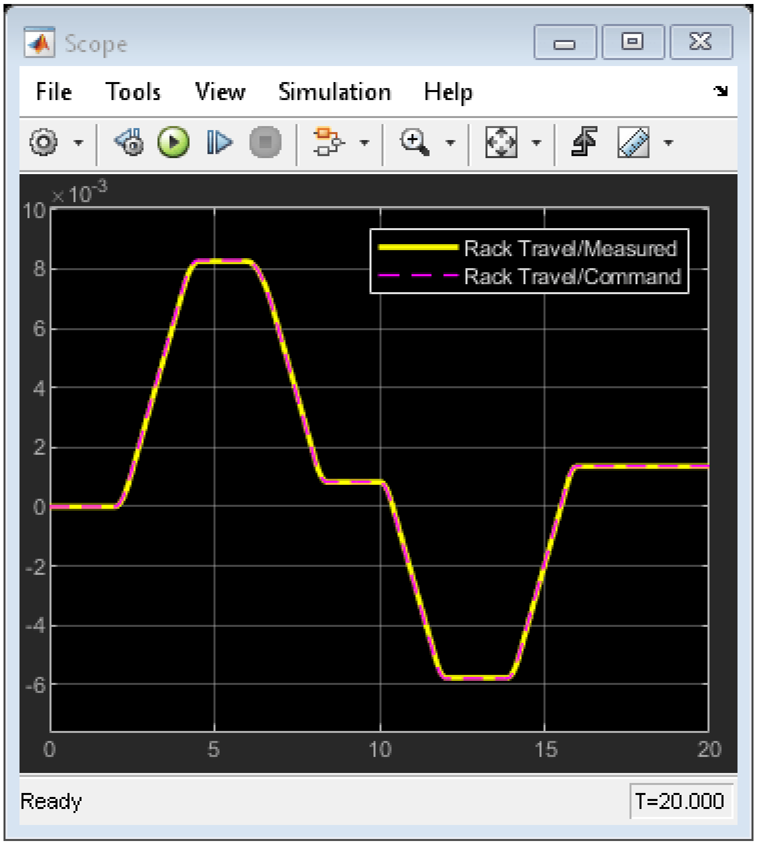
Copyright 2006-2022 The MathWorks, Inc.

4-Way Valve Opening Characteristics

This plot is generated by right-clicking on the 4-Way Directional Valve (IL) and selecting Fluids > Plot Valve Characteristics.

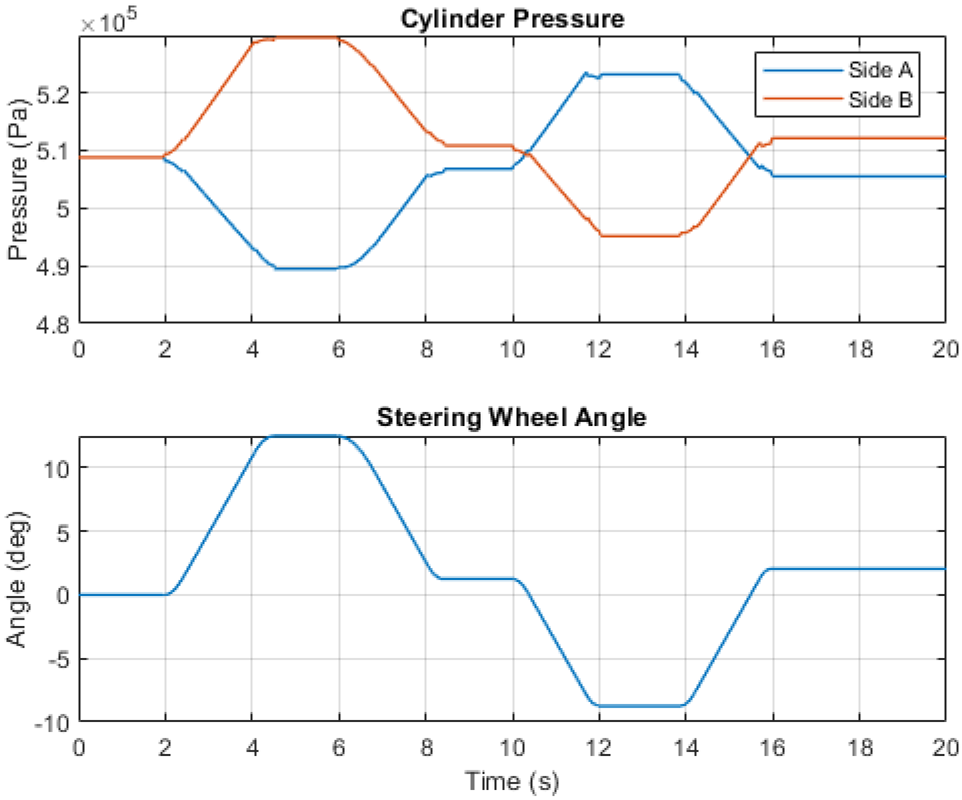

Reload Data
Area vs. Spool Position for Each Orifice
× 10⁻⁶
16
14
12
10
8
6
4
2
0
Orifice Area (m²)
Orifice P-A
Orifice P-B
Orifice A-T
Orifice B-T
-6
-4
-2
0
2
4
6
Spool position (m)
× 10⁻³

Simulation Results from Scopes

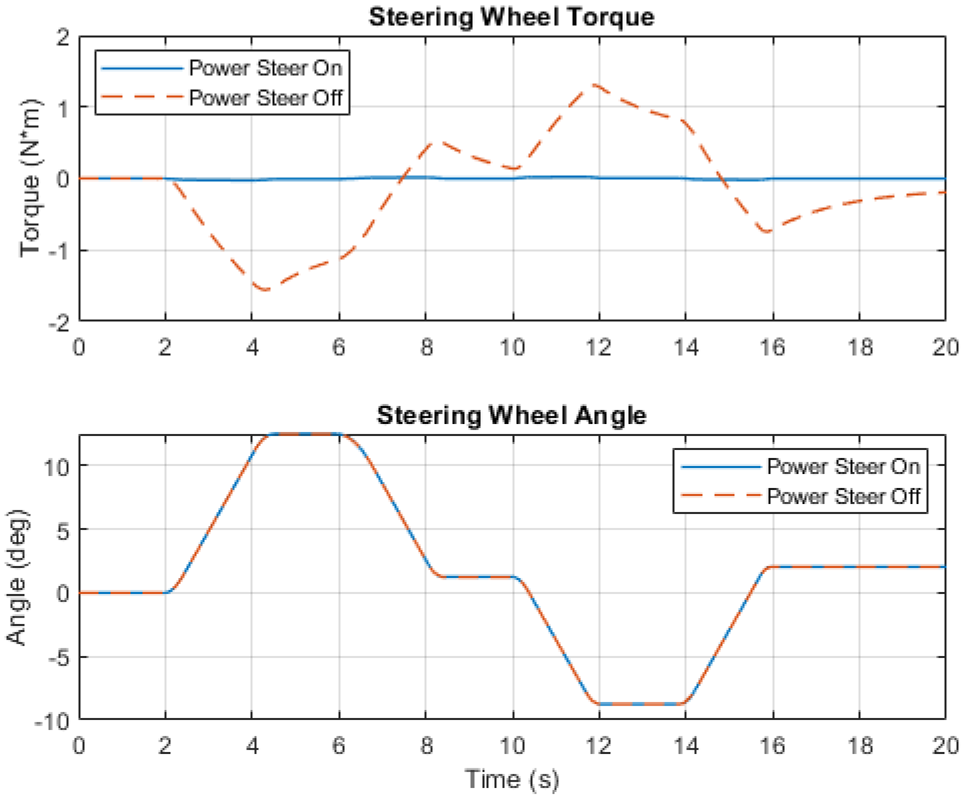


Simulation Results from Simscape Logging

The plots below show the amount of assistance the power steering system provides to the driver as the wheel is turned. The greater the difference in the cylinder chamber pressures, the more assistance that is provided.



The plots below compare the amount of effort required when the power steering is turned on and turned off. For the same steering sequence, the amount of torque required when the power steering is turned off is much higher. The power steering is turned off by setting the gain in Valve Opening to 0.

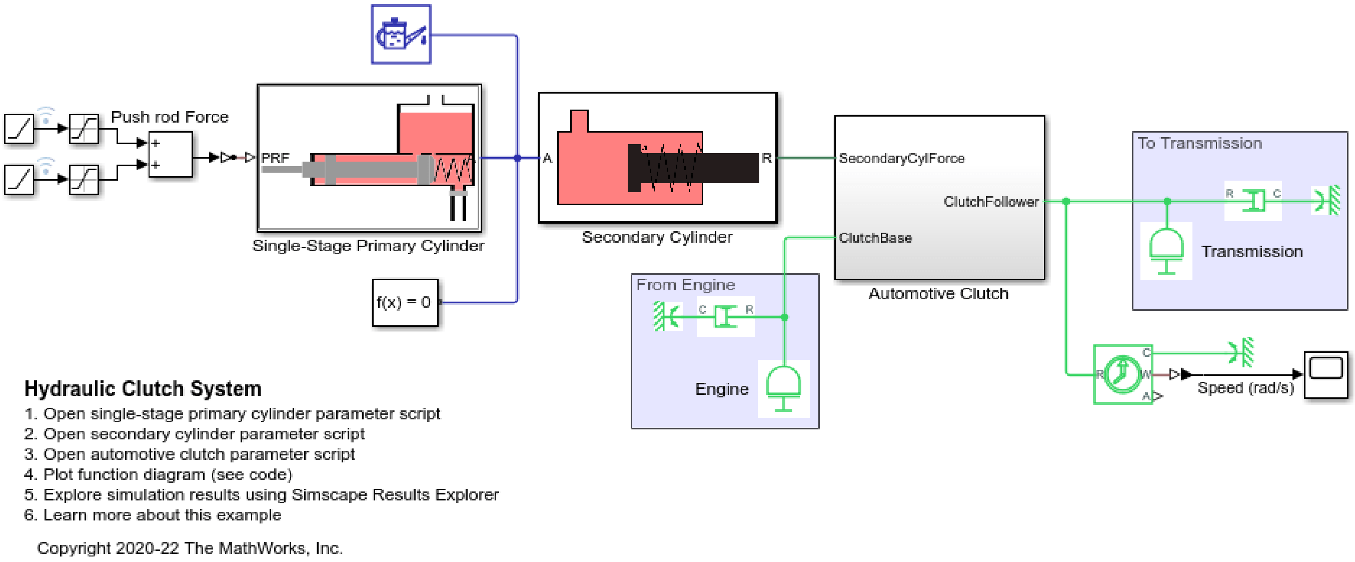

Steering Wheel Torque
Torque (N*m)
2
1
0
-1
-2
0
2
4
6
8
10
12
14
16
18
20
Power Steer On
Power Steer Off
Steering Wheel Angle
Angle (deg)
10
5
0
-5
-10
Time (s)
Power Steer On
Power Steer Off

Hydraulic Clutch System

This example shows how to model, parameterize, and test a hydraulic clutch system. The model is used to generate the plot of the engine and the transmission system speeds during a declutching and clutch re-engaging scenario.

Model

The following figure shows the model of a hydraulic clutch system in a test harness. The output of the model is a plot of speeds versus time of the engine and the transmission systems during the declutching and clutch re-engaging scenario.

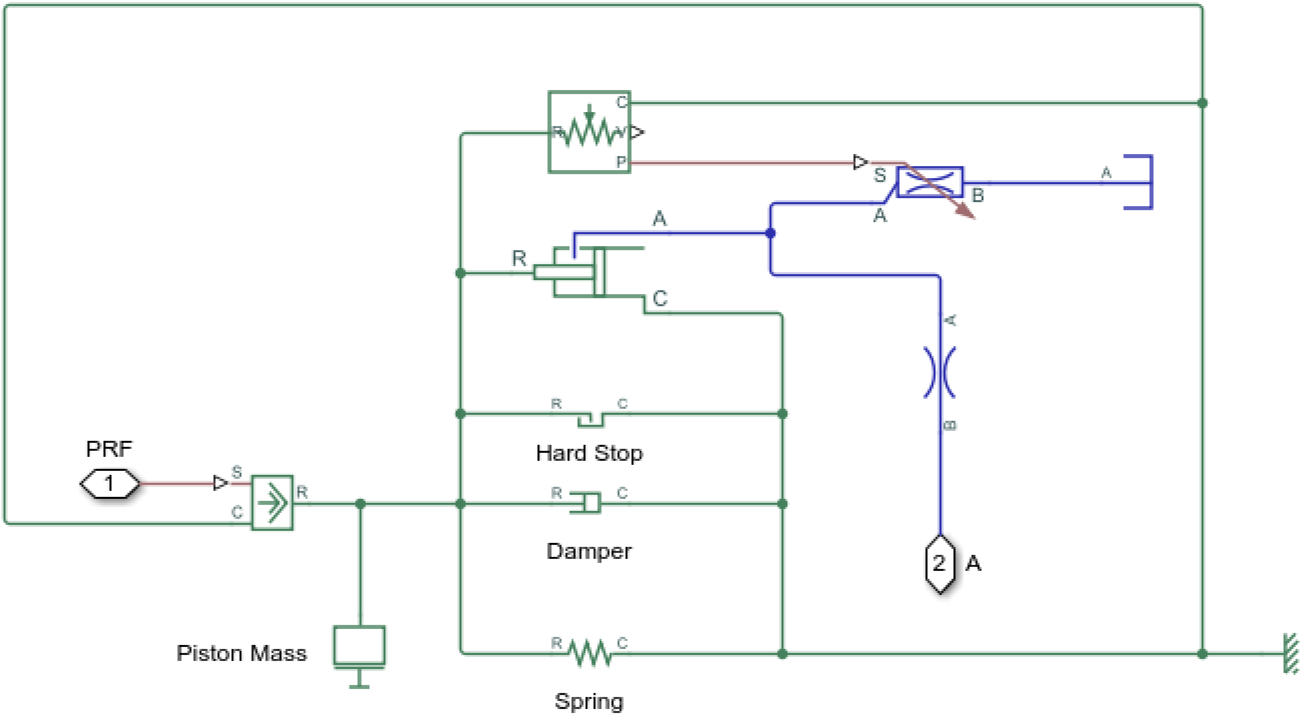


Hydraulic Clutch System
1. Open single-stage primary cylinder parameter script
2. Open secondary cylinder parameter script
3. Open automotive clutch parameter script
4. Plot function diagram (see code)
5. Explore simulation results using Simscape Results Explorer
6. Learn more about this example

Copyright 2020-22 The MathWorks, Inc.

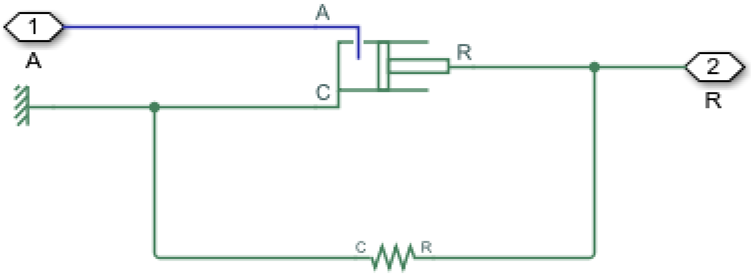
Single-Stage Primary Cylinder Subsystem

The following figure shows the modeling part of the single-stage primary cylinder.



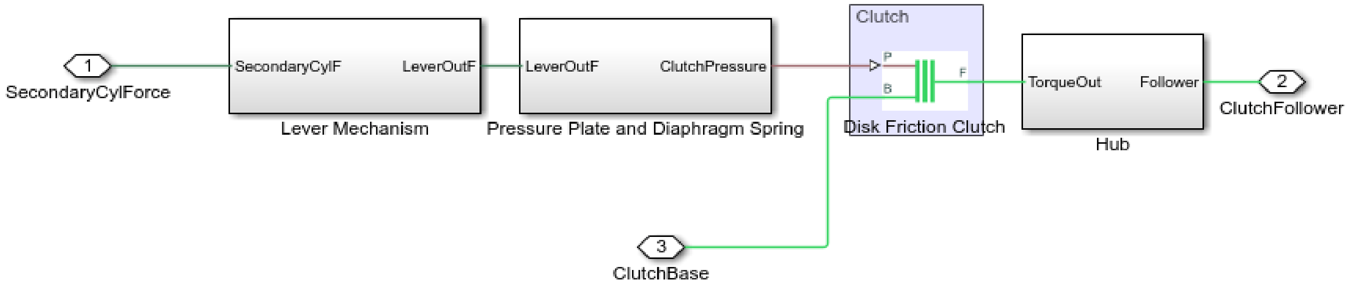
Secondary Cylinder Subsystem

The following figure shows the modeling part of the secondary cylinder.



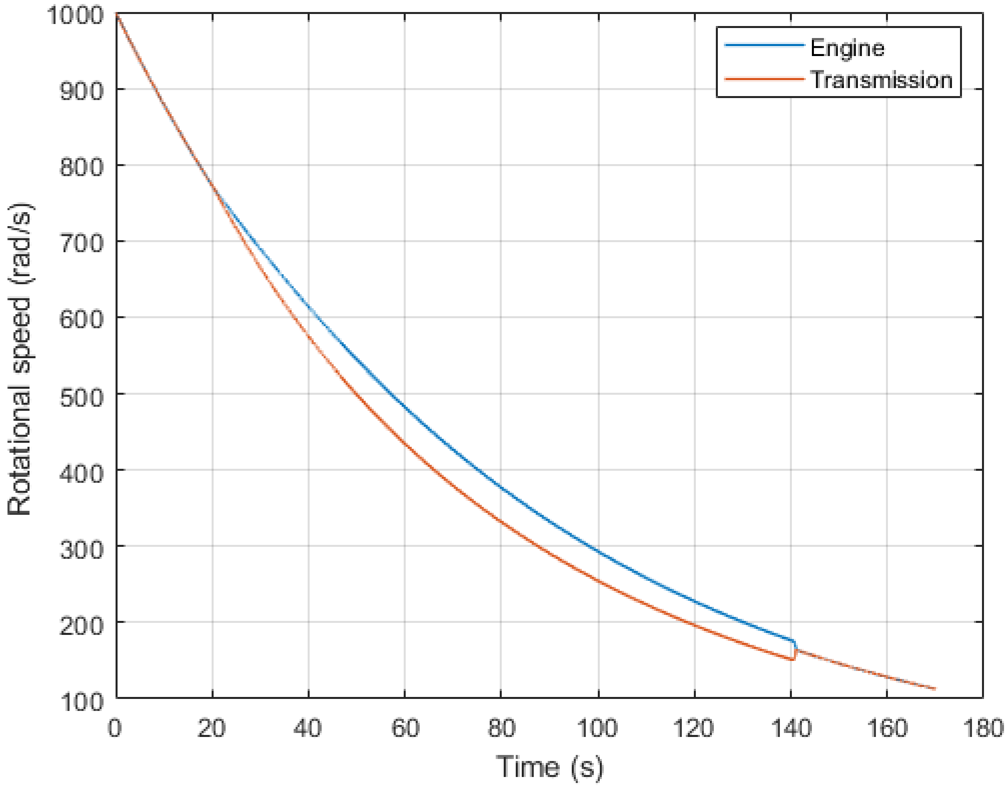
Automotive Clutch Subsystem

This subsystem shows how the clutch is modeled. The clutch consists of a lever mechanism, a diaphragm spring, a pressure plate, a set of clutch plates, and a hub. The lever transfers force to the diaphragm spring when a force is applied on the other end of the lever by the secondary cylinder. The applied force on the diaphragm spring causes the pressure plate to remove the pre-applied force on the clutch plate. The declutching takes place when the force on the clutch plate is removed through the removal of the force on the pressure plate by the diaphragm spring's action. No torque is transferred from the engine to the transmission after the declutching event.



Simulation Results from Simscape™ Logging

This model generates a plot of speeds versus time of the engine and the transmission systems during the declutching and clutch re-engaging scenario.

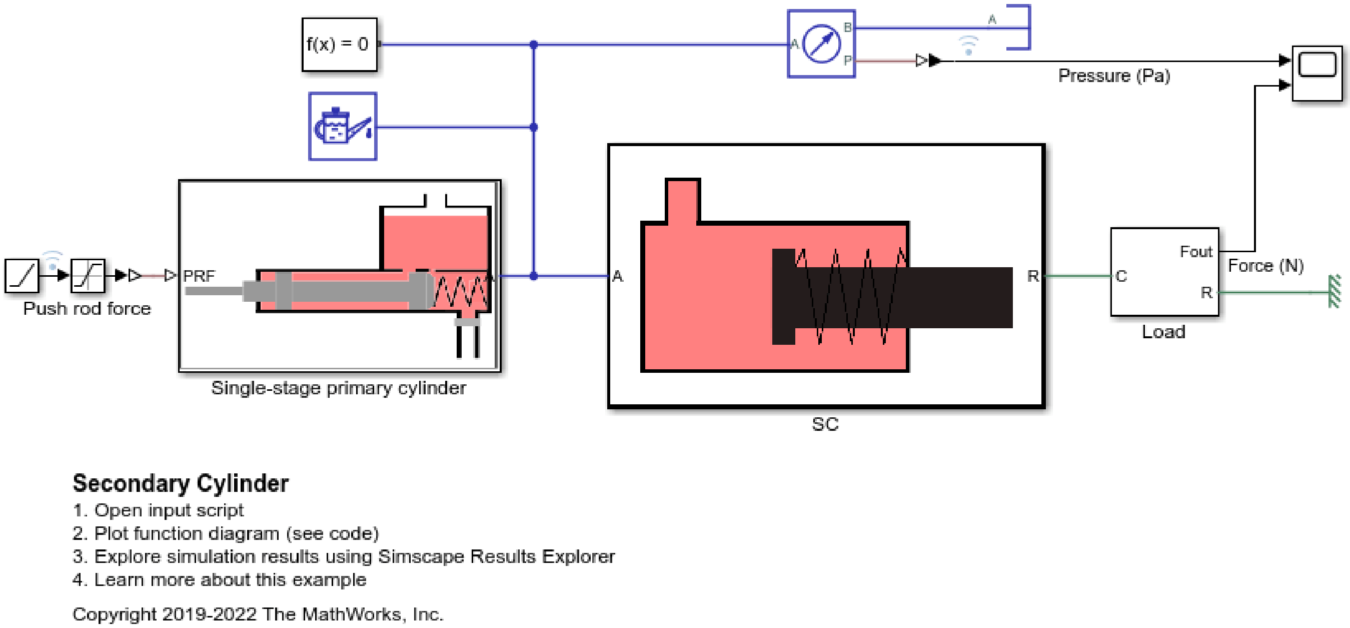


Secondary Cylinder

This example shows how to model, parameterize and, test a secondary cylinder. Given the numerical data extracted from the datasheet of the single-stage primary cylinder, the unknown parameters of the secondary cylinder are calculated. The model is simulated to generate the plot between the applied force to the single-stage primary cylinder and the pressure developed inside the secondary cylinder.

Model

The following figure shows the model of a secondary cylinder with the test harness. The output of the model is the plot between the applied force to the single-stage primary cylinder and the pressure developed inside the secondary cylinder.

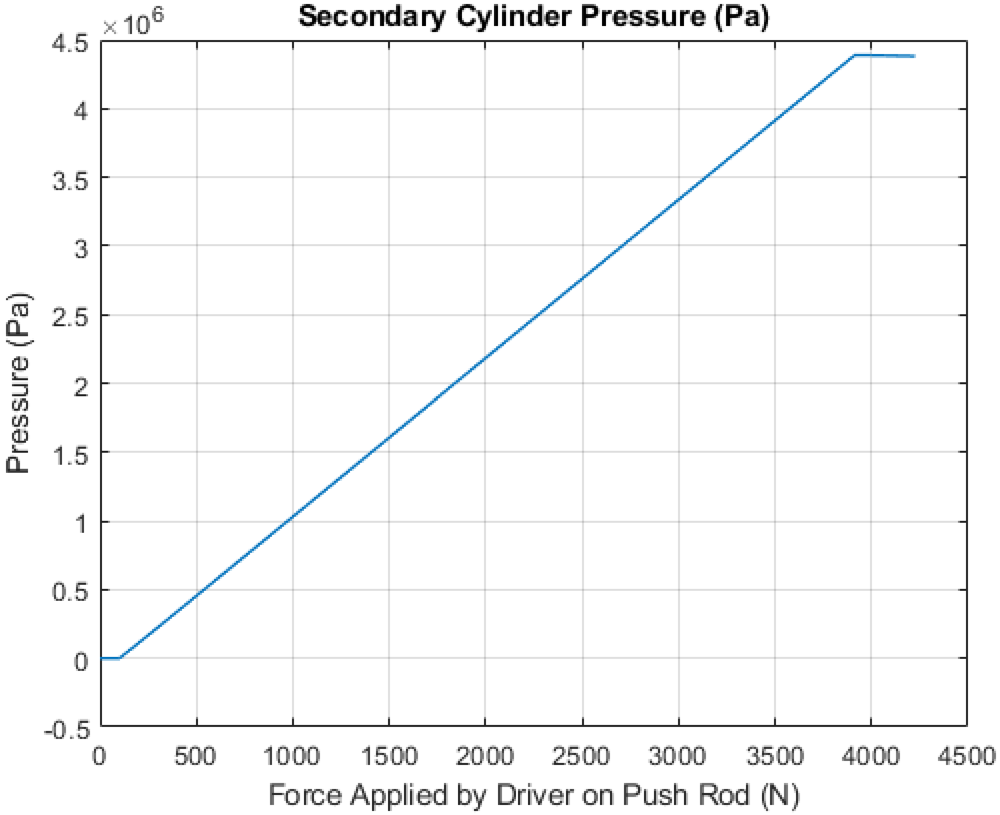


Secondary Cylinder

1. Open input script
2. Plot function diagram (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Simulation results from Simscape™ logging

This model generates a plot between the applied force to the single-stage primary cylinder and the pressure developed inside the secondary cylinder.

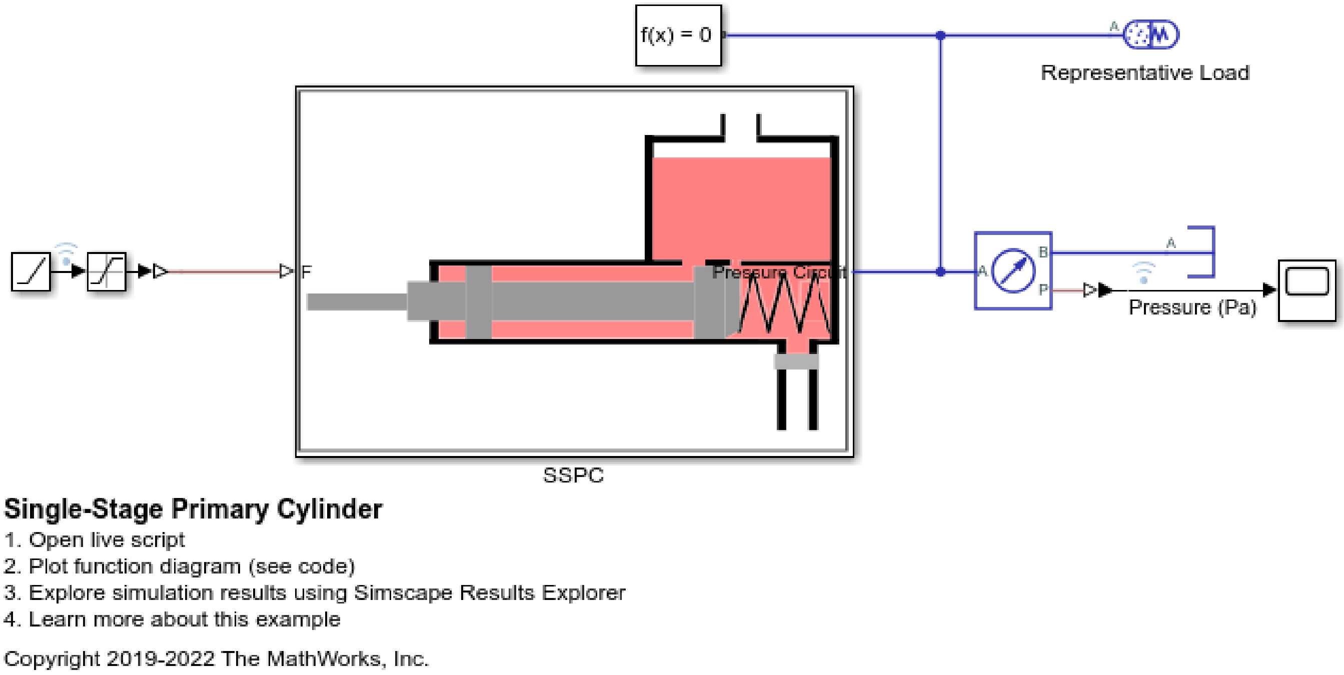

Secondary Cylinder Pressure (Pa)
×10^6
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
Pressure (Pa)
0
500
1000
1500
2000
2500
3000
3500
4000
4500
Force Applied by Driver on Push Rod (N)

Single-Stage Primary Cylinder

This example shows how to model, parameterize and test a single-stage primary cylinder starting from manufacturer datasheet information. Given the numerical data extracted from the datasheet, the unknown parameters are calculated. The model is then simulated and the resulting push rod force versus pressure relationship curve is compared with curve provided on the manufacturer datasheet.

Model

The following figure shows the model of a single-stage primary cylinder with the test harness. The output of the model is the plot of push rod force versus pressure relationship curve.



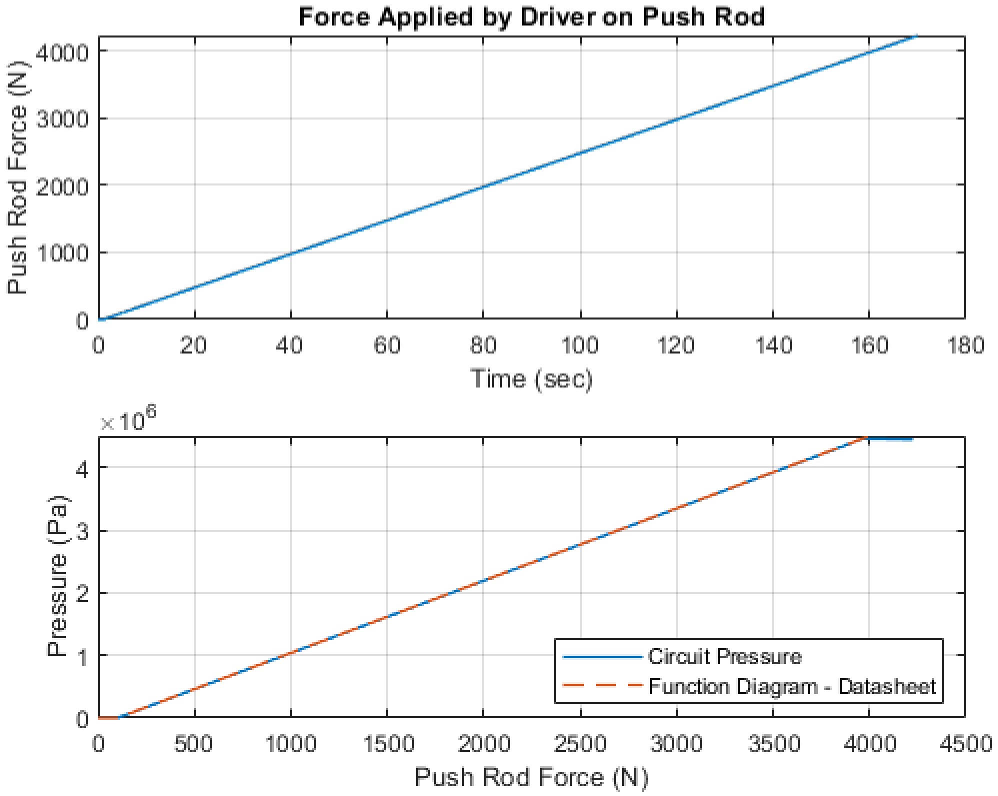
Single-Stage Primary Cylinder

1. Open live script
2. Plot function diagram (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2019-2022 The MathWorks, Inc.

Simulation results from Simscape™ logging

This script generates a plot of the push rod force versus pressure relationship in single-stage primary cylinder.

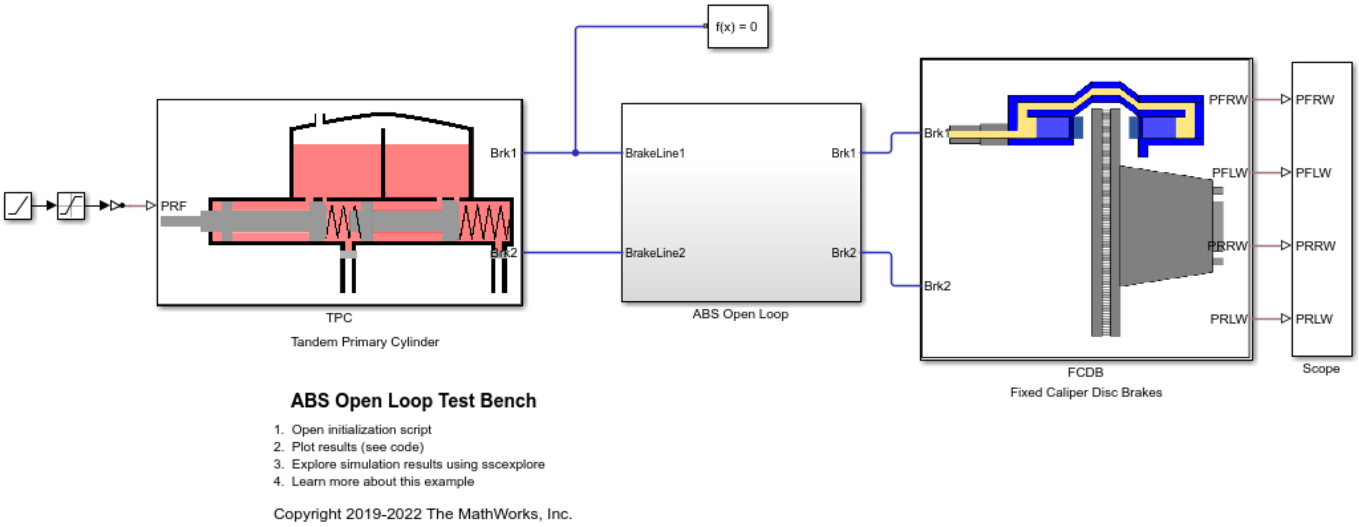

Force Applied by Driver on Push Rod
Push Rod Force (N)
4000
3000
2000
1000
0
0
20
40
60
80
100
120
140
160
180
Time (sec)
×10^6
Pressure (Pa)
4
3
2
1
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500
Push Rod Force (N)
Circuit Pressure
Function Diagram - Datasheet

ABS Open Loop Test Bench

This example shows a simple way of modeling of the anti-lock braking system (ABS) and using it with the manual braking components. The model simulates for the open loop ABS and shows the response of the pressures achieved in the caliper disc brakes. This model can be utilized in sizing of the release valve, apply valve and accumulator for the hydraulic control unit of the ABS.

Model

The following figure shows the model of an ABS system in open loop with the test harness. The output of the model is the plot of the pressures in the caliper disc brakes.



ABS Open Loop Test Bench

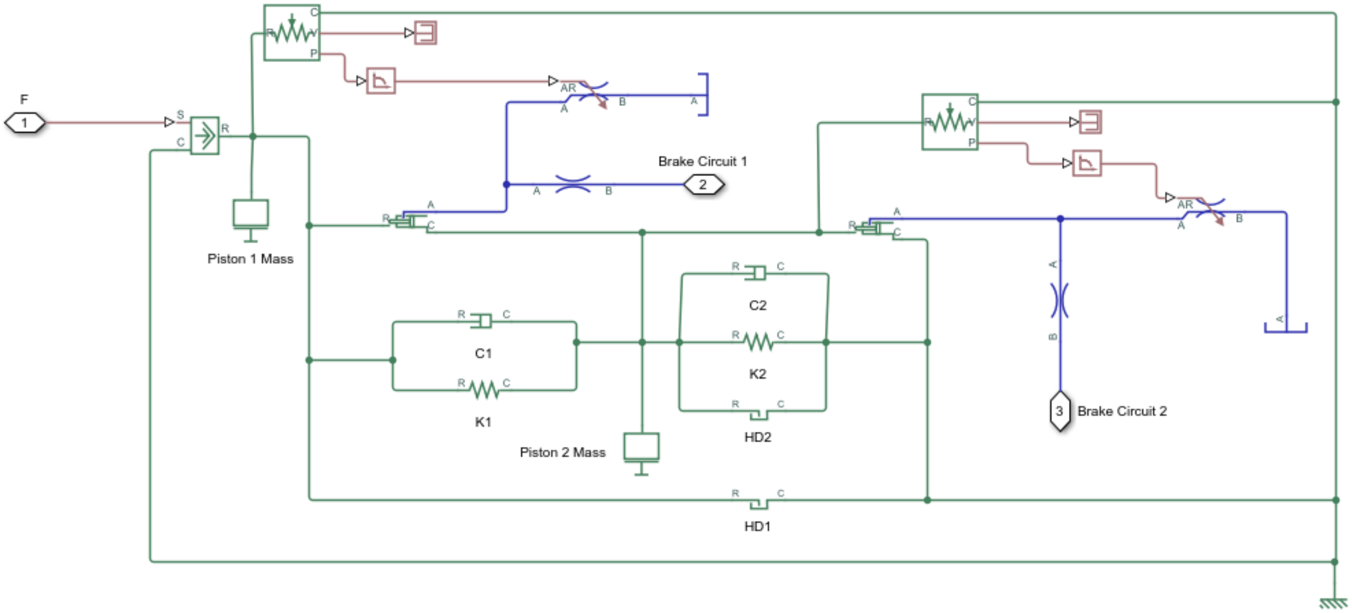
1. Open initialization script
2. Plot results (see code)
3. Explore simulation results using sscexplore
4. Learn more about this example

Copyright 2019-2022 The MathWorks, Inc.

Tandem Primary Cylinder Subsystem

This subsystem shows the modeling of the tandem primary cylinder.

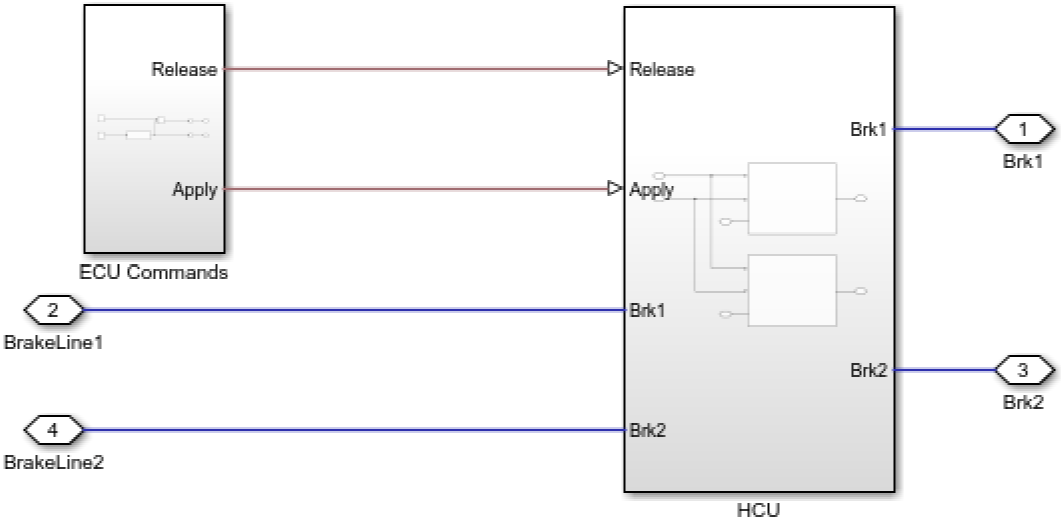
Open Subsystem



ABS Open Loop Subsystem

This subsystem shows the modeling of the ABS. It contains the electronic control unit (ECU) and the hydraulic control unit (HCU) subsystems.

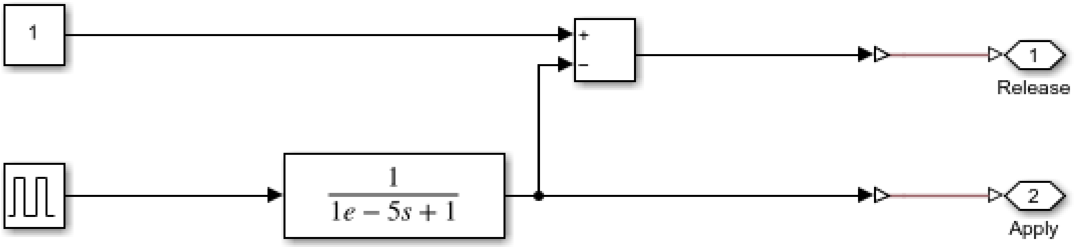
Open Subsystem



ECU Commands Subsystem

The ECU provides the control signals to the release and the apply valves.

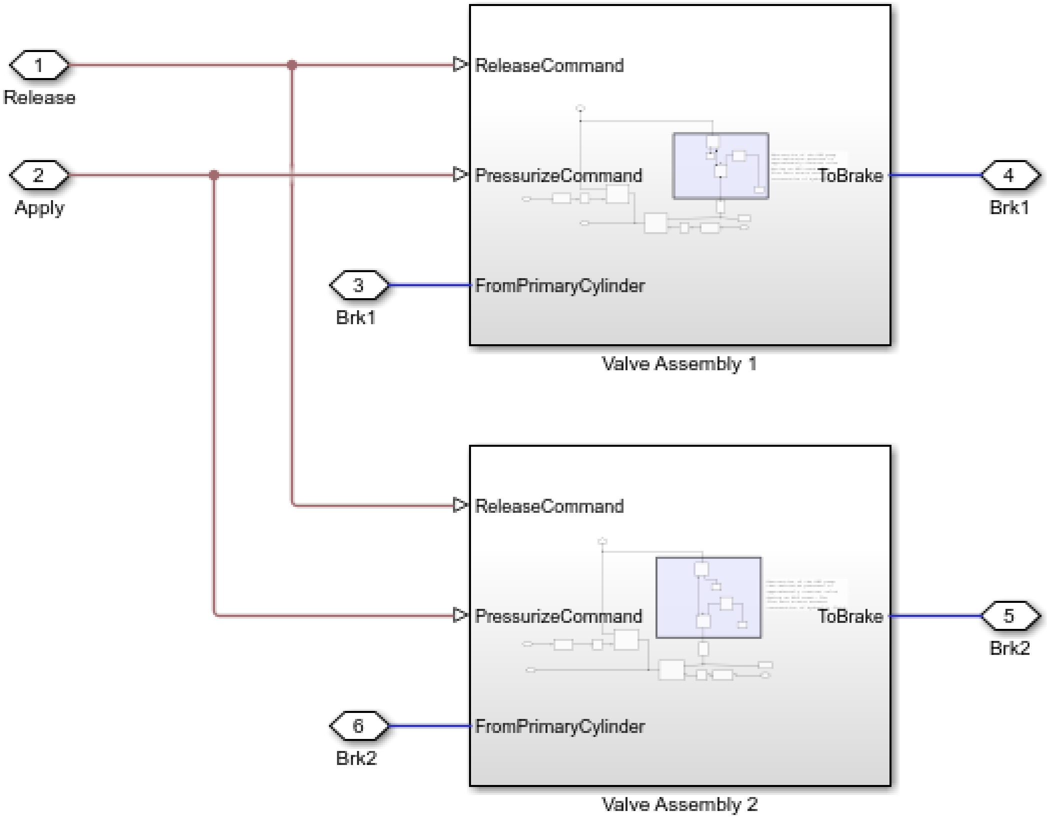
Open Subsystem



HCU Subsystem

The HCU subsystem shows, how to model the release valve, apply valve, pump and accumulator in the simplest possible way.

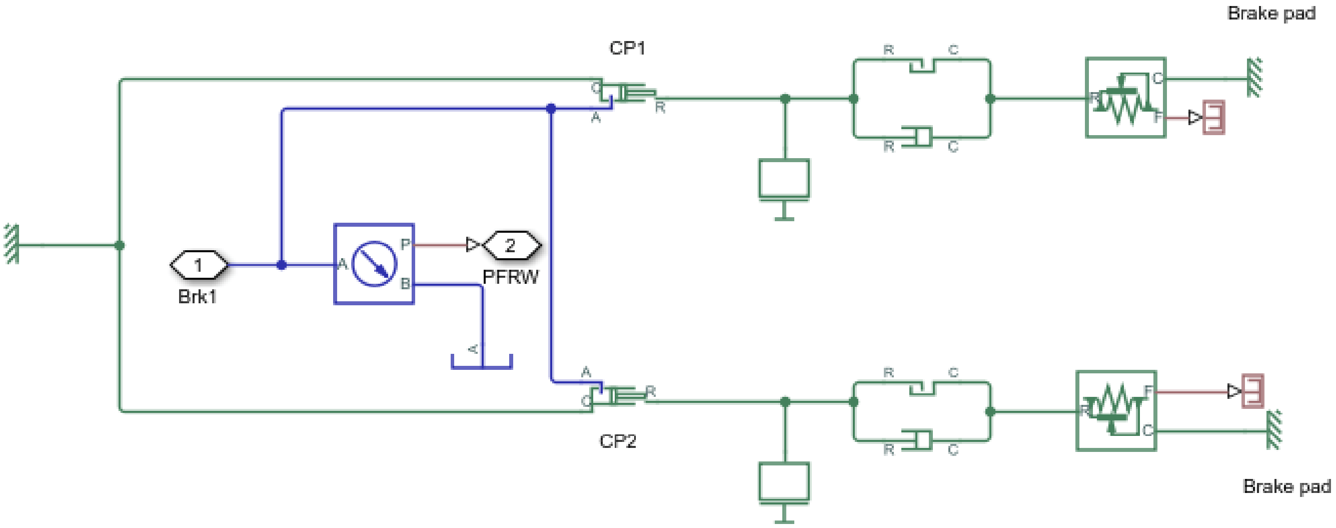
Open Subsystem



FRW (Front Right Wheel) Subsystem

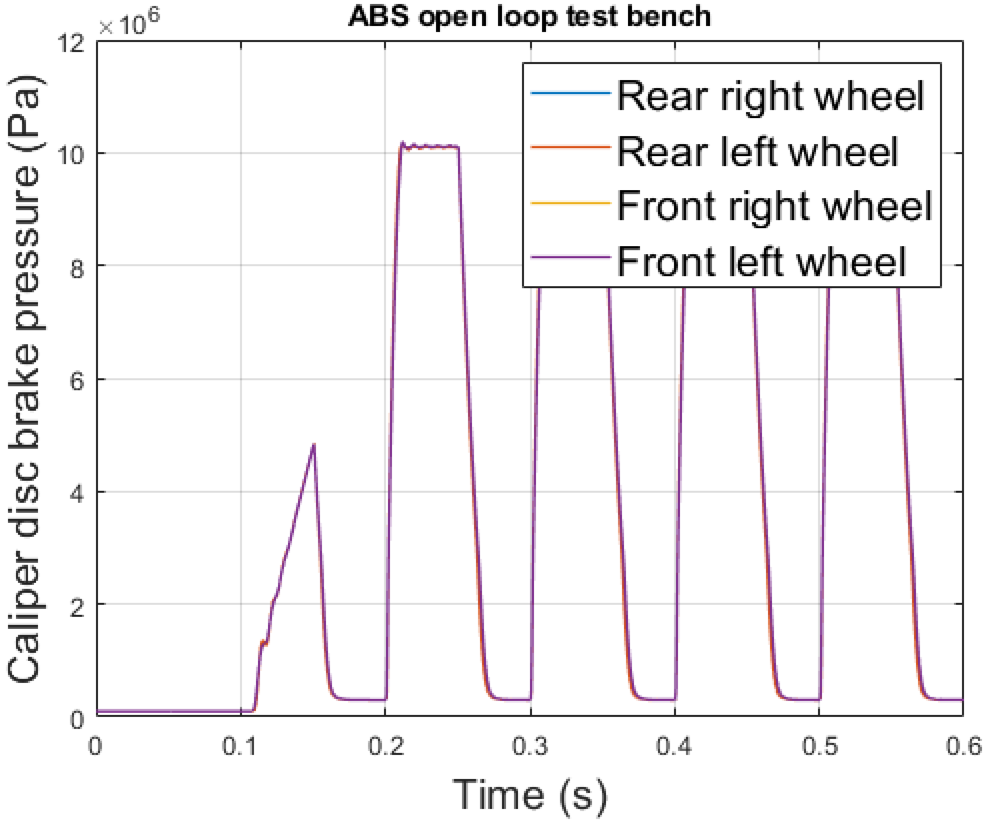
This subsystem shows the modeling of the fixed caliper disc brake's front right wheel subsystem. The friction plates are not modeled in this subsystem as the vehicle mass body is not attached to the brakes. Other wheels brakes are also modeled in the same way.

Open Subsystem



Simulation Results from Simscape™ Logging

The model has two inputs. The first input is a force applied to the tandem primary cylinder's push rod. The second input is the ECU commands applied to the release and apply valve in the ABS open loop subsystem. Based on the given inputs, the model generates a plot of the pressures in the caliper cylinders of the brakes.

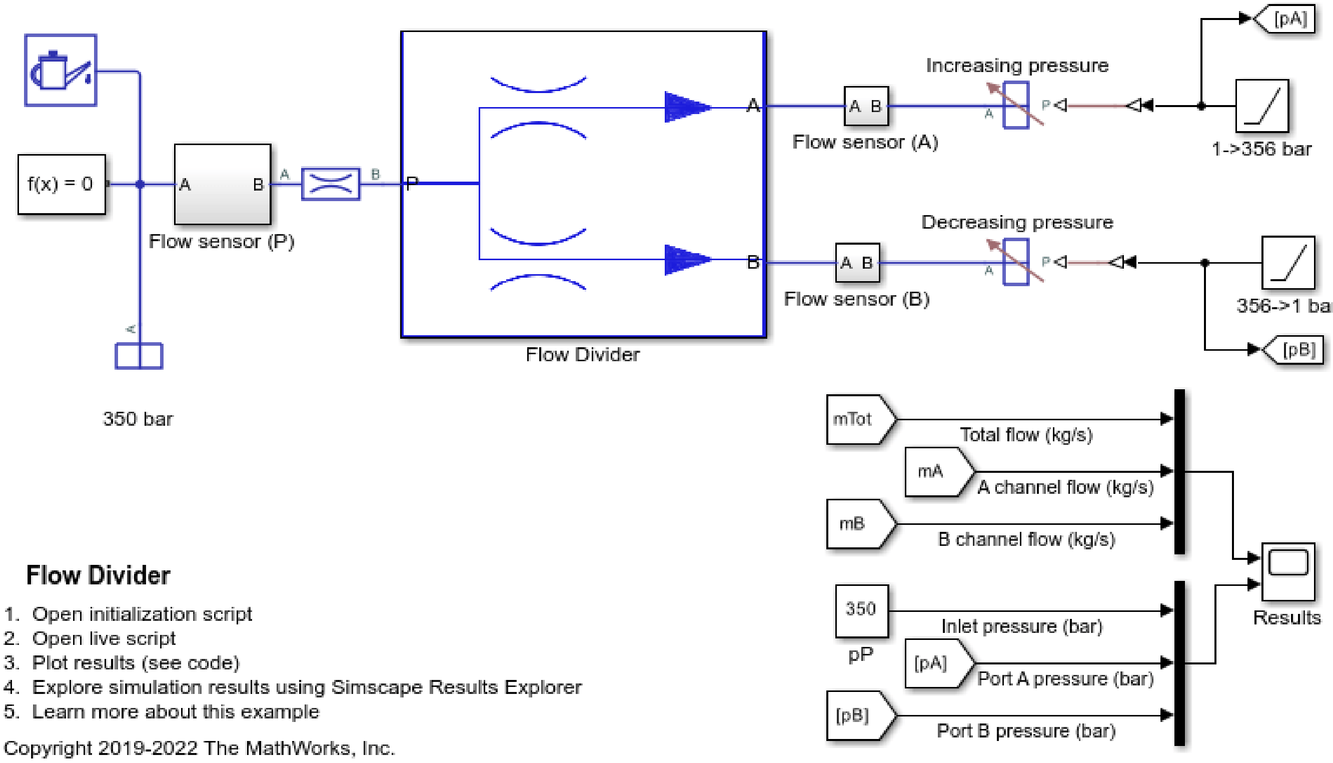

ABS open loop test bench
Caliper disc brake pressure (Pa)
×10^6
12
10
8
6
4
2
0
0
0.1
0.2
0.3
0.4
0.5
0.6
Time (s)
Rear right wheel
Rear left wheel
Front right wheel
Front left wheel

Flow Divider

This example shows a simple way of modeling of the flow divider and using it with the load harness. The flow divider helps in dividing the input flow in desired percentage when the loads connected to the two lines are unequal. Flow divider allows flow only in forward direction. Therefore when it is used with the loads where unloading is done through reverse flows, then it should be used in conjunction with the direction control valves.

Model

The following figure shows the model of a flow divider with the test harness. The output of the model is the plot of flow through the channels A and B for a given input flow.



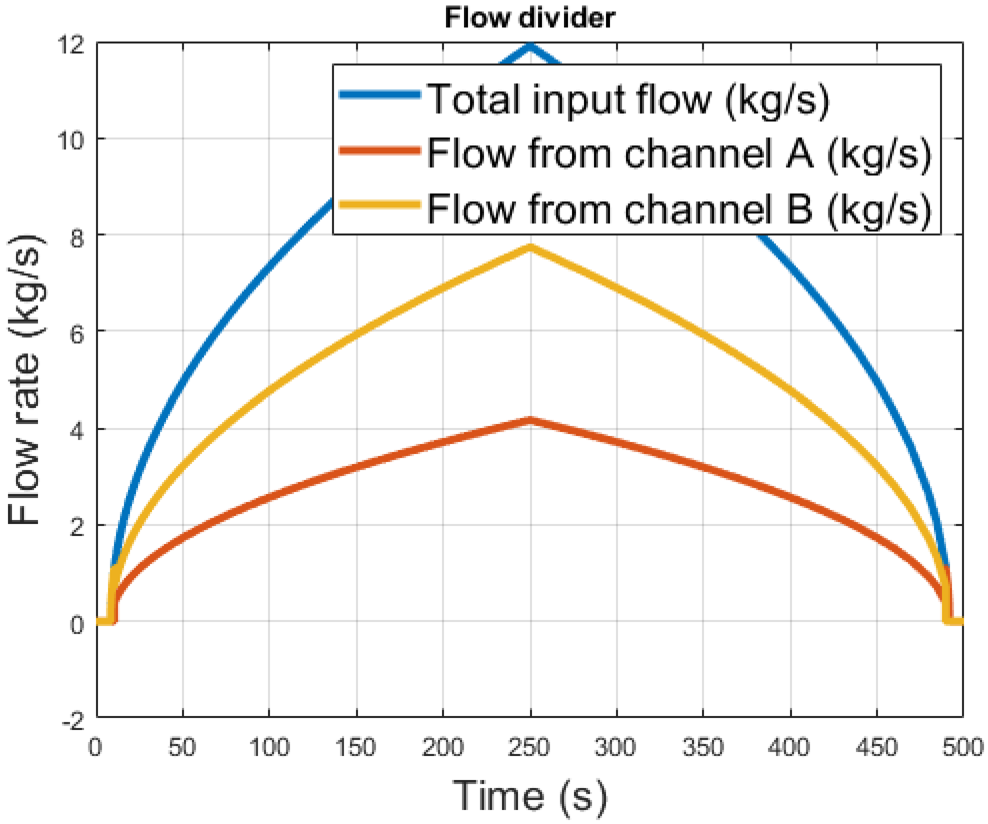
Flow Divider

1. Open initialization script
2. Open live script
3. Plot results (see code)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example

Copyright 2019-2022 The MathWorks, Inc.

Simulation Results from Simscape Logging

This script generates a plot of flow through the channels A and B for a given input flow.

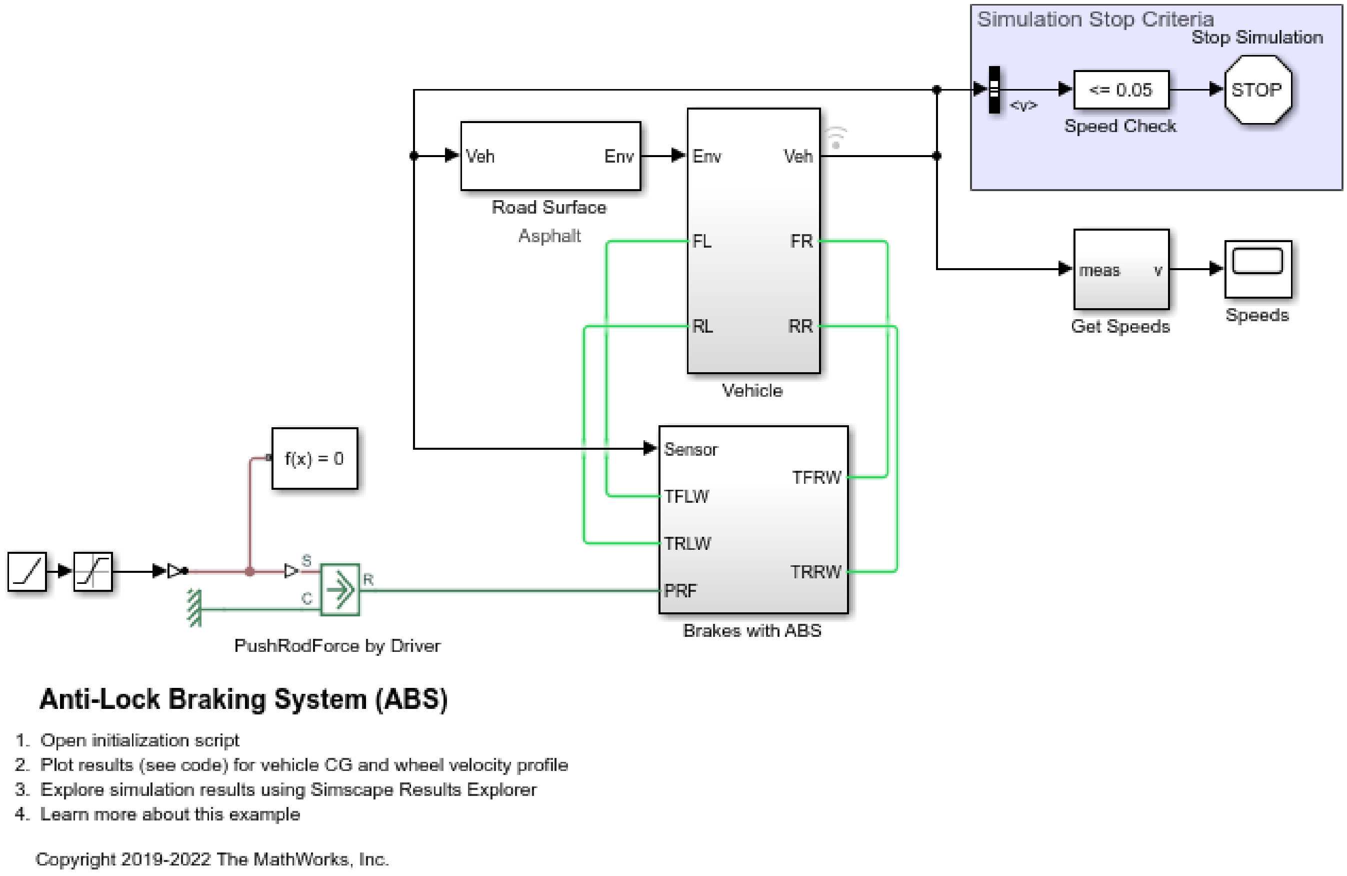

Flow divider
Total input flow (kg/s)
Flow from channel A (kg/s)
Flow from channel B (kg/s)
Flow rate (kg/s)
12
10
8
6
4
2
0
-2
0
50
100
150
200
250
300
350
400
450
500
Time (s)

Anti-Lock Braking System (ABS)

This example shows a simple way of modeling an ABS braking system. The model shows the velocity profile responses achieved for the vehicle CG and the wheels.

Model

The following figure shows the model of an ABS system with a test harness. The output of the model is the plot of the velocity profile achieved for the vehicle CG and the wheels.



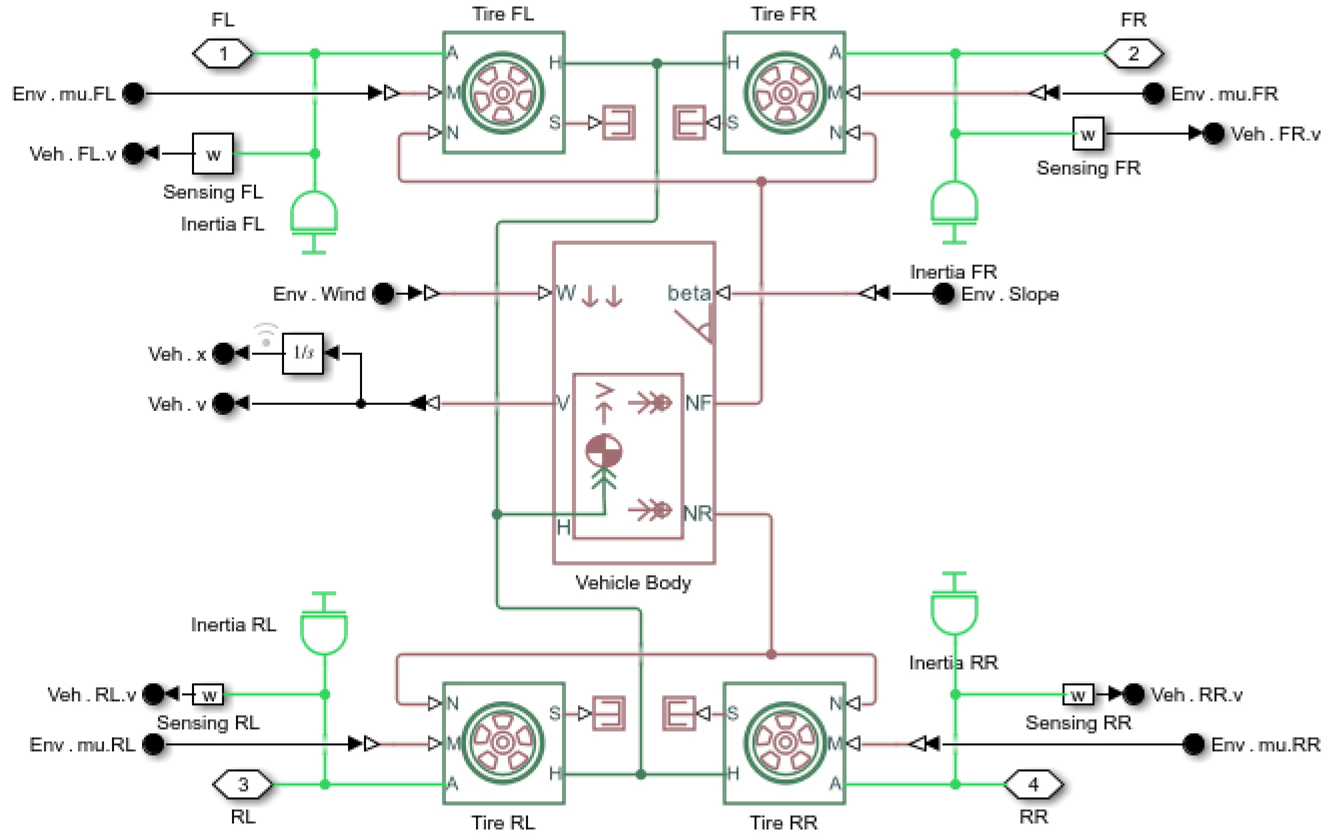
Anti-Lock Braking System (ABS)

1. Open initialization script
2. Plot results (see code) for vehicle CG and wheel velocity profile
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2019-2022 The MathWorks, Inc.

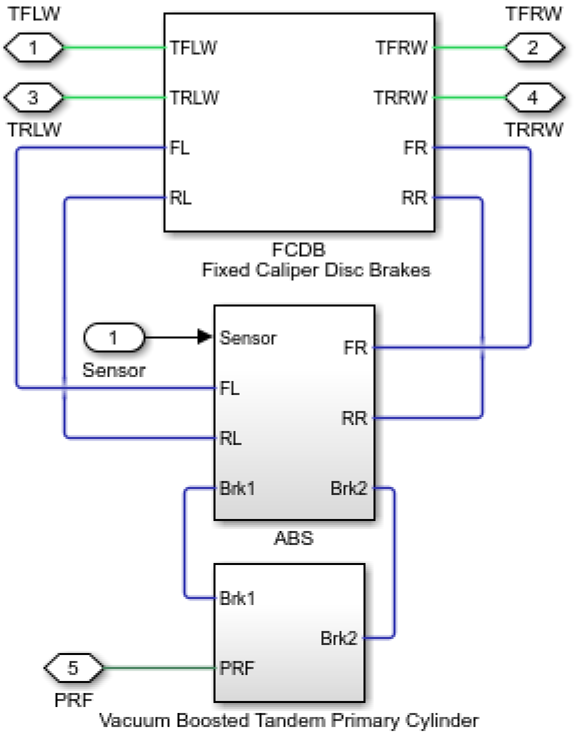
Vehicle Subsystem

This subsystem shows the modeling of the vehicle body with the four wheels.



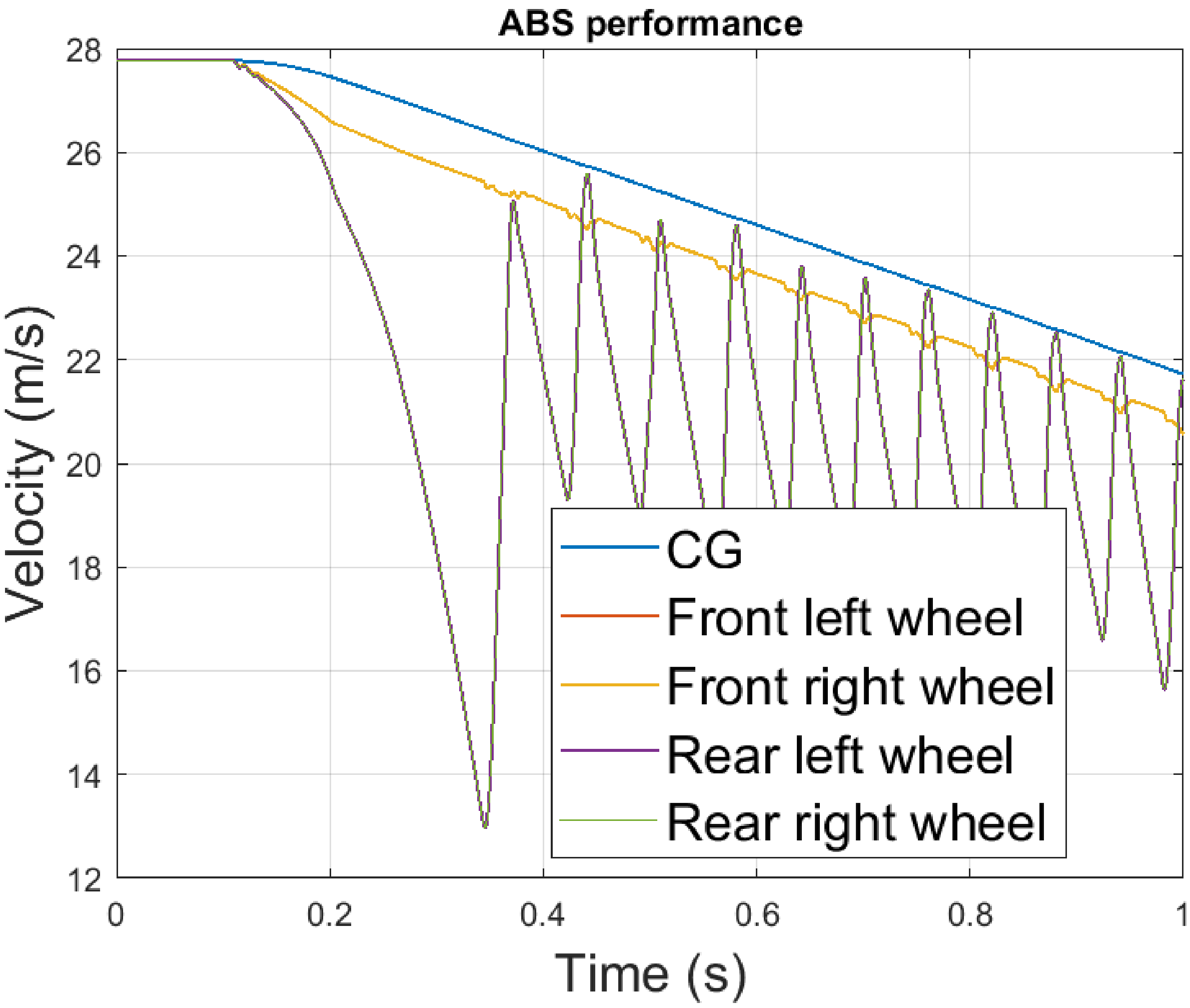
Brakes with ABS

This subsystem shows the modeling of the brakes with the ABS.



Simulation results from Simscape™ logging

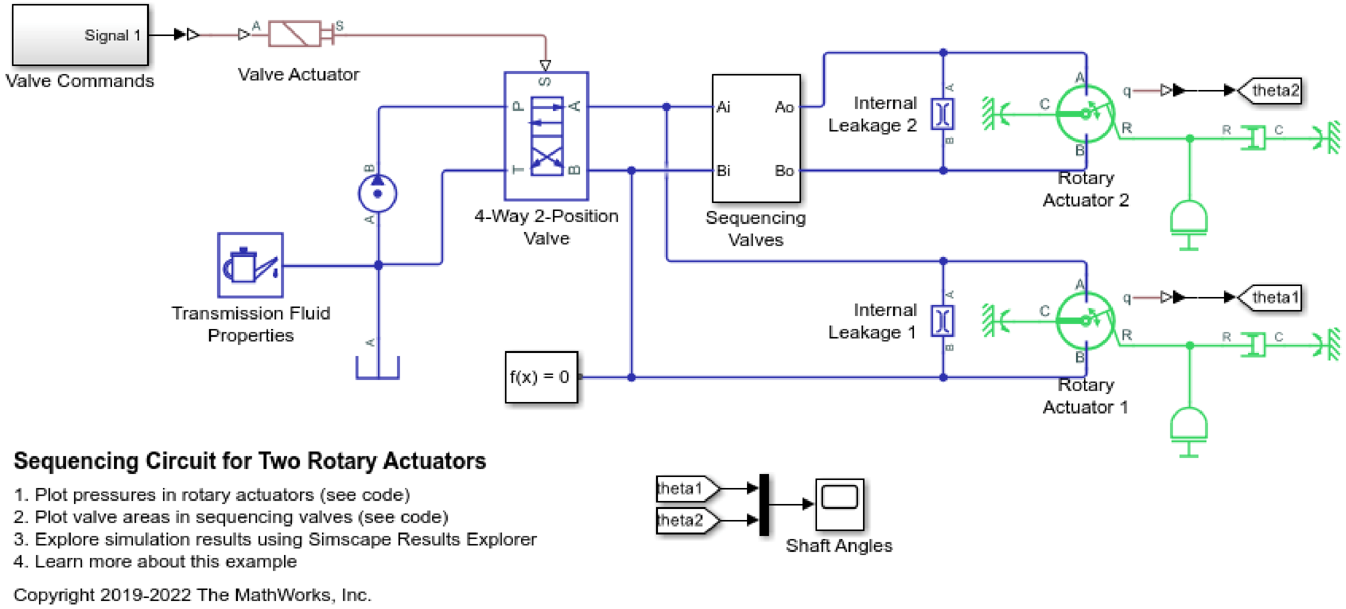
Based on the given inputs, the model generates a plot of the velocity profile achieved for the vehicle CG and the wheels.


ABS performance
Velocity (m/s)
28
26
24
22
20
18
16
14
12
0
0.2
0.4
0.6
0.8
1
Time (s)
CG
Front left wheel
Front right wheel
Rear left wheel
Rear right wheel

Sequencing Circuit for Two Rotary Actuators

This example shows a sequencing circuit that is based on four check valves installed in the pressure and return lines of the second rotary actuator. The cracking pressure of the meter-in check valves is set high enough to prevent flow into rotary actuator 2 while rotary actuator 1 is rotating, but lower than the pressure that develops once rotary actuator 1 reaches its hard stop. As a result, rotary actuator 2 starts moving only after rotary actuator 1 completes its stroke.

Model

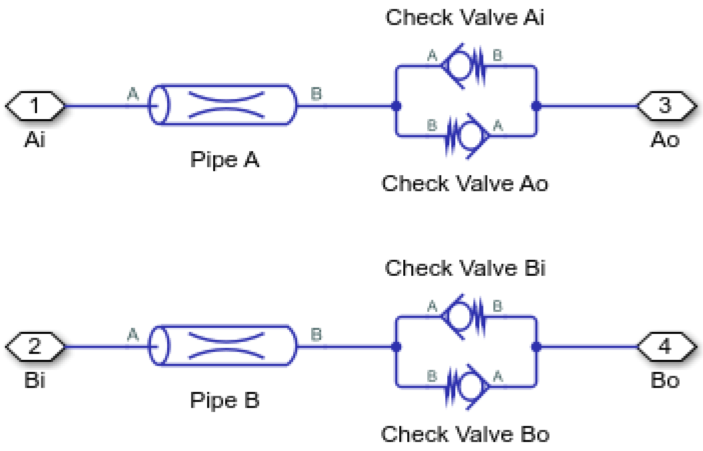


Sequencing Circuit for Two Rotary Actuators

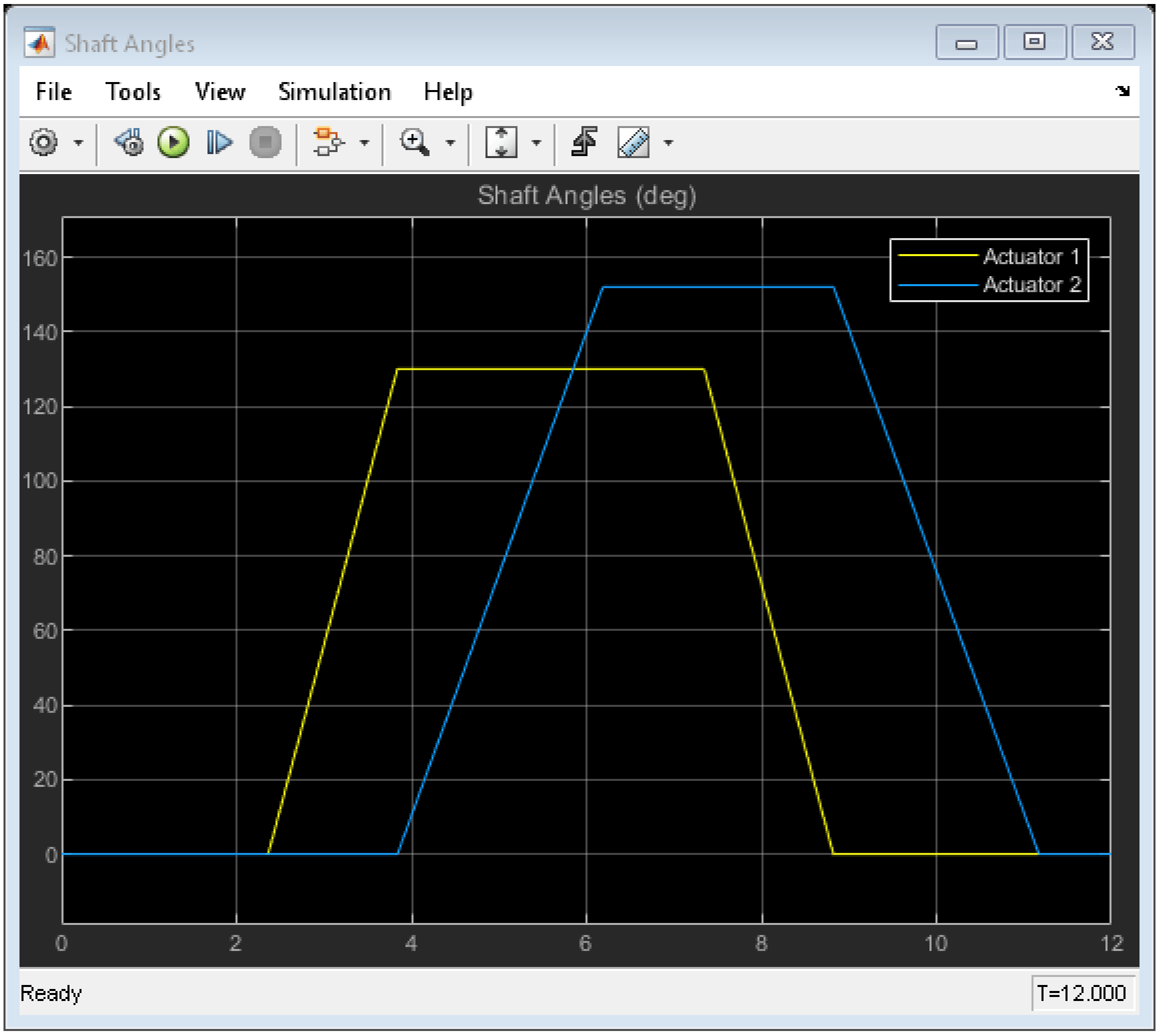
1. Plot pressures in rotary actuators (see code)
2. Plot valve areas in sequencing valves (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2019-2022 The MathWorks, Inc.

Sequencing Valves Subsystem

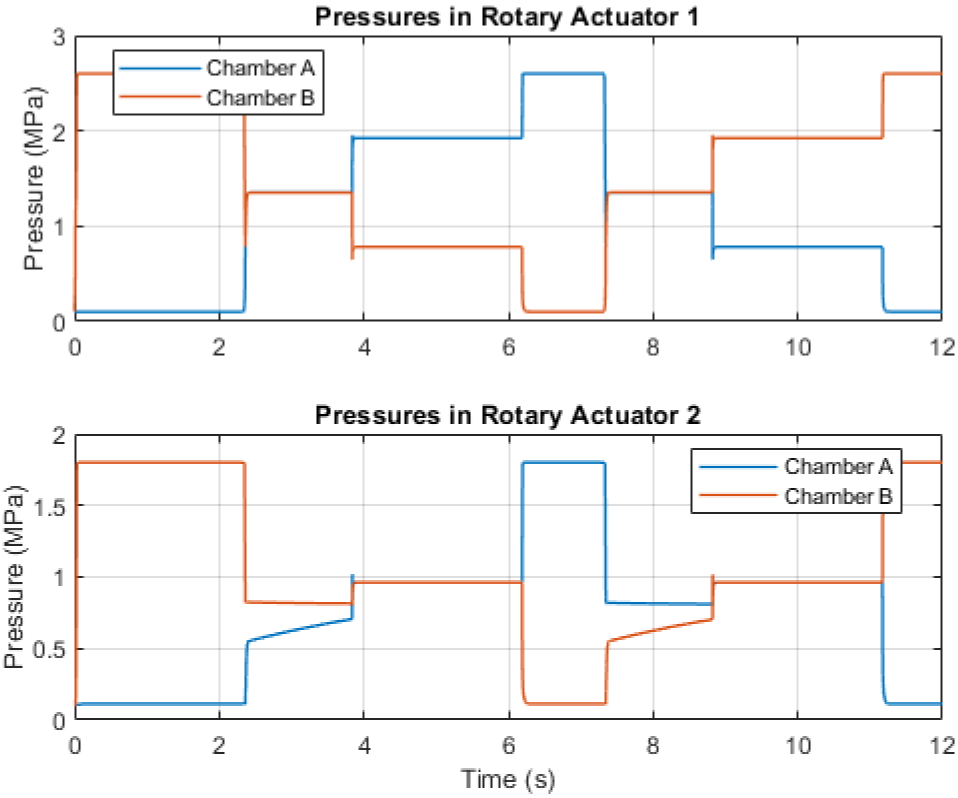


Simulation Results from Scopes

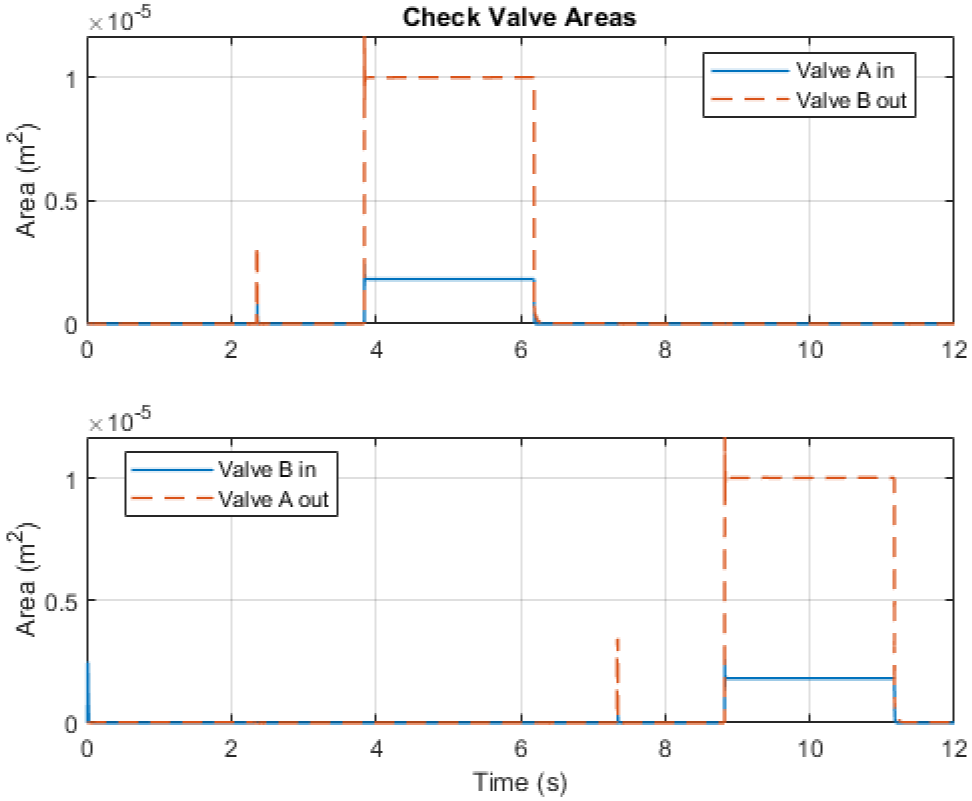


Simulation Results from Simscape Logging

This plot shows the sequence of pressures in the rotary actuator chambers. At the start of the simulation, both actuators are driven against their hard stops at 0 degree. At 2 seconds, the high pressure line is connected to chamber A of the actuators. The sequencing valves (check valves) prevent flow to rotary actuator 2 and maintain greater pressure in its chamber B over chamber A until rotary actuator 1 reaches its hard stop at 3.8 seconds. At that point, rotary actuator 1 is driven against its hard stop at maximum stroke and pressure builds up sufficiently to open the check valves. This lets flow into chamber A of rotary actuator 2 and causes it to rotate to its maximum stroke. At 7 seconds, the high pressure line switches and the opposite set of check valves repeat the sequence for the actuator rotation in the opposite direction.



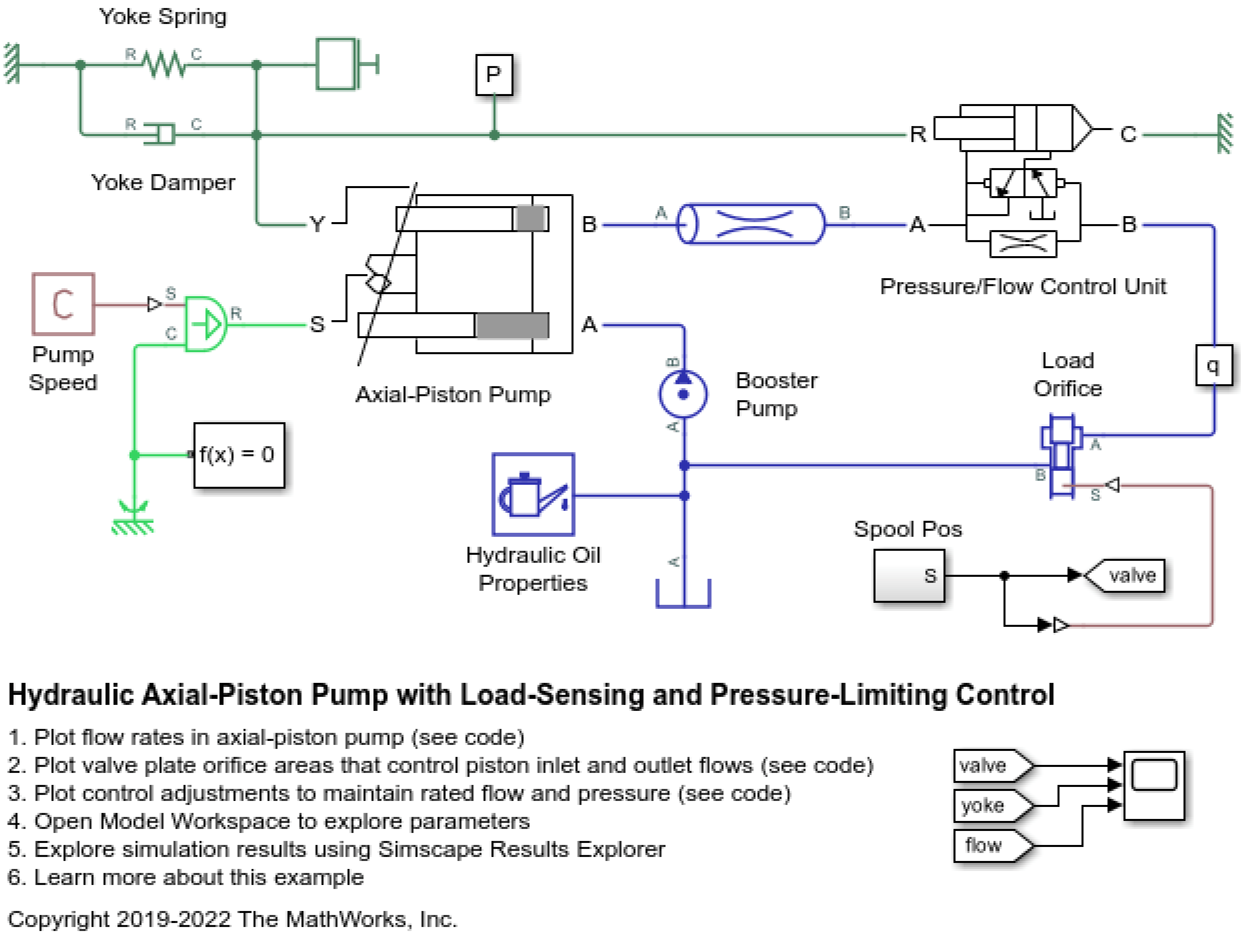
This plot shows the check valve areas that maintain the sequence of operations in the two rotary actuators. When rotary actuator 1 rotates, the meter-in check valve to chamber A of rotary actuator 2 remains closed until rotary actuator 1 completes its stroke. It then opens to enable flow into rotary actuator 2 so that it can rotate. The same sequence occurs when rotary actuator 1 rotates in the opposite direction.


Check Valve Areas
×10^-5
Area (m^2)
1
0.5
0
0
2
4
6
8
10
12
Valve A in
Valve B out
×10^-5
Area (m^2)
Valve B in
Valve A out
Time (s)

Hydraulic Axial-Piston Pump with Load-Sensing and Pressure-Limiting Control

This example shows a test rig designed to investigate the interaction between an axial-piston pump and a typical control unit simultaneously performing the load-sensing and pressure-limiting functions. To increase the fidelity of the simulation, this example uses a detailed model of the pump that accounts for the interaction between the pistons, swash plate, and valve plate.

Model



Hydraulic Axial-Piston Pump with Load-Sensing and Pressure-Limiting Control

1. Plot flow rates in axial-piston pump (see code)
2. Plot valve plate orifice areas that control piston inlet and outlet flows (see code)
3. Plot control adjustments to maintain rated flow and pressure (see code)
4. Open Model Workspace to explore parameters
5. Explore simulation results using Simscape Results Explorer
6. Learn more about this example

Copyright 2019-2022 The MathWorks, Inc.

Test Rig Description

The pump is modeled by the Axial-Piston Pump subsystem. The prime mover rotating the pump is represented by an angular velocity source. The pump's intake is connected to the output of a low pressure booster pump of 0.5 MPa. The pump output passes through a pipe, a control unit, and a variable orifice that acts as the load. To test the control unit's response to a variable load, the load orifice area is changed during simulation via the Spool Pos subsystem.

The control unit is modeled by the Pressure/Flow Control Unit subsystem. The load-sensing function of the pump control uses an orifice installed on the output flow path of the pump. The control unit keeps the pressure differential across this orifice constant regardless of the pump loading by adjusting the yoke displacement, which affects the tilt angle of the pump swash plate. This helps maintain the desired the flow rate through the orifice as well as prevent the pump pressure from exceeding the preset value.

The test rig basic parameters are:

```
Pump maximum displacement                         7.8877e-6 m^3/rad
Cylinder block pitch radius                       0.04 m
Piston diameter                                   0.015 m
Number of pistons                                 5
Maximum piston stroke                             0.06 m
Swash plate maximum angle                         35 deg (0.6109 rad)
Arm length between the control actuator
  and the swash plate pivoting point              0.055 m
Swash plate control actuator stroke               0.04 m
Diameter of the orifice at the bottom
  of the piston chamber                           0.007 m
Pump maximum rated speed                          2500 RPM
Maximum pressure                                  27 MPa
Rated flow rate                                   1.1e-3 m^3/s
```

Axial-Piston Pump Subsystem

The pump under investigation has five pistons. Its schematic is shown below:

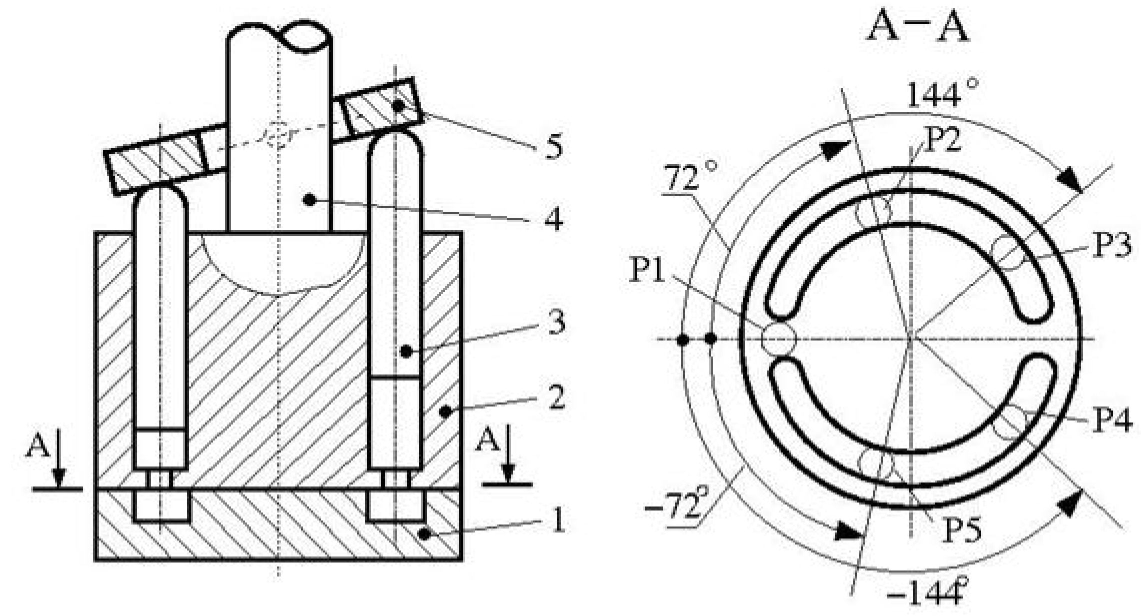


Figure 1: Axial-Piston Pump Schematic

where

1 - Valve plate (porting plate)

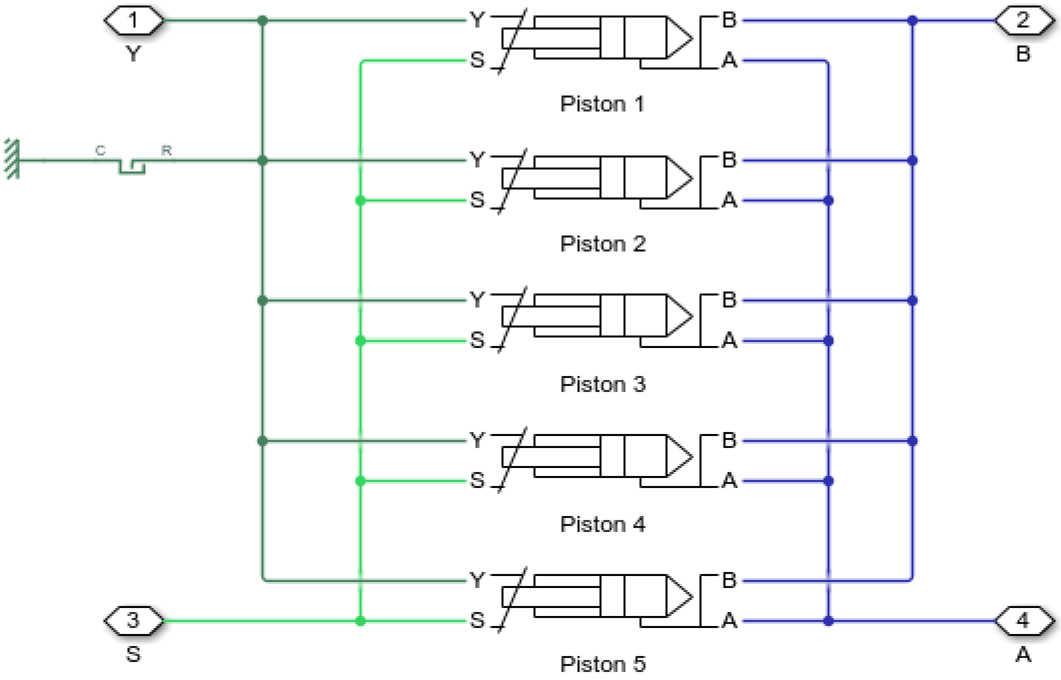
2 - Cylinder block (rotor)

3 - Piston

4 - Drive shaft

5 - Swash plate

The five pistons are modeled in the Axial-Piston Pump subsystem:

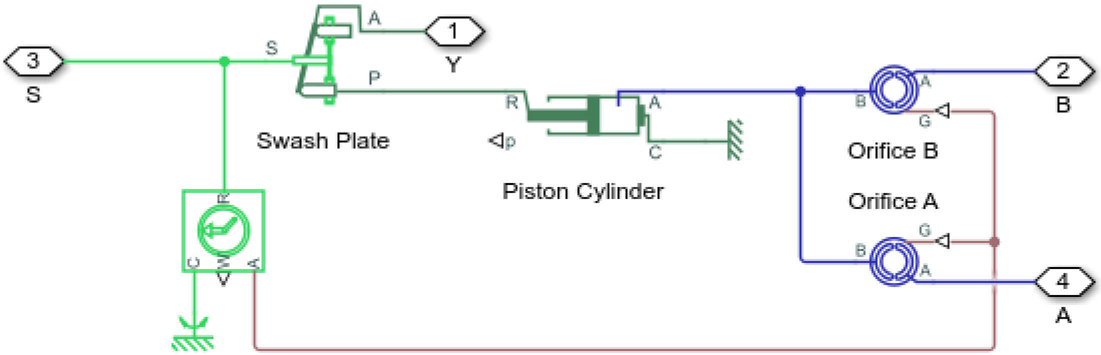


The pistons are identical and are modeled in their own identical subsystems. They are all connected to the following ports of the pump model:

- S - Drive shaft
- Y - Yoke connected with the inclined plate of the swash mechanism
- A - Pump inlet (intake) port
- B - Pump outlet (discharge) port

The yoke displacement is limited by a hard stop.

Piston Subsystem

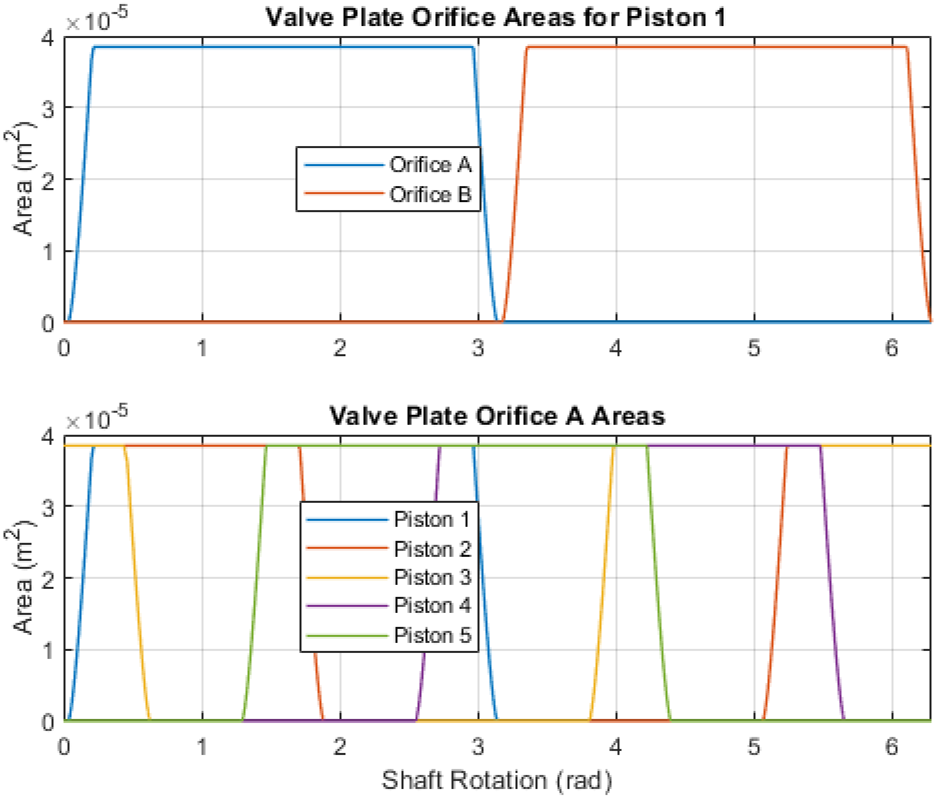


The piston model consists of a Single-Acting Actuator (IL) block connected mechanically to the drive shaft via the Swash Plate block. The piston is also hydraulically connected to ports A and B via two Valve Plate Orifice (IL) blocks. Ports A and B represent the pump intake and discharge ports, respectively.

The piston sucks in liquid during its upstroke and expels liquid during its downstroke. The two Valve Plate Orifice (IL) blocks models the interaction between the fixed orifice at the bottom of the piston chamber and the two crescent-shaped grooves on the valve plate so that the piston is connected to port A during its upstroke and port B during its downstroke.

The following plot shows the flow areas formed by the interaction between the piston and the two crescent-shaped grooves on the valve plate. As the cylinder block rotates, the piston chamber orifice slides over the groove to open the flow area and then slides off the groove to close the flow area. The two grooves are separated by a phase angle of 180 degrees so that the piston slides over the groove connected to the intake (A) during the upstroke and slides over the groove connected to the discharge (B) during the downstroke. This results in the alternately open flow areas as shown in the first subplot.

The pistons are spread evenly along the pitch circle of the cylinder block, as shown in Figure 1. This makes the angle between pistons 360/5 = 72 degrees. Therefore, the phase angles of the Valve Plate Orifice (IL) blocks of each piston is offset from the previous piston by 72 degrees. This results in the evenly distributed and overlapping open flow areas for the intake groove as shown in the second subplot.



The tilt of the swash plate is what produces the piston motion as the cylinder block rotates. A Swash Plate block is connected to the Single-Acting Actuator (IL) to actuate that piston. Similar to the Valve Plate Orifice (IL) blocks, the phase angle of the Swash Plate block of each piston is offset from the previous piston by 72 degrees.

A greater swash plate angle results in larger piston motion which produces greater flow volume. The piston stroke must be long enough to allow the piston to reciprocate even at the maximum swash plate angle:

$$x_{stroke} > 2r_p \tan\alpha_{max}$$

where $r_p = 0.04$ m is the pitch radius and $\alpha_{max} = 35$ deg is the maximum swash plate angle. This means the piston stroke must be greater than 0.056 m and is thus set to 0.06 m.

The piston initial positions must be equal to half of the stroke if the swash plate angle is zero. It is offset from half stroke depending on the phase angle of the piston if the initial swash plate angle is nonzero. The initial swash plate angle depends on the initial control actuator position. Therefore, the initial position of each piston is

$$x_{p0} = \frac{x_{stroke}}{2} - x_{a0}\frac{r_p}{r_a}\cos\gamma$$

where x_{a0} is the swash plate control actuator initial position, r_a is the swash plate control actuator arm, and γ is the phase angle of the piston in the cylinder block. This calculation is implemented in the Initialization tab of the Axial-Piston Pump subsystem mask.

Pressure/Flow Control Unit Subsystem

The purpose of the control unit is to implement two functions: load sensing and pressure limiting. Load sensing is implemented by maintaining a set pressure differential across a flow control orifice. The flow control orifice is installed in the output flow of the pump. The following figure shows the schematic of the control unit where the ports A and B are connected in series with the pump output.

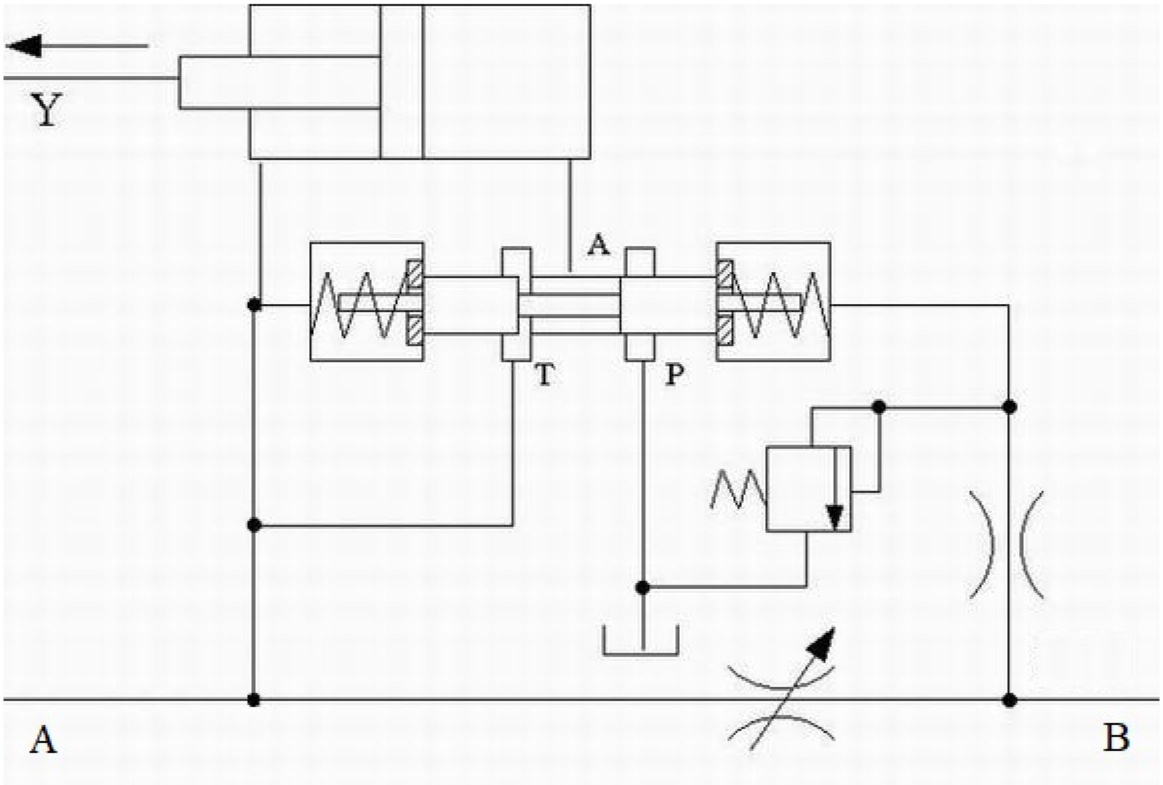
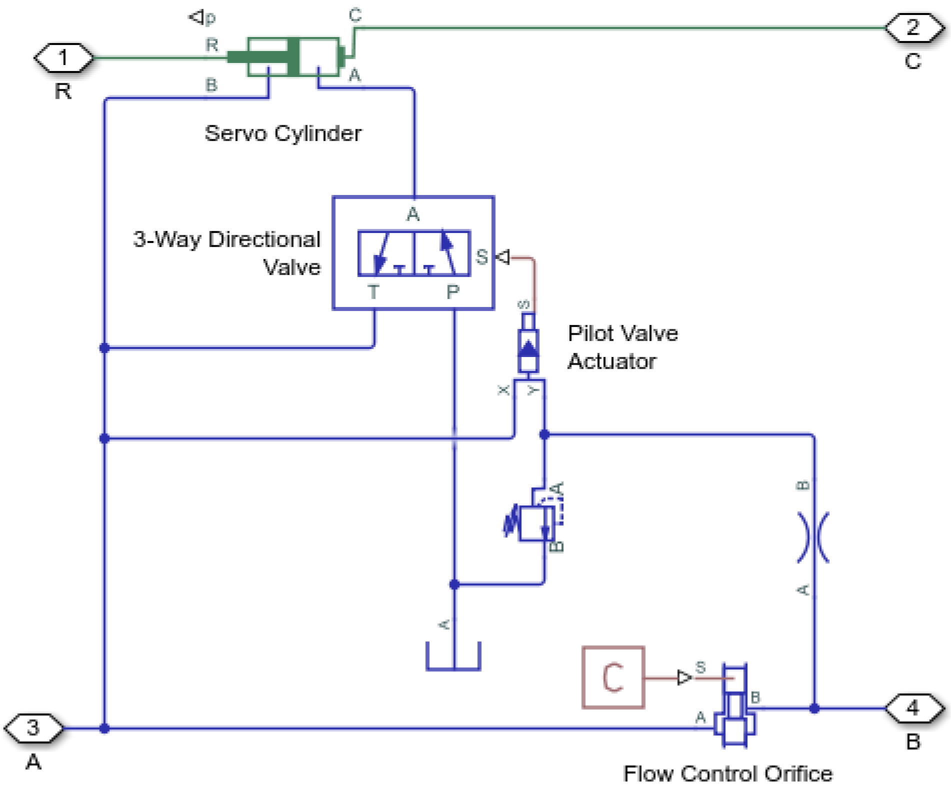


Figure 2: Pressure/Flow Control Unit Schematic

The pressure before and after the flow control orifice acts on the side faces of the 3-way directional valve and shifts the valve spool proportionally to the pressure difference and the setting of the two centering springs. The valve connections are selected such that an increase in pressure differential across the flow control orifice opens the directional valve path P-A and closes path A-T. The control actuator is arranged as a single-rod differential hydraulic cylinder with the rod connected to the yoke of the swash plate. The pump displacement is increased if the rod moves in the positive direction shown in the schematic. Due to the difference in piston areas of the two sides of the cylinder, the displacement is increased if both cylinder chambers are connected ot the pump and decreased if the chamber without the rod is connected to the tank. As a result, an increase in pressure differential across the flow control orifice causes the pump to decrease its displacement until it returns to the preset value.

The pressure-limiting function prevents the pump output pressure from exceeding the preset value. It is implemented with a pressure relief valve and a fixed orifice on the pressure line downstream of the flow control orifice. The pressure relief valve is set to the desired maximum value. When the pump pressure builds up to this value, the pressure relief valve opens to lower the pressure and causes the directional valve to increase path P-A decrease path A-T. This shifts the control actuator to decrease its displacement until pressure returns to the preset value.

The model of the control unit is shown below:



The control actuator is modeled with the Double-Acting Actuator (IL) block. The flow control orifice is modeled with the Spool Orifice (IL) block with the rectangular slot geometry option. The opening area of this orifice is chosen to give the desired rated flow rate of 1.1e-3 m^3/s for the load-sensing function.

The combination of the 3-Way Directional Valve (IL) block and the Pilot Valve Actuator (IL) block represents the directional valve in the schematic in Figure 2. Ports X and Y of the pilot valve senses the pressure acting on the two sides of the directional valve and adjusts the spool position of the 3-Way Directional Valve (IL) block. The centering springs in the Pilot Valve Actuator (IL) block returns the spool to the neutral position. Path A-T of the 3-Way Directional Valve (IL) block is set to be half open in the neutral position to force the pump to increase its displacement at the start of the operation. Other parameters such as spring stiffness, valve stroke, valve orifice areas, etc., are tuned in the model for accuracy and stability.

The set pressure differential in the pressure relief valve is set to 25 MPa. Since the load-sensing function maintains a roughly 2 MPa pressure difference across the flow control orifice, this causes the pressure-limiting function to maintain a maximum pressure of 27 MPa at port A of the control unit.

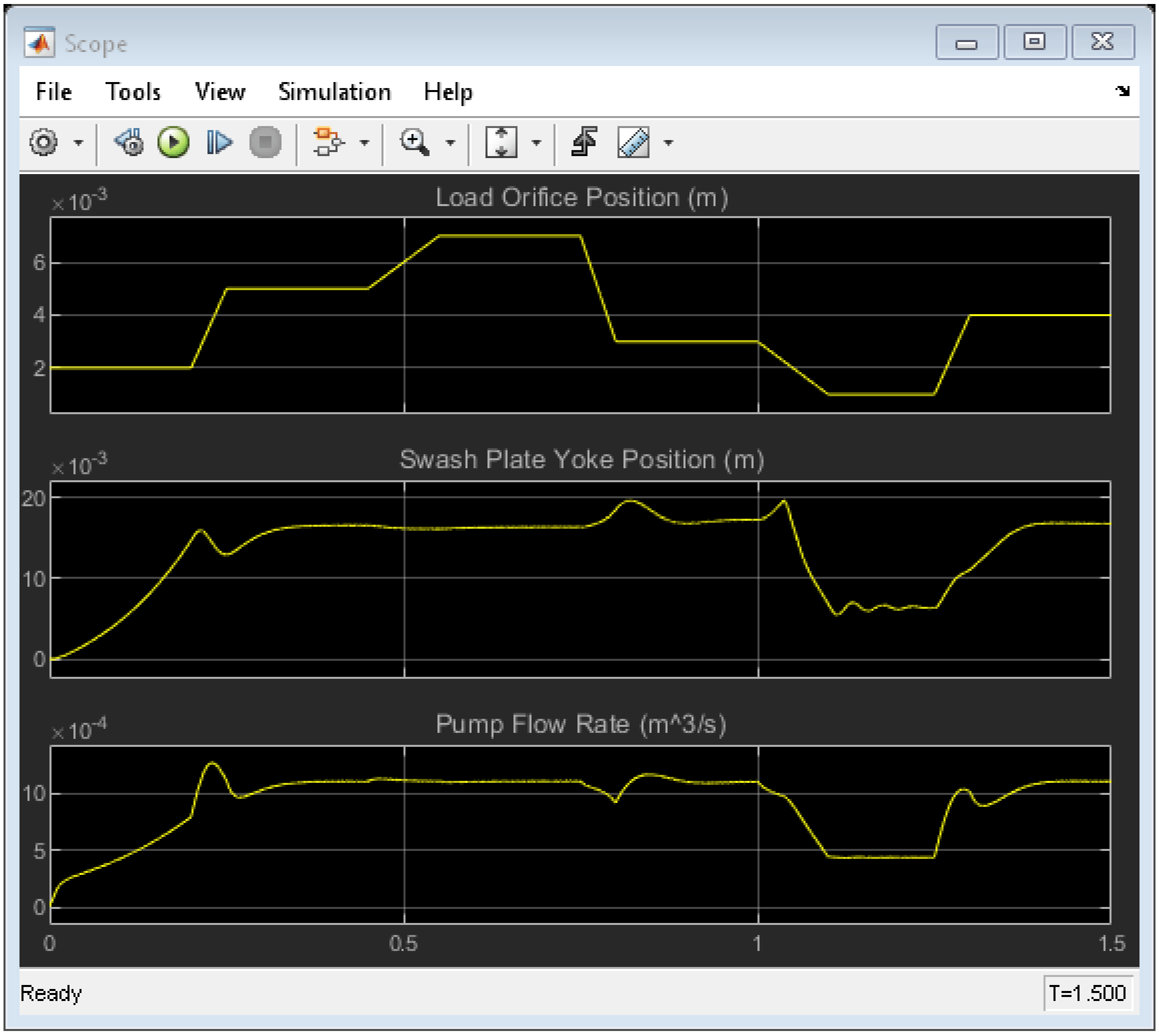
Test Rig Operation

The load of the test rig is represented by a Spool Orifice (IL) block. The simulation consists of six different loading conditions achieved by changing the spool position of the load orifice to 2 mm, 5 mm, 7 mm, 3 mm, 1 mm, and 4 mm. The control actuator initially increases the pump displacement to build up pressure. The process settles at around 0.35 s after the pressure difference across the flow control orifice becomes close to 2 MPa and the rated flow rate of 1.1e-3 m^3/s is achieved.

During the next three loading conditions, the control unit maintains the same pump delivery despite the varying load orifice opening.

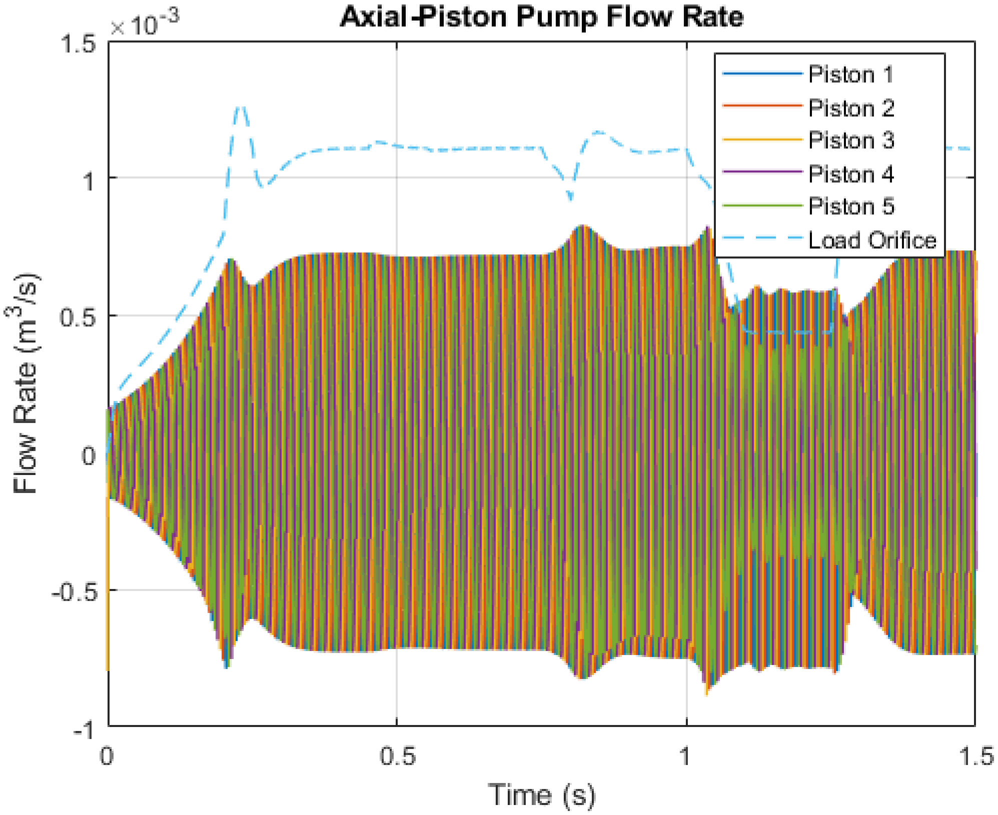
At 1 s, the load orifice is almost closed, causing pump pressure to rise. The pressure-limiting function becomes dominant as the pressure reaches 27 MPa. The control unit returns to the load-sensing mode after the pressure falls below 27 MPa.

Simulation Results from Scopes

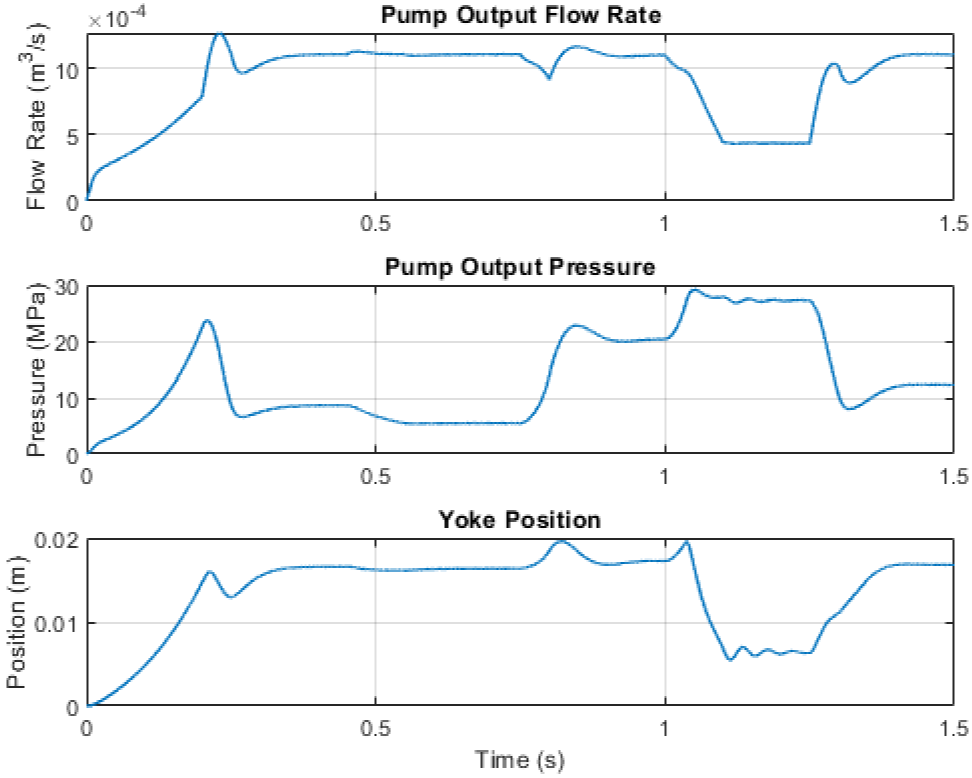


Simulation Results from Simscape Logging

This plot shows the flow rate within the axial-piston pump. The cyclic nature of the piston motion can be seen in the flow rate in each piston. The negative portion of the piston flow rate cycle corresponds to liquid being drawn in through valve plate A and the positive portion of the piston flow rate cycle corresponds to liquid being expelled through valve plate B. The combined effect of the five pistons results in the overall behavior of pump which produces a positive flow rate of about 1.1e-3 m^3/s through the load orifice.



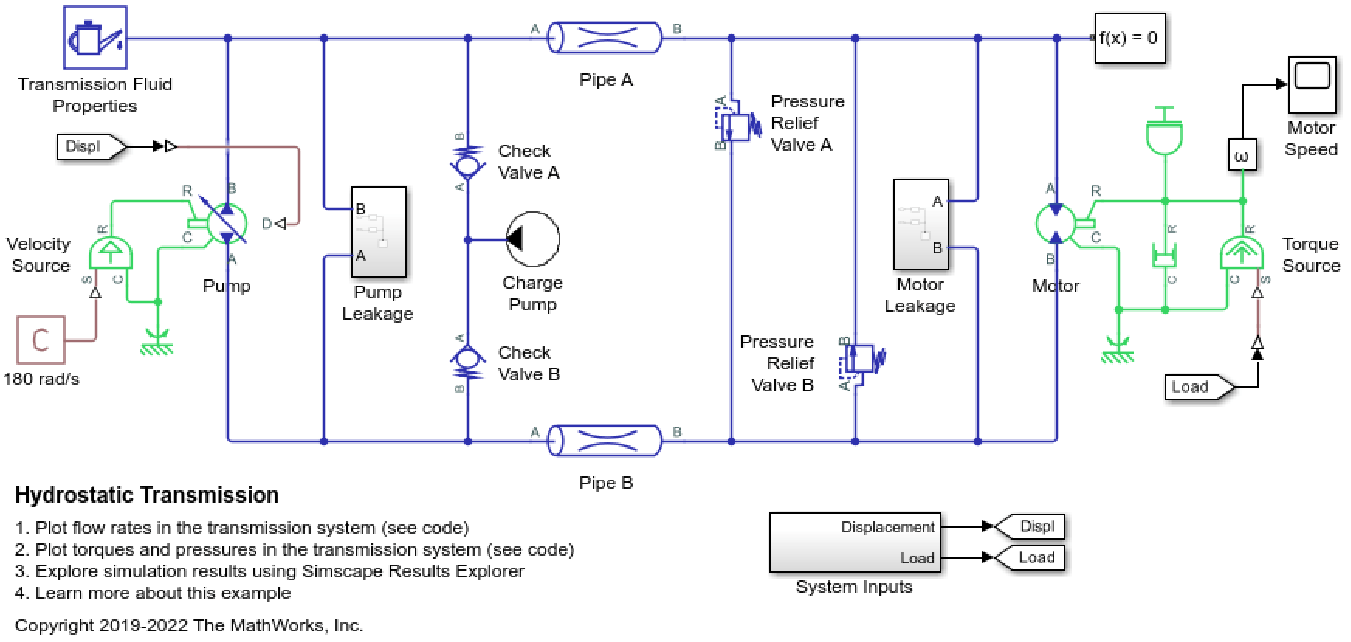
This plot shows the load-sensing and pressure-limiting control. The pump maintains its rated flow rate of 1.1e-3 m^3/s even as the load varies. However, as the pump output pressure rises to its maximum rated pressure of 27 MPa, the pressure-limiting control decreases the yoke position and the flow rate drops below its rated flow rate.


Pump Output Flow Rate
×10^-4
Flow Rate (m^3/s)
0
5
10
0
0.5
1
1.5
Pump Output Pressure
Pressure (MPa)
0
10
20
30
0
0.5
1
1.5
Yoke Position
Position (m)
0
0.01
0.02
0
0.5
1
1.5
Time (s)

Hydrostatic Transmission

This example shows a hydraulic transmission system built of a variable-displacement pump and a fixed-displacement motor. The pipes between the pump and the motor are connected by two replenishing valves (check valves) and a charge pump on the pump side and two pressure relief valves on the motor side. The motor drives a mechanical load consisting of an inertia, viscous friction, and time-varying torque. The system is tested with both positive and negative pump flow rate by varying the pump displacement.

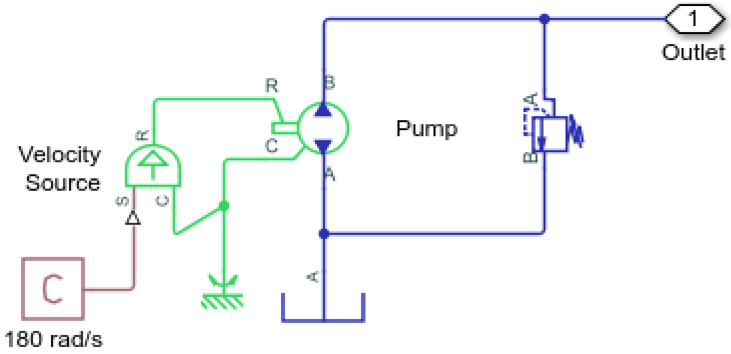
Model



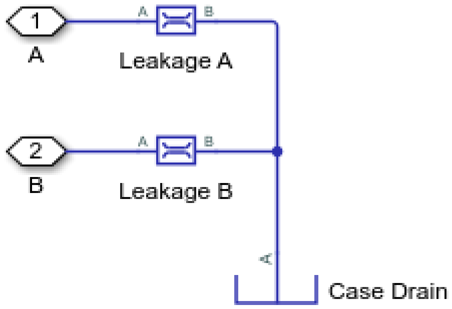
Hydrostatic Transmission

1. Plot flow rates in the transmission system (see code)
2. Plot torques and pressures in the transmission system (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

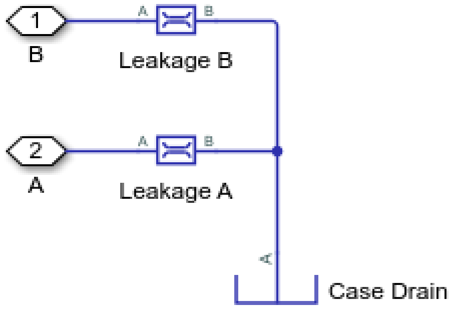
Charge Pump Subsystem



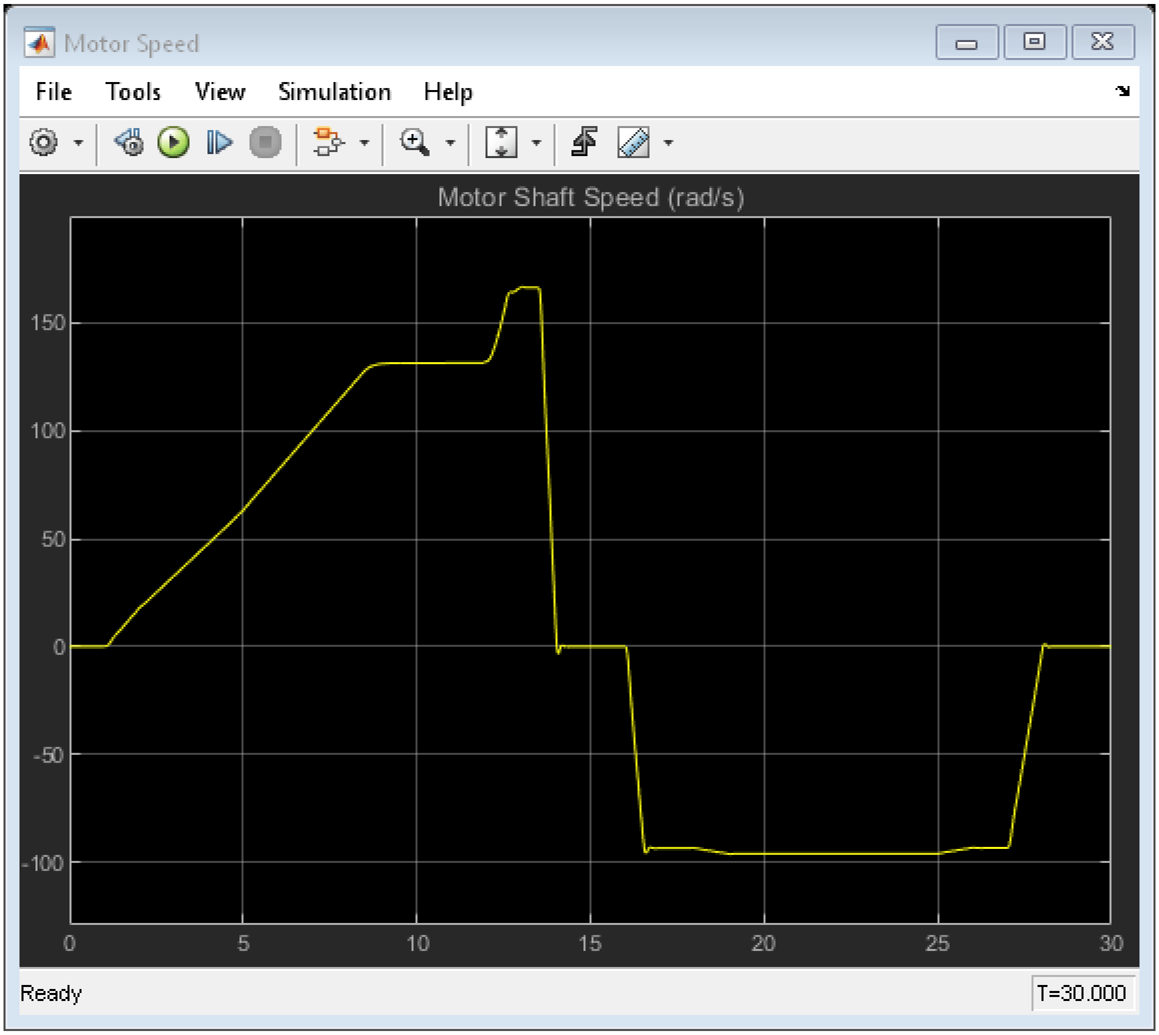
Motor Leakage Subsystem



Pump Leakage Subsystem

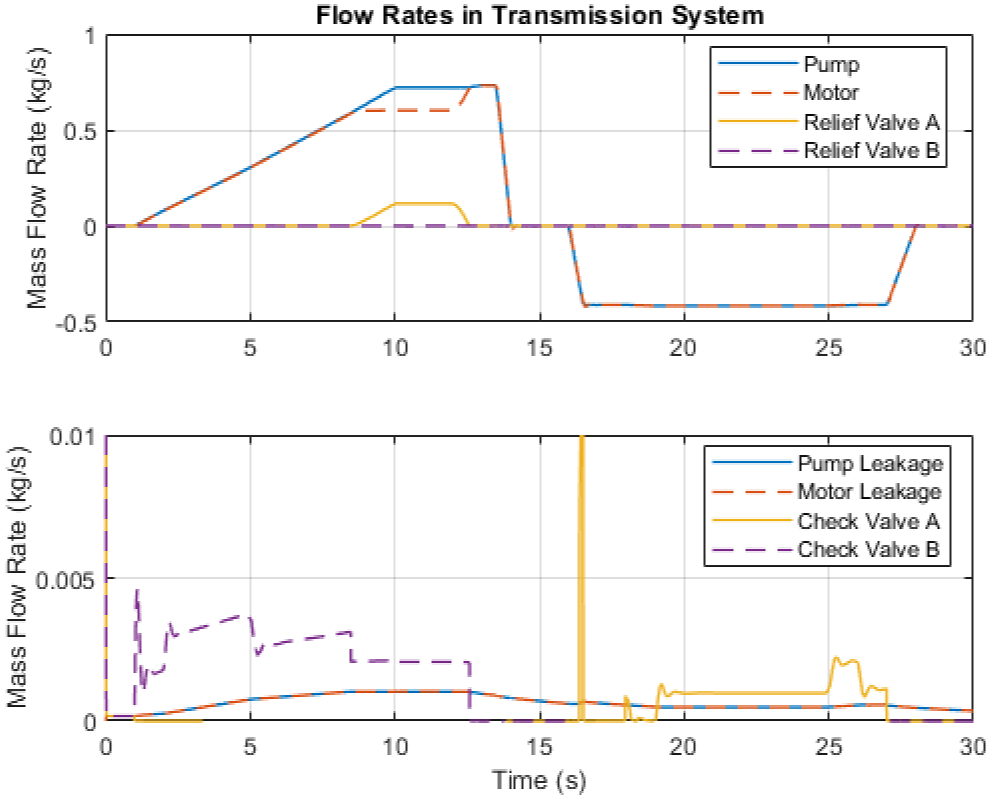


Simulation Results from Scopes

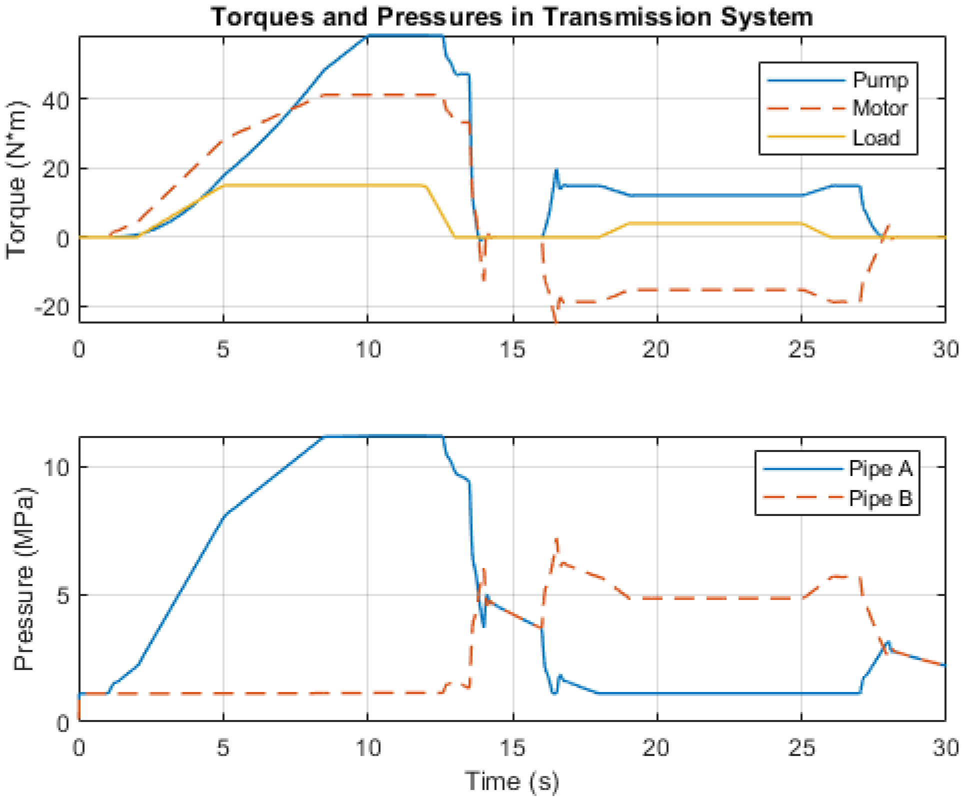


Simulation Results from Simscape Logging

This plot shows the mass flow rates through the pump, motor, and valves in the transmission system. Flow from the charge pump through the check valves compensate for external leakage through the pump and motor casing.



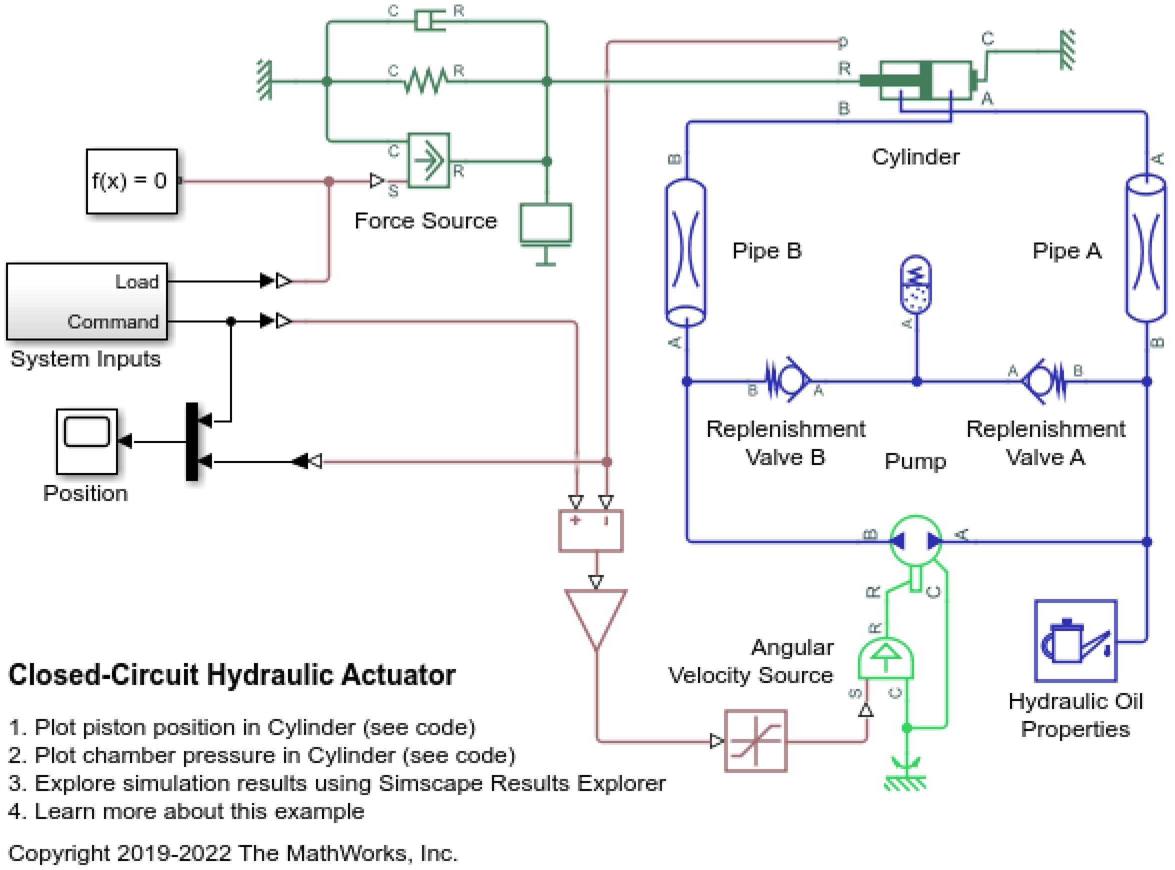
This plot shows the torque generated by the motor as well as the torque driving the pump and the mechanical load. It also shows the pressures in the two transmission pipes switch between high and low pressures as the pump is reversed.


Torques and Pressures in Transmission System
Torque (N*m)
40
20
0
-20
0
5
10
15
20
25
30
Pump
Motor
Load
Pressure (MPa)
10
5
0
0
5
10
15
20
25
30
Time (s)
Pipe A
Pipe B

Closed-Circuit Hydraulic Actuator

This example shows a closed-circuit hydraulic actuator driven by a variable-speed pump. The actuator is arranged as a closed fluid system with two replenishment valves (check valves) and a spring-loaded accumulator serving as a replenishment reservoir. The pump speed is controlled by the difference between the commanded and measured piston position. The actuator acts against a spring, a damper, and a time-varying load.

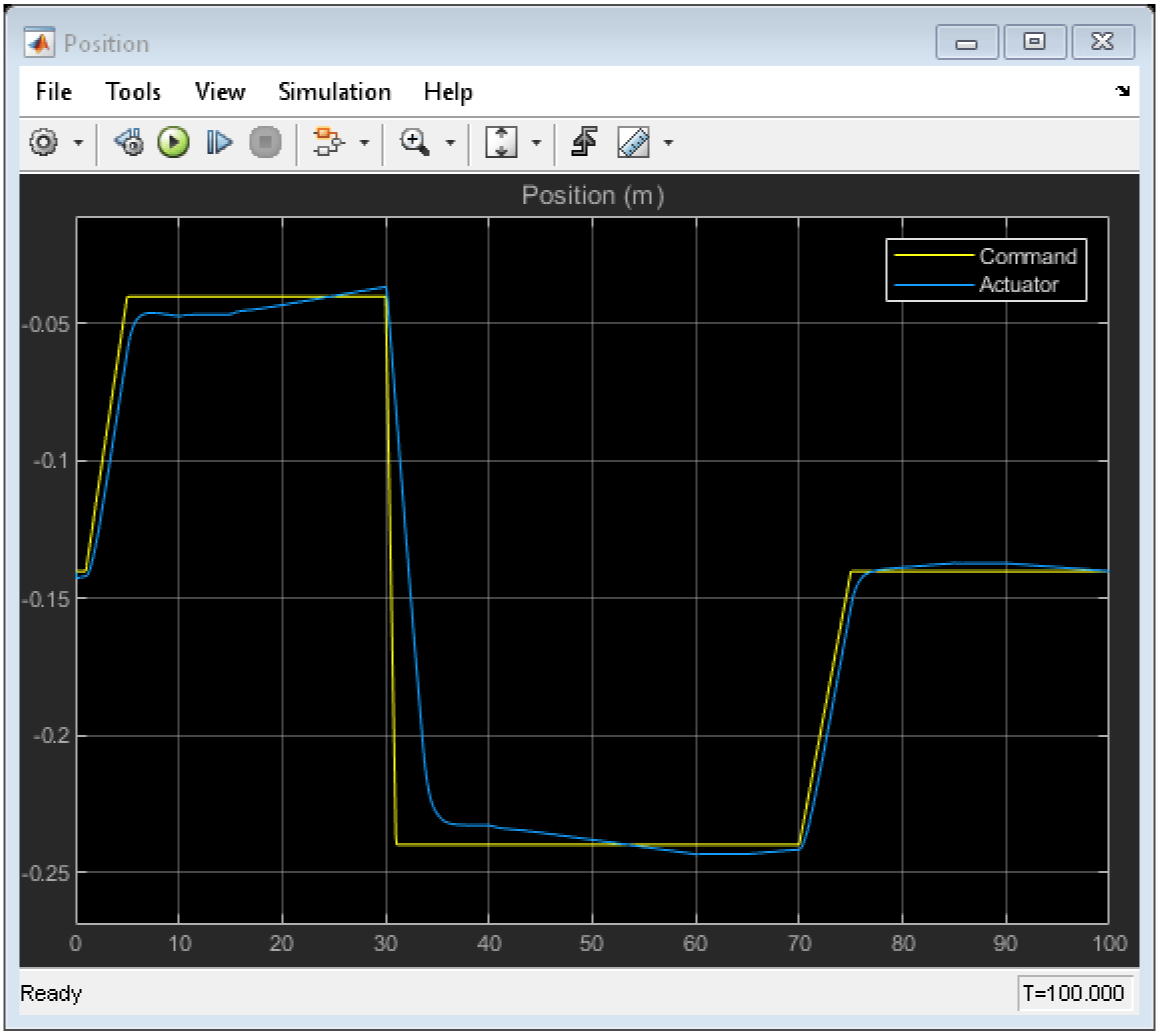
Model



Closed-Circuit Hydraulic Actuator

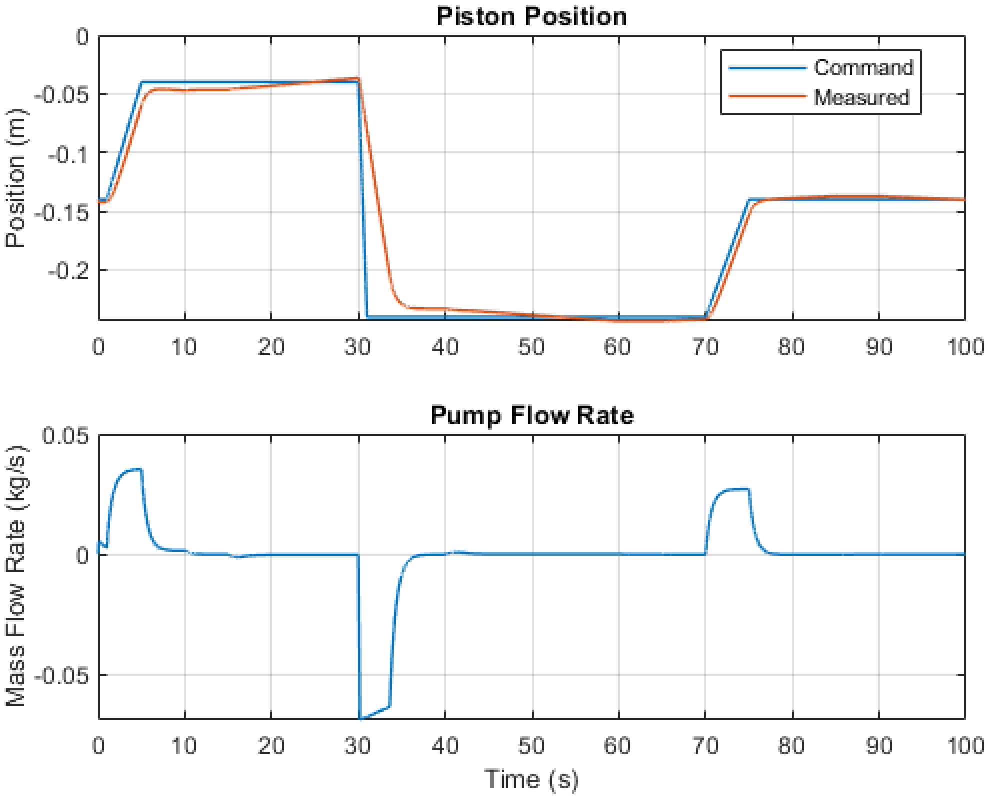
1. Plot piston position in Cylinder (see code)
2. Plot chamber pressure in Cylinder (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Simulation Results from Scopes

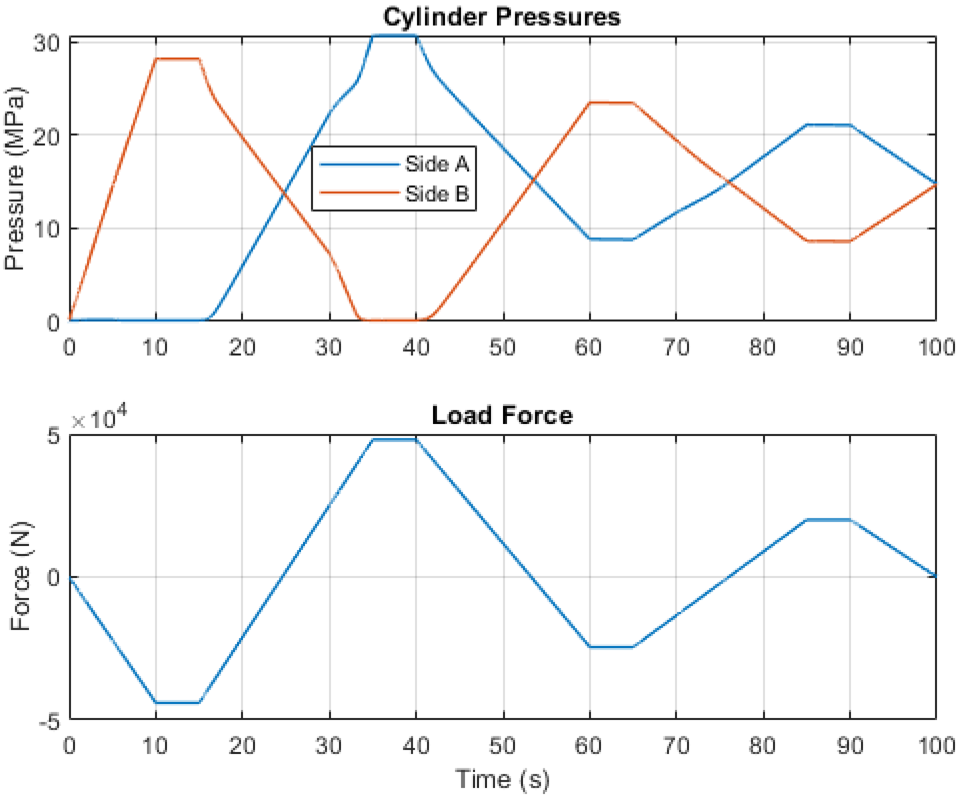


Simulation Results from Simscape Logging

This plot shows the position of the piston and the flow rate through the pump.



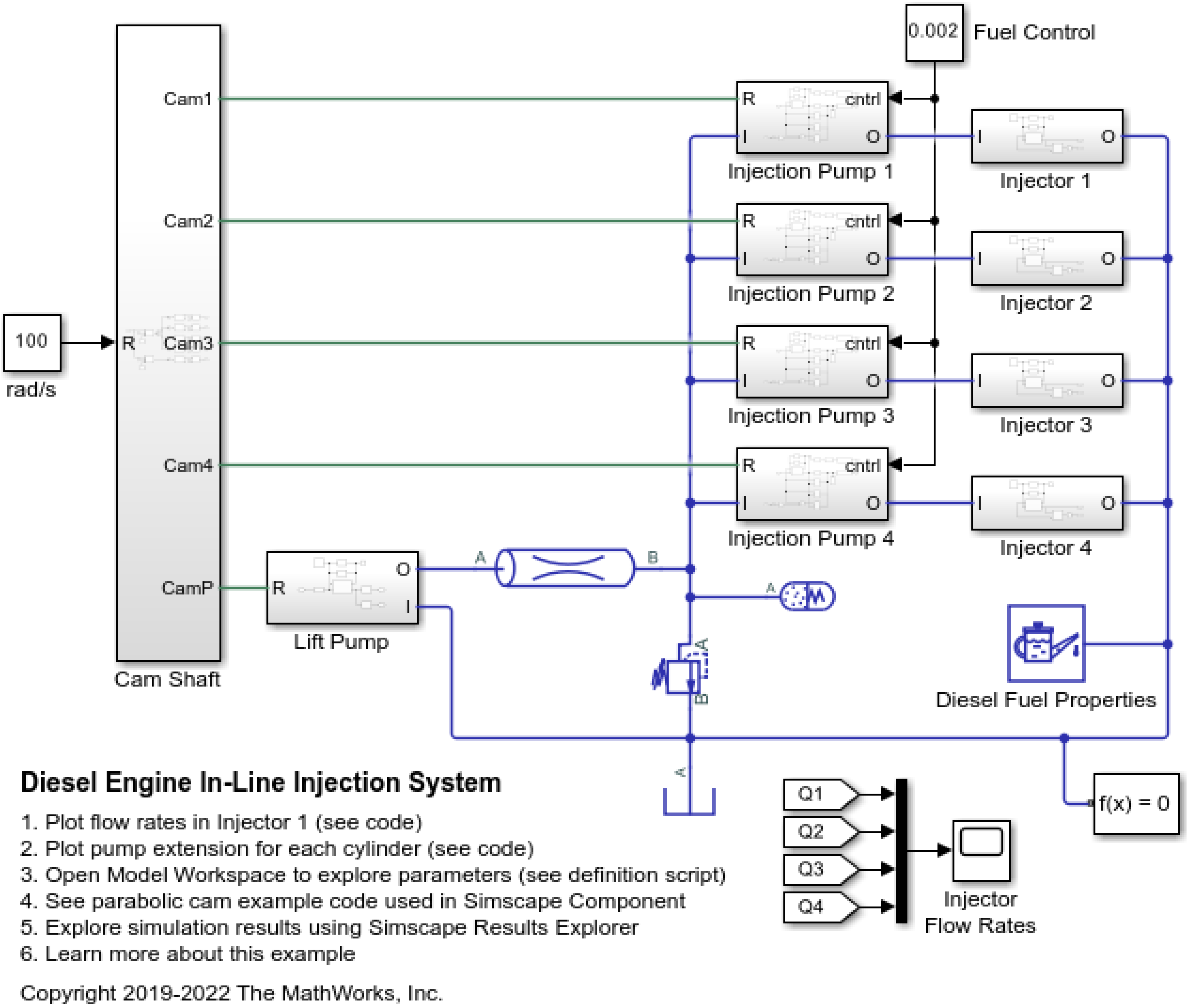
This plot shows the pressures inside the cylinder chambers and the time-varying load on the piston.


Cylinder Pressures
Pressure (MPa)
30
20
10
0
0
10
20
30
40
50
60
70
80
90
100
Side A
Side B
Load Force
×10^4
5
0
-5
Force (N)
0
10
20
30
40
50
60
70
80
90
100
Time (s)

Diesel Engine In-Line Injection System

This example shows an in-line multi-element diesel injection system. It consists of a cam shaft, a lift pump, four in-line injection pumps, and four injectors.

Model



Diesel Engine In-Line Injection System

1. Plot flow rates in Injector 1 (see code)
2. Plot pump extension for each cylinder (see code)
3. Open Model Workspace to explore parameters (see definition script)
4. See parabolic cam example code used in Simscape Component
5. Explore simulation results using Simscape Results Explorer
6. Learn more about this example

Copyright 2019-2022 The MathWorks, Inc.

Injection System Description

The diesel injection system simulated by this model is shown in the schematic below.

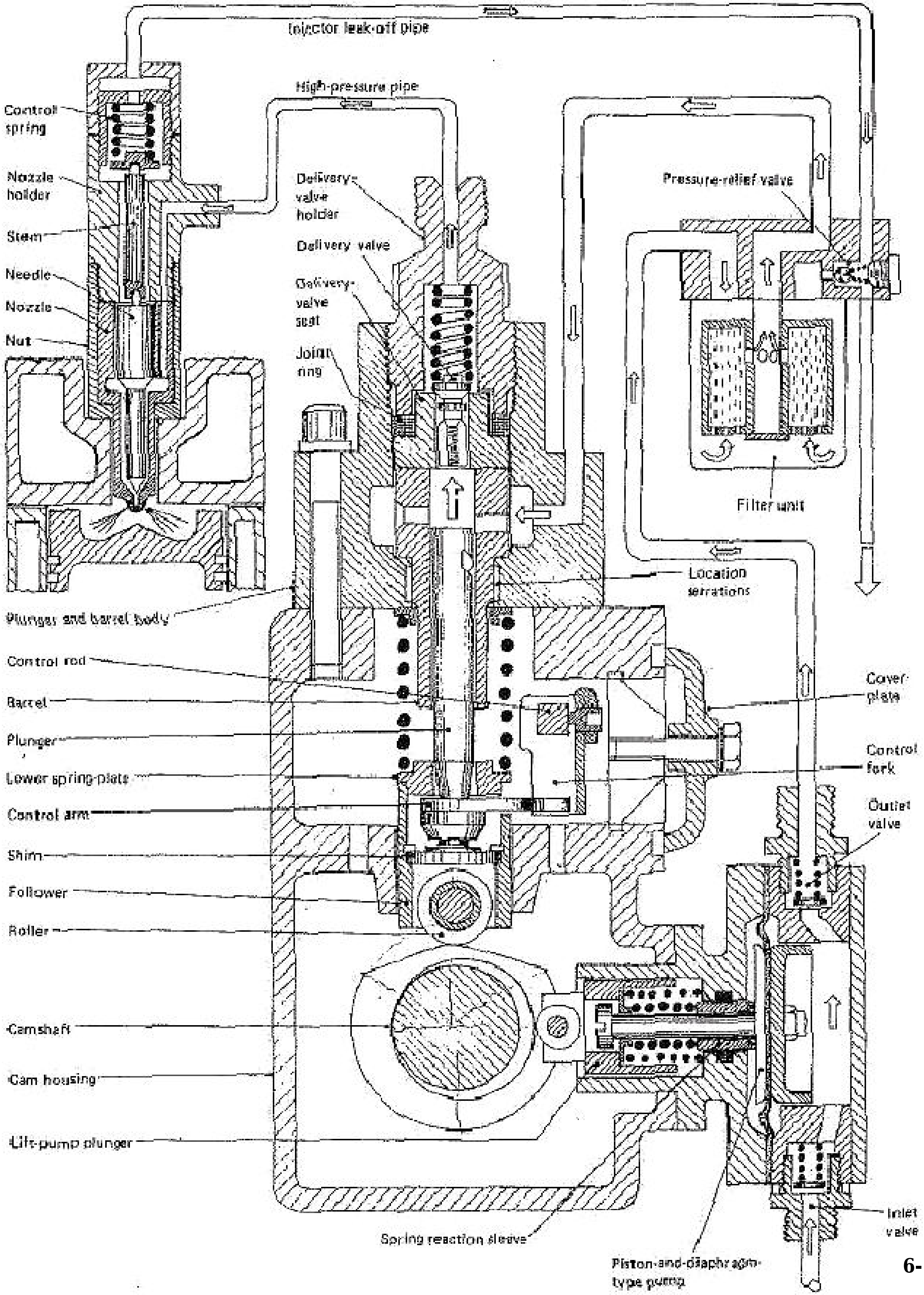

Injector leak-off pipe
High-pressure pipe
Control spring
Nozzle holder
Stem
Needle
Nozzle
Nut
Delivery-valve holder
Delivery valve
Delivery-valve seat
Joint ring
Pressure-relief valve
Filter unit
Location serrations
Plunger and barrel body
Control rod
Barrel
Plunger
Lower spring-plate
Control arm
Shim
Follower
Roller
Camshaft
Cam housing
Lift-pump plunger
Spring reaction sleeve
Piston-and-diaphragm-type pump
Cover-plate
Control fork
Outlet valve
Inlet valve

Figure 1: Injection System Schematic Diagram

The system structure is reproduced from H. Heisler, *Vehicle and Engine Technology* (second edition), 1999, and is categorized as an in-line multi-element injection system. It consists of the following basic units:

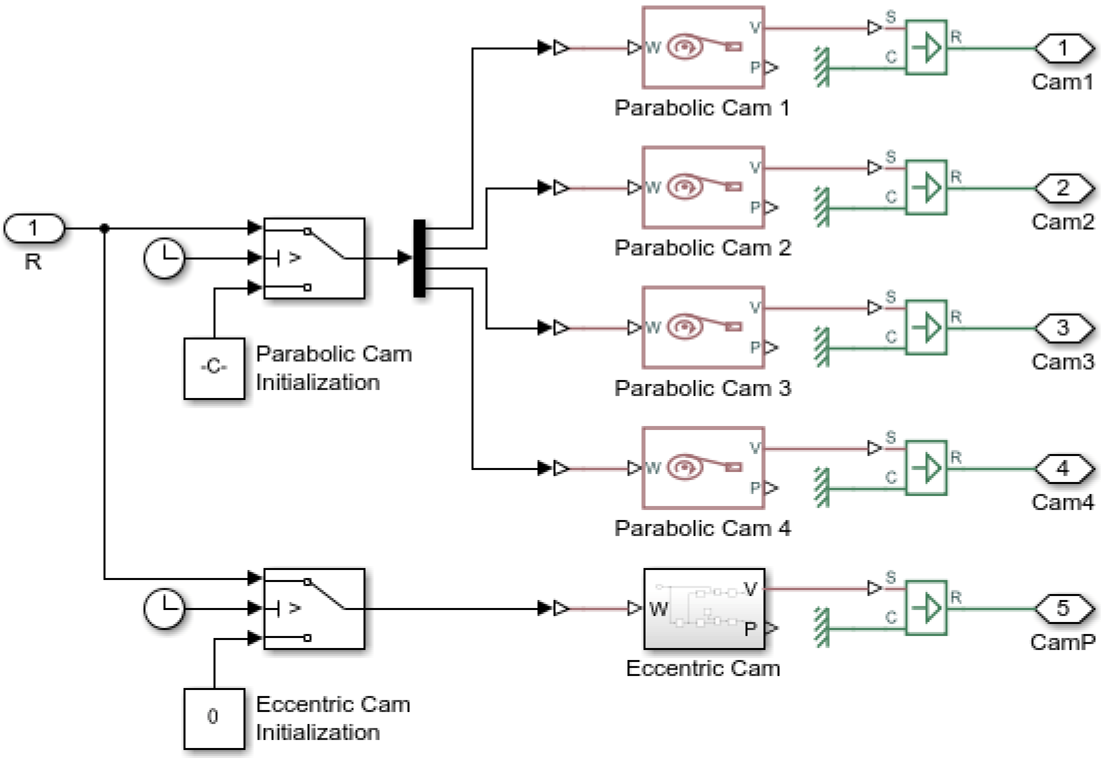
- Cam shaft
- Lift pump
- In-line injection pump with four pumping elements, one element per engine cylinder
- Injectors

The cam shaft carries five cams. The first one is the eccentric cam to actuate the lift pump. The remaining four drive the plungers of the injection pump. The cams are installed in such a way that pumping elements deliver fuel in the firing order and at the correct instant in the engine's cycle of operation. The lift pump supplies fuel to the intake of injection pump elements. Each element of the injection pump consists of a cam-driven plunger, a delivery valve, and the governor assembly. The purpose of the governor is to control the volume of the fuel delivered by the plunger to the engine cylinder.

The objective of the simulation is to investigate the entire system operation. The objective dictates the extent of idealization of each component in the system. All of the system units will be described in more detail in the following sections.

Note: The model of the system does not represent any particular injection system. All workspace parameters have been assigned based on practical considerations and do not represent any particular manufacturer parameters.

Cam Shaft Subsystem



The model of a cam shaft is built of five cam models. There are four parabolic profile cams and one eccentric cam. This motion profile is computed as a function of the shaft angle, obtained by integrating the shaft angular velocity. The cam that drives the plunger of the injection pump has a parabolic profile under which the follower moves back and forth at constant acceleration. It is described by the following set of equations:

$$x = 2s\left(\frac{\theta - \theta_E}{\alpha}\right)^2, \quad \text{if} \quad \theta_E \le \theta < \theta_E + \frac{\alpha}{2}$$

$$x = s\left[1 - 2\left(1 - \frac{\theta - \theta_E}{\alpha}\right)^2\right], \quad \text{if} \quad \theta_E + \frac{\alpha}{2} \le \theta < \theta_E + \alpha$$

$$x = s, \quad \text{if} \quad \theta_E + \alpha \le \theta < \theta_R$$

$$x = s\left[1 - 2\left(\frac{\theta - \theta_R}{\beta}\right)^2\right], \quad \text{if} \quad \theta_R \le \theta < \theta_R + \frac{\beta}{2}$$

$$x = 2s\left(1 - \frac{\theta - \theta_R}{\beta}\right)^2, \quad \text{if} \quad \theta_R + \frac{\beta}{2} \le \theta < \theta_R + \beta$$

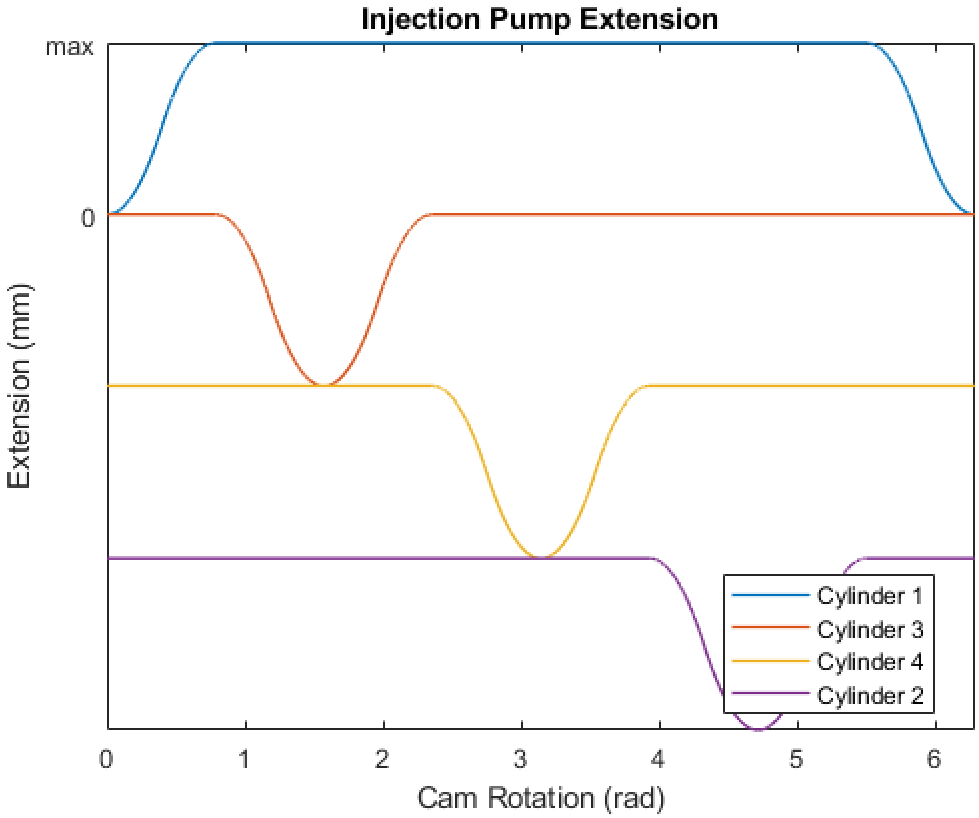
where

- x is the follower position
- s is the follower stroke
- θ is the cam angle (with phase angle accounted)
- θ_E is the start of extension angle
- α is the extension angle
- θ_R is the start of retraction angle
- β is the retraction angle

As a result, at the *start of extension angle* the follower starts moving up and reaches its top position after the shaft turns an additional *extension angle*. The follower starts the return stroke at the *start of retraction angle* and it takes *retraction angle* to complete this motion. The difference between the end of extension and the start of retraction sets the *dwell angle* at fully extended position. The profile is implemented in the Simscape language file `ParabolicCam.ssc` using the Simscape Component block.

Each of the four parabolic cams are offset from each other by a phase angle. The corresponding injection pump plunger positions are thus also offset from each other. To achieve this, for a brief period of `initialization_time` after the start of simulation, an appropriate initialization shaft speed is sent to each parabolic cam to rotate it to the correct phase angle. After the `initialization_time`, the main portion of the fuel injection system simulation begins.

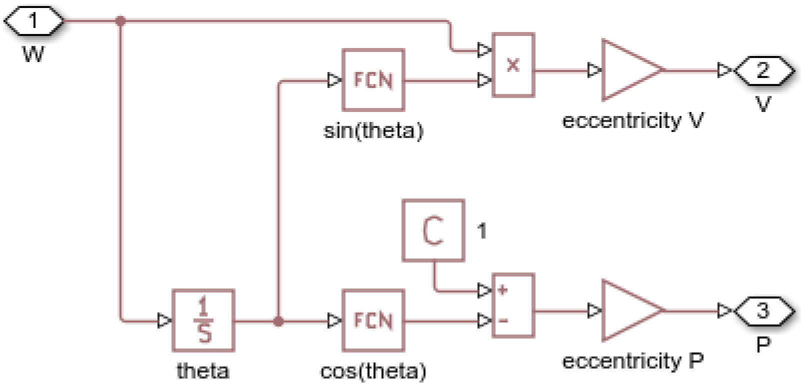
The following plot shows the extension of each injection pump as the cam shaft rotates. When the extension is zero, all of the fuel has been expelled from the pump. The firing order of the engine cylinders is 1-3-4-2.



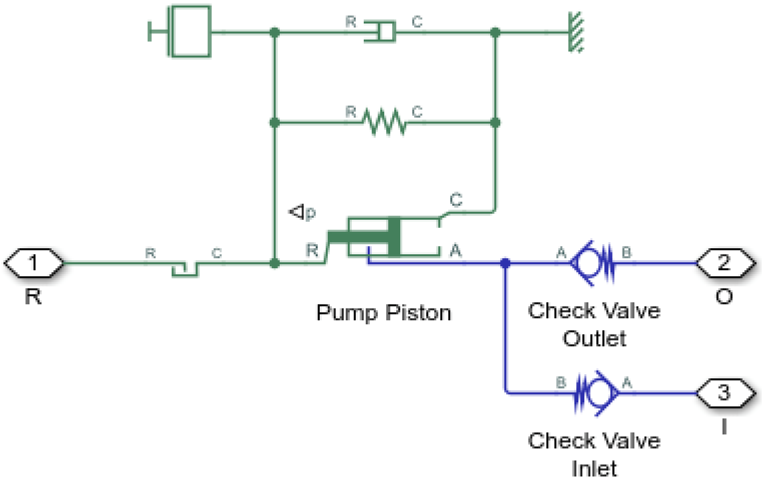
The profile of the eccentric cam is computed with the formula

$$x = \varepsilon\left(1 - \cos\theta\right)$$

where ε is the eccentricity. The equation is implemented using physical signal blocks in the Simscape Foundation Library.

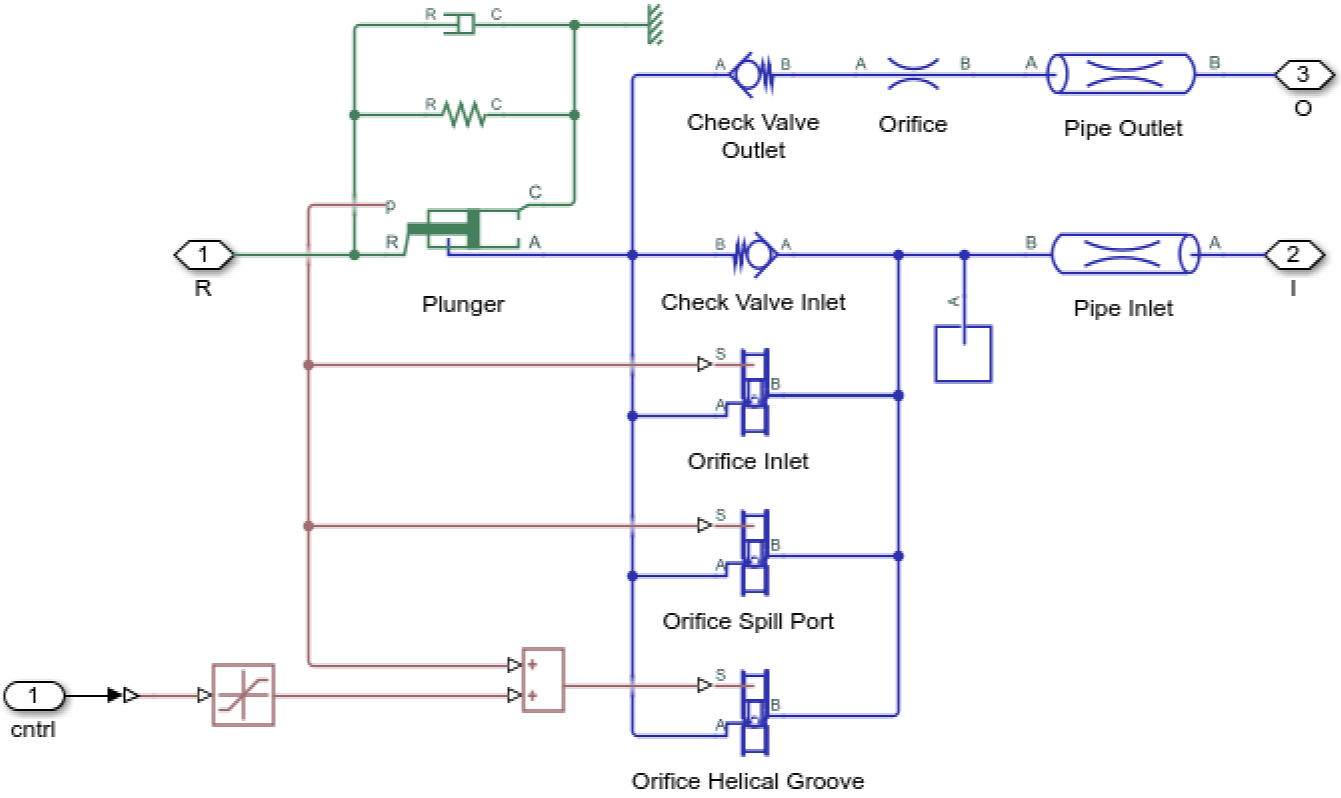


Lift Pump Subsystem



The model of the lift pump, which is a piston-and-diaphragm type pump, is built of a Single-Acting Actuator (IL) block and two Check Valve (IL) blocks. The check valves model inlet and outlet valves installed on both sides of the lift pump (see Figure 1). The contact between the pump rod roller and the cam is represented with the Translational Hard Stop block. The Translational Spring block models two springs in the pump that maintain permanent contact between the roller and the cam.

Injection Pump Subsystem



The in-line injection pump is a four-element pumping unit. Each element supplies fuel to injector which expels fuel to its engine cylinder. All four elements are identical by design. The Injection Pump subsystem represents the pump plunger and the pump control mechanism, while the Injector subsystem models the injector installed directly on the engine cylinder (see Figure 1).

The pump plunger oscillates inside the pump barrel driven by the cam (see Figure 1). The plunger is modeled as a Single-Acting Actuator (IL) block. When the plunger moves down, the plunger chamber is filled with fuel under pressure developed by the lift pump. The fuel enters through two orifices, the inlet port and the spill port (see Figure 2 below).

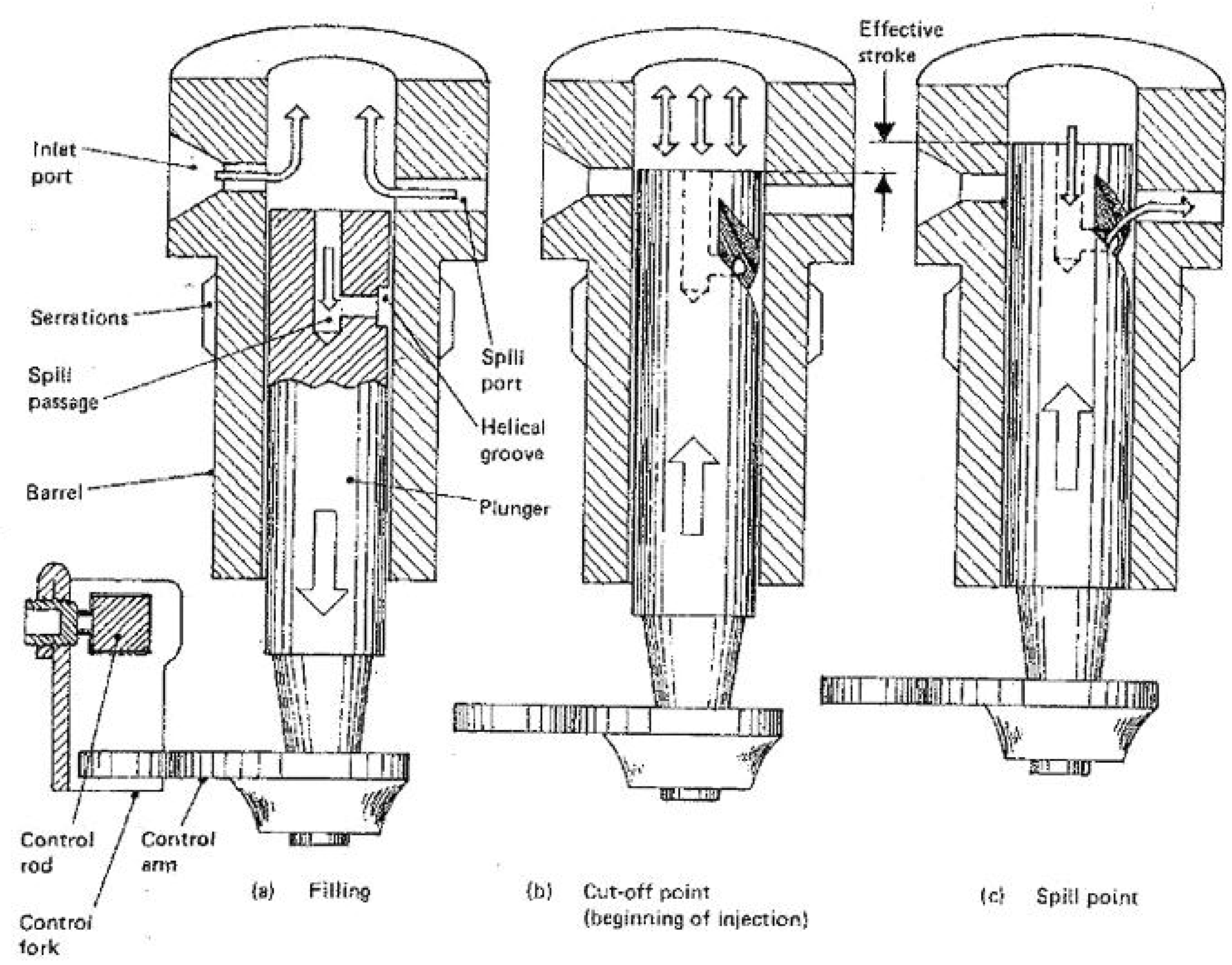


Figure 2: Plunger Interaction with Control Orifices in the Barrel

The model of the plunger control mechanism is based on the following assumptions:

1. There are three variable orifices modeled by Spool Orifice (IL) blocks in the control circuit: inlet port, spill port, and orifice formed by the helical groove and the spill port. The openings of the inlet and the spill orifices depend on the plunger motion, while the opening of the helical groove orifice is a function of the plunger motion and the plunger rotation. For the sake of simplicity, the displacement generated by the plunger rotation is represented as a source of linear motion which is combined with the plunger displacement.

2. The figure below shows all the dimensions necessary to parameterize the Spool Orifice (IL) blocks:

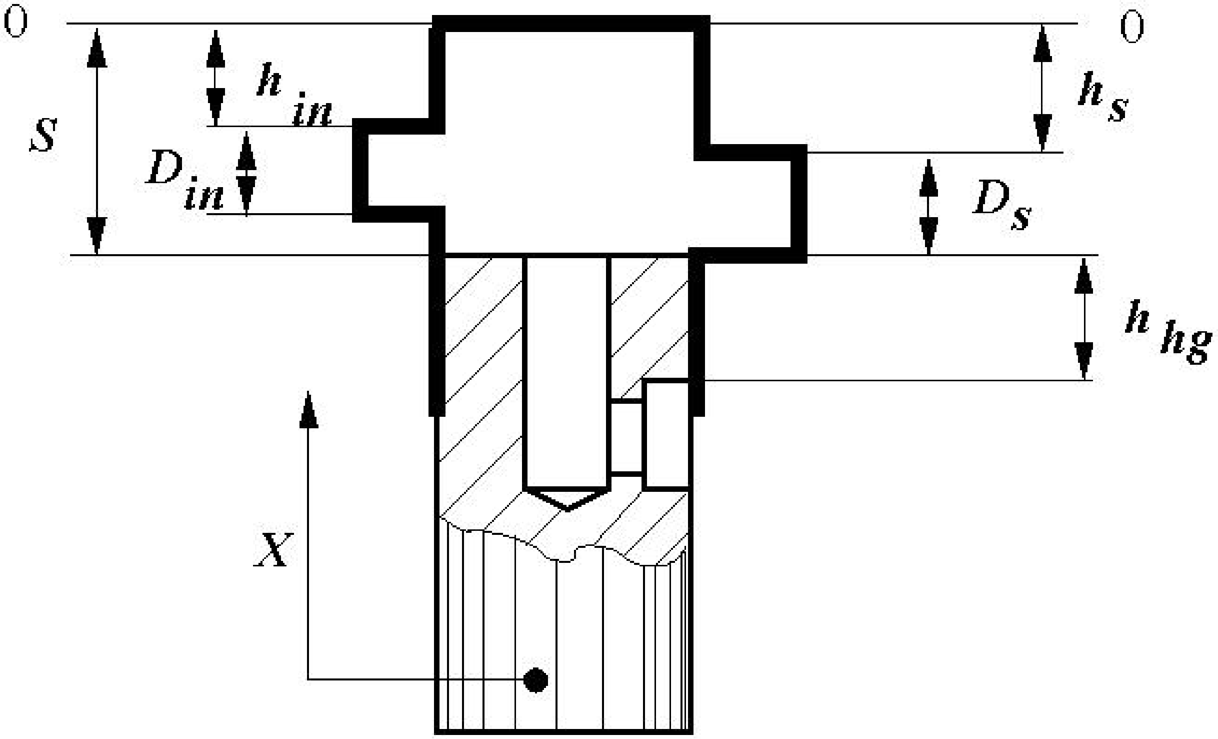


Figure 3: Schematic of injection plunger dimensions

where the schematic notations and corresponding workspace parameter names are

- D_{in} is the inlet port orifice diameter (`inlet_orifice_diameter`)
- D_s is the spill port orifice diameter (`spill_orifice_diameter`)
- S is the plunger stroke (`plunger_stroke`)
- h_{in} is the distance between the inlet orifice and the top plunger position (`-plunger_stroke + safety_gap + inlet_orifice_diameter + inlet_offset`)
- h_s is the distance between the spill port orifice and the top plunger position (`-plunger_stroke + safety_gap + spill_orifice_diameter`)
- h_{hg} is the distance between the spill port orifice and the upper edge of the helical groove

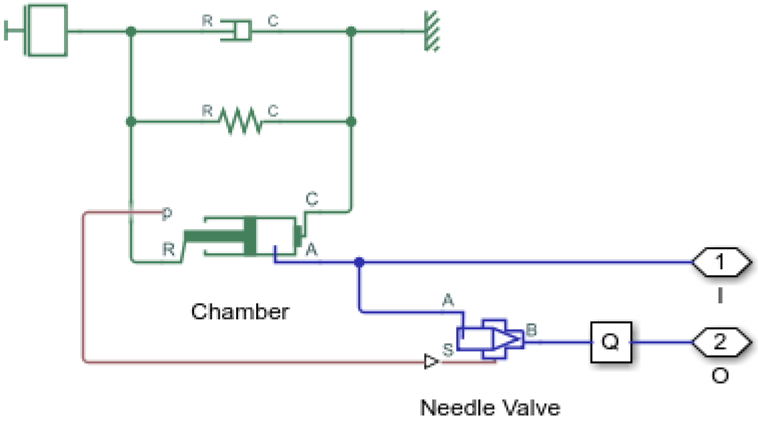
The inlet orifice is generally located higher than the spill orifice. This is represented by the workspace parameter `inlet_offset`.

3. In parameterizing the orifices, the plunger top position is taken as the origin and the motion in the upward direction is considered as the motion in positive direction. Therefore, the inlet and spill port orifices must be set to open with negative spool displacement, while the helical groove orifice must be set to open with positive spool displacement since it opens when the plunger moves upward.

4. The plunger effective stroke is `plunger_stroke - safety_gap - inlet_orifice_diameter - inlet_offset`. The distance between the spill port orifice and the upper edge of the helical

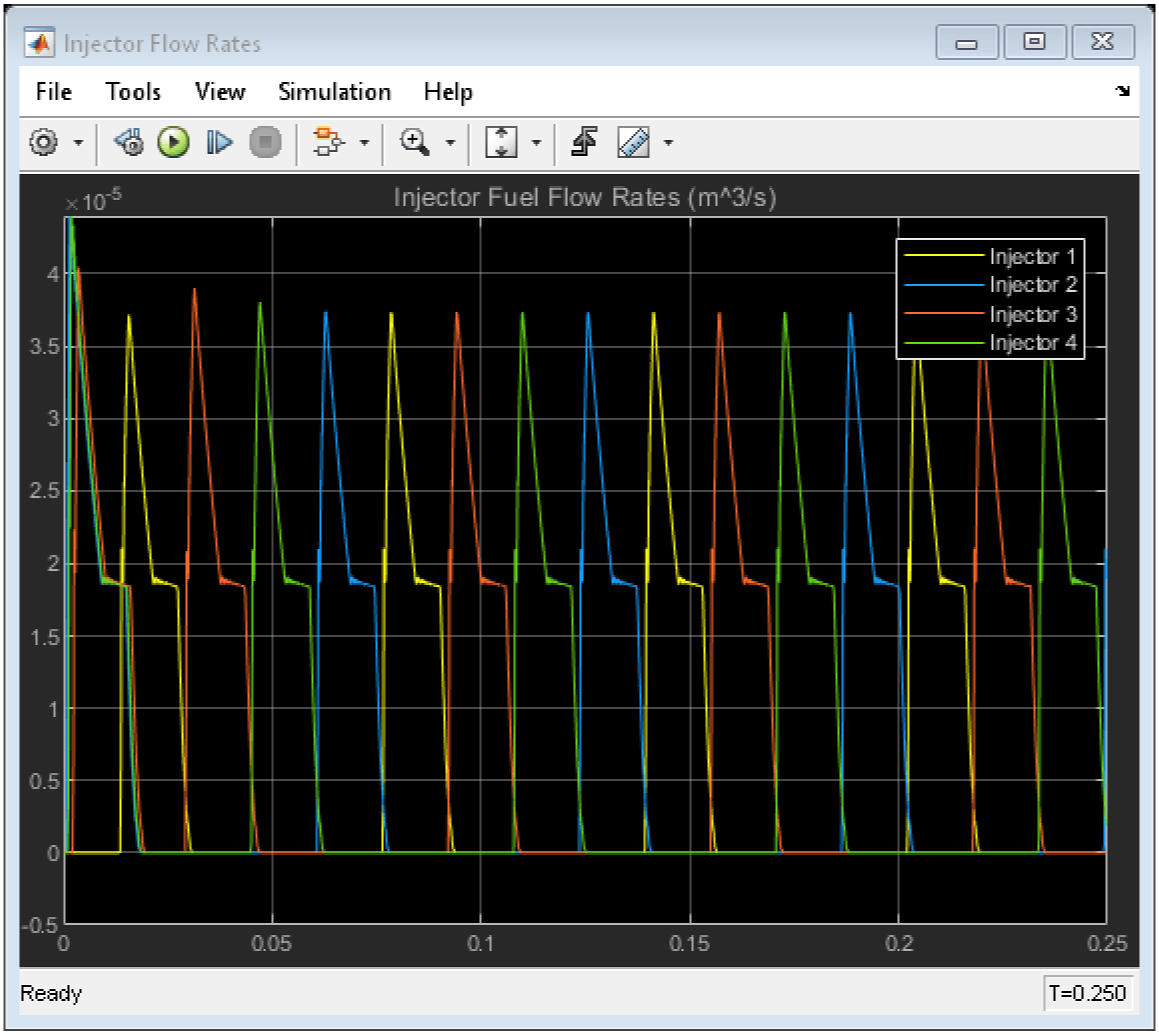
groove, h_{hg}, can be adjusted by rotating the plunger. This adjustment is modeled by shifting the plunger position signal to the helical groove orifice by a control signal. The larger the control signal, the sooner the helical groove orifice is opened, thus decreasing the volume of fuel delivered to the cylinder. The maximum value of the control signal is equal to the effective stroke. At this value, the helical groove orifice remains open all the time.

Injector Subsystem



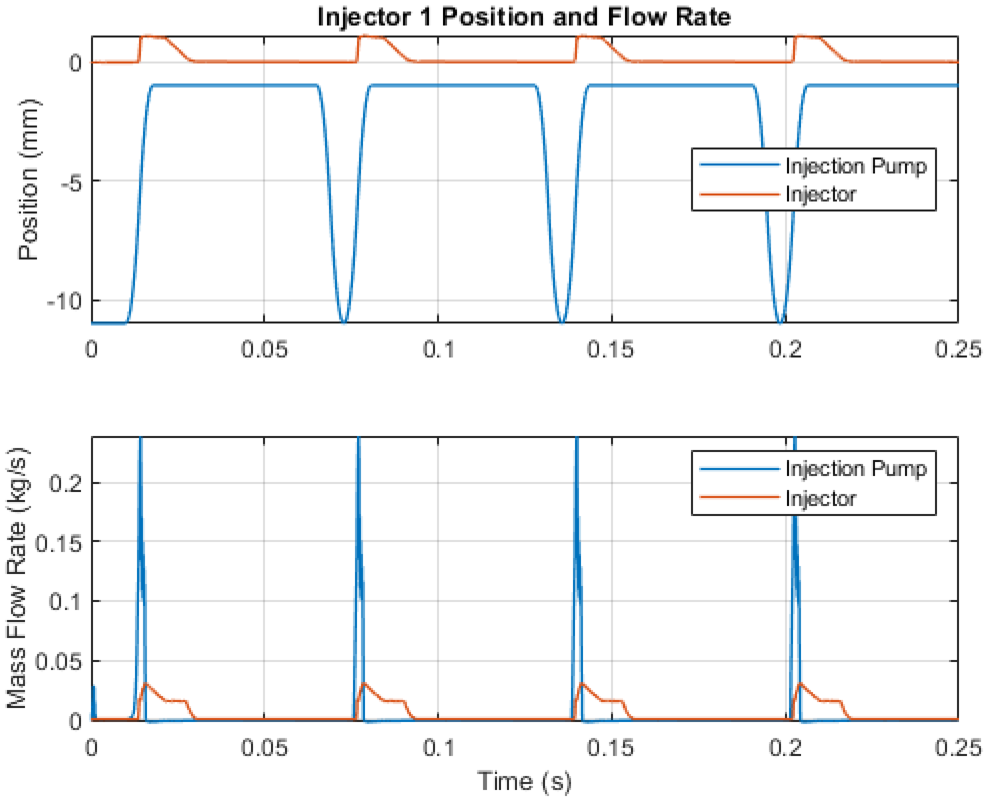
The model of the injector is based on the Single-Acting Actuator (IL) block and the Needle Valve (IL) block. The needle valve is normally closed by the force developed by the preloaded spring. As the force developed in the actuator chamber overcomes the spring force, the needle valve opens and allows fuel to be injected into the engine cylinder.

Simulation Results from Scopes



Simulation Results from Simscape Logging

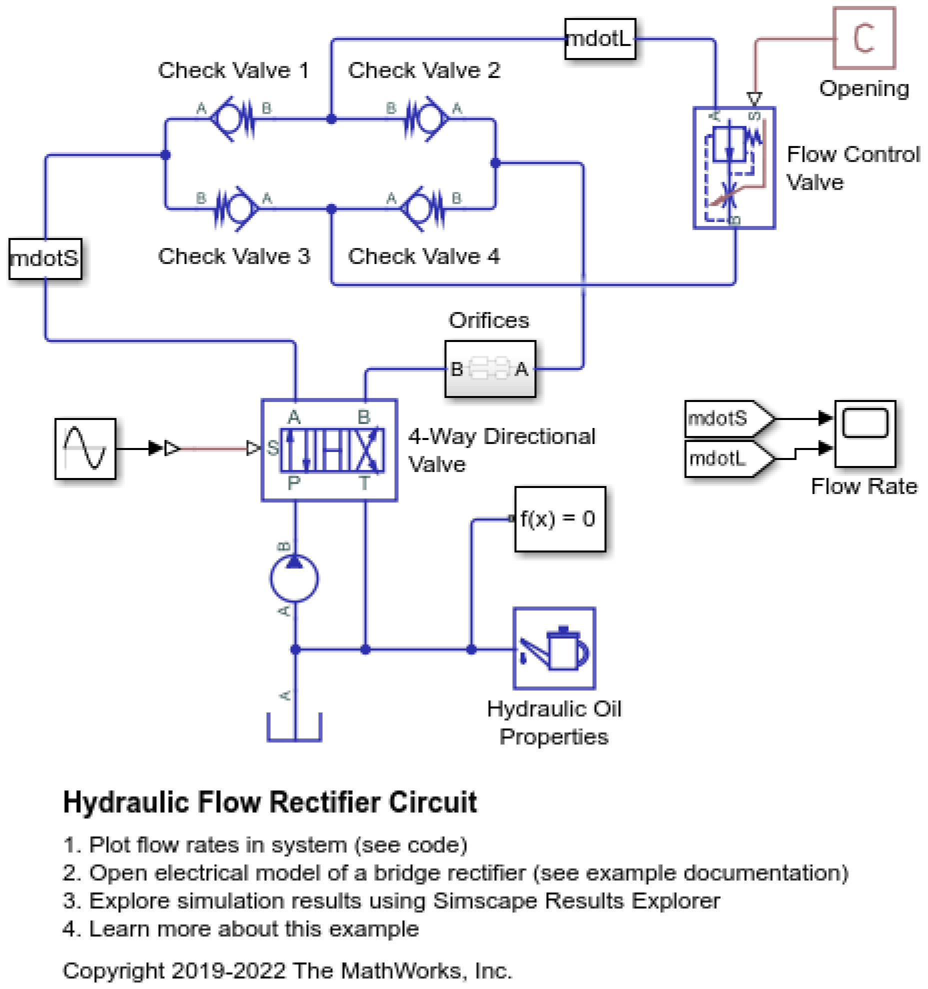
This plot shows the positions and outlet flow rates of Injection Pump 1 and Injector 1. The effect of the cam profile is shown in the displacement of Injection Pump 1. During the second half of the cam stroke, fuel exits the injection pump and passes into the injector. The fuel exits the injector via a needle valve. The injector is modeled as an actuator chamber with a preloaded spring which acts to store the fuel from the injection pump temporarily and push it out of the injector more smoothly.


Injector 1 Position and Flow Rate
Position (mm)
0
-5
-10
0
0.05
0.1
0.15
0.2
0.25
Injection Pump
Injector
Mass Flow Rate (kg/s)
0.2
0.15
0.1
0.05
0
0
0.05
0.1
0.15
0.2
0.25
Time (s)
Injection Pump
Injector

Hydraulic Flow Rectifier Circuit

This example shows a flow rectifier circuit with four check valves and a flow control valve. It is used to allow a single flow control valve to control fluid flow in both directions. Similar to a Graetz circuit implemented with diodes, the check valves are arranged in such a way that flow always passes through the flow control valve in the same direction. In the Orifices subsystem, there are two more check valves that are used to select the orifice that the flow passes through depending on the flow direction.

Model

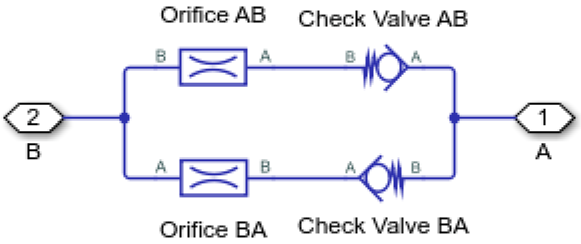


Hydraulic Flow Rectifier Circuit

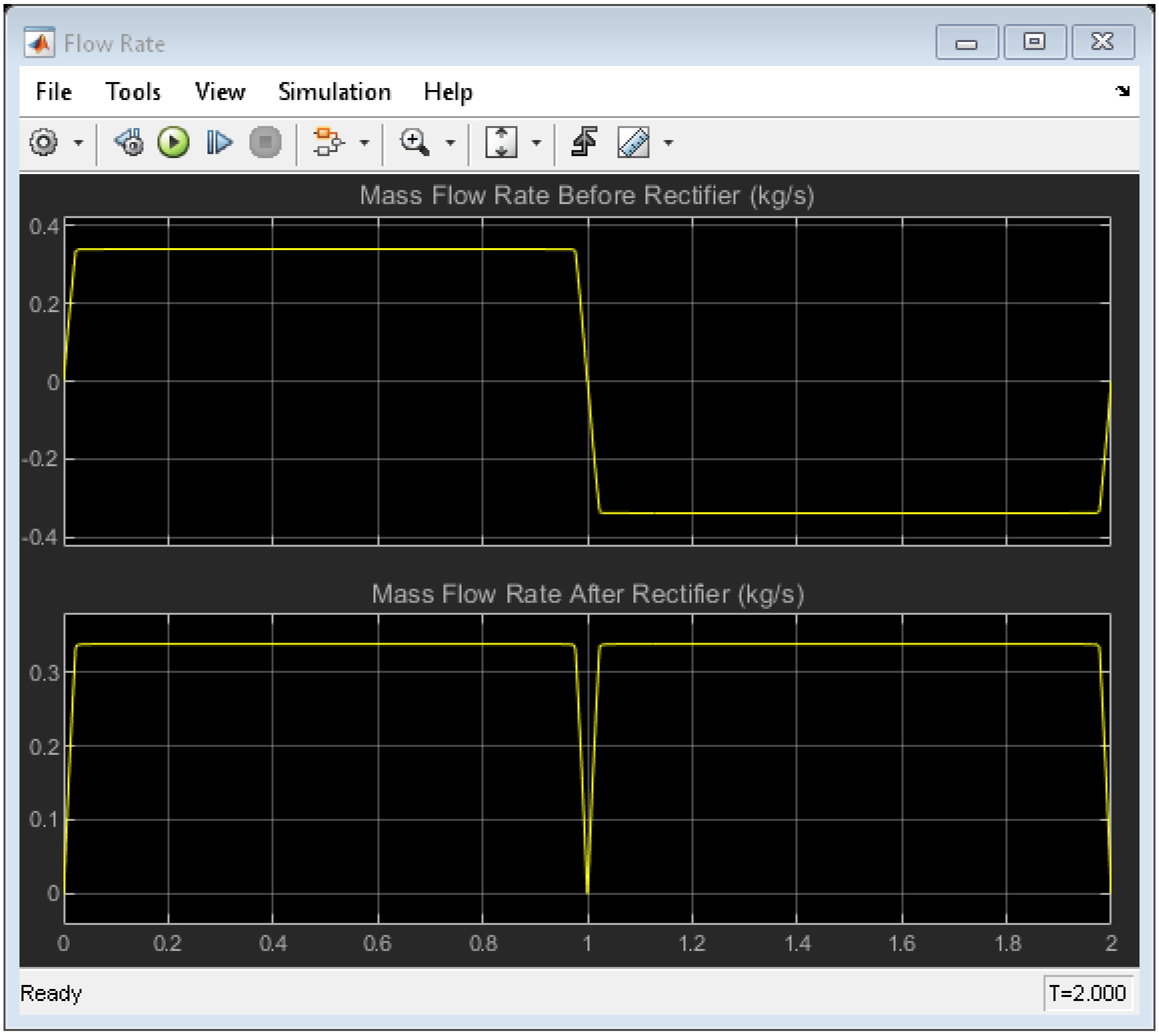
1. Plot flow rates in system (see code)
2. Open electrical model of a bridge rectifier (see example documentation)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2019-2022 The MathWorks, Inc.

Orifices Subsystem

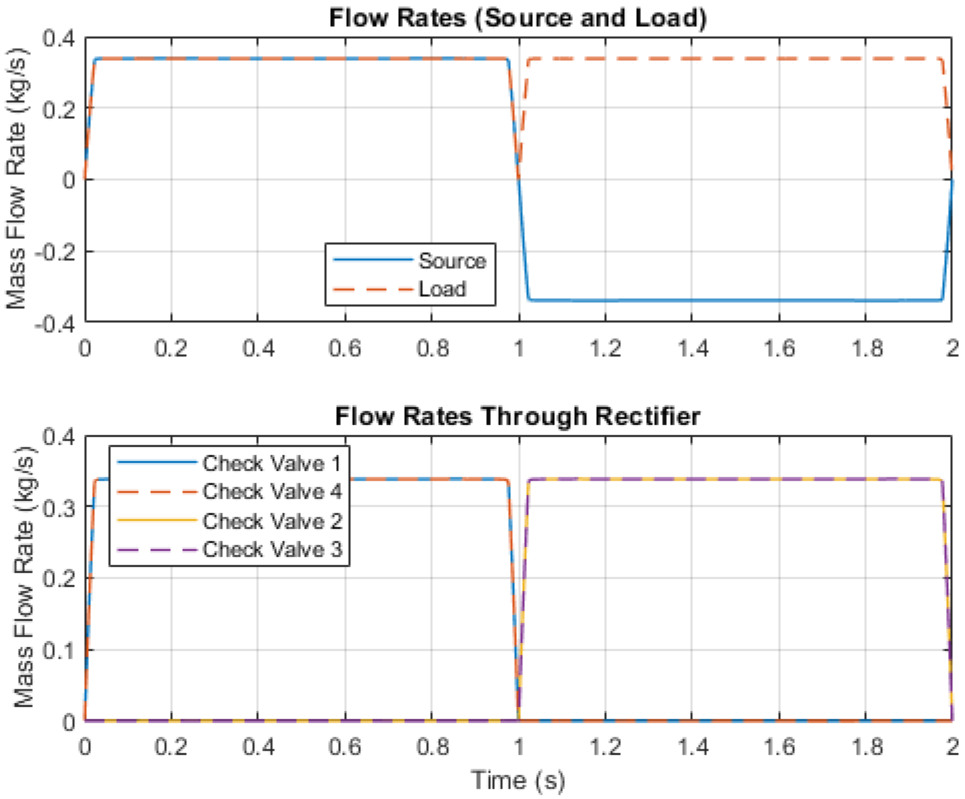


Simulation Results from Scopes



Simulation Results from Simscape Logging

This plot shows the mass flow rates through various components in the system. The flow rate from the source changes direction due to the 4-way directional valve. However, the flow rate to the load remains in the same direction due to the rectifier. When the flow rate from the source is positive, flow passes through check valves 1 and 4 and is blocked by check valves 2 and 3. When the flow rate from the source is negative, flow passes through check valves 2 and 3 and is blocked by check valves 1 and 4.



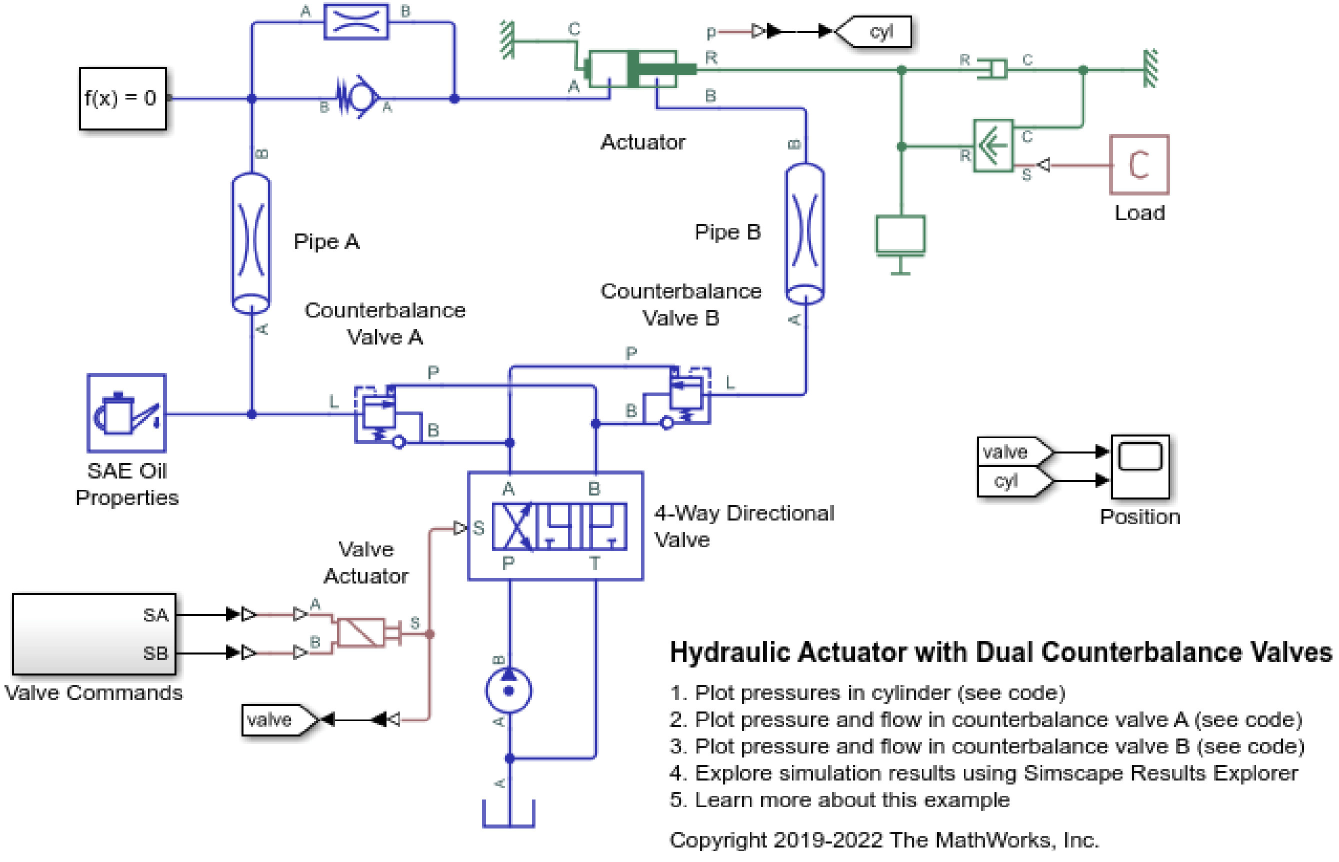
Hydraulic Actuator with Dual Counterbalance Valves

This example shows an actuator controlled by a 4-way directional valve and loaded with an overriding load, requiring the use of counterbalance valves to prevent the load from creeping when the directional valve is in the neutral position. In the neutral position, the directional valve connects ports A and B to the reservoir while blocking the pressure port P. The counterbalance valves block flow from returning to the reservoir, thus holding the actuator in place.

The valve actuator represents the electromagnets that control the directional valve position. When electromagnet B is energized, the directional valve is in the negative position and connects ports A and B to the pressure port P. This causes the actuator to extend due to the difference in the piston areas. Flow passes through the check valve in counterbalance valve A to enter cylinder chamber A, while pressure in the pilot line opens the orifice in counterbalance valve B to permit return flow from cylinder chamber B.

When electromagnet A is energized, the directional valve is in the positive position and connects port B to the pressure port P and port A to the reservoir. This causes the actuator to retract. Flow passes through the check valve in counterbalance valve B to enter cylinder chamber B, while pressure in the pilot line opens the orifice in counterbalance valve A to permit return flow from cylinder chamber A.

Model

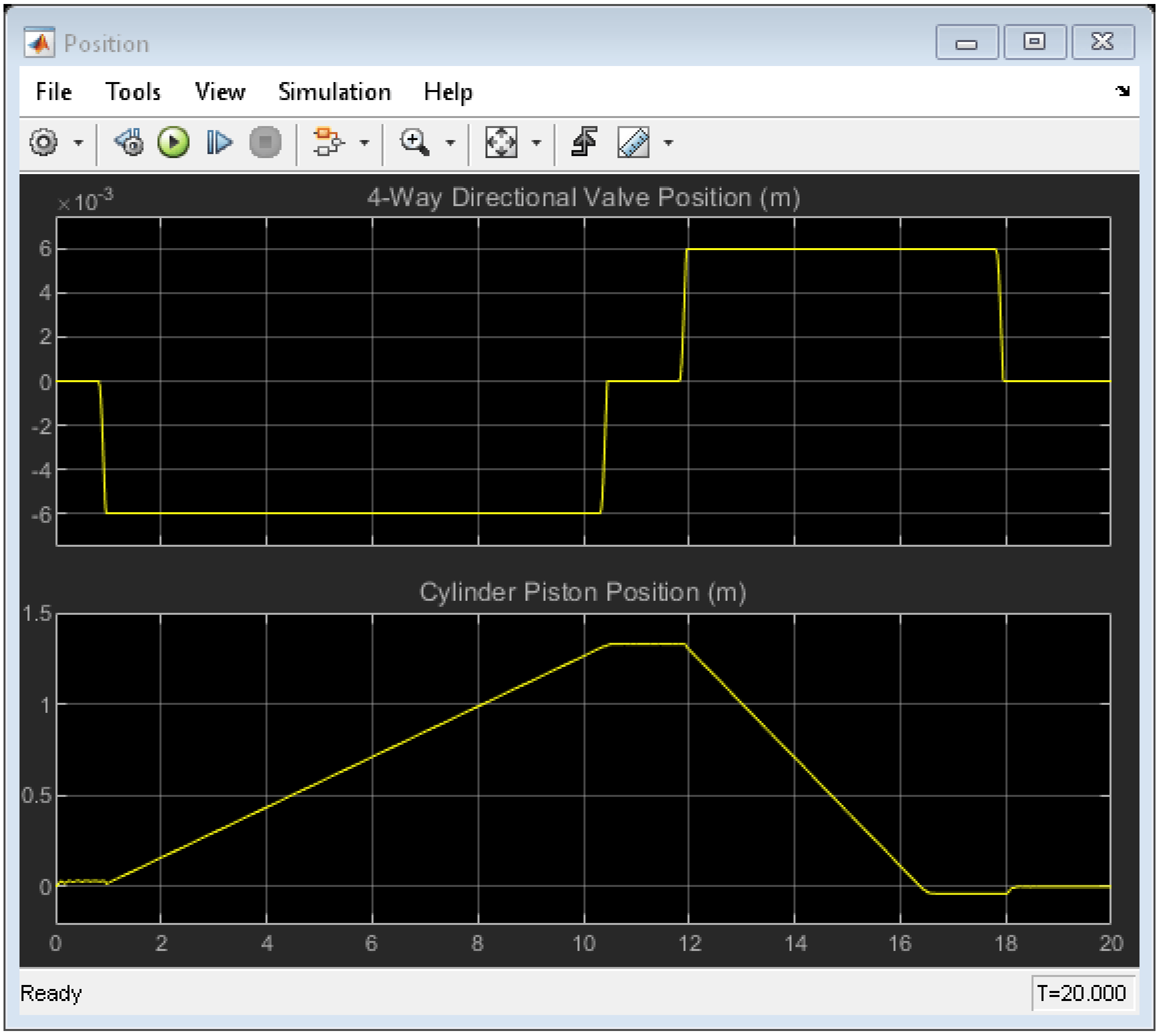


Hydraulic Actuator with Dual Counterbalance Valves

1. Plot pressures in cylinder (see code)
2. Plot pressure and flow in counterbalance valve A (see code)
3. Plot pressure and flow in counterbalance valve B (see code)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example

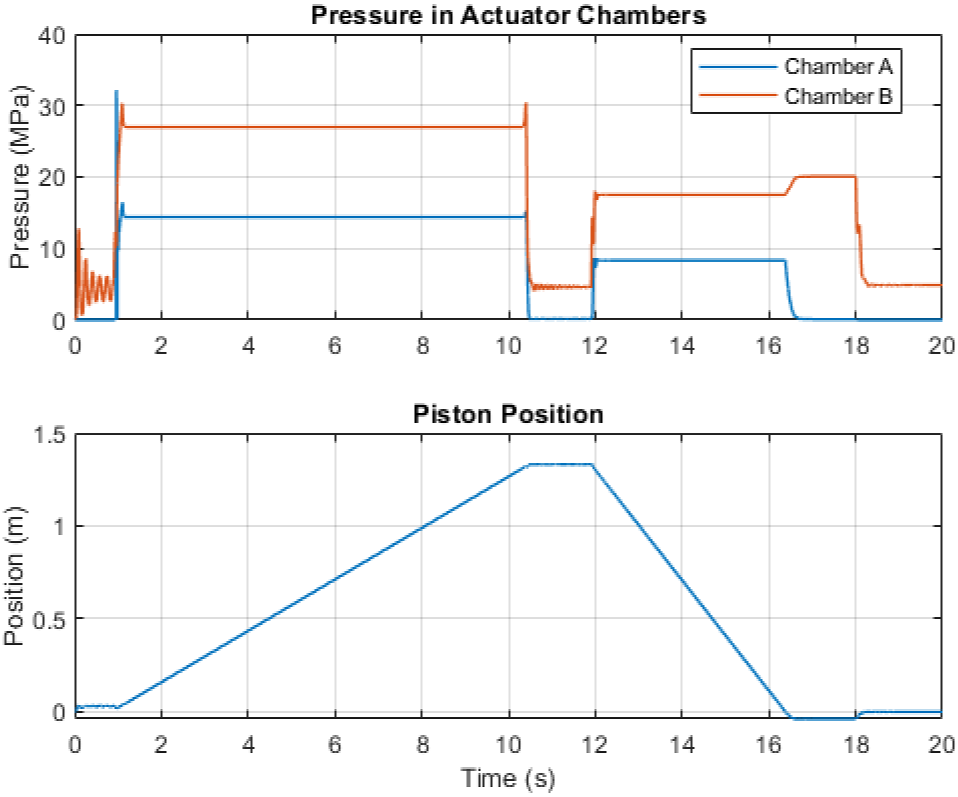
Copyright 2019-2022 The MathWorks, Inc.

Simulation Results from Scopes

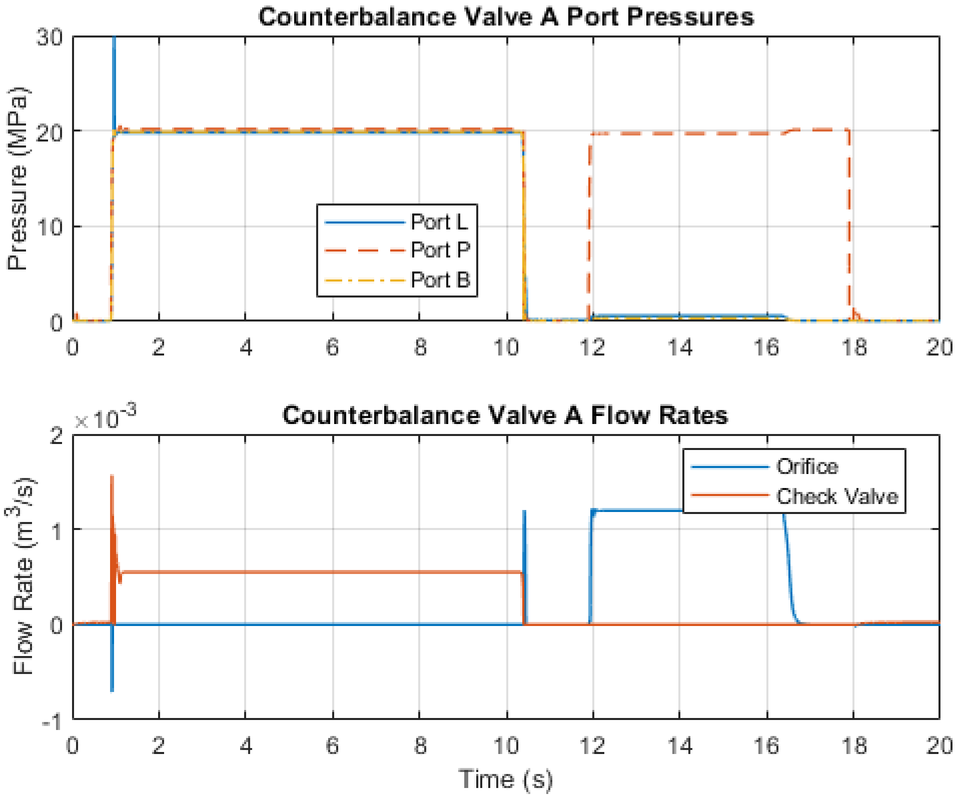


Simulation Results from Simscape Logging

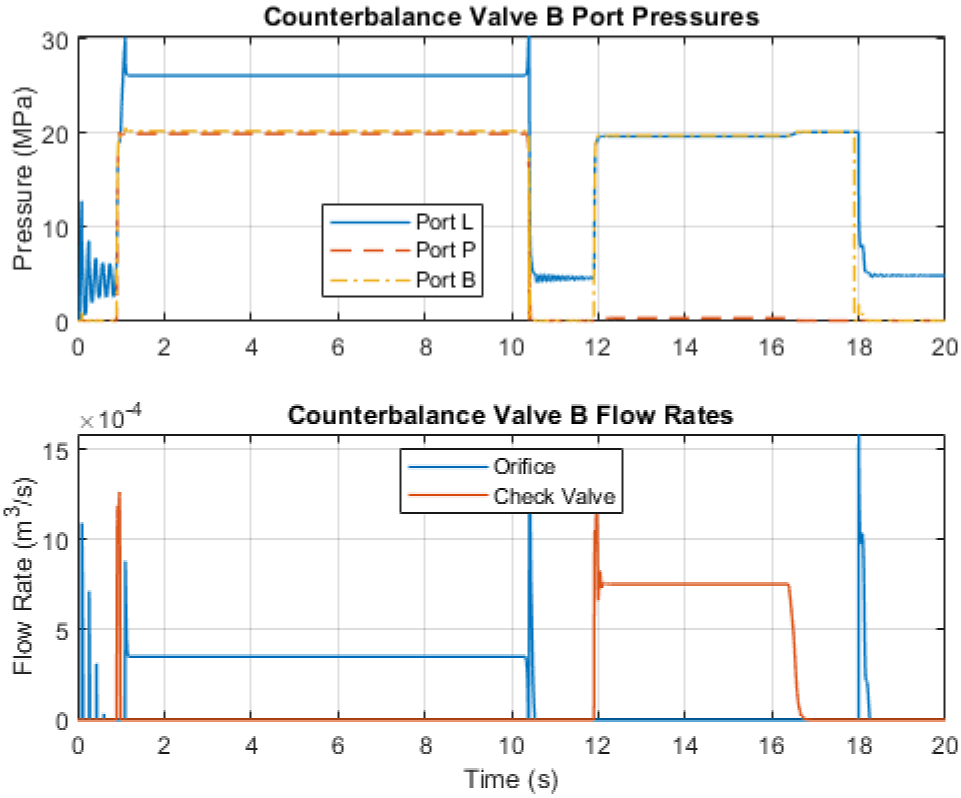
This plot shows the pressures in the two cylinder chambers and the position of the piston.



This plot shows the port pressures and flow rates for counterbalance valve A. In the first part of the simulation, flow passes through the check valve inside the counterbalance valve, allowing liquid to enter the cylinder chamber A to extend the piston. In the second part of the simulation, flow returning from the cylinder chamber A is controlled by the orifice inside the counterbalance valve. Pressures at port P and L must be sufficiently large to open the orifice and permit return flow.



This plot shows the port pressures and flow rates for counterbalance valve A. In the first part of the simulation, flow returning from the cylinder chamber B is controlled by the orifice inside the counterbalance valve. Pressures at port P and L must be sufficiently large to open the orifice and permit return flow. In the second part of the simulation, flow passes through the check valve inside the counterbalance valve, allowing liquid to enter the cylinder chamber B to retract the piston.


Counterbalance Valve B Port Pressures
Pressure (MPa)
0
10
20
30
Port L
Port P
Port B
0
2
4
6
8
10
12
14
16
18
20
Counterbalance Valve B Flow Rates
×10-4
Flow Rate (m3/s)
0
5
10
15
Orifice
Check Valve
0
2
4
6
8
10
12
14
16
18
20
Time (s)

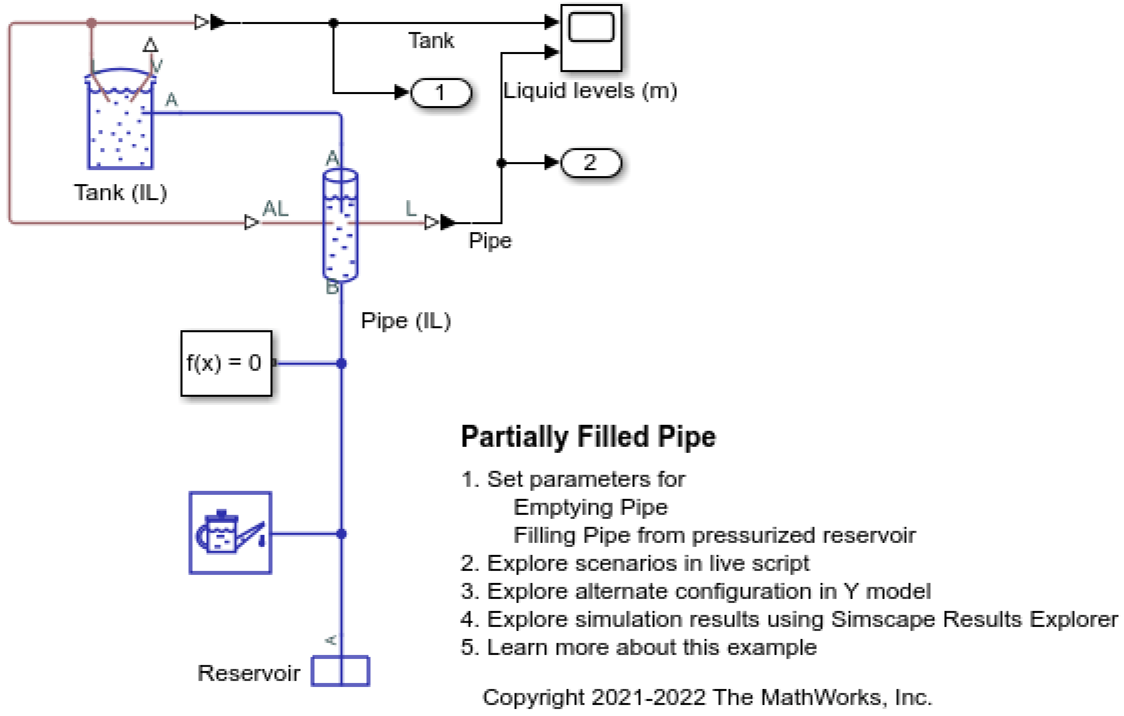
Partially Filled Pipe

This example shows the Partially Filled Pipe (IL) block used to model the emptying and filling of a tank in multiple configurations. You can use the live script provided with this example to understand the effect of different configurations.

The block represents an arbitrarily full pipe in an isothermal liquid network. The block requires Port A is higher than Port B. The physical signal input AL indicates the relative liquid level of the block that is connected to port A. When AL <= 0, the block assumes there is no liquid is entering port A.

Basic Model

This model demonstrates the basic function of the partially filled pipe using one Partially Filled Pipe (IL) block.

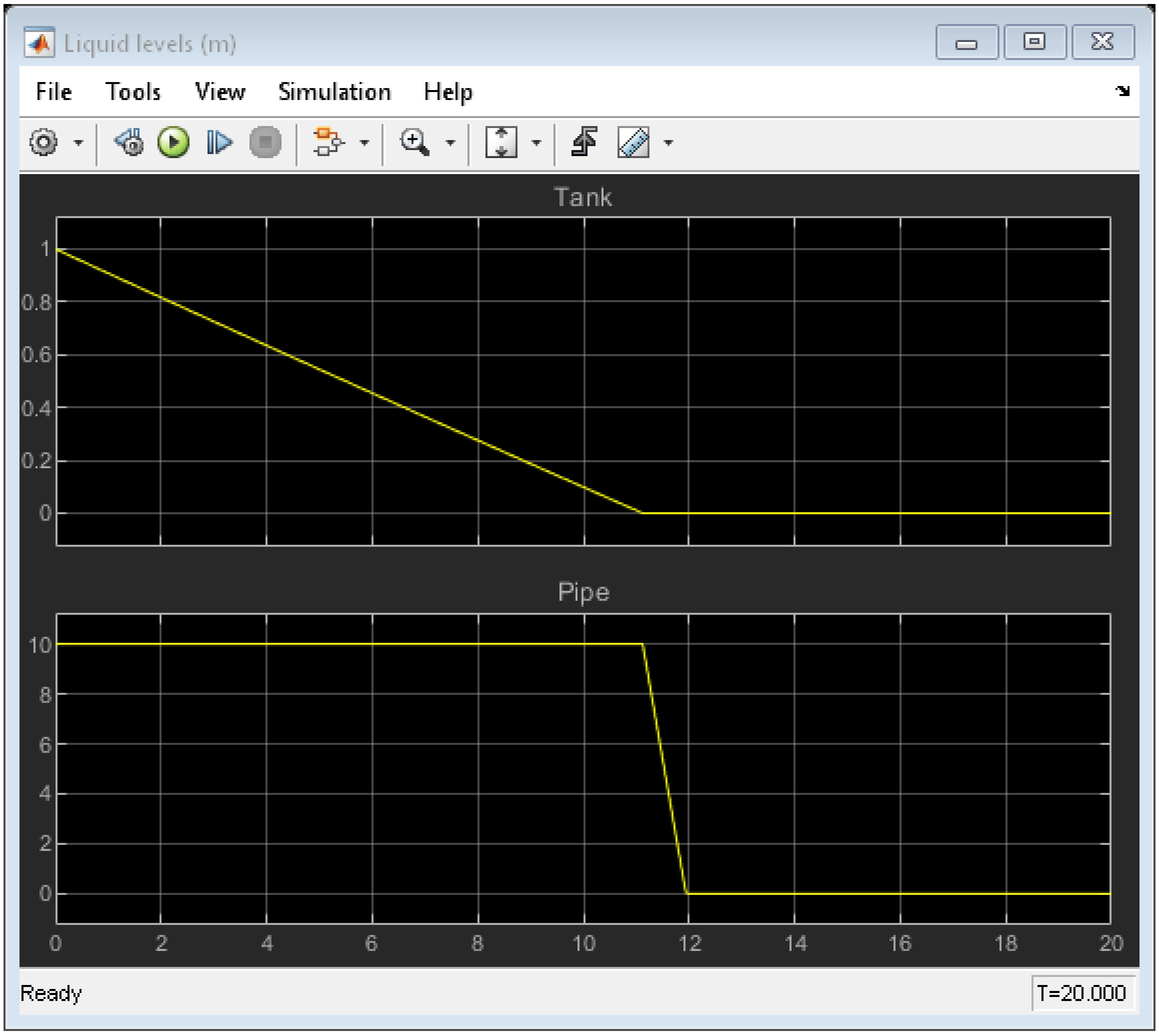


Partially Filled Pipe

1. Set parameters for
   Emptying Pipe
   Filling Pipe from pressurized reservoir
2. Explore scenarios in live script
3. Explore alternate configuration in Y model
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example

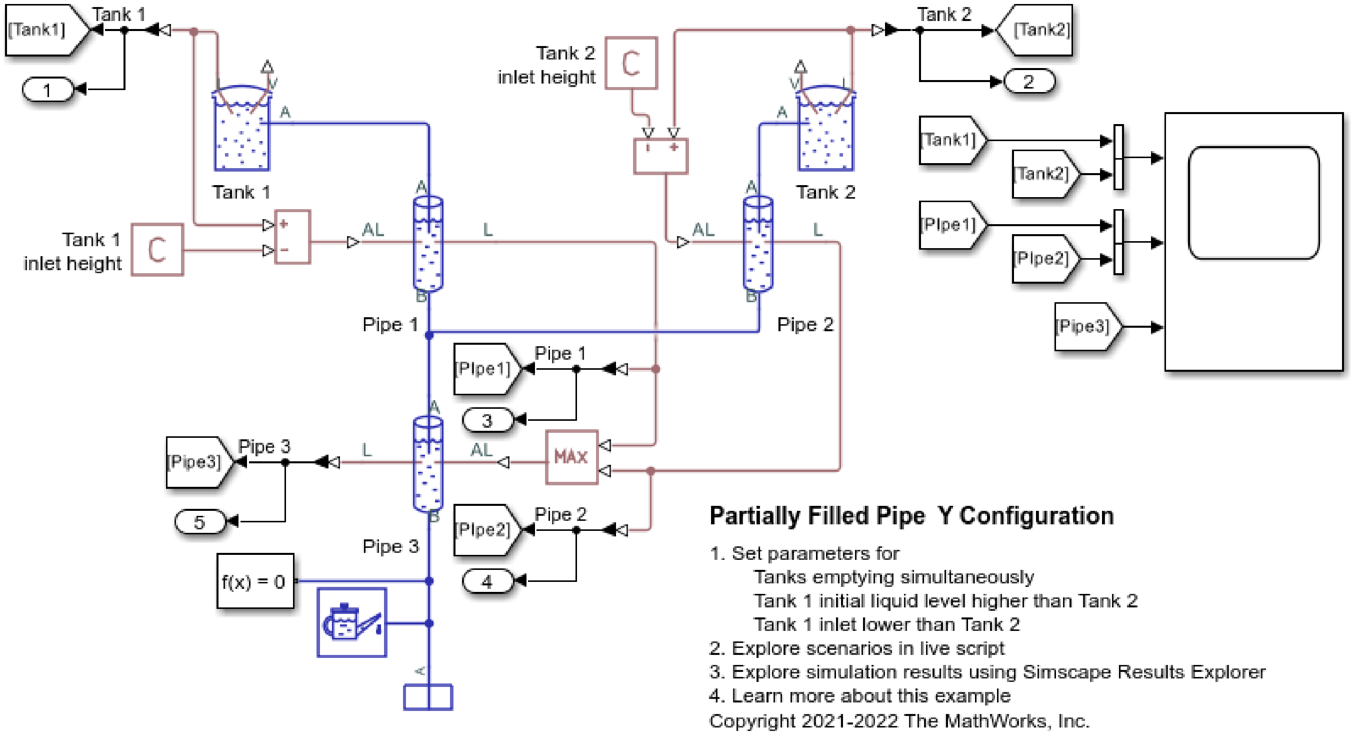
Copyright 2021-2022 The MathWorks, Inc.

Simulation Results from Scopes



Y configuration Model

This model demonstrates a more complex configuration which uses 3 Partially Filled Pipe (IL) blocks arranged in a Y configuration.

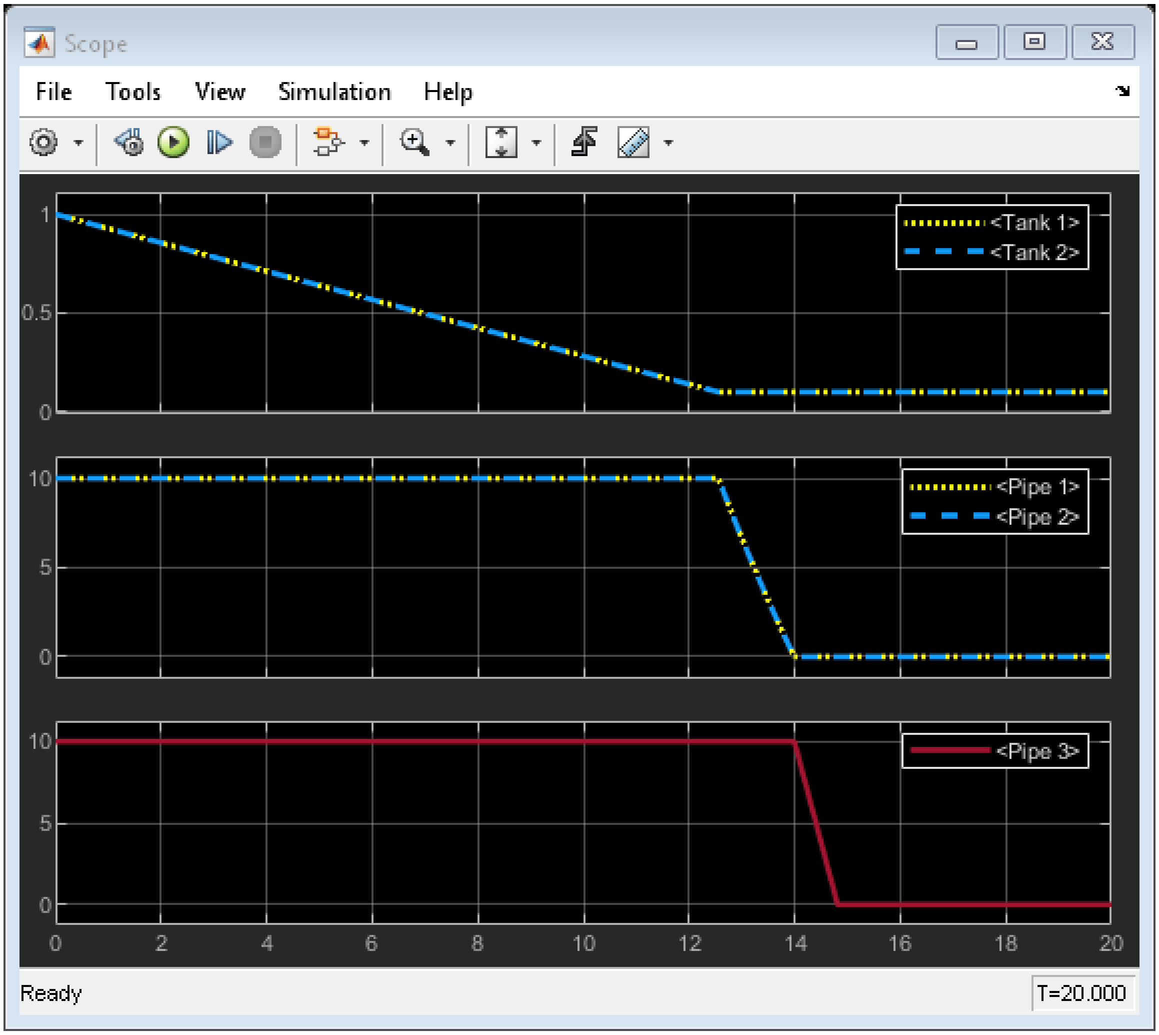


Partially Filled Pipe Y Configuration

1. Set parameters for
   Tanks emptying simultaneously
   Tank 1 initial liquid level higher than Tank 2
   Tank 1 inlet lower than Tank 2
2. Explore scenarios in live script
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2021-2022 The MathWorks, Inc.

Simulation Results from Scopes



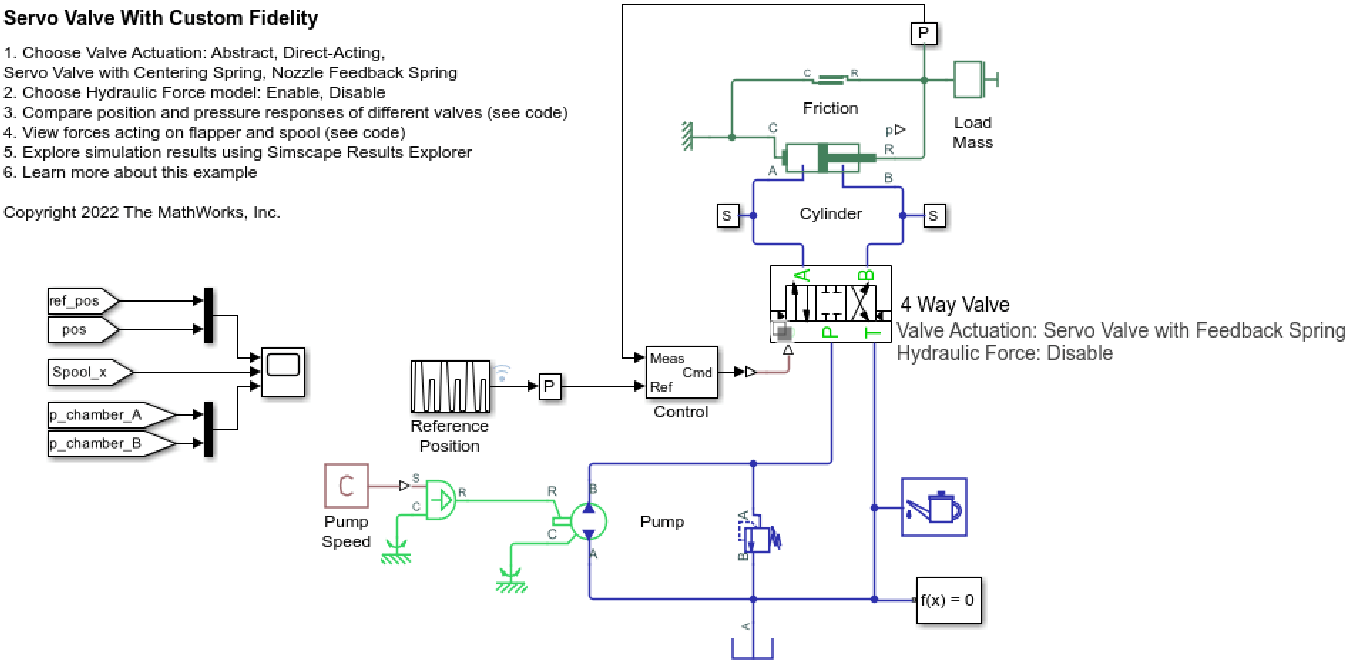
Servo Valve With Custom Fidelity

This example compares the mechanical performance of various spool actuation configurations and model fidelity levels for a hydraulic 4-way 3-position directional valve. The directional valve controls a simple double-acting cylinder in a closed-loop application. This example allows you to choose among four different spool actuation designs:

- abstract actuator,
- direct-acting electromagnetic actuator with a centering spring,
- servo valve with a centering spring,
- servo valve with a feedback spring.

For any of these valve types, you can enable or disable forces acting on the spool due to fluid flowing through the orifices.

Model



Servo Valve With Custom Fidelity

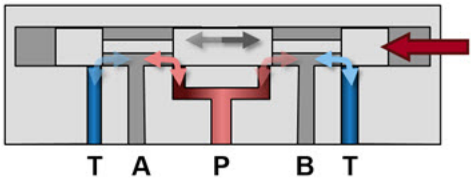
1. Choose Valve Actuation: Abstract, Direct-Acting, Servo Valve with Centering Spring, Nozzle Feedback Spring
2. Choose Hydraulic Force model: Enable, Disable
3. Compare position and pressure responses of different valves (see code)
4. View forces acting on flapper and spool (see code)
5. Explore simulation results using Simscape Results Explorer
6. Learn more about this example

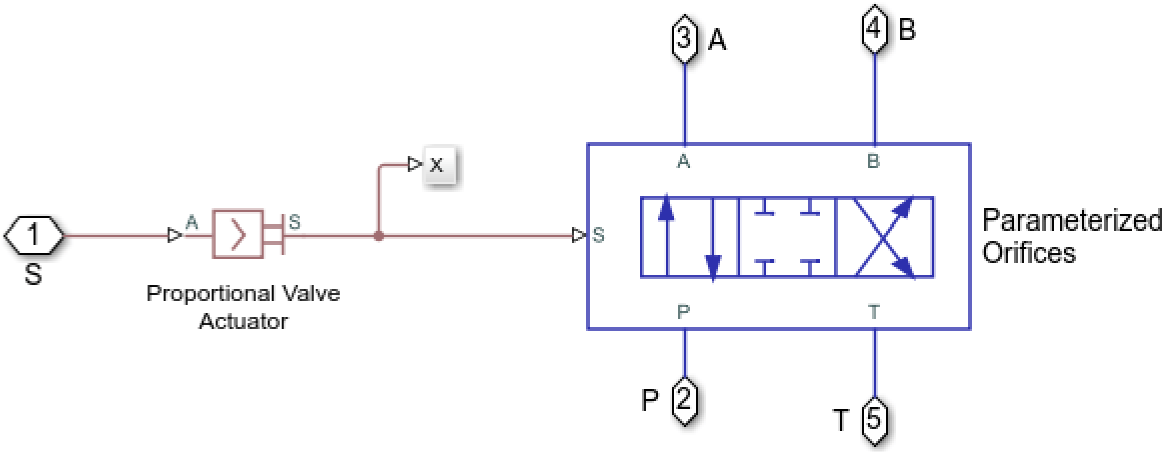
Copyright 2022 The MathWorks, Inc.

4 Way Valve
Valve Actuation: Servo Valve with Feedback Spring
Hydraulic Force: Disable

Valve with an Abstract Actuator

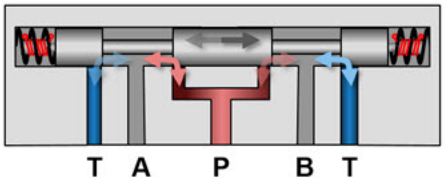
The simplest valve variant models the spool actuator abstractly as a first-order lag relation between the valve control command and the spool position.



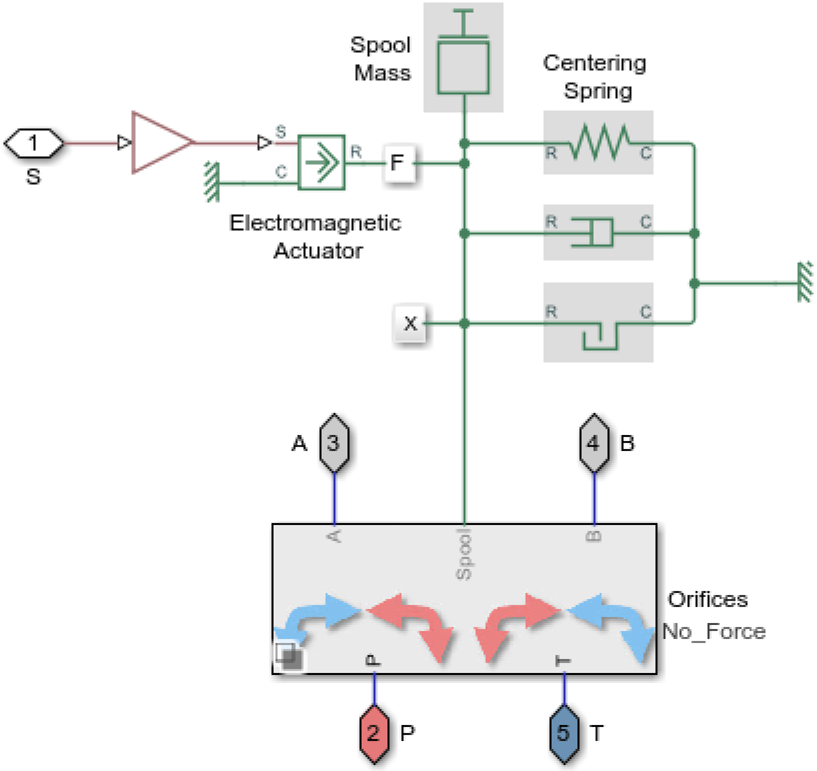


Direct-Operated Valve

The second valve variant models an electromagnetic actuator that directly drives the directional valve spool. The directional valve contains a centering spring.

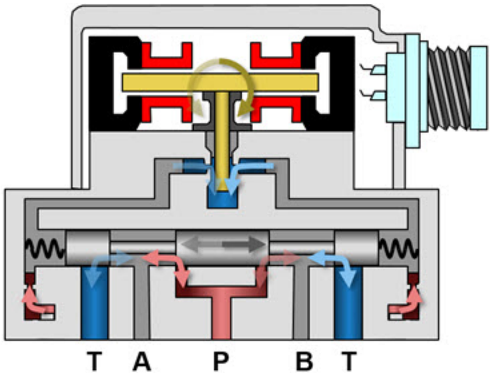


The subsystem for this valve variant contains an Ideal Force Source to abstractly represent the electromagnetic actuator and other translational mechanical elements to represent the spool and centering spring. The motion of this mechanical system drives the opening and closing of orifices that permit fluid flow in the Orifices subsystem.

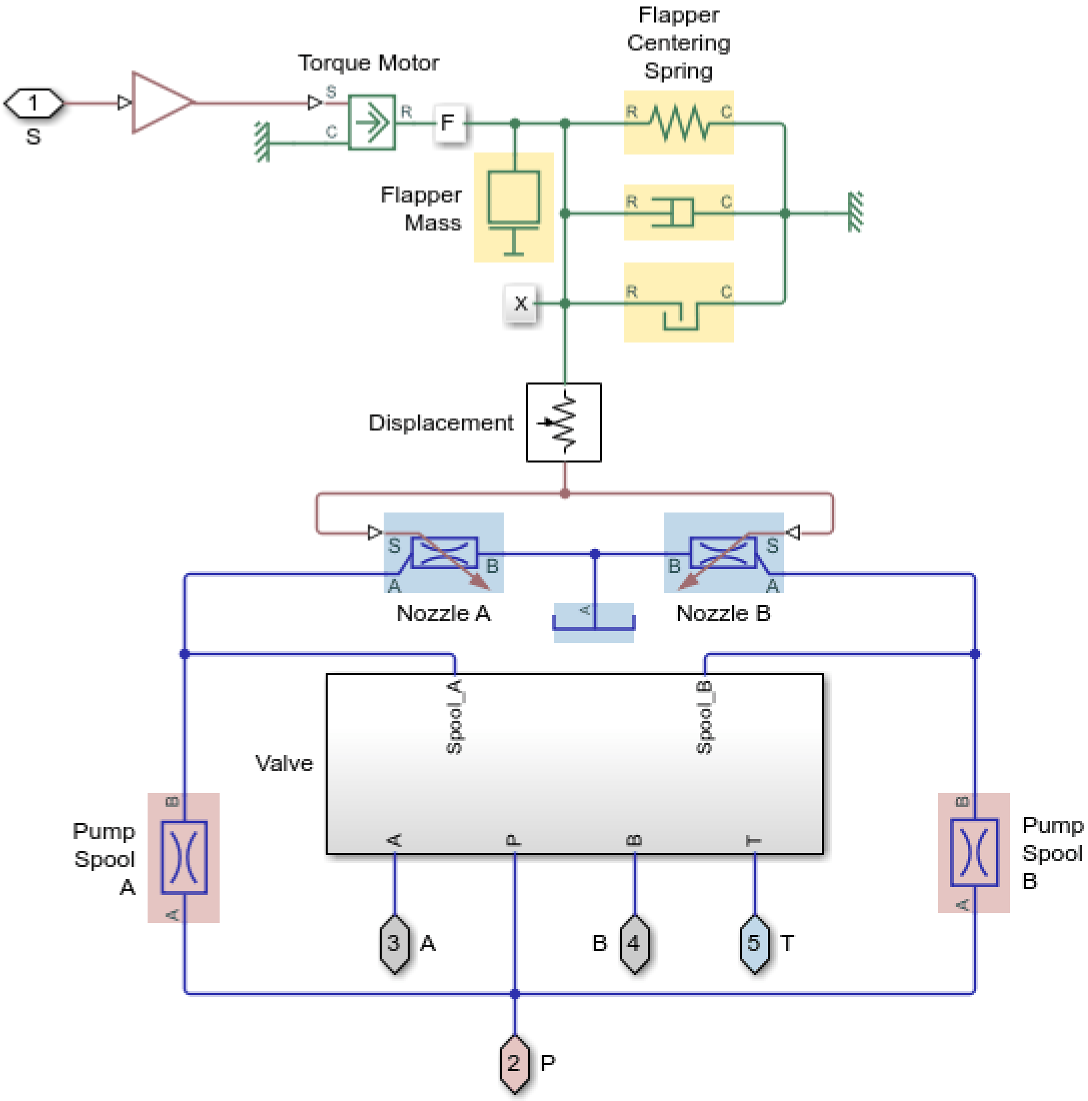


Servo Valve with a Centering Spring

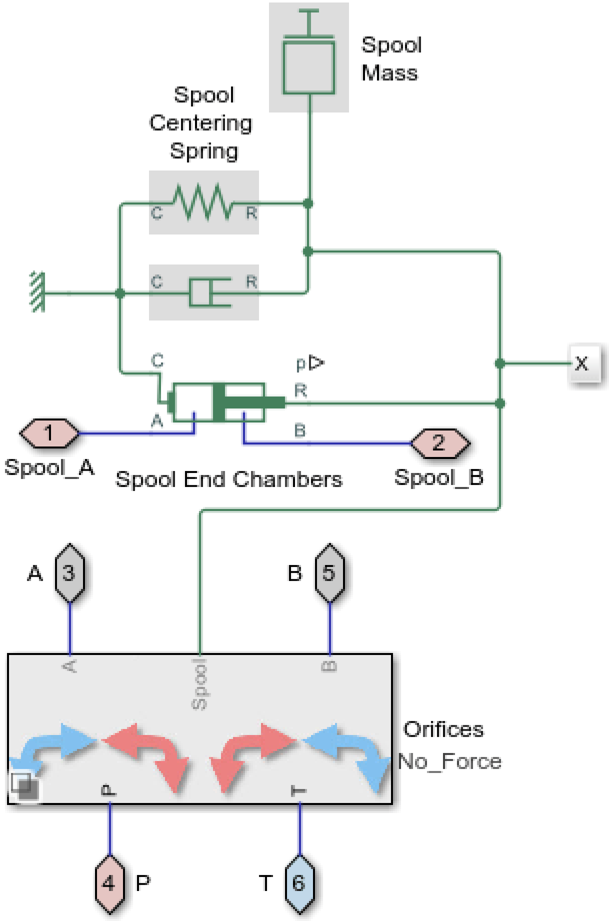
The third valve variant is a two-stage servo valve with a flapper-nozzle amplifier in the pilot stage and the 4-way 3-position spool valve in the power stage. This servo valve does not contain a feedback spring. Both the flapper and directional valve spool contain centering springs. The torque motor drives flapper motion, which varies the nozzle openings, and, in turn, varies the pressure acting on either side of the spool.



The subsystem for this valve variant contains an Ideal Force Source to represent the torque motor and other translational mechanical elements to represent the flapper and flapper centering spring. A hard stop constrains the flapper motion to remain within the allowable space in the valve cavity. The flapper displacement adjusts the nozzle openings represented by Orifice (IL) blocks. The nozzle openings affect the pressure of the fluid that feeds into the chambers on either end of the spool.

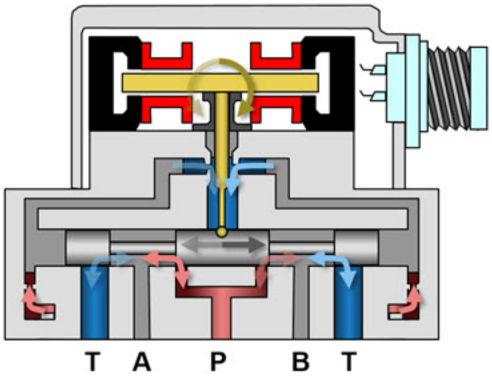


The subsystem for the power stage models the spool motion and spool centering spring using translational mechanical blocks. A double-Acting Actuator (IL) block couples the spool translational dynamics with the fluid pressure in the chambers on either end of the spool. The motion of this mechanical system drives the Orifices subsystem that models fluid flow through the spool orifices.

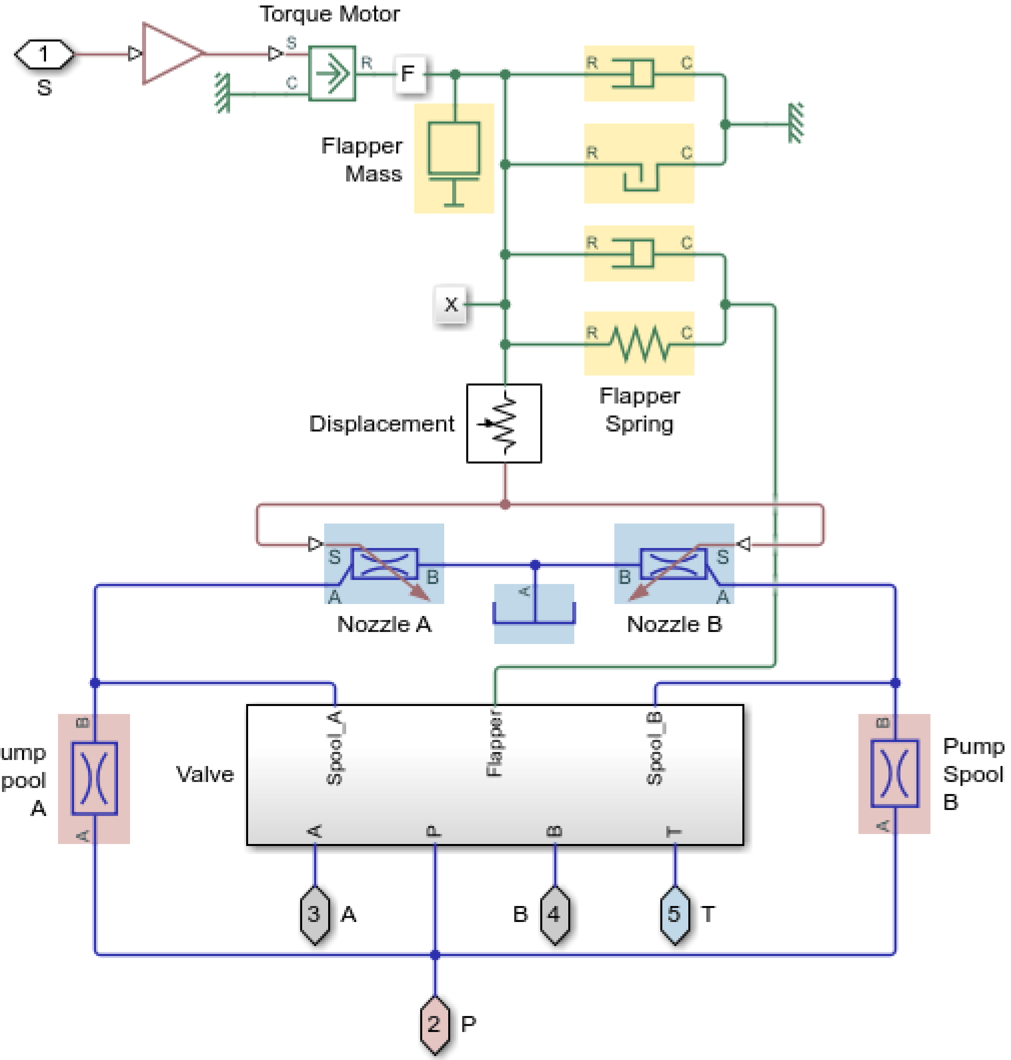


Servo Valve with a Flapper Nozzle Feedback Spring

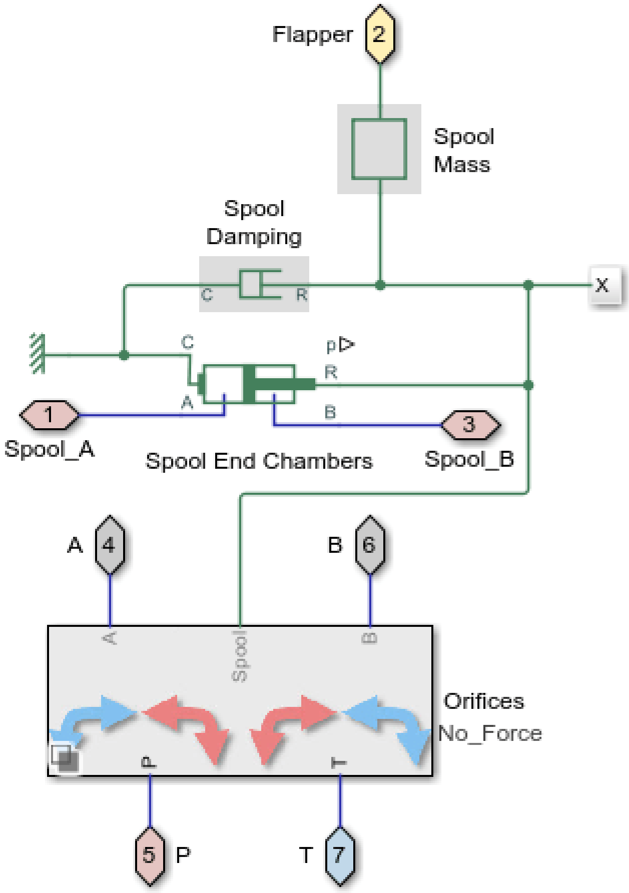
The servo valve with a feedback spring configuration is a two-stage servo valve with a 4-way, 3-position spool valve in the power stage and a flapper-nozzle amplifier in the pilot stage. The flapper is connected to the armature of a torque motor. The servo valve in this configuration is equipped with the spring feedback between the flapper and the spool of the main valve.



The subsystem for this valve variant is similar to the subsystem for the servo valve with a centering spring, except that instead of a centering spring that is grounded on one side, there is a translational mechanical flapper spring between the flapper and the spool.

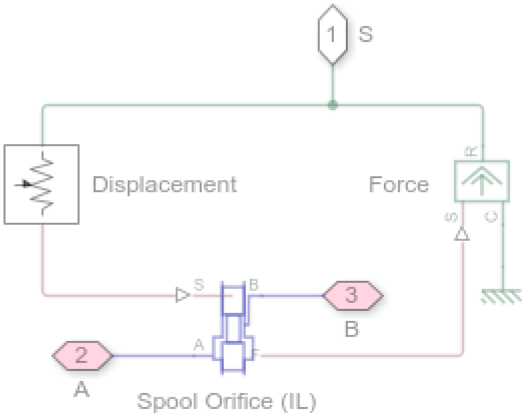


The subsystem for the power stage is similar to the subsystem for the servo valve with a centering spring, except this subsystem does not contain a spool centering spring.



Orifice AT Subsystem

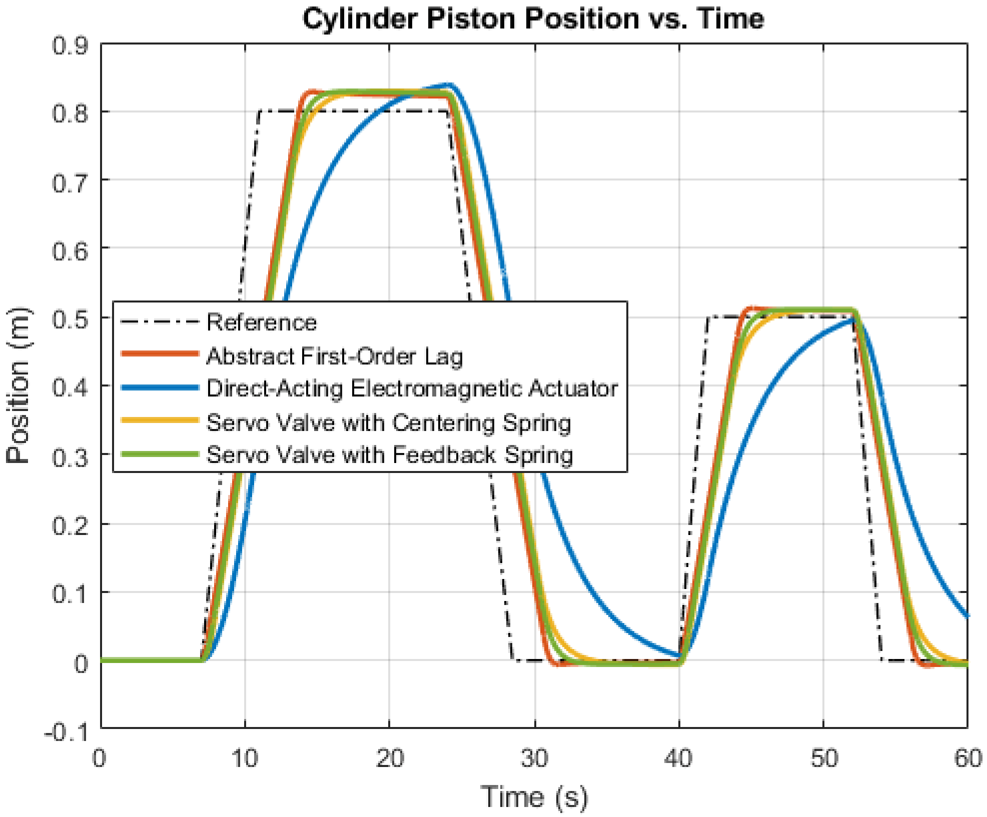
The fluid flow through each of the spool valve orifices is modeled using Spool Orifice (IL) blocks. These blocks can optionally model the steady state flow force acting on the spool.

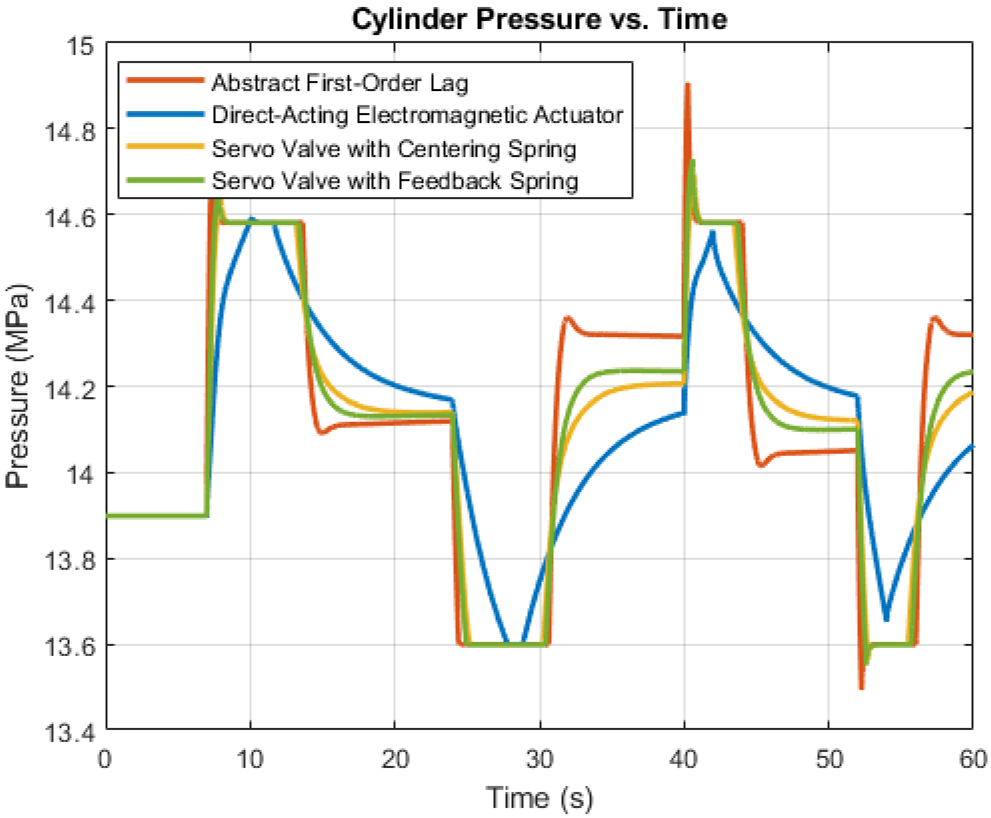


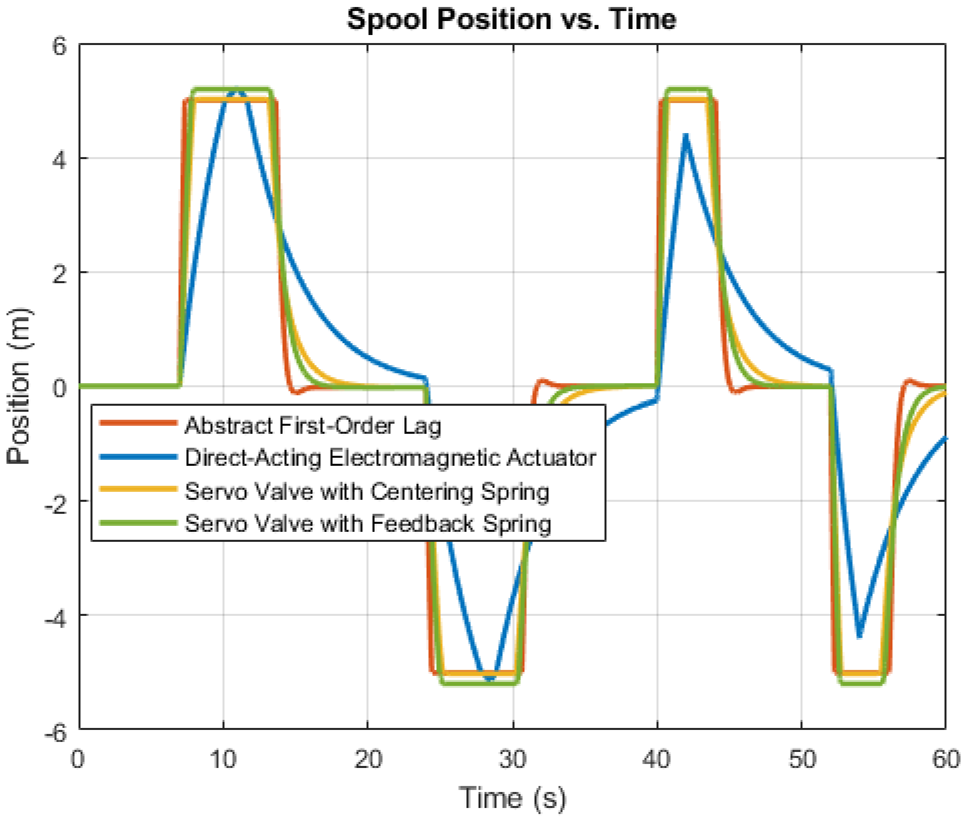
Simulation Results from Simscape Logging

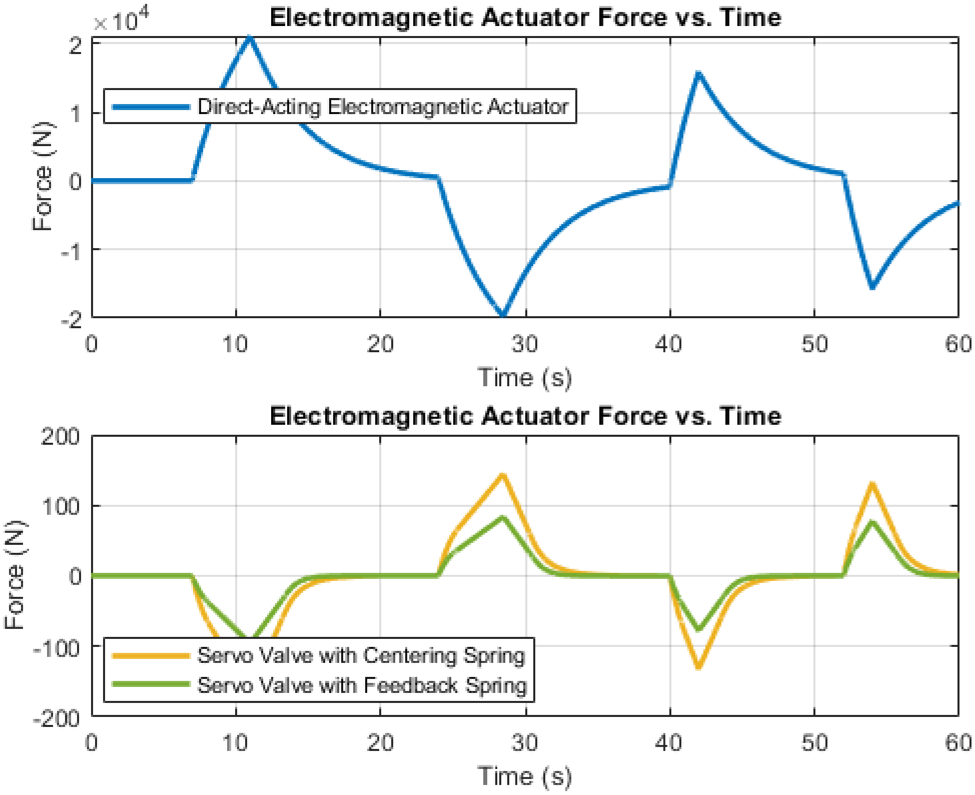
The plots below show the cylinder position, cylinder pressure, spool position, and electromagnetic actuator force versus time for the four different spool actuation designs. Among the non-abstract valve variants, the servo valve with the feedback spring responds the fastest to changes in the reference position command and its electromagnetic actuator requires the least amount of force. The direct-acting electromagnetic actuator needs to exert significantly more force than the servo valves to move the spool. The valve with the abstract first-order lag actuator has been tuned to match the motion response of the servo valve with the feedback spring.

```
Testing Valve Type: Abstract First-Order Lag. Hydraulic Forces: Disable
Testing Valve Type: Direct-Acting Electromagnetic Actuator. Hydraulic Forces: Disable
Testing Valve Type: Servo Valve with Centering Spring. Hydraulic Forces: Disable
Testing Valve Type: Servo Valve with Feedback Spring. Hydraulic Forces: Disable
```

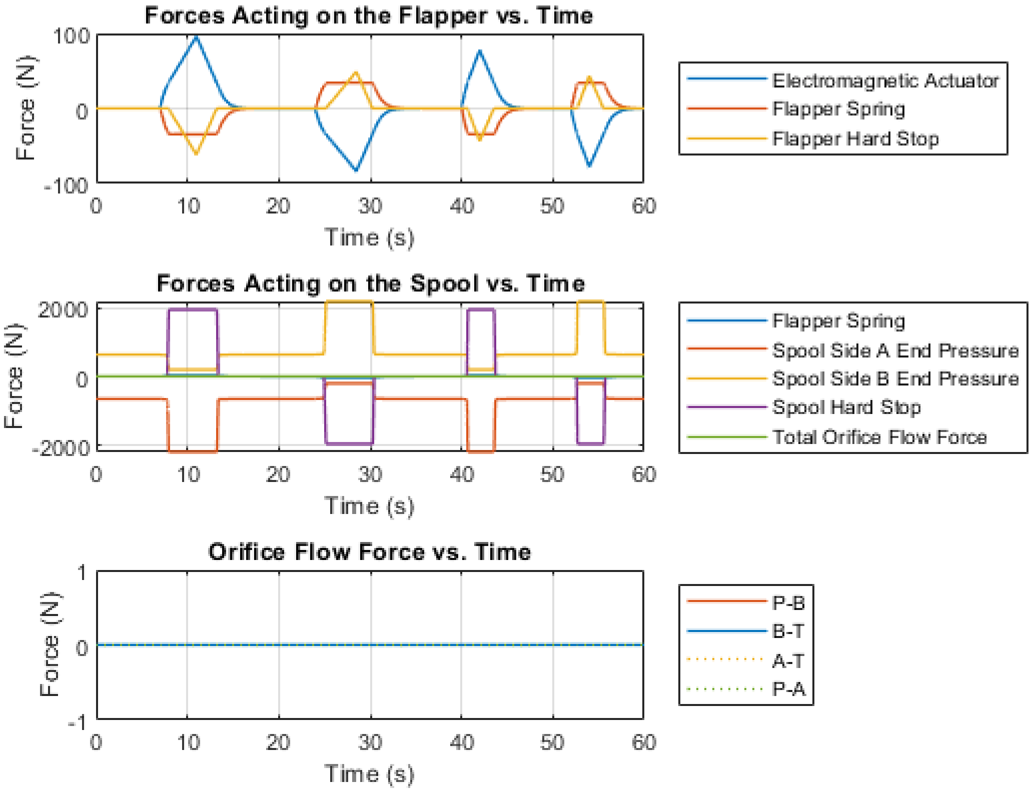



Cylinder Pressure vs. Time
Abstract First-Order Lag
Direct-Acting Electromagnetic Actuator
Servo Valve with Centering Spring
Servo Valve with Feedback Spring
Pressure (MPa)
15
14.8
14.6
14.4
14.2
14
13.8
13.6
13.4
0
10
20
30
40
50
60
Time (s)


Spool Position vs. Time
Position (m)
Time (s)
6
4
2
0
-2
-4
-6
0
10
20
30
40
50
60
Abstract First-Order Lag
Direct-Acting Electromagnetic Actuator
Servo Valve with Centering Spring
Servo Valve with Feedback Spring



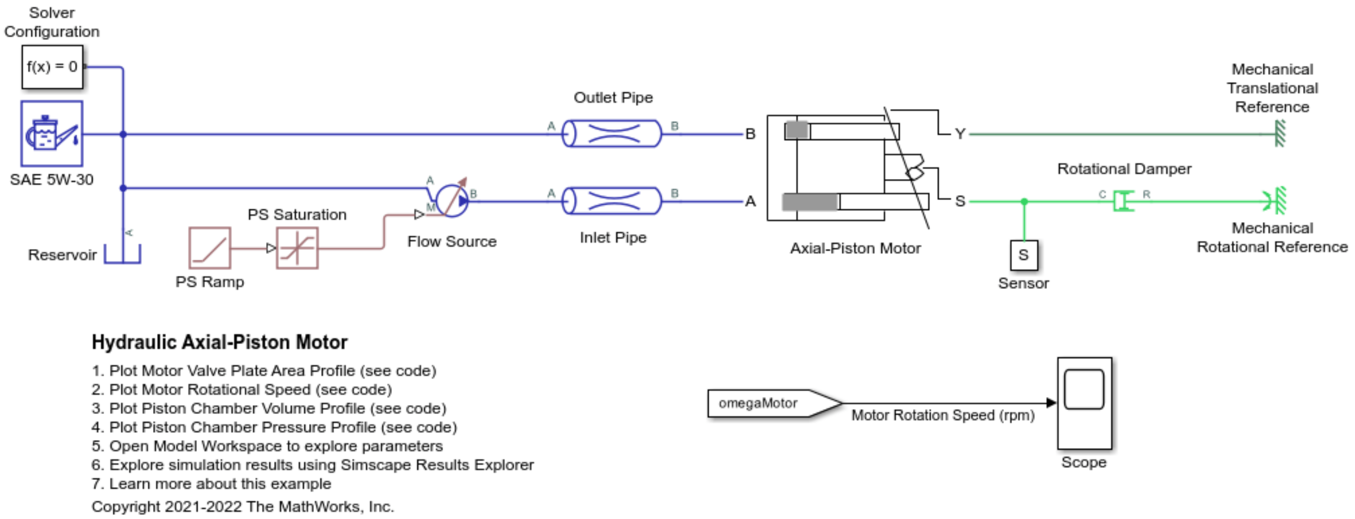
The plot below shows the dominant forces exerted on the flapper and the spool.


Forces Acting on the Flapper vs. Time
Force (N)
100
0
-100
0
10
20
30
40
50
60
Time (s)
Electromagnetic Actuator
Flapper Spring
Flapper Hard Stop
Forces Acting on the Spool vs. Time
Force (N)
2000
0
-2000
0
10
20
30
40
50
60
Time (s)
Flapper Spring
Spool Side A End Pressure
Spool Side B End Pressure
Spool Hard Stop
Total Orifice Flow Force
Orifice Flow Force vs. Time
Force (N)
1
0
-1
0
10
20
30
40
50
60
Time (s)
P-B
B-T
A-T
P-A

Hydraulic Axial-Piston Motor

This example shows an axial piston motor with a test harness. To increase the fidelity of the simulation, this model uses a detailed representation of the motor that accounts for the interaction between the pistons, swash plate, and valve plate.

Model



Hydraulic Axial-Piston Motor

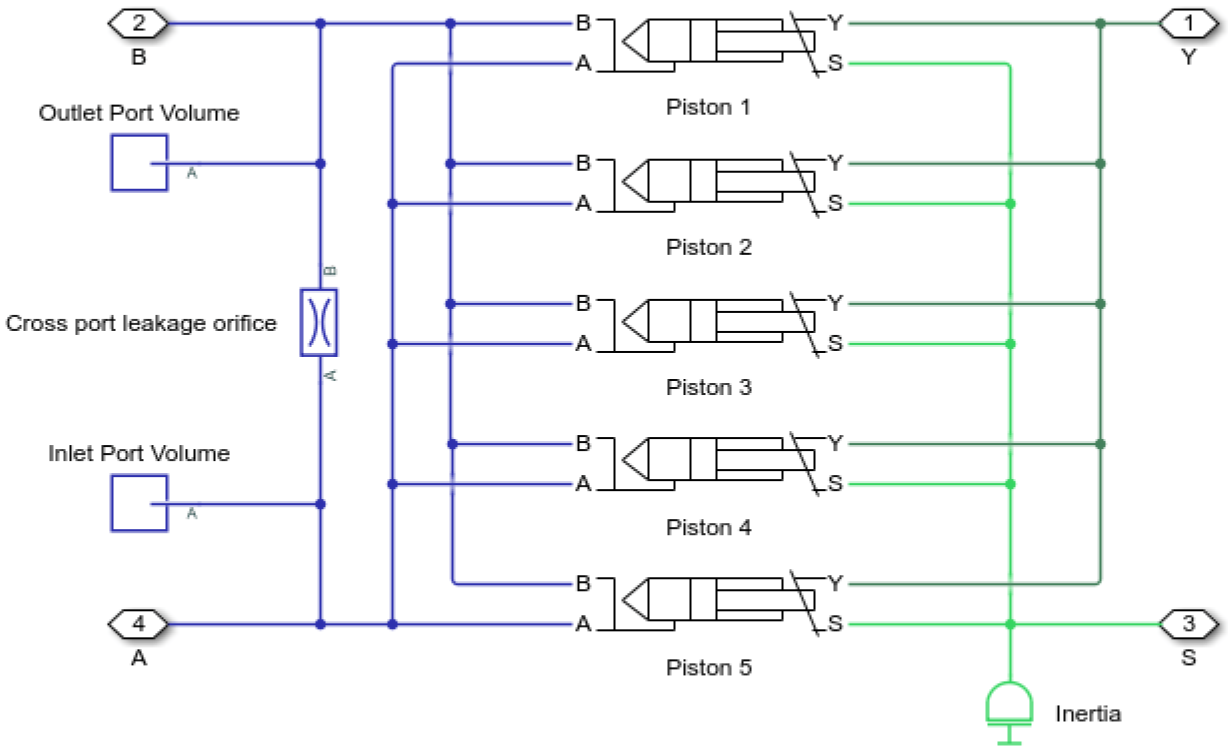
1. Plot Motor Valve Plate Area Profile (see code)
2. Plot Motor Rotational Speed (see code)
3. Plot Piston Chamber Volume Profile (see code)
4. Plot Piston Chamber Pressure Profile (see code)
5. Open Model Workspace to explore parameters
6. Explore simulation results using Simscape Results Explorer
7. Learn more about this example

Model Description

The Axial-Piston Motor subsystem contains the motor elements. The flow source connection at port A of the inlet pipe acts as the power source to the motor. The outlet pipe connects to the motor discharge at port B. The rotational damper acts as the load to the motor.

Axial-Piston Motor Subsystem

The five pistons are modeled in the Axial-Piston Motor subsystem:

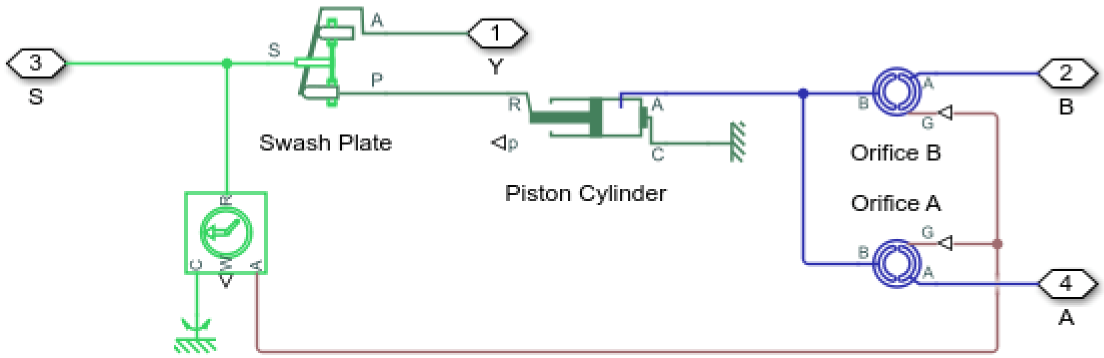


The pistons are identical and are modeled in their own identical subsystems. They are all connected to the following ports of the motor model:

- S - Motor output shaft
- Y - Yoke connected with the inclined plate of the swash mechanism
- A - Motor inlet (intake) port
- B - Motor outlet (discharge) port

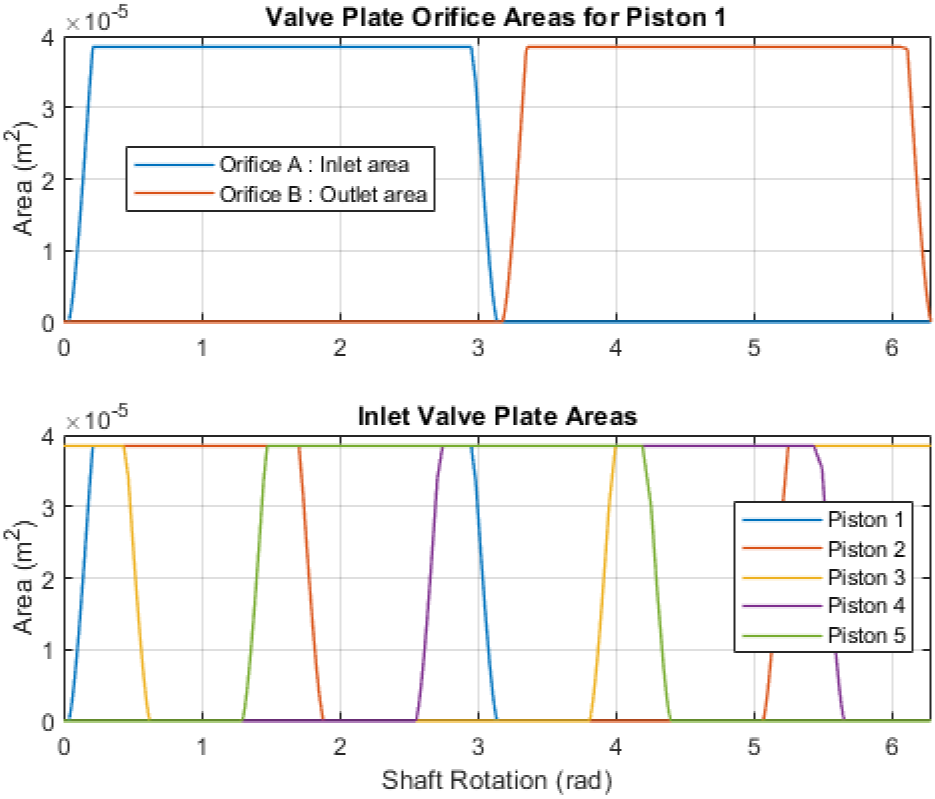
A Mechanical Translational Reference block keeps the yoke position constant.

Piston Subsystem



The piston model consists of a Single-Acting Actuator (IL) block connected mechanically to the drive shaft via the Swash Plate block. Two Valve Plate Orifice (IL) blocks hydraulically connect the piston to ports A and B. Ports A and B represent the motor intake and discharge ports, respectively.

The plot shows the flow area formation due to the piston and groove interaction.



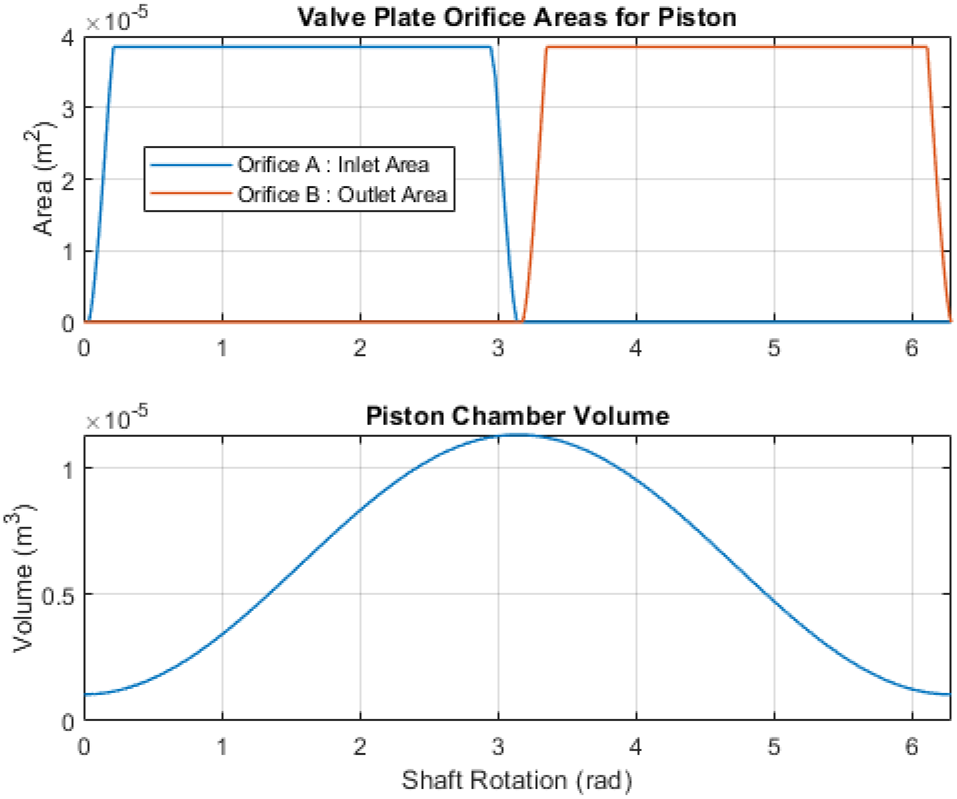
The tilting of the swash plate causes piston motion as the cylinder block rotates. Each Swash Plate block also has a 72 degree offset.

Increasing the swash plate angle increases the volumetric flow per output shaft rotation. Note that the stroke length needs to accommodate for piston motion at the maximum swash plate angle.

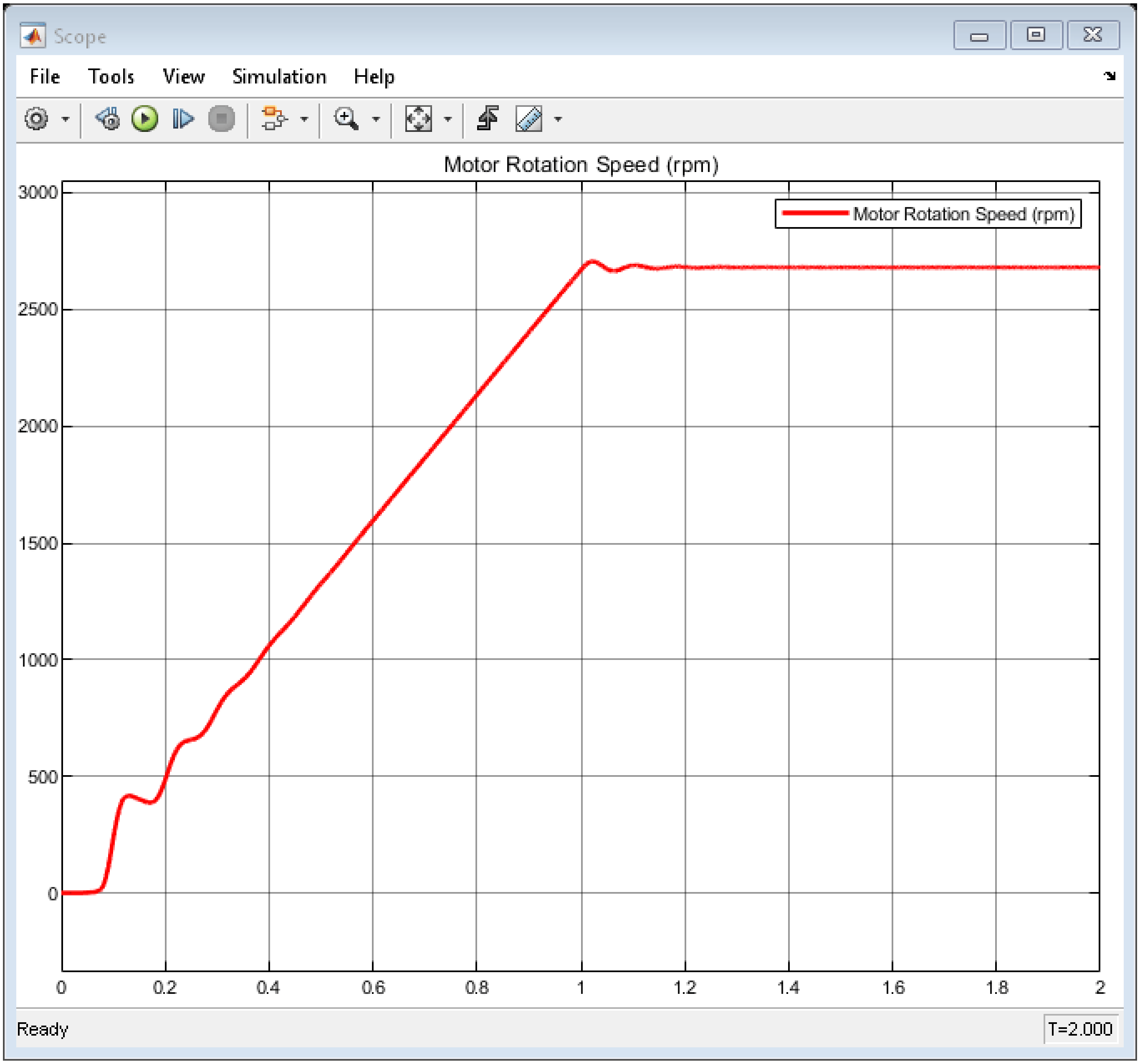
$$x_{stroke} > 2r_p \tan \alpha_{max}$$

where $r_p = 0.04$ m is the pitch radius and $\alpha_{max} = 35$ deg is the maximum swash plate angle. This means the piston stroke must be greater than 0.056 m. This example uses 0.06 m.

The plot shows how the valve plate area profiles correspond to the piston chamber volumes.

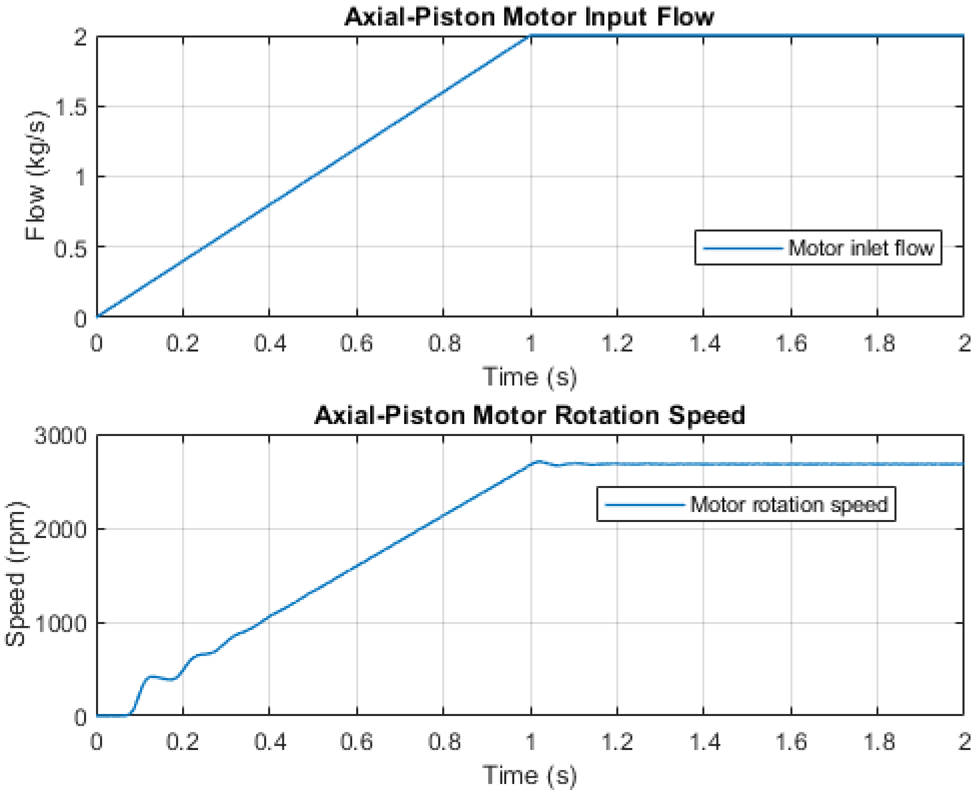

Valve Plate Orifice Areas for Piston
×10^-5
Area (m^2)
0
1
2
3
4
Orifice A : Inlet Area
Orifice B : Outlet Area
0
1
2
3
4
5
6
Piston Chamber Volume
×10^-5
Volume (m^3)
0
0.5
1
0
1
2
3
4
5
6
Shaft Rotation (rad)

Simulation Results from Scope

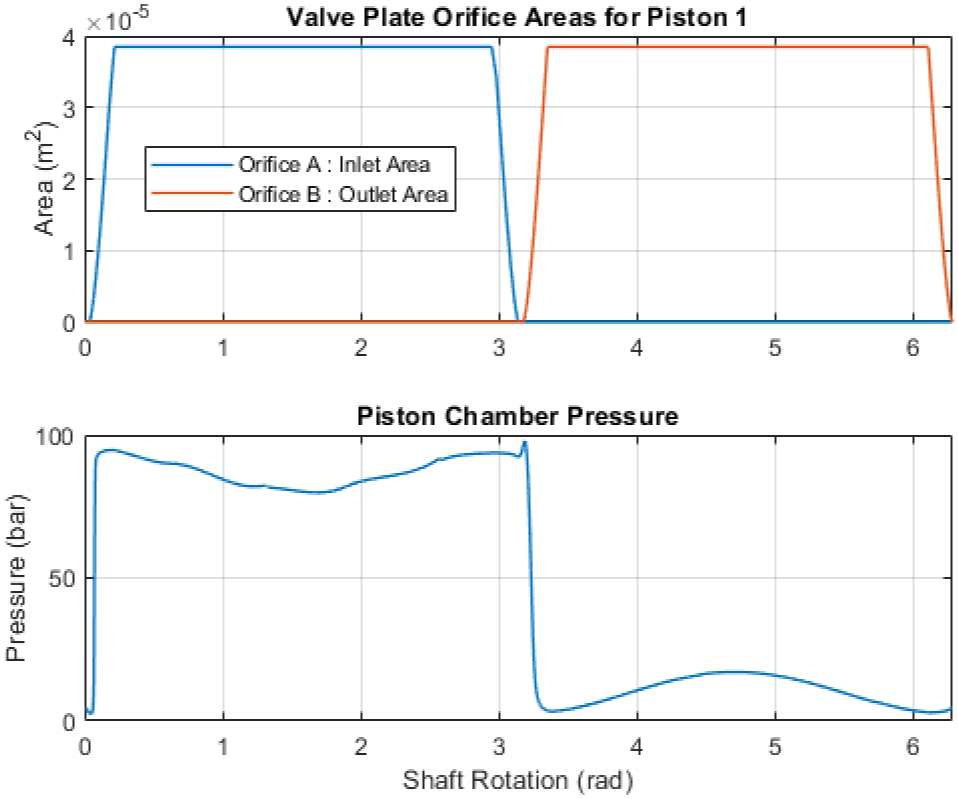

Scope
File Tools View Simulation Help
Motor Rotation Speed (rpm)
Motor Rotation Speed (rpm)
3000
2500
2000
1500
1000
500
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Ready
T=2.000

Simulation Results from Simscape Logging

This plot shows the speed of rotation of the axial-piston motor. The speed is proportional to the inlet flow to the motor.



This plot shows in the first subplot the area profiles of the valve plate geometry. The second subplot shows the piston chamber pressure variation with respect to the cylinder block rotation.

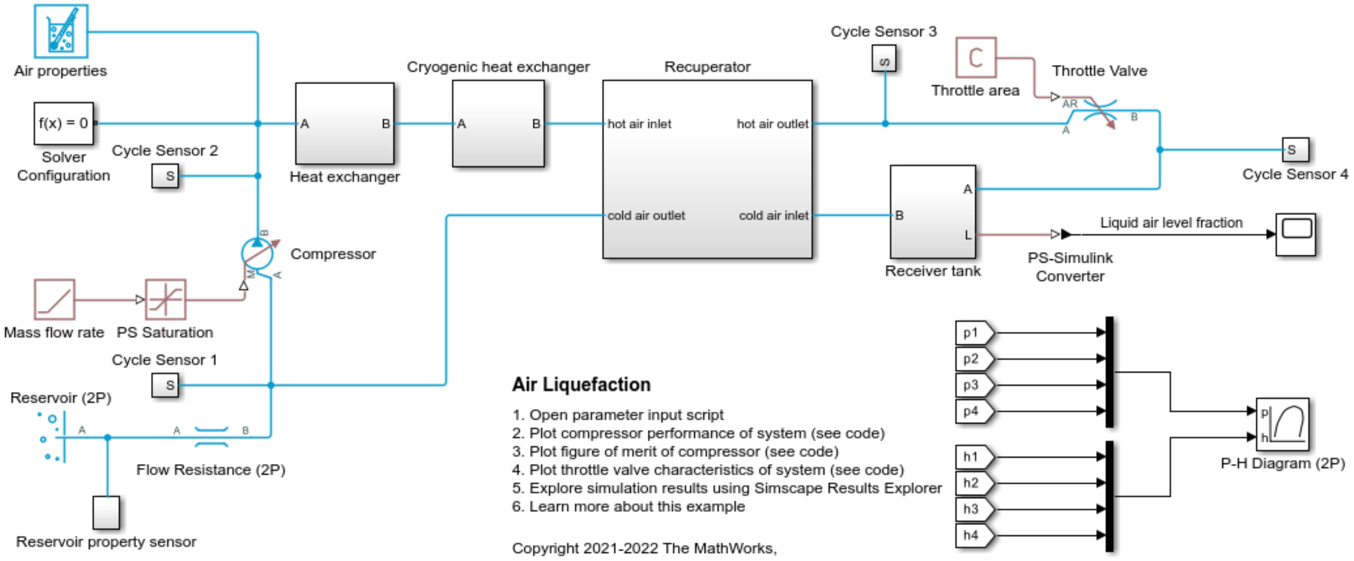

Valve Plate Orifice Areas for Piston 1
×10^-5
Area (m^2)
4
3
2
1
0
0
1
2
3
4
5
6
Orifice A : Inlet Area
Orifice B : Outlet Area
Piston Chamber Pressure
Pressure (bar)
100
50
0
0
1
2
3
4
5
6
Shaft Rotation (rad)

Air Liquefaction

This example models a Linde-Hampson air liquefaction cycle using the two-phase fluid components. The compressor drives the vapor air through two heat exchangers, recuperator, expansion valve, and storage tank. The first heat exchanger contains water at atmospheric temperature, while the second heat exchanger contains refrigerant. The pressure drops as the cooled air passes through the expansion valve which results in condensation. The air and entrained condensation then pass through a receiver tank where the condensation separates from the air. The liquid air is stored in a receiver tank while the vapor air is circulated back to the compressor.

Model

The figure shows the model of a air liquefaction cycle with its subcomponents.

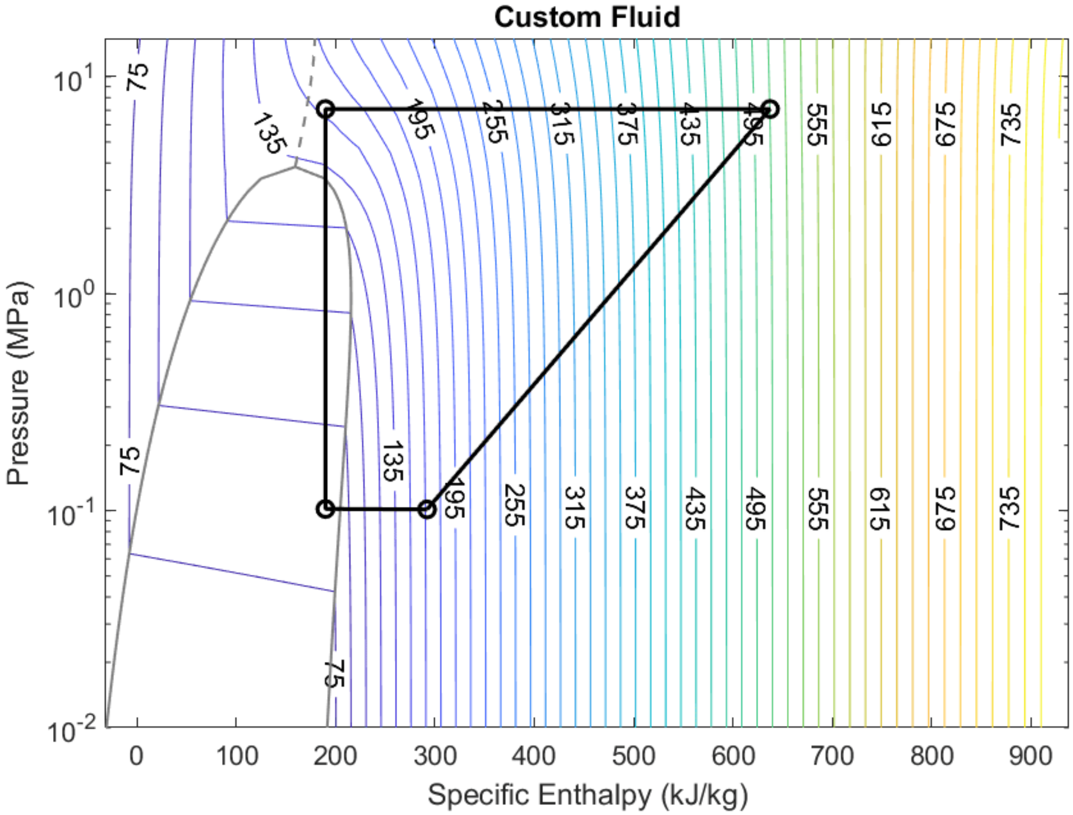


Fluid Properties

This example uses air as the working fluid, which is modeled as pseudo-pure.

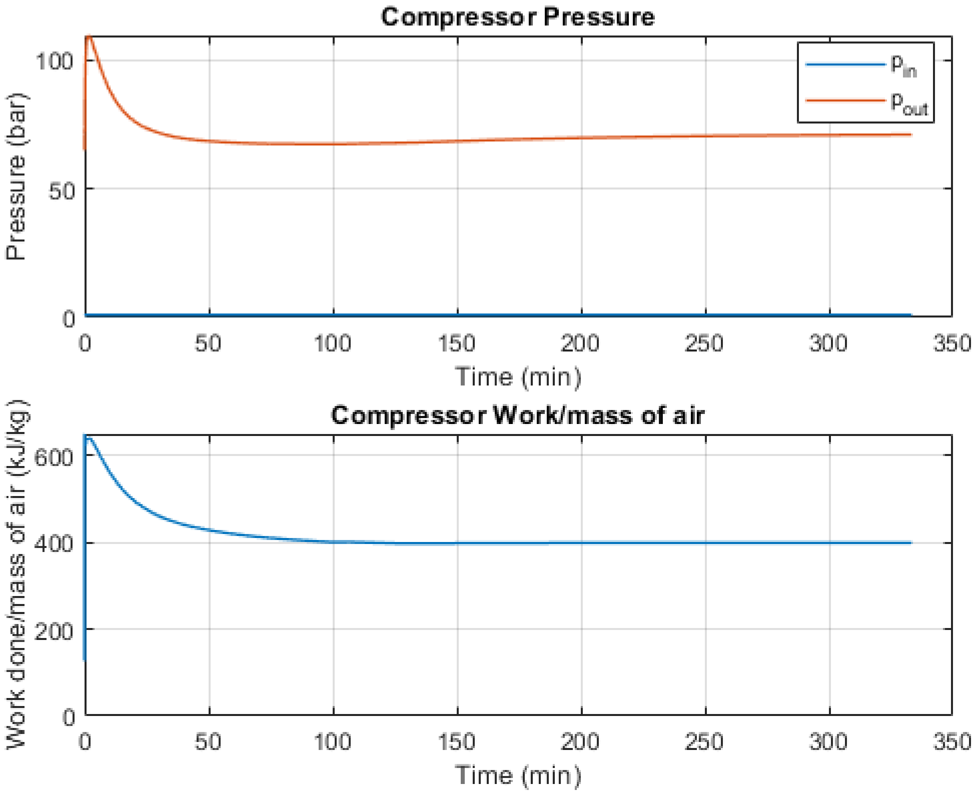
p-h Diagram of the Linde-Hampson Cycle

This figure shows the state of the fluid at the four points on the air liquefaction p-h diagram: at the compressor inlet, compressor outlet, expansion valve inlet, and expansion valve outlet. The black contour is the saturation dome of the air. Note that the lines joining the four states in the p-h diagram do not represent the fluid process.

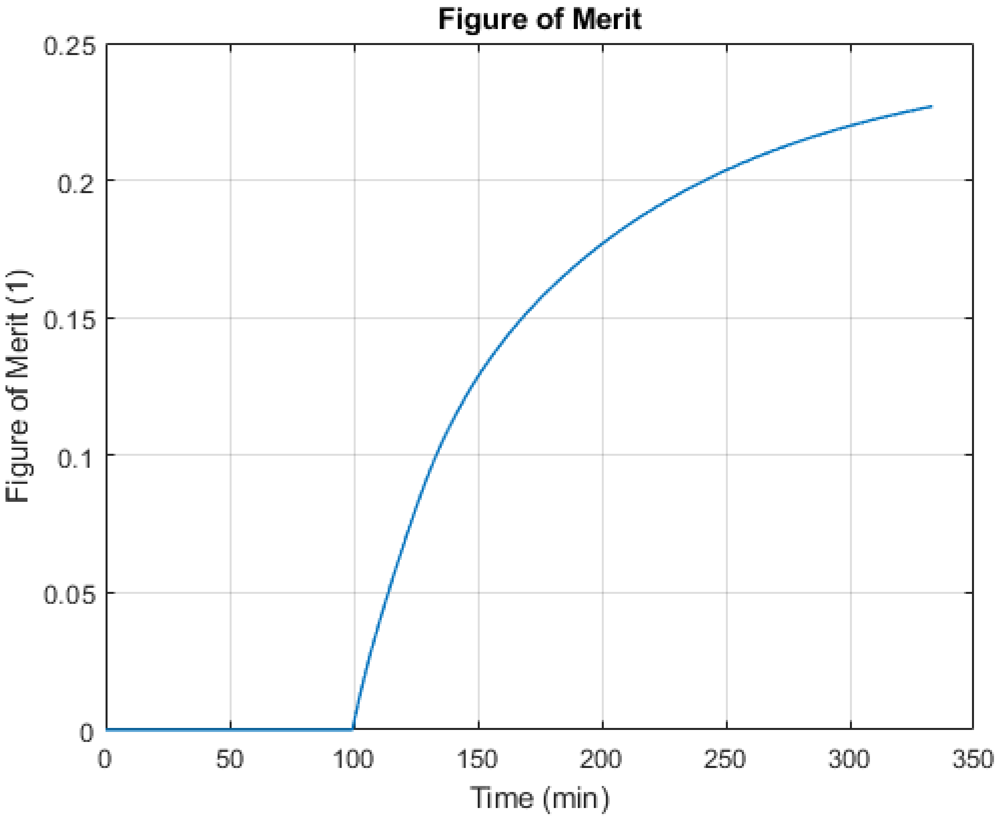


Simulation Results from Simscape™ Logging

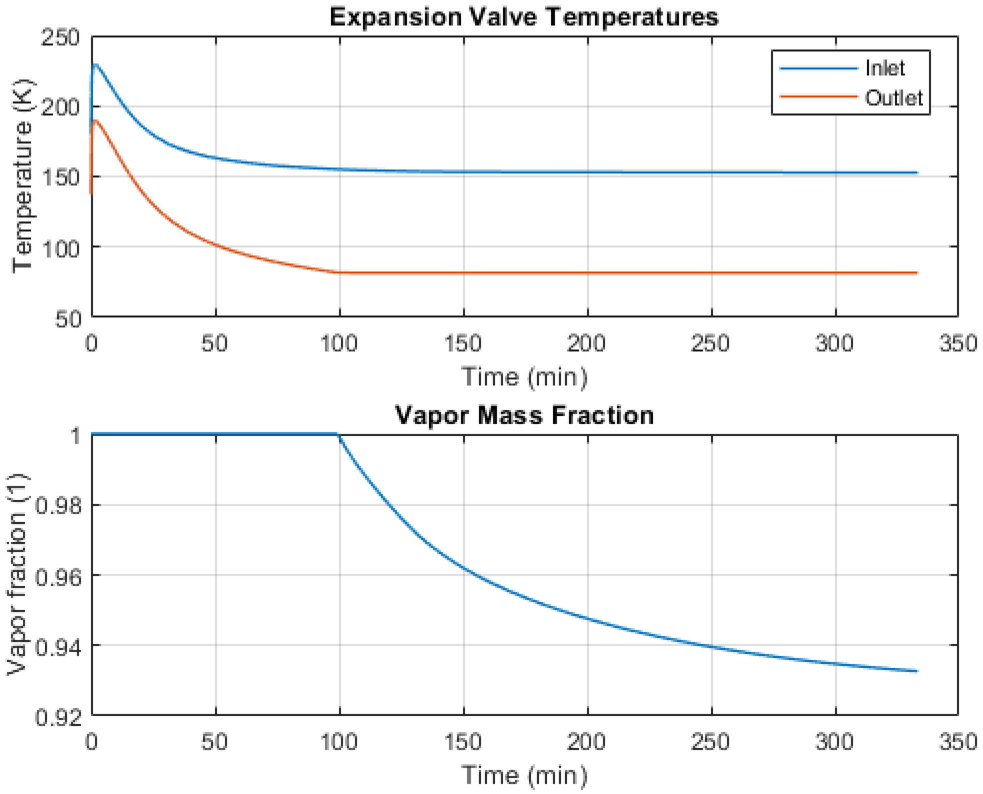
This figure shows the compressor parameters over the time. The output pressure of the compressor increases to ~110 bar initially and settles around ~62 bar. The pressure output of the compressor can be varied by changing the area of the throttle valve.



This figure shows the figure of merit of the compressor over the time. Figure of merit is the ratio of the ideal work done per unit mass of the air liquified to the actual work done per unit mass of the air liquified. It is a measurement of the efficiency of the compressor.



This figure shows the input and output temperatures for the throttle valve and vapor fraction of the fluid at the outlet of the throttle valve. The temperature drop obtained across the throttle valve is ~70K and the vapor fraction is ~0.87. The liquid fraction of the air gets accumulated in the receiver tank while the gas fraction of the air goes to the inlet of the compressor.


Expansion Valve Temperatures
Temperature (K)
250
200
150
100
50
0
50
100
150
200
250
300
350
Time (min)
Inlet
Outlet
Vapor Mass Fraction
Vapor fraction (1)
1
0.98
0.96
0.94
0.92
0
50
100
150
200
250
300
350
Time (min)

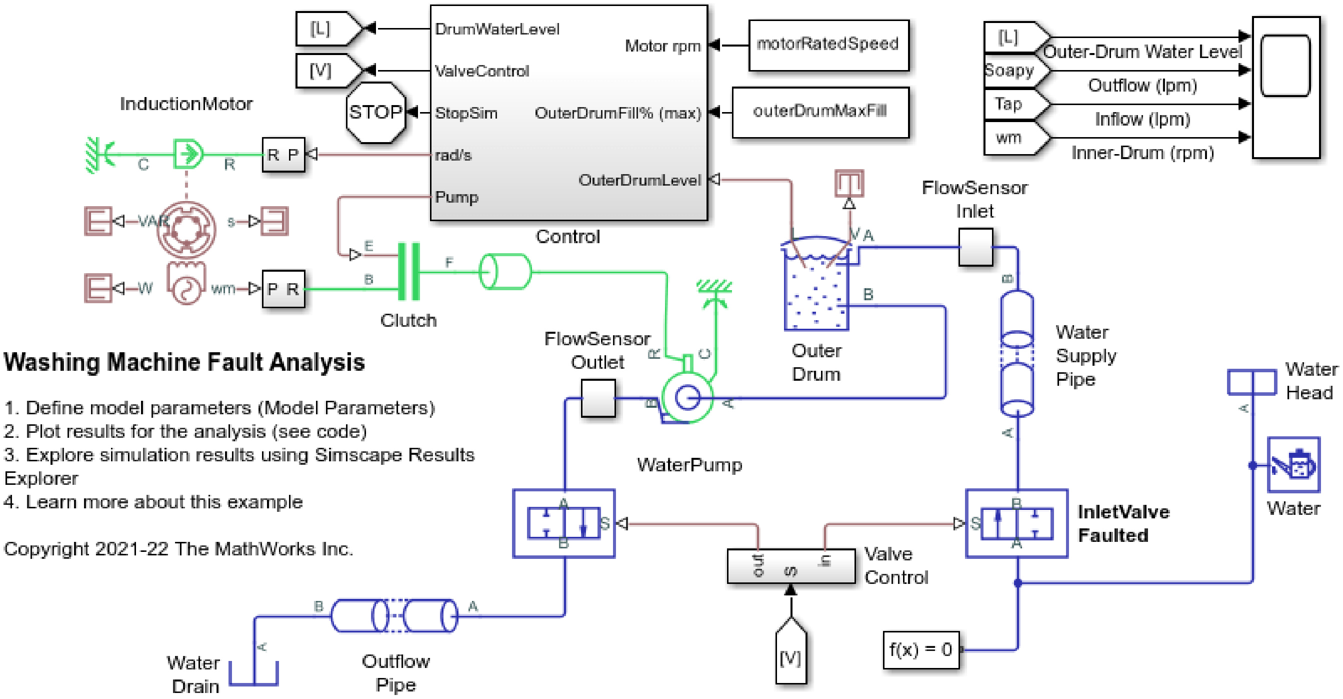
Washing Machine Fault Analysis

This example shows how to model a washing machine and introduce a fault in its operation. The machine water supply faults at a specific time and this example models the system response under this scenario. The electric motor switches off and the machine operation is aborted by discharging the partially filled outer drum.

Model Overview

```
open_system('sscv_washing_machine_fault_analysis.slx')
```

```
set_param(find_system('sscv_washing_machine_fault_analysis','FindAll', 'on','type','annotation',
```



Washing Machine Fault Analysis

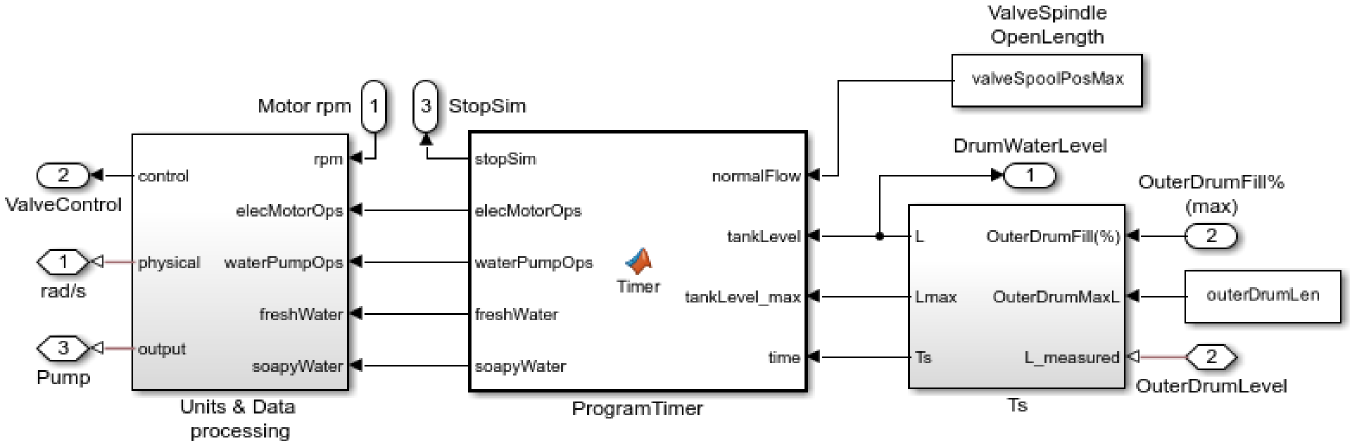
1. Define model parameters (Model Parameters)
2. Plot results for the analysis (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2021-22 The MathWorks Inc.

Control Overview

To modify or edit the machine wash cycle, use the ProgramTimer subsystem. The ProgramTime subsystem comprises the response to a fault.

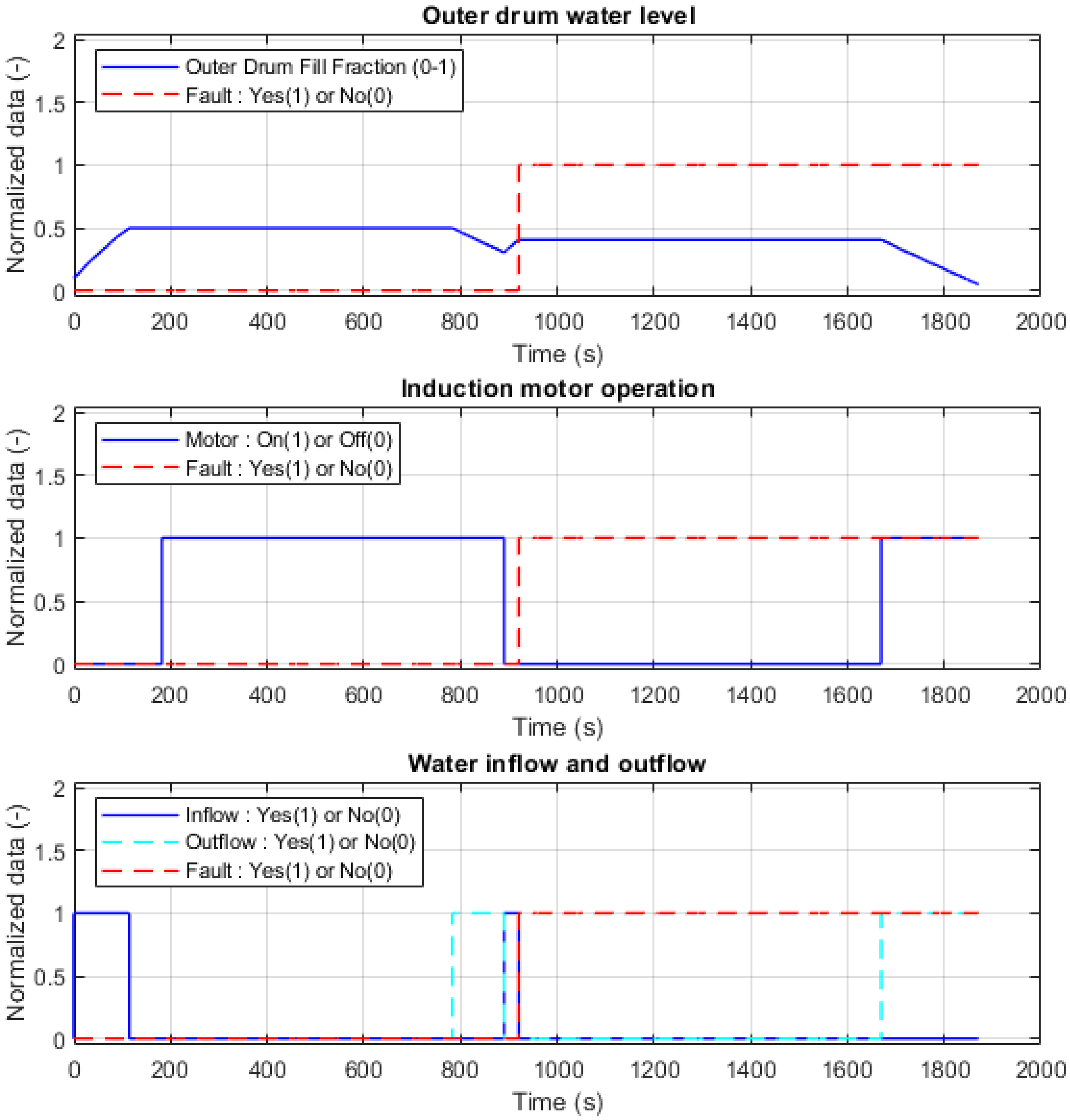
```
open_system('sscv_washing_machine_fault_analysis/Control','force')
```



Simulation Results from Simscape™ Logging

The plot below shows the water inlet and outlet valve operation (On/Off) and the pump operation with outer drum fill level. When the inlet valve faults, the pump switches off. After some time, the outlet valve opens to discharge the partially filled washing machine drum.

```
sscv_washing_machine_fault_analysis_plotResults;
```


Outer drum water level
Normalized data (-)
2
1.5
1
0.5
0
Outer Drum Fill Fraction (0-1)
Fault : Yes(1) or No(0)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Time (s)
Induction motor operation
Normalized data (-)
2
1.5
1
0.5
0
Motor : On(1) or Off(0)
Fault : Yes(1) or No(0)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Time (s)
Water inflow and outflow
Normalized data (-)
2
1.5
1
0.5
0
Inflow : Yes(1) or No(0)
Outflow : Yes(1) or No(0)
Fault : Yes(1) or No(0)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Time (s)

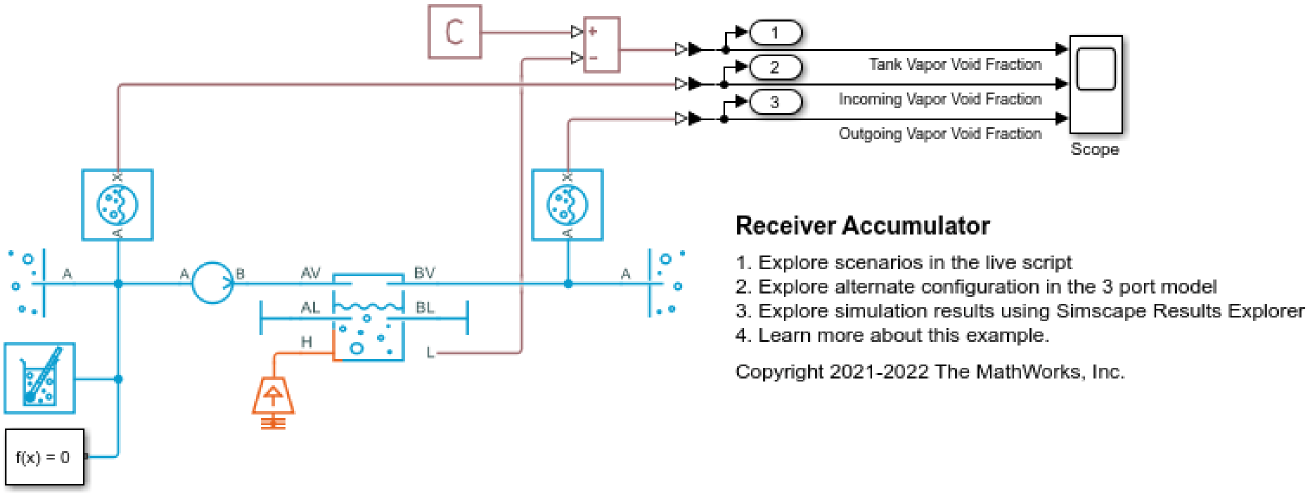
Receiver Accumulator

This example shows the liquid and vapor separation using the Receiver Accumulator (2P) block. The Receiver Accumulator (2P) block models a container of fluid in a two-phase fluid network with separate liquid and vapor ports. In an HVAC system, when this tank is placed between a condenser and an expansion valve, it acts as a receiver. Liquid connections to the block are made at ports AL and BL. When the tank is placed between an evaporator and a compressor, it acts as an accumulator. Vapor connections to the block are made at ports AV and BV. The fluid in the container can be fully liquid, fully vapor, or a mixture of both. Mass and energy exchange can occur between the fluid phases due to vaporization and condensation.

The physical signal port L of the Receiver Accumulator (2P) block reports the liquid volume fraction of the total container volume. In this example the vapor volume fraction of the signals connected to the receiver accumulator are reported using Vapor Quality Sensors. To create comparable plots for all the monitored values, the Vapor Quality sensor are set to report the vapor void fraction in the signals and the liquid volume fraction reported from the tank is converted to vapor volume fraction using a PS Subtract block.

Simple 2 Port Configuration

This model demonstrates the basic function of the Receiver Accumulator (2P) block by connecting 2 of the 4 ports of the block. Fluid with a low void fraction is sent into port AV. The plot shows that regardless of the void fraction inside the Receiver Accumulator (2P) block, port BV outputs only vapor, provided there is some vapor present in the tank.

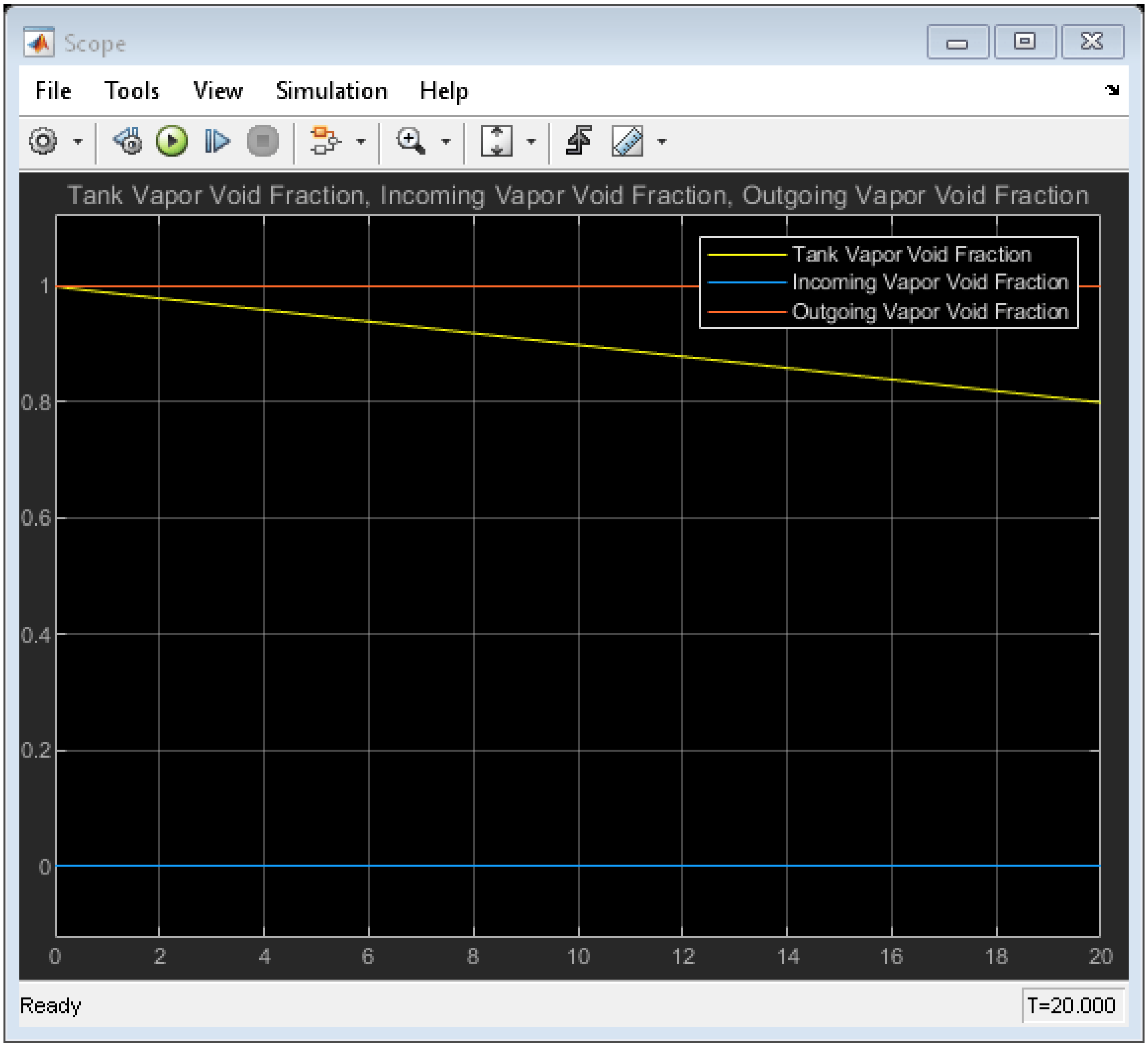


Receiver Accumulator

1. Explore scenarios in the live script
2. Explore alternate configuration in the 3 port model
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example.

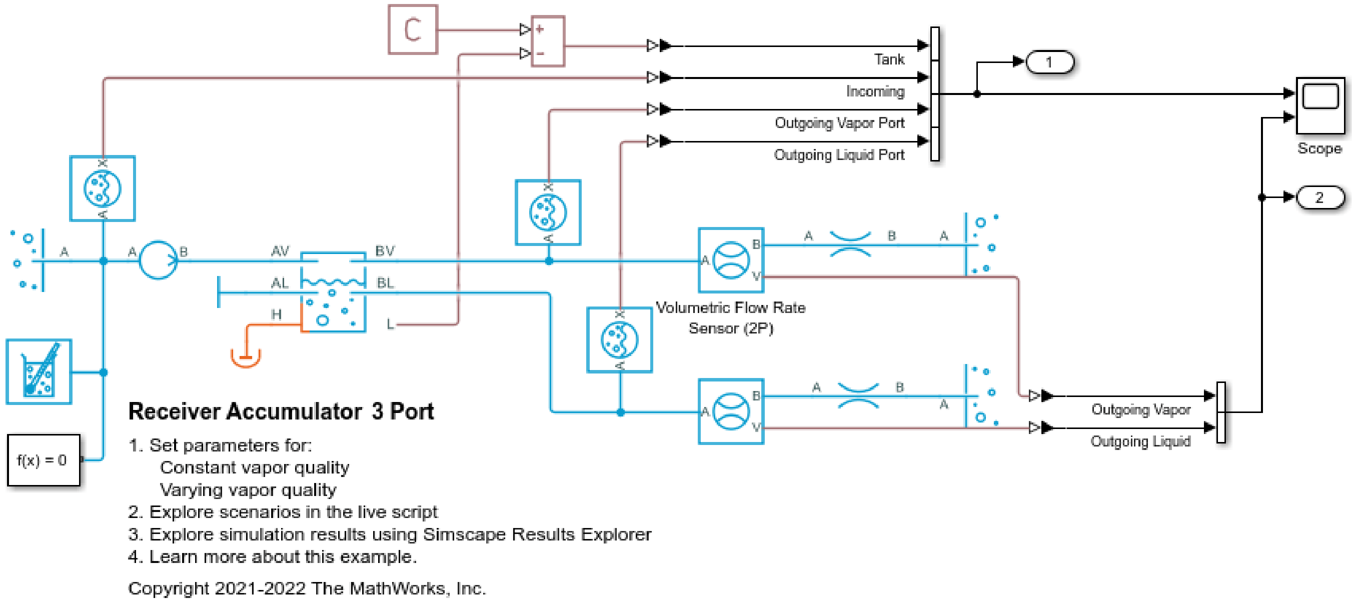
Copyright 2021-2022 The MathWorks, Inc.

Simulation Results from Scopes



3 Port Configuration

This configuration shows the Receiver Accumulator (2P) block with 3 ports connected, 1 incoming port and 2 outgoing ports. This model shows the separation of the liquid and vapor within the Receiver Accumulator (2P) block. It can be seen that as the vapor void fraction in the tank changes the vapor void fraction at ports and the mass flow rate out of the ports AV and BV remain constant and the mass flow rate out of the ports changes to adjust for the vapor void fraction of the tank.

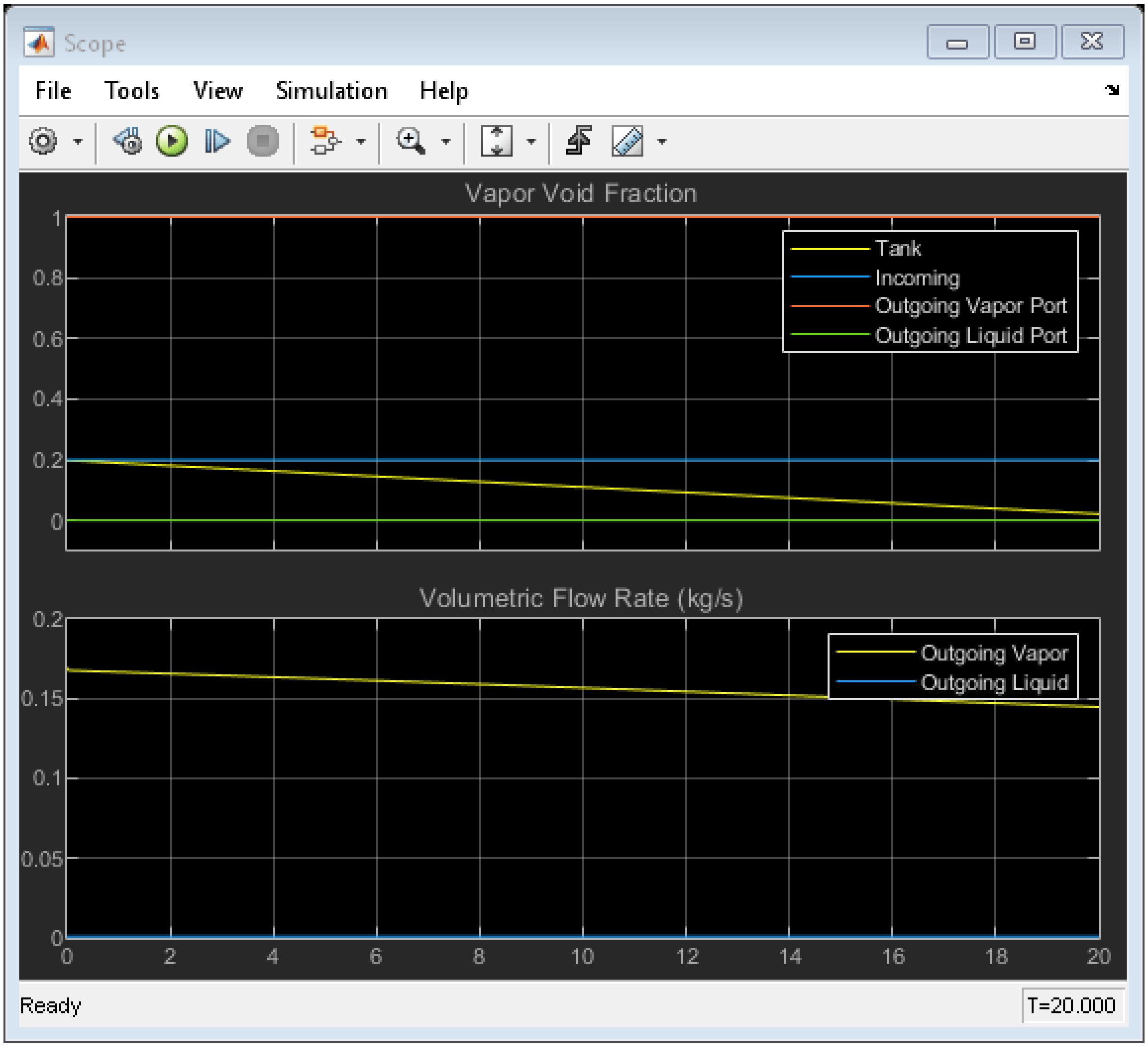


Receiver Accumulator 3 Port

1. Set parameters for:
   Constant vapor quality
   Varying vapor quality
2. Explore scenarios in the live script
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example.

Copyright 2021-2022 The MathWorks, Inc.

Simulation Results from Scopes



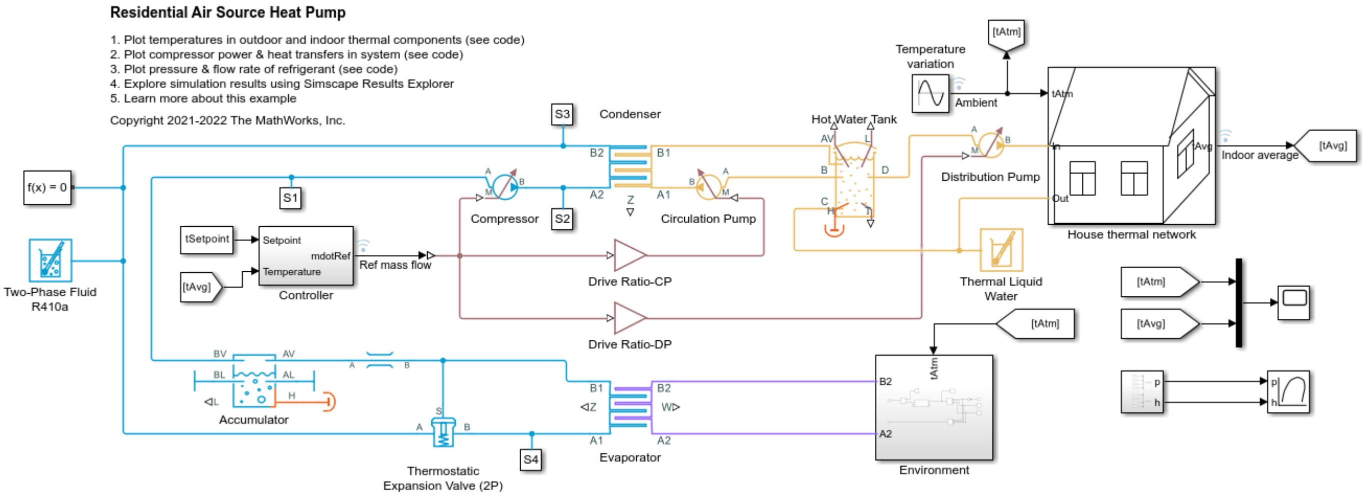
Residential Air Source Heat Pump

This example models an air source heat pump system that is used to heat a residential building having hot-water radiators for heat distribution. The two-phase fluid refrigerant takes up heat from the environment moist air mixture and transfers heat to water. The compressor drives the R410a refrigerant through a condenser, a thermostatic expansion valve, and an evaporator. An accumulator ensures that only vapor returns to the compressor.

A fan drives moist air flow over the evaporator. A circulation pump drives water flow through the condenser and hot water exiting condenser is stored in a hot water storage tank. A distribution pump drives hot water flow from the tank to the radiators. A PI controller modulates the refrigerant mass flow rate to keep the average air temperature of the building at around the setpoint temperature of 23 degC, as the ambient temperature varies between -1 degC to 11 degC.

The residential building has four radiators and four rooms. Each room exchanges heat with the environment through its exterior walls, roof, and windows. Each path is simulated as a combination of a thermal convection, thermal conduction, and the thermal mass. It is assumed that heat is not transferred internally between rooms. The initial indoor temperature of the rooms is equal to the ambient temperature of 5 degC, and the heat pump quickly increases the average indoor temperature to the setpoint temperature of 23 degC.

Model

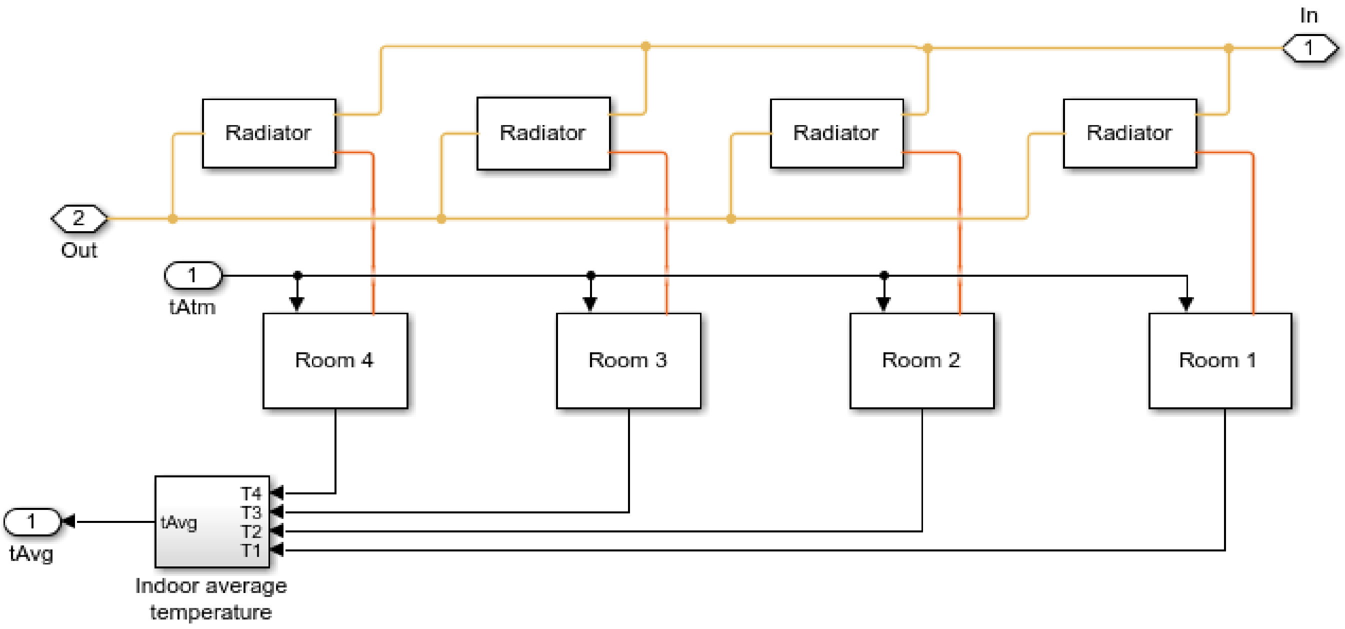


Residential Air Source Heat Pump

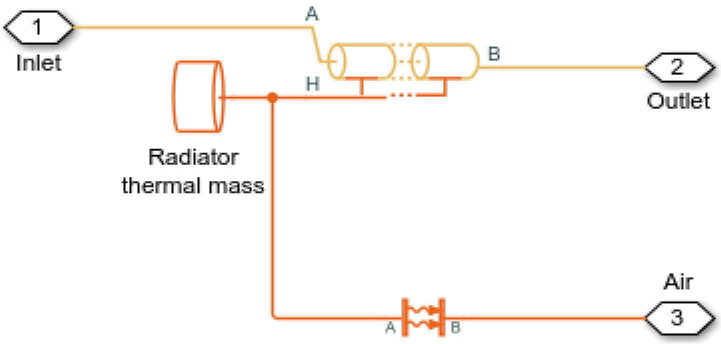
1. Plot temperatures in outdoor and indoor thermal components (see code)
2. Plot compressor power & heat transfers in system (see code)
3. Plot pressure & flow rate of refrigerant (see code)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example

Copyright 2021-2022 The MathWorks, Inc.

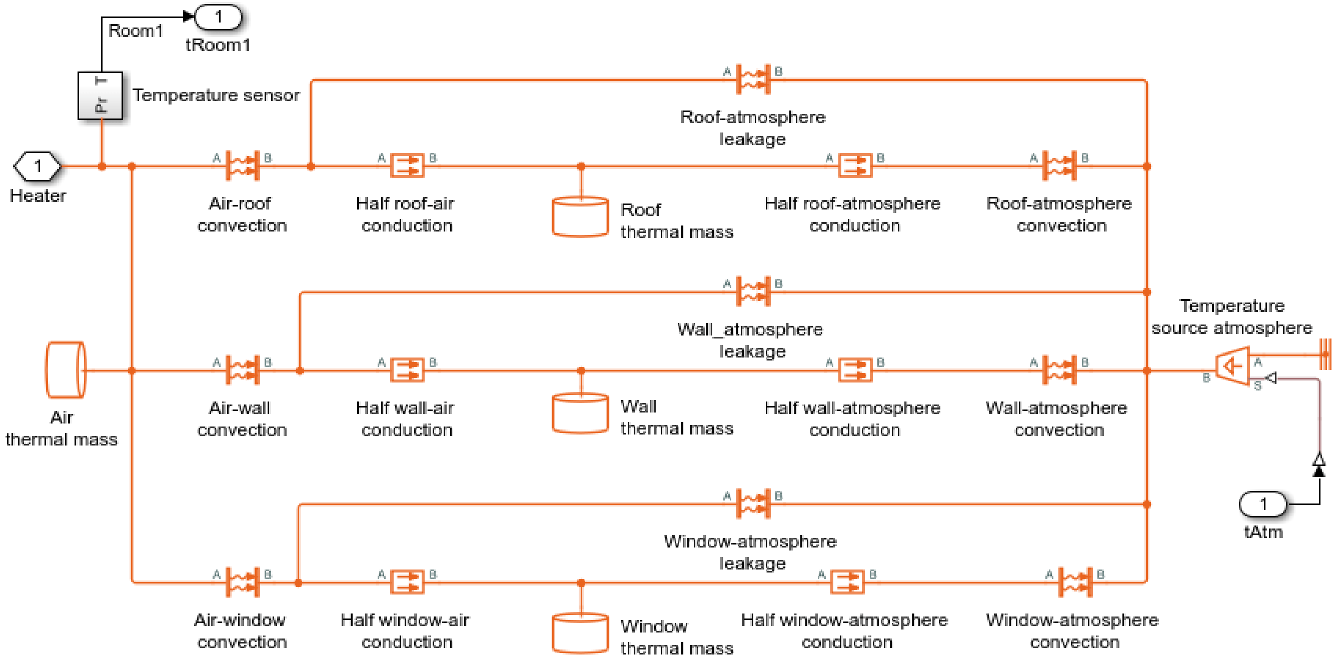
House Thermal Network Subsystem



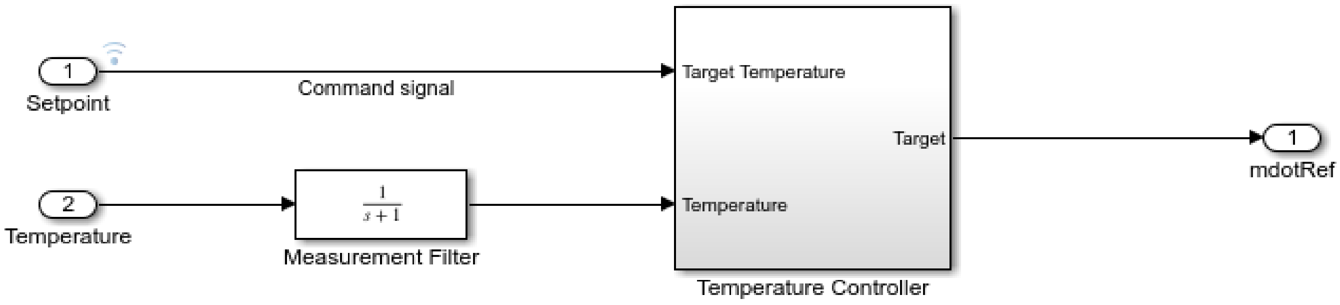
Radiator Subsystem



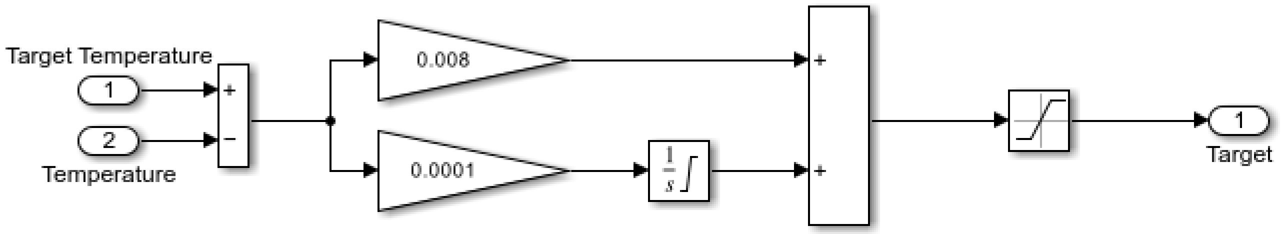
Room Subsystem



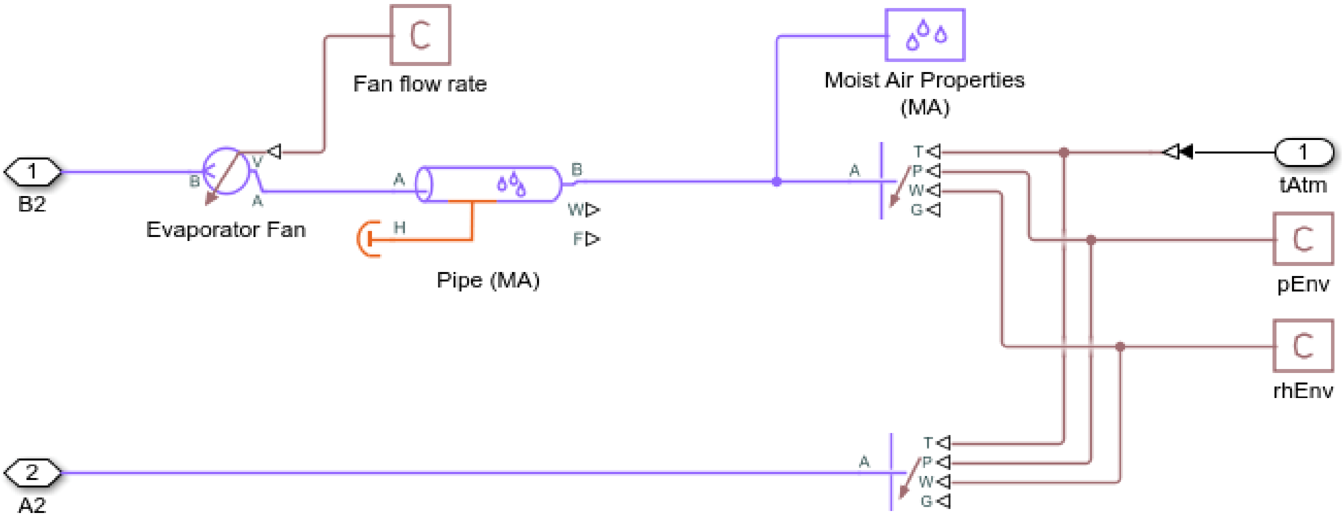
Controller Subsystem



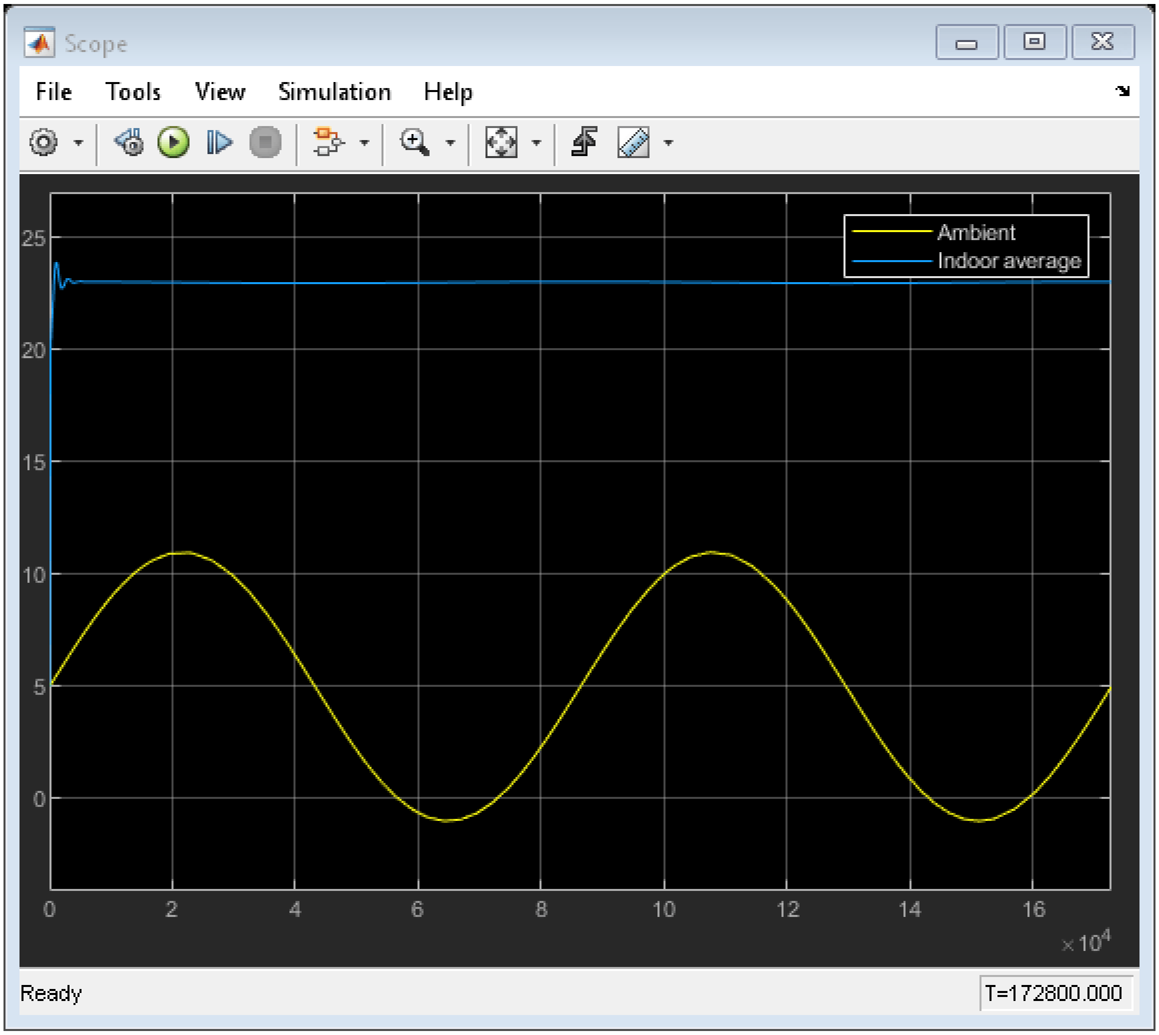
Temperature Controller Subsystem



Environment Subsystem

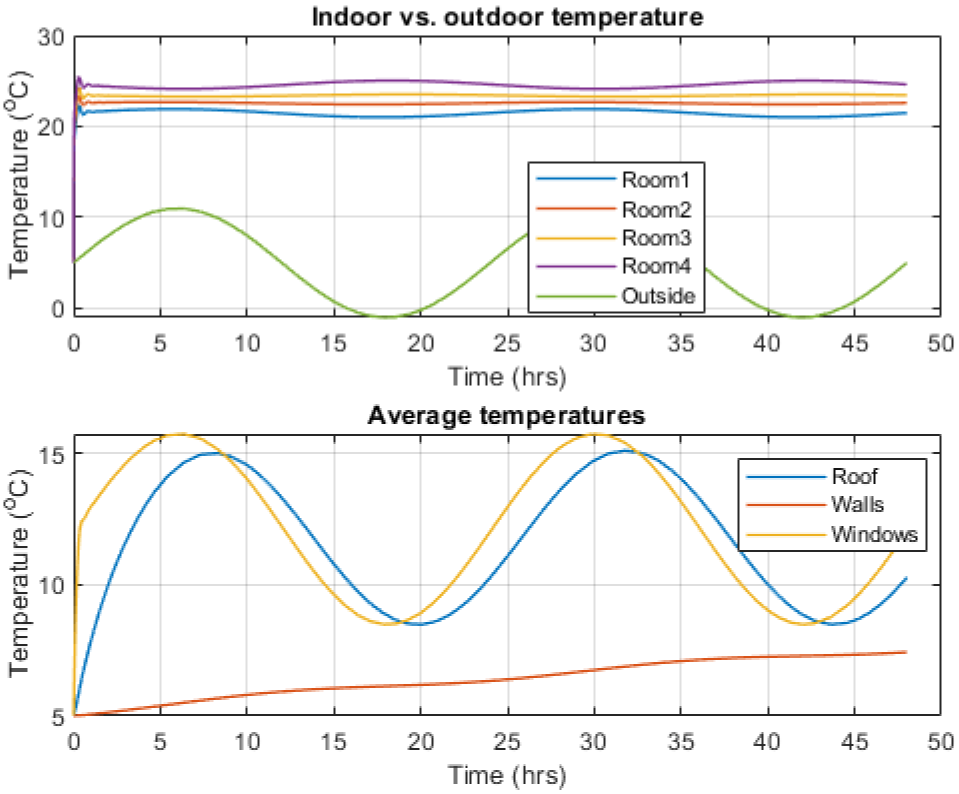


Simulation Results from Scopes

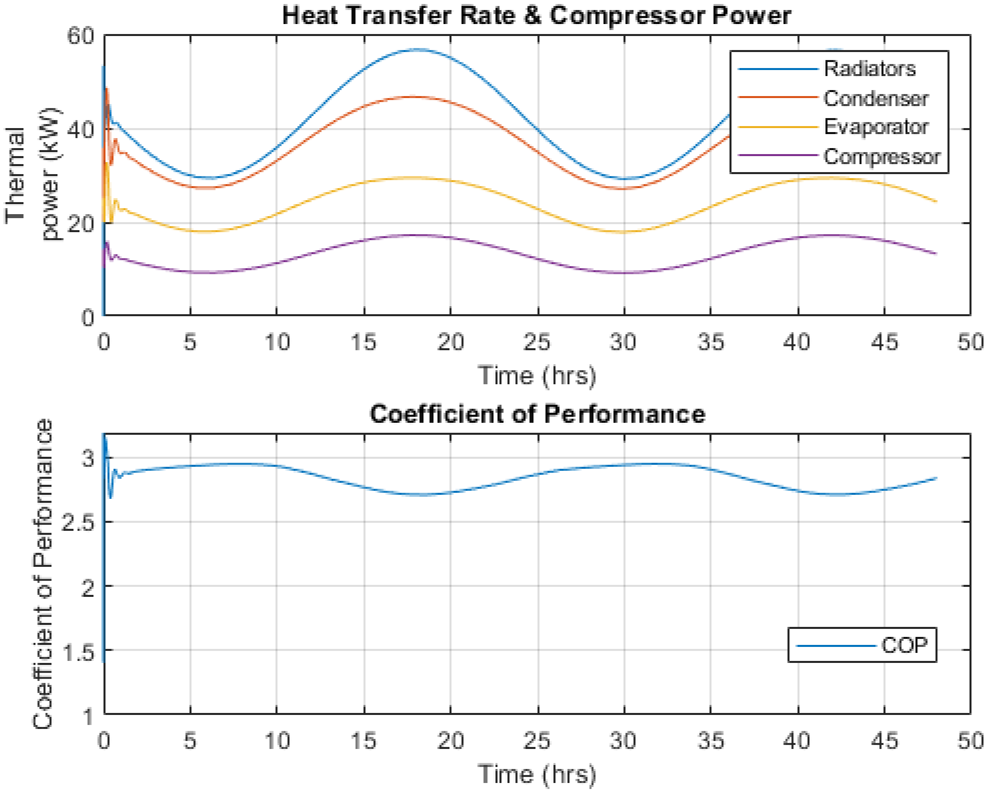


Simulation Results from Simscape Logging

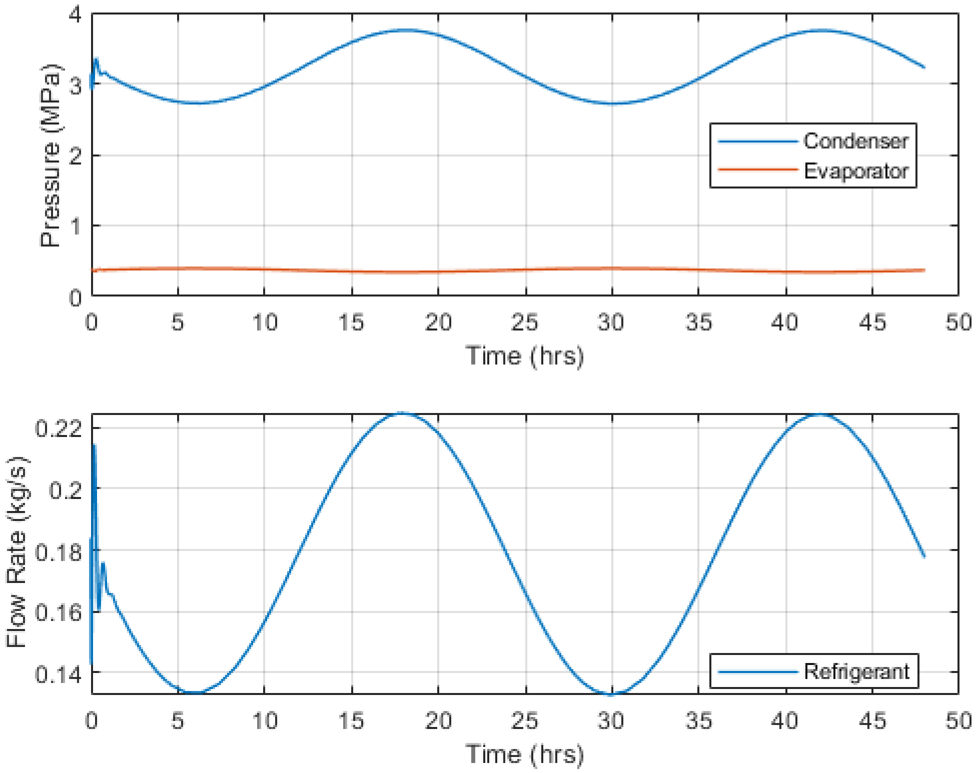
This top subplot shows the temperature variation of each room compared to the outside temperature fluctuations. This bottom subplot shows the average overall temperatures of roof, walls, and windows. The initial indoor temperature of the rooms is equal to the ambient temperature of 5 degC, and the heat pump quickly increases the average indoor temperature to the setpoint temperature of 23 degC.



This top subplot shows the power consumed by the compressor, the heating load of the heat pump system, which is the rate of heat transfer in the condenser, and the rate of heat transfer in the evaporator. This bottom subplot shows coefficient of performance which is the ratio of the heating load and the power consumed. These plots show that more energy is being consumed when heat load is higher, i.e when ambient temperature is lower, and less energy is being consumed when heat load is lower, i.e when ambient temperature is higher, while the coefficient of performance stays nearly constant for the entire duration.



This top subplot shows refrigerant pressure and this bottom subplot shows refrigerant mass flow rate. The high pressure line oscillates between 2.6 MPa to 3.7 MPa and the lower pressure line oscillates between 0.34 MPa to 0.4 MPa. The refrigerant flow rate oscillates between 0.13 kg/s to 0.23 kg/s. The higher pressures and flow rates correspond to lower ambient temperatures, when the heat pump needs to transfer more heat, while the lower pressure and flow rates correspond to higher ambient temperatures, when the heat pump needs to transfer less heat.


Pressure (MPa)
4
3
2
1
0
Condenser
Evaporator
0
5
10
15
20
25
30
35
40
45
50
Time (hrs)
Flow Rate (kg/s)
0.22
0.2
0.18
0.16
0.14
Refrigerant
0
5
10
15
20
25
30
35
40
45
50
Time (hrs)

PEM Fuel Cell System with a Custom Library

This example shows how to model a proton exchange membrane (PEM) fuel cell stack with a custom library and custom Simscape domain. The custom library includes blocks that are tailored to fuel cell modeling. The custom domain models a 4-species gas mixture that contains nitrogen (N2), oxygen (O2), hydrogen (H2), and water vapor (H2O).

The PEM fuel cell generates electrical power by consuming hydrogen and oxygen and producing water vapor. The anode contains nitrogen (N2), water vapor (H2O), and hydrogen (H2), representing the fuel. The hydrogen is stored in the fuel tank at 70 MPa. A pressure-reducing valve releases hydrogen to the fuel cell stack at around 0.16 MPa. Unconsumed hydrogen is recirculated back to the stack.

The cathode contains nitrogen (N2), water vapor (H2O), and oxygen (O2), representing air from the environment. A compressor brings air to the fuel cell stack at a controlled rate to ensure that the fuel cell is not starved of oxygen. A back pressure relief valve maintains a pressure of around 0.16 MPa in the stack and vents the exhaust to the environment.

The temperature and relative humidity in the fuel cell stack must be maintained at an optimal level to ensure efficient operation under various loading conditions. Higher temperatures increase thermal efficiency but reduce relative humidity, which causes higher membrane resistance. Therefore, in this model, the fuel cell stack temperature is kept at 80 degC. The cooling system circulates coolant between the cells to absorb heat and rejects it to the environment via the radiator. The humidifiers saturate the gas with water vapor to keep the membrane hydrated and minimize electrical resistance.

The MEA block is implemented in the Simscape code `FuelCell.ssc`.

The output ports xi of the anode and cathode gas channel blocks provide the gas mole fractions needed to model the fuel cell reaction. The block models the removal of H2 and O2 from the anode and cathode, respectively, and production of H2O due to the chemical reaction. Water vapor travels across the MEA. The heat generated by the reaction is sent through the thermal port H to the connected Thermal Mass block. Refer to the comments in the code for additional details on the implementation.

See also the “PEM Fuel Cell System” example that uses the moist air domain.

References:

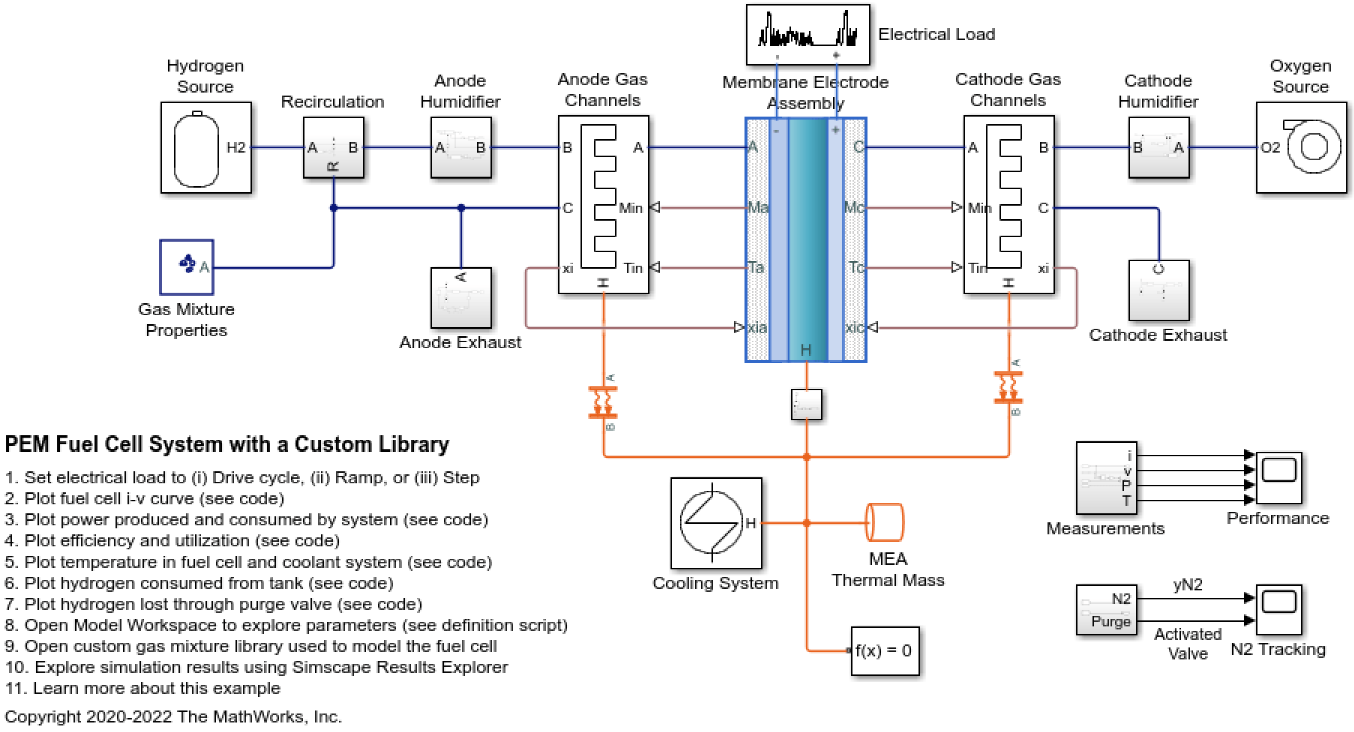
Dutta, Sandip, Sirivatch Shimpalee, and J. W. Van Zee. "Numerical prediction of mass-exchange between cathode and anode channels in a PEM fuel cell." *International Journal of Heat and Mass Transfer* 44.11 (2001): 2029-2042.

EG&G Technical Services, Inc. *Fuel Cell Handbook (Seventh Edition).* US Department of Energy, Office of Fossil Energy, National Energy Technology Laboratory, 2004.

Pukrushpan, Jay T., Anna G. Stefanopoulou, and Huei Peng. *Control of fuel cell power systems: principles, modeling, analysis and feedback design.* Springer-Verlag London, 2004.

Spiegel, Colleen. *PEM fuel cell modeling and simulation using MATLAB.* Elsevier, 2008.

Model

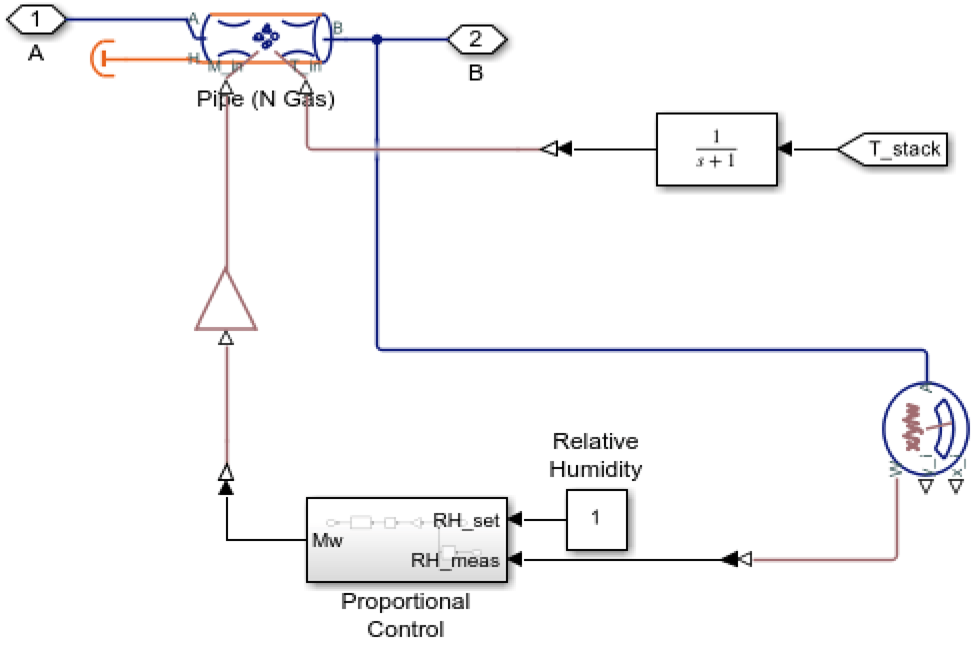


PEM Fuel Cell System with a Custom Library

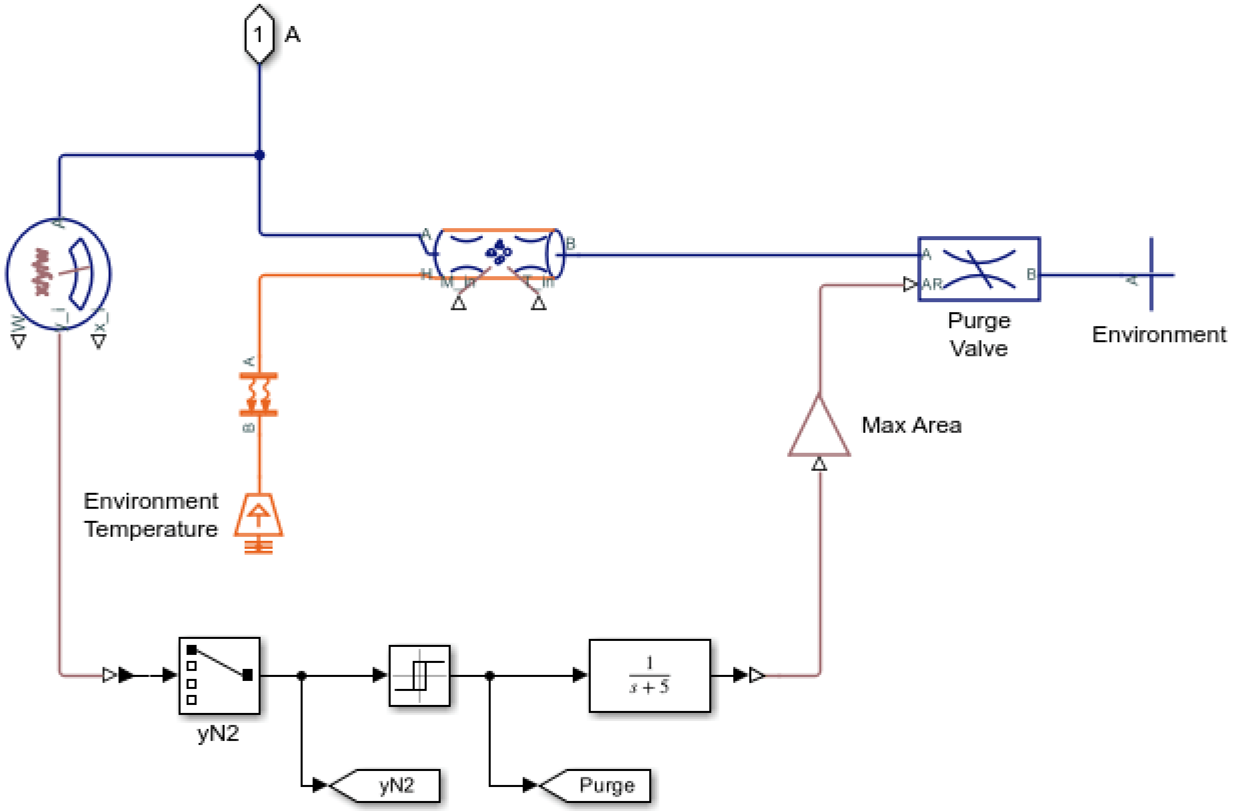
1. Set electrical load to (i) Drive cycle, (ii) Ramp, or (iii) Step
2. Plot fuel cell i-v curve (see code)
3. Plot power produced and consumed by system (see code)
4. Plot efficiency and utilization (see code)
5. Plot temperature in fuel cell and coolant system (see code)
6. Plot hydrogen consumed from tank (see code)
7. Plot hydrogen lost through purge valve (see code)
8. Open Model Workspace to explore parameters (see definition script)
9. Open custom gas mixture library used to model the fuel cell
10. Explore simulation results using Simscape Results Explorer
11. Learn more about this example

Copyright 2020-2022 The MathWorks, Inc.

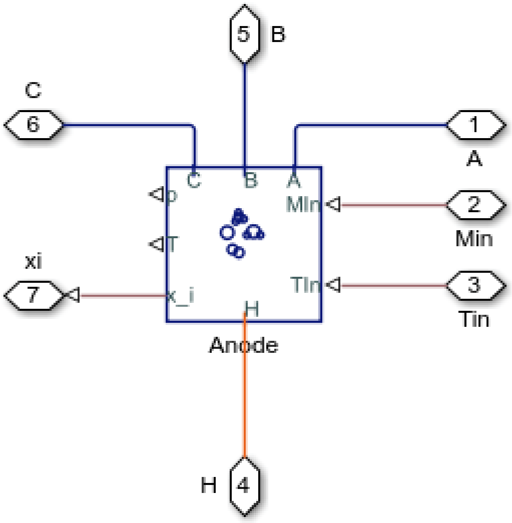
Anode Humidifier Subsystem



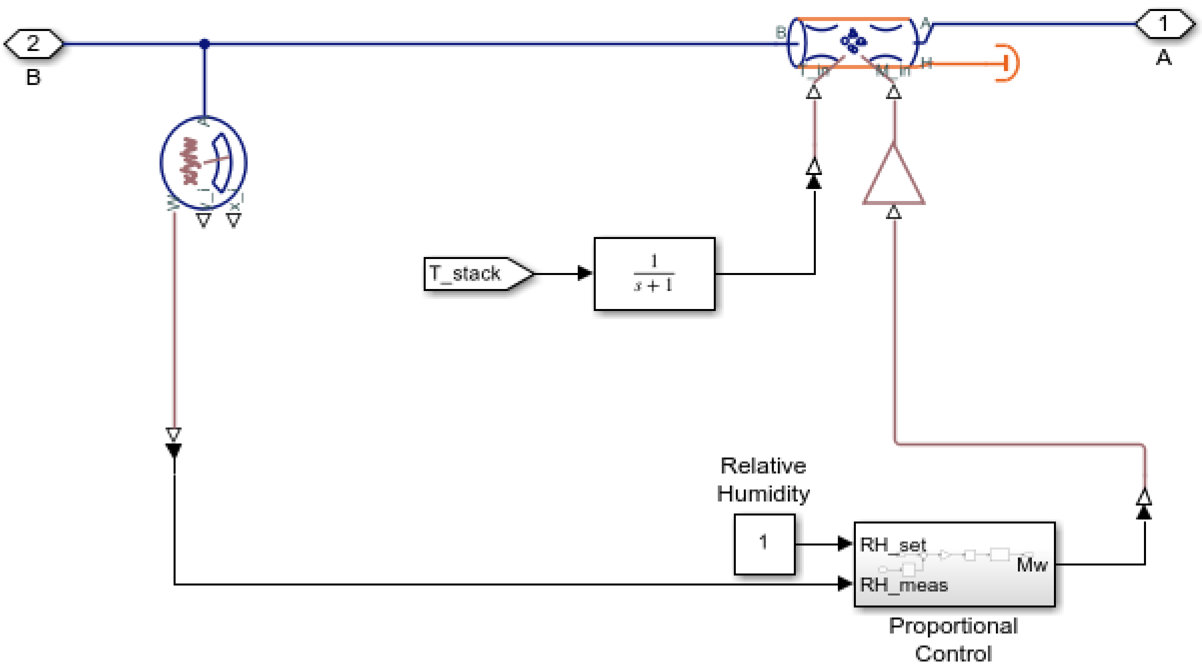
Anode Exhaust Subsystem



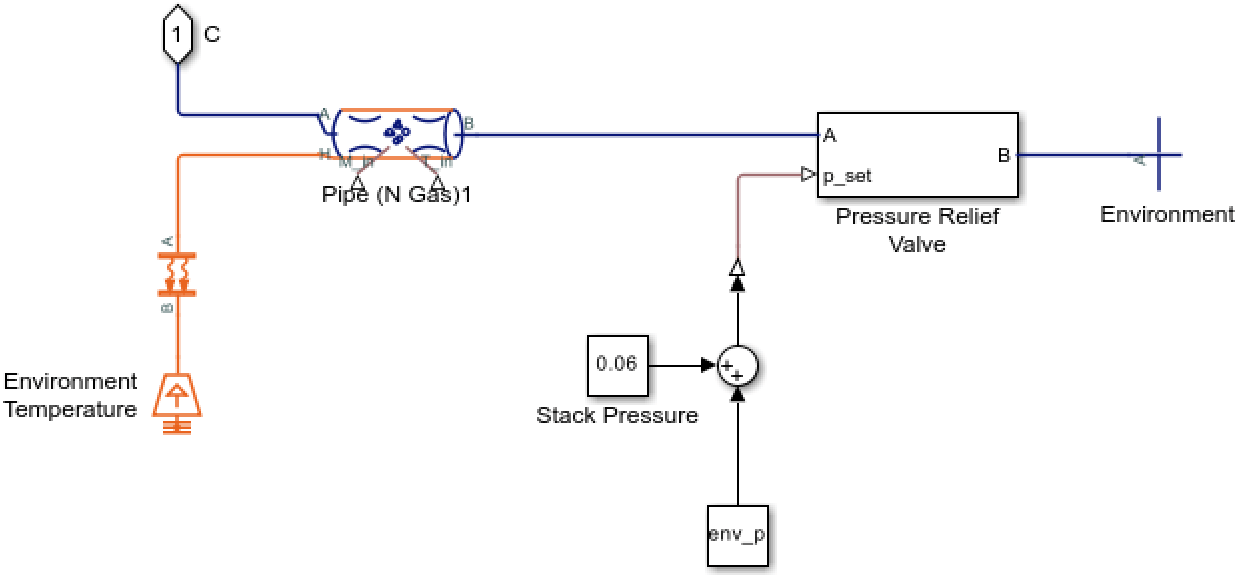
Anode Gas Channels Subsystem



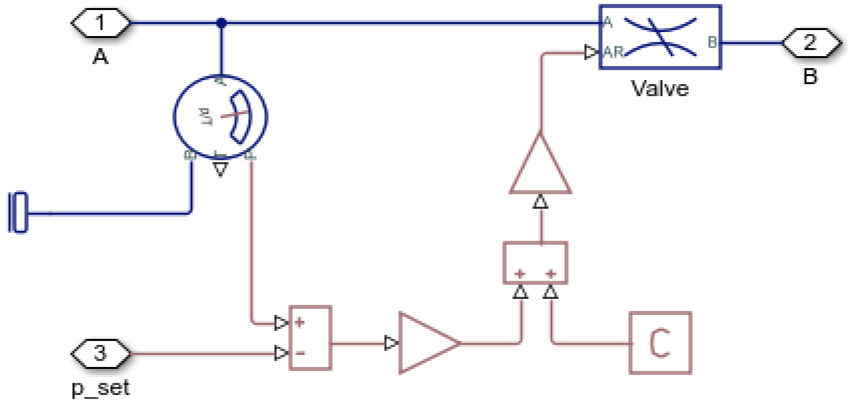
Cathode Humidifier Subsystem



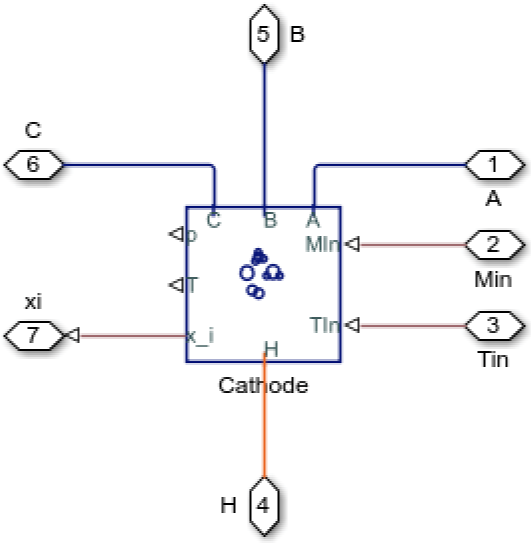
Cathode Exhaust Subsystem



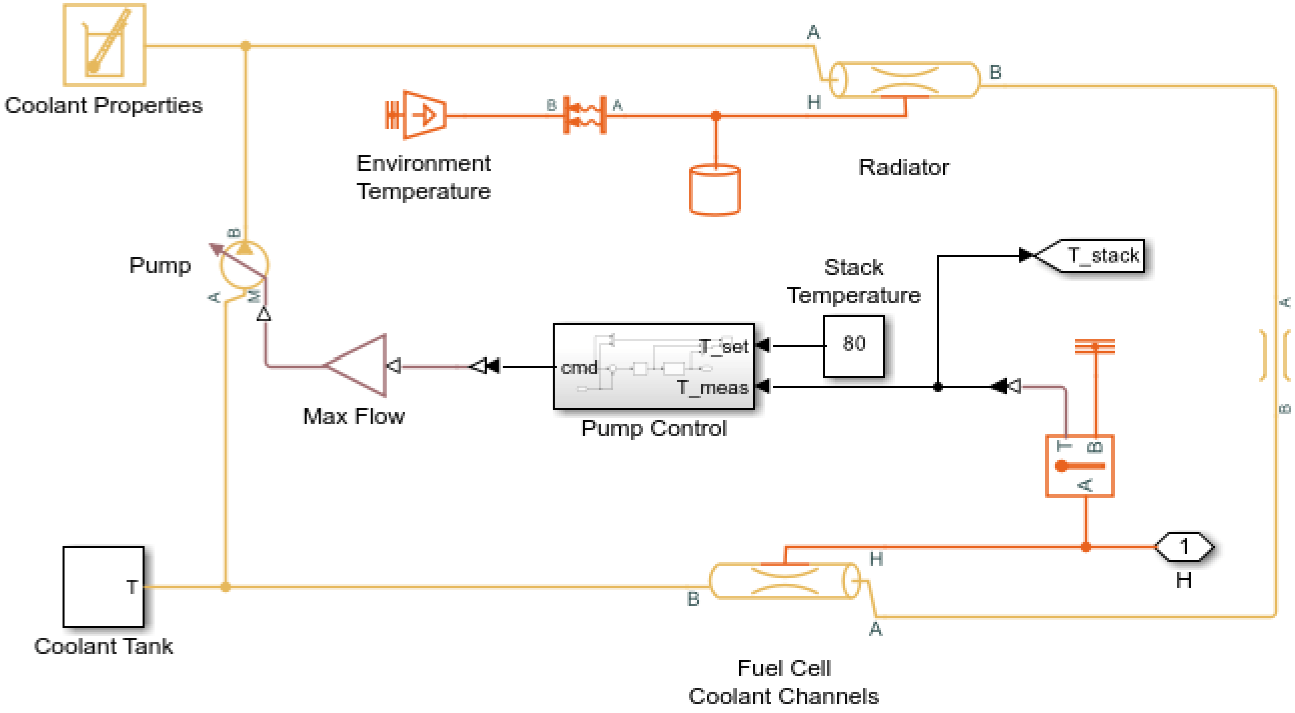
Pressure Relief Valve Subsystem



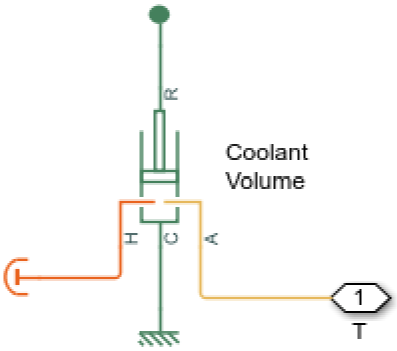
Cathode Gas Channels Subsystem



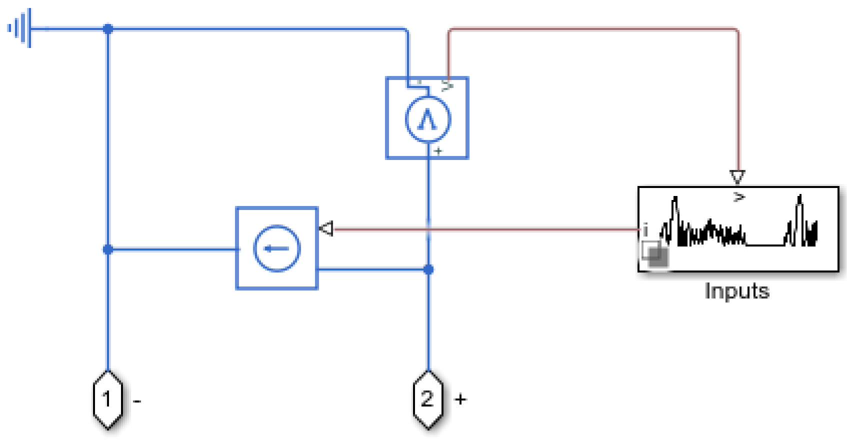
Cooling System Subsystem



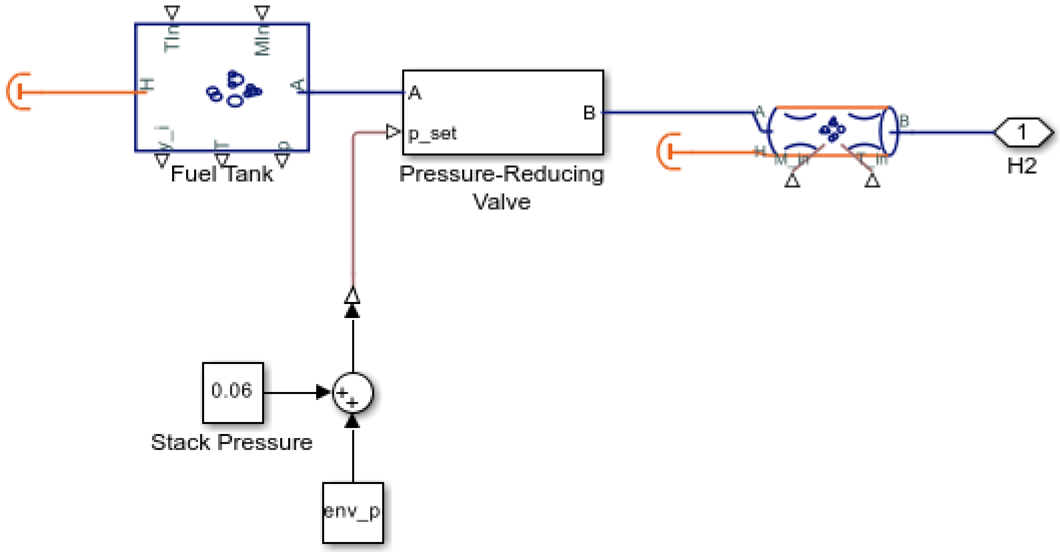
Coolant Tank Subsystem



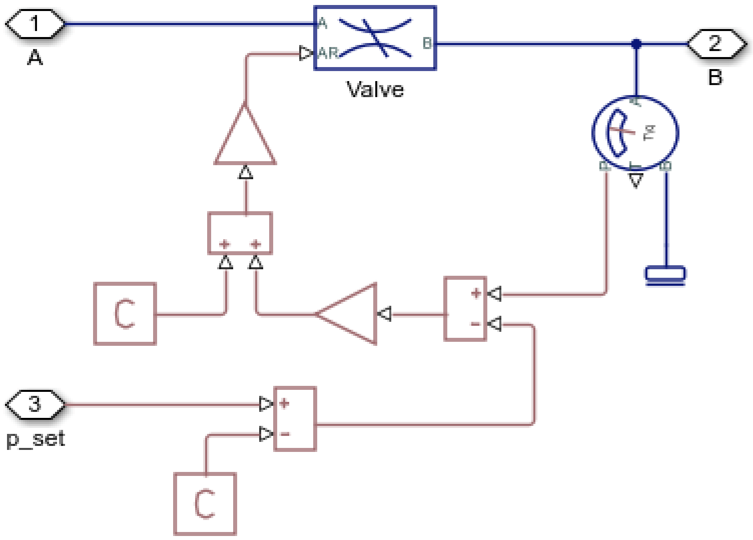
Electrical Load Subsystem



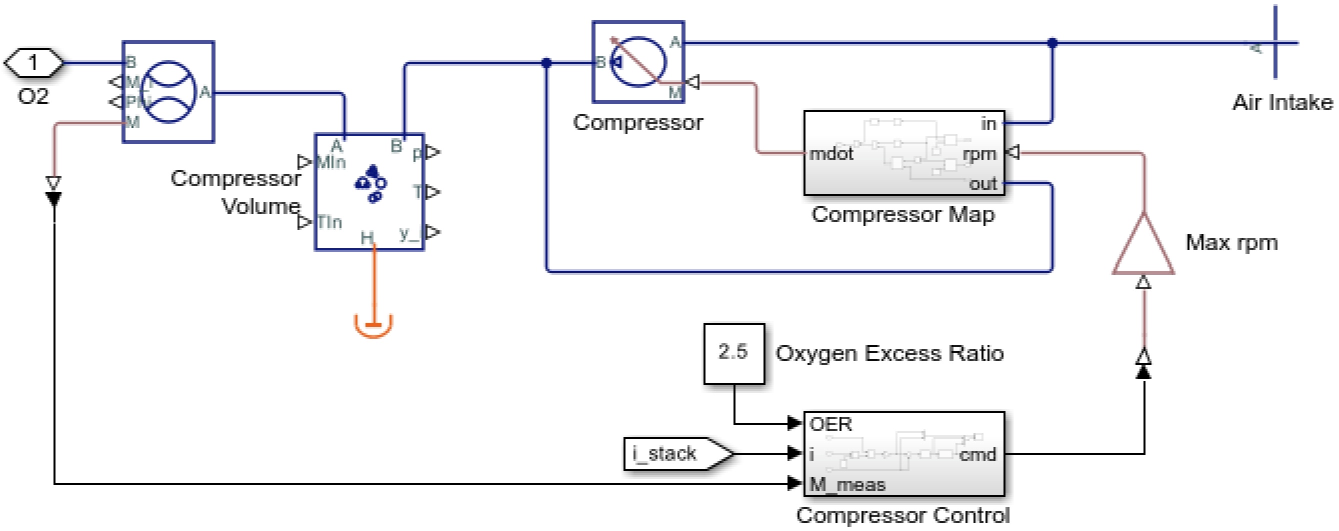
Hydrogen Source Subsystem



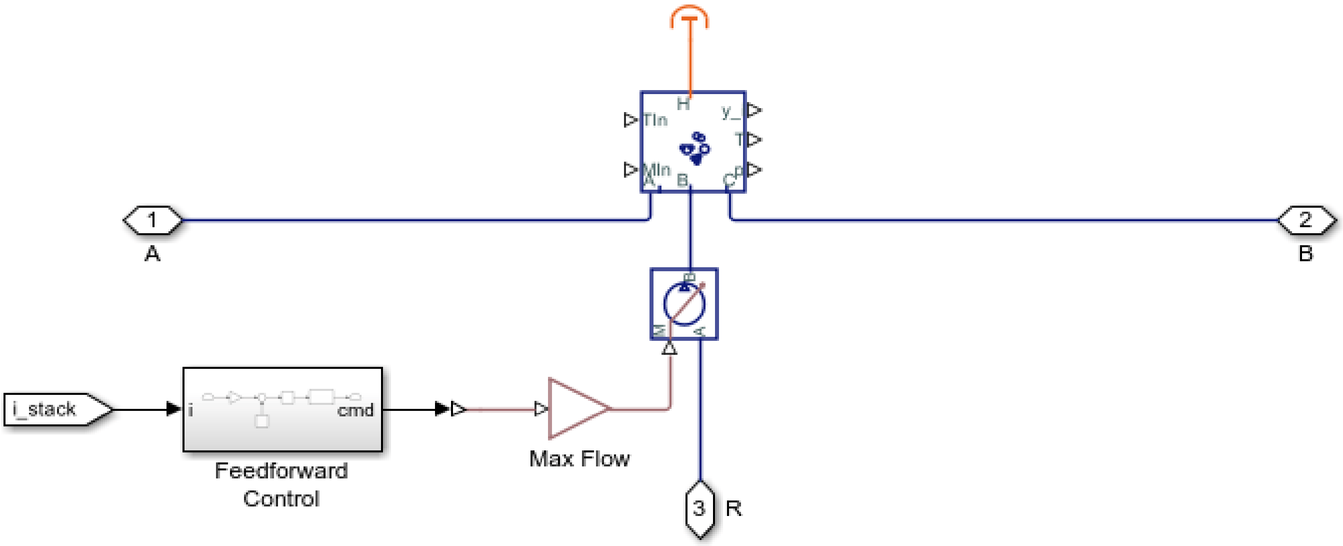
Pressure-Reducing Valve Subsystem



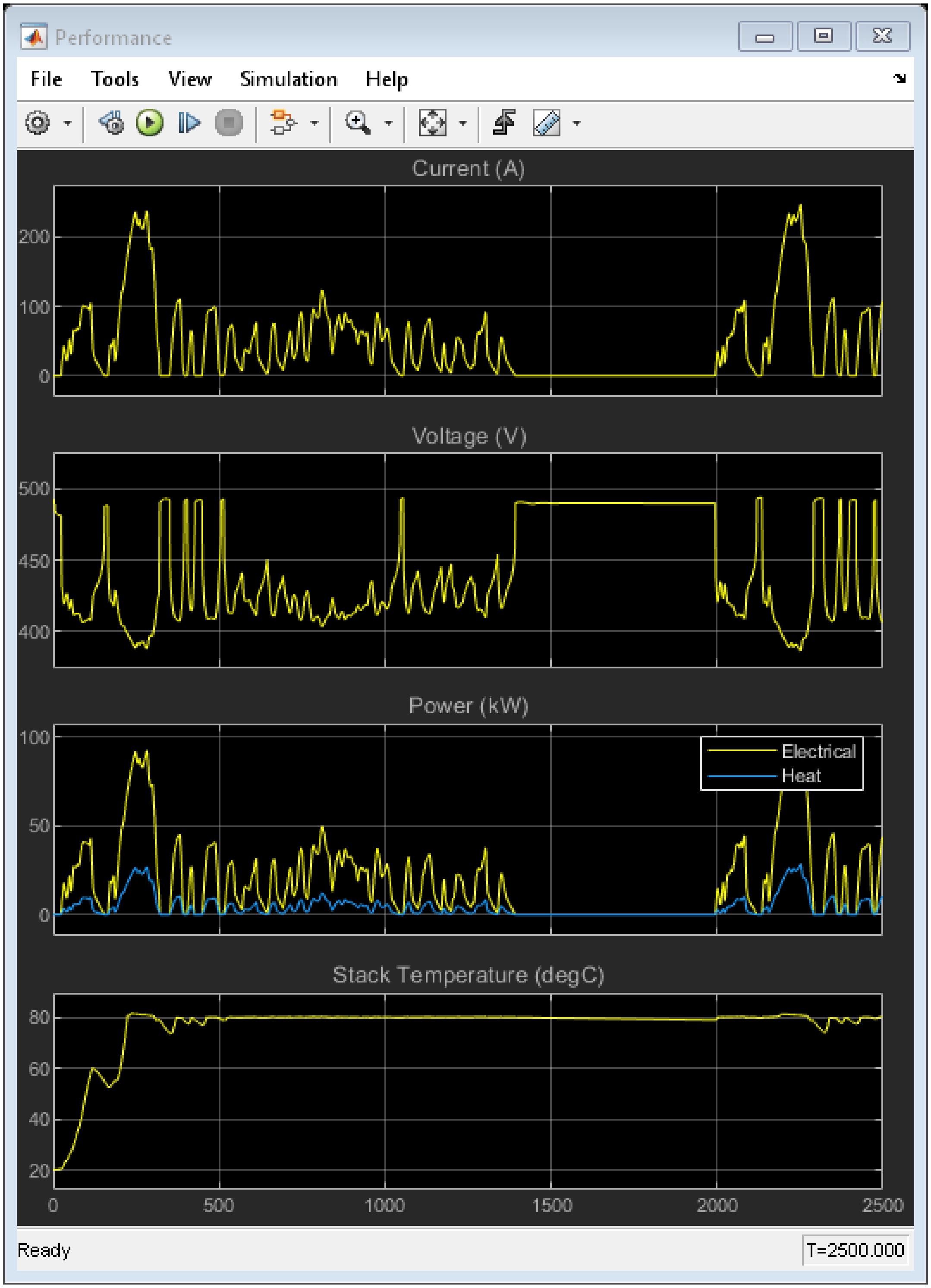
Oxygen Source Subsystem

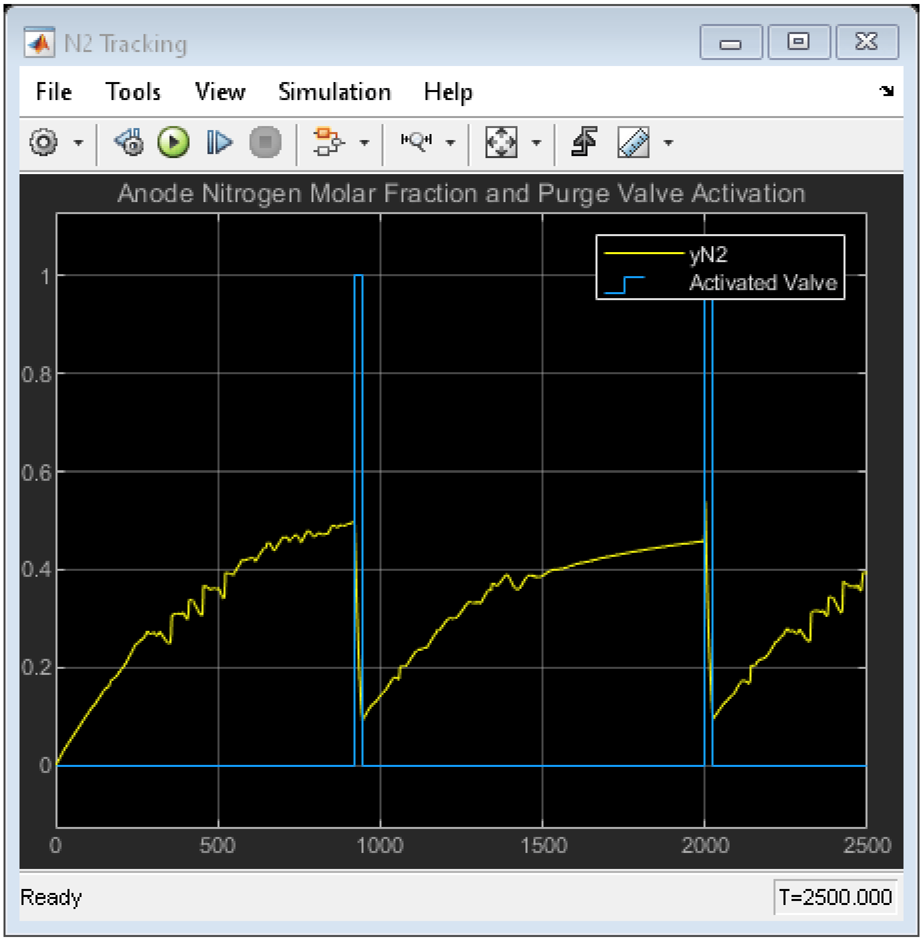


Recirculation Subsystem



Simulation Results from Scopes

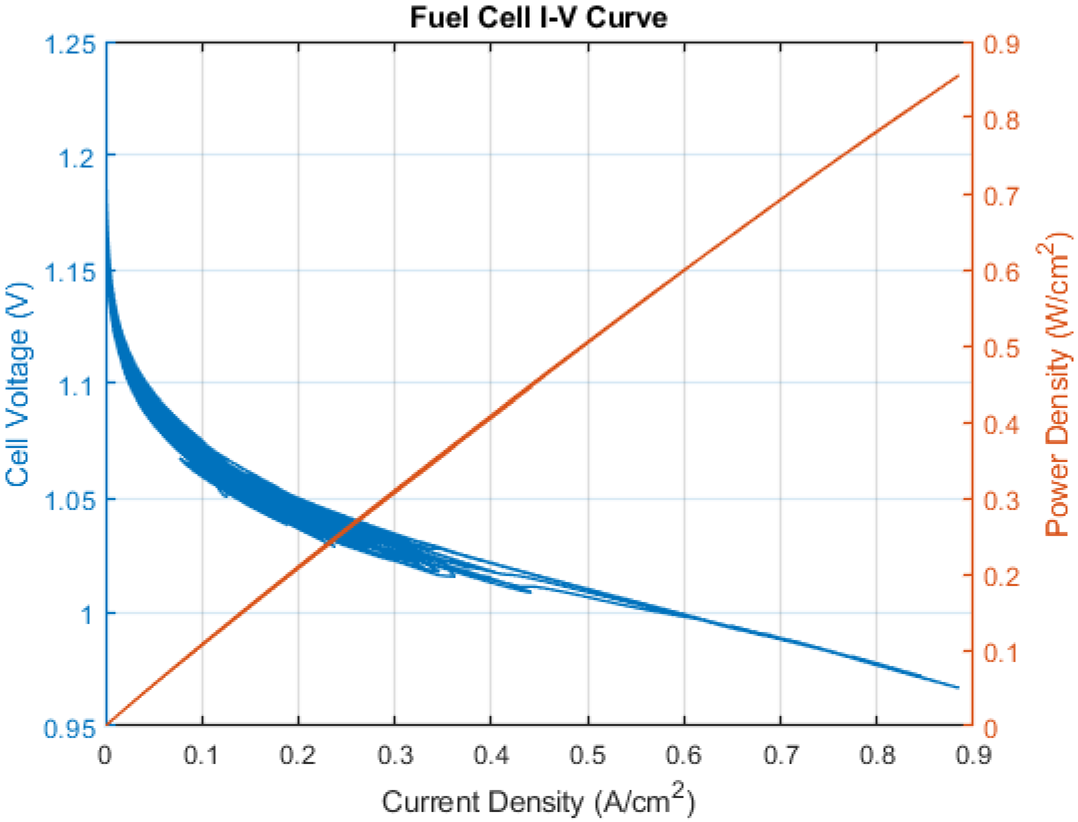




Simulation Results from Simscape Logging

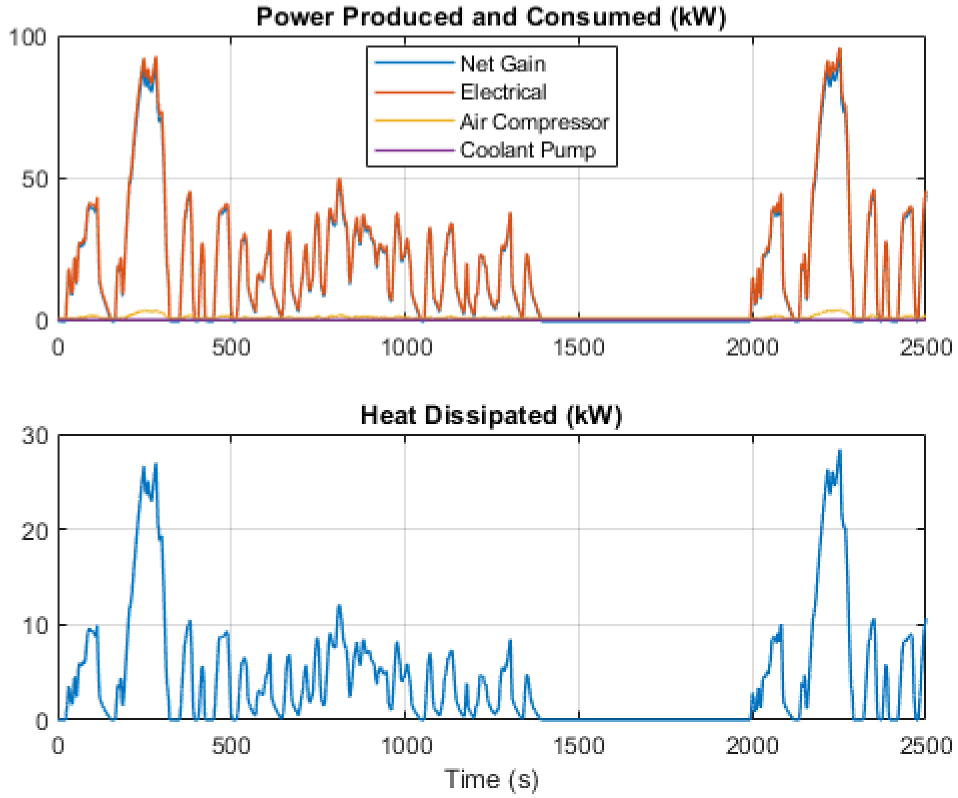
This plot shows the current-voltage (i-v) curve of a fuel cell in the stack. As the current ramps up, an initial drop in voltage occurs due to electrode activation losses, followed by a gradual decrease in voltage due to Ohmic resistances. Near maximum current, a sharp drop in voltage occurs due to gas-transport-related losses.

This plot also shows the power produced by the cell. When the ramp scenario is selected, the power increases until a maximum power output, then decreases due to the high losses near maximum current.



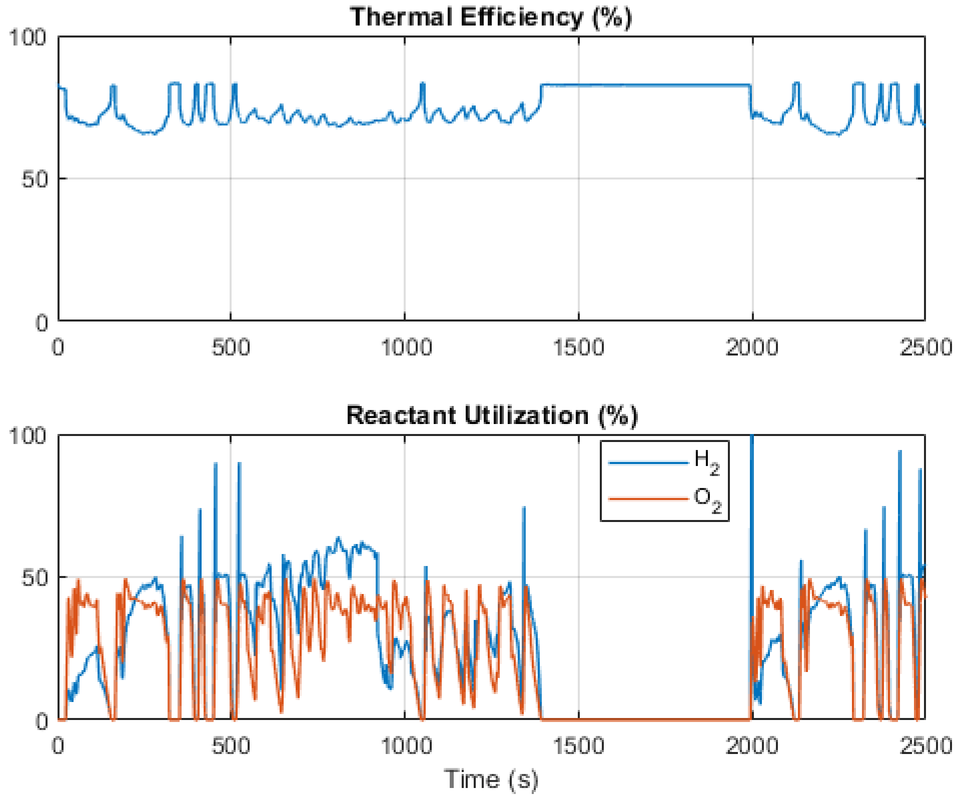
This plot shows the electrical power produced by the fuel cell stack as well as the power consumed by the cathode air compressor and the coolant pump to maintain stable and efficient system operation. As a result, the net power produced by the system is a few percent less than the power produced by the stack. Note that this model assumes an isentropic compressor. Accounting for compressor efficiency would decrease net power gain by another couple of percent.

This plot also shows the excess heat generated by the fuel cell stack, which must be removed by the cooling system. The maximum power produced by the fuel cell stack is 95 kW.



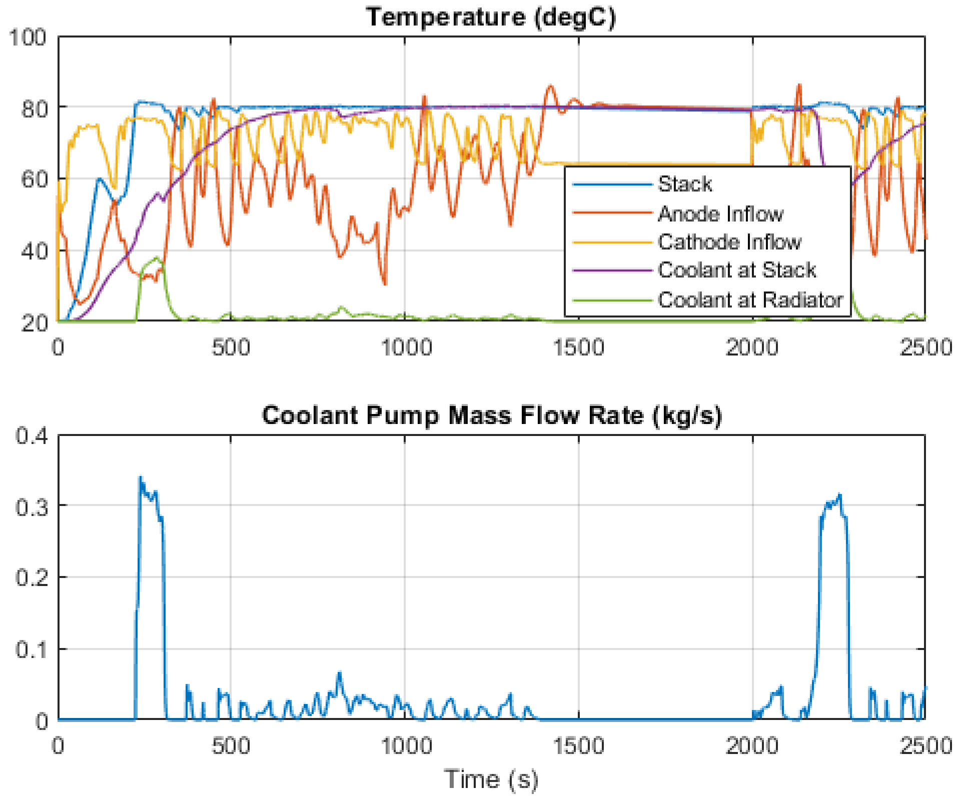
This plot show the thermal efficiency of the fuel cell and its reactant utilization fraction. The thermal efficiency indicates the fraction of the hydrogen fuel's energy that the fuel cell has converted to useful electrical work. The theoretical maximum efficiency for a PEM fuel cell is 83%. However, actual efficiency is around 60% due to internal losses. Near maximum current, the efficiency drops to around 45%.

The reactant utilization is the fraction of the reactants, H2 and O2, flowing into the fuel cell stack that has been consumed by the fuel cell. While higher utilization makes better use of the gases flowing through the fuel cell, it decreases the concentration of the reactants and thus reduces the voltage produced. Unconsumed O2 is vented to the environment, but unconsumed H2 is recirculated back to the anode to avoid waste. H2 is periodically purged to remove contaminants.

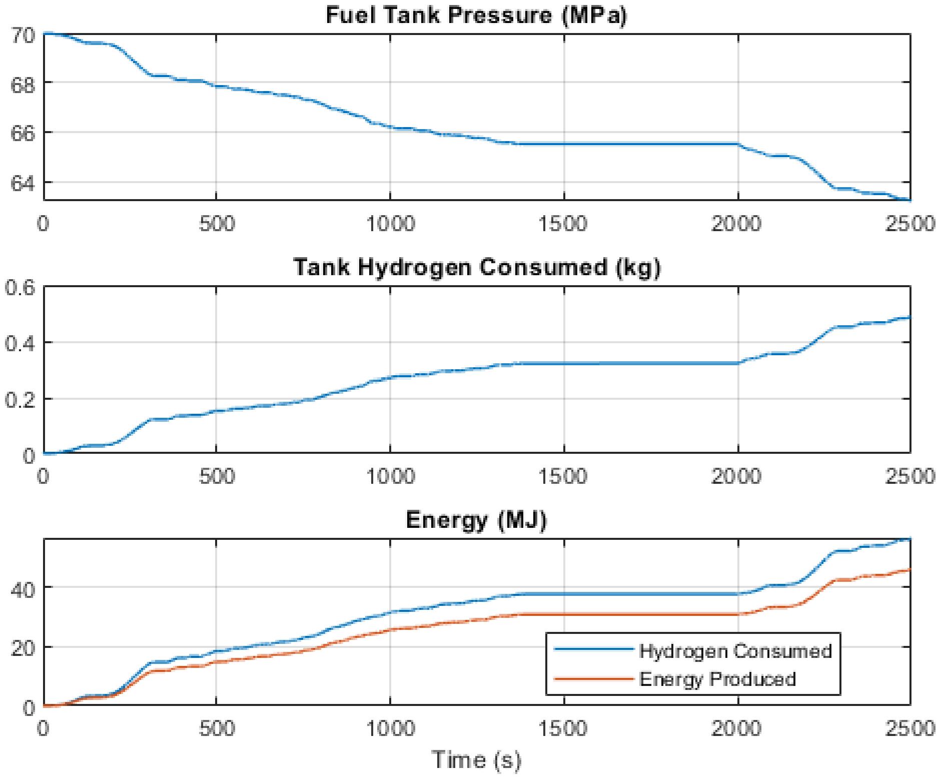


This plot shows the temperatures at various locations in the system. The fuel cell stack temperature is maintained at a maximum of 80 degC by the cooling system. Fuel flowing to the anode is warmed by the recirculated flow. Air flowing to the cathode is warmed by the compressor.

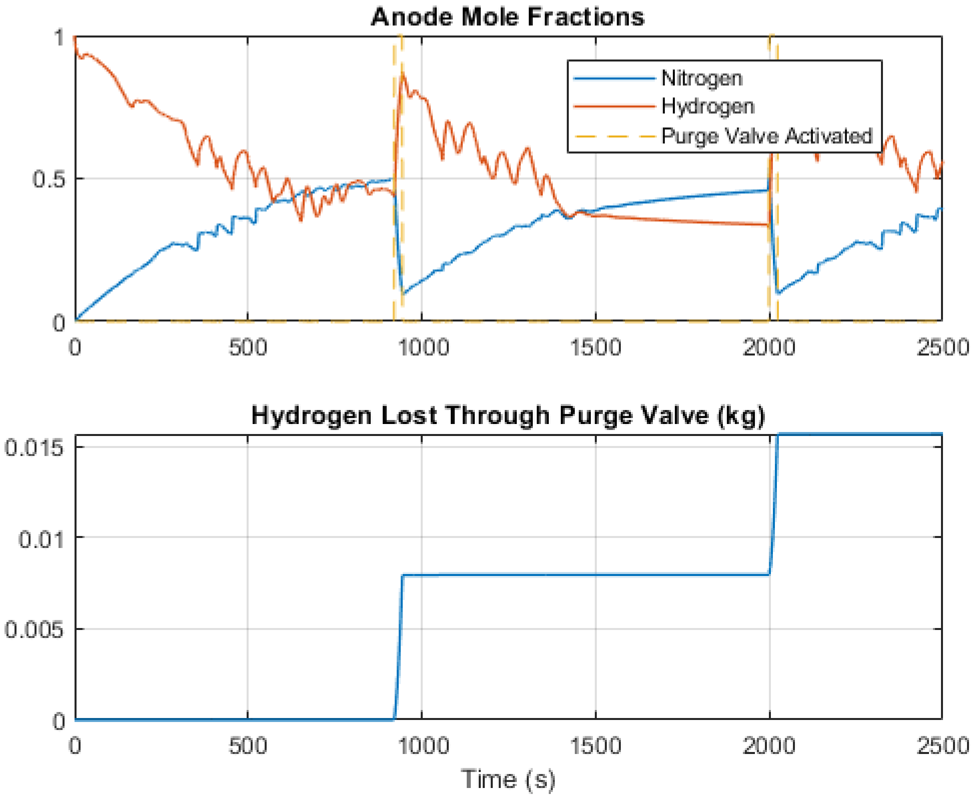
Maintaining an optimal temperature is critical to the operation of the fuel cell because higher temperatures lower the relative humidity which increases the membrane resistance. In this model, the cooling system is operated by a simple control of the coolant pump flow rate. The plot shows the temperature of the coolant after it has absorbed heat from the fuel cell stack and after it has rejected heat in the radiator.



This plot shows the mass of hydrogen used during operation and the corresponding decrease in the hydrogen tank pressure. The energy of the consumed hydrogen fuel is converted to electrical energy.



This plot shows the molar concentrations of nitrogen and hydrogen in the fuel cell anode. Higher concentrations of nitrogen correspond to lower concentrations of hydrogen and lower fuel cell performance. This system has a purge valve for the purpose of maintaining high hydrogen concentration in the anode. The valve opens when the nitrogen molar concentration in the anode exceeds 50% and closes when it returns to 10%. When the purge valve opens to expel nitrogen, it unfortunately also expels hydrogen. Therefore, fuel cell purge valves typically require carefully designed control strategies.

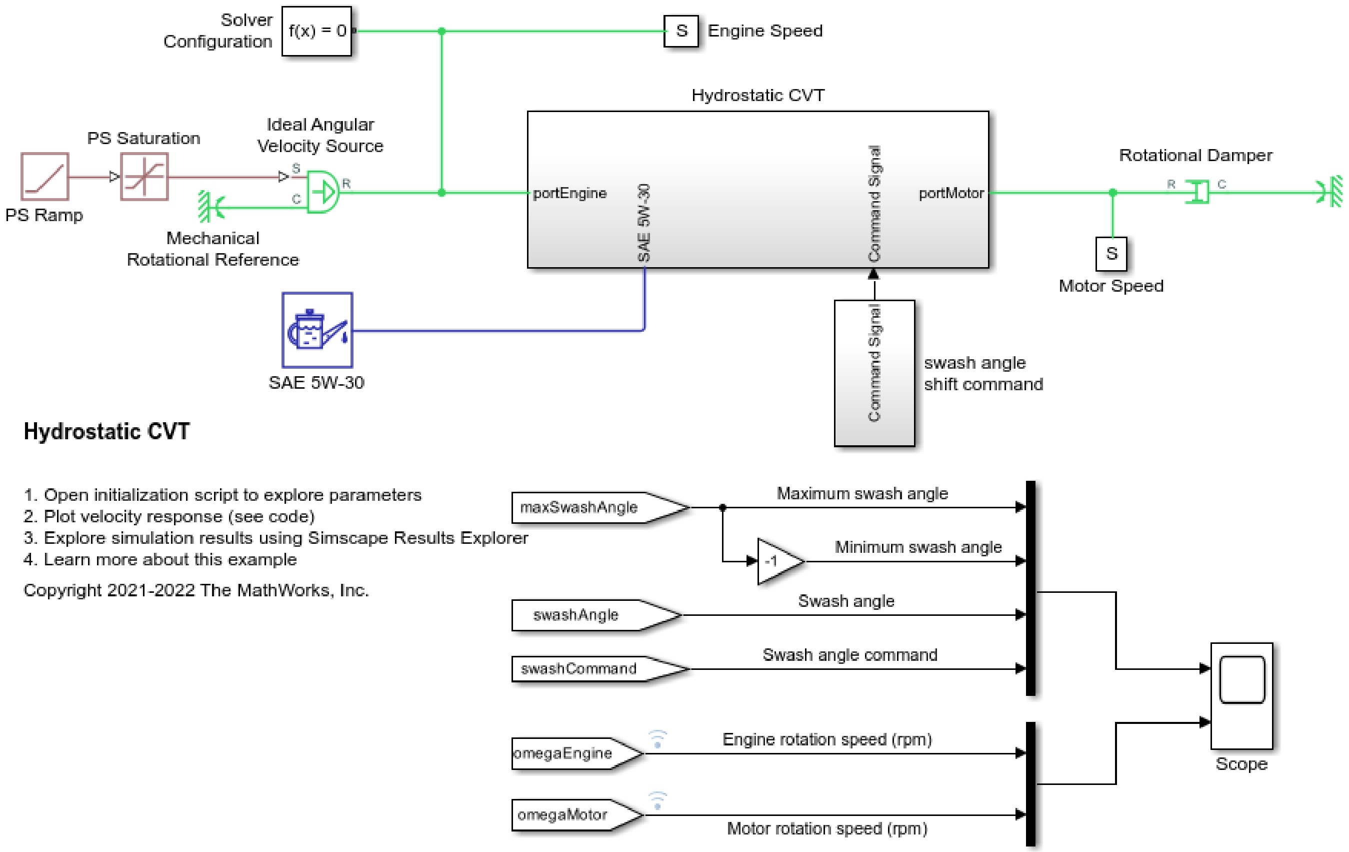

Anode Mole Fractions
Nitrogen
Hydrogen
Purge Valve Activated
1
0.5
0
0
500
1000
1500
2000
2500
Hydrogen Lost Through Purge Valve (kg)
0.015
0.01
0.005
0
0
500
1000
1500
2000
2500
Time (s)

Hydrostatic CVT

This example shows how to model, parameterize, and test a hydrostatic continuously variable transmission (CVT) with a swash angle shift command. When you run the plot function, it generates a comparison plot between the engine and the achieved rotational velocity in the hydraulic axial piston motor with respect to the time. Construction and agricultural equipment manufacturers use these transmissions.

Model

The following figure shows the model of a hydrostatic CVT. Here, connection portEngine is a physical signal port associated with the velocity source representing the engine. Connection portMotor is a physical signal port associated with the axial piston motor. Connection SAE 5W-30 is a physical signal port associated with the fluids properties of the hydraulic network. Connection Command Signal is a simulink input port associated with the axial piston pump swash plate angle displacement command. Driver provides the swash displacement command through speed pedal.



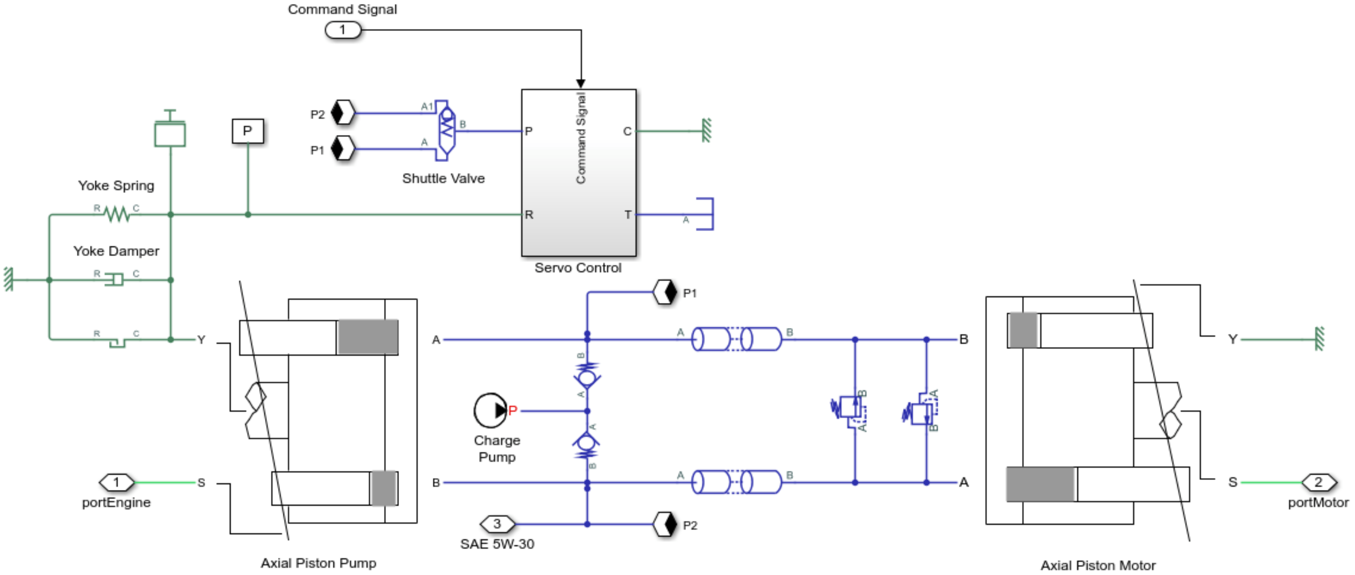
Hydrostatic CVT

1. Open initialization script to explore parameters
2. Plot velocity response (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2021-2022 The MathWorks, Inc.

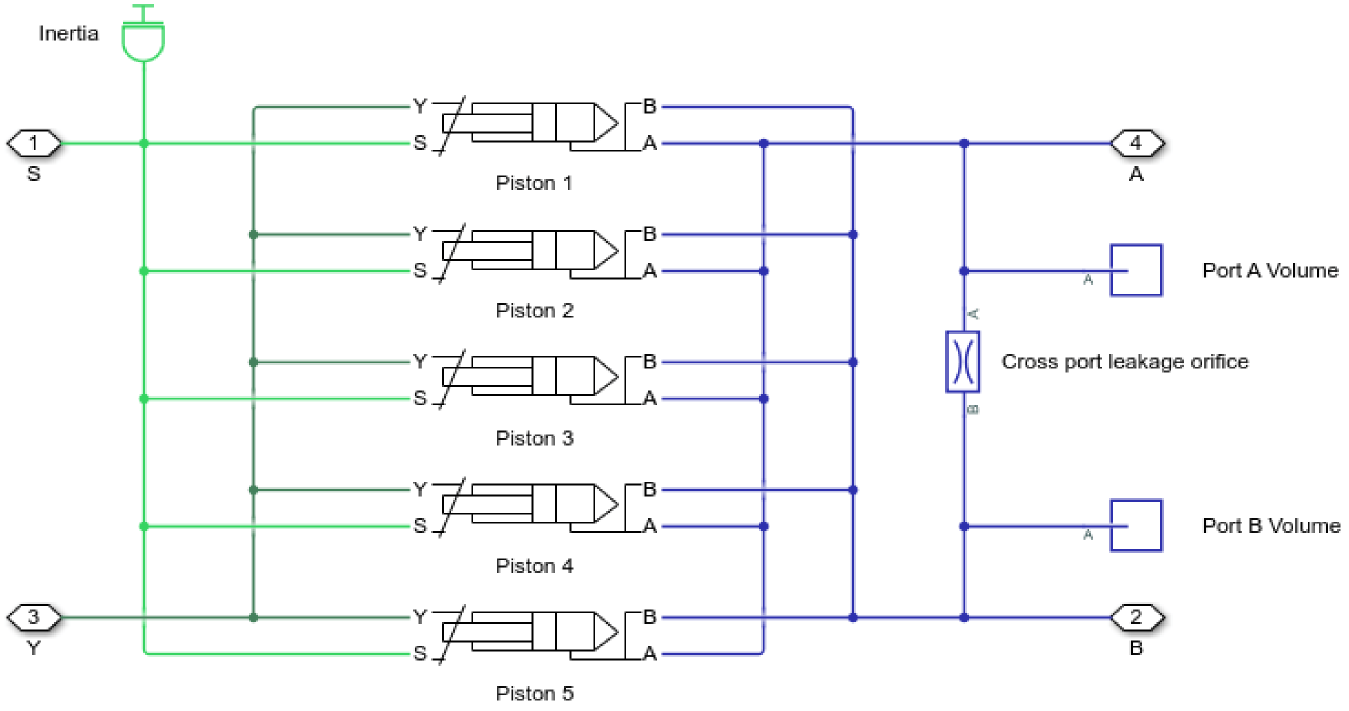
Hydrostatic CVT Subsystem

This subsystem demonstrates how to model and integrate the components of the hydrostatic CVT. It includes the axial piston pump, the axial piston motor, the servo control subsystem, the charge pump, the check valves, the pressure relief valves, the shuttle valve, the connection pipes and the spring, mass and damper of the servo cylinder. The charge pump maintains the pressure in the lower pressure hydraulic line between the motor and the pump at 20 bar. The charge pump required pressure is determined based on the hydraulic actuation application. The check valves connect the charge pump to the minimum pressure hydraulic line between the motor and the pump. The relief valves connect the high pressure hydraulic line to the low pressure hydraulic line between the motor and the pump. The output speed of the motor depends on the displacement of the pump, the pump input rotational speed, the leakages in the hydraulic network, and the displacement of the motor. The direction of the motor movement depends on the pump's swash angle sign. The motor rotates in positive/ forward direction for positive swash angle and rotates in the negative/reverse direction for negative swash angle of the pump.



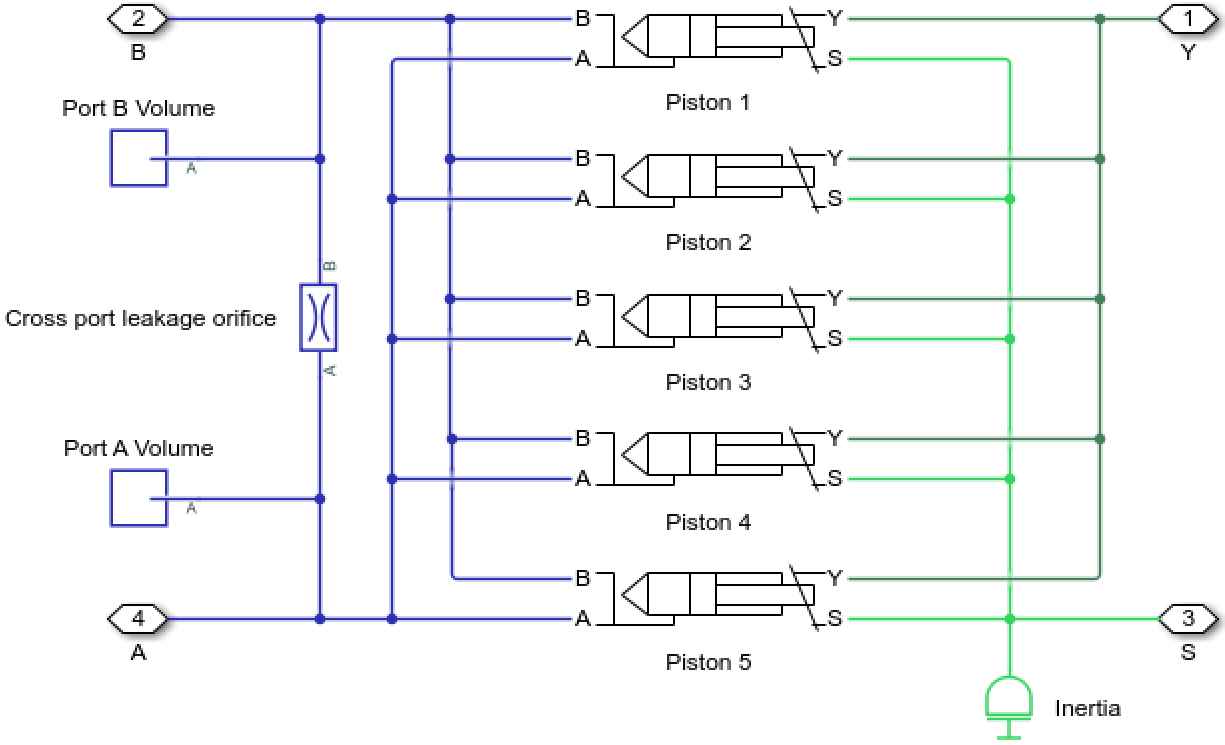
Axial Piston Pump Subsystem

This subsystem demonstrates how to model the axial piston pump. The hydraulic pump provides hydraulic power to the motor. The subsystem includes the pump inertia, the piston subsystems, the porting volume and the cross port leakage.



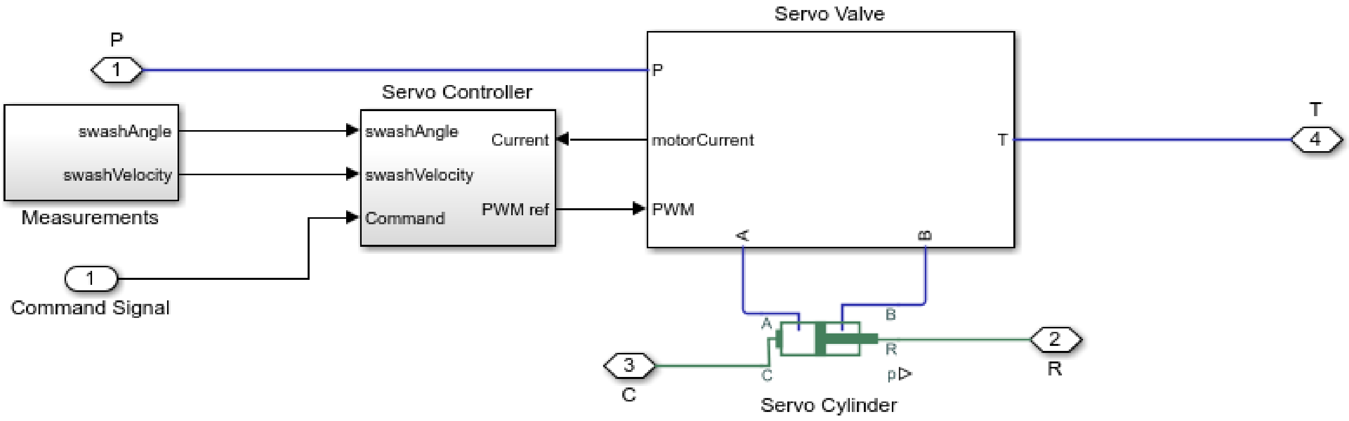
Axial Piston Motor Subsystem

This subsystem demonstrates how to model the axial piston motor. The hydraulic motor converts the hydraulic power to the mechanical power. The subsystem includes the motor inertia, the piston subsystems, the porting volume and the cross port leakage.



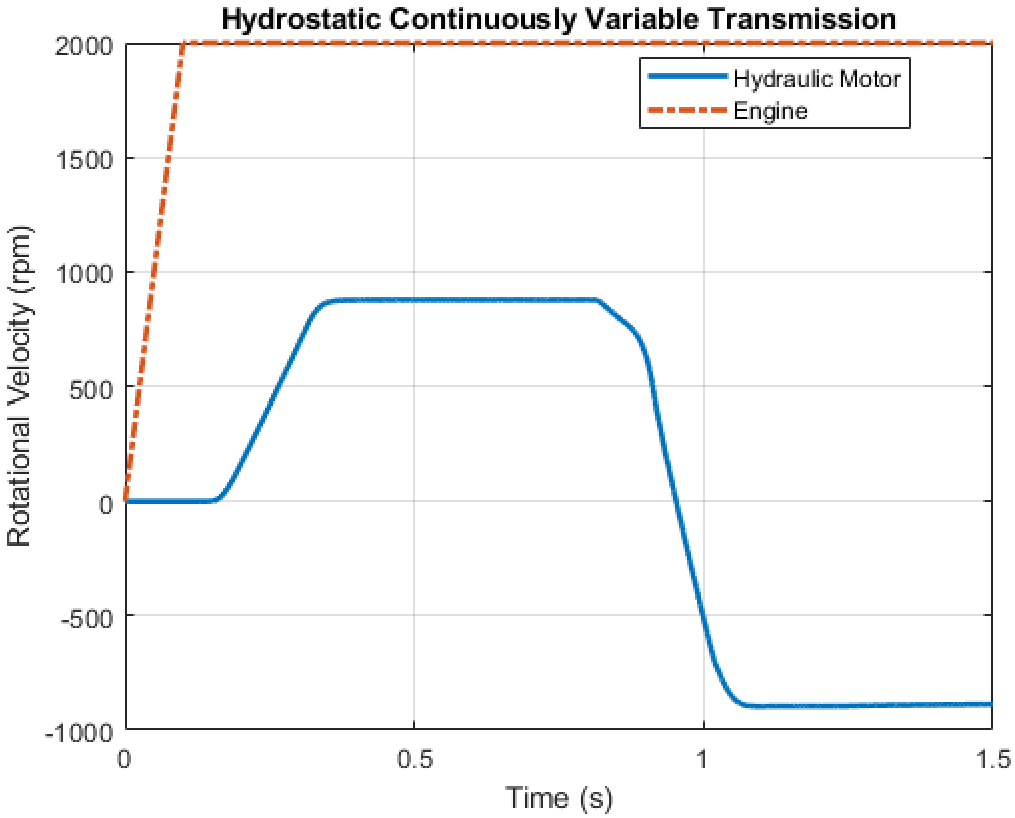
Servo Control Subsystem

This subsystem demonstrates how to model the servo cylinder controller. It includes the measurements, the servo controller, the servo valve subsystem, and the servo cylinder. The measurements subsystem computes the swash plate angle and the swash plate velocity. The servo controller subsystem models the closed loop cascade controller for the servo valve. The cascade control has three control loops. The inner loop controls the motor current, the middle loop controls the actuator velocity, and the outer loop controls the swash plate angle. The servo valve subsystem models a two stage flapper nozzle servo valve. The servo valve actuates the servo cylinder to control the swash angle of the axial piston pump.



Simulation Results from Simscape™ Logging

This model generates a comparison plot between the engine and the achieved velocity in the hydraulic motor with respect to the time. The system can achieve infinite gear ratios in the forward and reverse mode.



Liquid Air Energy Storage System

This example models a grid-scale energy storage system based on cryogenic liquid air. When there is excess power, the system liquefies ambient air based on a variation of the Claude cycle. The cold liquid air is stored in a low-pressure insulated tank until needed. When there is high power demand, the system expands the stored liquid air to produce power based on the Rankine cycle.

In the liquefaction system, a 4-stage intercooled compressor pressurizes air to 10 MPa. A portion of the high-pressure air is cooled by the chiller and then expanded via the throttle valve. This causes a some of the air to liquefy due to the Joule-Thomson effect. The remaining portion of high-pressure air is sent through an expander to cool the air while recovering some power. The cold low-pressure air that did not liquefy passes through the opposite side of the chiller to refrigerate the high-pressure air before returning to the compressor to complete the cycle.

In the power generation system, liquid air is pumped from the storage tank to the evaporator where it is heated from about 80 K to ambient temperature. This causes the liquid air to vaporize and build up 6.5 MPa of pressure. The high-pressure air is expanded through a 3-stage turbine with reheating to produce power.

To improve round-trip efficiency of the charge and discharge cycles, three thermal stores were added. The low-grade hot store captures waste heat from the compression process and uses it to boost the temperature of air going entering the turbines, increasing power production. The high-grade cold store captures waste cold of liquid air from the evaporation process and the low-grade cold store captures waste cold from the turbine exhaust. Both are used to help refrigerate high-pressure air in the chiller to reduce liquefaction power consumption.

This system consumes about 10 MW of power during charging and produces about 1.8 MW of power during discharging. About 15 MWh of energy is stored over a 10 hour period. The round-trip efficiency is about 15%.

Model

Chiller Subsystem

Cold Thermal Store Subsystem

Compressor Subsystem

Coolant Subsystem

Stage 1 Subsystem

Cooler Subsystem

Evaporator Subsystem

Expander Subsystem

Hot Thermal Store Subsystem

Liquid Air Tank Subsystem

Pump Subsystem

Separator Subsystem

Throttle Valve Subsystem

Turbine Subsystem

Reheater 1 Subsystem

Stage 1 Subsystem

Simulation Results from Scopes

The scope shows the net power consumed the liquefy air during the charge cycle and the net power produced from the stored liquid air during the discharge cycle. About 10 MW is consumed during charge and 1.8 MW is produced during discharge. The charge and discharge phases run for 10 hours each, allowing the system to store about 15 MWh of energy, calculated based on the enthalpy difference between atmospheric air and liquid air. The time-averaged efficiency of the charge cycle is about 26% and the time-averaged efficiency of the discharge cycle is about 56%, resulting in an overall round-trip efficiency of about 15%.

Simulation Results from Simscape Logging

The top subplot shows the work consumed by the charge cycle to liquefy a unit mass of air and the work produced by the discharge cycle from a unit mass of liquid air. The bottom subplot shows the liquefaction yield, defined as the mass of liquid air produced per unit mass of air compressed. Starting from the second charge cycle, the work to liquefy air and the yield is greater at the start of the cycle because of the boost from the cold thermal store. The work and yield settles down over the course of the cycle as the thermal store returns to environment temperature.

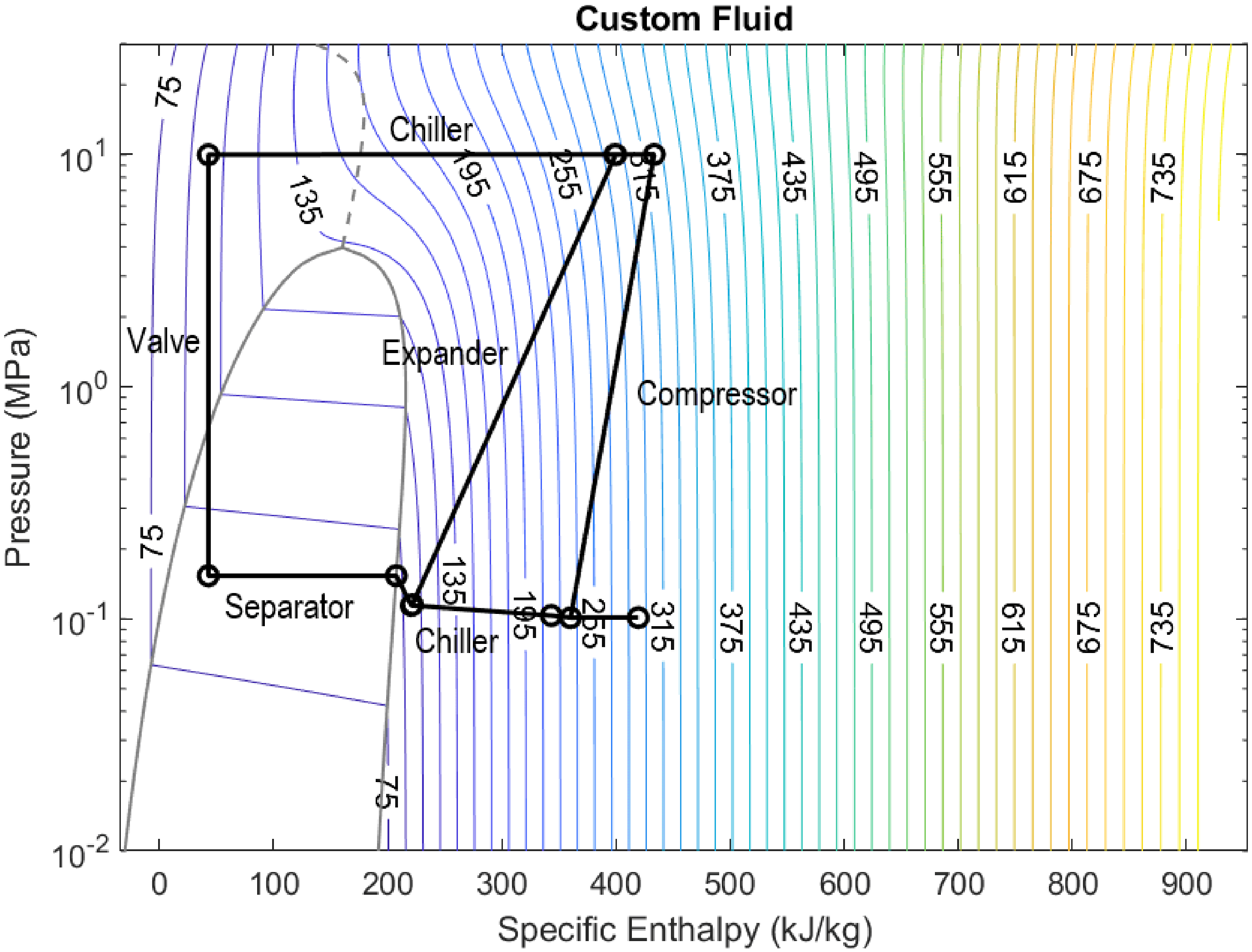
This figure shows the breakdown of the contributions to the net power consumed during the charge cycle and the net power produced during the discharge cycle. During the charge cycle, the expander in the liquefaction system offsets about 24% of the total power required, resulting in a net power consumption of about 10 MW. During the discharge cycle, the net power production is about 1.8 MW, settling down to about 1.7 MW as the hot thermal store is used up.

The top subplot shows the mass flow rates of air through the key components in the system. During the charge cycle, the compressor drives about 27 kg/s of air through the liquefaction system. Over 70% of the compressed air is diverted through the expander to recover some of the work and improve cycle efficiency. About 6.5 kg/s of liquid air is produced. During the discharge cycle, the pump consumes 7.5 kg/s of liquid air from the tank to run the turbines.

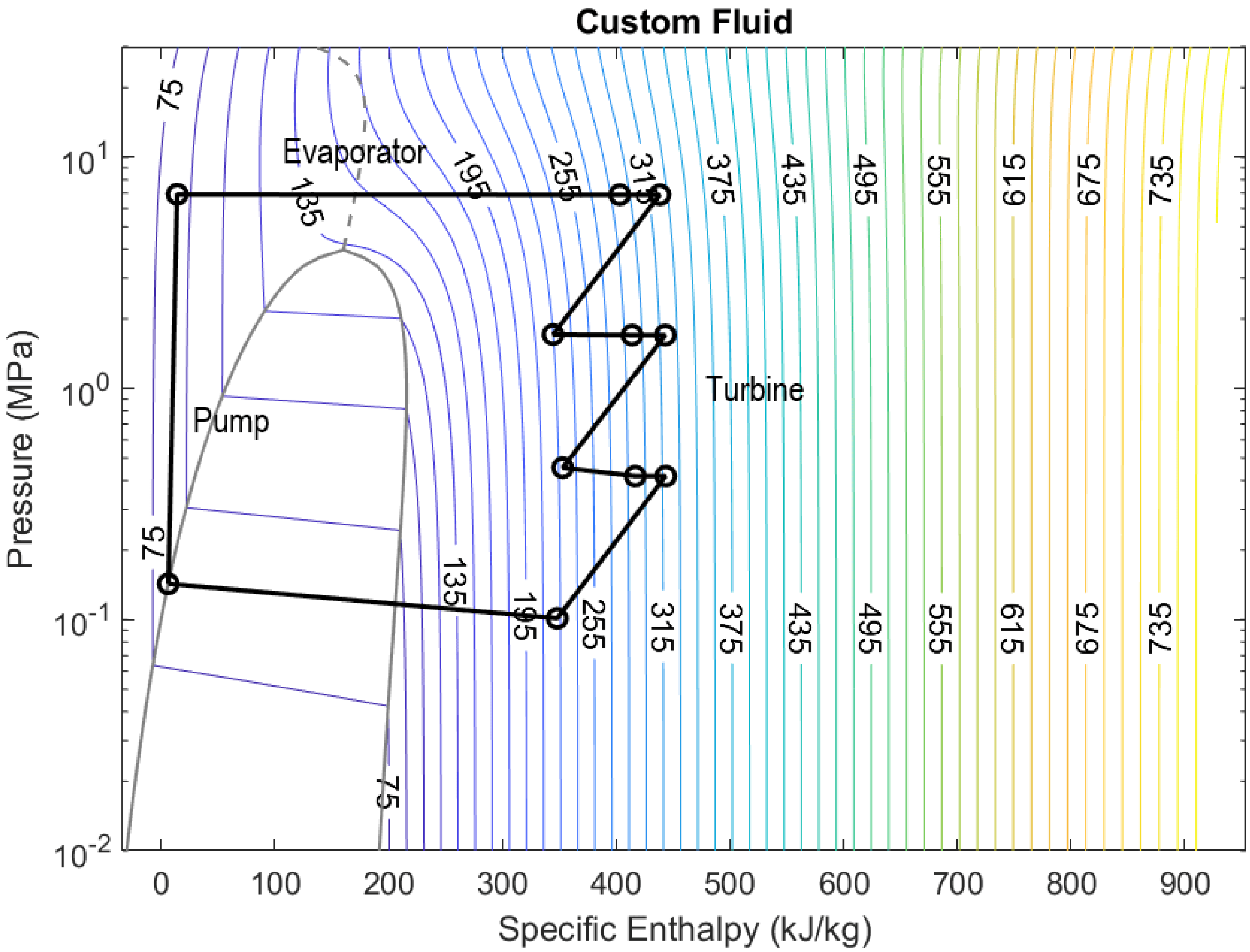
The bottom subplot shows the mass of liquid air in the tank. Starting from the second charge cycle, about 150 metric ton of liquid air is produced and stored in the tank. As seen in the scope, this corresponds to about 15 MWh of energy storage.

This figure shows the performance of the hot and cold thermal stores. The two cold stores capture about 5.1 MWh and 2.3 MWh of energy from the expansion of liquid air and releases about 3.8 MWh and 1.7 MWh of it to the charge cycle. Similarly, the hot store captures about 3.5 MWh of energy from the compression of air and releases about 2.2 MWh of it to the discharge cycle.

This figure shows the pressure-enthalpy diagram of the Claude cycle during the charging process.



This figure shows the pressure-enthalpy diagram of the Rankine cycle during the discharging process.


Custom Fluid
Evaporator
Pump
Turbine
Pressure (MPa)
Specific Enthalpy (kJ/kg)
10^1
10^0
10^-1
10^-2
0
100
200
300
400
500
600
700
800
900
75
135
195
255
315
375
435
495
555
615
675
735

Refrigeration Cycle (Air Conditioning)

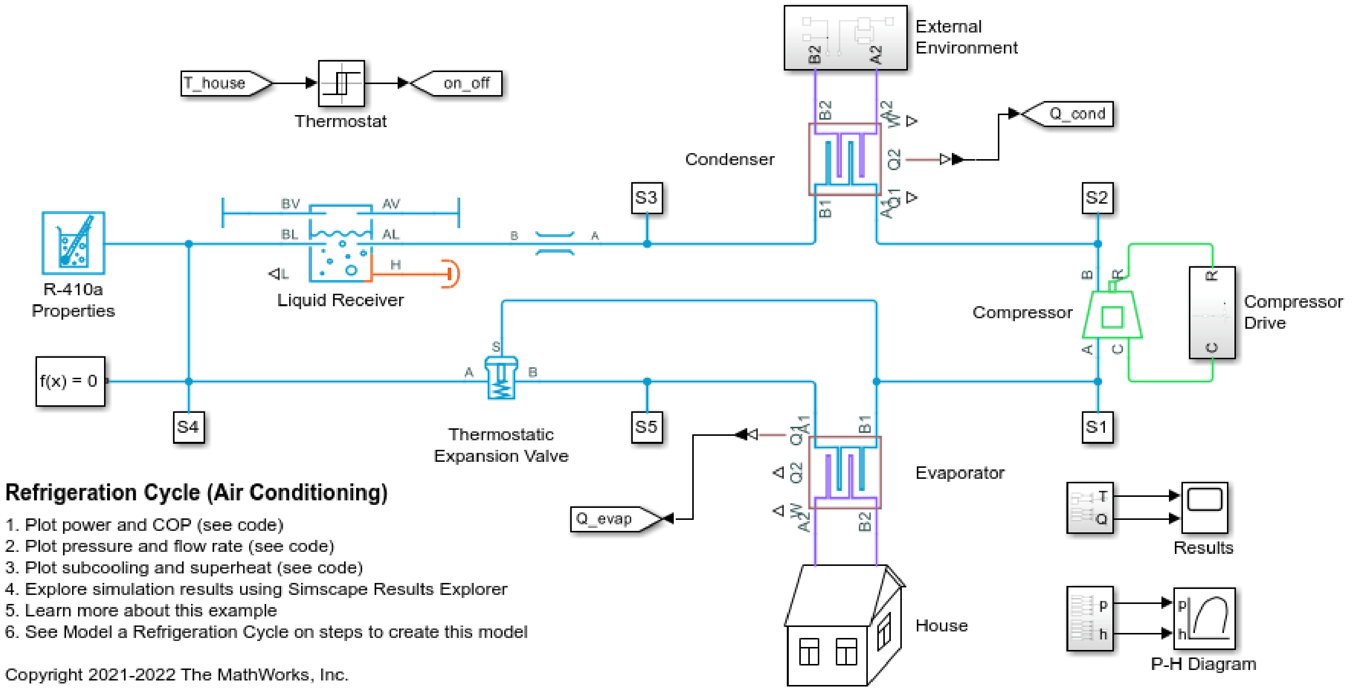
This example shows a refrigeration cycle for a home air conditioning system. See “Model a Refrigeration Cycle” on page 2-55 for the recommended steps to build this model in the two-phase fluid domain.

The refrigerant in this system is R-410a. The evaporator absorbs heat from the air in the house and turns the refrigerant into a superheated vapor. The compressor then pressurizes and drives the refrigerant through the condenser, where the heat absorbed by the refrigerant and the compression work is rejected to the external environment. This causes the refrigerant to condense into a subcooled liquid which is then stored in the liquid receiver. The valve controls the amount of refrigerant that flows from the liquid receiver to the evaporator to maintain the desired amount of superheat. The valve also causes a drop in pressure which cools the refrigerant and allows it to absorb heat in the evaporator.

The cooling load in this model is the house, represented as a volume of air in the moist air domain. A thermal network models heat transfer between the hot external environment and the air in the house via the walls, roof, and windows. Additionally, occupants and appliances generate heat inside the house. A fan circulates air between the house and the evaporator for cooling. The system is controlled by a thermostat which turns the system on and off to maintain a temperature of 22 degC.

See the example “Refrigeration Cycle (System-Level)” on page 6-249 for how to model a refrigeration cycle at a higher system level.

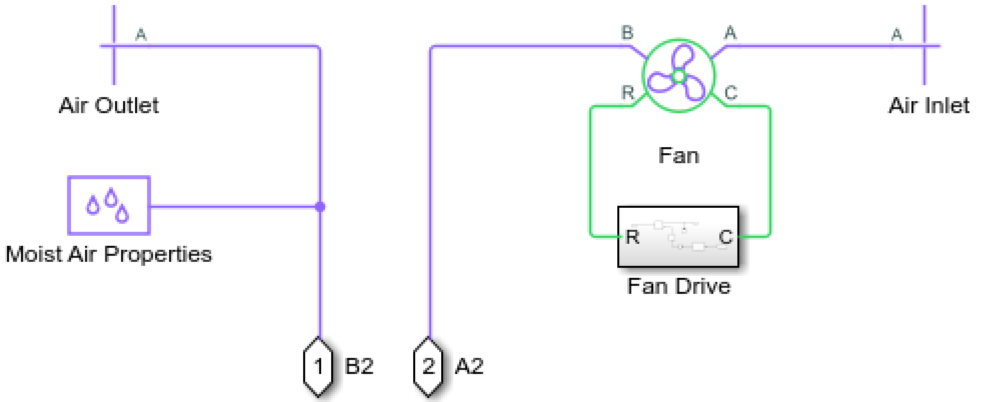
Model



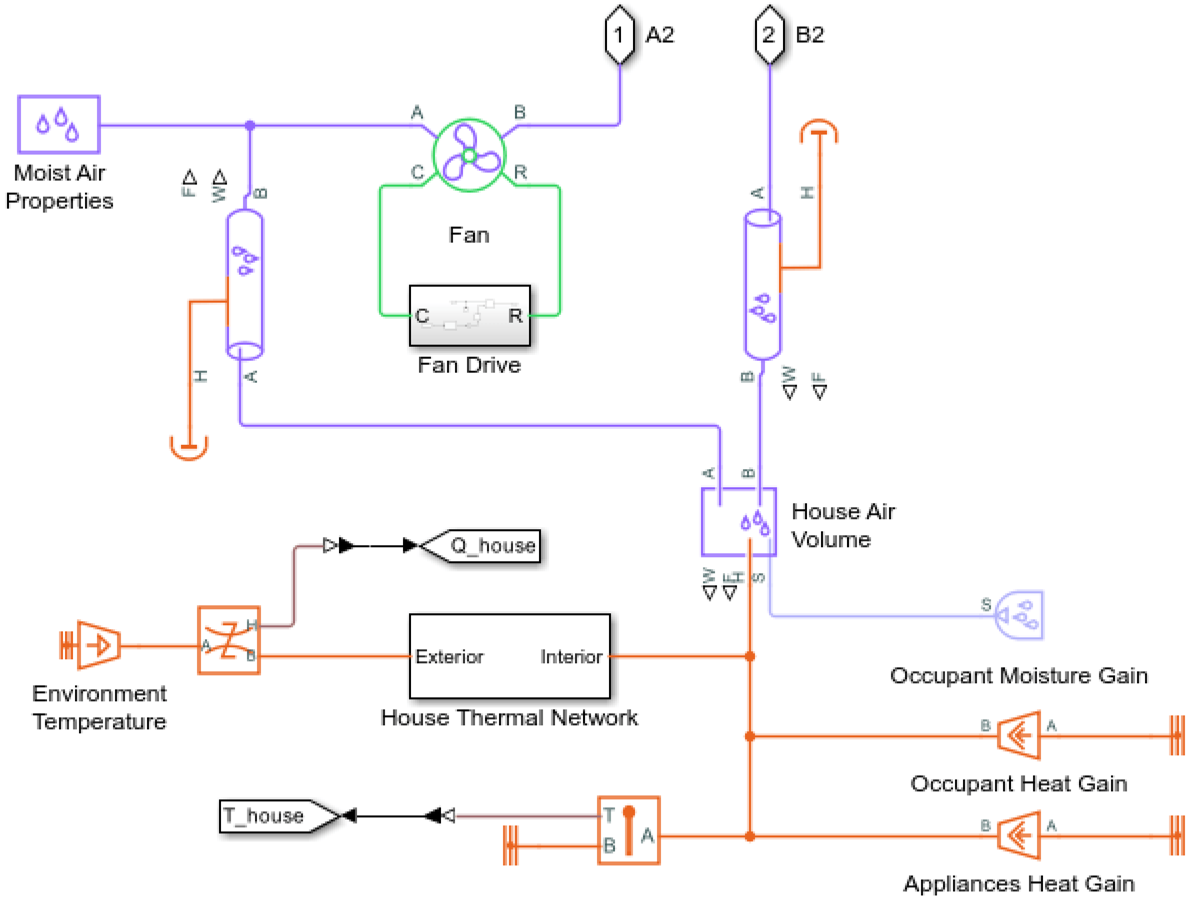
Refrigeration Cycle (Air Conditioning)

1. Plot power and COP (see code)
2. Plot pressure and flow rate (see code)
3. Plot subcooling and superheat (see code)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example
6. See Model a Refrigeration Cycle on steps to create this model

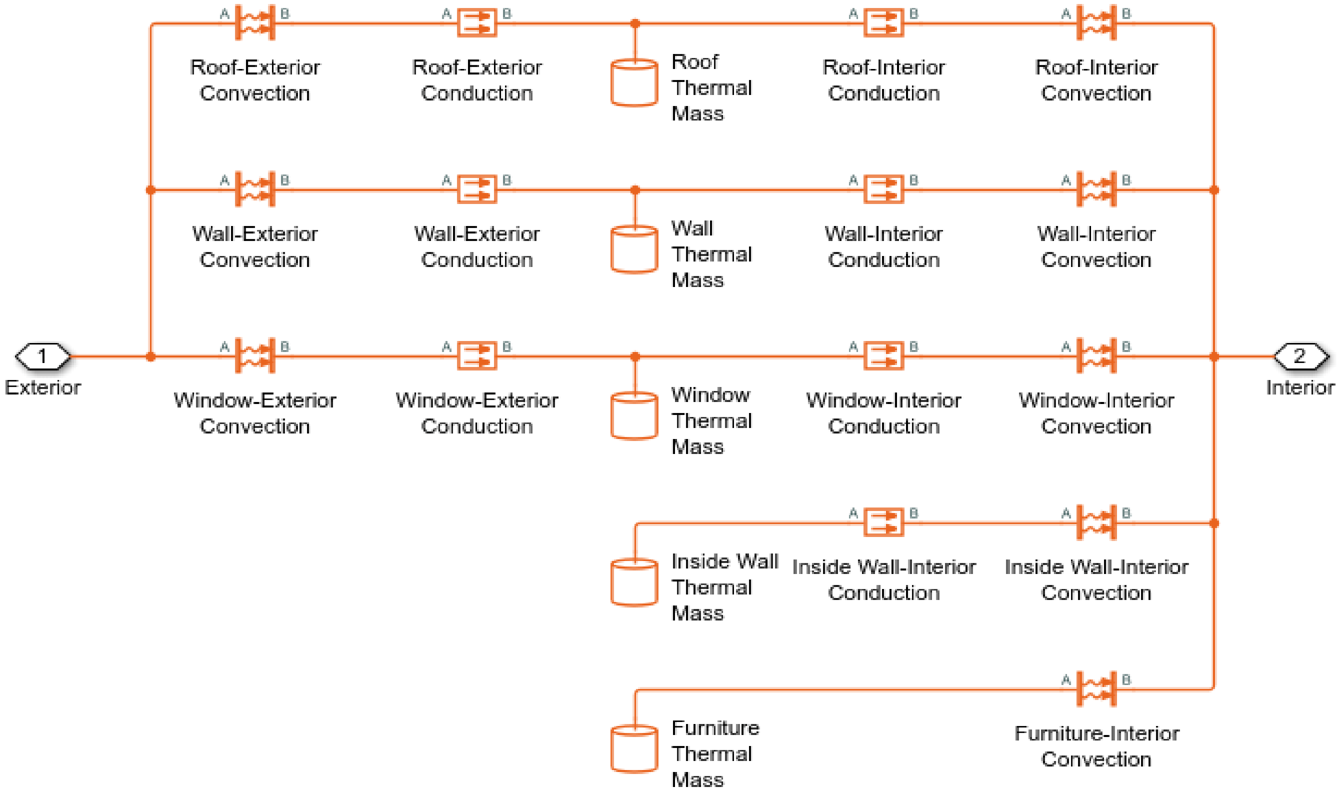
External Environment Subsystem



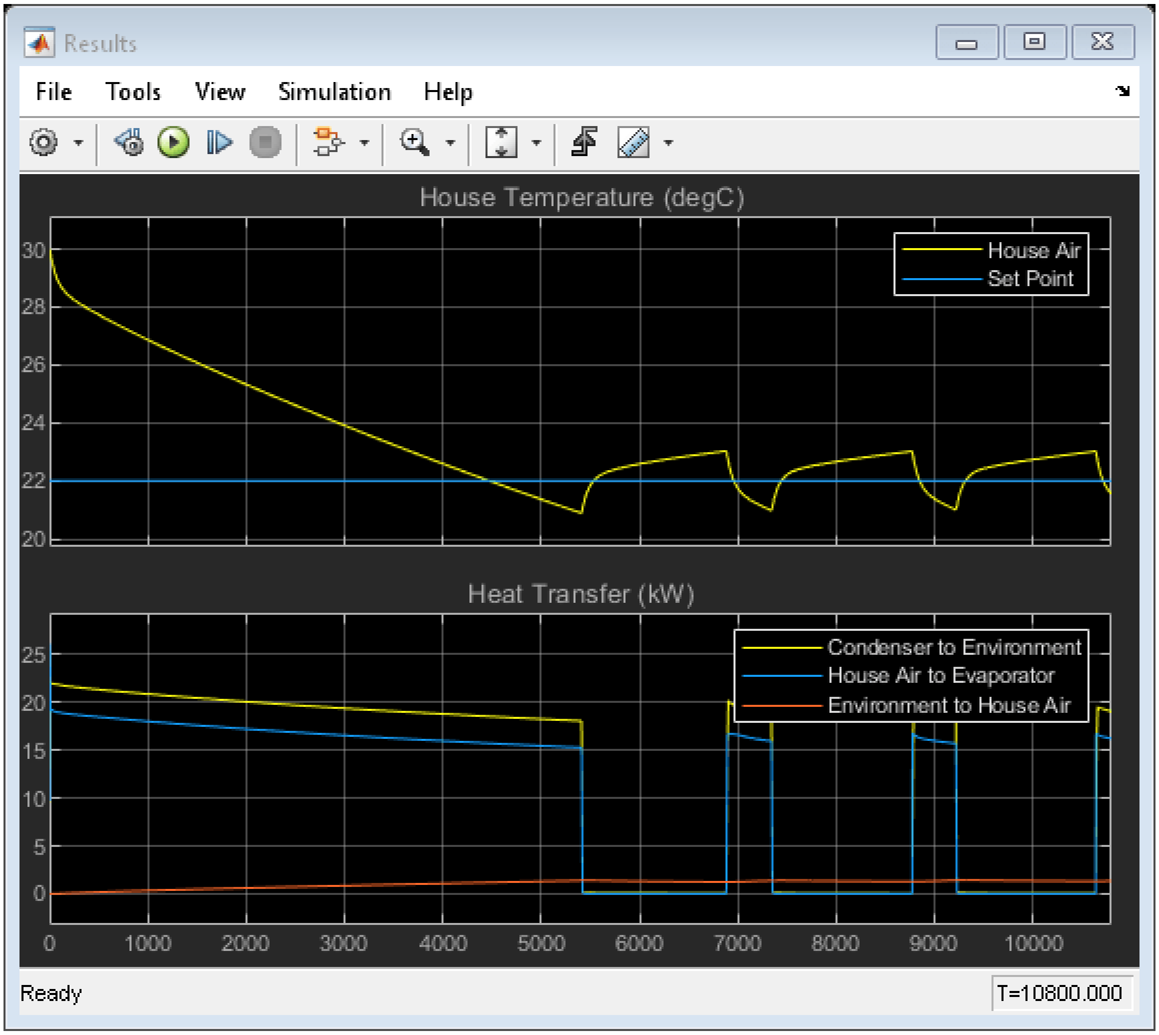
House Subsystem



House Thermal Network Subsystem

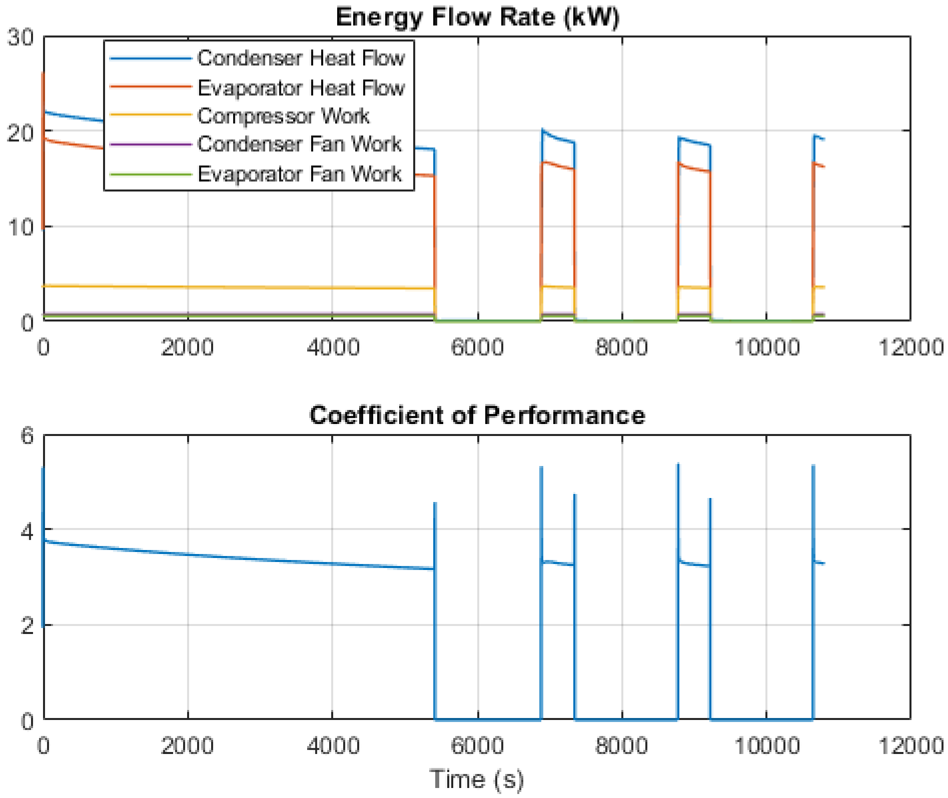


Simulation Results from Scopes

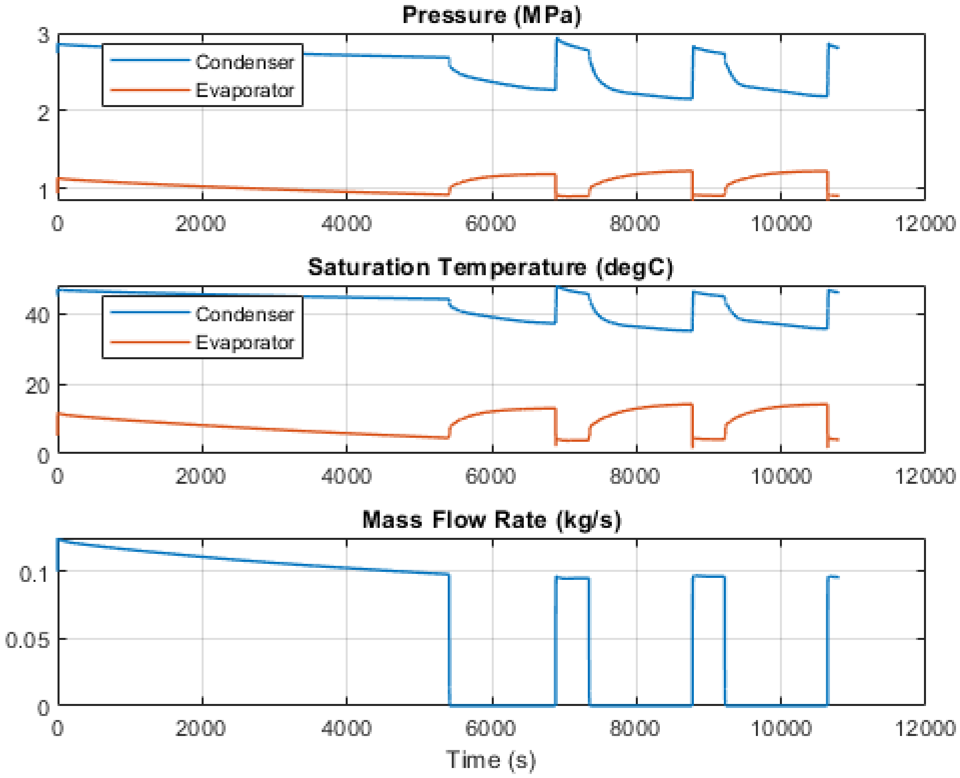


Simulation Results from Simscape Logging

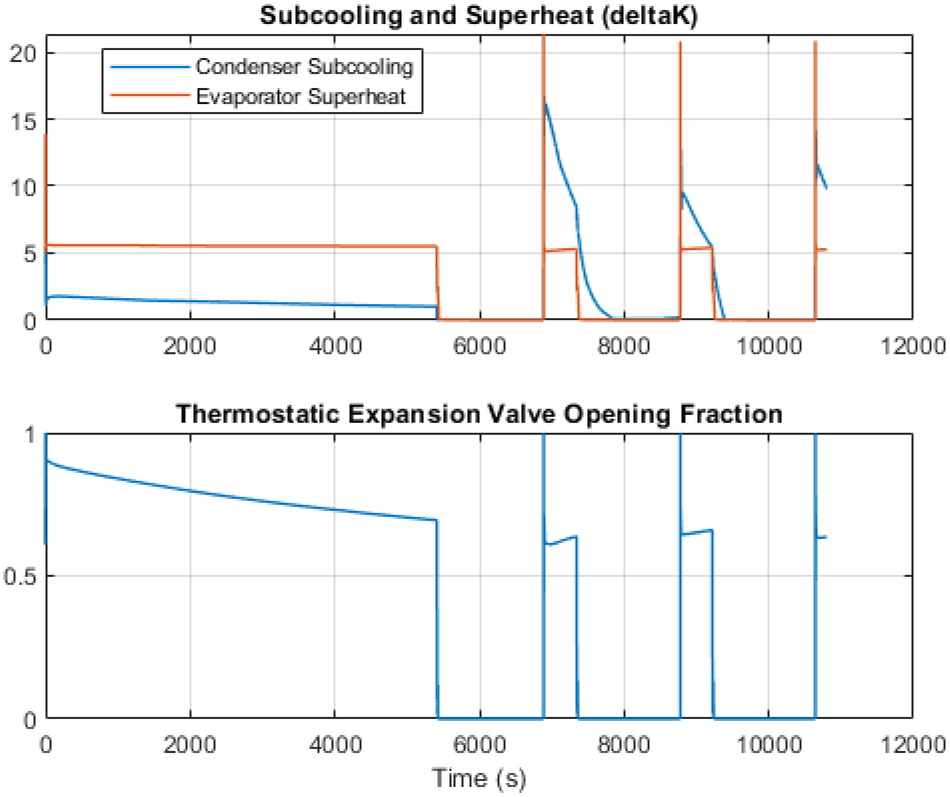
This plot shows the heat transfer in the condenser and the evaporator as well as the power consumed by the compressor and fans. The coefficient of performance is the ratio of the evaporator heat transfer to the total power consumed.



This plot shows the high and low pressure in the refrigeration cycle and the corresponding saturation temperatures. The design condensing temperature is 45 degC and the design evaporating temperature is 5 degC. The design refrigerant mass flow rate is 0.1 kg/s.

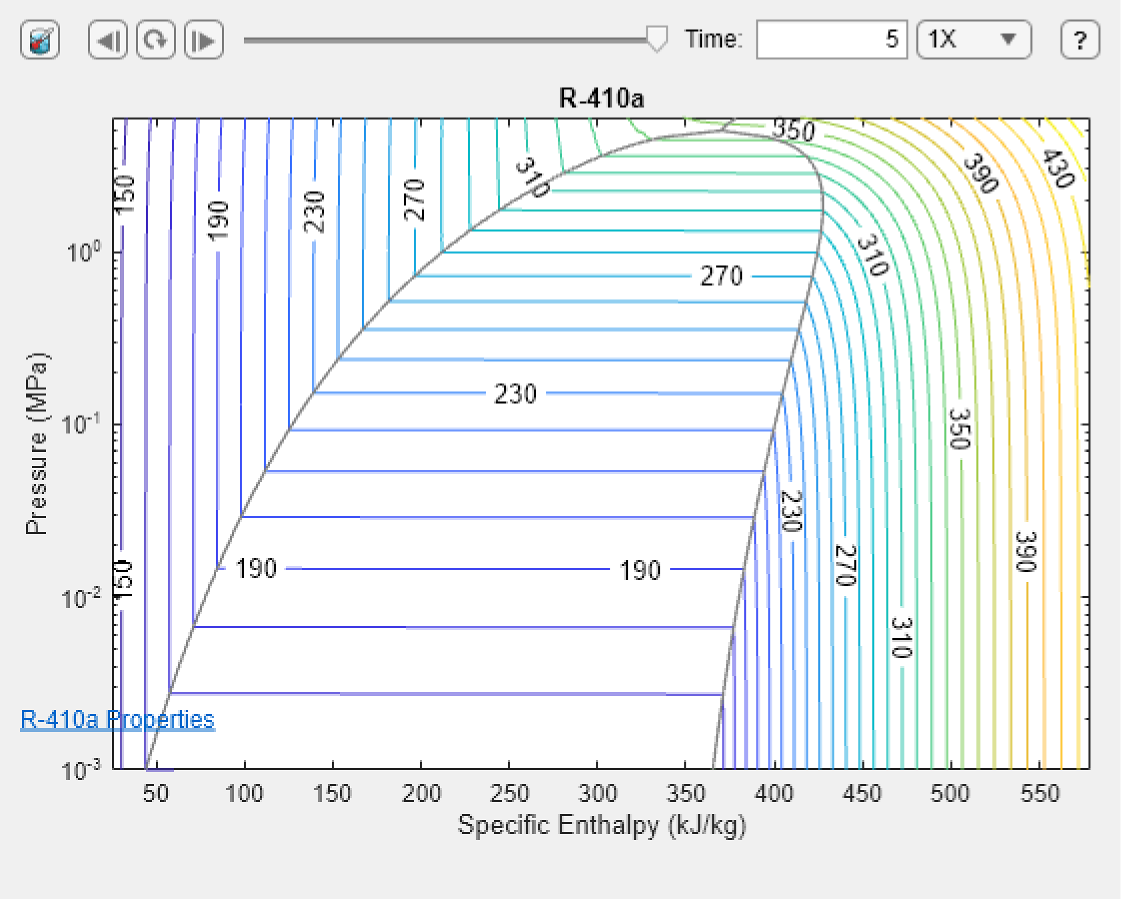


This plot shows the subcooling at the condenser outlet and the superheat at the evaporator outlet. The thermostatic expansion valve meters the flow into the evaporator in order to maintain a superheat of 5 degC.



Pressure-Enthalpy Diagram

This figure shows the fluid states in the refrigeration cycle plotted on a p-h diagram. The cycle points are compressor inlet, condenser inlet, liquid receiver inlet, thermostatic expansion valve inlet, and evaporator inlet. The background contours are lines of constant temperatures for the refrigerant R-410a and the grey curve represents the saturation dome.

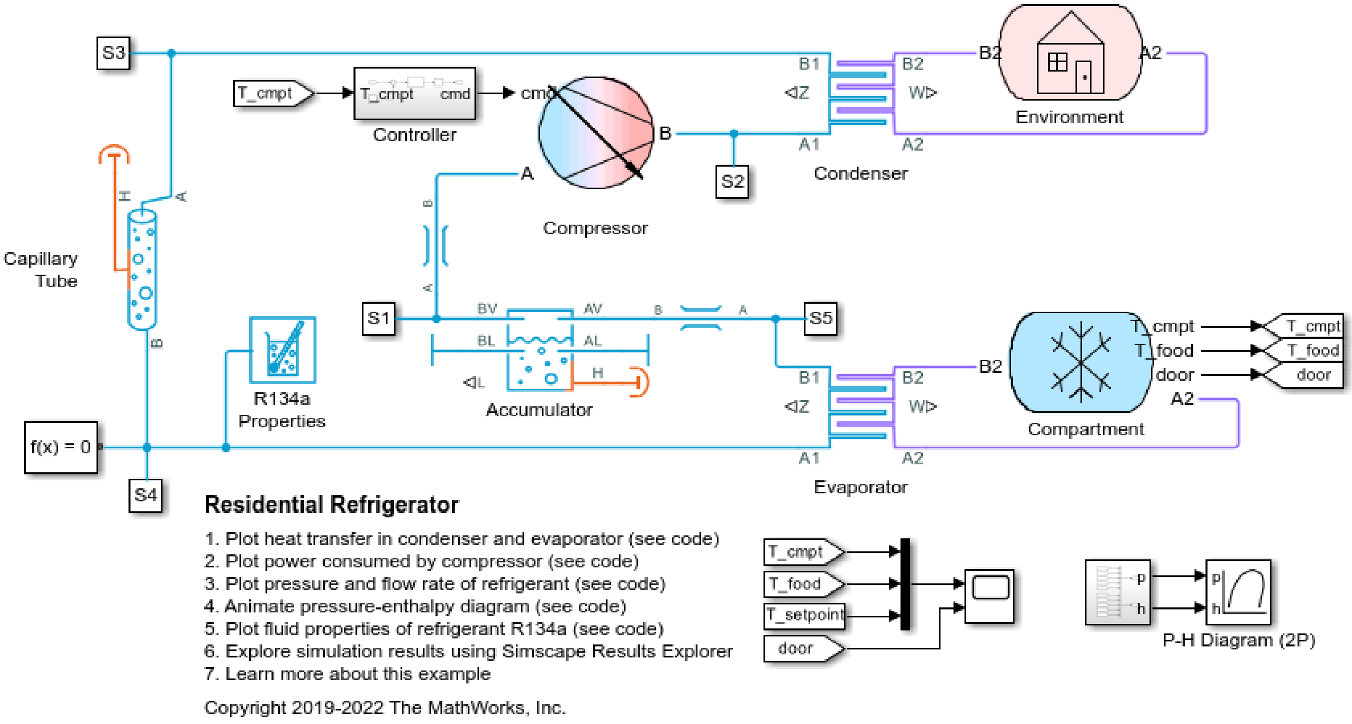

Time: 5
1X
?
R-410a
Pressure (MPa)
Specific Enthalpy (kJ/kg)
10^0
10^{-1}
10^{-2}
10^{-3}
50
100
150
200
250
300
350
400
450
500
550
150
190
230
270
310
350
390
430
270
230
190
190
150
310
350
230
270
390
310
R-410a Properties

Residential Refrigerator

This example models a basic refrigeration system that transfers heat between the refrigerant two-phase fluid and the environment moist air mixture. The compressor drives the R134a refrigerant through a condenser, a capillary tube, and an evaporator. An accumulator ensures that only vapor returns to the compressor.

Two fans drive moist air flow over the condenser and the evaporator. The evaporator air flow is divided between the freezer compartment and the regular compartment. The controller turns the compressor on and off to keep the compartment air temperature at around 4 degC.

Model

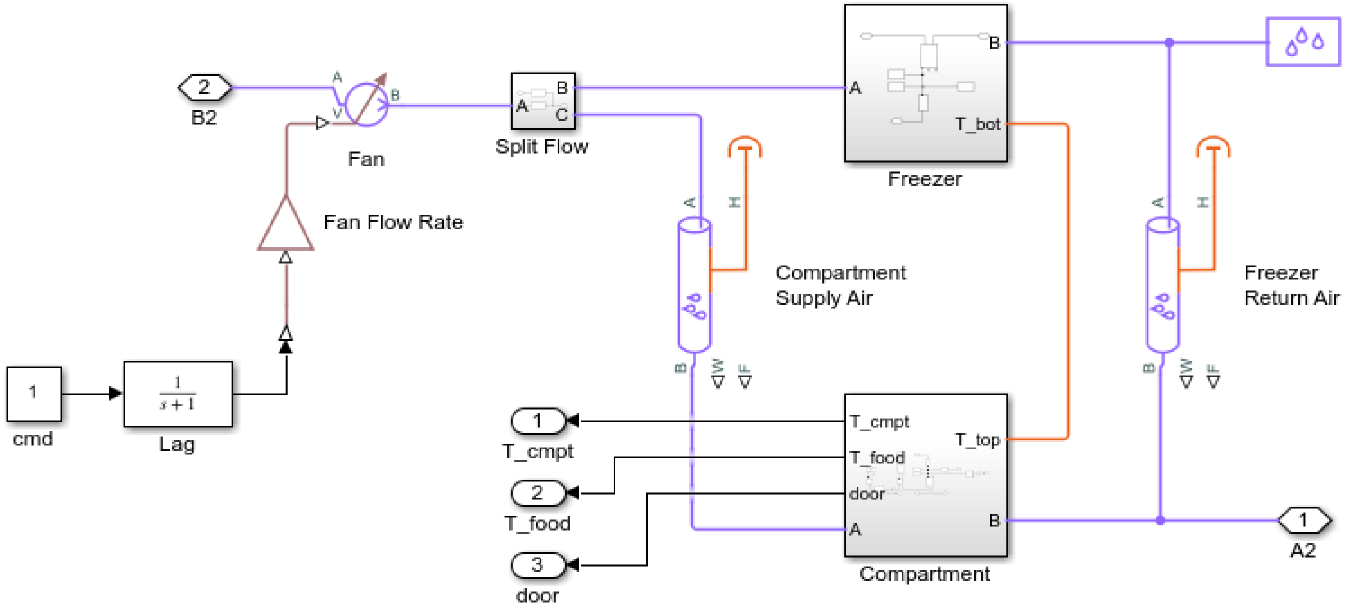


Residential Refrigerator

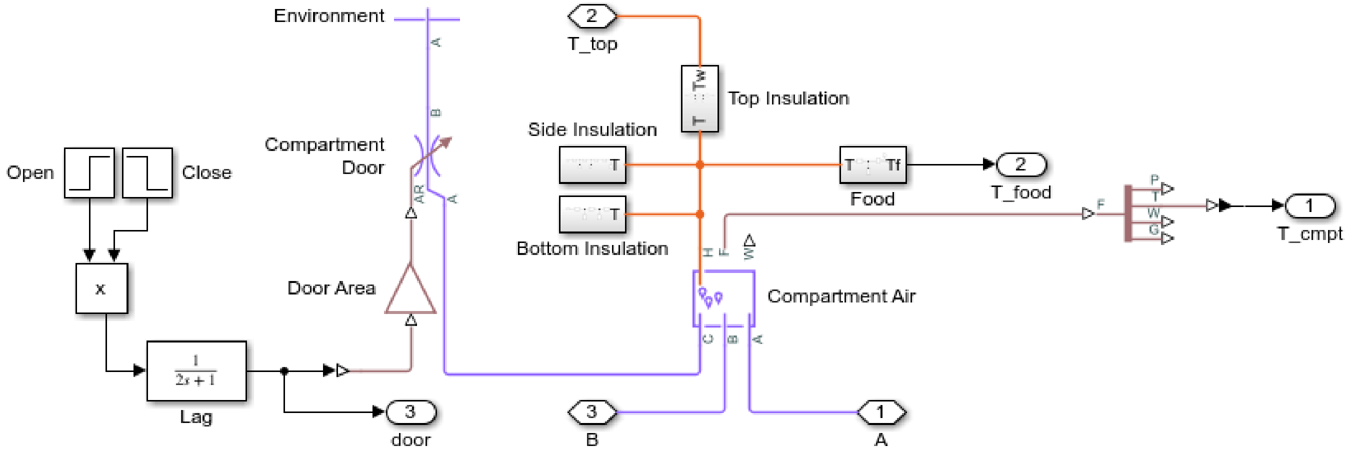
1. Plot heat transfer in condenser and evaporator (see code)
2. Plot power consumed by compressor (see code)
3. Plot pressure and flow rate of refrigerant (see code)
4. Animate pressure-enthalpy diagram (see code)
5. Plot fluid properties of refrigerant R134a (see code)
6. Explore simulation results using Simscape Results Explorer
7. Learn more about this example

Copyright 2019-2022 The MathWorks, Inc.

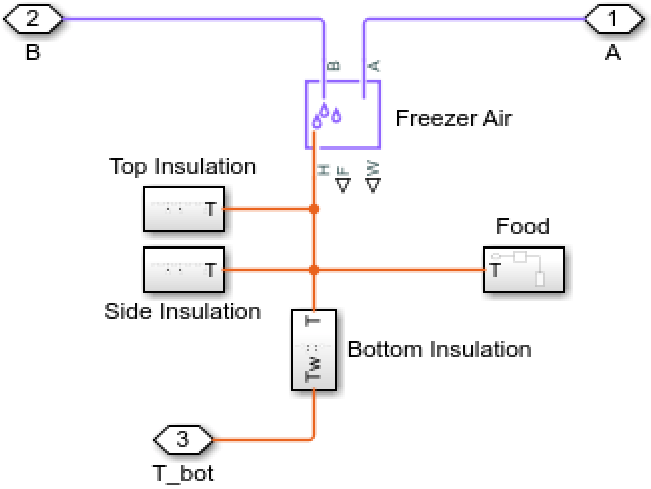
Compartment Subsystem



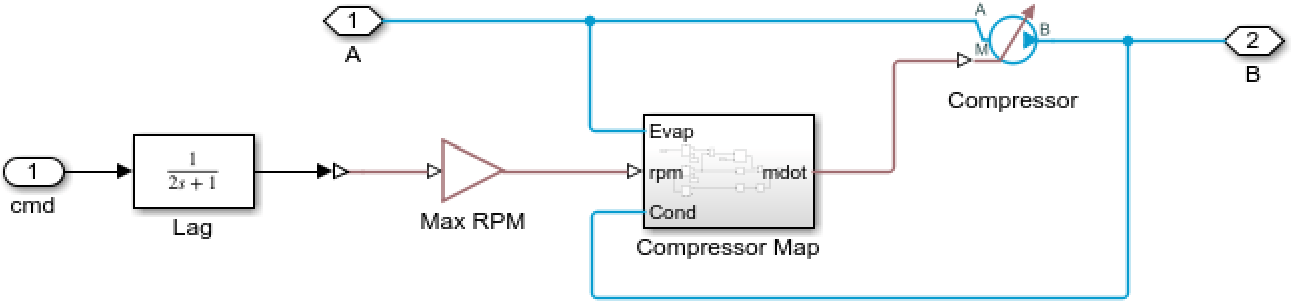
Compartment Subsystem



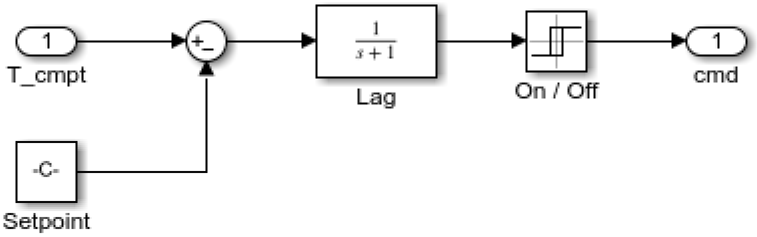
Freezer Subsystem



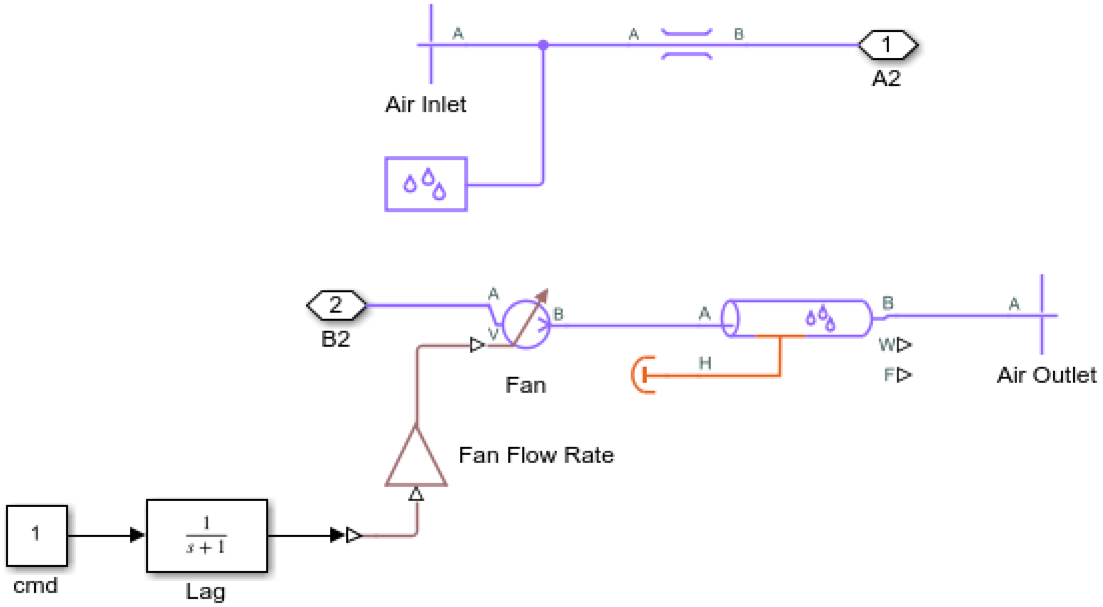
Compressor Subsystem



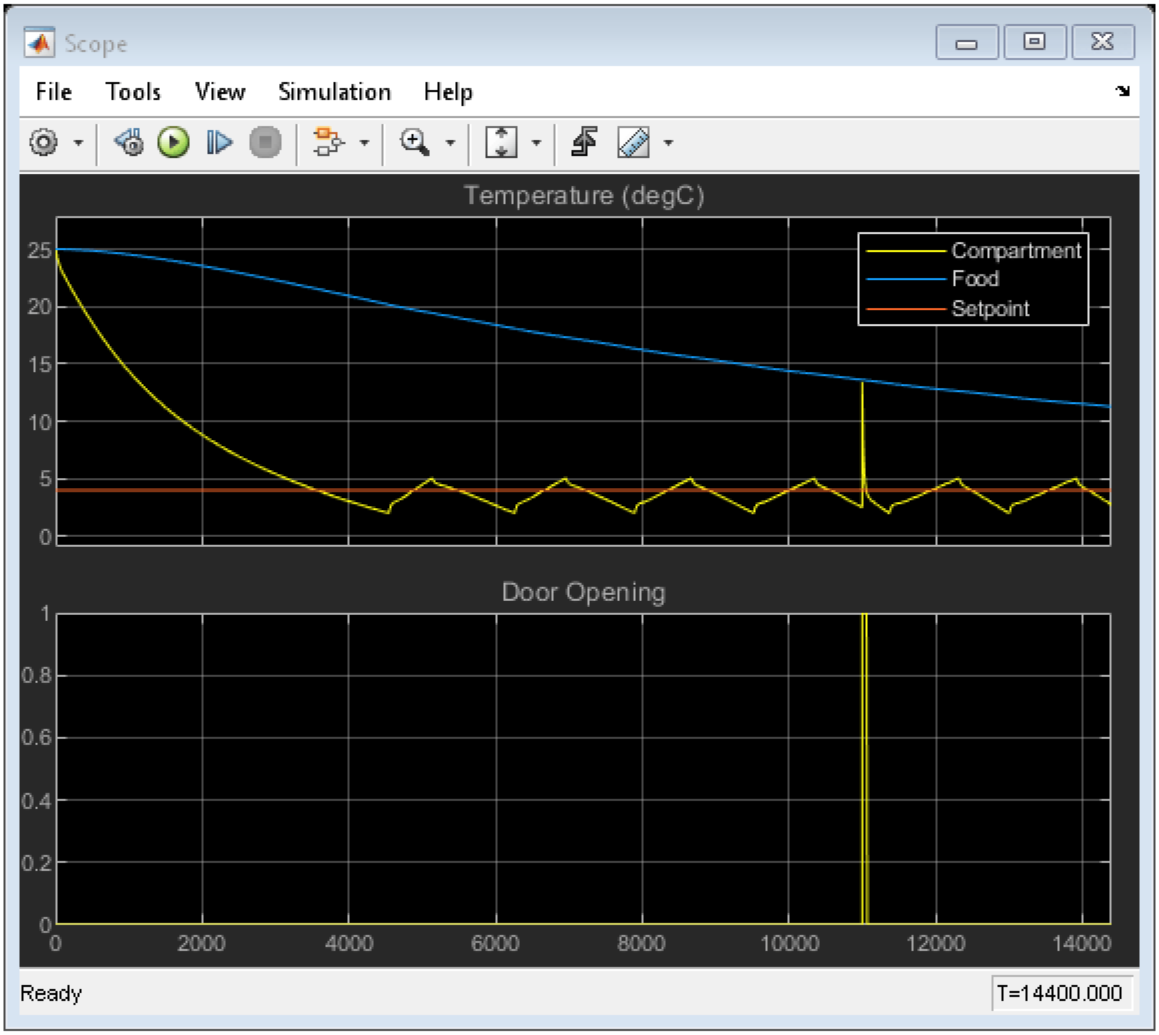
Controller Subsystem



Environment Subsystem

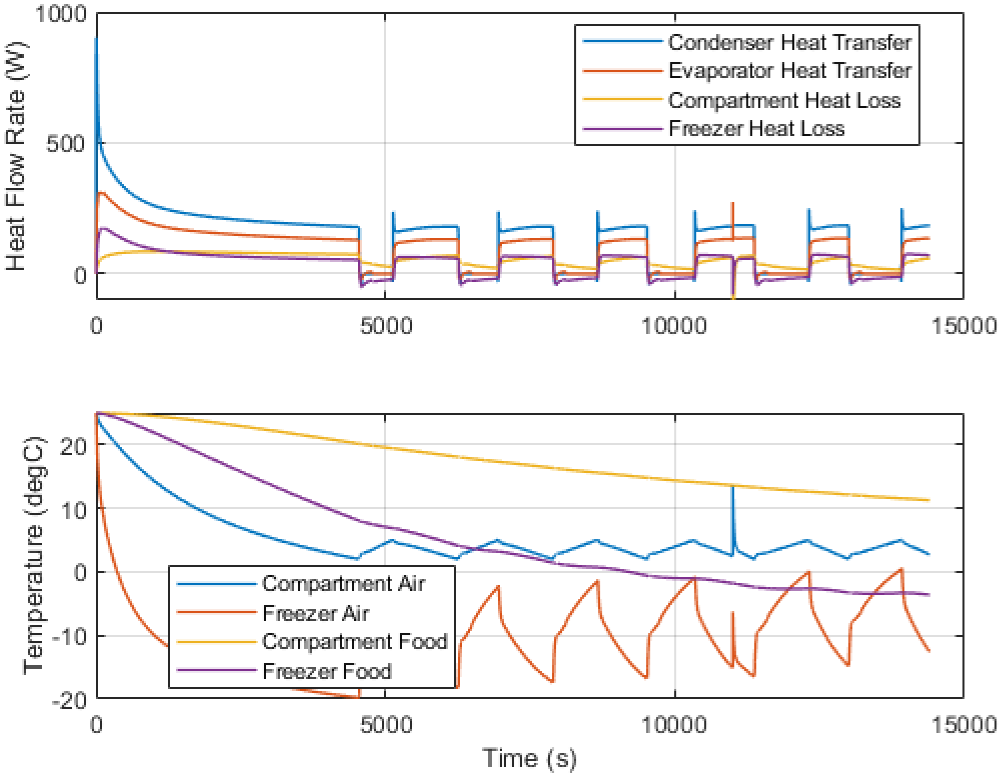


Simulation Results from Scopes

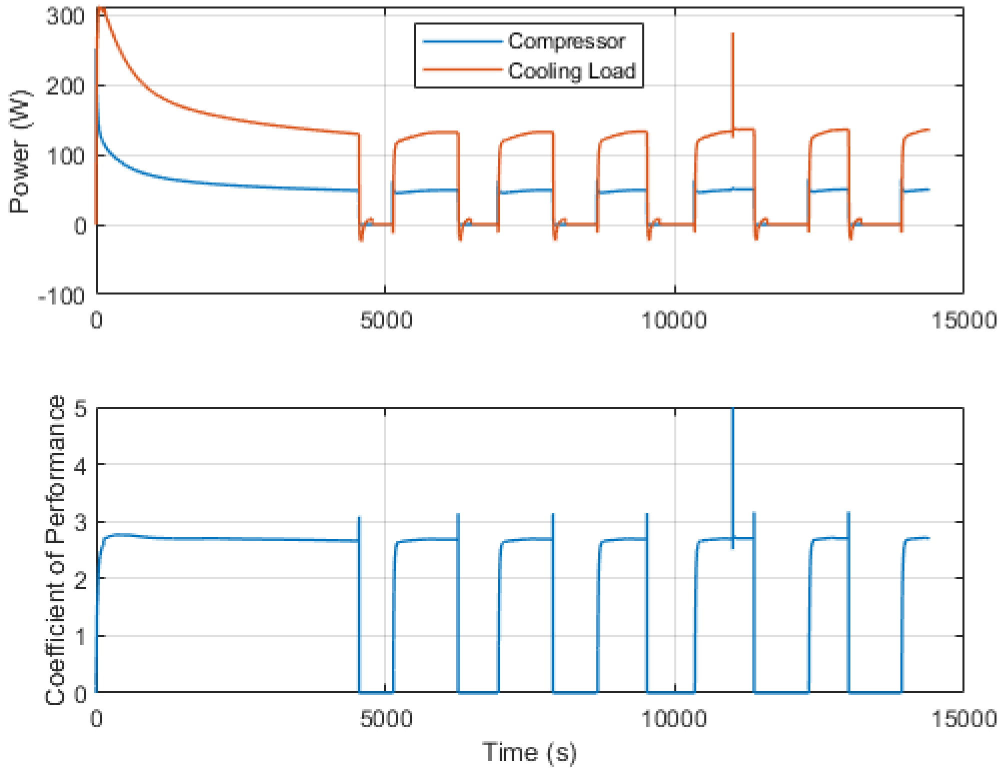


Simulation Results from Simscape Logging

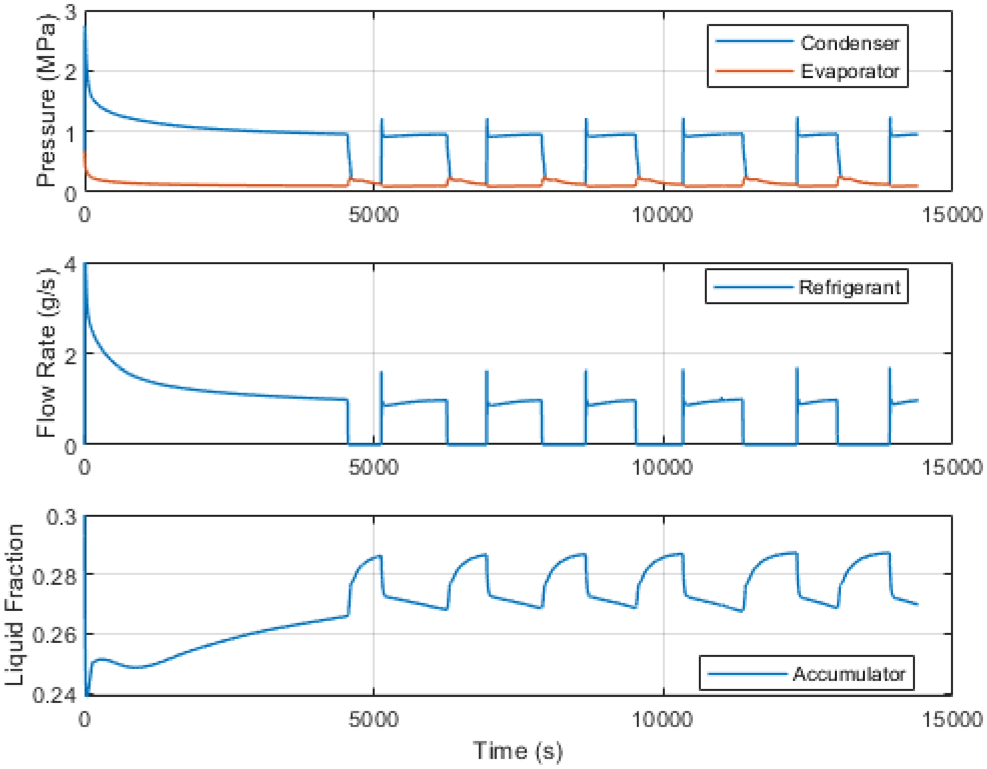
This plot shows the rate of heat transfer between refrigerant and moist air in the condenser and evaporator as well as the rate of heat loss through the insulation of the compartment and freezer. It also shows the temperature of cold air and food in the compartment and freezer. At 11000 s, the compartment door is opened for 60 s, resulting in a spike in compartment temperature.



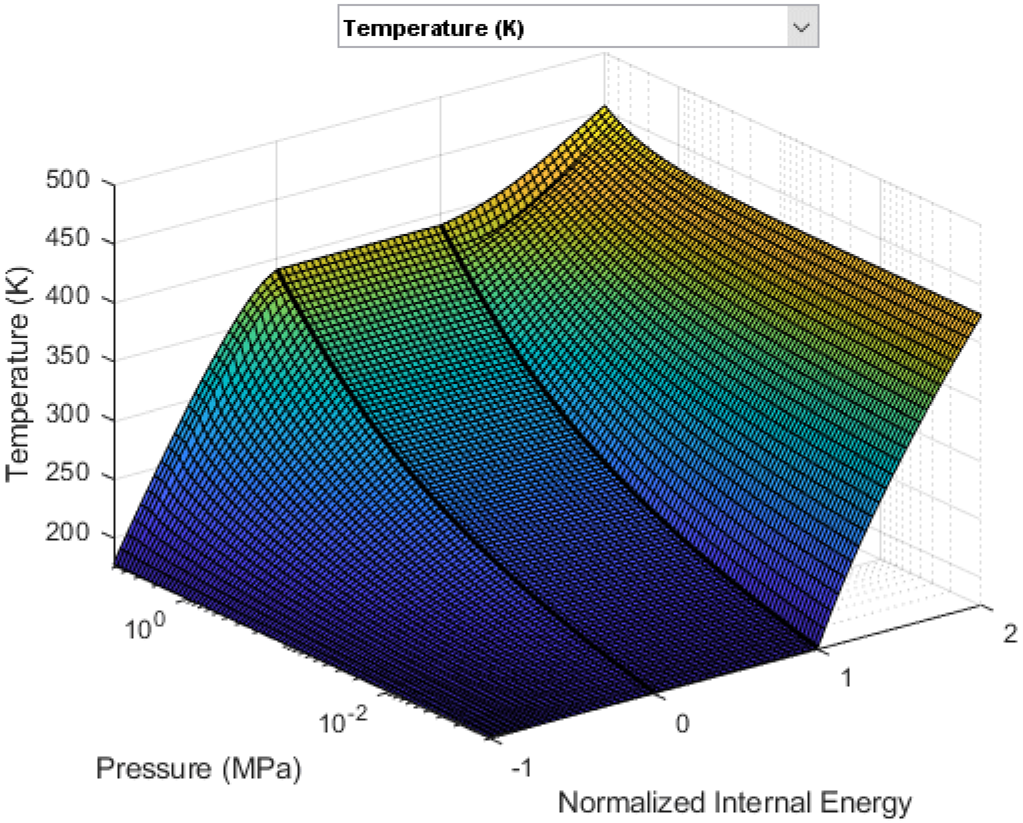
This plot shows the power consumed by the compressor and the cooling load of the refrigeration system, which is the rate of heat transfer in the evaporator. The coefficient of performance is the ratio of the cooling load and the power consumed.

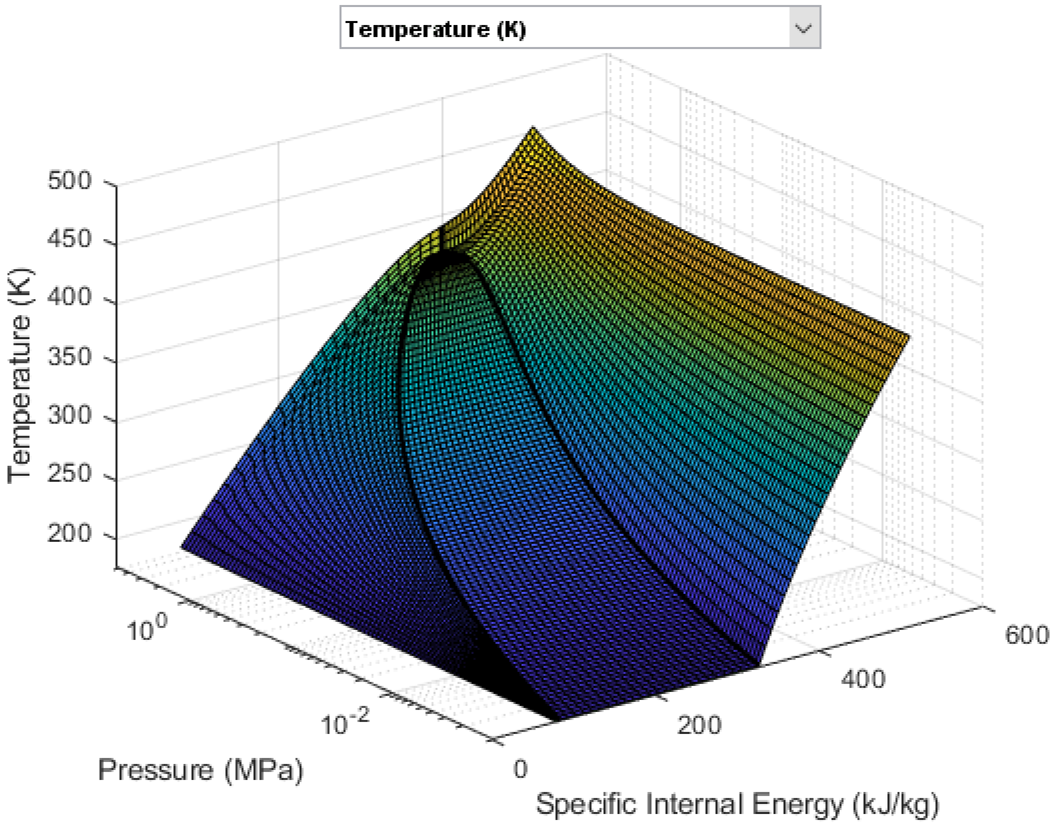


This plot shows refrigerant pressure and mass flow rate. The high pressure line is at around 1 MPa and the lower pressure line is at around 0.1 MPa. The nominal refrigerant flow rate is 1 g/s. The plot also shows the liquid volume fraction in the accumulator.


Pressure (MPa)
3
2
1
0
Condenser
Evaporator
0
5000
10000
15000
Flow Rate (g/s)
4
2
0
Refrigerant
0
5000
10000
15000
Liquid Fraction
0.3
0.28
0.26
0.24
Accumulator
0
5000
10000
15000
Time (s)

Fluid Properties

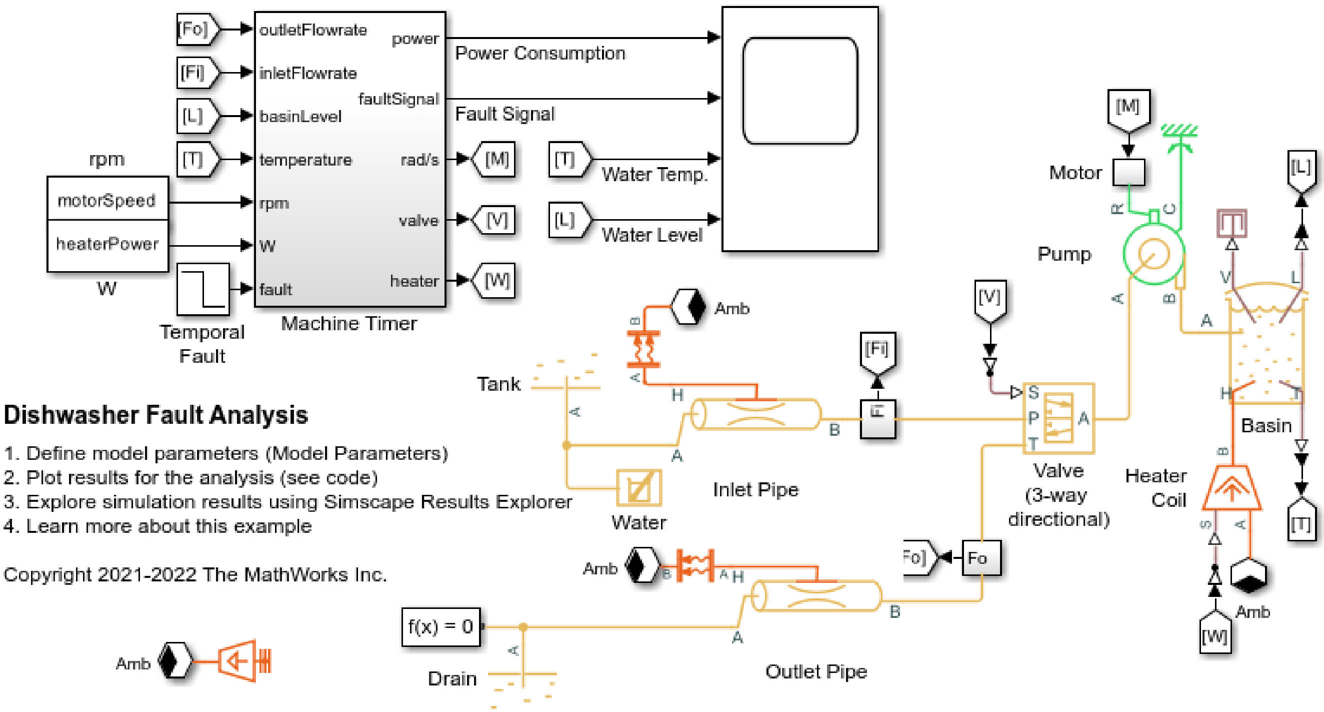



Temperature (K)
Temperature (K)
500
450
400
350
300
250
200
10^0
10^{-2}
Pressure (MPa)
0
200
400
600
Specific Internal Energy (kJ/kg)

Dishwasher Fault Analysis

This example shows how to model a dishwasher and introduce a fault in its operation. Observe the system response when a fault occurs in the water heater during operation. The electric motor switches off, and the dishwasher aborts operation.

Model

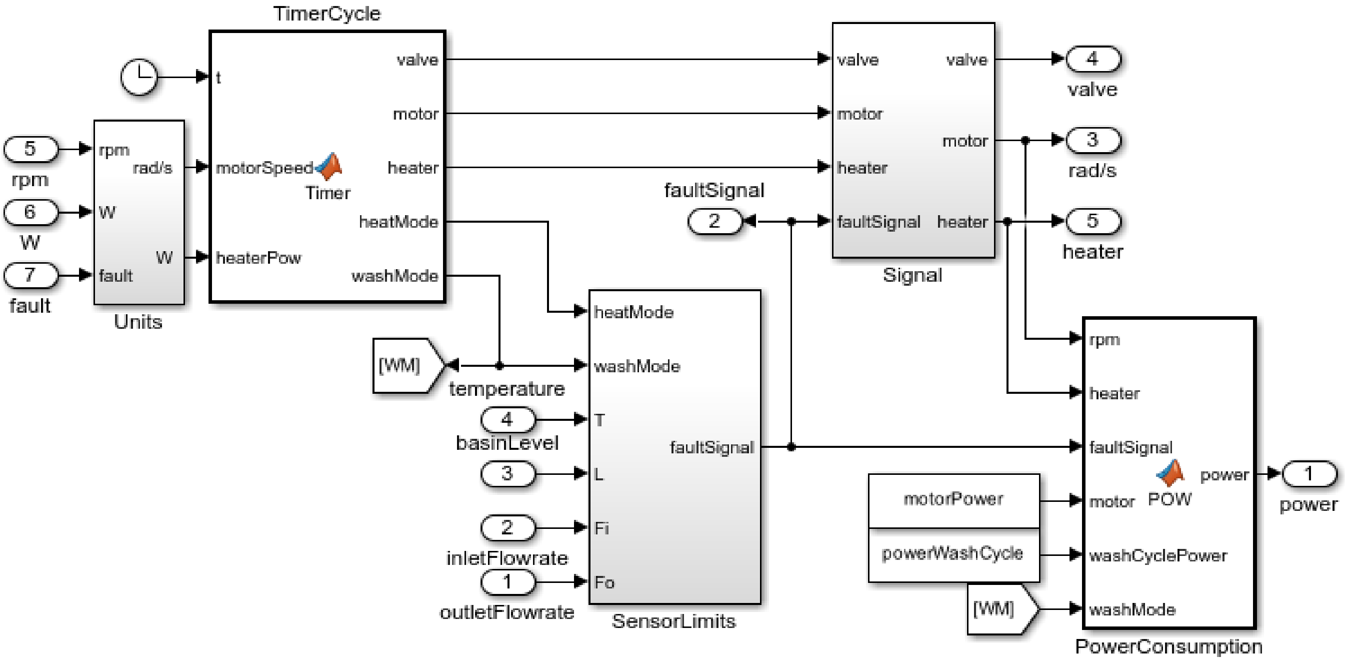


Dishwasher Fault Analysis

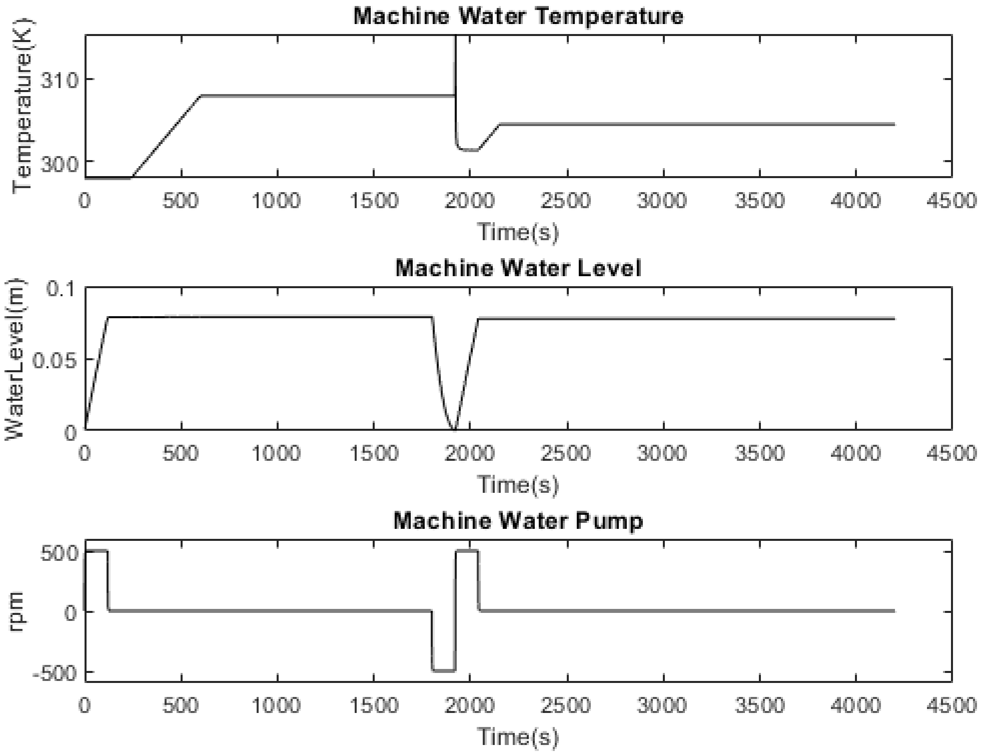
1. Define model parameters (Model Parameters)
2. Plot results for the analysis (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2021-2022 The MathWorks Inc.

Machine Timer Subsystem



Simulation Results from Simscape™ Logging

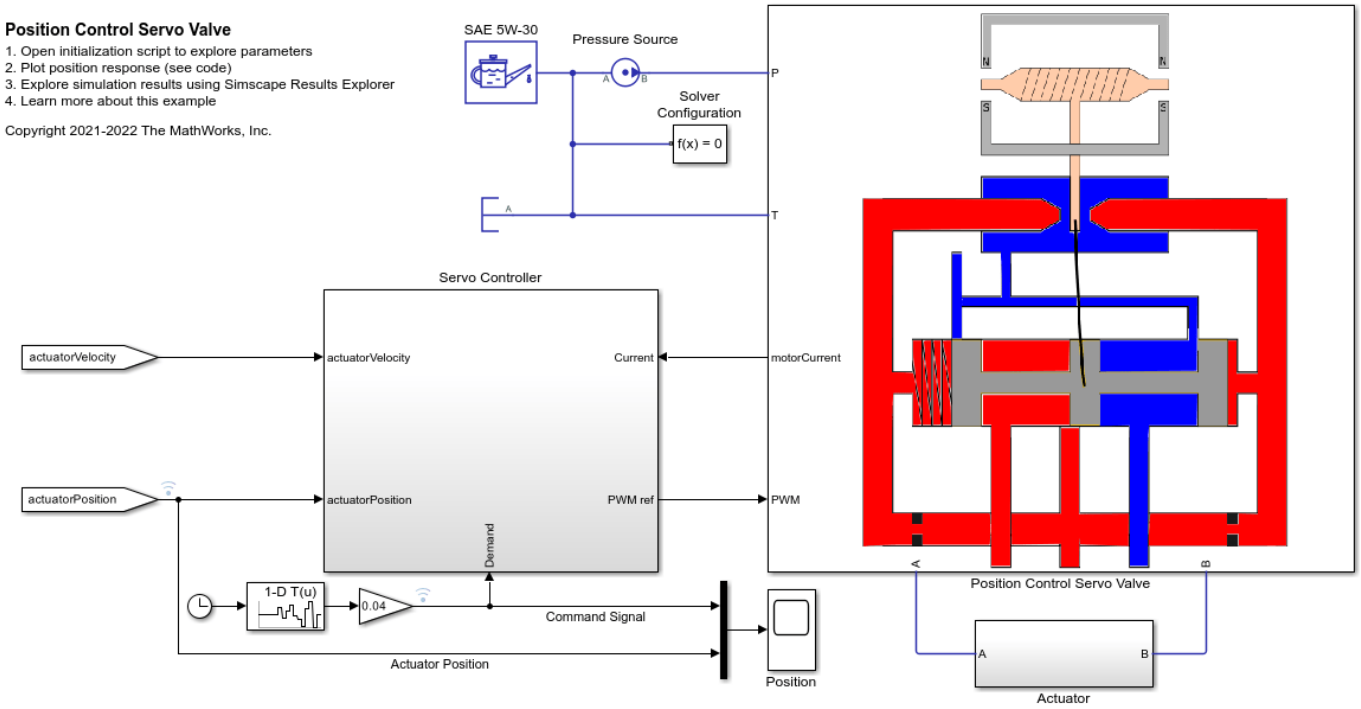


Position Control Servo Valve

This example shows how to model, parameterize, and test a position control servo valve with a closed loop control. When you run the model, it generates a comparison plot between the commanded and the achieved position in the actuator with respect to the time. A position control servo valve provides a precise and fast control of the position in the actuator with a very small electrical signal to the torque motor. Aerospace, construction and agricultural equipments manufacturers use these valves for safety critical applications.

Model

The following figure shows the model of a position control servo valve. Here, connection P is a physical signal port associated with the pressure source. Connection T is physical signal port associated withe the reservoir. Connections A and B are physical signal ports associated with the actuator control.



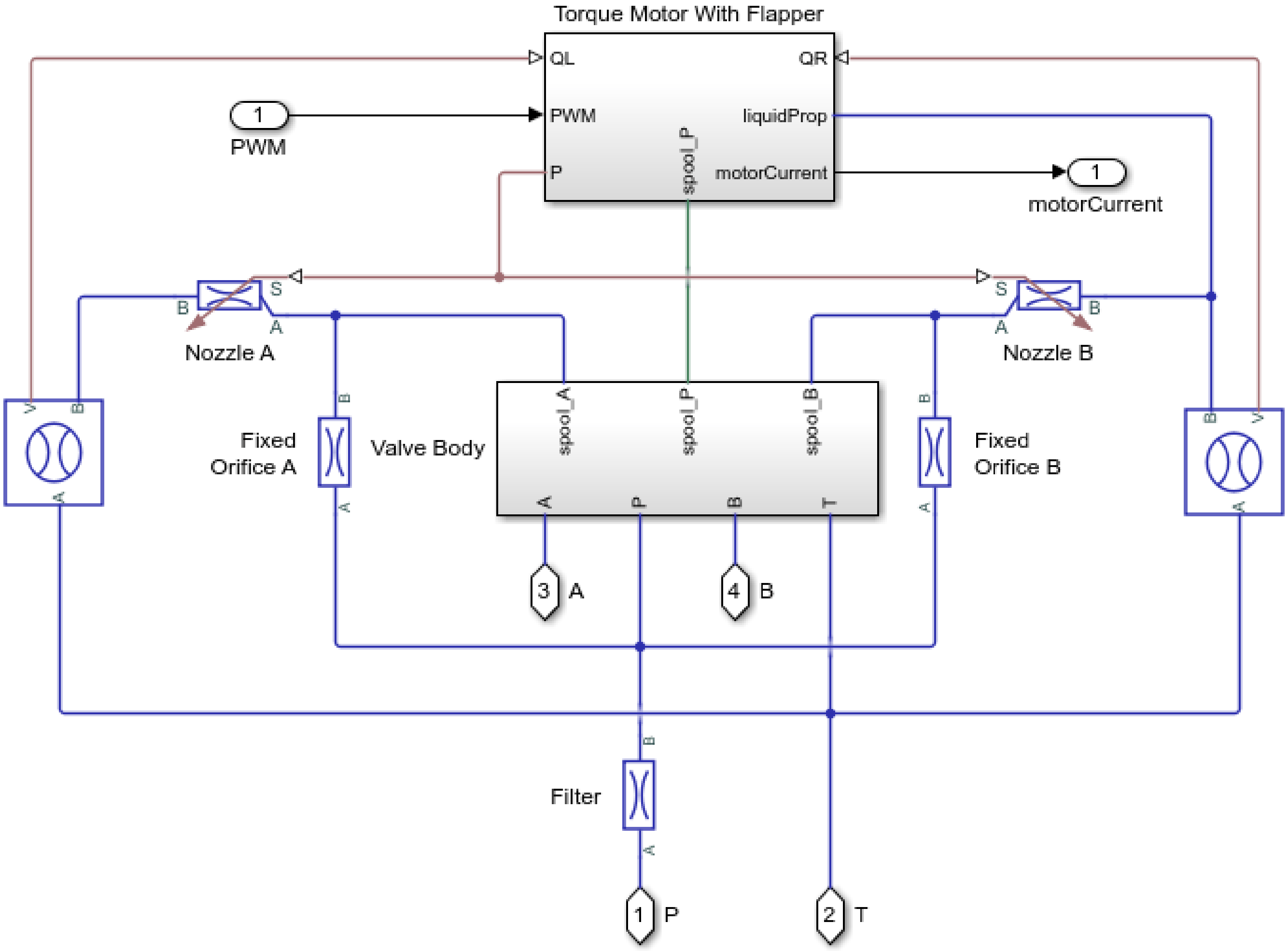
Position Control Servo Valve

1. Open initialization script to explore parameters
2. Plot position response (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2021-2022 The MathWorks, Inc.

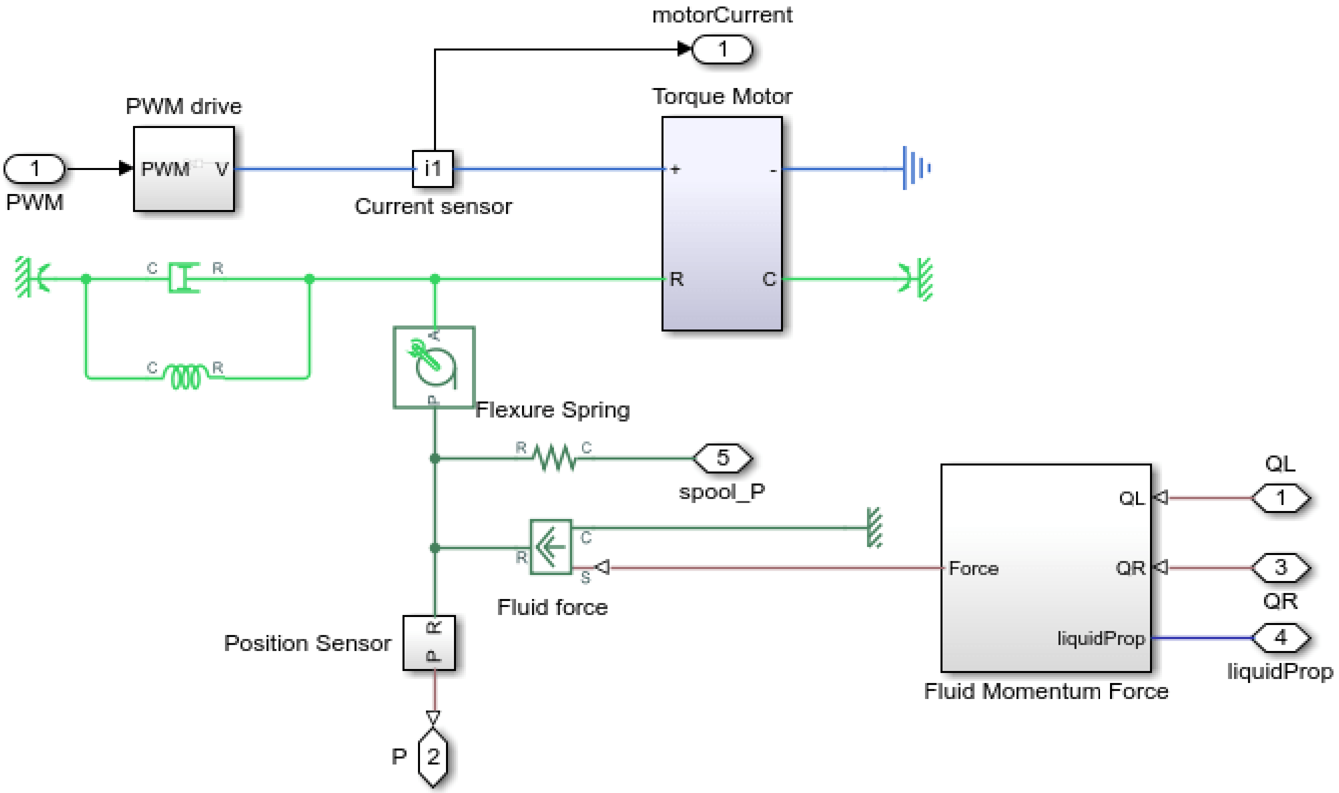
Position Control Servo Valve Subsystem

This subsystem demonstrates how to model the position control servo valve. It includes the flapper nozzle amplifier circuit, the torque motor, and the valve body. The flapper nozzle amplifier circuit includes nozzles and fixed orifices on either side of the flapper as well as a filter. This circuit controls the pressure on both sides of the spool.



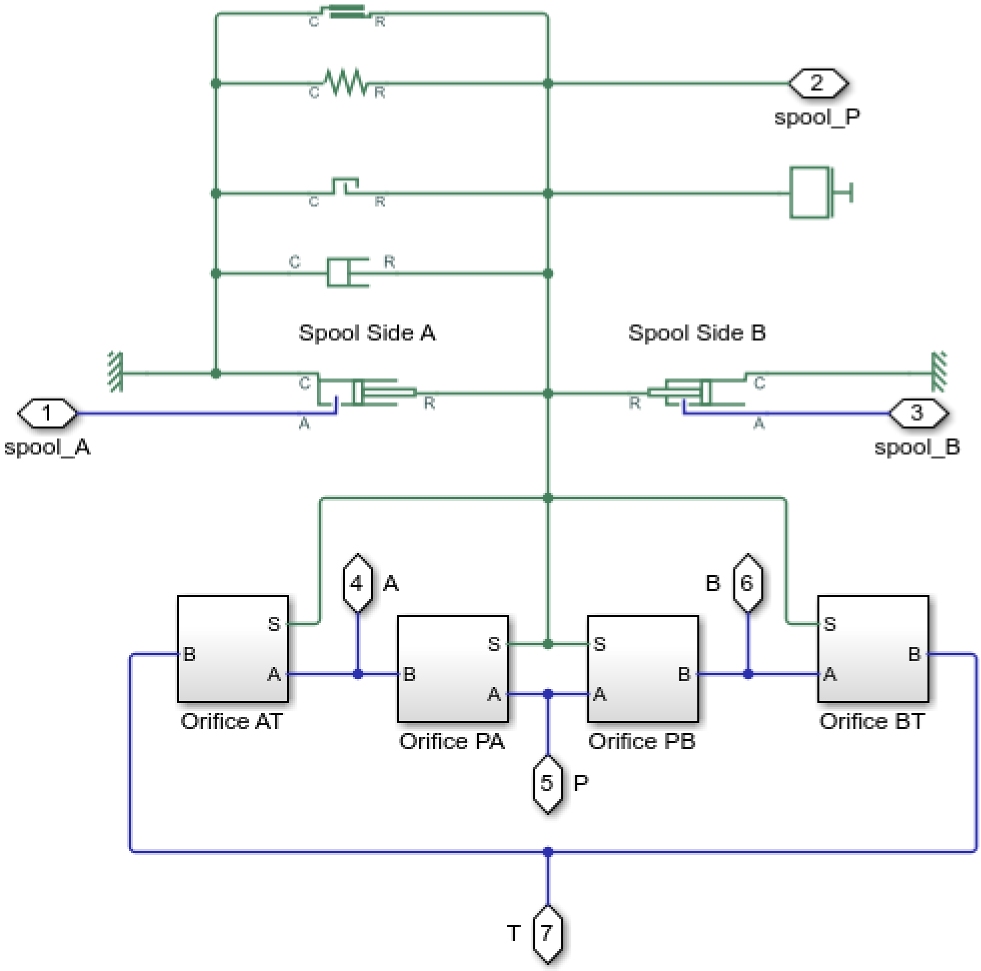
Torque Motor With Flapper Subsystem

This subsystem demonstrates how to model the torque motor and flapper. The flapper transfers the load torque to the torque motor. Forces on the flapper from the flexure tube compression, and nozzle jets result in a load torque.



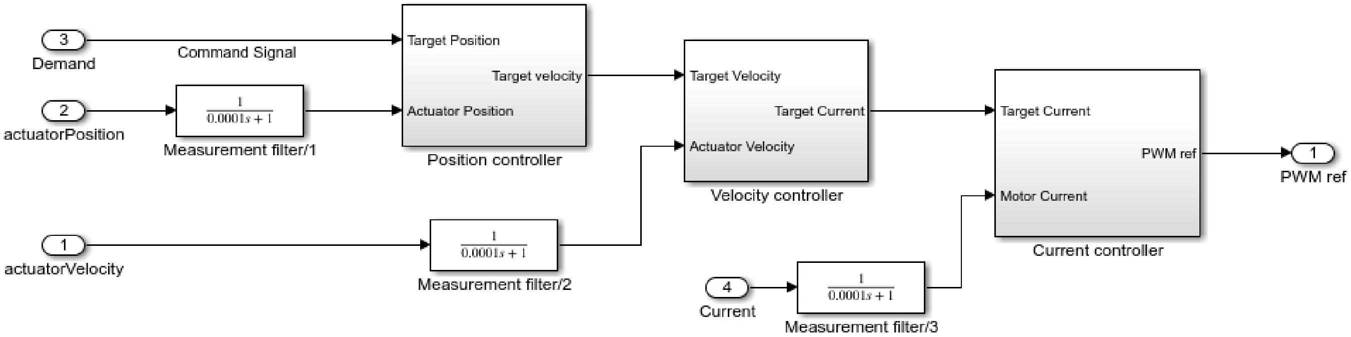
Valve Body Subsystem

This subsystem demonstrates how to model the valve body and spool dynamics including the forces on the spool due to pressure, flexure tube compression, change in flow direction, friction, damping, spool spring compression, and the hard stop. Spool position dictates whether the orifices between the P to A, A to T, P to B, and B to T ports are open or shut.



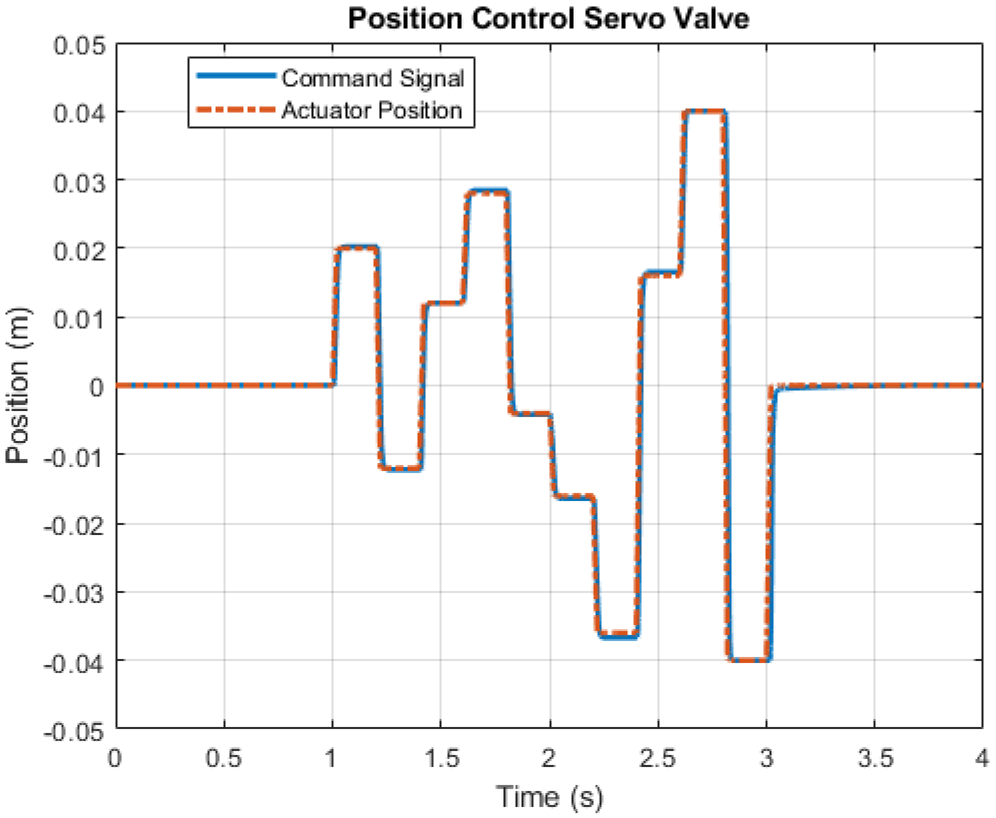
Servo Controller Subsystem

This subsystem demonstrates how to model the servo controller. There are three control loops. The inner loop controls the motor current, the middle loop controls the actuator velocity, and the outer loop controls the actuator position.



Simulation Results from Simscape™ Logging

This model generates a comparison plot between the commanded and the achieved position in the actuator with respect to the time.

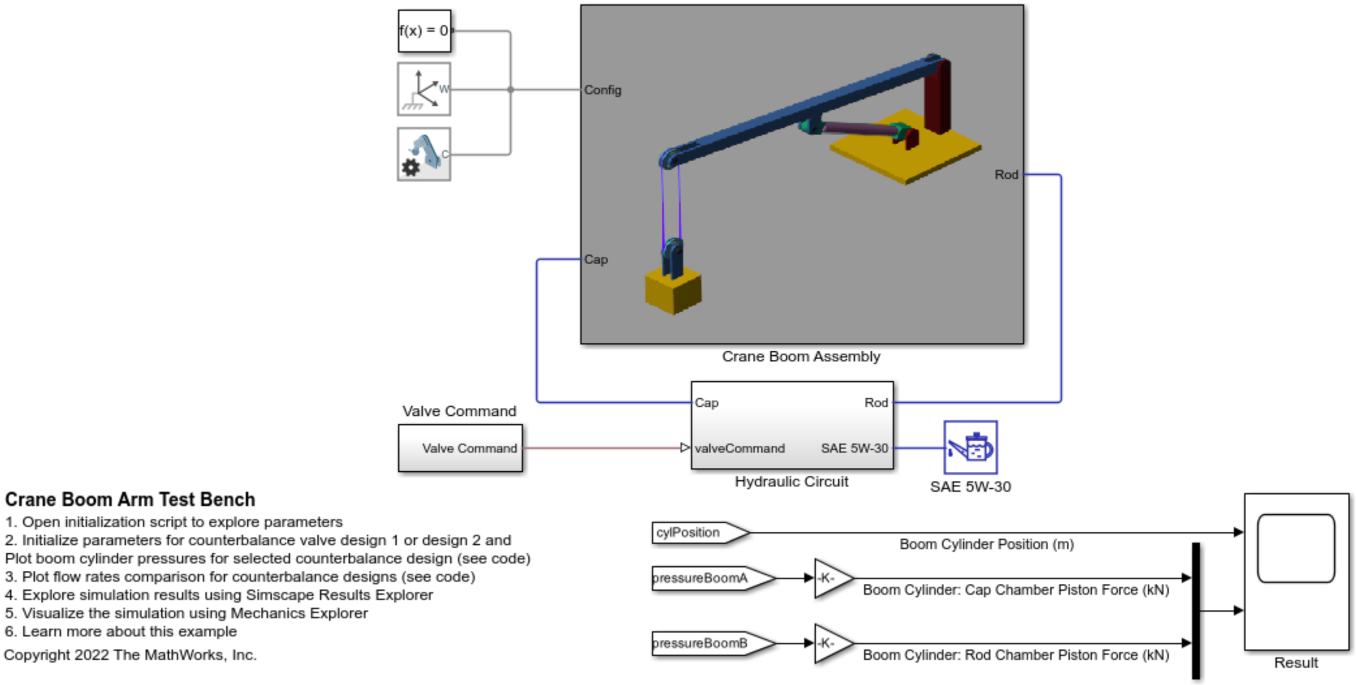


Crane Boom Arm Test Bench

This example shows how to combine hydraulics modeling in Simscape™ Fluids™ with six-degree-of-freedom motion and animation in Simscape Multibody™. This example also shows how to parameterize, test, and compare design cases for a counterbalance valve in a crane boom arm test bench. A counterbalance valve allows an upstream flow from the back port to the load port through the check valve stage during cylinder rod extension, and it allows a downstream flow from the load port to the back port through the relief valve stage during cylinder rod retraction in the boom arm mechanism. A counterbalance valve help with the load holding, load actuation, speed control, and safety by controlling the piston motion of the boom cylinder.

Model

The following figure shows the model of a crane boom arm test bench.

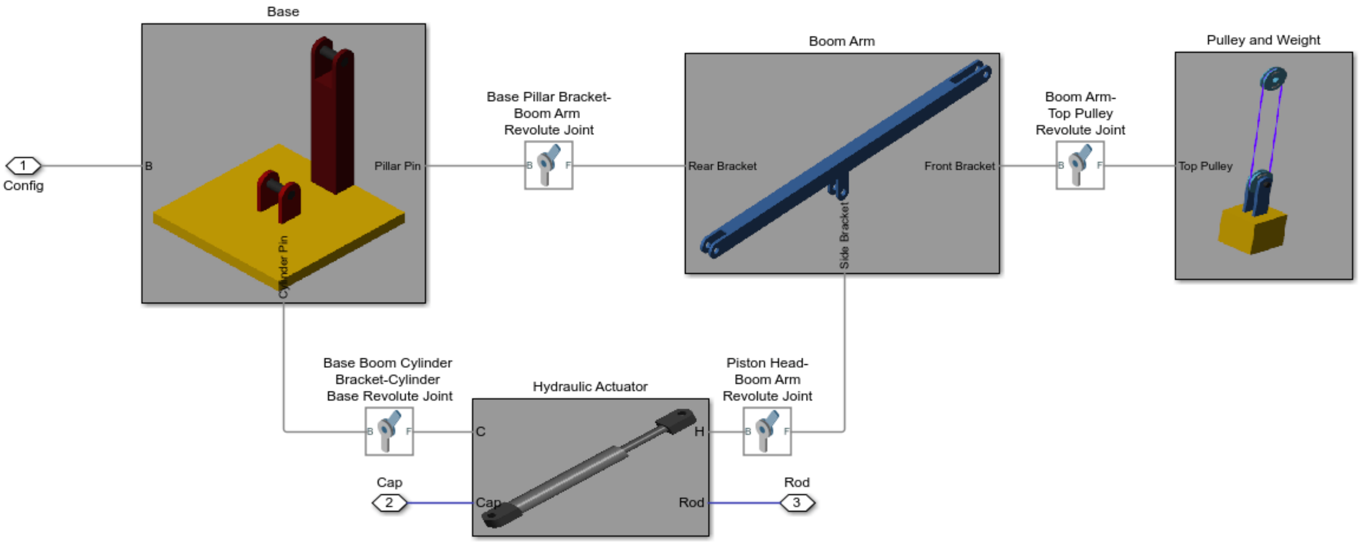


Crane Boom Arm Test Bench

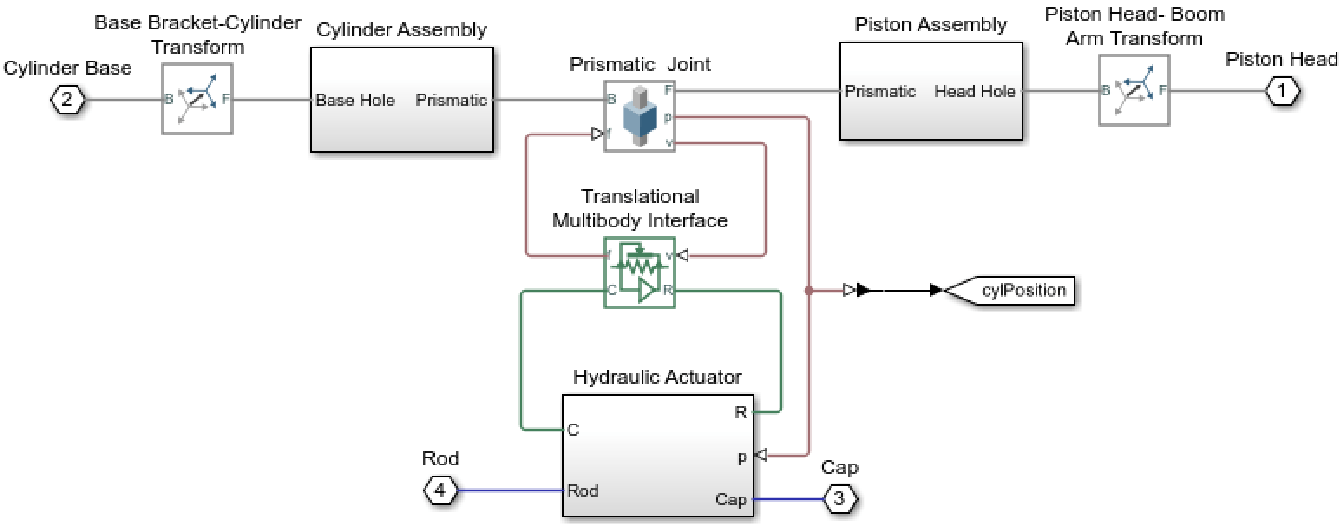
1. Open initialization script to explore parameters
2. Initialize parameters for counterbalance valve design 1 or design 2 and Plot boom cylinder pressures for selected counterbalance design (see code)
3. Plot flow rates comparison for counterbalance designs (see code)
4. Explore simulation results using Simscape Results Explorer
5. Visualize the simulation using Mechanics Explorer
6. Learn more about this example

Copyright 2022 The MathWorks, Inc.

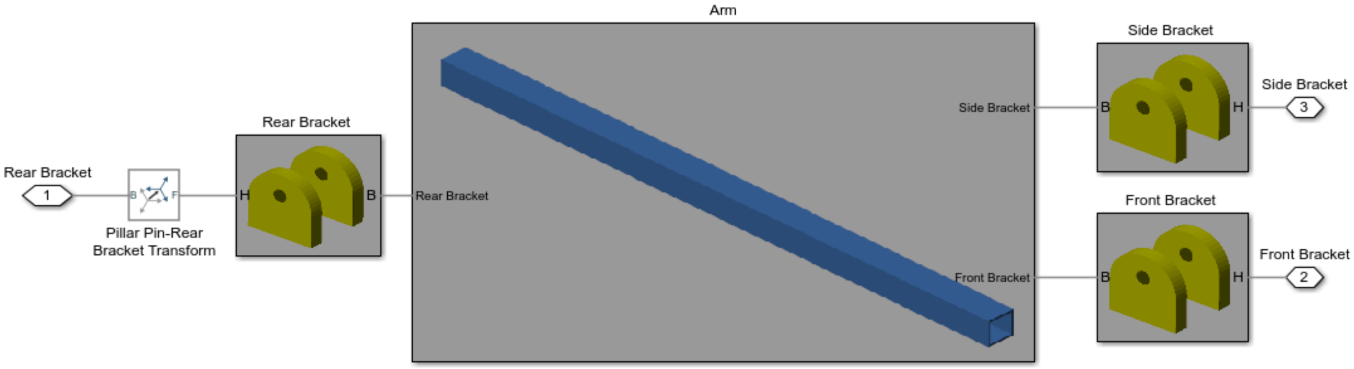
Crane Boom Assembly Subsystem: This subsystem models the crane boom system assembly.



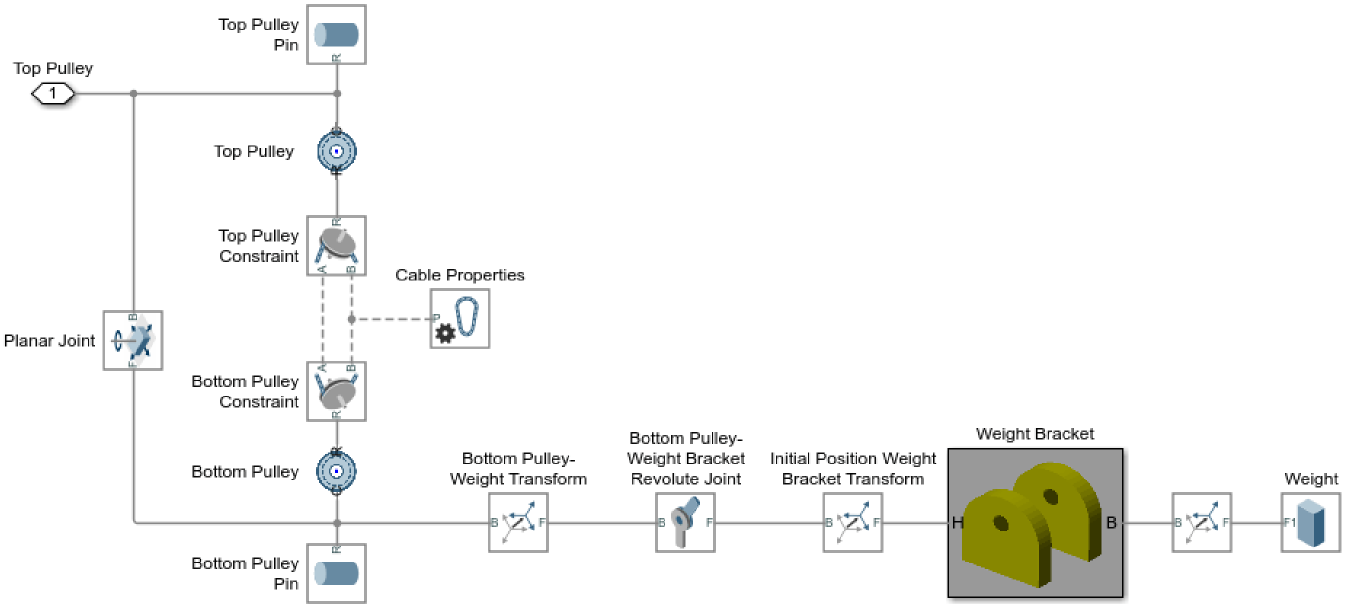
Hydraulic Actuator Subsystem: Simscape Multibody connections Cylinder Base and Piston Head are the Multibody ports associated with the bracket holes of the cylinder and piston rod respectively. Simscape Fluids connections Cap and Rod ports are associated with the cylinder and rod side hydraulic fluid respectively.



Boom Arm Subsystem: This subsystem models the multibody boom arm.



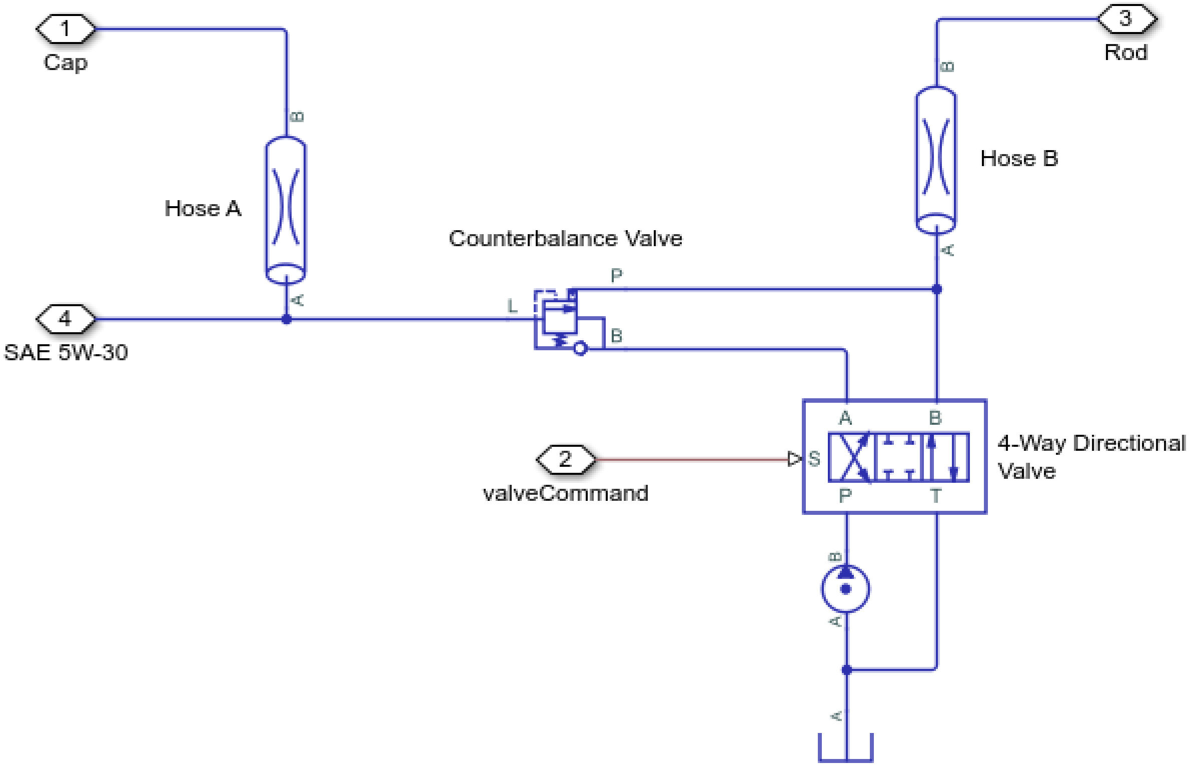
Pulley and Weight Subsystem: This subsystem models the top pulley connected to the front bracket of the boom arm and the bottom pulley connected to the top pulley with the rope. The bottom pulley is connected to the weight.



Hydraulic Circuit Subsystem: This subsystem models the hoses connecting to the boom cylinder rod and cap sides. The hoses are connected with the counterbalance valve and 4-way 3-position valve.

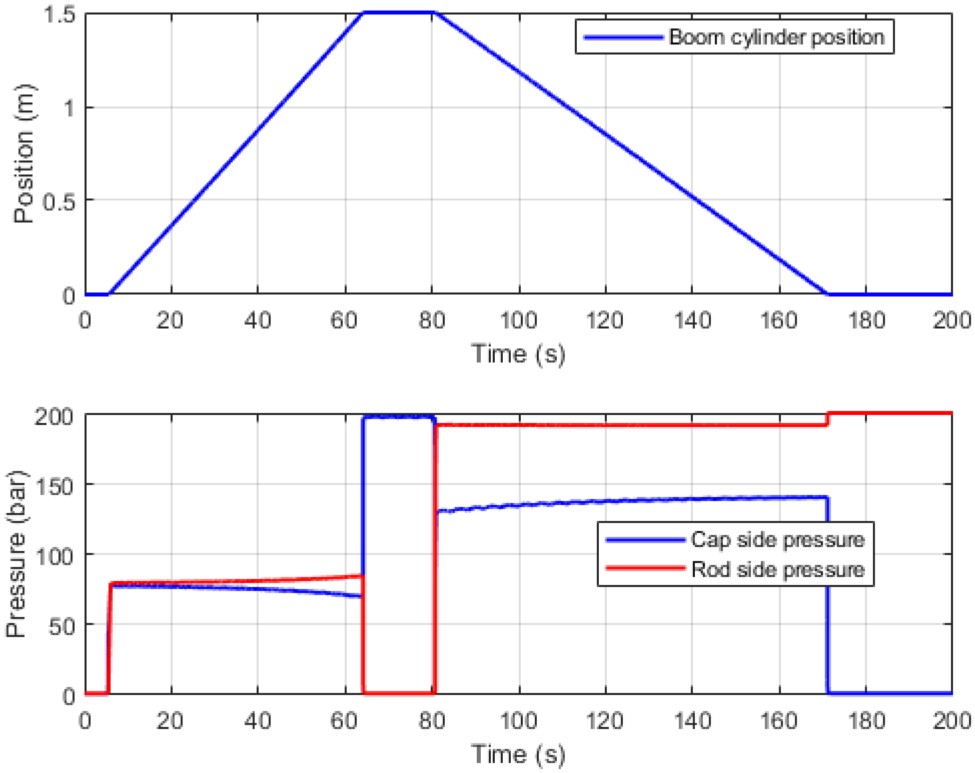
The rod extends the boom arm for the negative displacement valve command. The hydraulic fluid flows into the cap side of the boom cylinder through the check valve stage of the counterbalance valve.

The rod retracts the boom arm for the positive displacement valve command. The hydraulic fluid flows out of the cap side of the boom cylinder through the pilot assisted relief valve stage of the counterbalance valve.

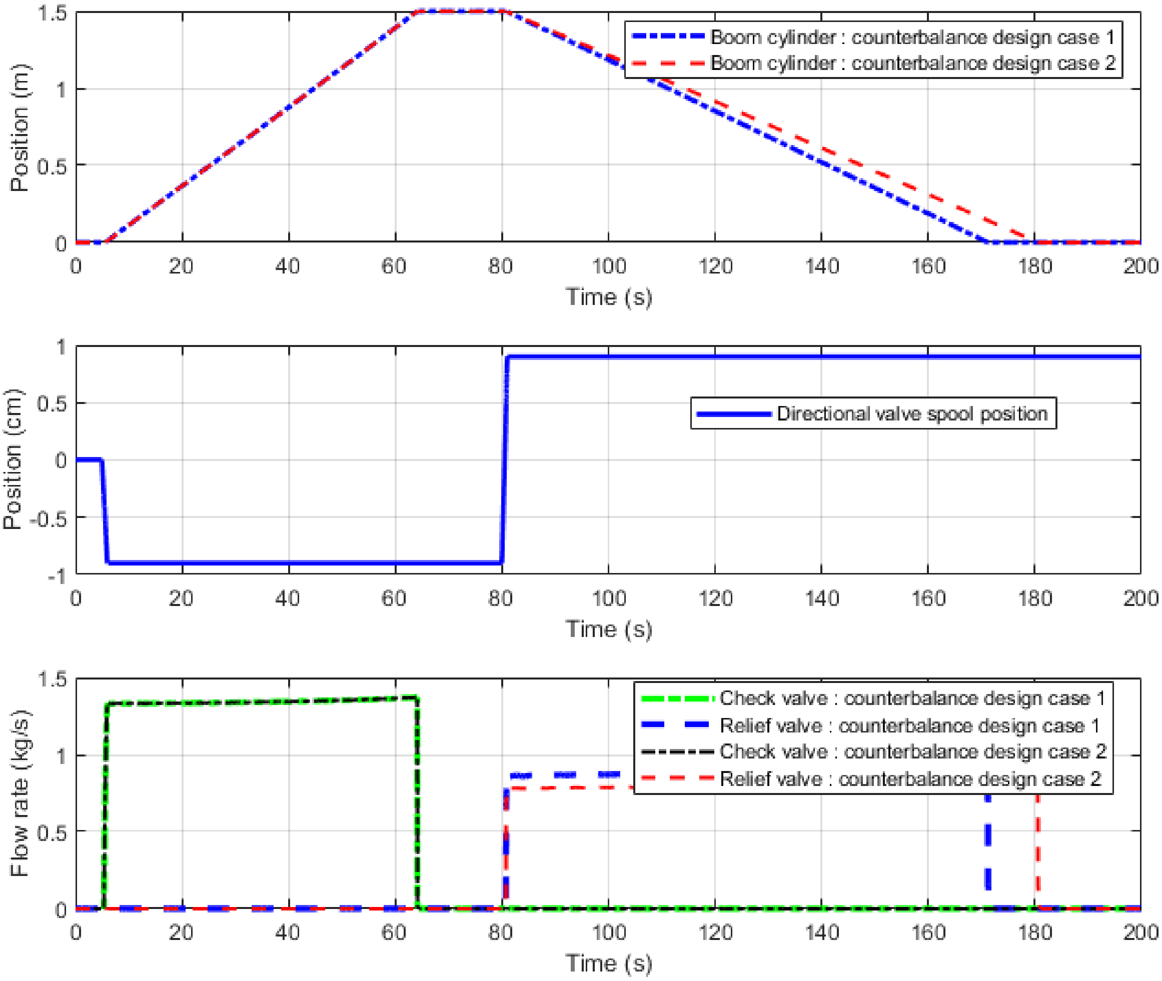


Simulation Results from Simscape Logging

This figure shows the boom cylinder position and the pressures at the cap side and rod side of the cylinder piston.

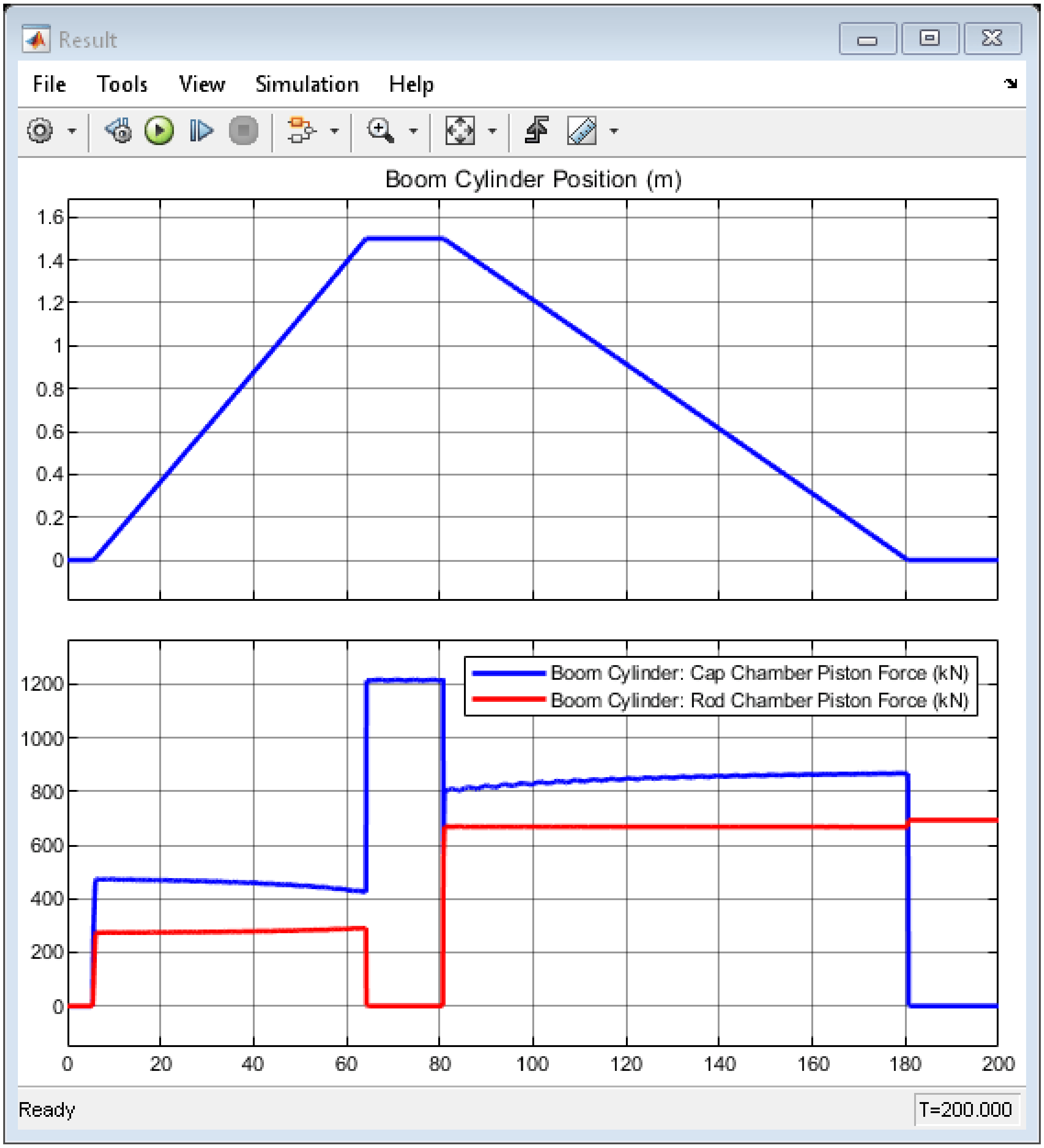


This figure shows the boom cylinder position, directional control valve spool position, and the flow rates through the counterbalance valve for the design cases 1 and 2 of the counterbalance valve. The boom cylinder extends when the directional control valve is in the negative spool position and retracts when the directional control valve is in the positive spool position.



Simulation Results from Scope

This figure shows the boom cylinder position and the hydraulic forces at the cap side and rod side of the cylinder piston.

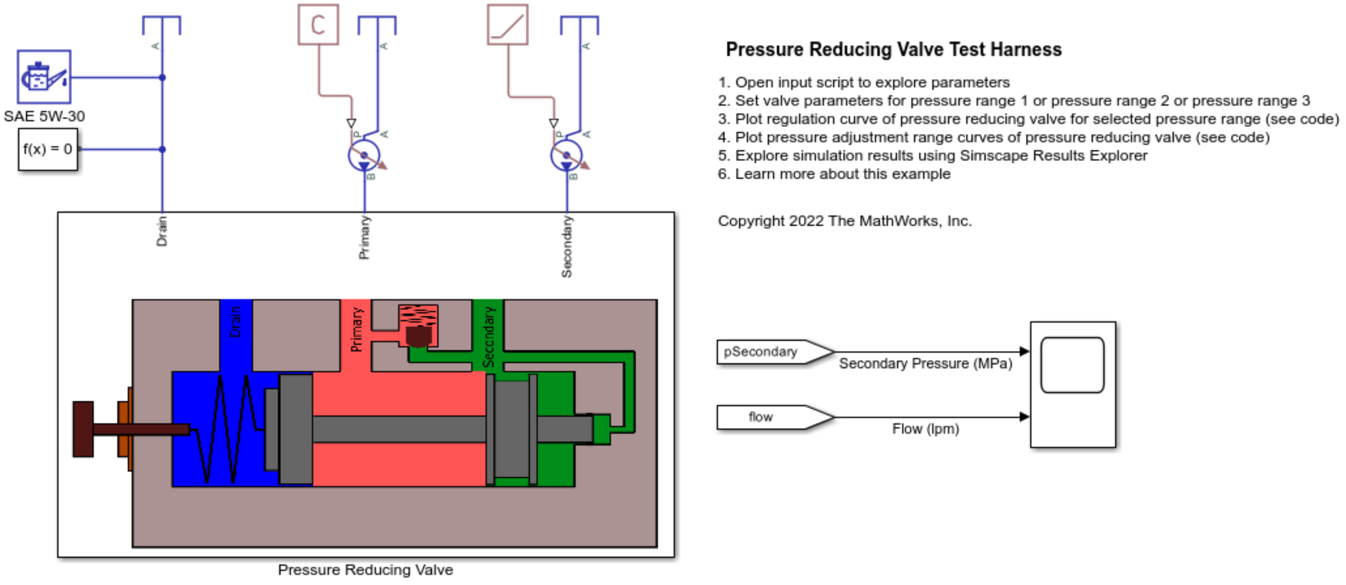

Result
File
Tools
View
Simulation
Help
Boom Cylinder Position (m)
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Boom Cylinder: Cap Chamber Piston Force (kN)
Boom Cylinder: Rod Chamber Piston Force (kN)
1200
1000
800
600
400
200
0
0
20
40
60
80
100
120
140
160
180
200
Ready
T=200.000

Pressure Reducing Valve Test Harness

This example shows how to model, parameterize, and test a pressure reducing valve. The model is used to generate a plot of the regulation curve i.e. a plot of flow vs regulated pressure. This plot can be compared with the characteristics from the manufacturer datasheet to verify the test harness. The regulation curve displays two valve functions namely the reducing function and the relieving function. The reducing valve remains open when the pressure at the secondary port is less than a specified pressure, known as the set pressure. When the pressure at the secondary port meets or surpasses the set pressure, the reducing valve closes. In the regulation curve, the region of positive flow represents the phase when the reducing valve is open and gradually closes. The pressure reducing valve used in this example has a check valve that relieves pressure when the secondary pressure is about to surpass the set pressure. During this phase, the hydraulic fluid escapes via the check valve to the primary port keeping the secondary pressure constant at the set pressure. The region of negative flow in the regulation curve represents the phase when the pressure reducing valve is closed and the check valve opens gradually. Pressure reducing valves are used in industry to limit and maintain pressure during operations like hydraulic pressing, punching, drilling and stamping.

Model

The following figure shows the model of a pressure reducing valve. The primary port of the valve is connected to a pressure source that receives a constant input. The pressure source here represents a pump or prime mover which is connected to a reservoir. The pressure at the secondary port is gradually increased with a ramp input. The drain port is connected to a reservoir. This model represents a direct acting pressure reducing valve with an external drain and a check valve.



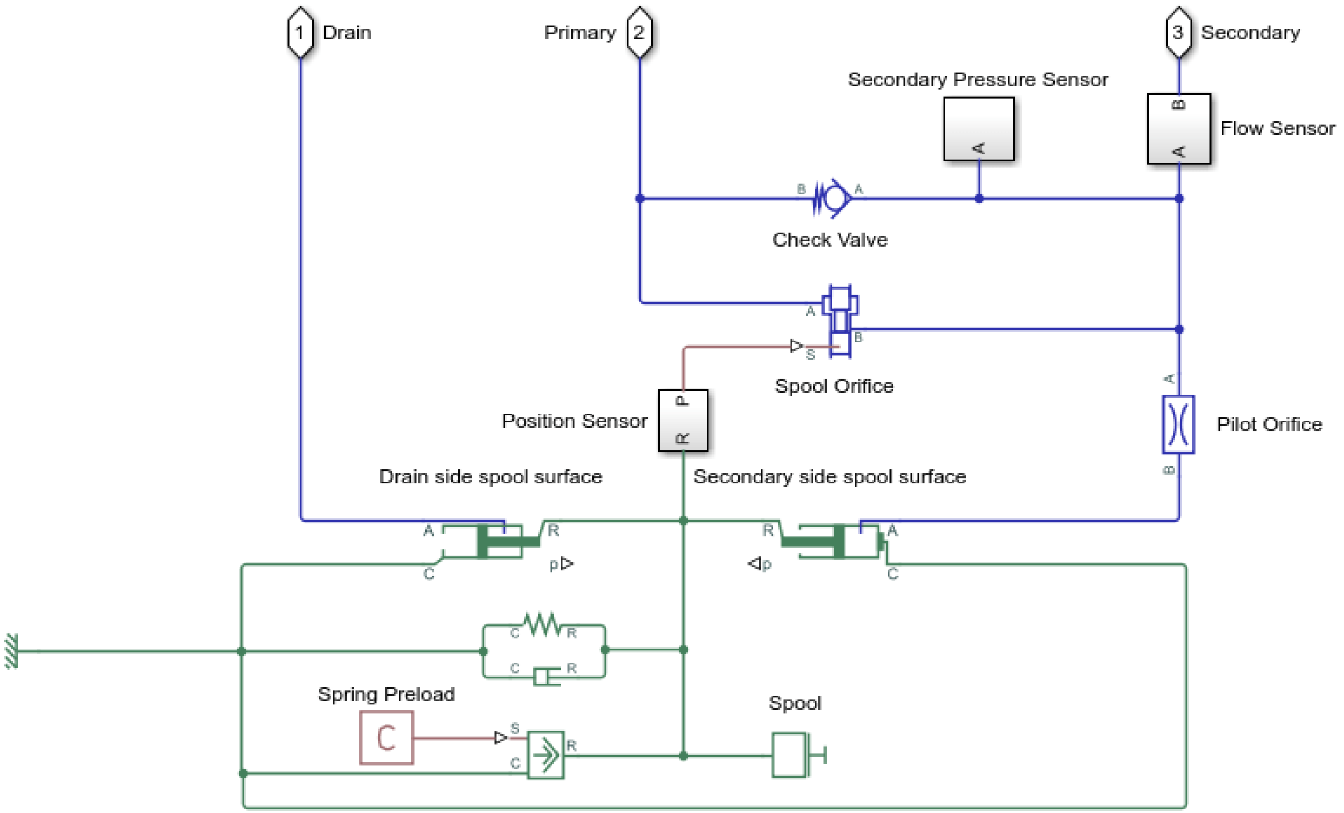
Pressure Reducing Valve Test Harness

1. Open input script to explore parameters
2. Set valve parameters for pressure range 1 or pressure range 2 or pressure range 3
3. Plot regulation curve of pressure reducing valve for selected pressure range (see code)
4. Plot pressure adjustment range curves of pressure reducing valve (see code)
5. Explore simulation results using Simscape Results Explorer
6. Learn more about this example

Copyright 2022 The MathWorks, Inc.

Pressure Reducing Valve Subsystem

A pressure reducing valve uses a spring-loaded spool to control the downstream pressure or the secondary pressure. If the secondary pressure is below the valve set pressure, fluid flows freely from the primary side to the secondary side. An internal passage transmits the fluid pressure from secondary side to the rightmost spool piston face. This pressure is called pilot pressure. The force due to pilot pressure opposes the spring load on the leftmost piston face. The spring may be preloaded and the preload can be adjusted with a handle or adjusting screw. The spring end of the spool is connected to a drain. When the secondary pressure reaches the set pressure, the spool quickly moves to close the secondary port and thus limits and maintains secondary pressure to the set value.



The pressure reducing valve test harness model is parameterized for different pressure adjustment ranges provided by the manufacturer. Each such adjustment setting is achieved by turning the handle or screw attached to the spring to the desired extent. The stroke length of the spool is different for each such setting. To determine the stroke length, the following equations have been used.

$$Q_{max} = C_d.A_{max}.\sqrt{(2.\Delta p/\rho)}$$

$$A_{max} = w.\Delta s_{max}$$

where,

1. Q_{max} is the maximum flow rate of the pressure reducing valve
2. C_d is the coefficient of discharge
3. A_{max} is the maximum opening area of the spool orifice
4. Δp is the pressure differential between the primary port and the secondary port at maximum flow
5. ρ is the fluid density

6 w is the spool orifice width

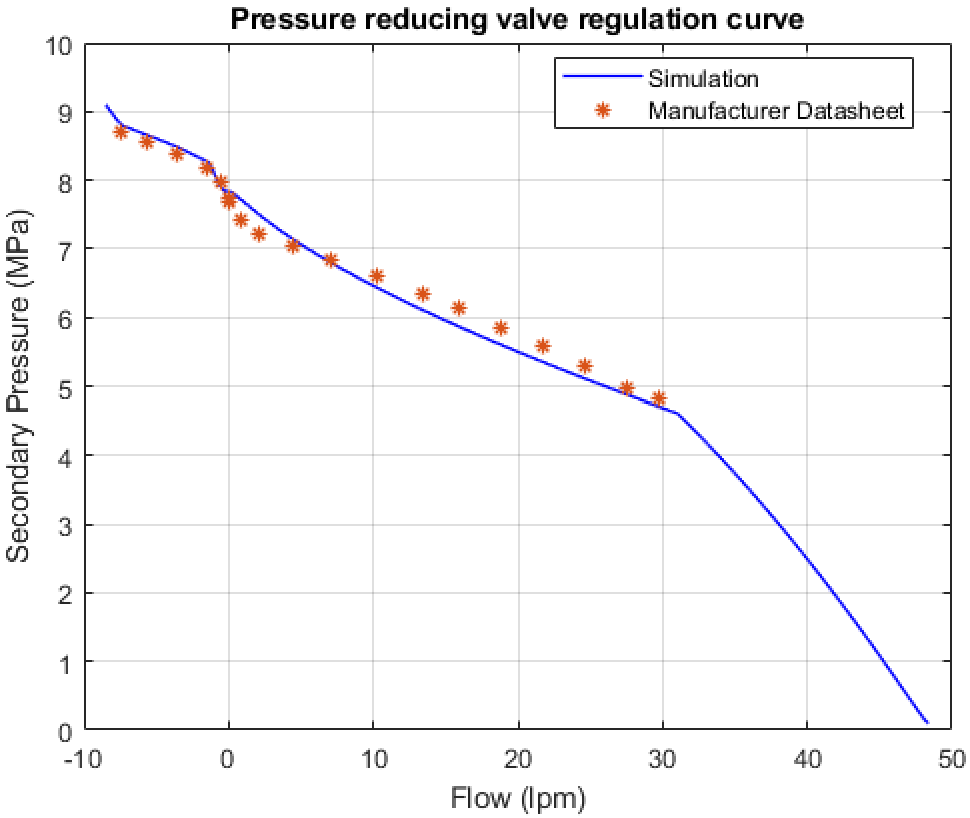
7 Δs_{max} is the spool stroke length

The values of Q_{max}, Δp, w are derived from the manufacturer datasheet. The value of C_d is assumed to be 0.64. The spring constant of the reducing valve is chosen such that the slope of the simulation output regulation curve matches that of the datasheet. The spring preload is chosen such that the secondary pressure at the maximum flow of the simulation output regulation curve matches with the datasheet.

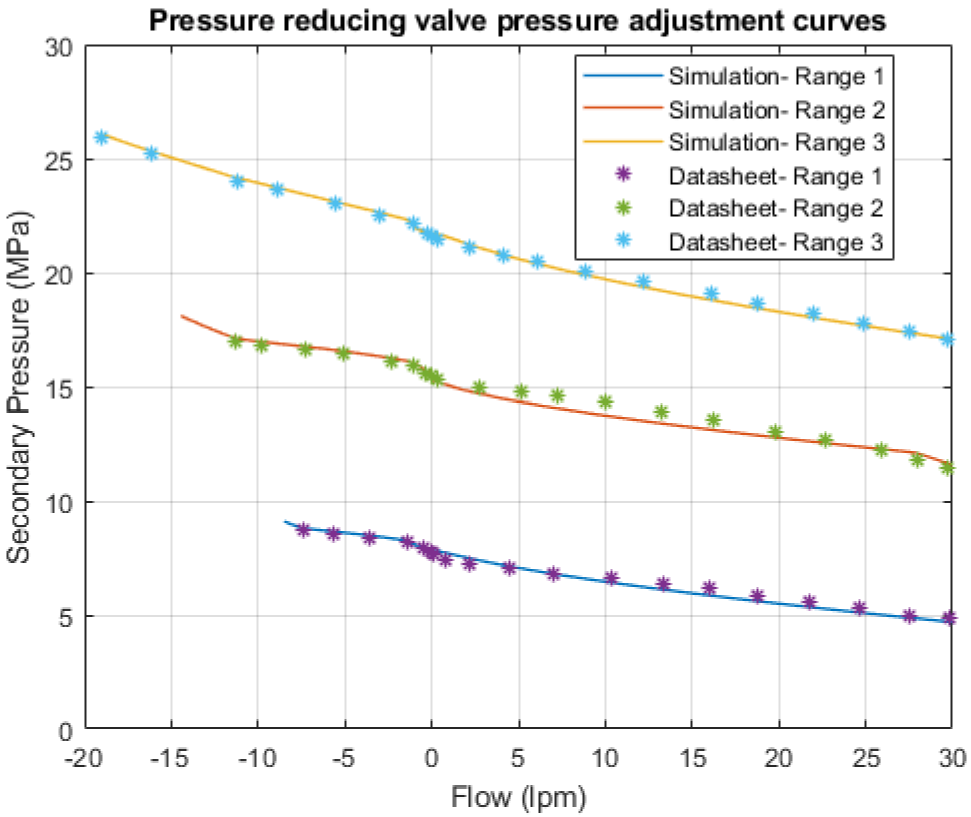
The model assumes that the flow forces due to fluid inertia and friction are negligible. The leakage area of the spool orifice is also assumed and may not match with the manufacturer data. The simulation output regulation curve does not match perfectly with the manufacturer datasheet due to certain ambiguity regarding the exact test setup and the above factors.

Simulation Results from Simscape™ Logging

The model can be used to generate a plot of the regulation curve or a plot of flow vs regulated pressure for the selected pressure adjustment range. This plot can be compared with the characteristics from the manufacturer datasheet to verify the test harness using the *sscfluids_prv_test_harness_reducing_characteristics* script. The below figure shows the regulation curve for the first pressure adjustment range.



The model can be used to generate a plot of the secondary pressure variation with the flow for three pressure adjustment ranges. This plot can be compared with the characteristics from the manufacturer datasheet using the *sscfluids_prv_test_harness_adjustment_range_characteristics* script.



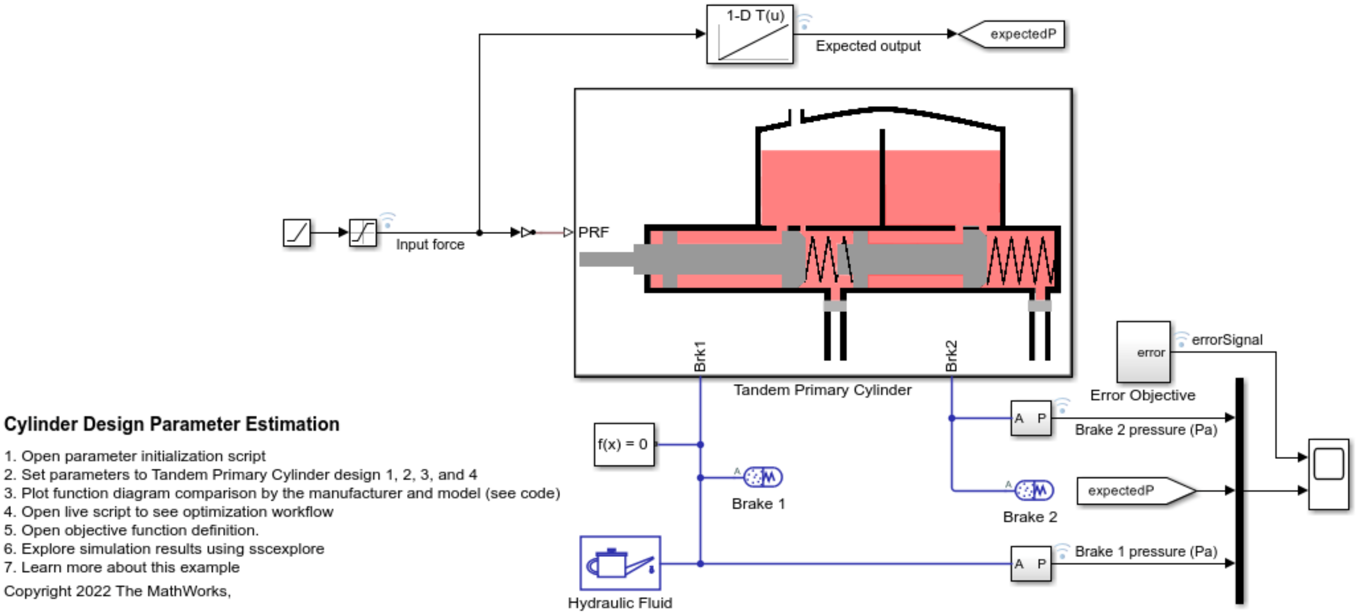
Cylinder Design Parameter Estimation

This example shows how to parameterize and test a tandem primary cylinder starting from a manufacturer's datasheet. This example uses optimization to determine remaining unknown parameters, given the numerical data extracted from the datasheet. After the model is simulated, it compares the resulting push rod force - brake pressure relationship curve to the curve provided on the manufacturer datasheet. Understanding the behavior of the tandem primary cylinder is important when selecting other braking system components.

The live script requires the Optimization Toolbox™ and Global Optimization Toolbox to be installed and licensed to run.

Model

The following figure shows the tandem primary cylinder test harness. The model uses spring-loaded accumulators for loading in the 1^{st} and the 2^{nd} circuits in place of disk or drum brakes. The input to the model is the push rod force coming from the lever mechanism or the brake booster. Output of the model is the pressures achieved in the brake circuits 1 and 2 respectively.



Cylinder Design Parameter Estimation

1. Open parameter initialization script
2. Set parameters to Tandem Primary Cylinder design 1, 2, 3, and 4
3. Plot function diagram comparison by the manufacturer and model (see code)
4. Open live script to see optimization workflow
5. Open objective function definition.
6. Explore simulation results using sscexplore
7. Learn more about this example

Copyright 2022 The MathWorks,

Manufacturer datasheet data

The supplier has provided the following data in the datasheet.

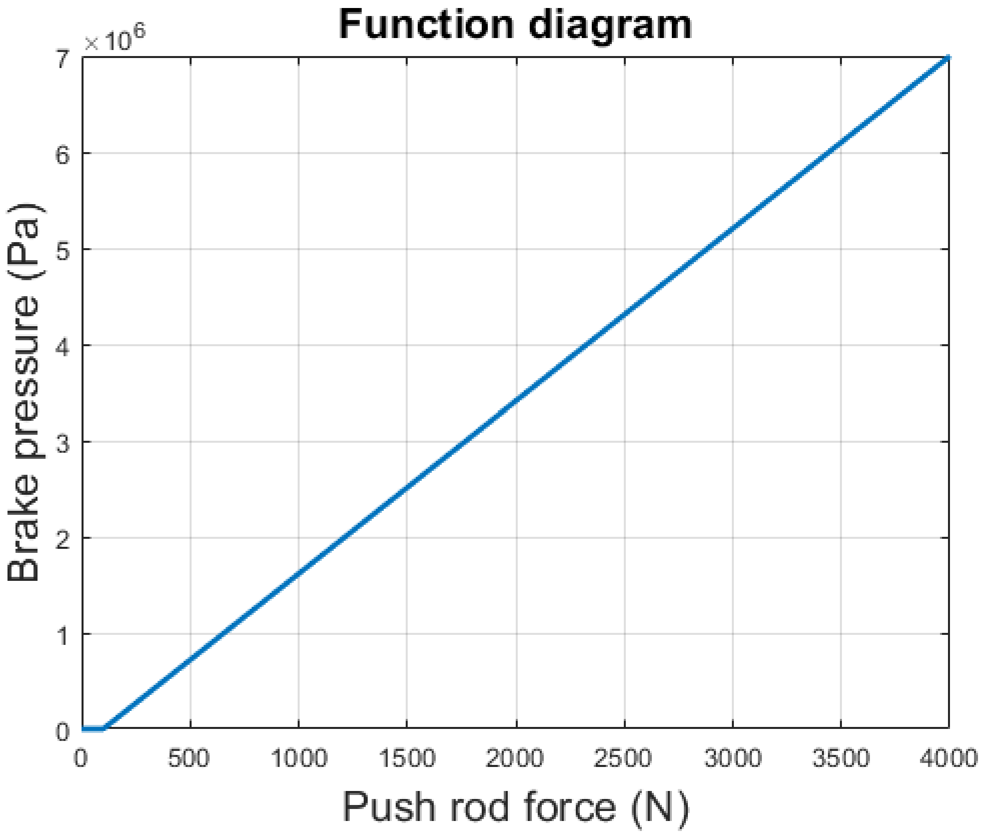
1. *tpcPistonDiameter* : first and second circuit piston diameter.
2. *tpcPistonStroke1* : first circuit piston stroke.
3. *tpcPistonStroke2* : second circuit piston stroke.
4. *tpcTotalStroke* : overall stroke.
5. *tpcCircuit1Disp* : first circuit displacement.
6. *tpcCircuit2Disp* : second circuit displacement.

7 *tpcTotalDisp* : net displacement.

8 _tpcMaxPressure _ : max pressure.

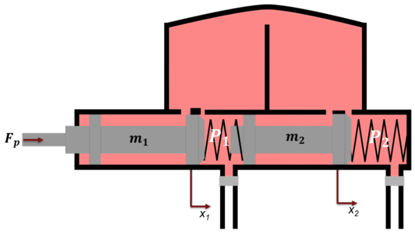
Pressure development characteristics

The supplier datasheet gives the following function diagram for the push rod force-brake pressure curve for the tandem primary cylinder which should be followed by the brake circuit 1 and 2 as per supplier catalogue. It is required for a tandem master cylinder to have same brake pressures created in brake circuits for uniform braking.



Equations of motion (EoM) of system

The following figure shows the tandem primary cylinder. Mass 1 and mass 2 use the coordinate frames shown below. This description only includes the mechanical EoM of the system. The fluid dynamics equations are not shown.



$$m_1\frac{d^2x_1}{dt^2} + c_1(\frac{dx_1}{dt} - \frac{dx_2}{dt}) + k_1(x_1 - x_2) = F_p - p_1A - Preload_1 + R_1, \text{ when } x_1 = 0$$

$$m_2\frac{d^2x_2}{dt^2} + c_1(\frac{dx_2}{dt} - \frac{dx_1}{dt}) + c_2\frac{dx_2}{dt} + k_1(x_2 - x_1) + k_2x_2 = p_1A - p_2A + Preload_1 - Preload_2 + R_2, \text{ when } x_2 = 0$$

$$m_1\frac{d^2x_1}{dt^2} + c_1(\frac{dx_1}{dt} - \frac{dx_2}{dt}) + k_1(x_1 - x_2) = F_p - p_1A - Preload_1, \text{ when } x_1 > 0$$

$$m_2\frac{d^2x_2}{dt^2} + c_1(\frac{dx_2}{dt} - \frac{dx_1}{dt}) + c_2\frac{dx_2}{dt} + k_1(x_2 - x_1) + k_2x_2 = p_1A - p_2A + Preload_1 - Preload_2, \text{ when } x_2 > 0$$

where,

x_1 and x_2 are positions of piston mass 1 and mass 2 from the left side assumed hard stop.

m_1 and m_2 are masses of piston 1 and 2 respectively.

c_1 and c_2 are damping coefficients of piston 1 and 2 respectively.

k_1 and k_2 are stiffness coefficients of left (Spring 1) and right (spring 2) respectively.

F_p is force applied on push rod of primary cylinder.

P_1 and P_2 are pressures in brake circuit 1 and 2 respectively.

R_1 and R_2 are reaction forces on mass 1 and 2 respectively when both masses are at the leftmost position.

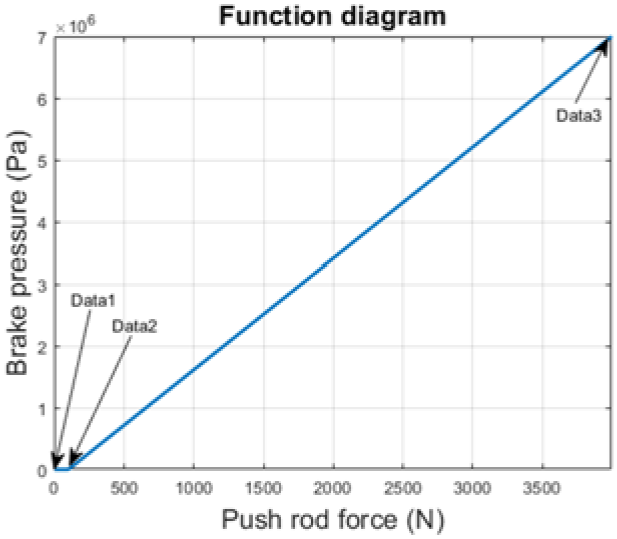
$Preload_1$ and $Preload_2$ are preloads on left and right side springs respectively.

Parameter Estimation Scheme

The following steps outline how to estimate unknown feasible parameters of the tandem primary cylinder.

1. Identify Parameters to be Estimated

Some parameters, which are required for component model simulation, are not provided in the manufacturer's datasheet. Therefore, in order to validate the model with supplier function diagram it is necessary to estimate these parameters. The supplier function diagram gives quasi- steady state behaviour of the tandem primary cylinder.



Where,

Data1, Data2, and Data3 are 3 points on the manufacturer function diagram.

Data1 point represents the leftmost points of the pistons in their respective coordinate frames.

Data3 point represent the right most points of the pistons in their respective coordinate frame, which are given in the datasheet.

Data2 point represent pistons movement at which brake circuits 1 and 2 get disconnected from the brake oil reservoir by the closing of the compensating ports. Movement beyond Data2 point in the right direction results in pressure building in the brake circuits. These points are not given in the datasheet and have to be estimated.

Left and right spring stiffness coefficients and pre-compressions are unknown.

Therefore there are 6 unknown parameters.

1. tpcPosOpen1, which is piston 1 movement up to which compensating port remains connected with the brake circuit 1.
2. tpcPosOpen2, which is piston 2 movement up to which compensating port remains connected with the brake circuit 2.
3. tpcStiffness1Coeff, which is first spring stiffness coefficient
4. tpcStiffness2Coeff, which is second spring stiffness coefficient
5. tpcSpring1Comp, which is first spring pre-compression displacement
6. tpcSpring2Comp, which is second spring pre-compression displacement. This parameter can be computed from the constraint that the Preload1 and Preload2 are equal at rest in the steady state of the system.

The datasheet does not provide the step test data, therefore the overshoot, rise time, settling time, and recovery time are not known. Therefore, the inertia and damping coefficient can't be estimated.

2. Provide Lower, Upper Bound, Scaling Factor and Initialization Point for the Parameter Estimation through Optimization

Defining scaling factor for better numerical performance of the optimization algorithm

Defining lower bound vector for optimization

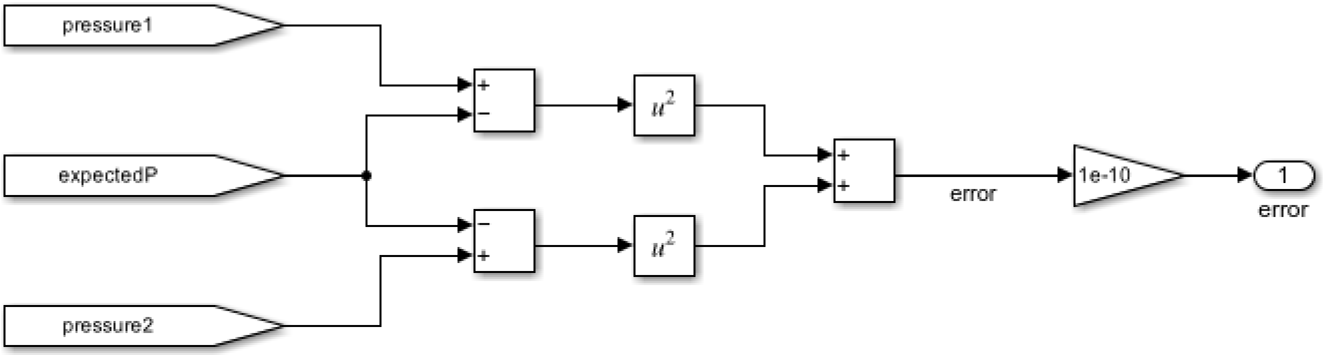
Defining upper bound vector for optimization

Computing initialization point between lower and upper bound for the optimization algorithm for parameter estimation.

Defining random initialization point for the optimization

3. Define Objective Function

The optimization algorithm minimizes the objective function to estimate the required unknown parameters. The function used here defines the least squared error between the simulation output and the expected output from the datasheet. The Error Objective subsystem computes the least squared error vector.

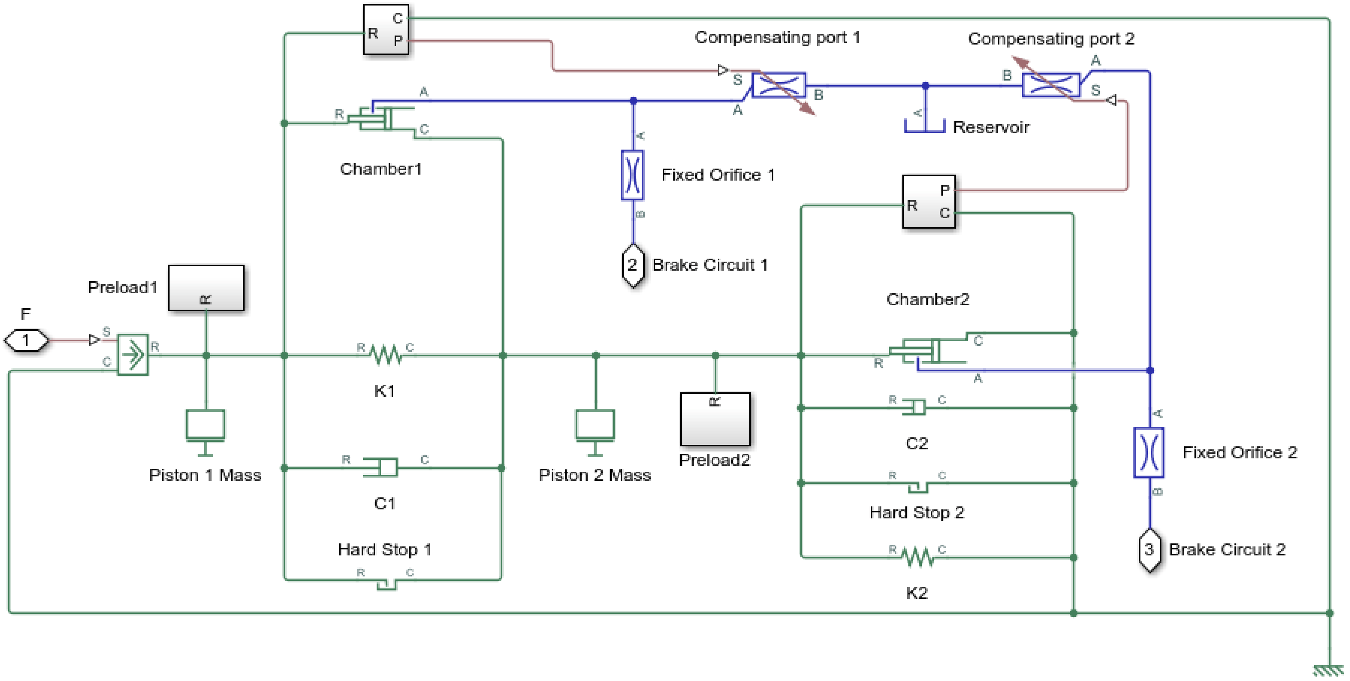


Open objective function definition file to see the cumulative error estimation code.

The optimization algorithm assumes that the objective function has one input `x`, where `x` has as many elements as the number of unknown variables in the problem. The objective function computes the scalar value of the objective function and returns it in its single output argument `y`.

4. Setting up model for Fast Restart

You can save computational time by setting up the model in Fast Restart mode. Set parameter configurability to Run-Time so that optimization algorithm can be executed in Fast Restart mode. This requires modeling spring pre-compressions and preload through Ideal force Source blocks in the model.



5. Parameter Estimation Using Optimization Algorithm

It is important to choose an algorithm which matches the problem complexity and type. No single optimization algorithm can perform the same for all types of problems.

Defining objective function

Defining lower bound

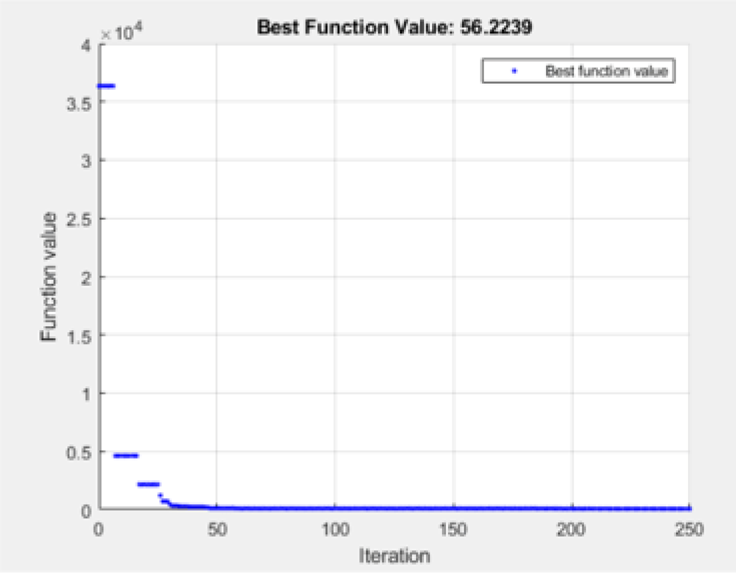
Defining upper bound

Defining optimization initialization point

You can use a hybrid algorithm to estimate unknown parameters. In a hybrid algorithm, multiple algorithms are re used together. For example, first a Global optimization algorithm, such as Simulated Annealing, could perform the optimization iterations, and then feed the output to a gradient based optimization algorithm such as fmincon to refine the optimization output.

You can also use other existing optimization algorithm functions such as Pattern Search, lsqnonlin, Genetic Algorithm, or Particle Swarm. However, you need to reconfigure optimization options for better results.

You can use Surrogate optimization to estimate the unknown parameters. Here optimization is performed using Surrogate optimization.



```
surrogateopt stopped because it exceeded the function evaluation limit set by
'options.MaxFunctionEvaluations'.
```

The estimated parameters are converted to their respective values through the defined scale.

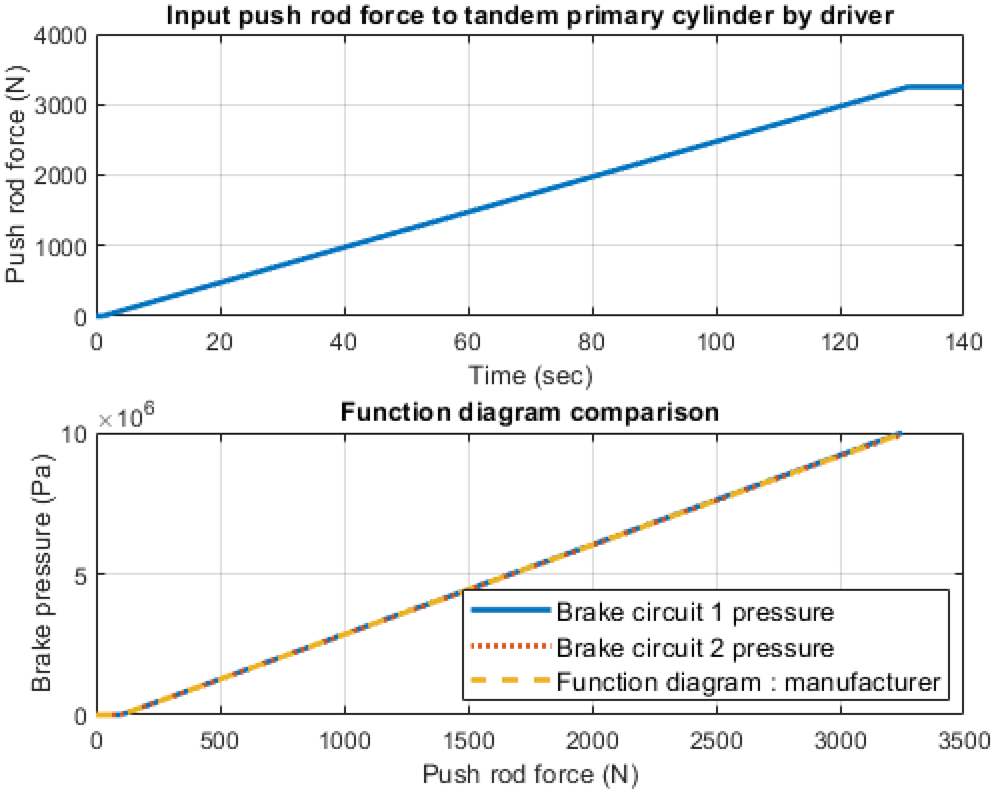
Converting scaled parameters to actual values.

Switching off Fast Restart

Simulation

The model generates push rod force-brake pressure plots for the selected manufacturer design based on the opimization algorithm parameter estimation. The applied push rod force to the tandem primary cylinder is ramped at 25 N/sec rate in the simulation .

Plotting output from Simscape™ tandem primary cylinder model and comparing with function diagram


Input push rod force to tandem primary cylinder by driver
Push rod force (N)
4000
3000
2000
1000
0
0
20
40
60
80
100
120
140
Time (sec)
Function diagram comparison
×10^6
10
5
0
Brake pressure (Pa)
Brake circuit 1 pressure
Brake circuit 2 pressure
Function diagram : manufacturer
0
500
1000
1500
2000
2500
3000
3500
Push rod force (N)

Refrigeration Cycle (System-Level)

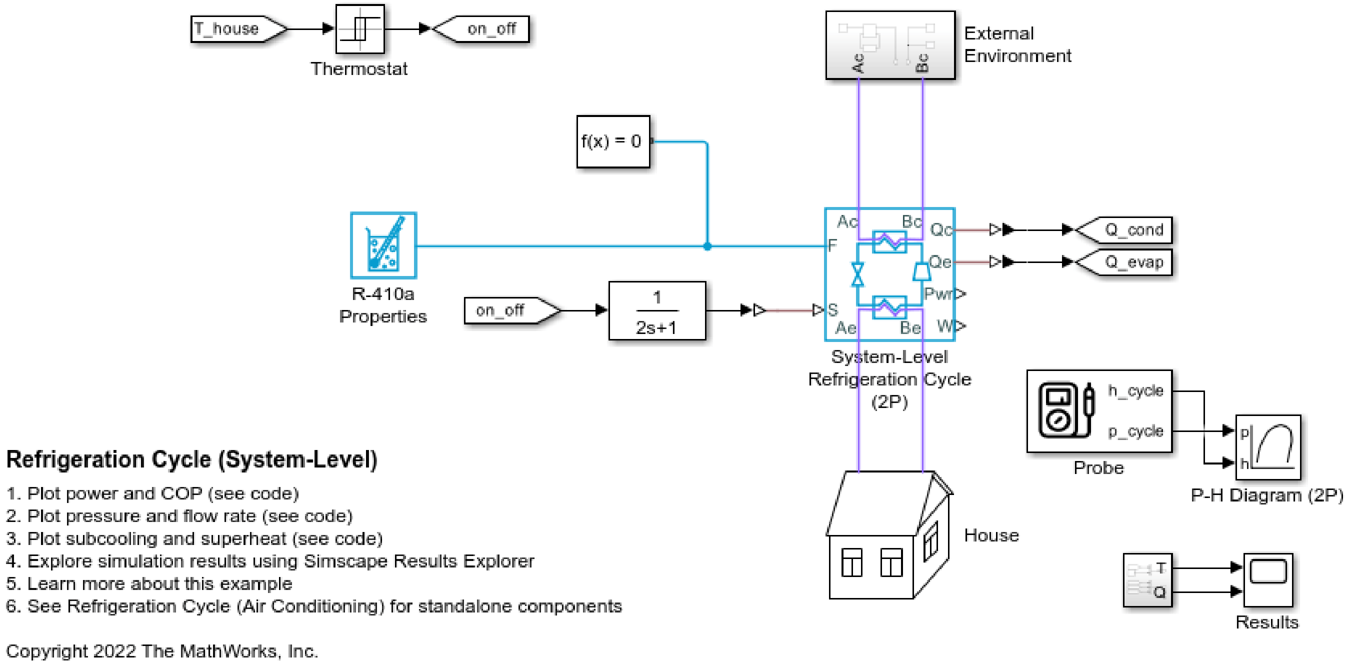
This example shows how to model a refrigeration cycle for a home air conditioning system at an abstract system level using the System-Level Refrigeration Cycle (2P) block. This block simplifies the set up of the refrigeration cycle by encapsulating the entire refrigerant loop in one block.

The System-Level Refrigeration Cycle (2P) block consists of several internal components. The evaporator absorbs heat from the air in the house and turns the refrigerant into a superheated vapor. The compressor then pressurizes and drives the refrigerant through the condenser, where the heat absorbed by the refrigerant and the compression work is rejected to the external environment. This causes the refrigerant to condense into a subcooled liquid which is then stored in the liquid receiver. The valve controls the amount of refrigerant that flows from the liquid receiver to the evaporator to maintain the desired amount of superheat. The valve also causes a drop in pressure which cools the refrigerant and allows it to absorb heat in the evaporator.

The cooling load in this model is the house, represented as a volume of air in the moist air domain. A thermal network models heat transfer between the hot external environment and the air in the house via the walls, roof, and windows. Additionally, occupants and appliances generate heat inside the house. A fan circulates air between the house and the evaporator for cooling. The system is controlled by a thermostat which turns the system on and off to maintain a temperature of 22 degC.

See the example “Refrigeration Cycle (Air Conditioning)” on page 6-204 for how to model a refrigeration cycle using standalone components.

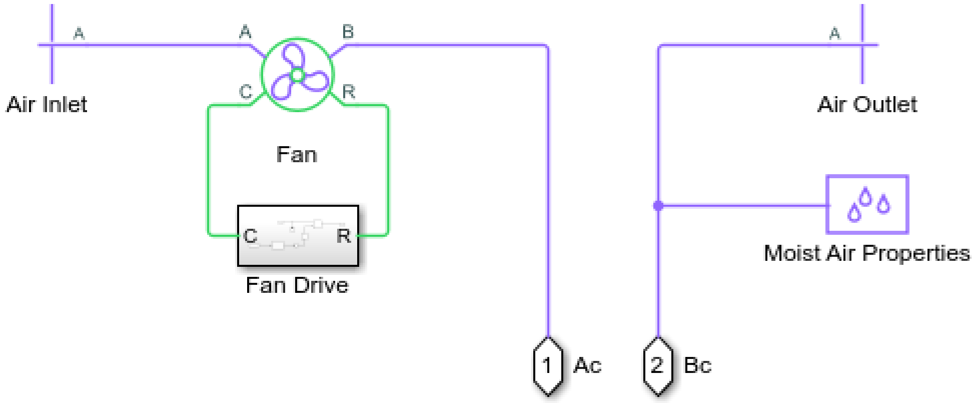
Model



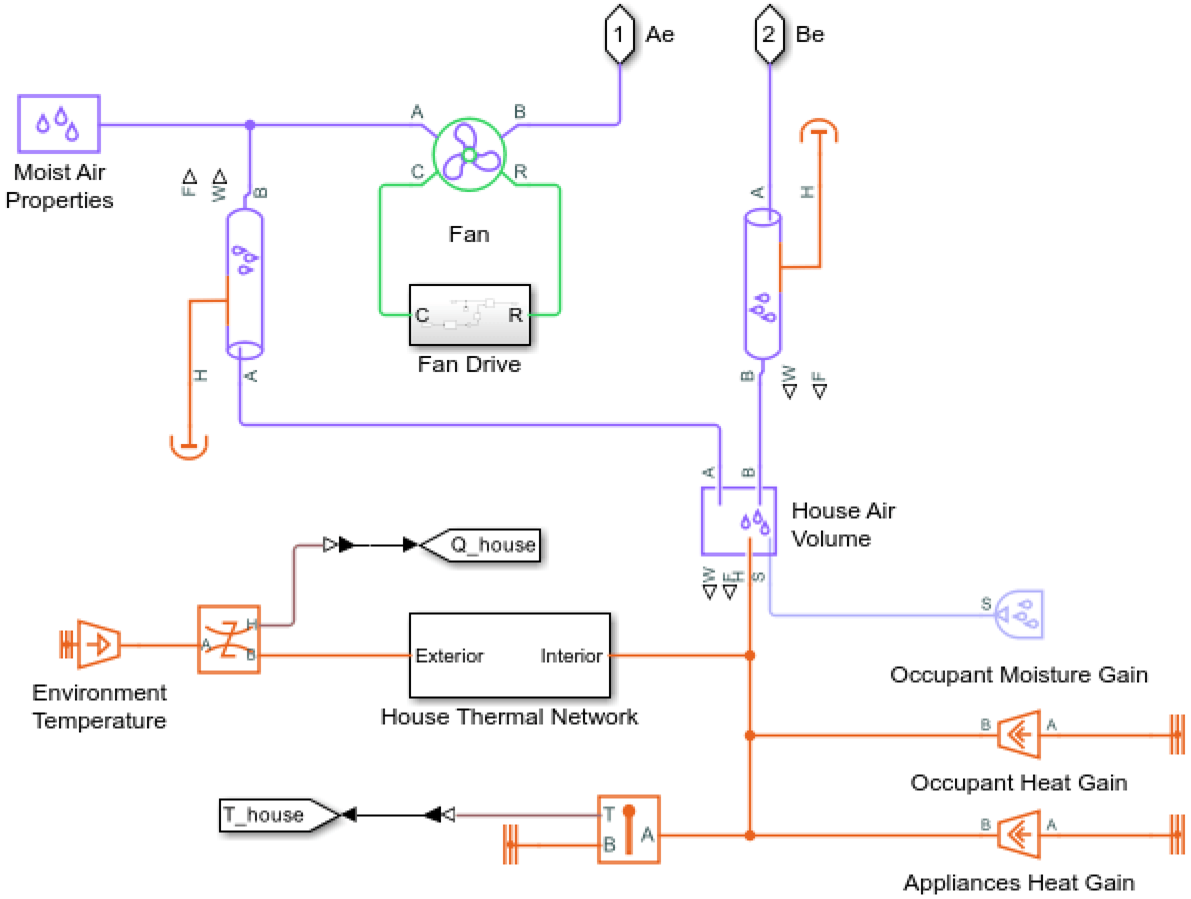
Refrigeration Cycle (System-Level)

1. Plot power and COP (see code)
2. Plot pressure and flow rate (see code)
3. Plot subcooling and superheat (see code)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example
6. See Refrigeration Cycle (Air Conditioning) for standalone components

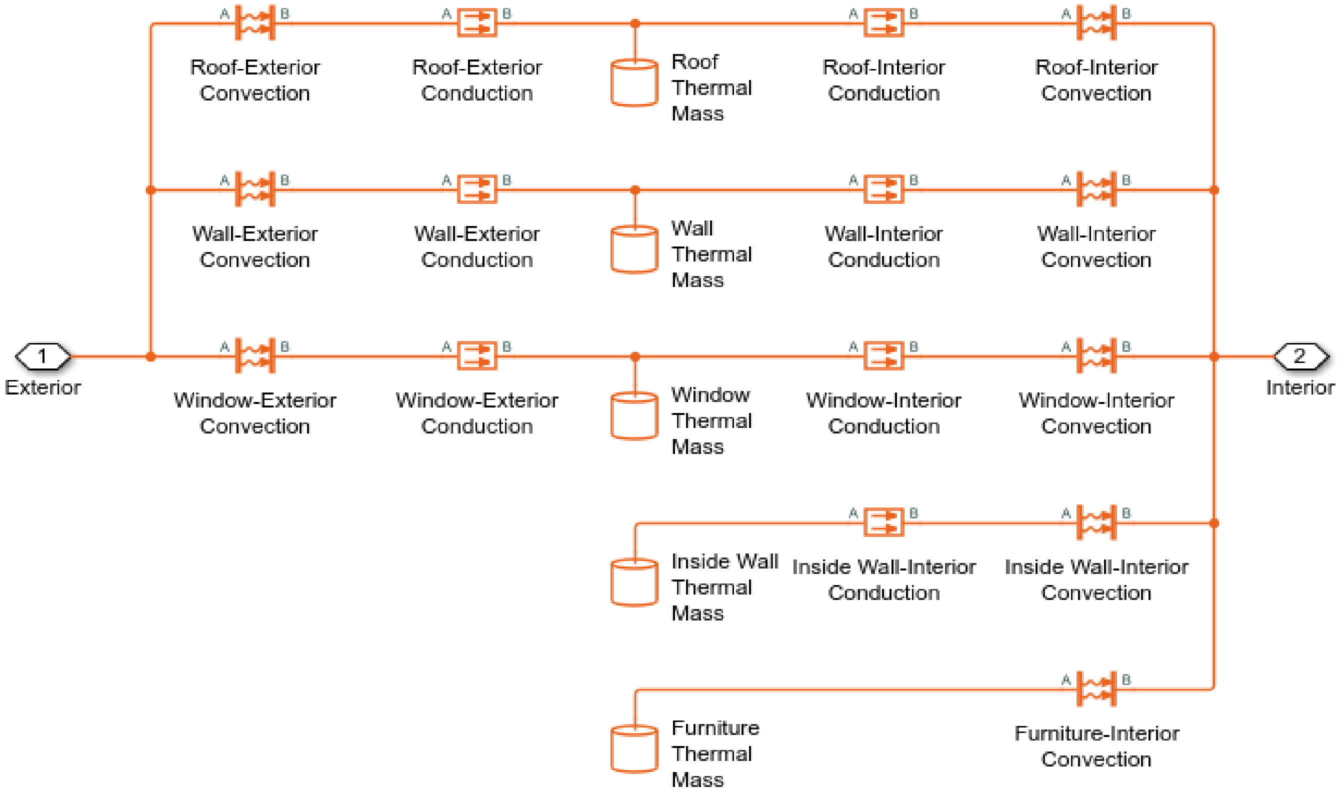
External Environment Subsystem



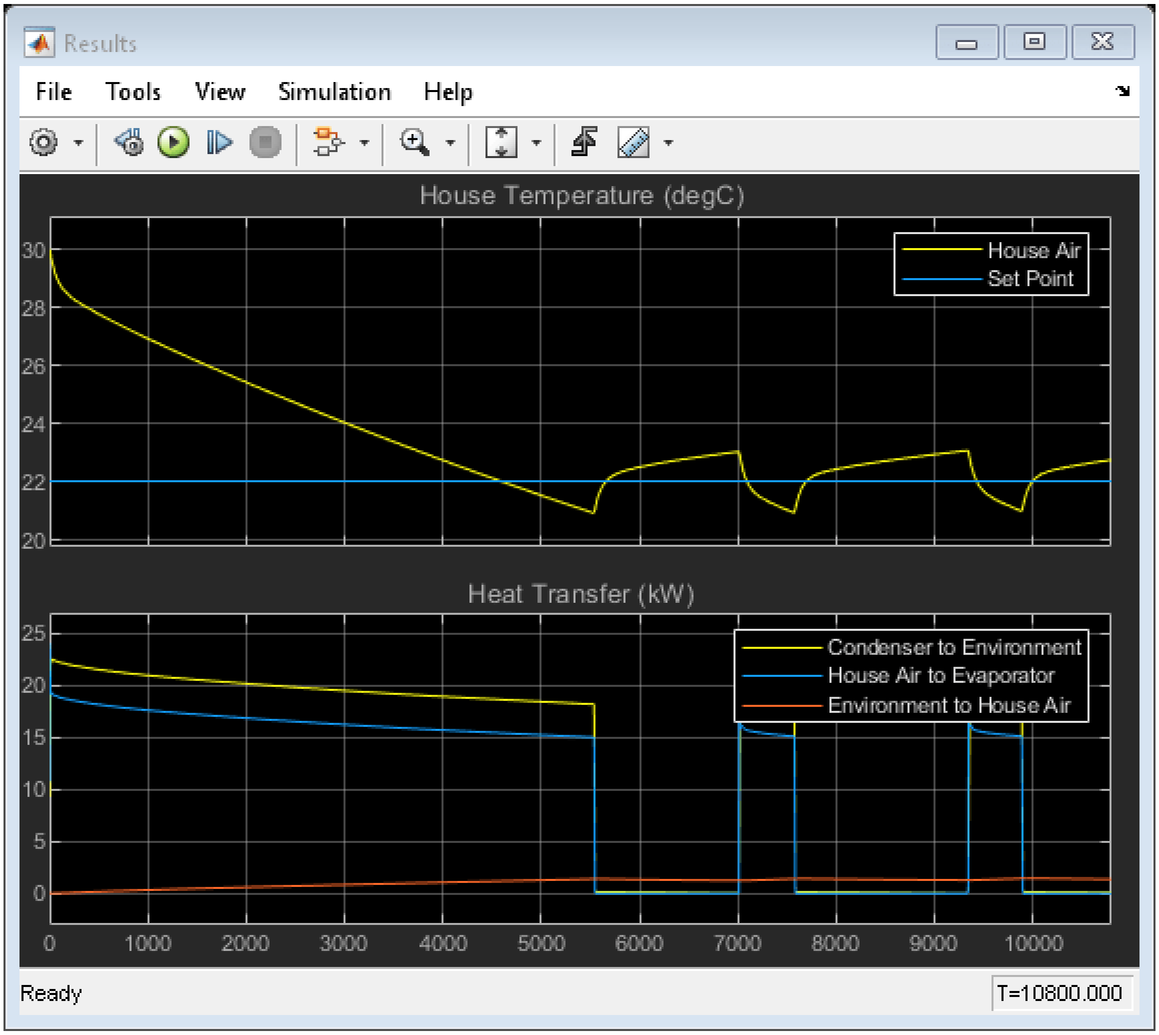
House Subsystem



House Thermal Network Subsystem

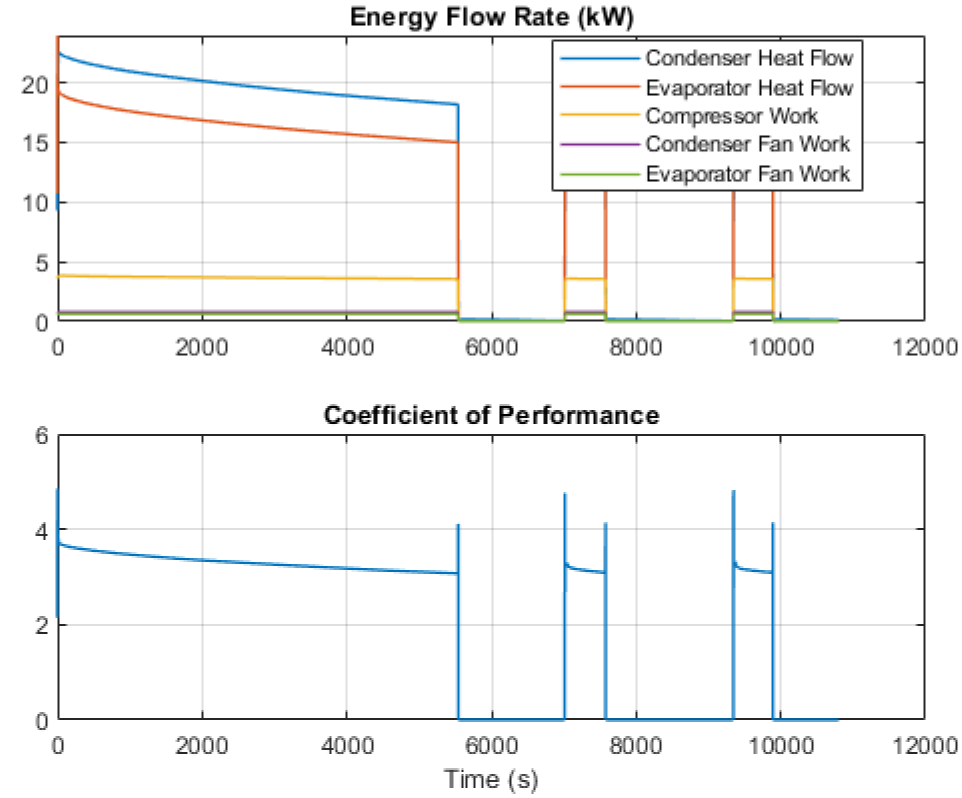


Simulation Results from Scopes

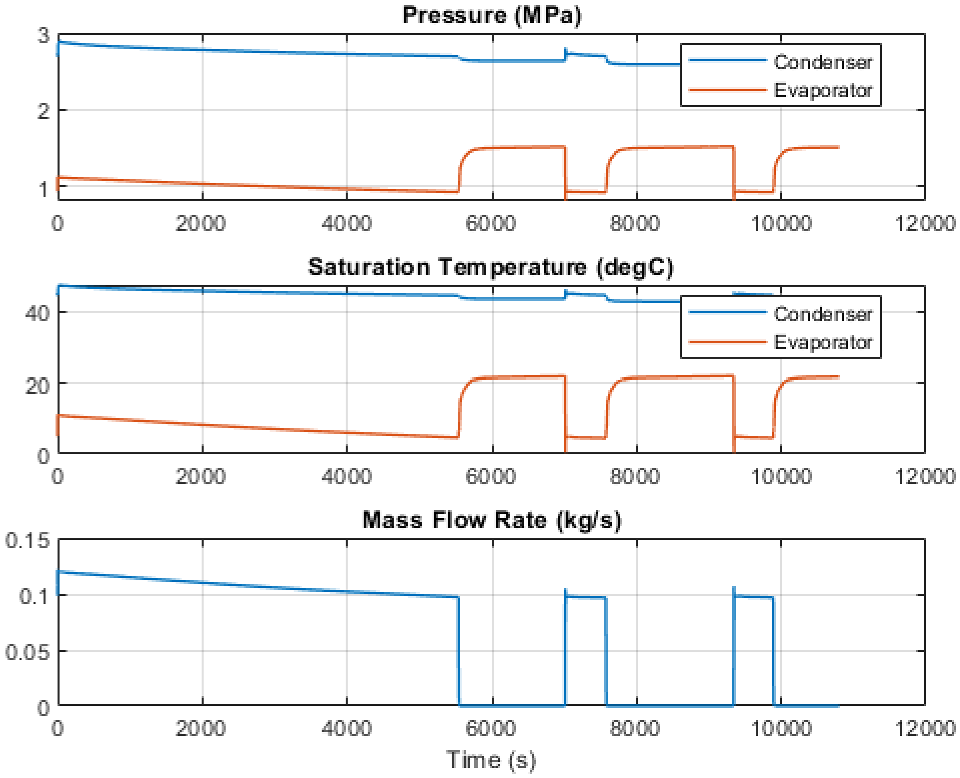


Simulation Results from Simscape Logging

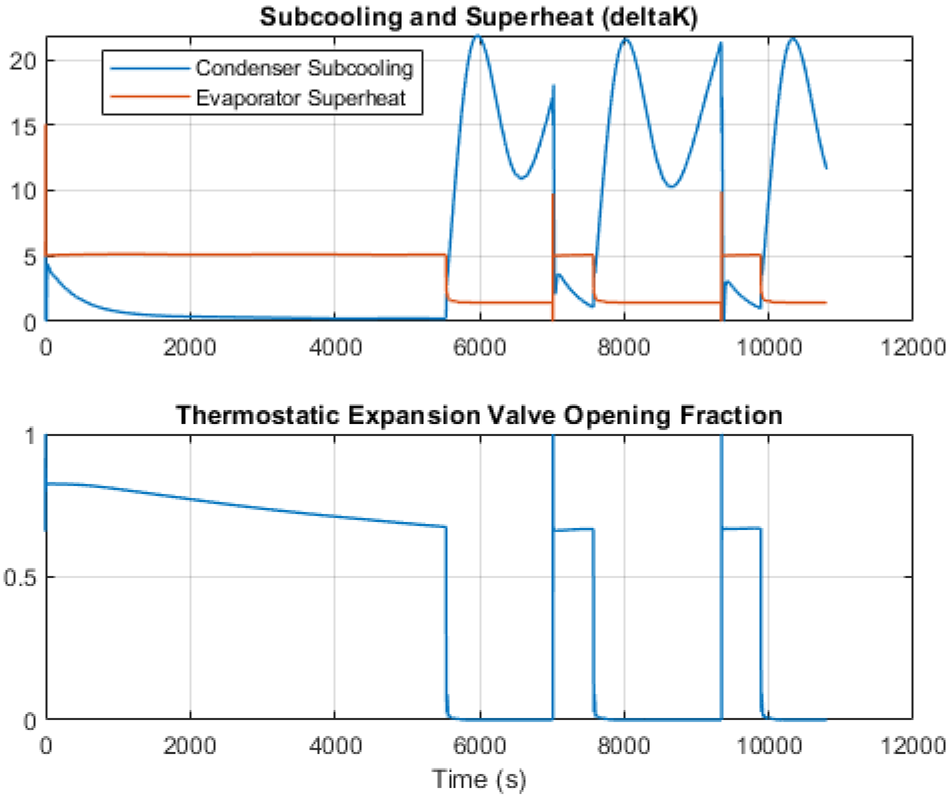
This plot shows the heat transfer in the condenser and the evaporator as well as the power consumed by the compressor and fans. The coefficient of performance is the ratio of the evaporator heat transfer to the total power consumed.



This plot shows the high and low pressure in the refrigeration cycle and the corresponding saturation temperatures. The design condensing temperature is 45 degC and the design evaporating temperature is 5 degC. The design refrigerant mass flow rate is around 0.1 kg/s.



This plot shows the subcooling at the condenser outlet and the superheat at the evaporator outlet. The thermostatic expansion valve meters the flow into the evaporator in order to maintain a superheat of 5 degC.


Subcooling and Superheat (deltaK)
Condenser Subcooling
Evaporator Superheat
0
5
10
15
20
0
2000
4000
6000
8000
10000
12000
Thermostatic Expansion Valve Opening Fraction
0
0.5
1
0
2000
4000
6000
8000
10000
12000
Time (s)

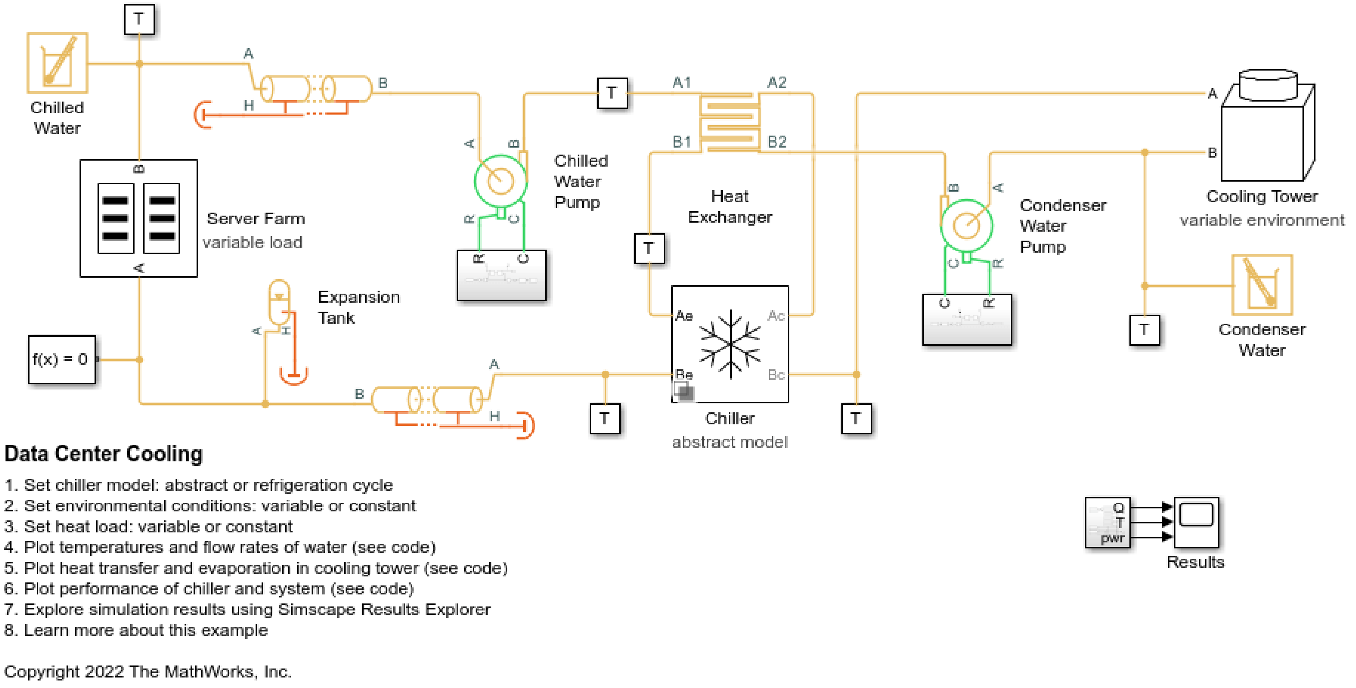
Data Center Cooling

This example models the cooling system of a data center. The system consists of two separate water loops: a chilled water loop and a condenser water loop. The chilled water loop absorbs heat from the server farm and transfers it to the condenser water loop. The condenser water loop rejects this heat to the environment using a cooling tower.

The heat transfer between the chilled water loop and condenser water loop occurs directly via the water-water heat exchanger and indirectly via the chiller. The heat exchanger is inexpensive to operate as it does not consume power. However, it must be supplemented by the chiller to provide sufficient cooling to the chilled water loop. The chiller is a refrigeration cycle that consumes power to transfer heat from the evaporator to the condenser via the R-134a refrigerant. It is the largest contributor to the energy consumption of the overall system.

The model includes several variants. The chiller can be modeled directly using the System-Level Refrigeration Cycle (2P) block or abstractly based on pipe heat transfer. Pipe heat transfer is simpler and allows for more efficient simulation. The environmental conditions and the server farm heat load can be variable or constant. Variable conditions and load provide a more realistic scenario. Constant conditions and load allow for simpler component sizing and analyses.

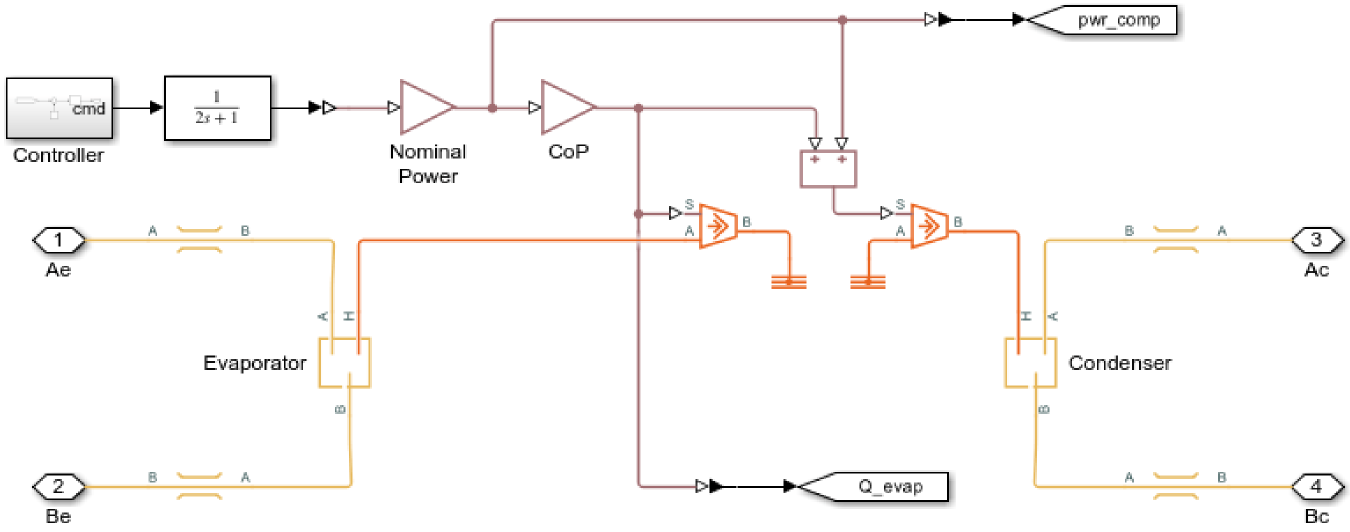
Model



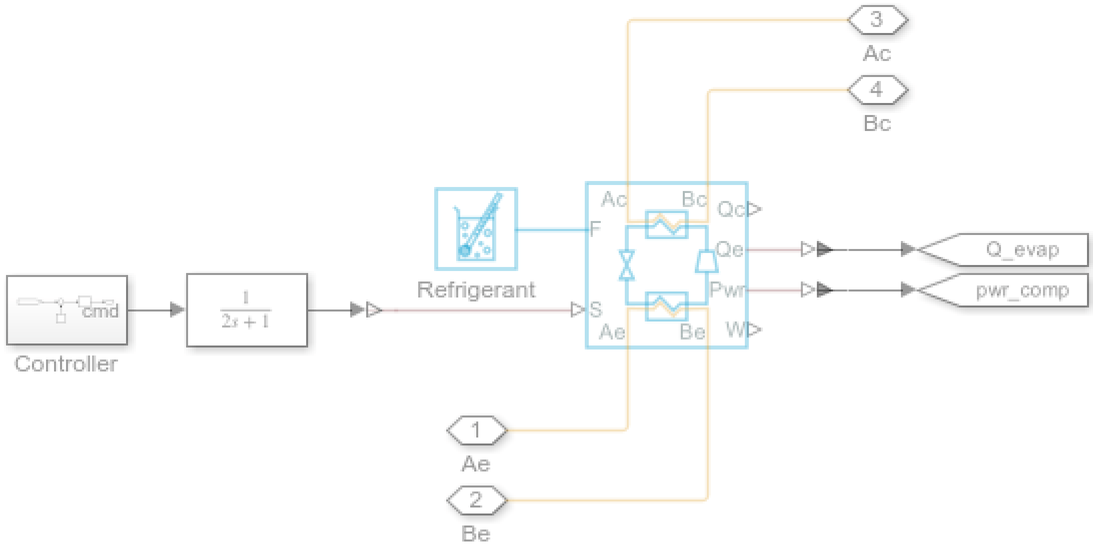
Data Center Cooling

1. Set chiller model: abstract or refrigeration cycle
2. Set environmental conditions: variable or constant
3. Set heat load: variable or constant
4. Plot temperatures and flow rates of water (see code)
5. Plot heat transfer and evaporation in cooling tower (see code)
6. Plot performance of chiller and system (see code)
7. Explore simulation results using Simscape Results Explorer
8. Learn more about this example

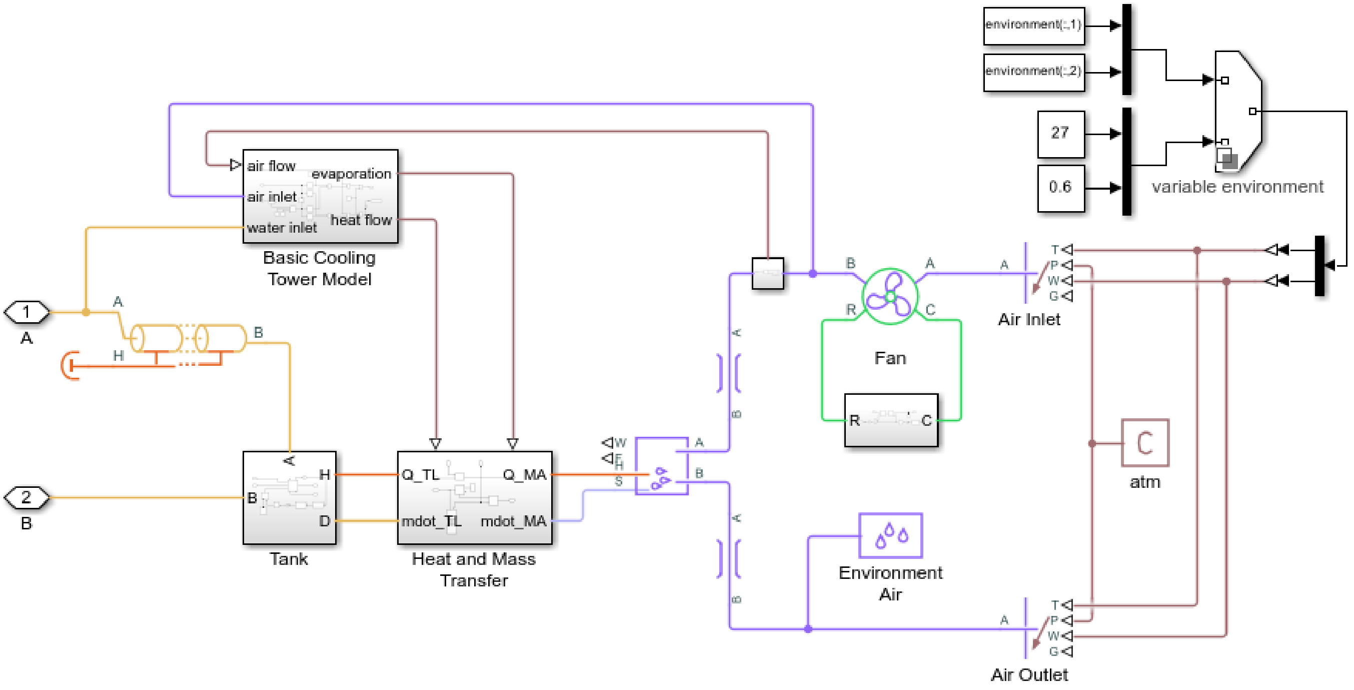
Chiller - Abstract Model Subsystem



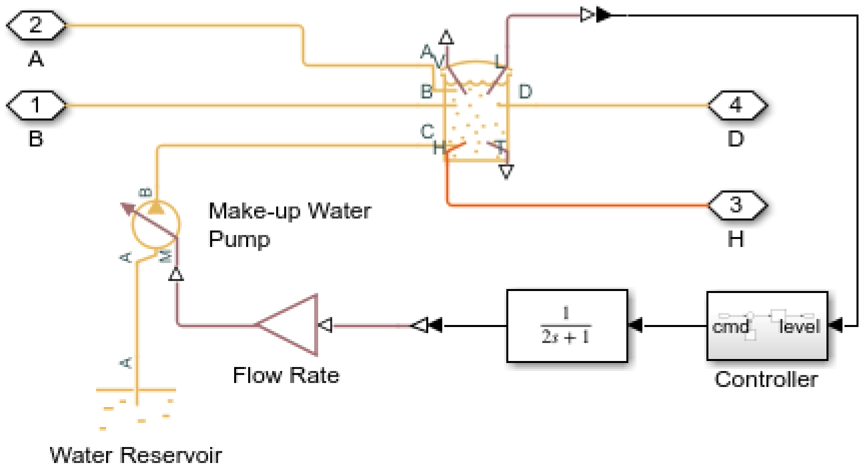
Chiller - Refrigeration Cycle Subsystem



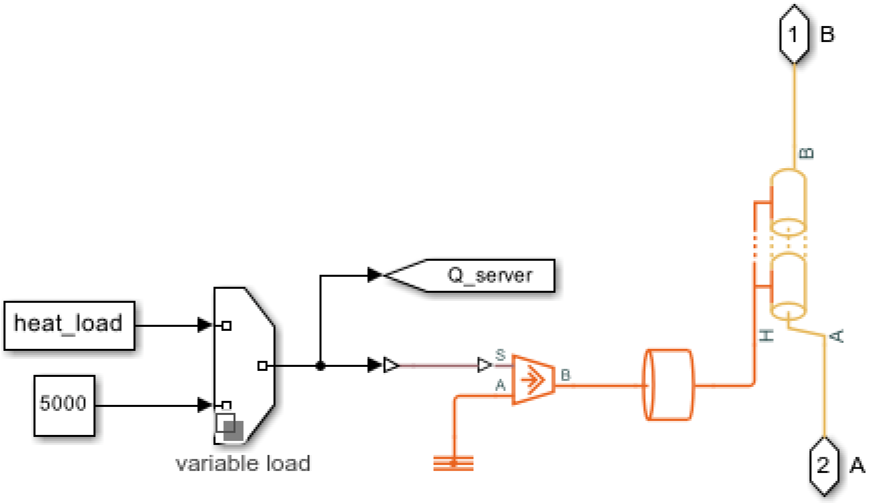
Cooling Tower Subsystem



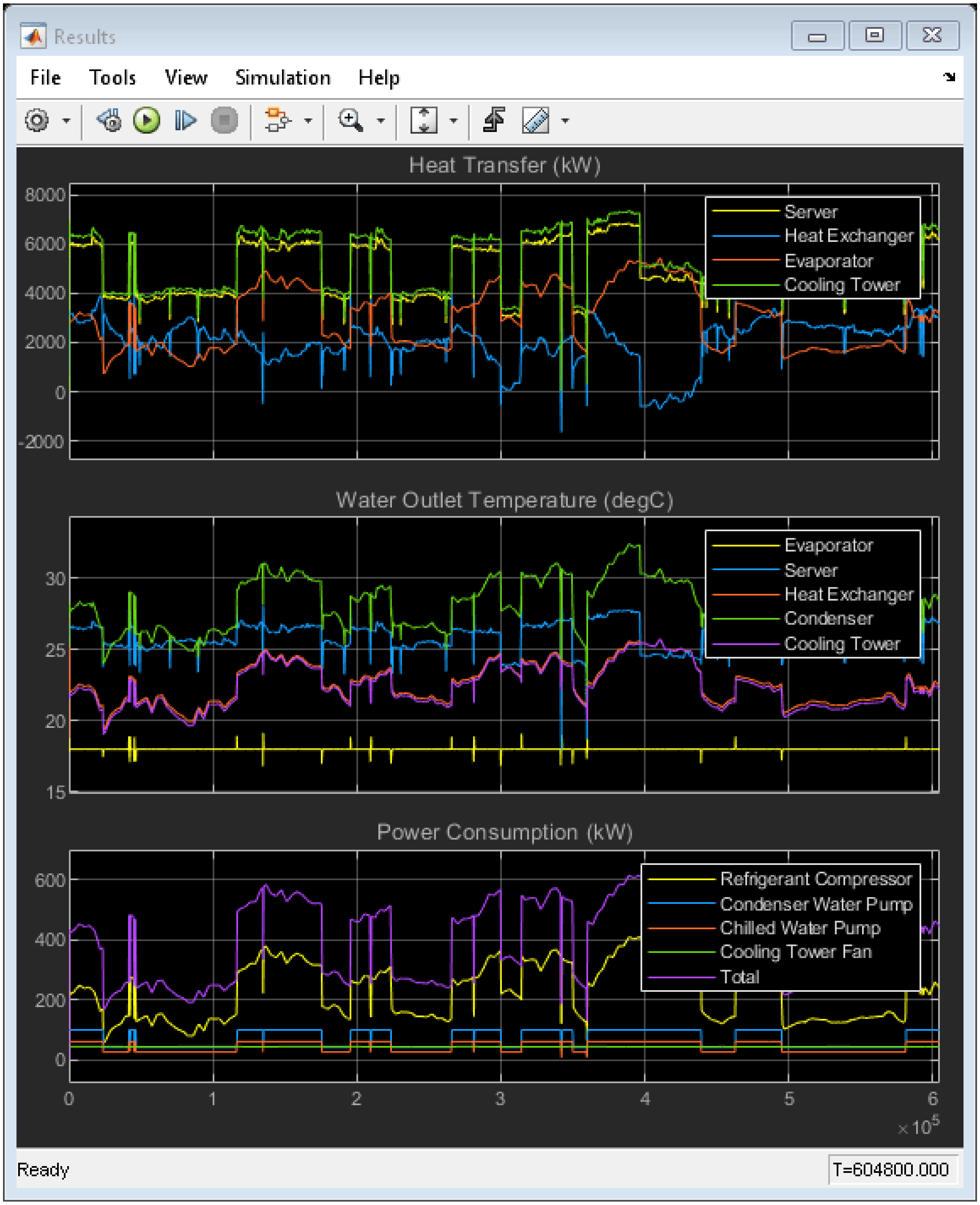
Tank Subsystem



Server Farm Subsystem



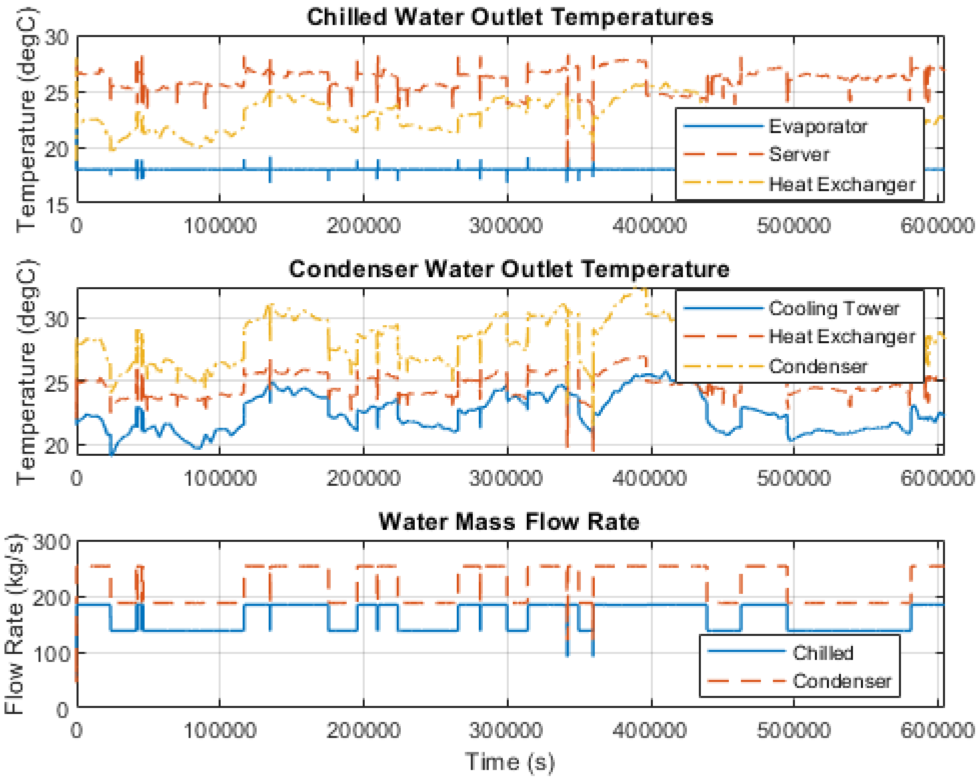
Simulation Results from Scopes



Simulation Results from Simscape Logging

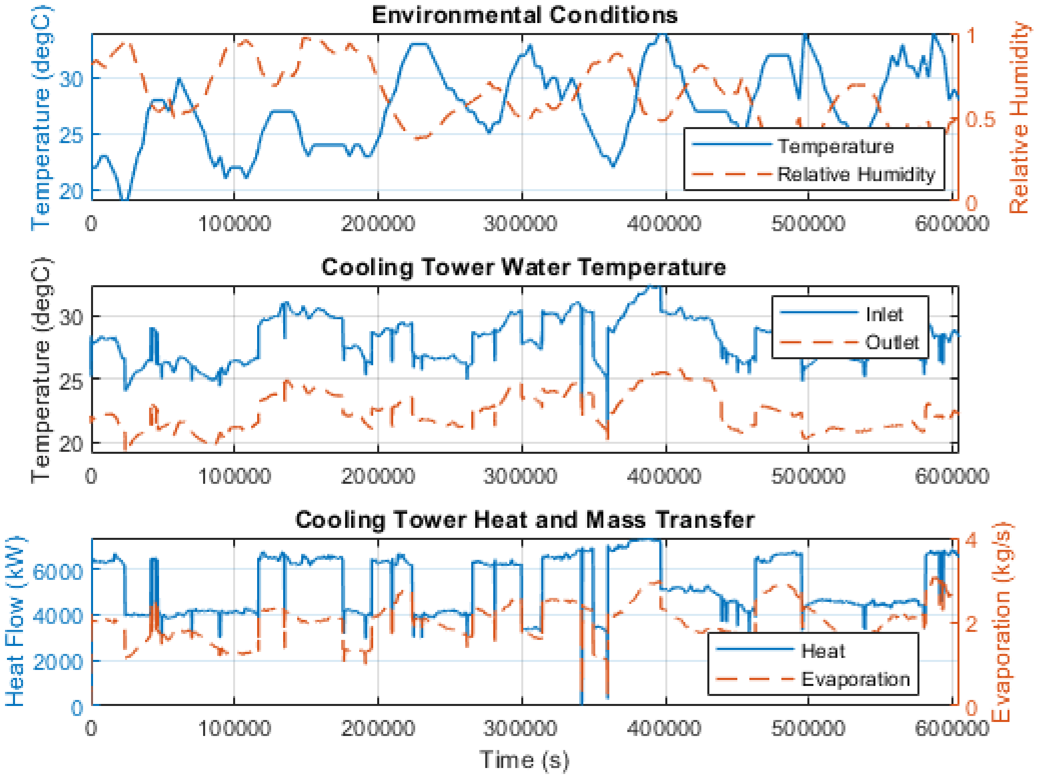
This plot shows the mass flow rates and temperatures of the chilled water loop and the condenser water loop. In the chilled water loop, the chiller's evaporator maintains a temperature of 18 degC. The chilled water absorbs heat from the server farm, transfers part of the heat to the condenser water via the heat exchanger, and transfers the remaining heat to the condenser water via the chiller. The chilled water pump adjusts the flow rate to keep the water temperature at the server outlet at around 28 degC.

In the condenser water loop, the cooling tower cools the water as much as possible based on environmental conditions. The condenser water absorbs heat from the chilled water via the heat exchanger and the chiller, then rejects all heat to the environment via the cooling tower. The condenser water pump adjusts the flow rate to keep the water temperature difference across the heat exchanger and condenser to around 6 degC.

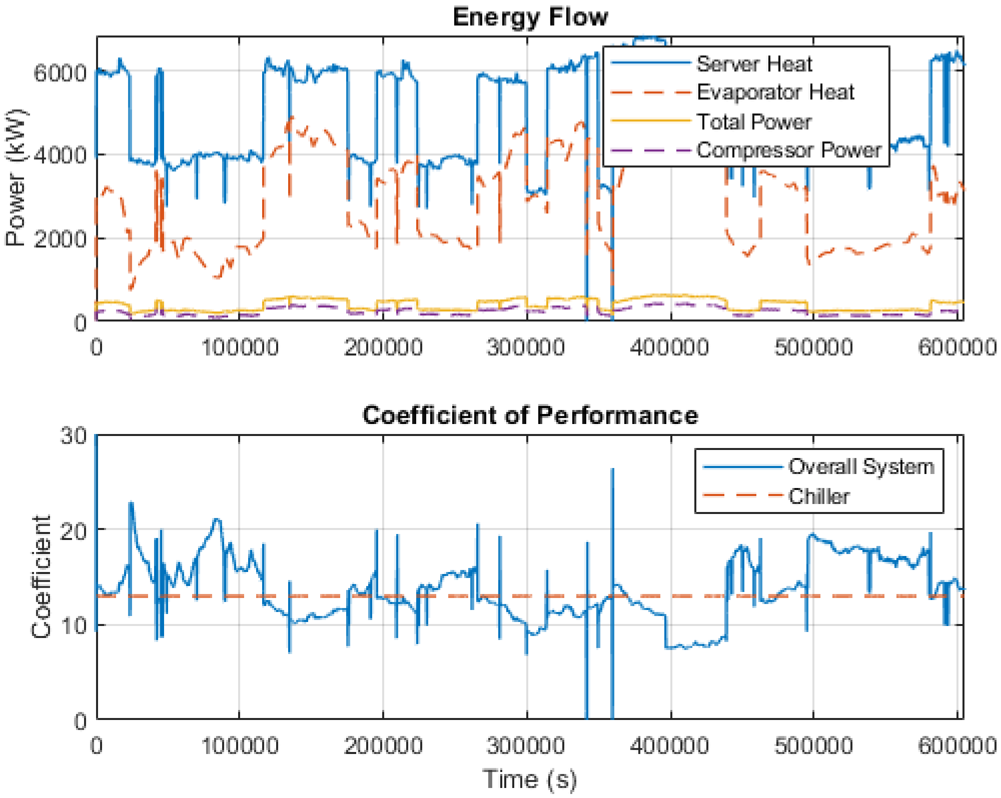


This plot shows the evaporative cooling performance of the cooling tower. The cooling tower removes heat from the condenser water by allowing some of the water to evaporate into the environment. The heat of vaporization needed for evaporation comes from the water and thus it cools the water. The fan draws in fresh air from the environment and pushes out the hot humid air. A colder and dryer environment increases the performance of the cooling tower. Unlike a regular water-air heat exchanger, the cooling tower can reduce the water temperature to below the environment temperature. The drawback is the loss of around 1% of the water flow, which must be replenished.

The evaporative process is modeled approximately using the e-NTU method based based on an analogy between heat and mass transfer. For sensible heat transfer, the e-NTU method obtains the heat flow rate by multiplying the theoretical maximum temperature difference with an effectiveness factor. For evaporative heat transfer, the analogous e-NTU method multiplies the theoretical maximum mixture enthalpy difference with an effectiveness factor. The use of mixture enthalpy accounts for differences in both temperature and humidity. The maximum occurs when air is assumed to become fully saturated with the evaporated water.



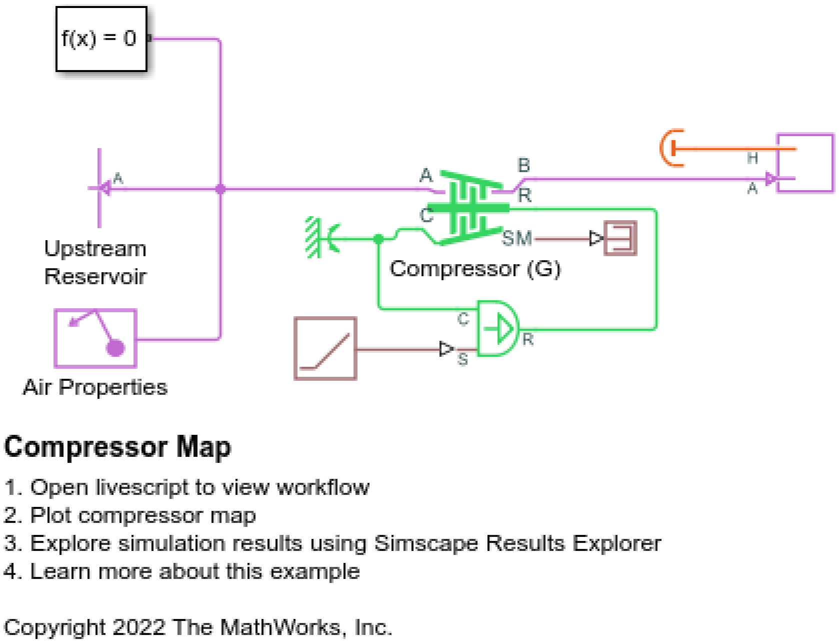
This plot shows the performance of the chiller and the overall system. The chiller's coefficient of performance (CoP) is the ratio of the heat transfer in the evaporator to the power consumed by the compressor. Applying the same concept to the entire system, the system's CoP is the ratio of the heat transfer in the server farm to the total power consumed by the compressor, fan, and all pumps. Both CoP values are around 13, which is higher than typical air conditioning units. This is because heat transfer between the refrigerant and water is more effective than heat transfer between the refrigerant and air. This allows the refrigeration cycle to have a smaller temperature difference and thus smaller pressure difference between the condenser and evaporator, which reduces the work required by the compressor.


Energy Flow
Power (kW)
6000
4000
2000
0
0
100000
200000
300000
400000
500000
600000
Server Heat
Evaporator Heat
Total Power
Compressor Power
Coefficient of Performance
Coefficient
30
20
10
0
Overall System
Chiller
Time (s)

Compressor Map

This example shows how to read and format compressor map data for use with the Compressor (G) block. Open the livescript to see the full workflow.

Model



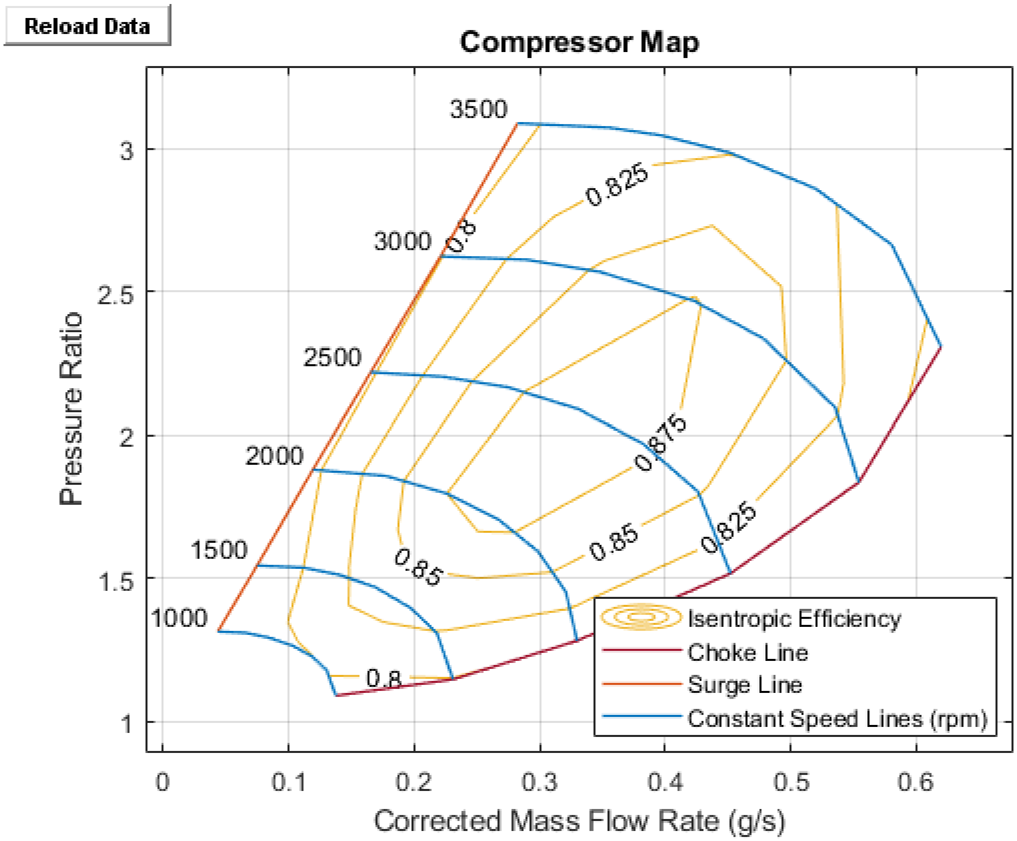
Compressor Map

1. Open livescript to view workflow
2. Plot compressor map
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2022 The MathWorks, Inc.

Workflow Results

This plot shows the compressor map resulting from the demonstrated workflow.

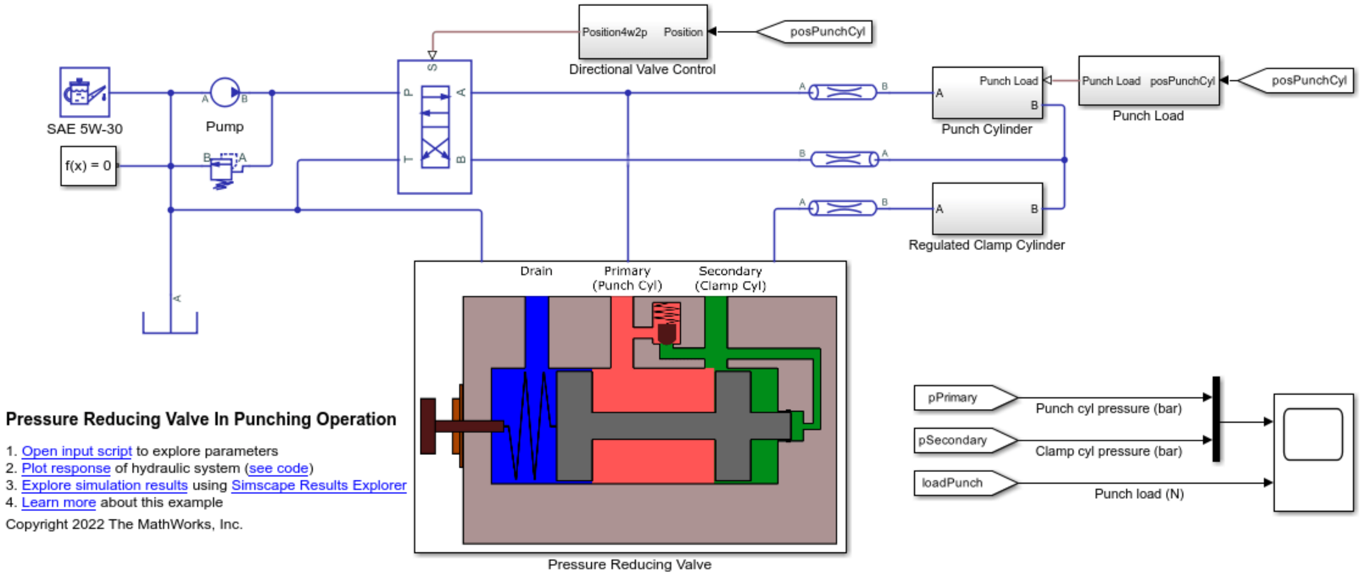

Reload Data
Compressor Map
3500
3000
2500
2000
1500
1000
0.825
0.8
0.875
0.85
0.85
0.825
0.8
Isentropic Efficiency
Choke Line
Surge Line
Constant Speed Lines (rpm)
Pressure Ratio
1
1.5
2
2.5
3
0
0.1
0.2
0.3
0.4
0.5
0.6
Corrected Mass Flow Rate (g/s)

Optimize Pressure Reducing Valve Model for Real-Time Simulation

This example shows how to identify simulation slowdowns, resolve performance issues, and configure a Simscape Fluids™ model for real-time simulation. This example uses the Pressure Reducing Valve In Punching Operation model, `sscfluids_prv_punch_operation` which is called here as the original model. In this example, you measure the simulation time, identify the factors that affect simulation speed, optimize the model, and prepare the model for code generation.

Open and Examine the Model

Open the original model.



Pressure Reducing Valve In Punching Operation

1. Open input script to explore parameters
2. Plot response of hydraulic system (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

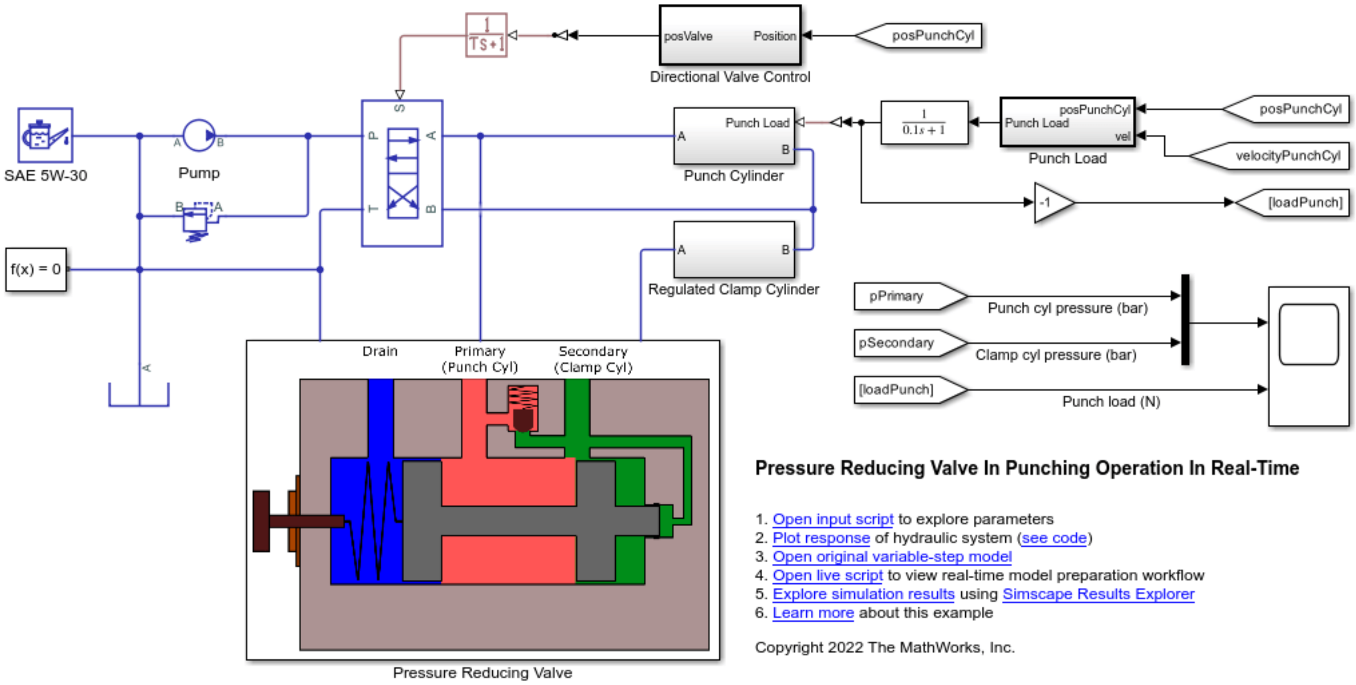
Copyright 2022 The MathWorks, Inc.

The model has been configured for real-time simulation by performing these steps:

1. Enable the **Use local solver** option in the Solver Configuration block. Set **Solver type** to `Backward Euler` and **Sample time** to `2e-4`. Enable the **Use fixed-cost runtime consistency iterations** option in the Solver Configuration block. Expand the **Use fixed-cost runtime consistency iterations** option and set **Nonlinear iterations** to `3`.
2. Open the Configuration Parameters window, click **Solver**, under **Solver selection**, set **Type** to `Fixed-step`, set **Solver** to `ode1be (Backward Euler)`, and under **Solver details**, set **Fixed-step size** to `2e-4`.
3. Replace Single-Acting Actuator (IL) blocks with Foundation Library blocks such as the Translational Mechanical Converter (IL), Translational Spring, Translational Damper and Translational Hard Stop blocks. Set **Hard stop model** in Translational Hard Stop block to `Full stiffness and damping applied at bounds, damped rebound`.
4. Convert the `Punch Load` and `Directional Valve Control` subsystems to Atomic Subsystems. Right click on the subsystems, then click **Block Parameters (subsystem)**. Enable the **Treat as atomic unit** option and set the **Sample time** to `1e-3`. Add Rate Transition blocks at the input and output of the subsystems.

5. Set the **Smoothing factor** parameter of the 4-Way 2-Position Directional Valve and the Check Valve blocks to `0.1`.
6. Remove the Pipe (IL), Pipe (IL)1 and Pipe (IL)2 blocks. To compensate for the pipe volume, increase the dead volume of the single-acting actuators that represent the punch cylinder, clamp cylinder and pressure reducing valve spool. Navigate inside `Punch Cylinder` subsystem. Increase the value of **Dead volume in chamber A** and **Dead volume in chamber B** parameters of the `Punch Cylinder` block from `1e-6 m^3` to `1e-3 m^3`. Navigate inside `Regulated Clamp Cylinder` subsystem. Increase the value of **Dead volume in chamber A** and **Dead volume in chamber B** parameters of the `Clamp Cylinder` block from `1e-6 m^3` to `1e-3 m^3`. Navigate inside `Pressure Reducing Valve` subsystem. Increase the value of **Dead volume** parameter of the Single-Acting Actuator (IL) and Single-Acting Actuator (IL)1 blocks from `1e-5 m^3` to `1e-4 m^3`.

Open the real-time model.



Pressure Reducing Valve In Punching Operation In Real-Time

1. Open input script to explore parameters
2. Plot response of hydraulic system (see code)
3. Open original variable-step model
4. Open live script to view real-time model preparation workflow
5. Explore simulation results using Simscape Results Explorer
6. Learn more about this example

Copyright 2022 The MathWorks, Inc.

Now, again open and examine the `sscfluids_prv_punch_operation` model.

This model includes interactions between Simscape™, Simulink® and Stateflow®. The Simscape blocks represent the physical model of the pressure-reducing valve and the punch system. The Simulink blocks are used to measure and transmit signals across the model. Stateflow models the conditional actuation of a punch cylinder that involves a feedback loop. The model includes several Goto, From, and PMC Port blocks that the subsystems use to interact with each other. This complex communication across the model causes simulation slowdown.

The Switch, Saturation, Hard Stop, Translational Friction blocks cause non-linear behavior that causes the solver to iterate using lower time steps. The 4-Way 2-Position Directional Valve causes changes in the flow direction which causes zero-crossings when the punch cylinder starts to displace or comes to a stop. The clamp cylinder and the pressure reducing valve spool cause similar behavior when they start to displace or come to a stop. These zero-crossings contribute to simulation slowdown.

The compressibility of the fluid is an essential aspect of modeling a hydraulic system. When the **Fluid dynamic compressibility** parameter of the Pipe block is set to `On`, an imbalance of mass inflow and mass outflow can cause liquid to accumulate or diminish in the pipe. As a result, pressure in the pipe volume can rise and fall dynamically, which provides some compliance to the system and modulates rapid pressure changes. This setting can contribute to simulation slowdown.

All the above factors can thus impede a real-time simulation. To achieve real-time simulation, the run time or execution time must be less than the Simulation stop time. To achieve real-time simulation for this model, you must use a fixed-step solver with a large enough time step.

Use a Variable-Step Solver to Get Reference Results

A variable-step solver adjusts the step size to capture events and dynamics in the system. Since, the variable step sizes cannot be mapped to the real-time clock of a target system, you must use an implicit or explicit fixed-step solver for real-time simulation.

To ensure that the results obtained with the fixed-step solver are accurate, compare them with reference results obtained by simulating the system with a variable-step solver.

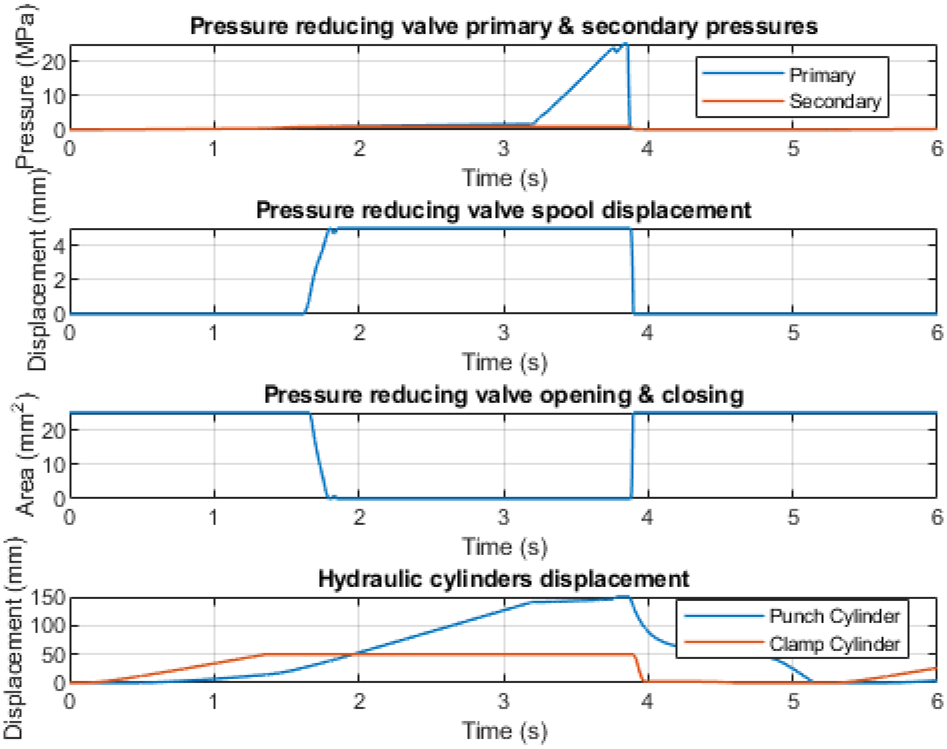
To run your model using a variable-step solver, ensure that the **Use local solver** in the Solver Configuration block is set to `off`. This example focuses on reducing the elapsed simulation time for one cycle of punching operation. Set **Simulation stop time (sec)** to `6`.

Before running the model, open the Configuration Parameters window, click **Data Import/Export**, then enable the **Time** parameter under **Save to workspace or file** tab. Enable the **Single simulation output** parameter to return simulation data as a single `Simulink.SimulationOutput` object. The object name is specified as `out` by default. Alternatively, enable the parameter by entering:

Create a `Simulink.SimulationInput` object with **Simulation stop time (sec)** as an input argument.

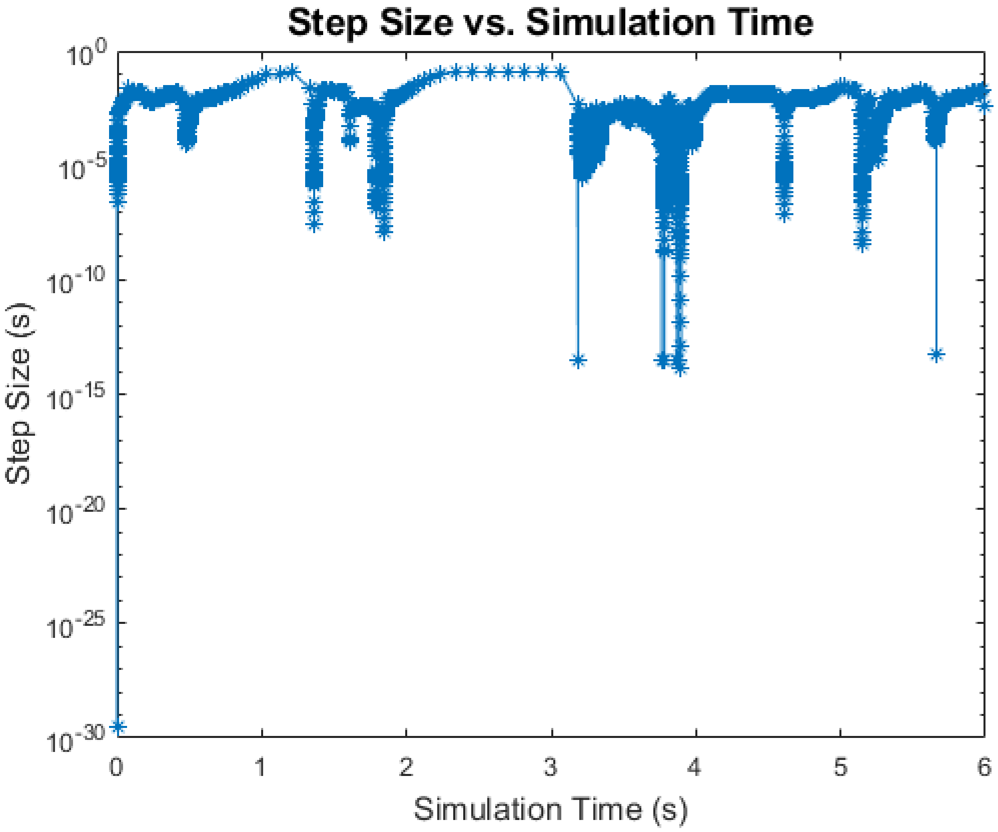
Specify a `Simulink.SimulationOutput` object. Run the model.

Plot the results.



Examine Step Sizes during the Simulation

A variable-step solver will vary the step size to stay within the error tolerances and to react to zero-crossing events . If the solver abruptly reduces the step size to a small value (e.g., 1e-15s), this indicates that the solver is trying to accurately identify a zero-crossing event. A fixed-step solver may have trouble capturing these events at a step size large enough to permit real-time simulation. These MATLAB® commands can be used to generate a plot that shows how the time step varies during the simulation:



This plot shows the step sizes the solver takes during simulation. The sharp plunge in step size are examples of zero-crossing events.

This analysis should provide a rough idea of a step size that can be used to run the simulation. Determining what effects are causing these events and modifying or eliminating them will make it easier to run the system with a fixed-step solver at a larger step size and produce results comparable to the variable-step simulation.

A stiff system has continuous dynamics that vary slowly and quickly. Implicit solvers are particularly useful for stiff problems. Using an explicit solver to solve a stiff system can lead to incorrect results. If a nonstiff solver uses a very small step size to solve a model, this is a sign that your system is stiff. This plot shows that the variable-step solver takes very less step sizes and thus the system is stiff.

Use a Fixed-Step Solver to Measure Simulation Time

Run your model using fixed-step, fixed-cost configurations. It is recommended to enable the **Use local solver** option in the Solver Configuration block. Local solvers discretize the continuous states. Using local solvers permits configuring implicit solvers on the stiff portions of the model and explicit solvers on the remainder of the model, minimizing execution time while maintaining accuracy. This example uses the Backward Euler solver. Set **Solver type** to `Backward Euler` and **Sample time** to `1e-4`. Enable the **Use fixed-cost runtime consistency iterations** option in the Solver Configuration block. Expand the **Use fixed-cost runtime consistency iterations** option and set **Nonlinear iterations** to 3.

Run the model.

View the metadata and the timing information.

To see the elapsed simulation time, in seconds, enter:

The simulation time depends on the CPU on which you execute the model. There may be a difference between the time measured by you and the time in this example.

Compare Solvers and Determine if the Simulation runs Real-Time

Next, determine the fixed-step solver most appropriate for your model. To do this, use the local solvers in the Solver Configuration block and compare the elapsed simulation time for each. The available solvers are:

- Backward Euler
- Trapezoidal Rule
- Partitioning

The Backward Euler tends to damp out oscillations, but is more stable, especially if you increase the time step.

The Trapezoidal Rule solver captures oscillations better but is less stable.

The Partitioning solver lets you increase real-time simulation speed by partitioning the entire system of equations that correspond to a Simscape network into a cascade of smaller equation systems. Not all networks can be partitioned. However, when a system can be partitioned, this solver provides a significant increase in real-time simulation speed. To configure the Partitioning solver, set **Solver type** to `Partitioning` and **Sample time** to `1e-4`. Set **Partition method** to `Robust simulation` and **Partition storage method** to `Exhaustive`. Enable the **Use fixed-cost runtime consistency iterations** option in the Solver Configuration block. Expand the **Use fixed-cost runtime consistency iterations** option and set **Nonlinear iterations** to `3`.

This system cannot be partitioned with a time step of 1e-4 s or greater.

Next, set **Solver type** to `Backward Euler` and **Sample time** to `1e-4`. Enable the **Use fixed-cost runtime consistency iterations** option in the Solver Configuration block. Expand the **Use fixed-cost runtime consistency iterations** option and set **Nonlinear iterations** to `3`.

Run the model.

After running the model, access the metadata and the timing information.

View the elapsed simulation time, in seconds:

Then, set **Solver type** to `Trapezoidal Rule` and **Sample time** to `1e-4`. Retain the fixed-cost configurations.

Run the model.

After running the model, access the metadata and the timing information.

View the elapsed simulation time, in seconds:

Because the elapsed simulation time is more than the **Simulation stop time**, the model does not run in real-time.

Identify Factors that Affect Speed and Choose an Appropriate Solution

Because the simulation is not real-time capable, you can explore some options.

- **Determine New Settings that Reduce the Execution Time**

You can either permit a larger step size or reduce the number of nonlinear iterations.

Again, set **Solver type** to `Backward Euler`. Now, set **Sample time** to `2e-4`. Enable the **Use fixed-cost runtime consistency iterations** option in the Solver Configuration block. Expand the **Use fixed-cost runtime consistency iterations** option and set **Nonlinear iterations** to 3.

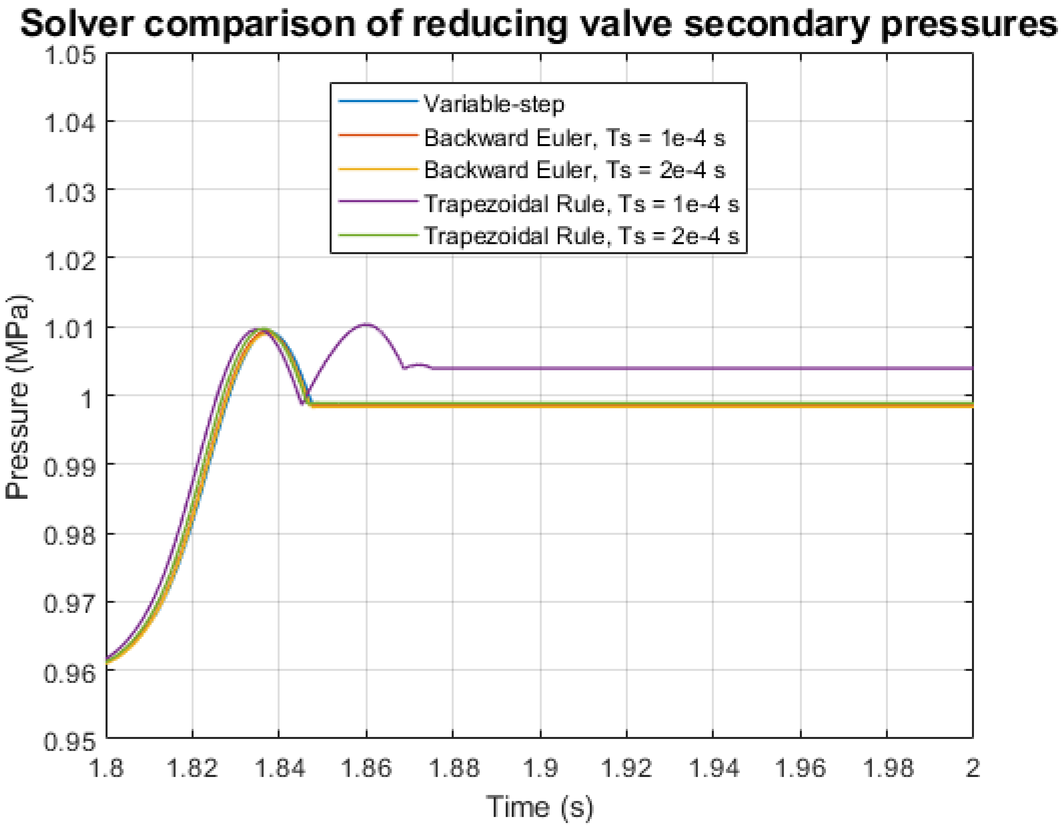
The global solver solves for the Simulink and Stateflow blocks. To ensure that the Simulink and Stateflow blocks also run at higher time steps without affecting accuracy, open the Configuration Parameters window, click **Solver**, under **Solver selection**, set **Type** to `Fixed-step`, set **Solver** to `ode1be (Backward Euler)`, and under **Solver details**, set **Fixed-step size** to `2e-4`.

Run the model.

Then, set **Solver type** to `Trapezoidal Rule` and **Sample time** to `2e-4`. Retain the fixed-cost configurations.

Run the model.

Now, compare the results of simulation with large step size with the reference results obtained from variable-step solver.



This zoomed in version of the figure shows the comparison of solvers for secondary pressure. Increasing the step size, increases the error, but the error is within acceptable limits for Backward Euler solver with sample time of 2e-4 s.

- **Use the Solver Profiler to Identify Factors that Affect Speed**

The system stiffness is high. High stiffness is responsible for simulation slowdown.

The **Simscape Stiffness** option in the **Solver Profiler** tool lets you analyze the Simscape networks and determine which variables and equations have the most impact on system stiffness. You can then modify the values of parameters involved in these equations to make the system less stiff.

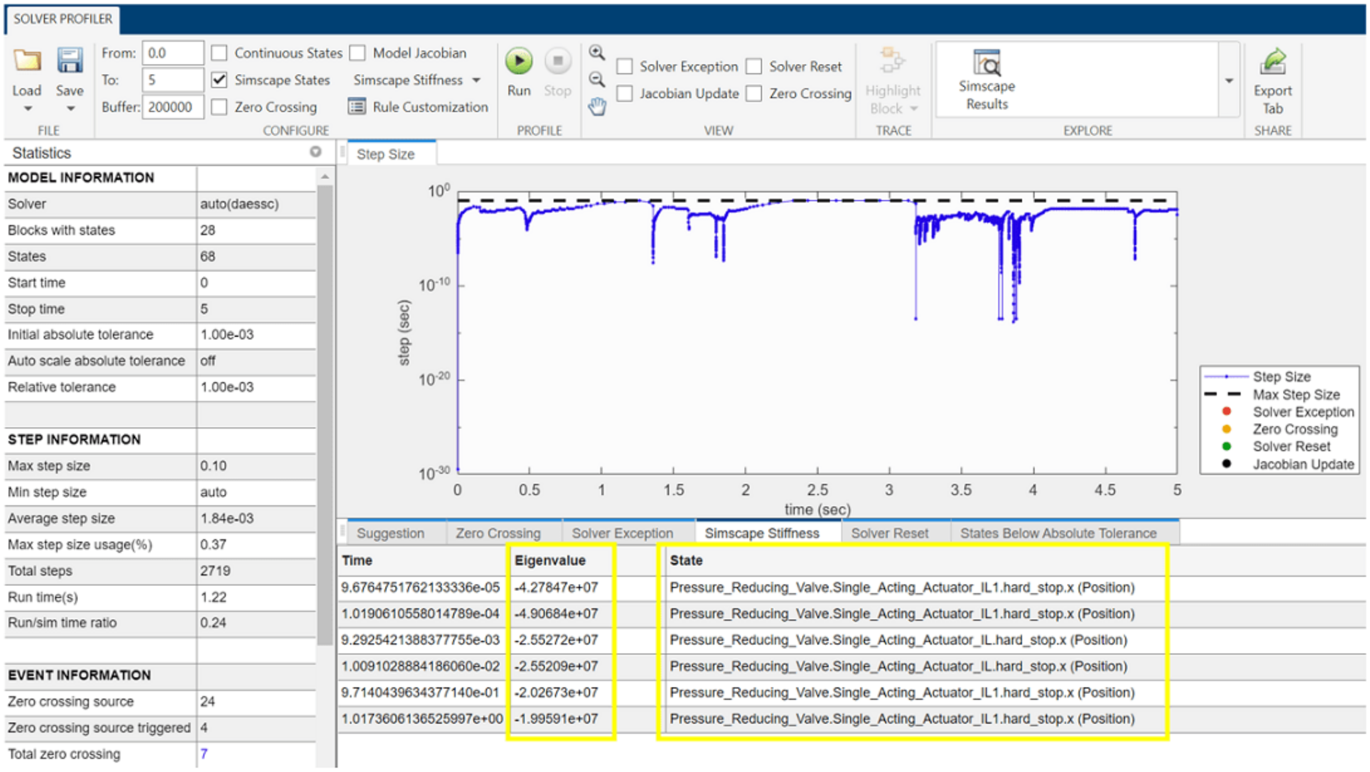
Ensure that you select a Variable-step Global Solver for this analysis. Open the Configuration Parameters window, click **Solver**, under **Solver selection**, set **Type** to `Variable-step`. Ensure that **Use local solver** in the Solver Configuration block is set to `off`.

To access the **Simscape Stiffness** tool, on the **Debug** tab, under **Performance Advisor**, click **Solver Profiler**. To perform stiffness analysis on the Simscape model, in the Solver Profiler window, click the **Simscape Stiffness** drop-down button, enter the stiffness analysis checkpoints as a vector, and click Run.

This example uses the vector `[1e-4, 1e-2, 1, 3.76]`.

The **Simscape Stiffness** tab in the bottom pane shows that the variable with the most impact on the system stiffness is Pressure_Reducing_Valve.Single_Acting_Actuator_IL1.hard_stop.x(Position), and the corresponding eigenvalue is of the order of -1e+07 at each interval. When the time steps do not line up with the times in your vector, the tool performs stiffness analysis at the closest time steps before and after the instances you specify, t- and t+, respectively.

The greater the magnitude of a negative eigenvalue, the more unstable the system becomes due to stiffness.



From: 0.0
To: 5
Buffer: 200000

□ Continuous States
☑ Simscape States
□ Zero Crossing
□ Model Jacobian
Simscape Stiffness
Rule Customization

□ Solver Exception
□ Jacobian Update
□ Solver Reset
□ Zero Crossing

Statistics

MODEL INFORMATION	
Solver	auto(daessc)
Blocks with states	28
States	68
Start time	0
Stop time	5
Initial absolute tolerance	1.00e-03
Auto scale absolute tolerance	off
Relative tolerance	1.00e-03
STEP INFORMATION	
Max step size	0.10
Min step size	auto
Average step size	1.84e-03
Max step size usage(%)	0.37
Total steps	2719
Run time(s)	1.22
Run/sim time ratio	0.24
EVENT INFORMATION	
Zero crossing source	24
Zero crossing source triggered	4
Total zero crossing	7

Time	Eigenvalue	State
9.6764751762133336e-05	-4.27847e+07	Pressure_Reducing_Valve.Single_Acting_Actuator_IL1.hard_stop.x (Position)
1.0190610558014789e-04	-4.90684e+07	Pressure_Reducing_Valve.Single_Acting_Actuator_IL1.hard_stop.x (Position)
9.2925421388377755e-03	-2.55272e+07	Pressure_Reducing_Valve.Single_Acting_Actuator_IL.hard_stop.x (Position)
1.0091028884186060e-02	-2.55209e+07	Pressure_Reducing_Valve.Single_Acting_Actuator_IL.hard_stop.x (Position)
9.7140439634377140e-01	-2.02673e+07	Pressure_Reducing_Valve.Single_Acting_Actuator_IL1.hard_stop.x (Position)
1.0173606136525997e+00	-1.99591e+07	Pressure_Reducing_Valve.Single_Acting_Actuator_IL1.hard_stop.x (Position)

The **Solver Profiler** tool shows that the eigenvalues of the states involving hard stop forces of the Single-Acting Actuator (IL) blocks are high, which is an indication that the hard stop is contributing to a stiffer model. You can implement the next steps to reduce numerical stiffness.

The hard stop represents the interaction of the rightmost piston face of the reducing valve spool with the physical obstruction of the pilot line. Replace Single-Acting Actuator (IL) blocks with Foundation Library blocks such as the Translational Mechanical Converter (IL), Translational Spring, Translational Damper and Translational Hard Stop blocks. This eliminates one hard stop. Using only one hard stop instead of two is sufficient to represent the behavior and reduces non-linearity due to hard stop behavior.

The Translational Hard Stop block has the **Hard stop model** parameter set to `Stiffness and damping applied smoothly through transition region, damped rebound` setting by default. As a result, the block implements smoothing near zero, when the reducing valve spool starts to move. This smoothing introduces a nonlinearity that the solver takes time to resolve. Set **Hard stop model** to `Full stiffness and damping applied at bounds, damped rebound`.

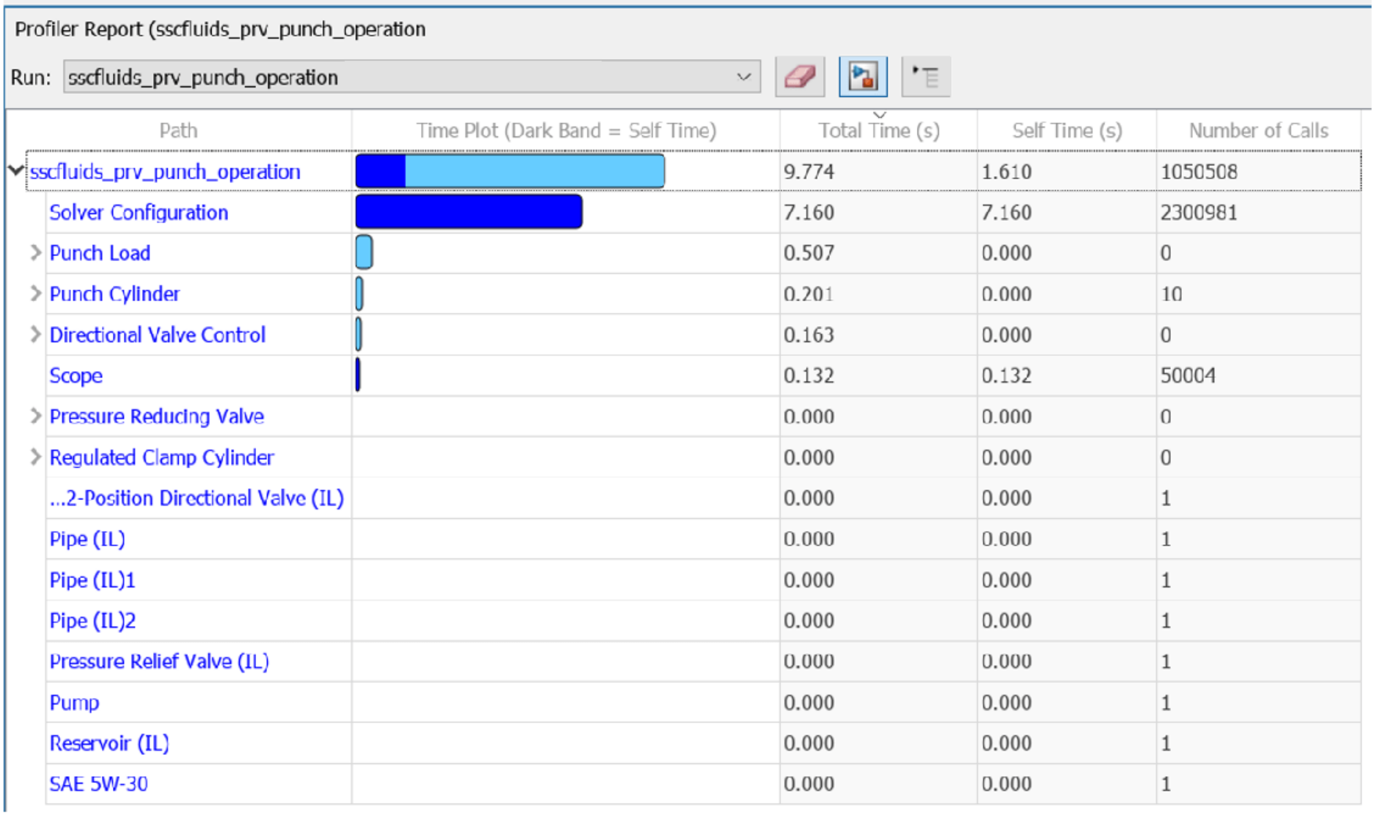
Set the **Use local solver** option in the Solver Configuration block to `On`. Run the model again with new blocks and new hard stop model.

- **Use the Simulink Profiler to Identify Factors that Affect Speed**

The **Simulink Profiler** identifies how much time each Simulink block and all Simscape blocks take to simulate. The **Simulink Profiler** can be used to identify simulation slowdowns and manually resolve performance bottlenecks.

To open the **Simulink Profiler**, on the **Debug** tab, under **Performance Advisor**, click **Simulink Profiler**. On the **Profile** tab, click **Profile** to run the model and collect profiler data.

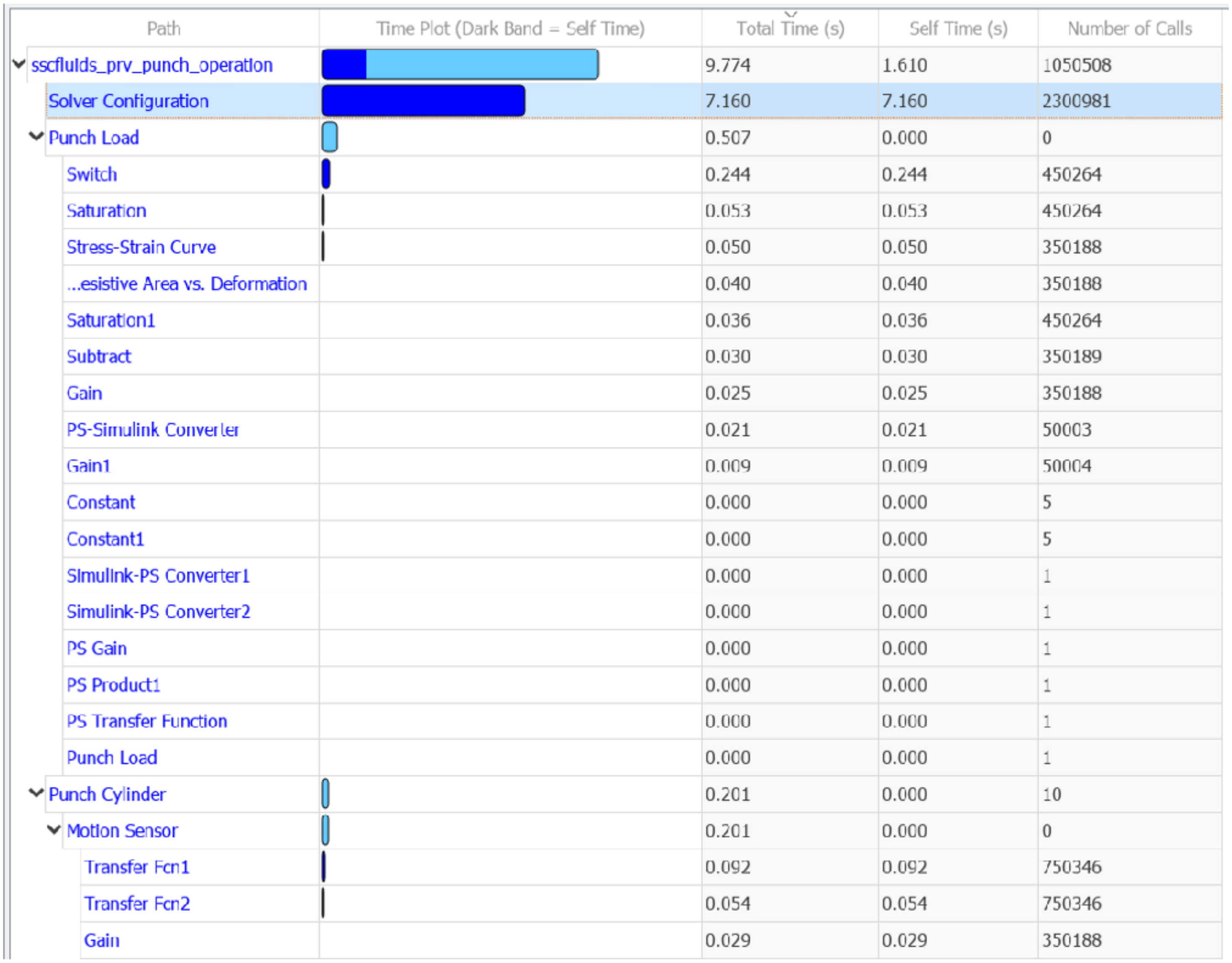
When simulation is complete, the **Profiler Report** pane displays the simulation profile for the model. Expand **sscfluids_prv_punch_operation**. By default, the profiler sorts the blocks at the same level of the model in descending order by the values in the **Total Time(s)** column. The contribution of the Simscape blocks is grouped in the **Solver Configuration** row. Although the Simscape blocks take up the most time (`8.82 s`) out of the total time (`12.326 s`), the time consumed by the Simulink blocks can be minimized. The `Punch Load` subsystem takes up the most time (`0.646 s`), followed by the `Punch Cylinder` subsystem (`0.242 s`) and the `Directional Valve Control` subsystem (`0.188 s`). Additionally, logging the results to the Scope block leads to a slowdown of `0.15 s`. Expand the subsystems hierarchy to see how much time each block takes up.



Profiler Report (sscfluids_prv_punch_operation

Run: sscfluids_prv_punch_operation

Path	Time Plot (Dark Band = Self Time)	Total Time (s)	Self Time (s)	Number of Calls
sscfluids_prv_punch_operation		9.774	1.610	1050508
Solver Configuration		7.160	7.160	2300981
Punch Load		0.507	0.000	0
Punch Cylinder		0.201	0.000	10
Directional Valve Control		0.163	0.000	0
Scope		0.132	0.132	50004
Pressure Reducing Valve		0.000	0.000	0
Regulated Clamp Cylinder		0.000	0.000	0
...2-Position Directional Valve (IL)		0.000	0.000	1
Pipe (IL)		0.000	0.000	1
Pipe (IL)1		0.000	0.000	1
Pipe (IL)2		0.000	0.000	1
Pressure Relief Valve (IL)		0.000	0.000	1
Pump		0.000	0.000	1
Reservoir (IL)		0.000	0.000	1
SAE 5W-30		0.000	0.000	1



Path	Time Plot (Dark Band = Self Time)	Total Time (s)	Self Time (s)	Number of Calls
sscfluids_prv_punch_operation		9.774	1.610	1050508
Solver Configuration		7.160	7.160	2300981
Punch Load		0.507	0.000	0
Switch		0.244	0.244	450264
Saturation		0.053	0.053	450264
Stress-Strain Curve		0.050	0.050	350188
...esistive Area vs. Deformation		0.040	0.040	350188
Saturation1		0.036	0.036	450264
Subtract		0.030	0.030	350189
Gain		0.025	0.025	350188
PS-Simulink Converter		0.021	0.021	50003
Gain1		0.009	0.009	50004
Constant		0.000	0.000	5
Constant1		0.000	0.000	5
Simulink-PS Converter1		0.000	0.000	1
Simulink-PS Converter2		0.000	0.000	1
PS Gain		0.000	0.000	1
PS Product1		0.000	0.000	1
PS Transfer Function		0.000	0.000	1
Punch Load		0.000	0.000	1
Punch Cylinder		0.201	0.000	10
Motion Sensor		0.201	0.000	0
Transfer Fcn1		0.092	0.092	750346
Transfer Fcn2		0.054	0.054	750346
Gain		0.029	0.029	350188

You can see that within `Punch Load` subsystem, the blocks such as Switch and Saturation lead to minor slowdown due to their non-linear behavior. The execution of these subsystems can be sped up by implementing these steps:

Convert the `Punch Load` and `Directional Valve Control` subsystems to Atomic Subsystems with a higher sample time. The subsystems named `Punch Load` and `Directional Valve Control` are virtual subsystems. Virtual subsystems are treated as if all the blocks existed at the same level. Atomic subsystems let you control the execution of the model. Right click on the subsystems, then click **Block Parameters (subsystem)**. Enable the **Treat as atomic unit** option and set the **Sample time** to `1e-3`, that is higher than the sample time of the local solver (1e-4) so that the blocks of the subsystem run faster.

Add Rate Transition blocks at the input and output of the subsystems. The Rate Transition block transfers data from the output of a block operating at one rate to the input of a block operating at a different rate. Use the block parameters to ensure data integrity and deterministic transfer for faster response or lower memory requirements.

Run the model again and check if the simulation time has improved.

You can also run the **Performance Advisor** to automatically fix performance-related issues in the model. To open the **Performance Advisor**, on the **Debug** tab, under **Performance Advisor**, click **Performance Advisor**. To learn more, see “Improve Simulation Performance Using Performance Advisor”.

Configure the Model

You can configure the model to experience further improvement in speed.

- **Modify or Remove Elements that Create Discontinuities**

A valve with zero smoothing factor causes discontinuities during near-open and near-closed positions. These discontinuities cause the solver to take smaller time steps and slows the simulation down. A non-zero smoothing factor implements a piecewise quadratic smoothing function that helps avoid these discontinuities during near-open and near-closed positions. A higher value of smoothing factor increases the effect of the smoothing function.

The 4-Way 2-Position Directional Valve and the Check Valve blocks have their **Smoothing factor** parameters set to `0.01`. Set the **Smoothing factor** parameter of these blocks to `0.1` to observe a slight improvement in simulation speed.

- **Modify or Remove Elements with Small Time Constant**

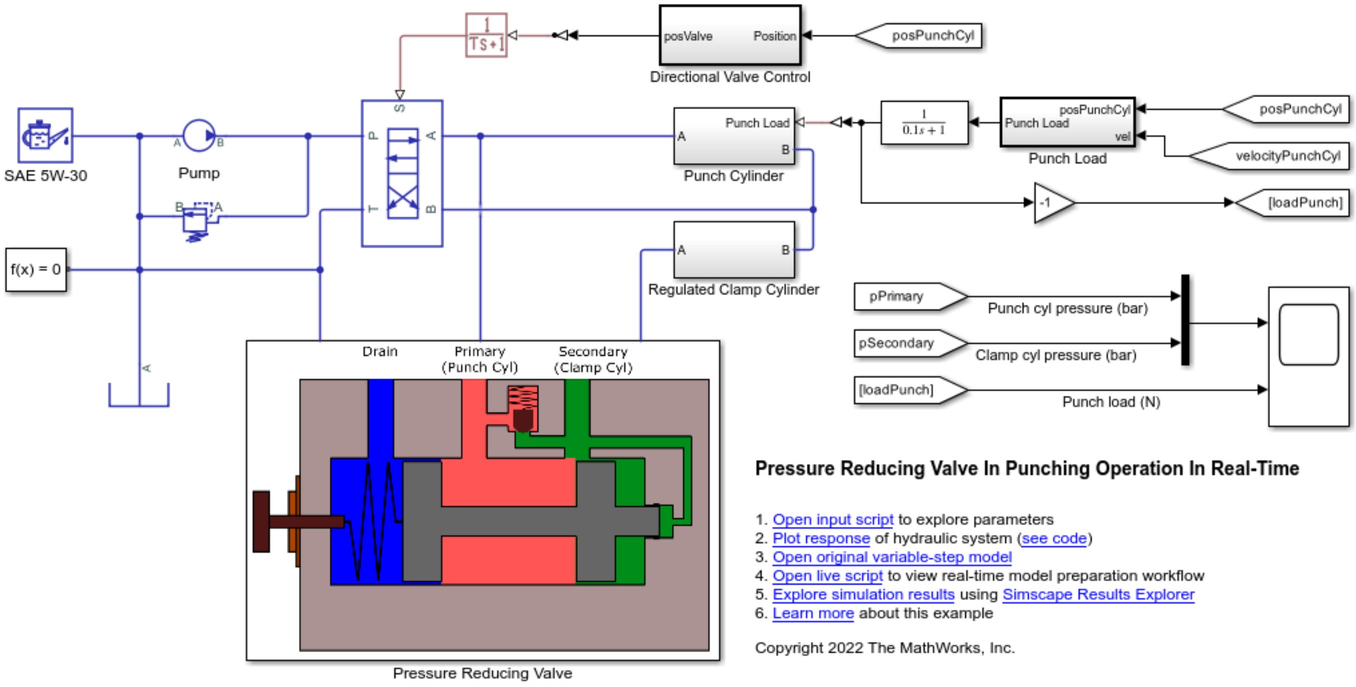
The Pipe (IL) blocks in the model have the **Fluid dynamic compressibility** parameter set to `On`. When dynamic compressibility is on, an imbalance of mass inflow and mass outflow can cause liquid to accumulate or diminish in the pipe. As a result, pressure in the pipe volume can rise and fall dynamically, which provides some compliance to the system and modulates rapid pressure changes. These pressure changes increase the model complexity and the simulation time. Also, the pipe friction can induce discontinuities.

Remove the Pipe (IL), Pipe (IL)1 and Pipe (IL)2 blocks. To compensate for the pipe volume, increase the dead volume of the single-acting actuators that represent the punch cylinder, clamp cylinder and pressure reducing valve spool. Navigate inside `Punch Cylinder` subsystem. Increase the value of **Dead volume in chamber A** and **Dead volume in chamber B** parameters of the `Punch Cylinder` block from `1e-6 m^3` to `1e-3 m^3`. Navigate inside `Regulated Clamp Cylinder` subsystem. Increase the value of **Dead volume in chamber A** and **Dead volume in chamber B** parameters of the `Clamp Cylinder` block from `1e-6 m^3` to `1e-3 m^3`. Navigate inside `Pressure Reducing Valve` subsystem. Increase the value of **Dead volume** parameter of the Single-Acting Actuator (IL) and Single-Acting Actuator (IL)1 blocks from `1e-5 m^3` to `1e-4 m^3`. These changes lead to an improvement in speed with no marked difference in the results.

Now you can close the original model.

Simulate the Real-Time Model

Open the sscfluids_prv_punch_operation_RT model, that is already optimized for real-time simulation.



Pressure Reducing Valve In Punching Operation In Real-Time

1. Open input script to explore parameters
2. Plot response of hydraulic system (see code)
3. Open original variable-step model
4. Open live script to view real-time model preparation workflow
5. Explore simulation results using Simscape Results Explorer
6. Learn more about this example

Copyright 2022 The MathWorks, Inc.

Open the Configuration Parameters window, click **Data Import/Export**, then enable the **Single simulation output** parameter to return simulation data as a single `Simulink.SimulationOutput` object and set the object name to `out_sscfluids_prv_punch_operation_RT`.

Set **Simulation stop time (sec)** to `6`.

Enable the **Use local solver** option in the Solver Configuration block. Set **Solver type** to `Backward Euler` and **Sample time** to `2e-4`. Enable the **Use fixed-cost runtime consistency iterations** option in the Solver Configuration block. Expand the **Use fixed-cost runtime consistency iterations** option and set **Nonlinear iterations** to `3`.

Open the Configuration Parameters window, click **Solver**, under **Solver selection**, set **Type** to `Fixed-step`, set **Solver** to `ode1be (Backward Euler)`, and under **Solver details**, set **Fixed-step size** to `2e-4`.

Create a `Simulink.SimulationInput` object with **Simulation stop time (sec)** as an input argument. Specify a `Simulink.SimulationOutput` object. Run the model optimized for real-time simulation.

After running the model, access the metadata and the timing information.

To view the elapsed simulation time, in seconds, enter:

The elapsed run time is less than the simulation time, that indicates the model runs faster than real-time.

Generate and Build Code

After you optimize the model for real-time simulation, you can generate code from the model and build the code.

- **Generate Code from the Model**

To generate code from the real-time model

1. In the **Modeling** tab, click **Model Settings**. The Configuration Parameters dialog opens. Navigate to the **Code Generation** tab, under **Build process**, select the **Generate code only** parameter, and click **Apply**.
2. Note that the system target file must be `grt.tlc`. In the Configuration Parameters window, click **Code Generation**. In the right pane, set **System Target File** to `grt.tlc`.
3. In the **Apps** tab, under **Code Generation**, click **Simulink Coder**. In the **C Code** tab, click **Generate Code**.

After the model finishes generating code, a new subfolder called `sscfluids_prv_punch_operation_RT_grt_rtw` is created in your current MATLAB working folder and the Code Generation Report window opens. This subfolder contains the files created by the code generation process, including those that contain the generated C source code.

- **Build the Generated Code**

To build an executable, you must set up a supported C/ C++ compiler. For a list of compilers supported in the current release, see Supported and Compatible Compilers.

To set up your compiler, run this command in the MATLAB command prompt:

After you set up your compiler, you can build and run the compiled code. The `sscfluids_prv_punch_operation_RT` model is currently configured to only generate code. To build the generated code:

1. In the **Modeling** tab, click **Model Settings**. The Configuration Parameters dialog opens. Click **Code Generation**, then clear the **Generate code only** parameter and click **Apply**.
2. In the **C Code** tab, click **Build**.

The code generator builds the executable and generates the Code Generation Report. The code generator places the executable in the working folder. The executable generated is `sscfluids_prv_punch_operation_RT.exe`.

You can run the generated executable from MATLAB command prompt with this command to create a MAT-file that contains the same variables as those generated by simulating the model.

Cylinder Linearization

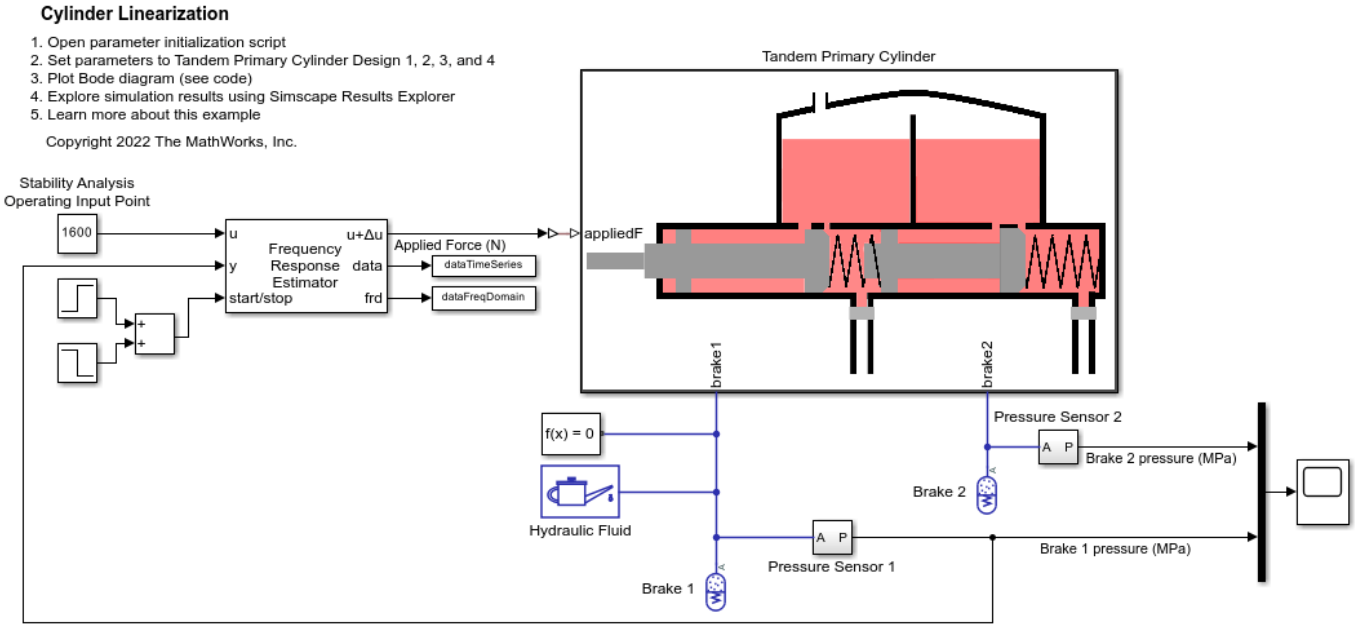
This example shows how you can linearize a Simscape™ hydro-mechanical system model to check stability of an open-loop system around an operating point. This example uses the Simulink Control Design™ *Frequency Response Estimator* block to perform a frequency response estimation experiment and store the data for later estimation offline.

Model

The model contains a plant, Tandem Primary cylinder in an open-loop configuration. The *Frequency Response Estimator* block accepts the input signal *u*. It feeds a sum of input signal and a Sinestream perturbation into the *Simulink-PS converter* block. The *Simulink-PS converter* block injects the sum of the operating point input and the sinestream to the Tandem Primary Cylinder subsystem.

```
open_system('sscv_cylinder_linearization')
```

```
set_param(find_system('sscv_cylinder_linearization','FindAll', 'on','type','annotation','Tag','Mo
```



Cylinder Linearization

1. Open parameter initialization script
2. Set parameters to Tandem Primary Cylinder Design 1, 2, 3, and 4
3. Plot Bode diagram (see code)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example

Copyright 2022 The MathWorks, Inc.

Offline Frequency Response Estimation

Frequency response describes the steady state response of a system to a sinusoidal input signal. If the system is linear *G(s)*, the output signal is a sine wave of the same frequency with a different magnitude and a phase shift. A frequency response data (*frd*) model that stores frequency response information at multiple frequencies is useful for tasks such as analyzing plant dynamics, understanding open-loop and closed-loop stability, validating linearization results, designing a control system, and estimating a parametric model.

You can use the Simulink Control Design *frestimate* command or use the Model Linearizer app to run a simulation with some perturbation signals on the Simscape model. Afterwards, the plant frequency responses are estimated offline based on the collected experiment data. This approach is called offline estimation.

When generating data from simulating the model, it is important that the physics should not encounter non-linearities like friction, saturation, hardstop, and switching. This is required for the correct frequency response estimation for the system of interest.

Hydro-mechanical System Stability Estimation Using Sinestream Mode

The model contains the *Frequency Response Estimator* block in the *control signal + perturbation* output configuration setting. You can use the *start/stop signal* to start and stop a frequency response estimation experiment in this configuration setting.

During the experiment, when the *Experiment Mode* is in *Sinestream* setting, the block injects sinusoidal signals into to the *Simulink-PS converter* block one frequency after another, from the lowest to the highest.

Frequency Response Estimator block has a *data* outport that allows you to log a time series experiment data from simulation. You can process that data set offline with the *frestimate* command to generate an *frd* object.

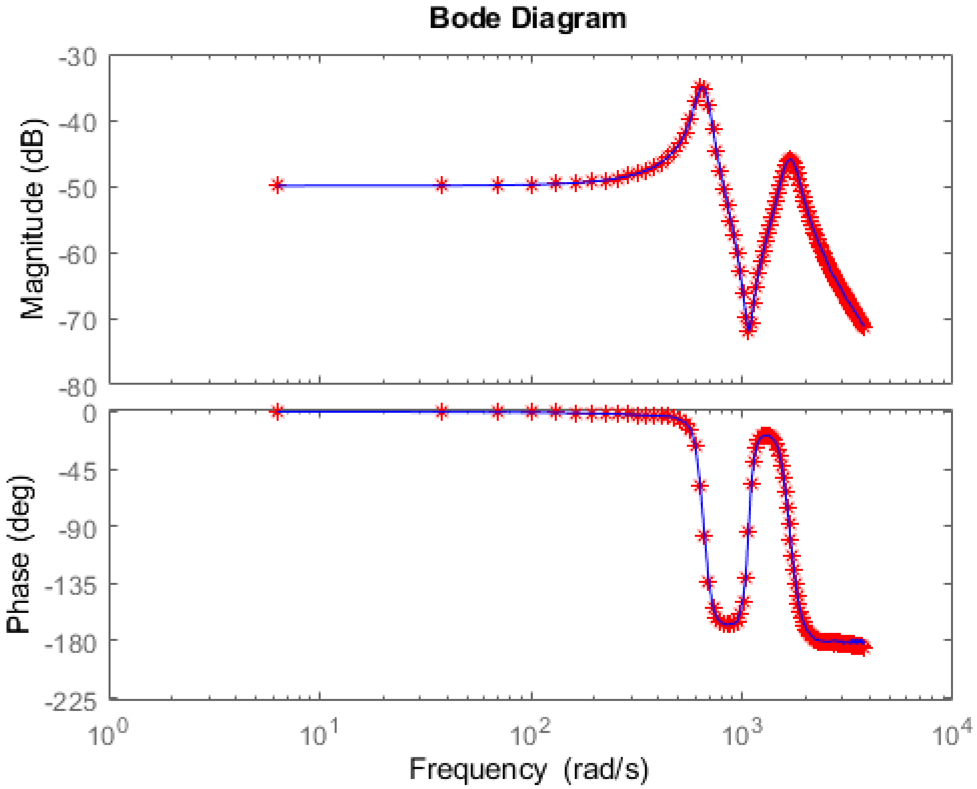
Frequency Response Estimator block also has an *frd* outport that allows you to log the frequency response data from simulation. You can process that data set offline with the *frd* command to generate an *frd* object.

```
sim('sscv_cylinder_linearization')
sysEstimFRD1 = frestimate(dataTimeSeries,omega,'rad/s');
sysEstimFRD2  = frd(dataFreqDomain,omega);
```

You can use generated *frd* objects to create the bode plot for the system. The generated bode plot will be same for the generated *frd* objects, sysEstimFRD1 and sysEstimFRD2.

```
figure;
bode(sysEstimFRD1, 'r*')
hold;
bode(sysEstimFRD2,'b')
```

```
Current plot held
```

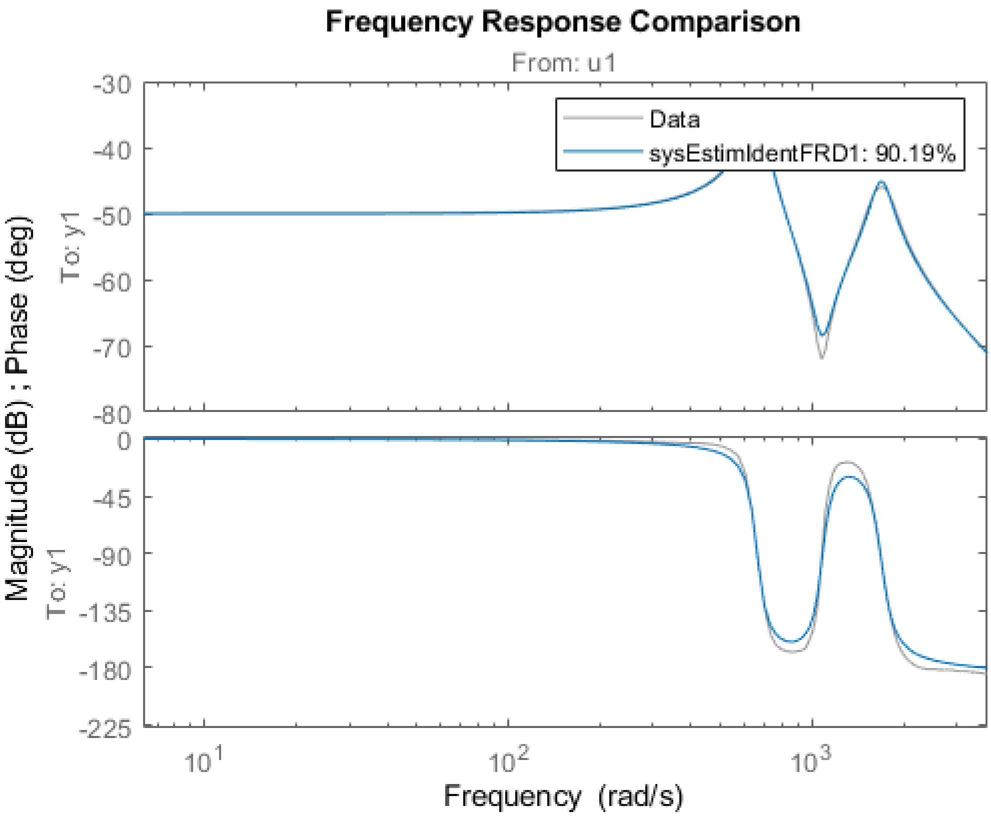


You can use the System Identification Toolbox™ *pem* or *ssest* command to identify a state-space model fitting the *frd* object. The sysEstimFRD2 *frd* object will create a continuous state-space model, however sysEstimFRD1 *frd* object will create a discrete state-space model.

```
sysEstimIdentFRD1 = ssest(sysEstimFRD1,4, 'Ts', sysEstimFRD1.Ts);
sysEstimIdentFRD2 = ssest(sysEstimFRD2,4);
```

You can use the System Identification Toolbox *compare* command to check the fit quality between the estimated state-space model, sysEstimIdentFRD1 or sysEstimIdentFRD2 and the generated *frd* object, sysEstimFRD1 or sysEstimFRD2 respectively.

```
figure;
compare(sysEstimFRD1,sysEstimIdentFRD1)
```



You can use the Control System Toolbox™ *pole* command to estimate the poles of the system using the identified state-space model. If the poles are on the left side of the s-plane for the continuous state-space model, sysEstimIdentFRD2, then the open-loop hydro-mechanical system of interest is a stable system around the selected operating point.

```
poles = pole(sysEstimIdentFRD2)
```

```
poles =

   1.0e+03 *

  -0.0392 + 0.6568i
  -0.0392 - 0.6568i
  -0.1250 + 1.6856i
  -0.1250 - 1.6856i
```

Similarly, for the discrete state-space model, sysEstimIdentFRD1, if the poles lie within the unit circle then the open-loop hydro-mechanical system of interest is a stable system around the selected operating point.

```
poles = pole(sysEstimIdentFRD1)
magnitude = abs(poles)
```

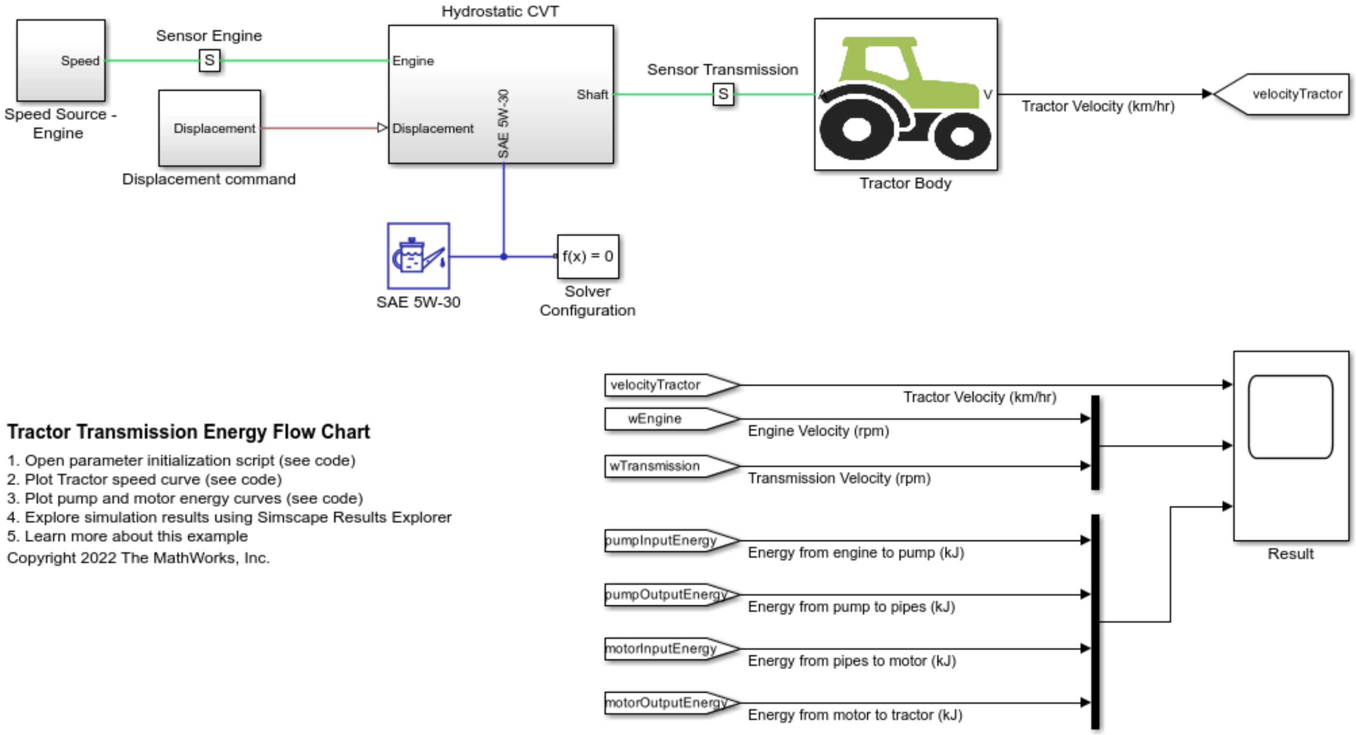
```
poles =
      0.9736 + 0.1657i
      0.9736 - 0.1657i
      0.9939 + 0.0654i
      0.9939 - 0.0654i


magnitude =

    0.9876
    0.9876
    0.9961
    0.9961
```

Tractor Transmission Energy Flow Chart

This example shows how to model, parameterize, and test a tractor with a hydrostatic continuously variable transmission (CVT). When you run one of the plot functions, you can view plots of the tractor velocity, engine speed, transmission speed, and pump and motor energy computations.

Model

The figure shows the hydrostatic CVT tractor model.



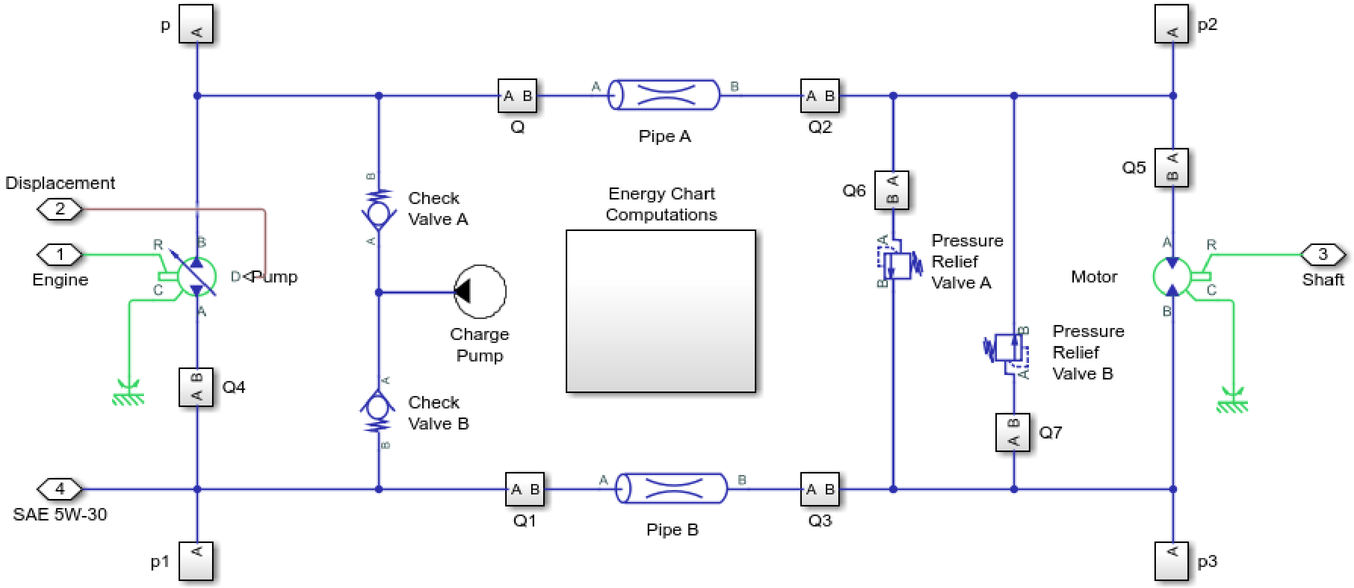
Tractor Transmission Energy Flow Chart

1. Open parameter initialization script (see code)
2. Plot Tractor speed curve (see code)
3. Plot pump and motor energy curves (see code)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example

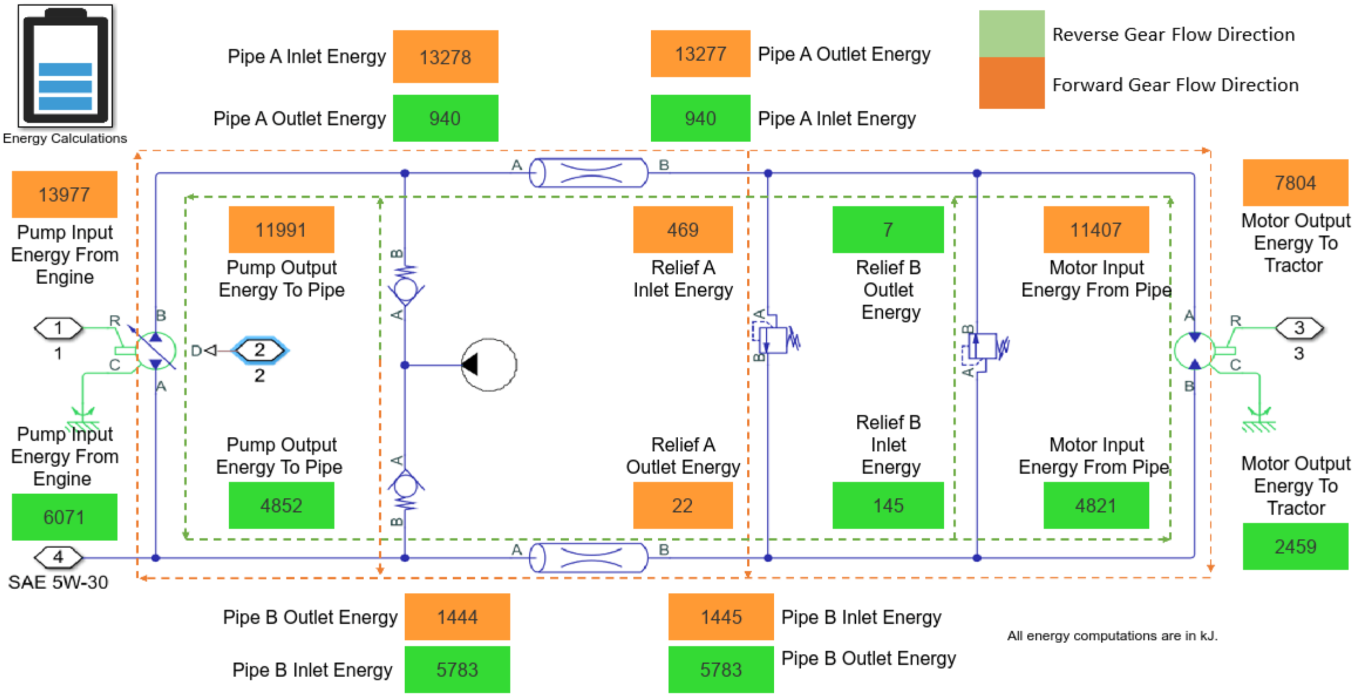
Copyright 2022 The MathWorks, Inc.

Hydrostatic CVT Subsystem

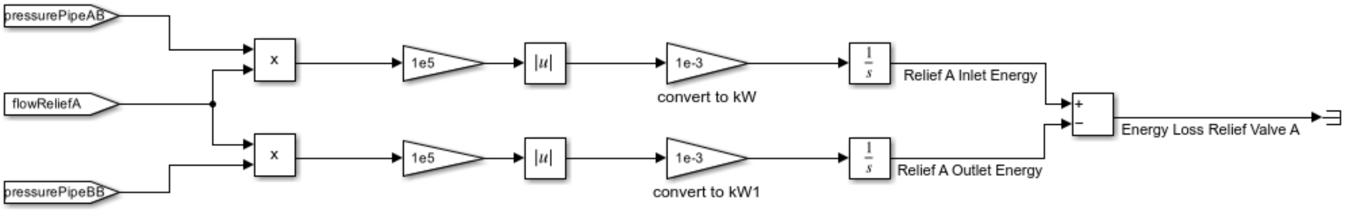
This subsystem demonstrates how to model the hydrostatic transmission. Here, connection Engine is a physical signal port associated with the speed source representing the engine. Connection Shaft is a physical signal port associated with the transmission shaft connecting with the final drive of the Tractor. Connection SAE 5W-30 is a physical signal port associated with the fluid properties of the Isothermal Liquid network that represents the transmission hydraulics. Connection Displacement is a physical signal port associated with the axial piston pump displacement command. The driver provides the swash displacement command in a tractor. The charge pump maintains the minimum pressure in the low pressure line of the circuit. The relief valves limit the maximum pressure in the circuit's high pressure line.



Energy Chart Computations: This subsystem shows the energy chart of the tractor transmission system. The green dashed line shows the fluid flow path while in reverse, and the orange line shows the fluid flowpath while in forward gears. The pump displacement is positive for the forward gear and negative for the reverse gear tractor operation. You can compute the available, useful, recirculated, wasted energies as well as the efficiency of the components from the energy flow chart.

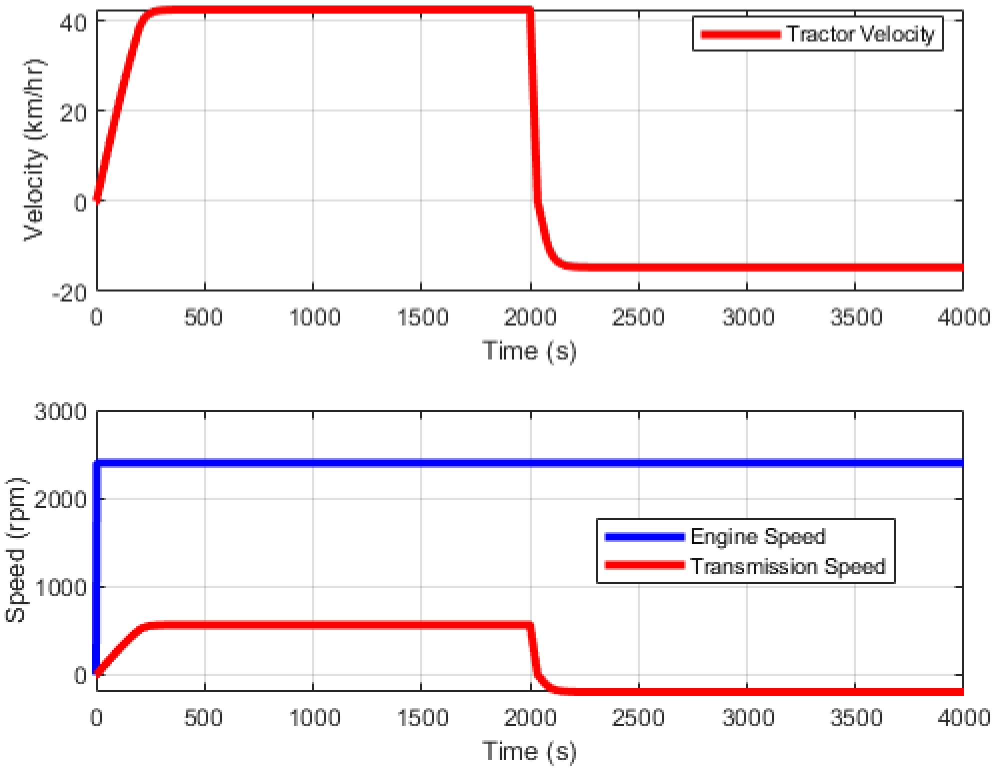


Energy Calculations: This subsystem Relief Valve A Energy Computation shows how to perform energy calculations for a relief valve.

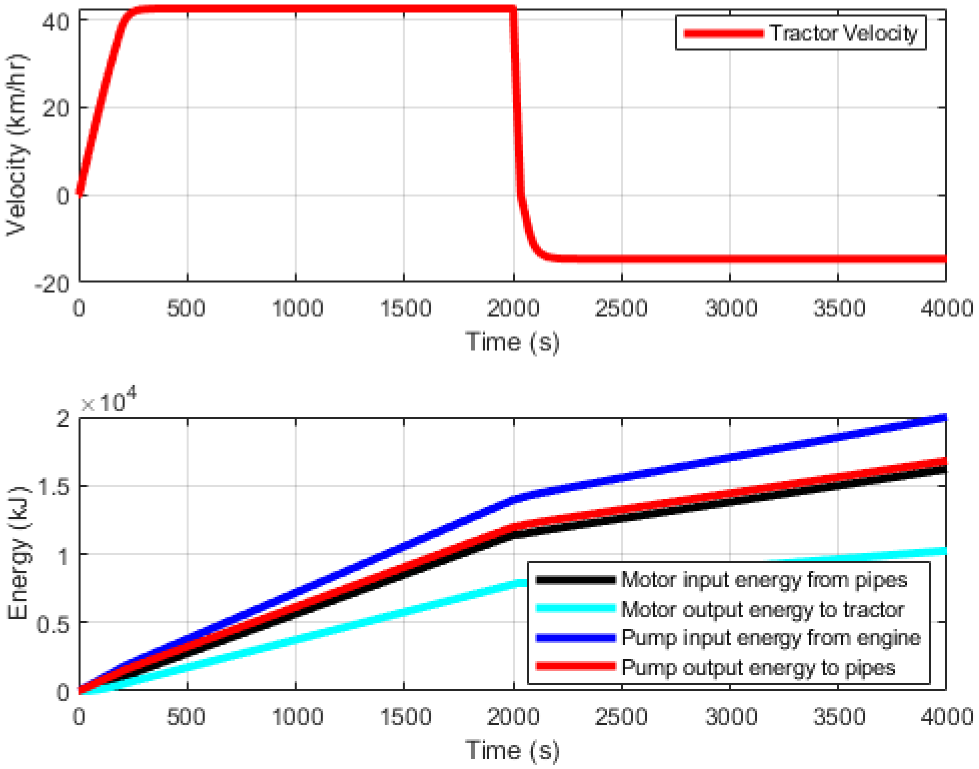


Simulation Results from Simscape™ Logging

This model generates a plot of the tractor velocity, engine speed, and transmission speed.



This model also generates a plot of the input and output energies of the pump and motor.

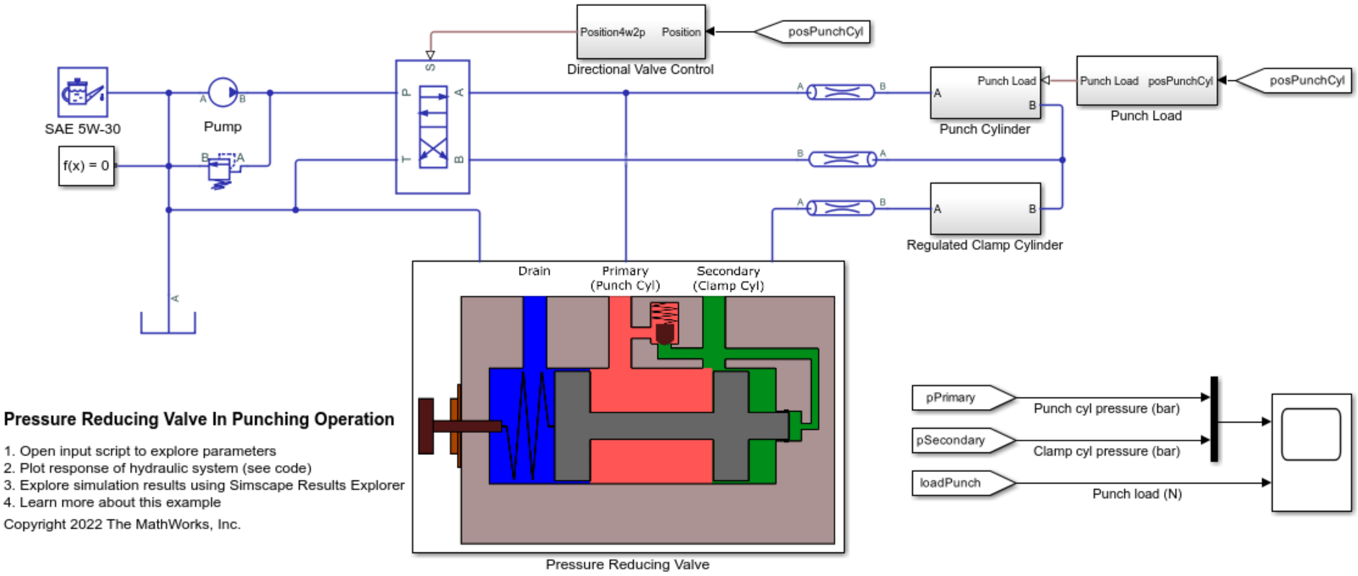

40
20
0
-20
Velocity (km/hr)
Tractor Velocity
0
500
1000
1500
2000
2500
3000
3500
4000
Time (s)
×10^4
2
1.5
1
0.5
0
Energy (kJ)
Motor input energy from pipes
Motor output energy to tractor
Pump input energy from engine
Pump output energy to pipes
0
500
1000
1500
2000
2500
3000
3500
4000
Time (s)

Pressure Reducing Valve In Punching Operation

This example models a hydraulic system with a direct operated, pressure reducing valve. This system helps to limit and maintain pressure in a hydraulic punching machine. Pressure reducing valves are common in hydraulic pressing, drilling, and stamping applications.

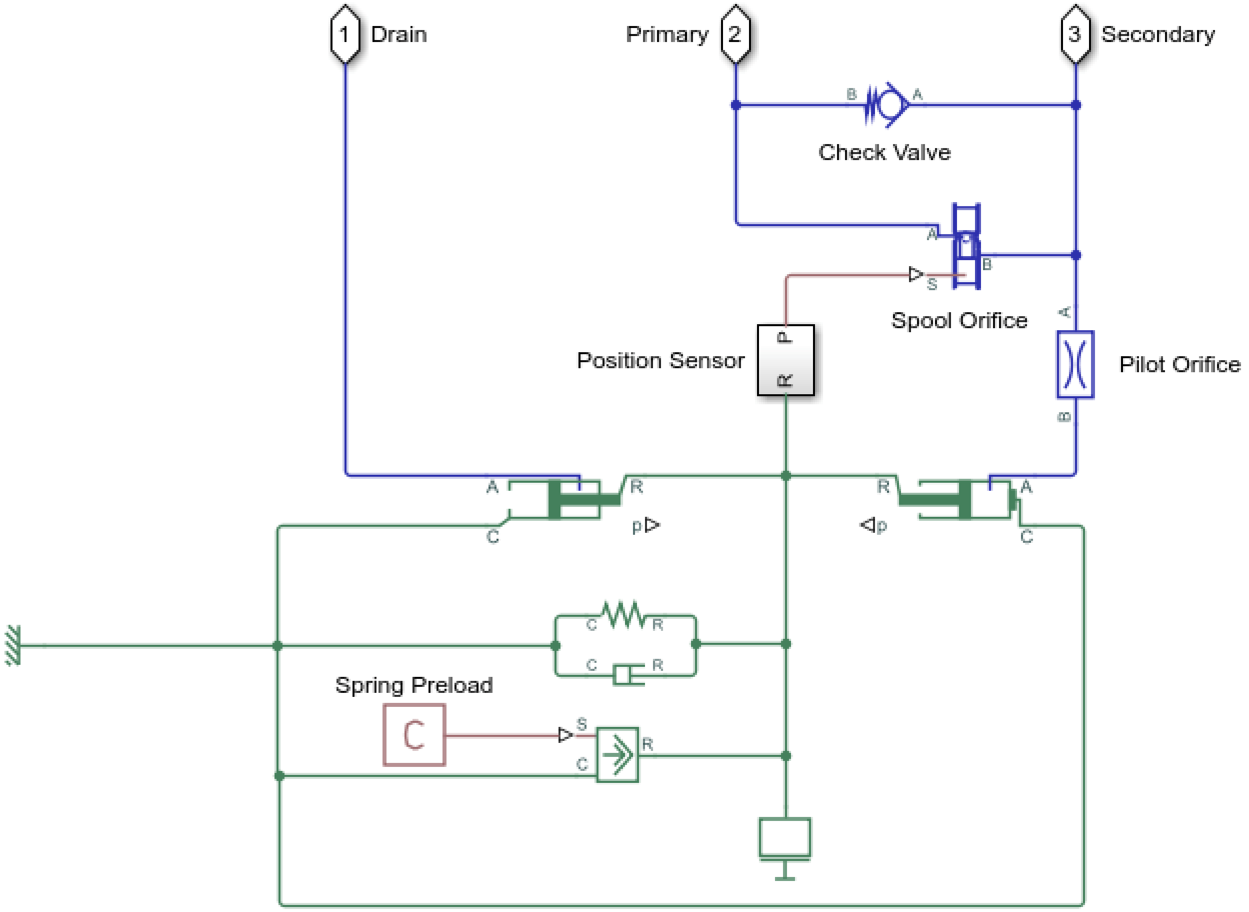
Model

The pressure reducing valve is upstream of the punch and downstream of the pump, pressure relief valve, and a 4-way 2-position directional valve The maximum pressure of the system is limited by the pressure relief valve.



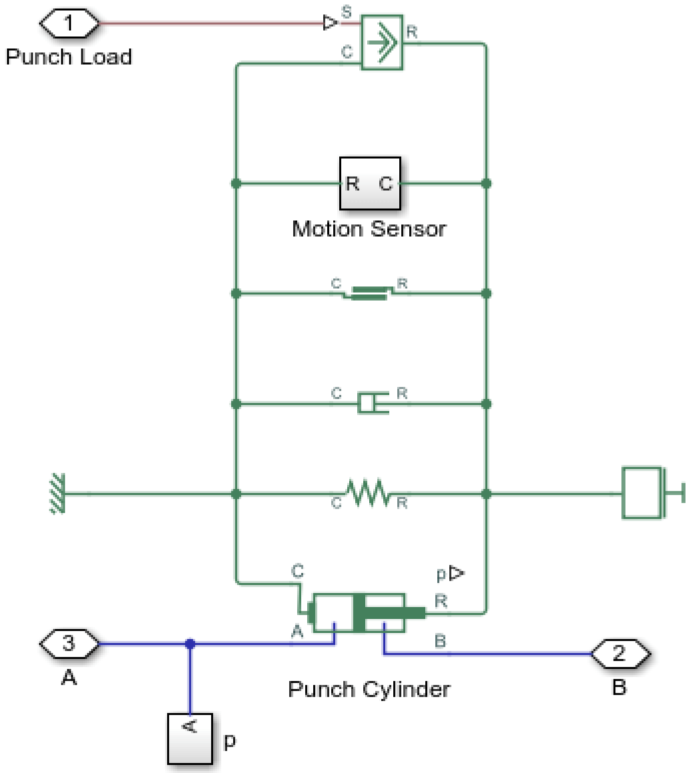
Pressure Reducing Valve Subsystem

A pressure reducing valve uses a spring-loaded spool to control the downstream pressure or the secondary pressure. If the secondary pressure is below the valve set pressure, fluid flows freely from the primary side to the secondary side. An internal passage transmits the fluid pressure from secondary side to the rightmost spool piston face. This pressure is called pilot pressure. The force due to pilot pressure opposes the spring load on the leftmost piston face. The spring may be preloaded. The spring end of the spool is connected to a drain. When the secondary pressure reaches the set pressure, the spool quickly moves to close the secondary port and thus limits and maintains secondary pressure to the set value.



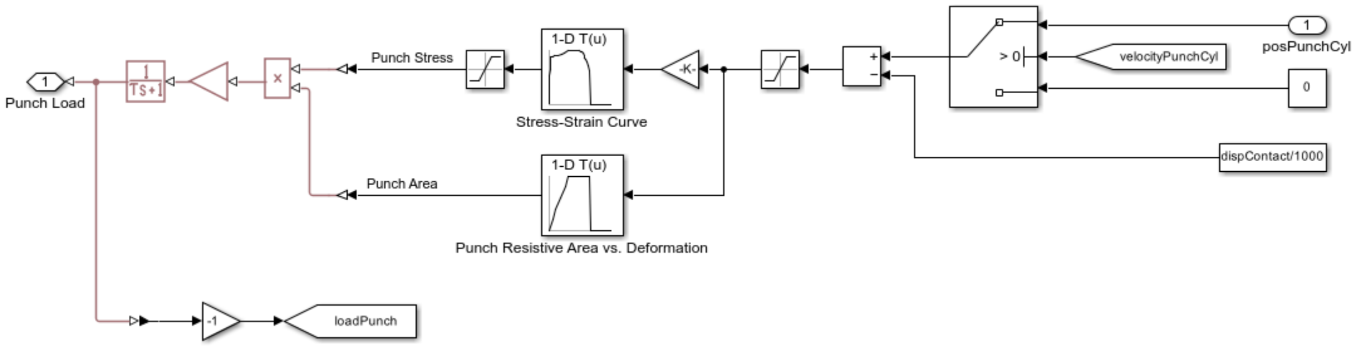
Punch Cylinder Subsystem

The punching operation has three phases. Phase one is the extension of the punch cylinder till the punch comes in contact with the metal work piece. In phase two, the punch cylinder extends more as the punch cuts through the work piece. During this phase, the pressure reducing valve controls a hydraulic clamp to keep the work piece steady. In phase three, the punch retracts. It is triggered by the control of the 4-way 2-position directional valve when the punch cylinder completes its extension stroke. A check valve present inside the reducing valve body, opens to relieve the secondary pressure during the retraction phase. The punch cylinder again starts to extend after retraction.



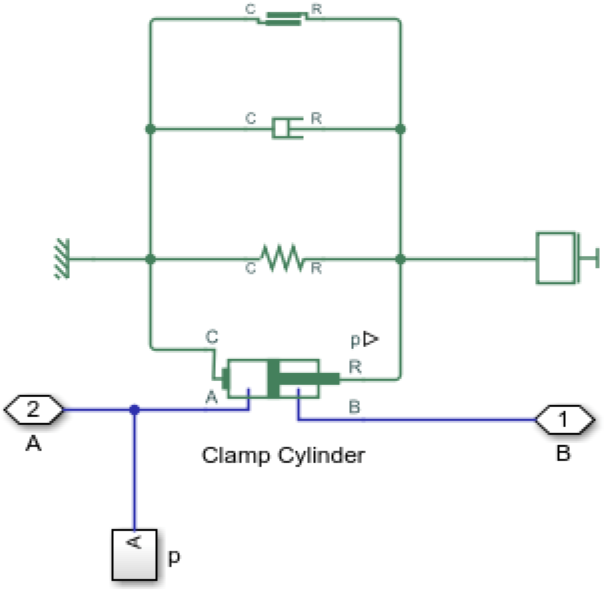
Punch Load Subsystem

The subsystem titled 'Punch Load' calculates the resistive force on the punch cylinder during punching operation. The example uses the work piece deformation and the stress-strain curve to calculate the punch stress. The punching area enlarges as the punch moves into the work piece while a hole is punched.



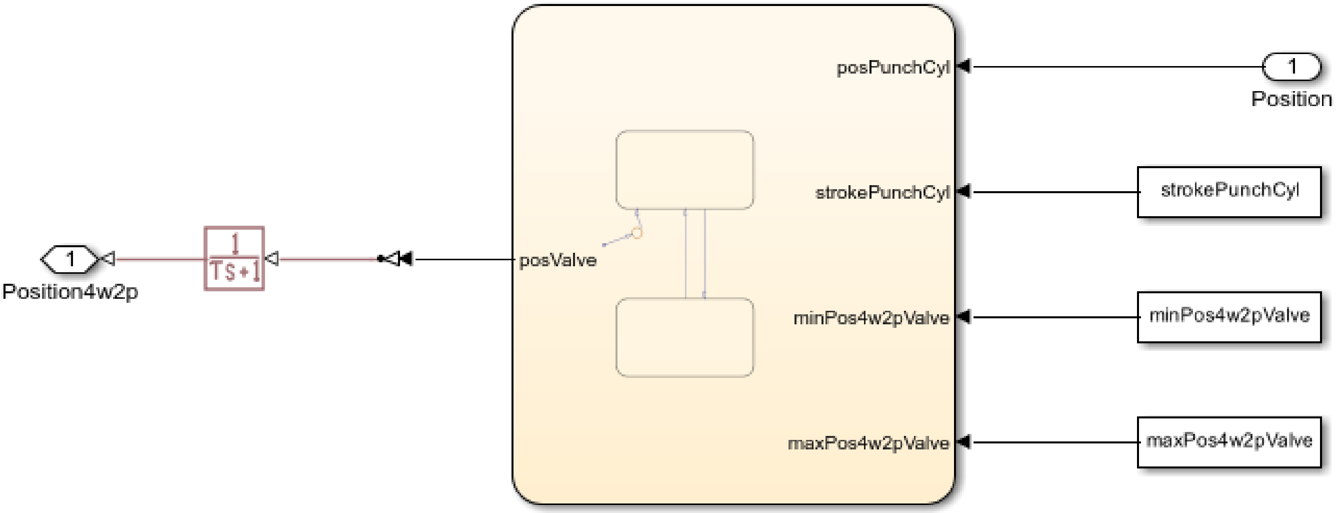
Regulated Clamp Cylinder Subsystem

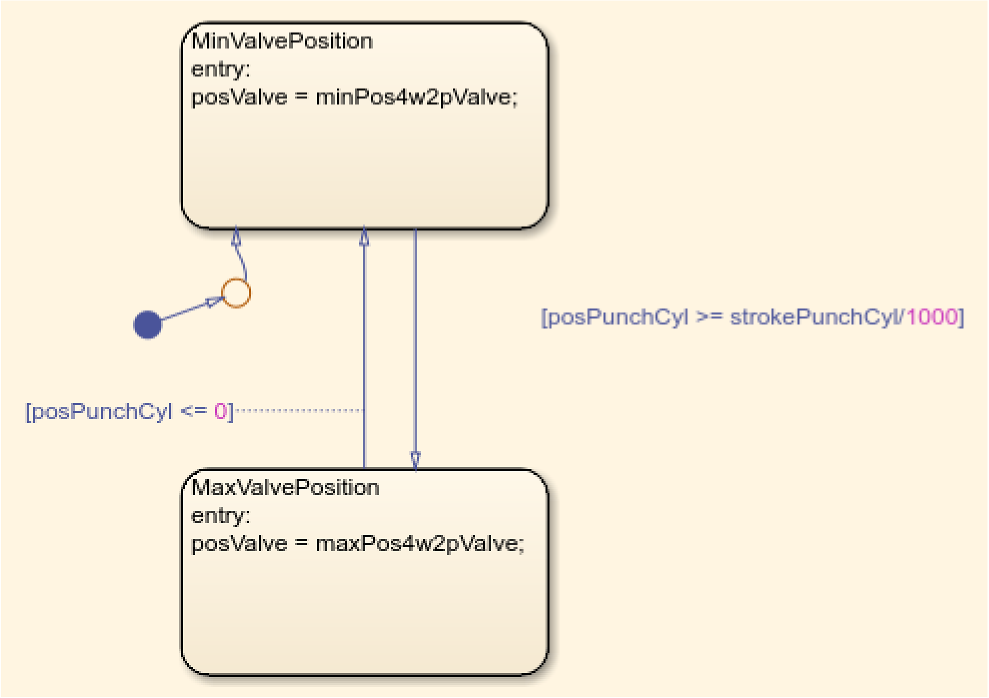
As the pump drives fluid into the punch and clamp cylinders, the pressure in the cap end chambers of both cylinders rises and both the cylinders extend. The primary port of the pressure reducing valve is connected to the cap end of the punch cylinder, and the secondary port is connected to the cap end of the clamp cylinder. As the clamp cylinder pressure reaches the set pressure of the pressure reducing valve, the spool orifice of the reducing valve closes and thus limits the clamp cylinder pressure, keeping the work piece steady. The punch cylinder pressure continues to rise till the punch stroke is completed, while the clamp cylinder pressure is maintained same.



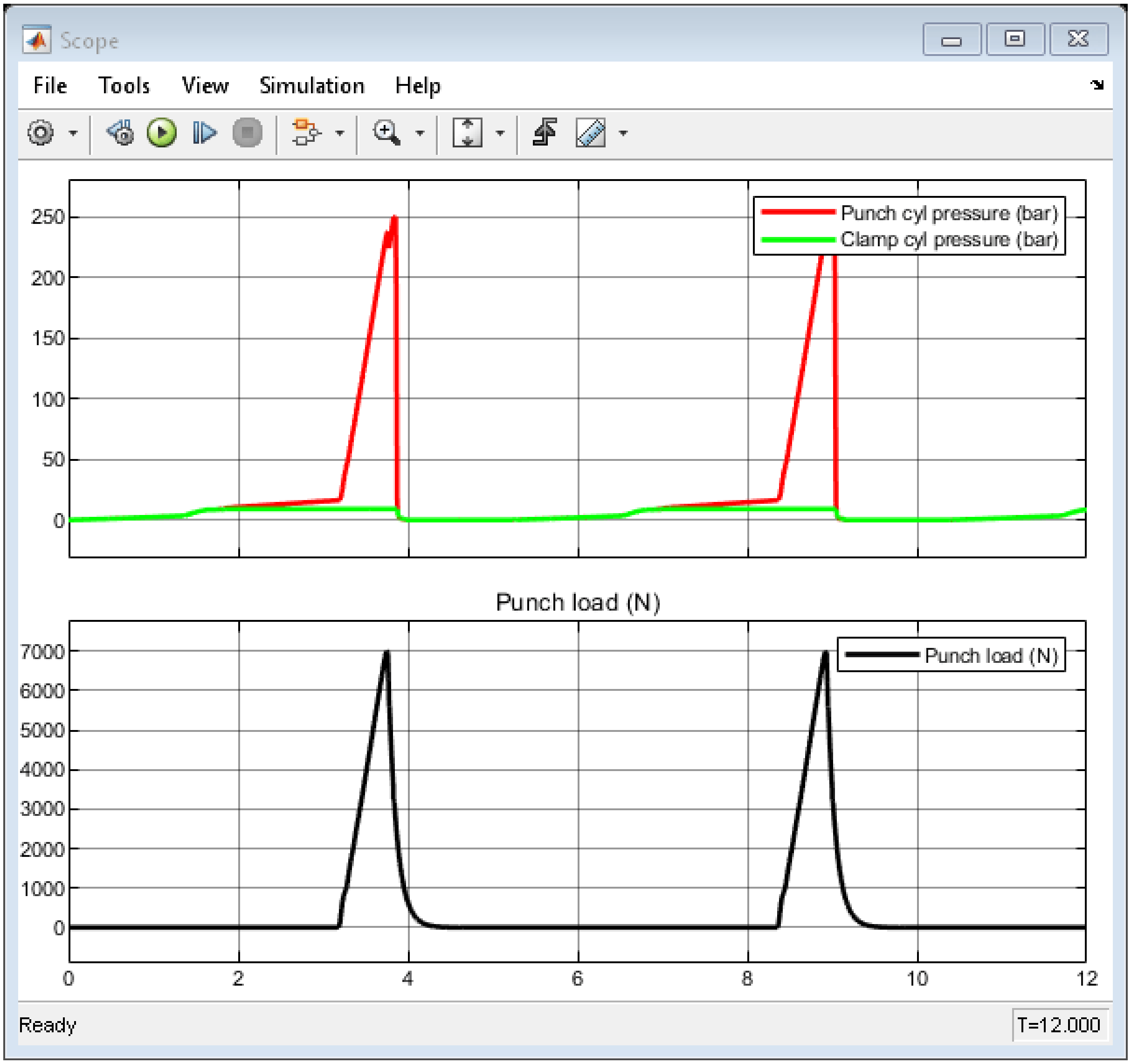
Directional Valve Control Subsystem

A motion sensor senses the punch cylinder displacement and transmits a feedback signal to the directional valve controller. This example uses a Stateflow® chart to model a simple controller that shifts the position of the directional valve. When the punch cylinder is fully retracted, or in other words, at zero displacement, flow occurs from the P port to the A port and from the B port to the T port. When the punch cylinder extends to its maximum position, the directional valve position is shifted such that flow occurs from the P port to the B port and from the A port to the T port.



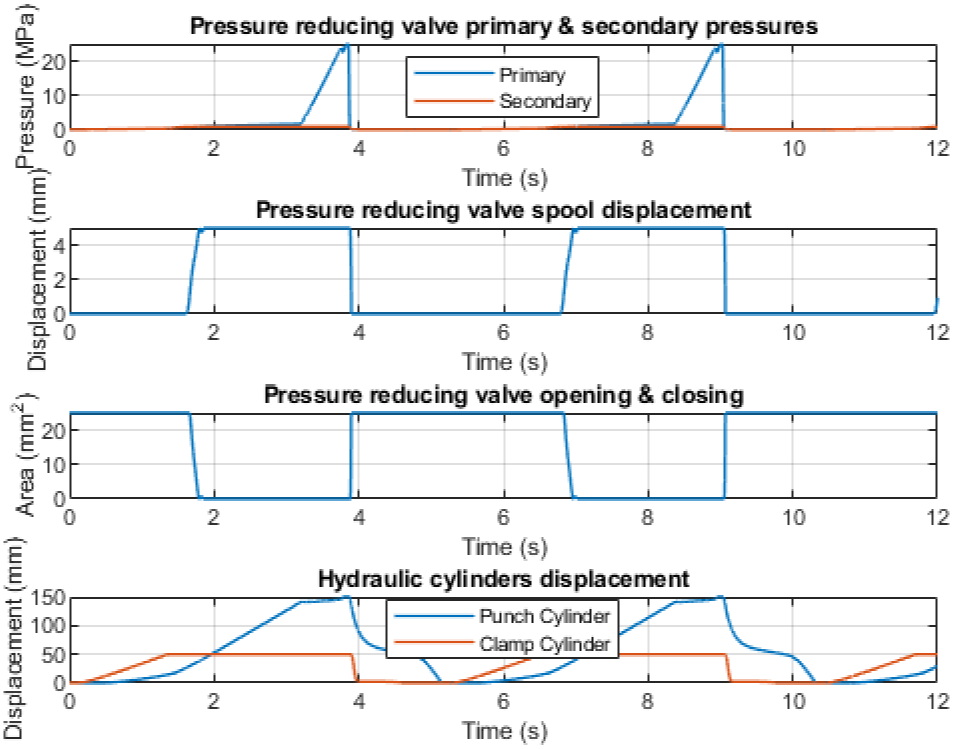

MinValvePosition
entry:
posValve = minPos4w2pValve;
[posPunchCyl >= strokePunchCyl/1000]
[posPunchCyl <= 0]
MaxValvePosition
entry:
posValve = maxPos4w2pValve;

Simulation Results from Scope



Simulation Results from Simscape™ Logging

The first and topmost subplot shows the pressure variation of the punch cylinder and the clamp cylinder during extension and retraction phases. The second subplot shows the spool travel of the pressure reducing valve. The third subplot shows the spool orifice area of the valve indicating valve opening and closing. The fourth and bottommost subplot shows the displacement of the hydraulic cylinders used for punching and clamping the work piece.


Pressure reducing valve primary & secondary pressures
Pressure (MPa)
0
10
20
Primary
Secondary
0
2
4
6
8
10
12
Time (s)
Pressure reducing valve spool displacement
Displacement (mm)
0
2
4
Time (s)
Pressure reducing valve opening & closing
Area (mm^2)
0
10
20
Time (s)
Hydraulic cylinders displacement
Displacement (mm)
0
50
100
150
Punch Cylinder
Clamp Cylinder
Time (s)

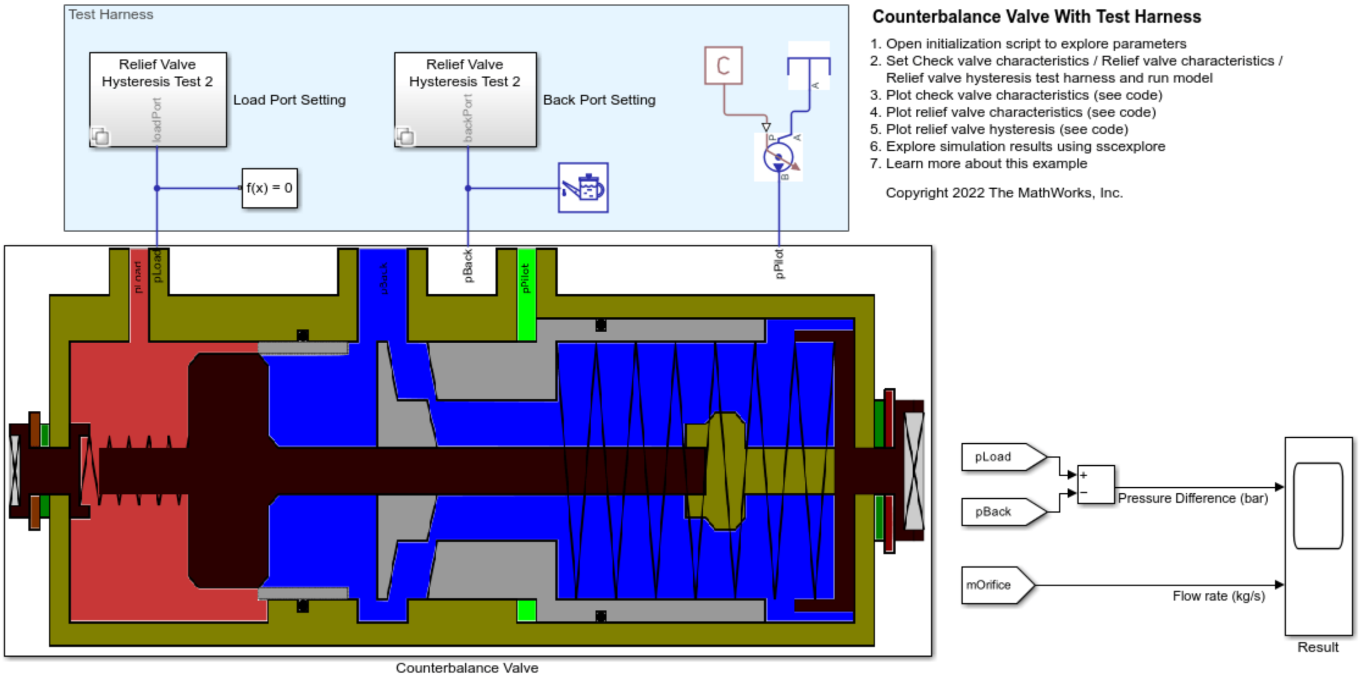
Counterbalance Valve With Test Harness

This example shows how to model, parameterize, and test a counterbalance valve. When you run the model, it generates a plot of the flow from the load port to the back port due to the relief action of the counterbalance valve. A counterbalance valve allows an upstream flow from the back port to the load port through the check valve stage, and it allows a downstream flow from the load port to the back port through the relief stage. Counterbalance valves help with the load holding, load actuation speed control, and safety.

Model

The following figure shows the model of a counterbalance valve. Here, connection pLoad is a physical signal port associated with the counterbalance valve load port. Connection pBack is a physical signal port associated with the counterbalance valve back port. This can be the prime mover or the reservoir depending on the mode of operation. Connection pPilot is a physical signal port associated with the external pilot signal. This represents an internal-external pilot and internal drain type counterbalance valve.

The figure shows a cutaway view, where the gray areas belong to the same, solid valve body. The left spring is the check valve stage spring. The right spring is the relief valve stage spring.



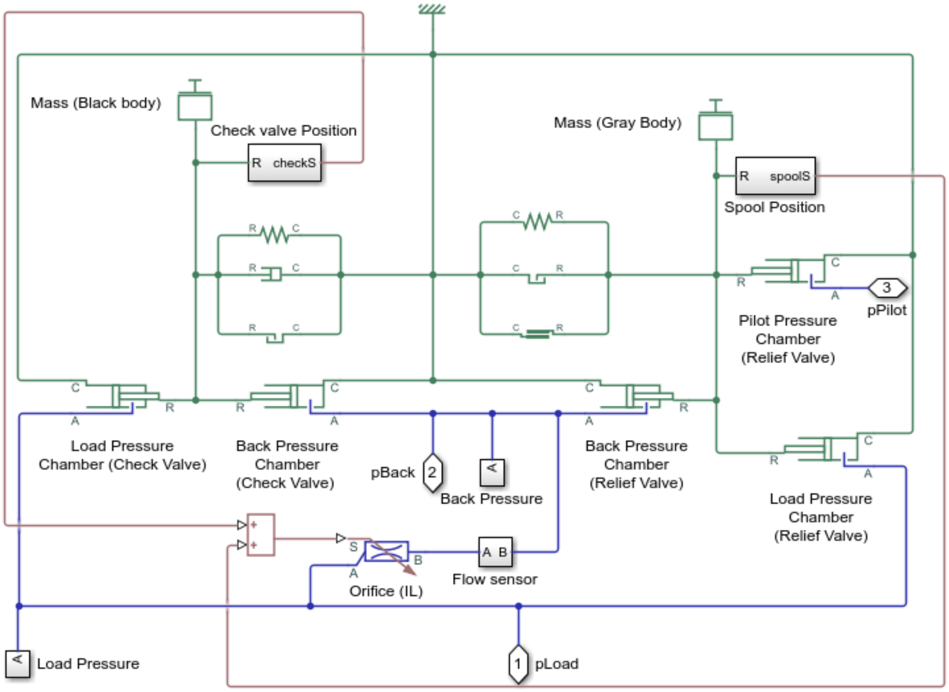
Counterbalance Valve With Test Harness

1. Open initialization script to explore parameters
2. Set Check valve characteristics / Relief valve characteristics / Relief valve hysteresis test harness and run model
3. Plot check valve characteristics (see code)
4. Plot relief valve characteristics (see code)
5. Plot relief valve hysteresis (see code)
6. Explore simulation results using sscexplore
7. Learn more about this example

Copyright 2022 The MathWorks, Inc.

Counterbalance Valve Subsystem

This subsystem demonstrates how to model the counterbalance valve. It includes the modeling of the *Check Valve Function*, *Relief Function*, and *Orifice (IL)*. The *Orifice (IL)* models the flow area created due to movement of the check valve poppet and the relief stage spool.



The Orifice permits flow from the load port to the back port when the relief function is active. This represents the main function of the counterbalance valve. This occurs during lowering of the load. The relief stage of the counterbalance valve provides load holding and load actuation speed control functions, which improves safety.

The Orifice also permits flow from the back port to the load port when the check valve stage is active. When the check valve cracks, free upstream flow happens from back pressure line to actuator load line.

The *Check Valve Function* models the mass dynamics of the check stage poppet (Black colored body). The check stage operation allows the free upstream flow from the pBack port to the pLoad port of the counterbalance valve.

The *Relief Function* models the mass dynamics of the relief valve stage spool (Gray colored body). The relief valve stage operation helps with the load holding, load actuation speed control, and safety in the machine operations. The relief valve stage operation allows the controlled reverse downstream flow from the pLoad port to the pBack port of the counterbalance valve.

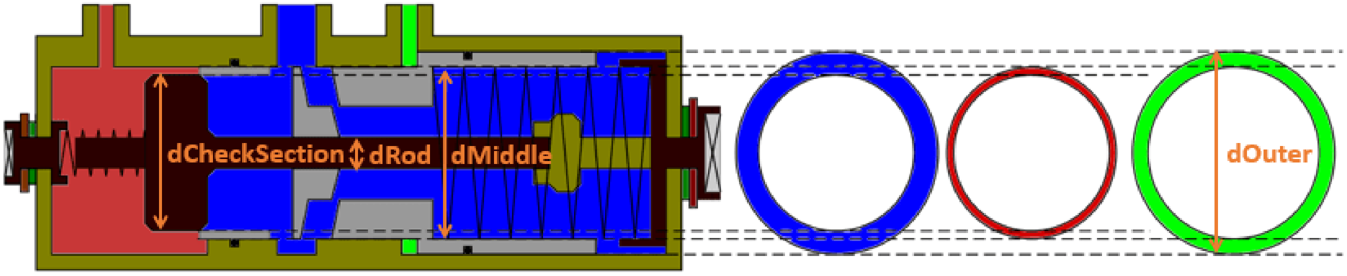
The following equation represents the mass dynamics of the relief stage spool.

$$m_R.\frac{d^2x_R}{dt^2} + k_R.x_R = p_L.A_L + p_P.A_P - p_B.A_B - k_R.xPre_R - friction - fHardStop$$

where,

1. x_R is the displacement of the relief stage spool from the steady state position.
2. m_R is the mass of the relief stage spool
3. k_R is the spring stiffness of the relief stage's spring
4. $xPre_R$ is the spring pre-compression of the relief stage's spring
5. $friction$ is the friction force
6. $fHardStop$ is the hardstop force
7. A_L is the annular area of the red colored ring
8. p_L is the pressure that acts on $areaLoad$ area
9. A_B is the annular area of the blue colored ring

10. p_B is the pressure that acts on $areaBack$ area
11. A_P is the annular area of the green colored ring
12. p_P is the pressure that acts on $areaPilot$ area

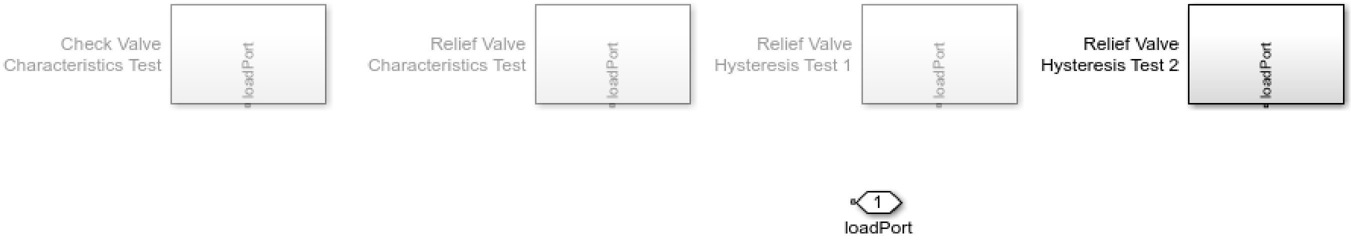


The model assumes that the flow forces due to fluid inertia and momentum change at the ports are negligible. To perform hysteresis and stability analysis, you must include these forces. You can model a flow force using a known force equation or data generated from CFD analysis. Note that these analyses are sensitive to the friction model type and parameters.

Load Port Setting and Back Port Setting Subsystem

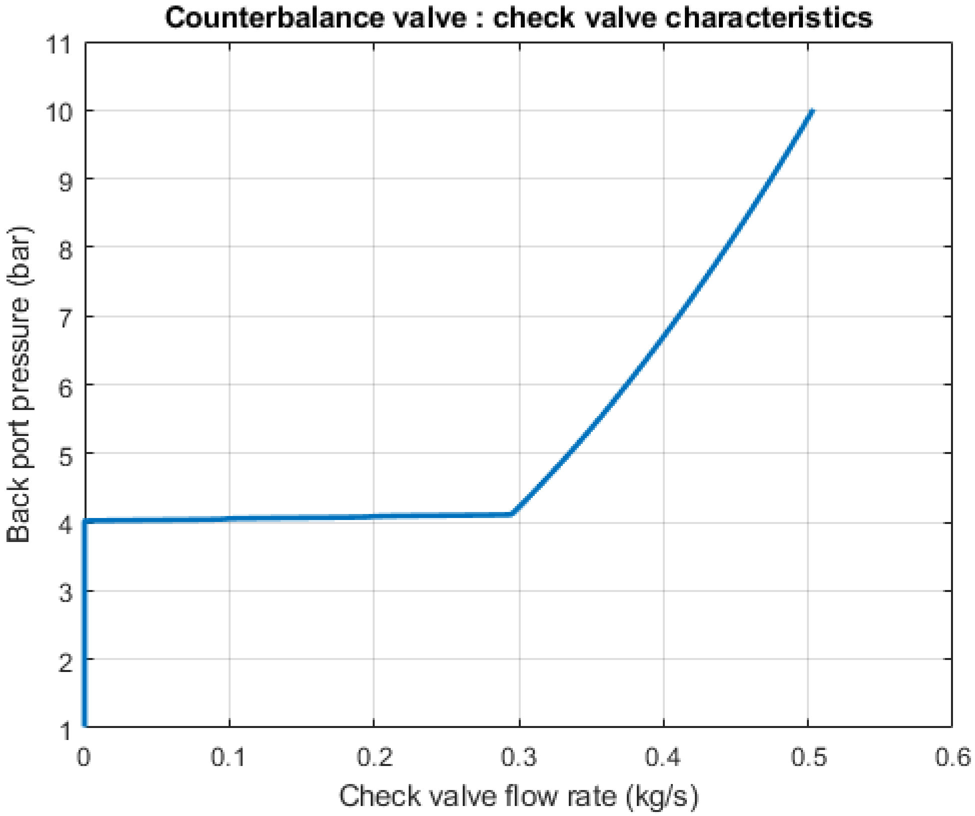
These subsystems set up the boundary condition at the load port and back port. These subsystems use Simulink® variants to set up the test harness for the check stage and relief stage testing.

1. *portSetting=1;* sets up the model to test the check valve stage characteristics.
2. *portSetting=2;* sets up the model to test the relief valve stage characteristics.
3. *portSetting=3; or portSetting=4;* sets up the model to test the relief valve stage hysteresis. The model includes two flow profiles. However, there are numerous valid flow profiles because the operator can not precisely control the flow injected to the load port while testing the valve.

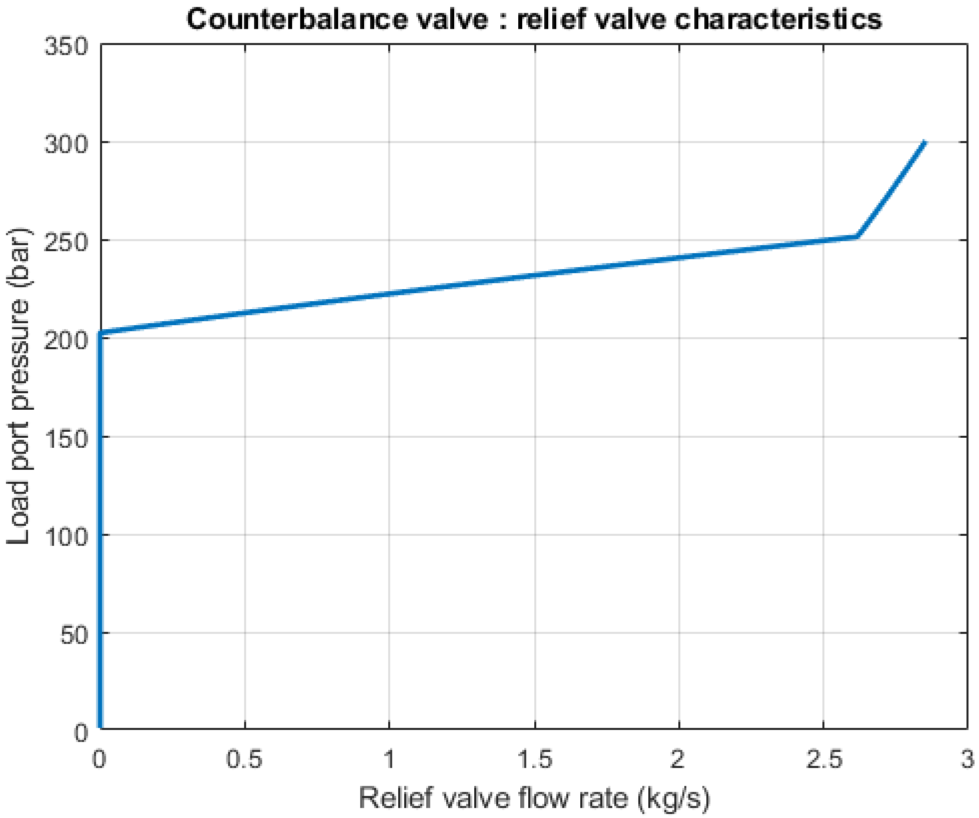


Simulation Results from Simscape™ Logging

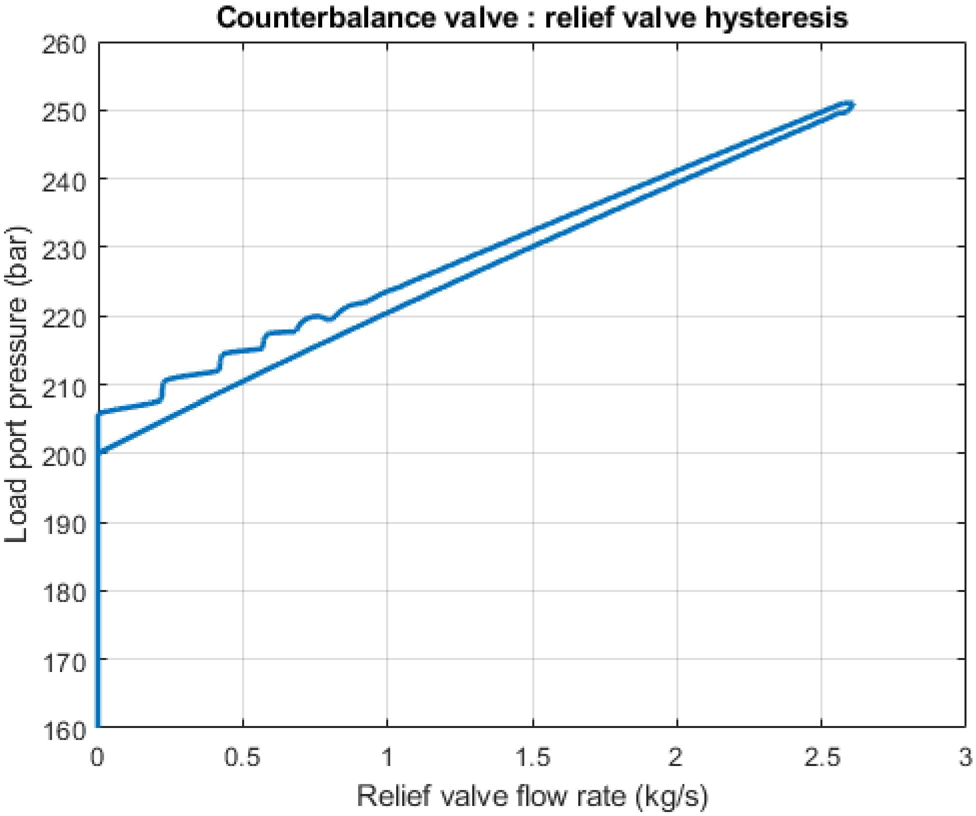
You can use this model to generate the check valve stage characteristics of the counterbalance valve using *sscfluids_counterbalance_valve_check_characteristics* script.



You can use this model to generate the relief valve stage characteristics of the counterbalance valve using *sscfluids_counterbalance_valve_relief_characteristics* script.



You can use this model to generate the relief valve stage hysteresis curve of the counterbalance valve using *sscfluids_counterbalance_valve_relief_hysteresis* script.

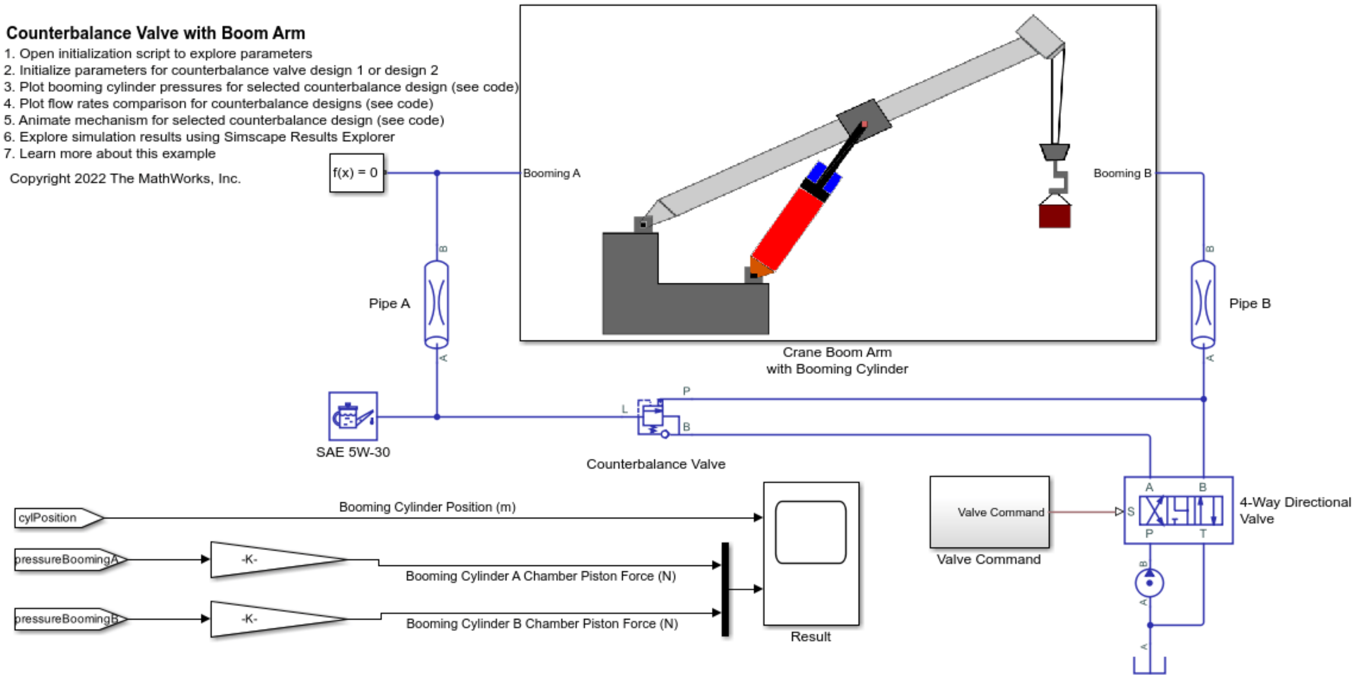

Counterbalance valve : relief valve hysteresis
Load port pressure (bar)
260
250
240
230
220
210
200
190
180
170
160
0
0.5
1
1.5
2
2.5
3
Relief valve flow rate (kg/s)

Counterbalance Valve with Boom Arm

This example shows how to model, parameterize, test, and compare design cases for a counterbalance valve in a boom arm mechanism. A counterbalance valve allows an upstream flow from the back port to the load port through the check valve stage during cylinder rod extension, and it allows a downstream flow from the load port to the back port through the relief valve stage during cylinder rod retraction in the boom arm mechanism. Counterbalance valves help with the load holding, load actuation, speed control, and safety by controlling the piston motion of the booming cylinder. When you run the model, it generates a plot of the booming cylinder position and the forces at the A side and B side of the cylinder piston.

Model

The following figure shows the model of a counterbalance valve with a boom arm.

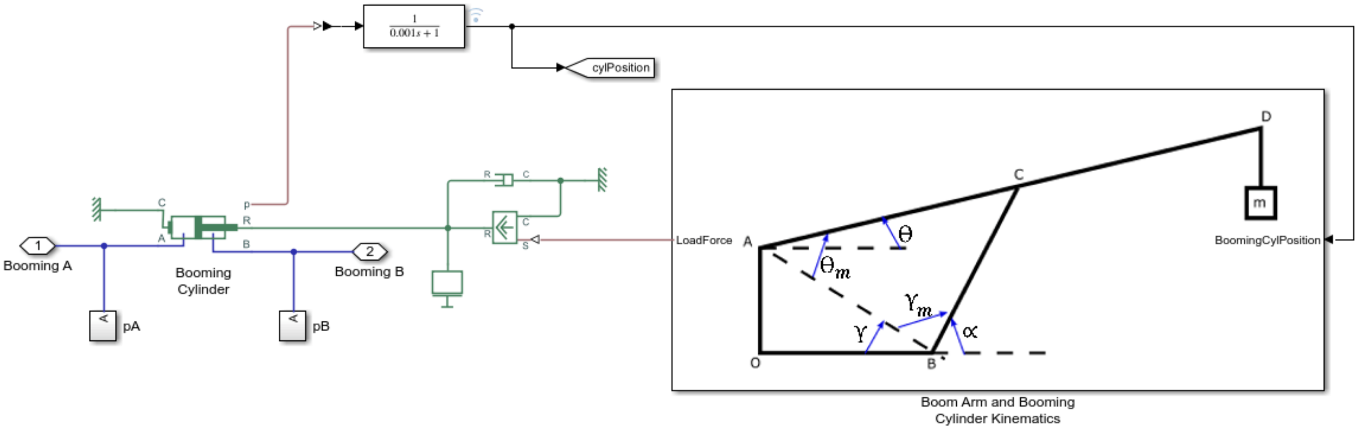


Counterbalance Valve with Boom Arm

1. Open initialization script to explore parameters
2. Initialize parameters for counterbalance valve design 1 or design 2
3. Plot booming cylinder pressures for selected counterbalance design (see code)
4. Plot flow rates comparison for counterbalance designs (see code)
5. Animate mechanism for selected counterbalance design (see code)
6. Explore simulation results using Simscape Results Explorer
7. Learn more about this example

Copyright 2022 The MathWorks, Inc.

Crane Boom Arm with Booming Cylinder Subsystem: Connection Booming A and Booming B are the isothermal liquid ports associated with the cap and rod side hydraulic nodes of the booming cylinder.



Boom Arm and Booming Cylinder Kinematics Subsystem: This subsystem demonstrates how to compute the link angles relative to one another and the force applied by the boom arm on the booming cylinder rod.

$$tan(\gamma) = \frac{bracketYPosition}{bracketXPosition}$$

$$cos(\theta_m) = \frac{bracketXPosition^2 + bracketYPosition^2 + lengthConnect^2 - (lengthBoomingCylinder + cylPosition)^2}{2.lengthConnect.(bracketXPosition^2 + bracketYPosition^2)^{0.5}}$$

$$cos(\gamma_m) = \frac{bracketXPosition^2 + bracketYPosition^2 + (lengthBoomingCylinder + cylPosition)^2 - lengthConnect^2}{2.(lengthBoomingCylinder + cylPosition).(bracketXPosition^2 + bracketYPosition^2)^{0.5}}$$

$$\alpha = 180^o - (\gamma + \gamma_m)$$

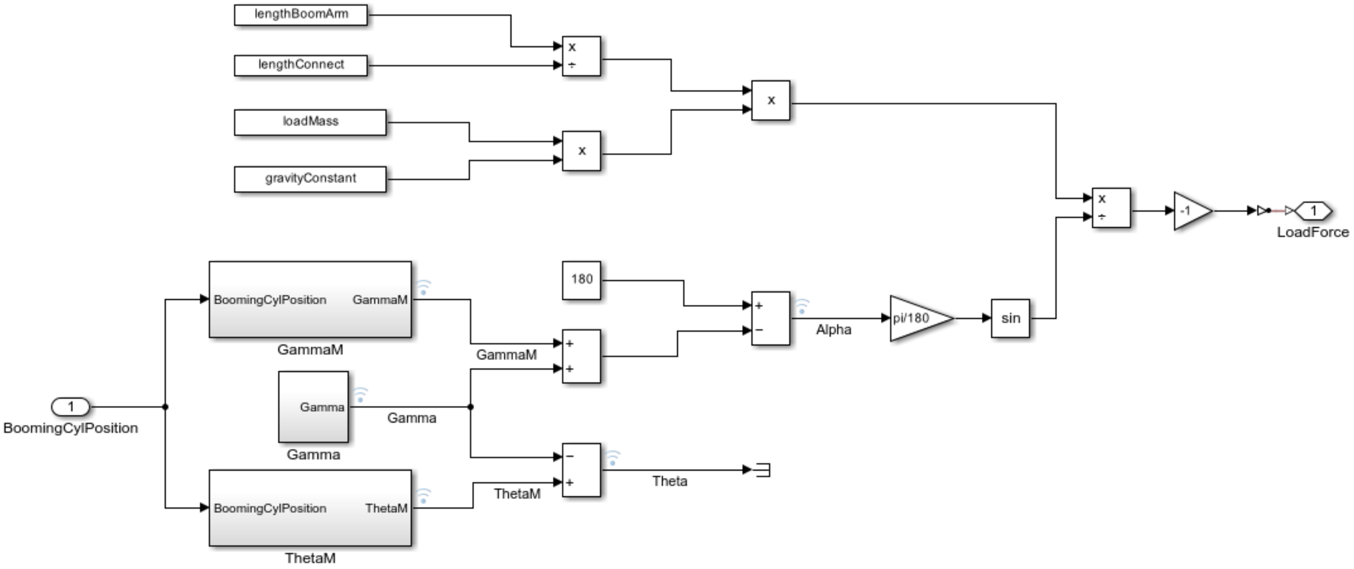
$$\theta = \theta_m - \gamma$$

$$R = \frac{mg.lengthBoomArm}{lengthConnect.sin(\alpha)}$$

where,

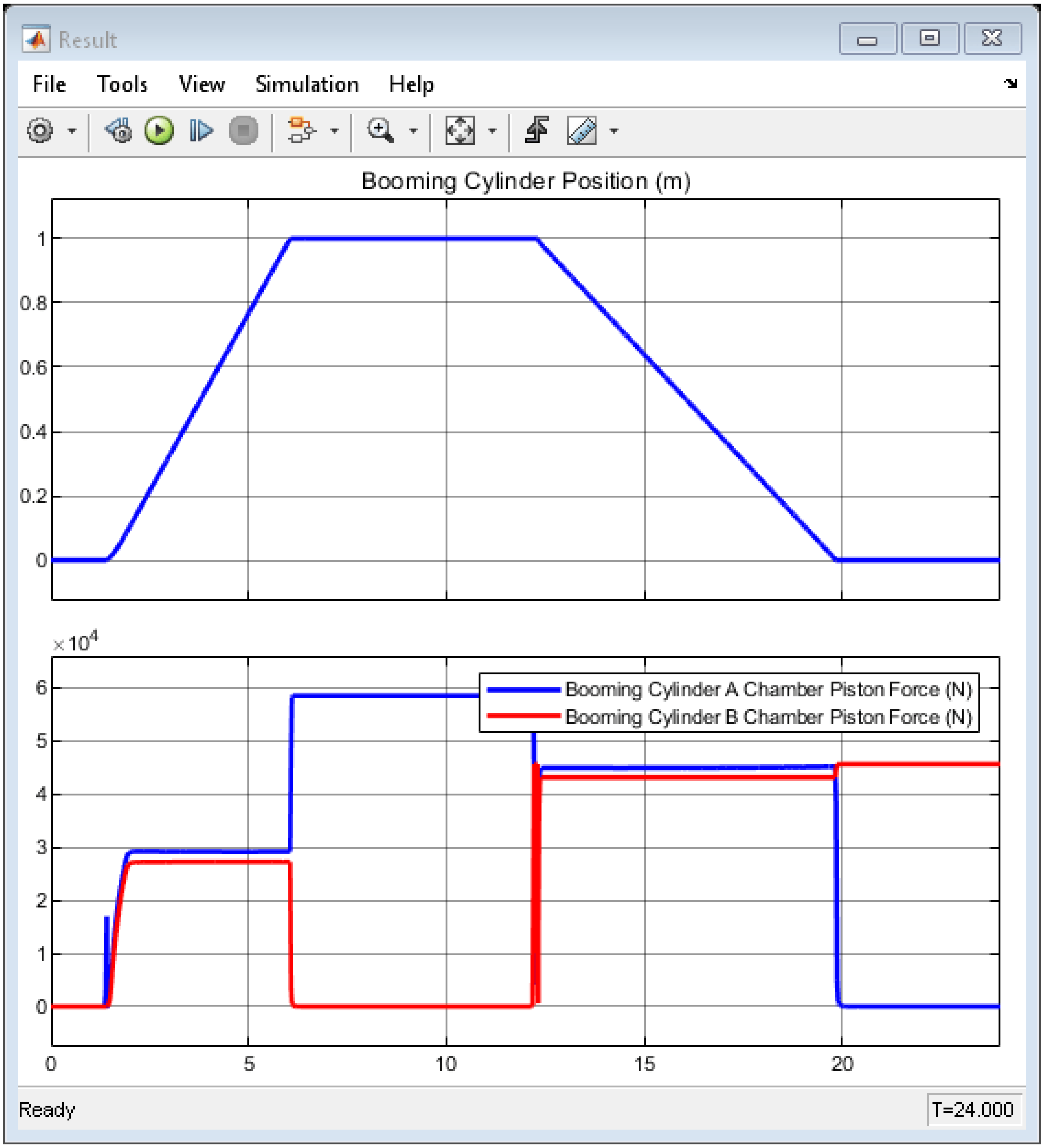
1. AOB is the bracket
2. BC is the booming cylinder
3. AD is the boom arm
4. $bracketXPosition$ is the booming cylinder connection location to the bracket (OB)
5. $bracketYPosition$ is the boom arm connection location to the bracket (OA)
6. $lengthBoomArm$ is the boom arm length
7. $lengthConnect$ is the distance between the bracket and the connection point of the boom arm and the booming cylinder (AC)
8. $cylPosition$ is the booming cylinder rod extension length
9. $lengthBoomingCylinder$ is the booming cylinder length

10 R is the force applied to the booming cylinder by the boom arm



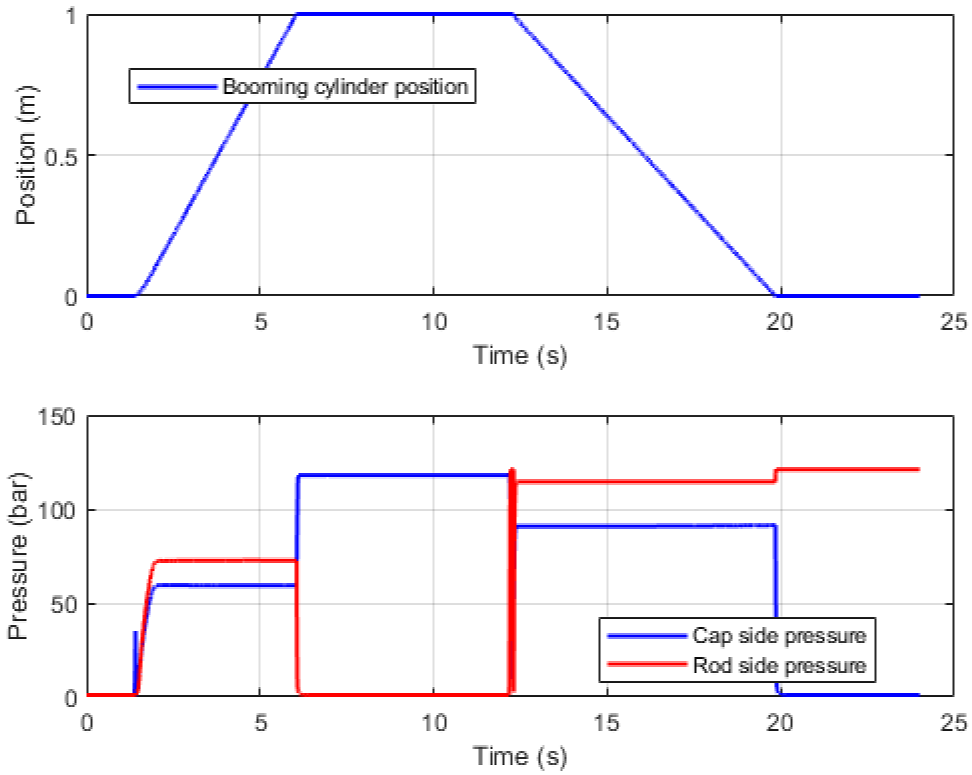
Simulation Results from Scope

This figure shows the booming cylinder position and the forces at the A side and B side of the cylinder piston.

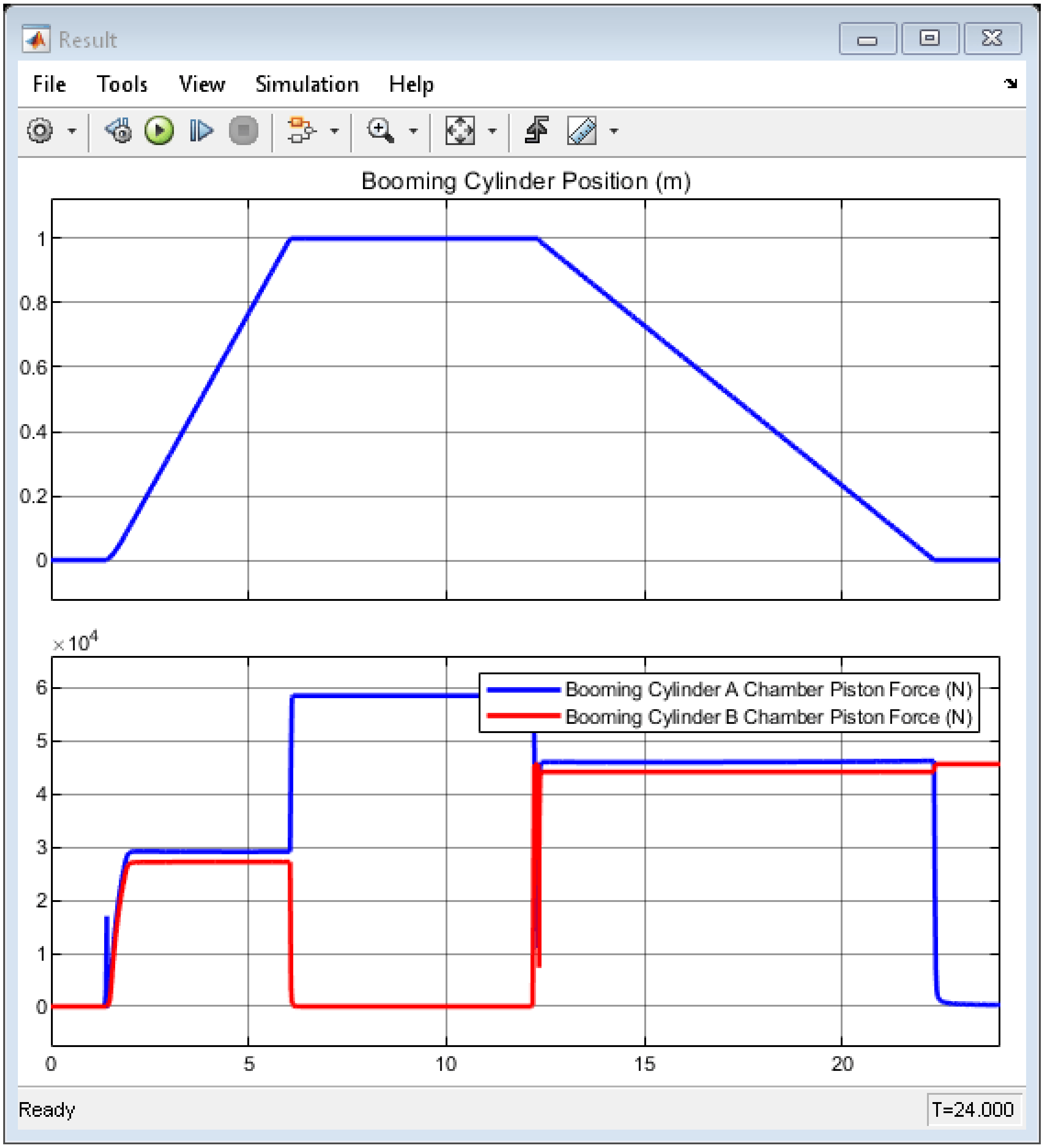


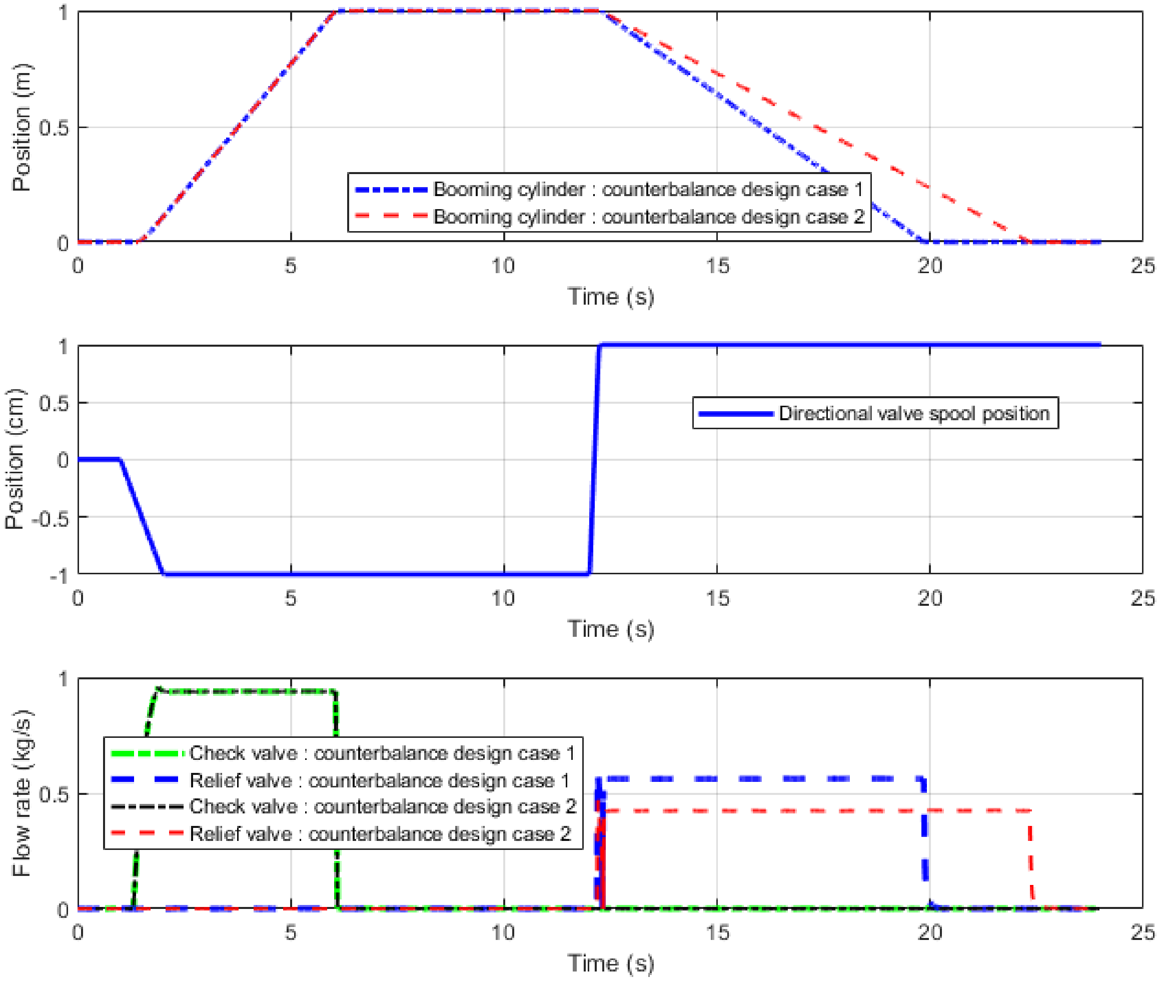
Simulation Results from Simscape™ Logging

This figure shows the booming cylinder position and the pressures at the A side and B side of the cylinder piston.



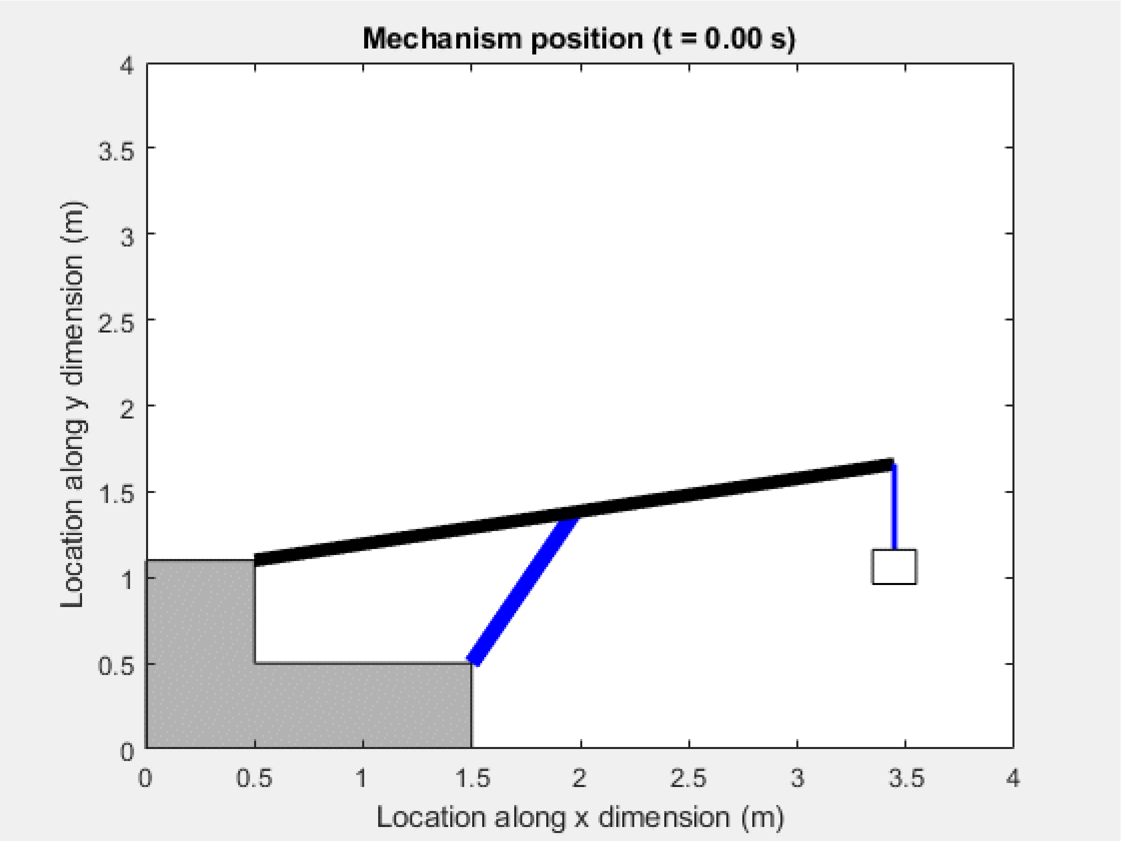
This figure shows the booming cylinder position, directional control valve position, and the flow rates through the counterbalance valve for the design cases 1 and 2 of the counterbalance valve. The booming cylinder extends in the negative position of the direction control valve and retracts in the positive position of the direction control valve.


Result
File
Tools
View
Simulation
Help
Booming Cylinder Position (m)
1
0.8
0.6
0.4
0.2
0
×10^4
6
5
4
3
2
1
0
Booming Cylinder A Chamber Piston Force (N)
Booming Cylinder B Chamber Piston Force (N)
0
5
10
15
20
Ready
T=24.000



Animation of Simscape Logging Results

This animation shows the change in the position of the booming cylinder and boom arm mechanism. The motion in the mechanism is based on the flows to the cap side and rod side of the piston chamber of the booming cylinder through the counterbalance valve.


Mechanism position (t = 0.00 s)
Location along y dimension (m)
Location along x dimension (m)
0
0.5
1
1.5
2
2.5
3
3.5
4

Residential Ground Source Heat Pump

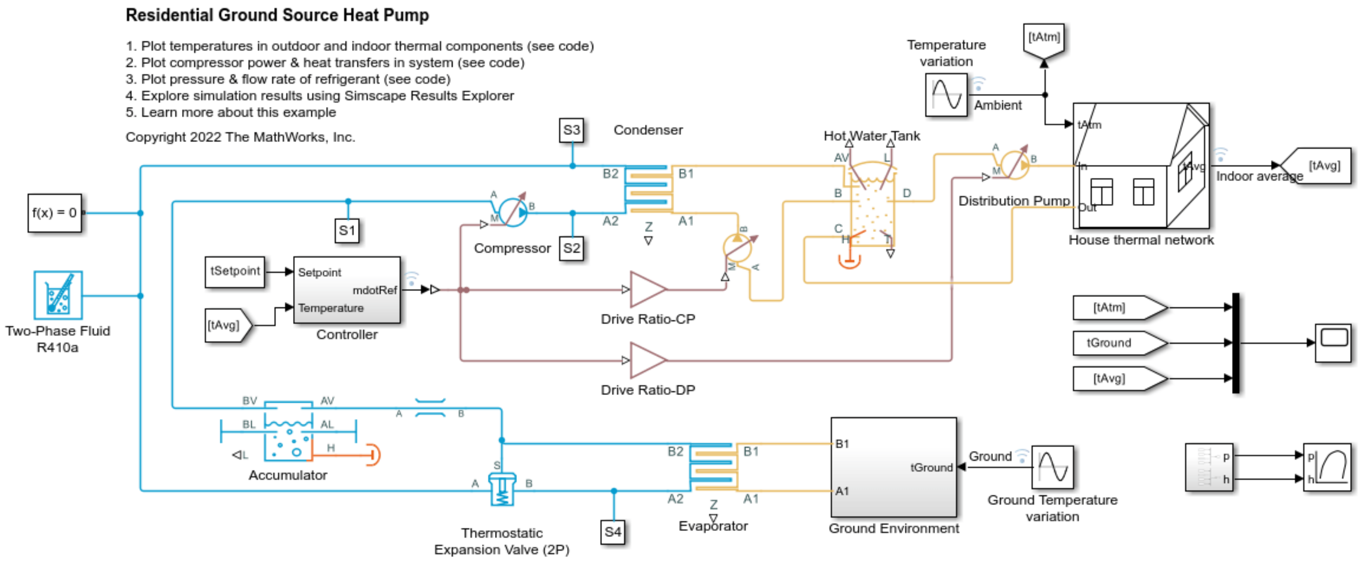
This example models a ground source heat pump system that is used to heat a residential building having hot-water radiators for heat distribution. The ground source heat pump uses R410a, a two-phase fluid refrigerant, as the working fluid. The heat pump takes up the naturally existing heat stored in the ground and transfers the heat to the hot-water radiators. The compressor drives the refrigerant through a condenser, a thermostatic expansion valve, and an evaporator. An accumulator ensures that only vapor returns to the compressor.

The ground acts as a heat source during winters. A series of connected pipes, known as ground loop, is buried horizontally in the ground at a shallow depth near the residential building. It circulates a mixture of water and antifreeze that absorbs heat from the surrounding soil. The antifreeze used in this example is Propylene glycol which is considered to be non-toxic. A ground loop pump drives the mixture of water and antifreeze through the header pipes, the ground loop pipes and the evaporator. A circulation pump drives water flow through the condenser. The hot water exiting the condenser is stored in a hot water storage tank. A distribution pump drives the hot water flow from the tank to the radiators. A PI controller modulates the refrigerant mass flow rate to keep the average air temperature of the building around the setpoint temperature of 23 degC, while the ambient temperature varies between -1 degC to 11 degC. The ground temperature varies between 8 degC to 12 degC.

The residential building has four radiators and four rooms. Each room exchanges heat with the environment through its exterior walls, roof, and windows. Each path is simulated as a combination of a thermal convection, thermal conduction, and the thermal mass. It is assumed that heat is not transferred internally between rooms. The initial indoor temperature of the rooms is equal to the ambient temperature of 5 degC and the initial ground temperature is equal to 10 degC. The heat pump quickly increases the average indoor temperature to the setpoint temperature of 23 degC.

This example can be used to keep the average air temperature of the building up to the setpoint temperature of 28 degC while the ambient temperature varies between -1 degC to 11 degC and the ground temperature varies between 8 degC to 12 degC.

Model

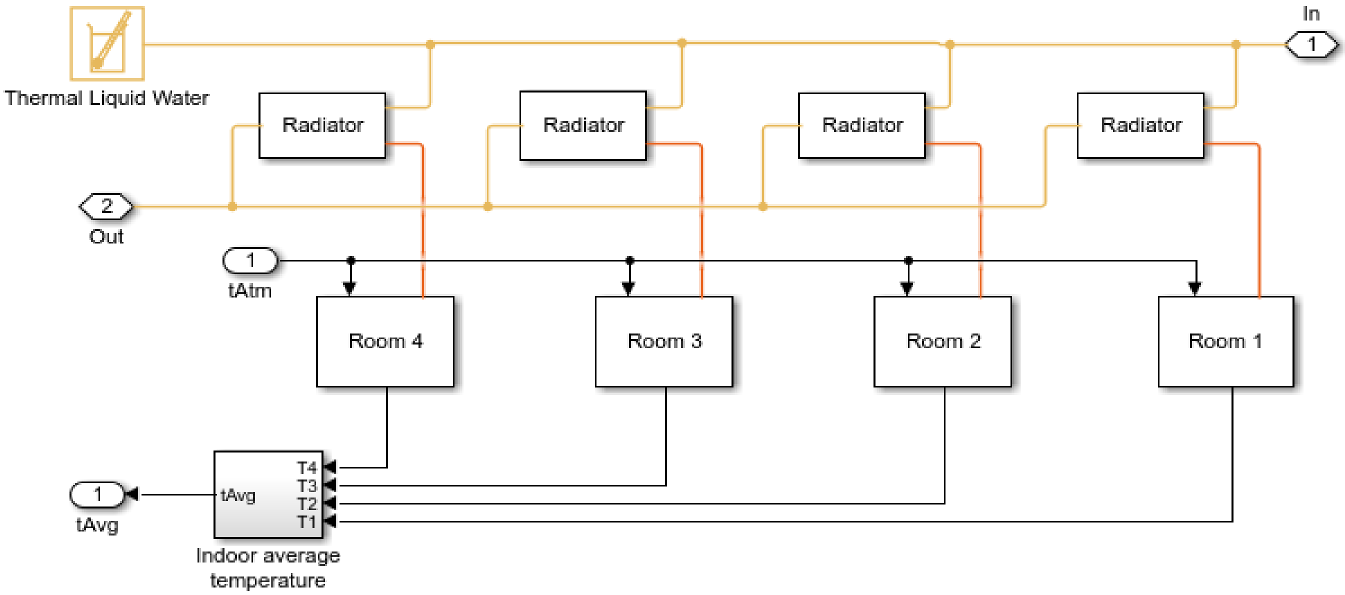


Residential Ground Source Heat Pump

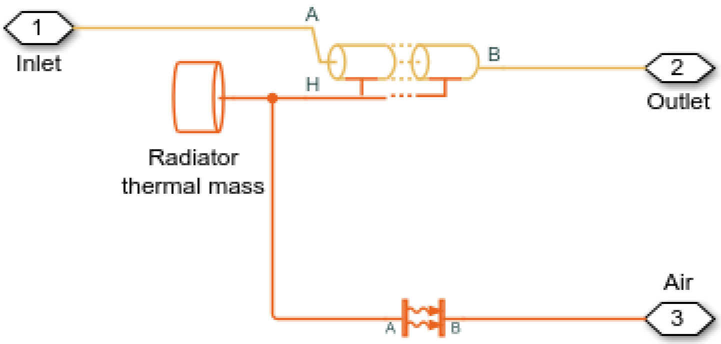
1. Plot temperatures in outdoor and indoor thermal components (see code)
2. Plot compressor power & heat transfers in system (see code)
3. Plot pressure & flow rate of refrigerant (see code)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example

Copyright 2022 The MathWorks, Inc.

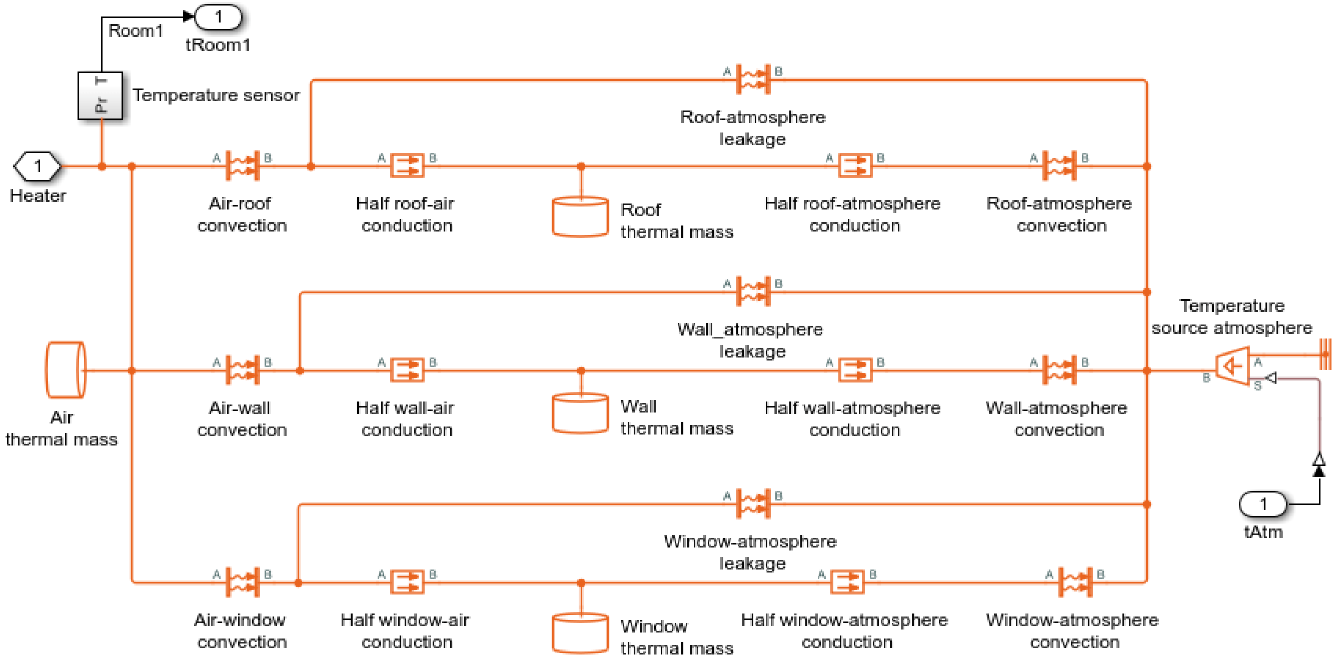
House Thermal Network Subsystem



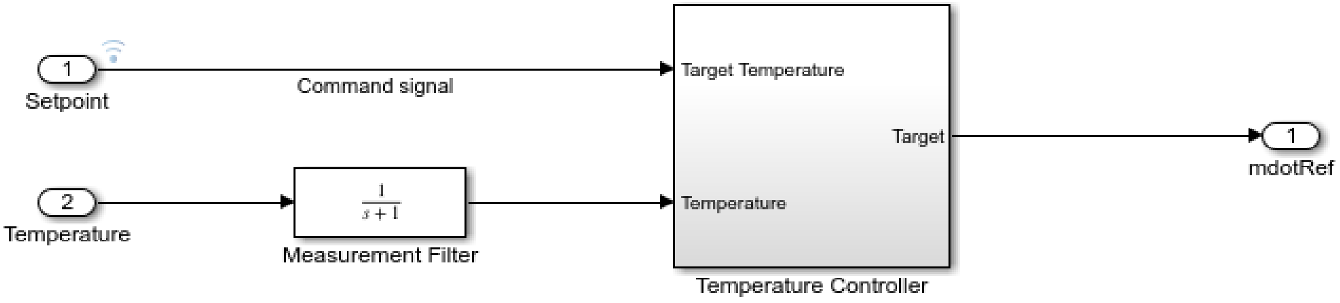
Radiator Subsystem



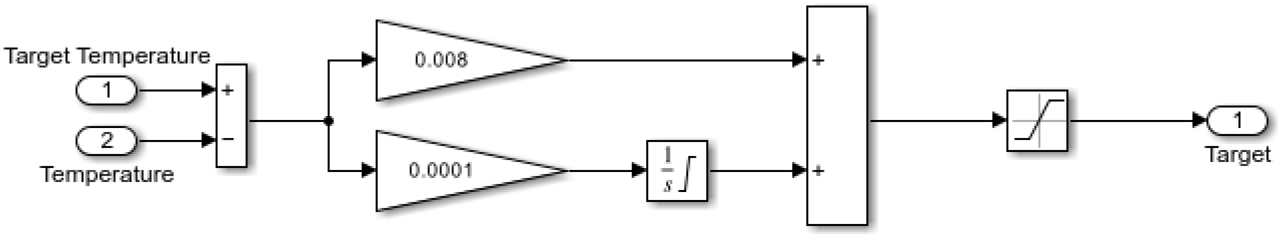
Room Subsystem



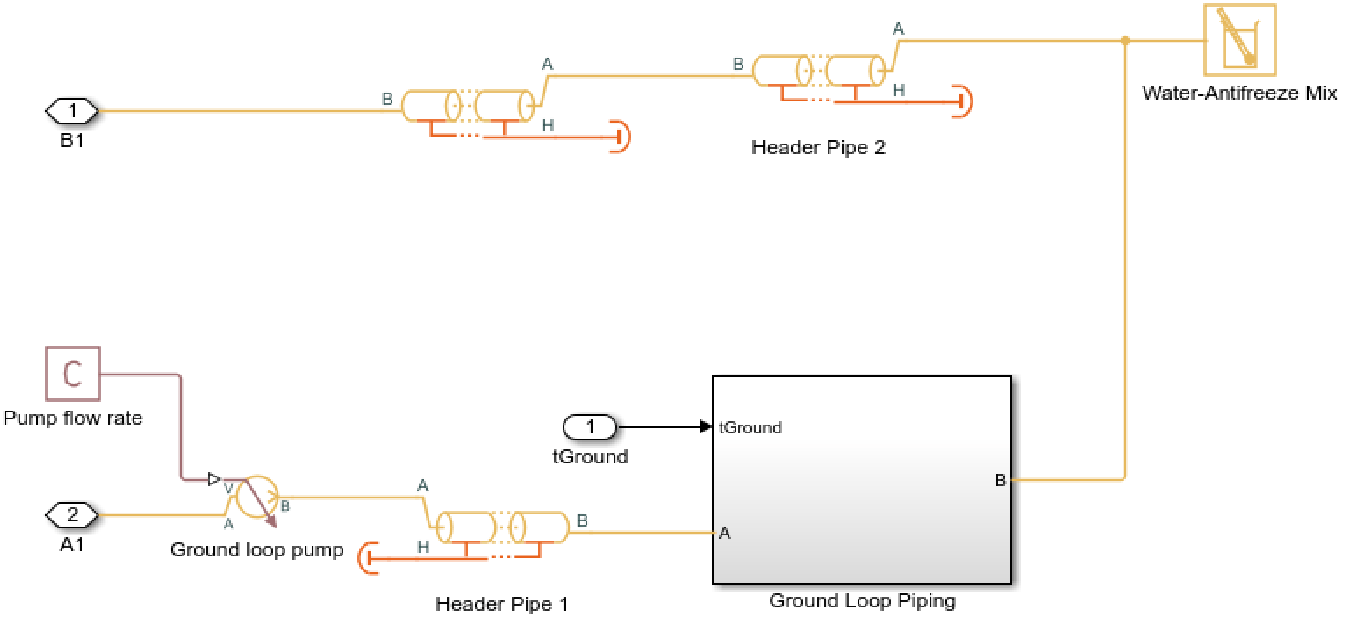
Controller Subsystem



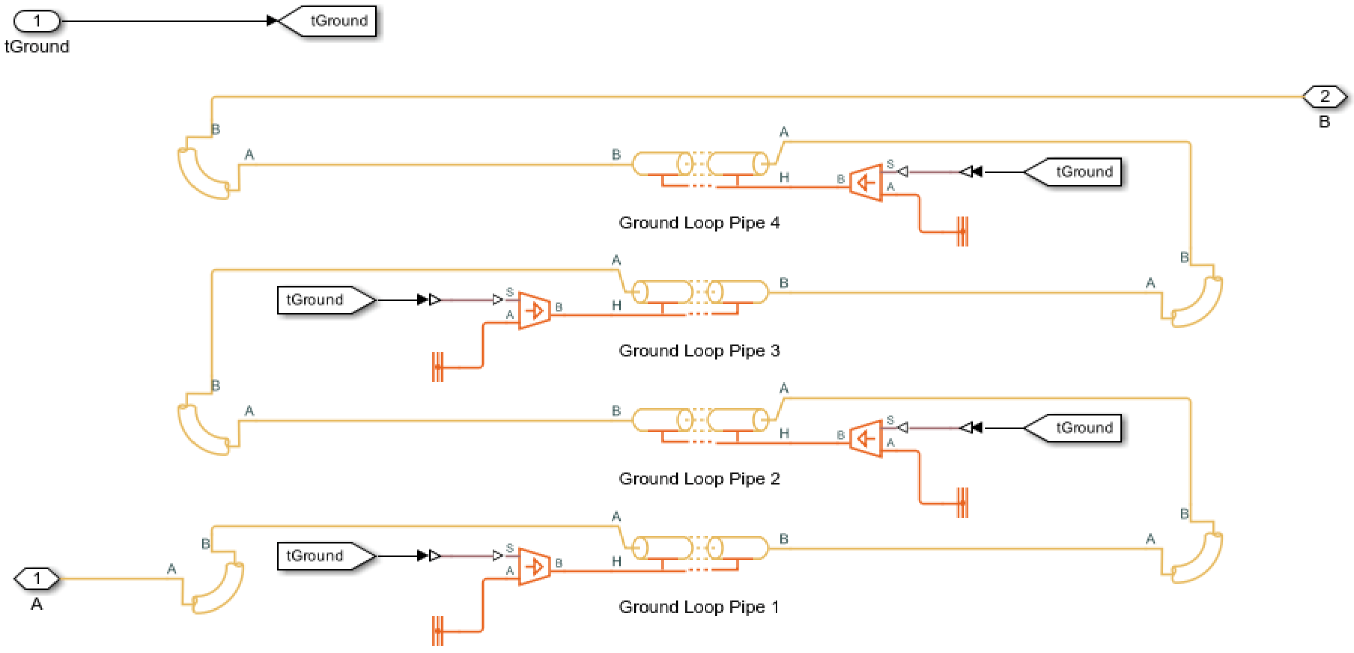
Temperature Controller Subsystem



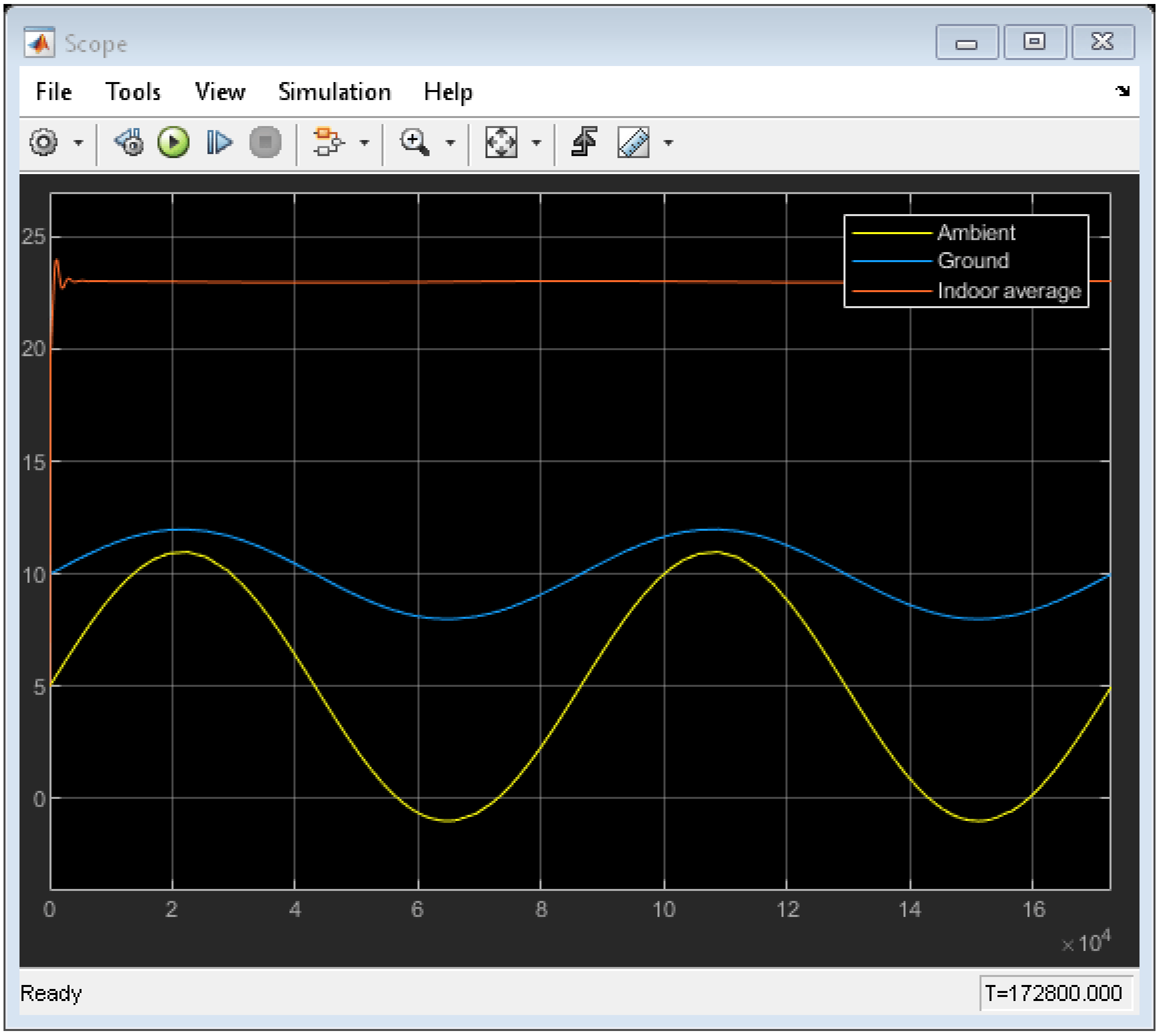
Ground Environment Subsystem



Ground Loop Piping Subsystem

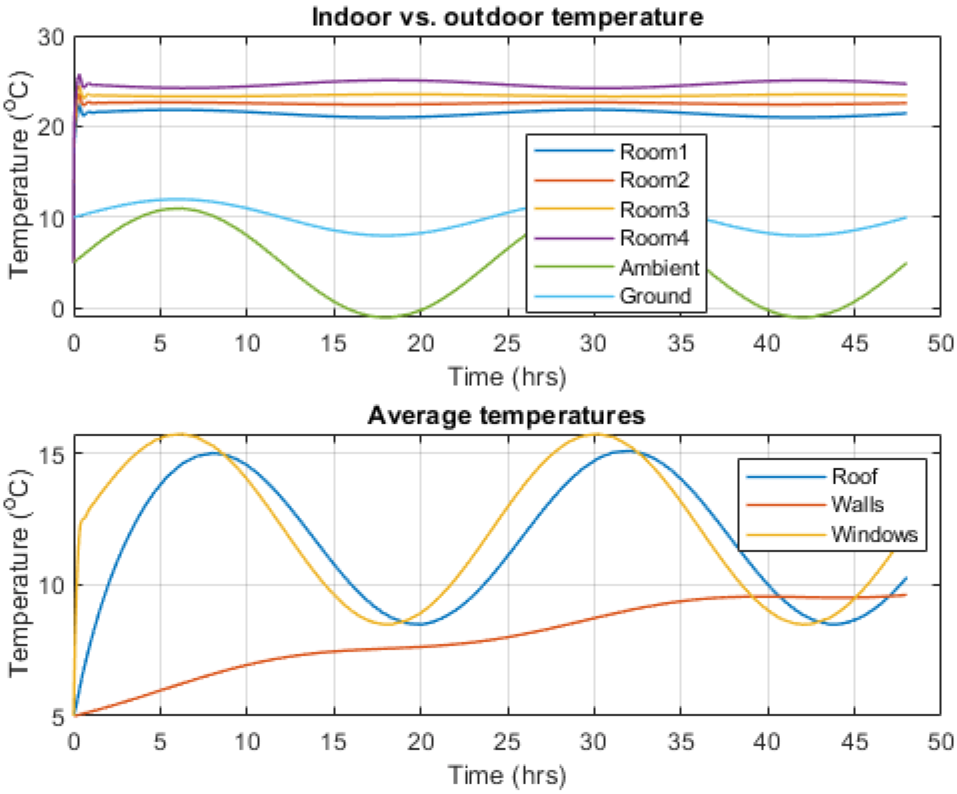


Simulation Results from Scopes

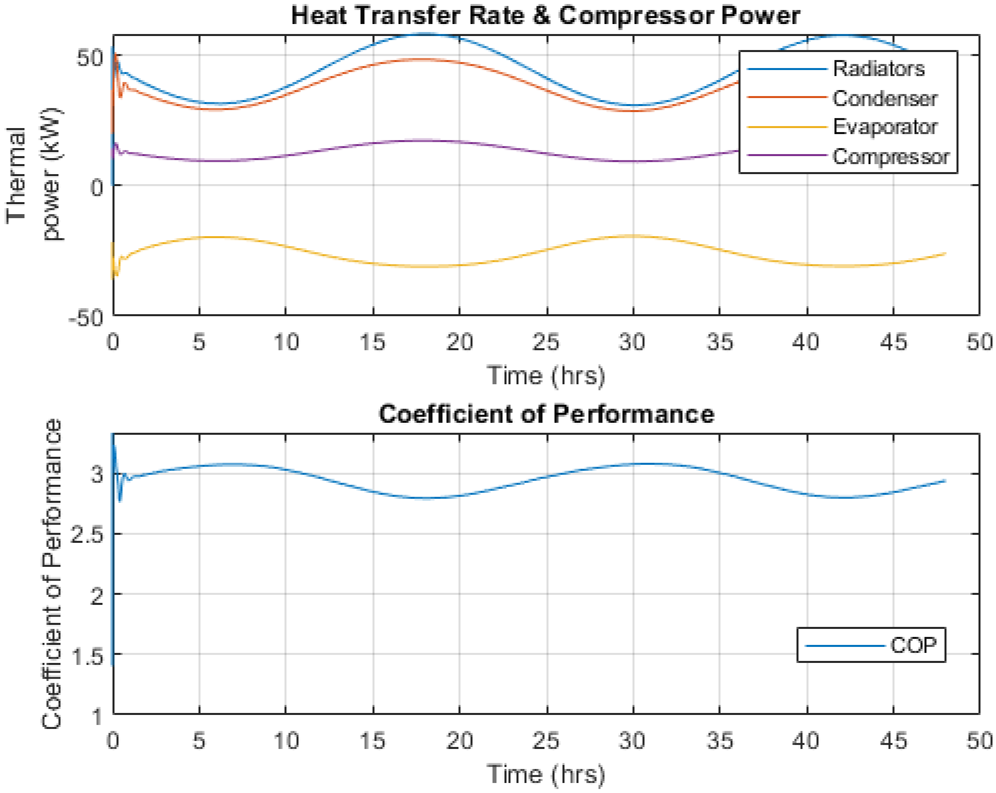


Simulation Results from Simscape Logging

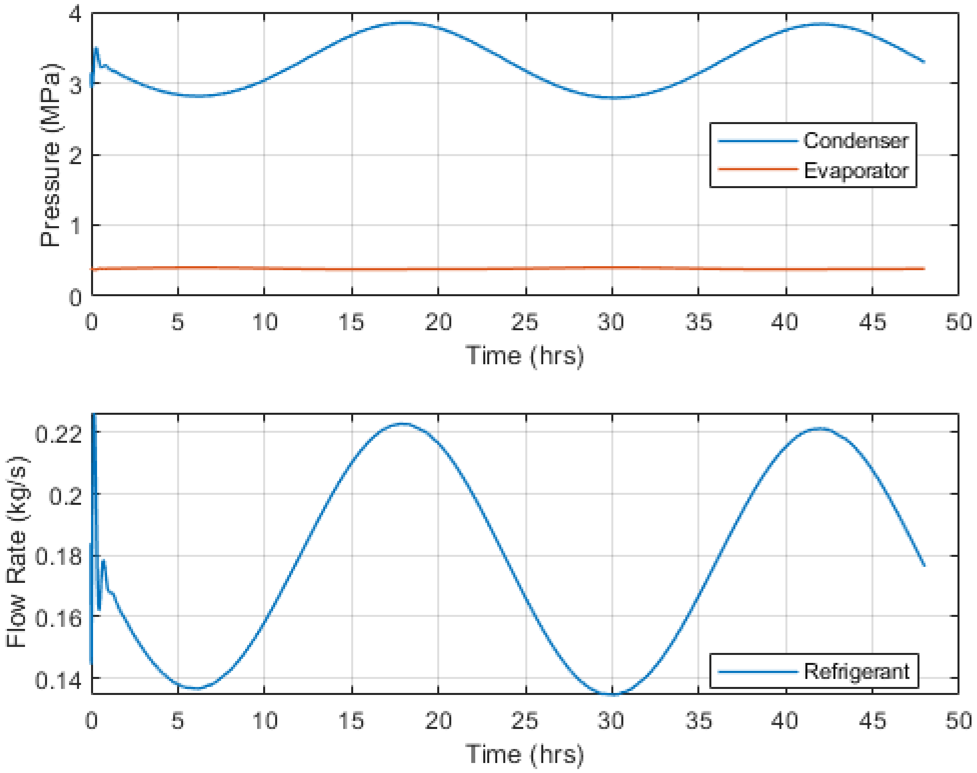
This model can be used to plot temperature variation of indoors and outdoors using sscfluids_residential_ground_source_heat_pump_plot1T script. This top subplot shows the temperature variation of each room compared to the outside air and ground temperature fluctuations. This bottom subplot shows the average overall temperatures of roof, walls, and windows. The initial indoor temperature of the rooms is equal to the ambient temperature of 5 degC, and the heat pump quickly increases the average indoor temperature to the setpoint temperature of 23 degC. The initial ground temperature is equal to 10 degC.



This model can be used to plot power consumption and heat load of different components of the heat pump using sscfluids_residential_ground_source_heat_pump_plot2Energy script. This top subplot shows the power consumed by the compressor, the heating load of the heat pump system, which is the rate of heat transfer in the condenser, and the rate of heat transfer in the evaporator. This bottom subplot shows coefficient of performance which is the ratio of the heating load and the power consumed. These plots show that more energy is being consumed when heat load is higher, i.e when ambient temperature is lower, and less energy is being consumed when heat load is lower, i.e when ambient temperature is higher, while the coefficient of performance stays nearly constant for the entire duration.



This model can be used to plot pressure and flow variation of the refrigerant using sscfluids_residential_ground_source_heat_pump_plot3pmdot script. This top subplot shows refrigerant pressure at the condenser and at the evaporator and this bottom subplot shows refrigerant mass flow rate. The higher pressures and flow rates correspond to lower ambient temperatures, when the heat pump needs to transfer more heat, while the lower pressure and flow rates correspond to higher ambient temperatures, when the heat pump needs to transfer less heat.


Pressure (MPa)
0
1
2
3
4
Condenser
Evaporator
Time (hrs)
0
5
10
15
20
25
30
35
40
45
50
Flow Rate (kg/s)
0.14
0.16
0.18
0.2
0.22
Refrigerant
Time (hrs)
0
5
10
15
20
25
30
35
40
45
50

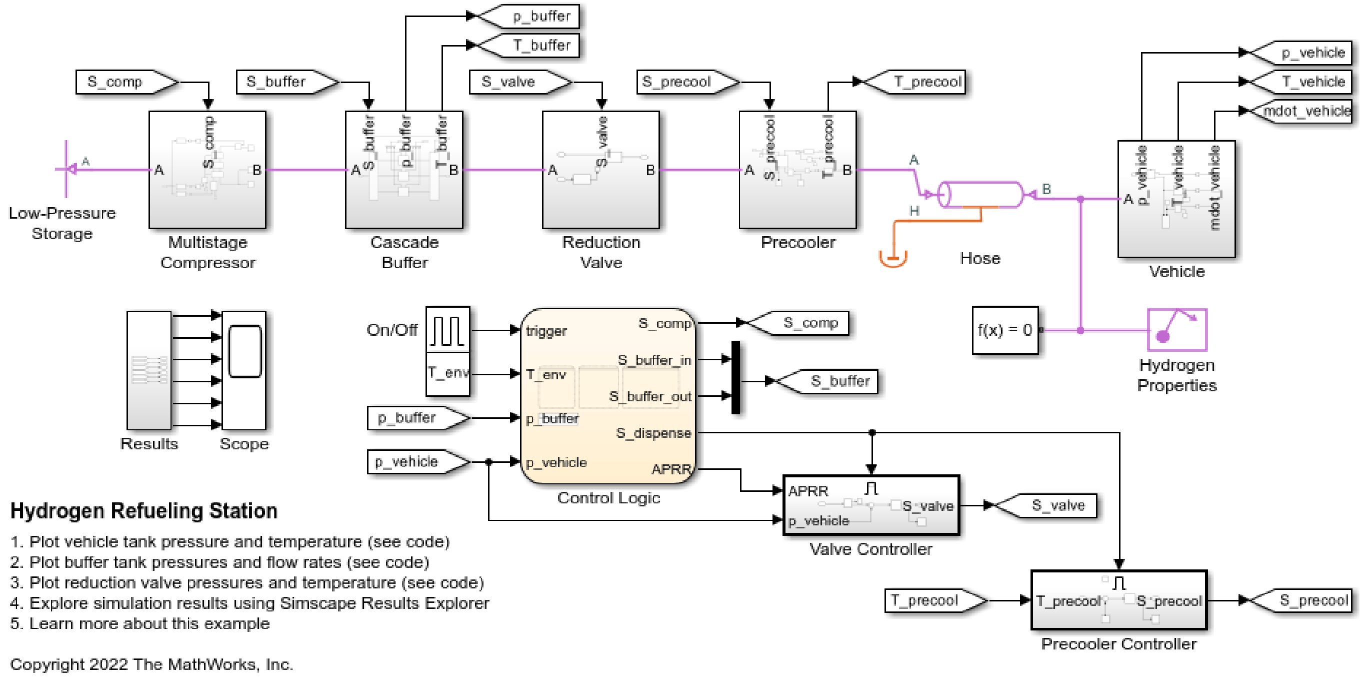
Hydrogen Refueling Station

This example models a hydrogen refueling station. Hydrogen is stored in low-pressure storage tanks at 200 bar at the station. A 3-stage intercooled compressor maintains the necessary pressure in a cascade buffer storage system so that the station is ready to dispatch hydrogen to any connected vehicles. The buffer is divided into high-pressure tanks at 950 bar, medium-pressure tanks at 650 bar, and low-pressure tanks at 450 bar. To avoid wasting compression energy, the lowest pressure buffer that is greater than the vehicle tank pressure is used to dispatch hydrogen. Priority valves switches between the different buffer tanks to control which buffer tanks to fill and discharge from.

When dispensing, a reduction valve controls the flow between the cascade buffer storage and the vehicle tank. A precooler chills the hydrogen before it is dispatched to the vehicle to avoid excessive temperature rise in the vehicle tank. This example is modeled after a type A70 refueling station from the SAE J2601 fueling protocols, which means that hydrogen is delivered at -40 degC and up to 700 bar.

A Stateflow® chart is used to model the logic needed to control the operations of the station. It determines when to turn on the compressor to recharge the buffer, when to dispatch hydrogen to the vehicle, and the switching logic of the cascade buffer priority valves. When dispensing, the Valve Controller and the Precooler Controller subsystems are enabled. They contain PI controllers that maintain the flow rate and temperature of the hydrogen fuel being dispatched.

Model

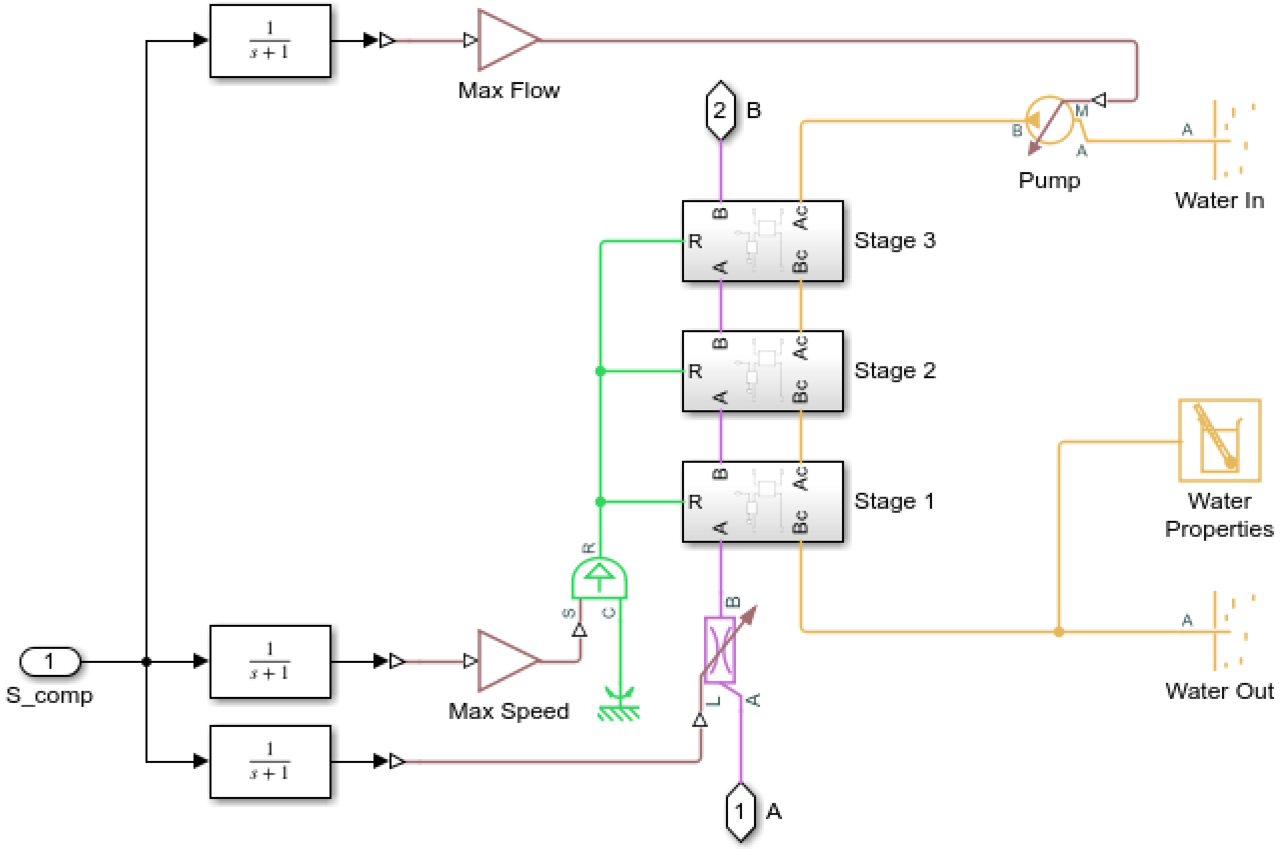


Hydrogen Refueling Station

1. Plot vehicle tank pressure and temperature (see code)
2. Plot buffer tank pressures and flow rates (see code)
3. Plot reduction valve pressures and temperature (see code)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example

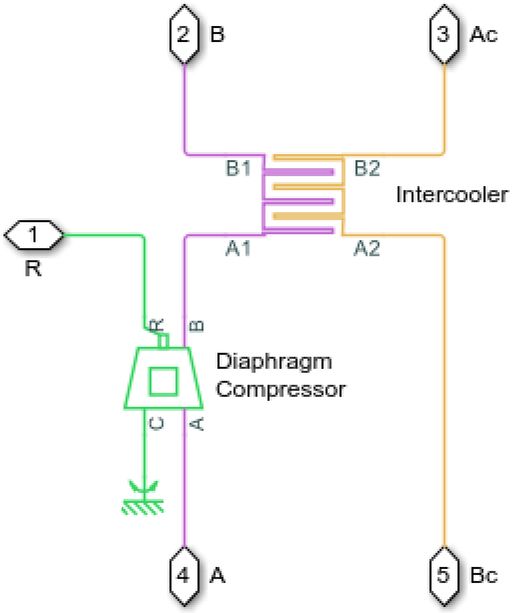
Multistage Compressor Subsystem

The compressor is comprised of three positive-displacement compressors with intercooler. The cooling water is assumed to be maintained near environment temperature by a radiator or another cooling system (not modeled).

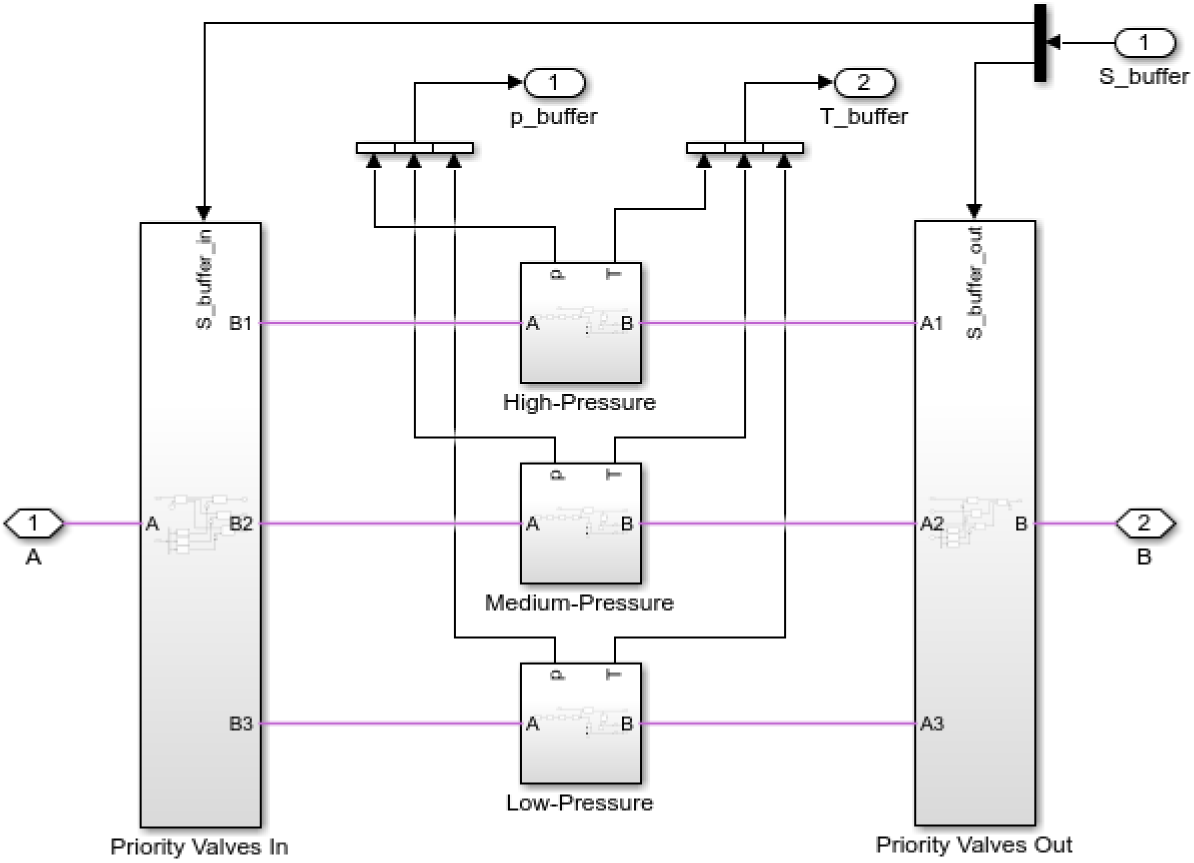


Stage 1 Subsystem

Stages 2 and 3 have identical components, but the compressor displacement is reduced.

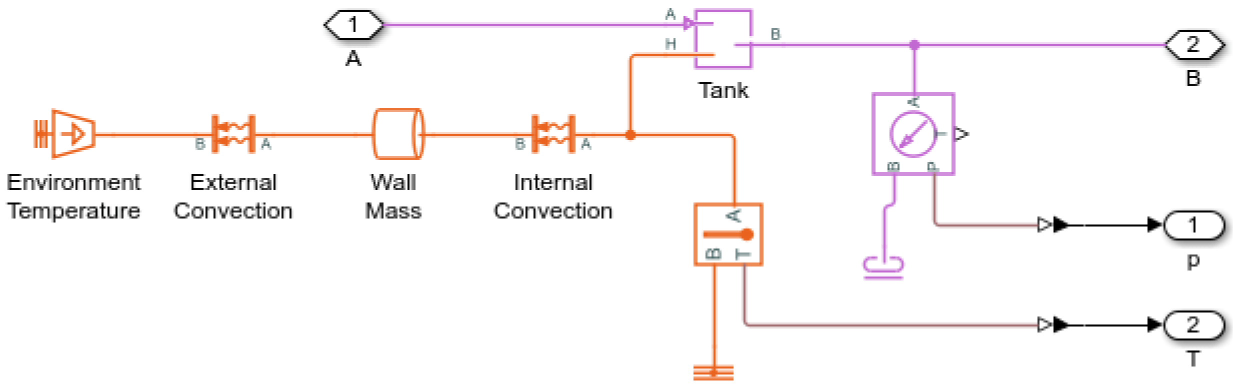


Cascade Buffer Subsystem

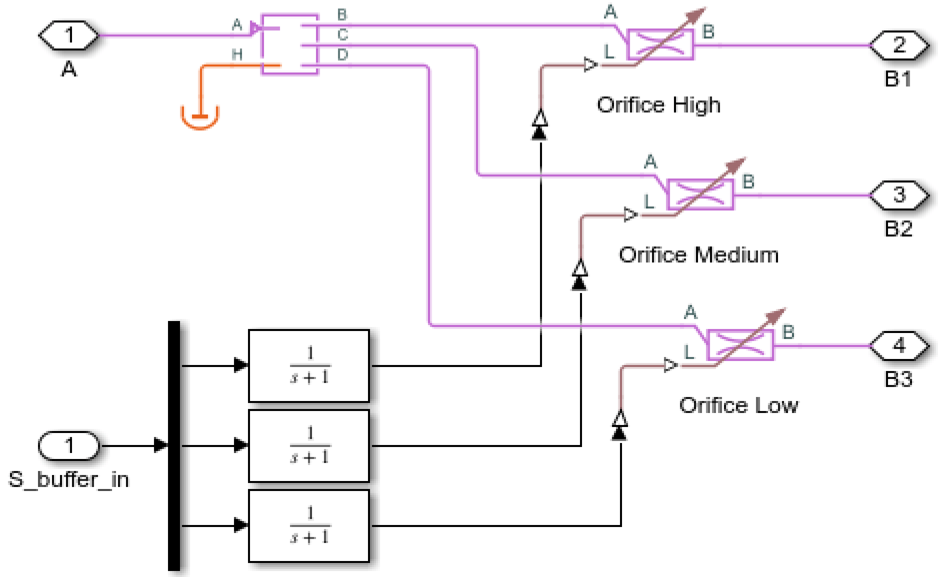


High-Pressure Subsystem

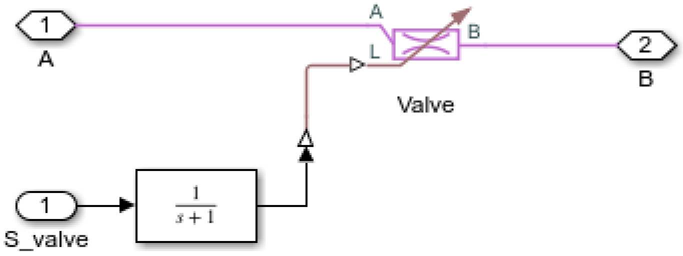
The medium-pressure and low-pressure buffer tanks are identical to the high-pressure buffer tank.



Priority Valves In Subsystem

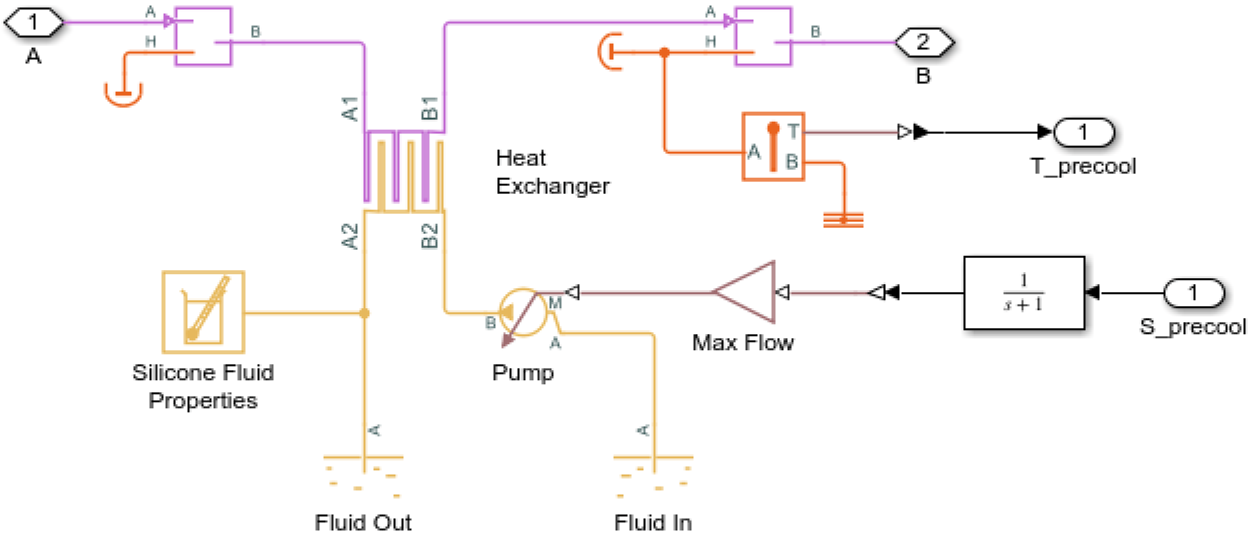


Reduction Valve Subsystem

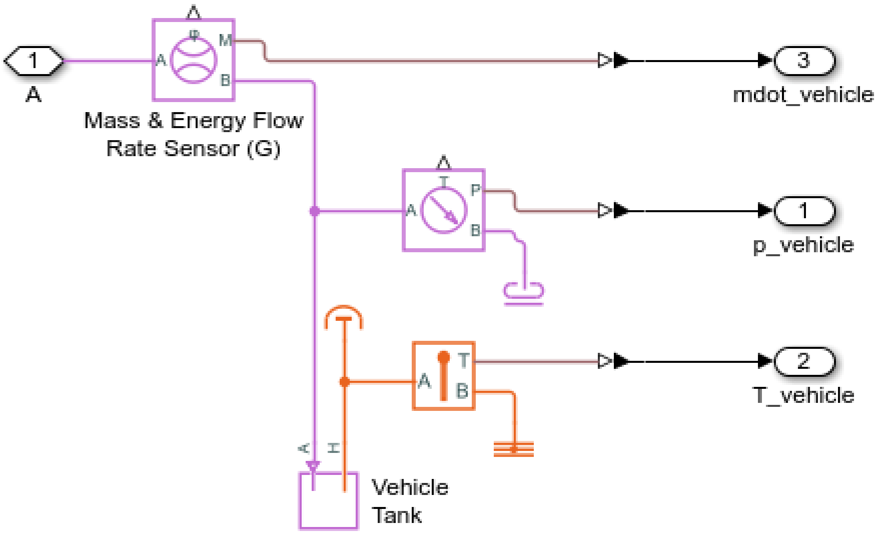


Precooler Subsystem

A silicone heat transfer fluid is used to cool the hydrogen before it is dispatched into the vehicle. The heat transfer fluid is assumed to be maintained at -60 degC by a chiller (not modeled).



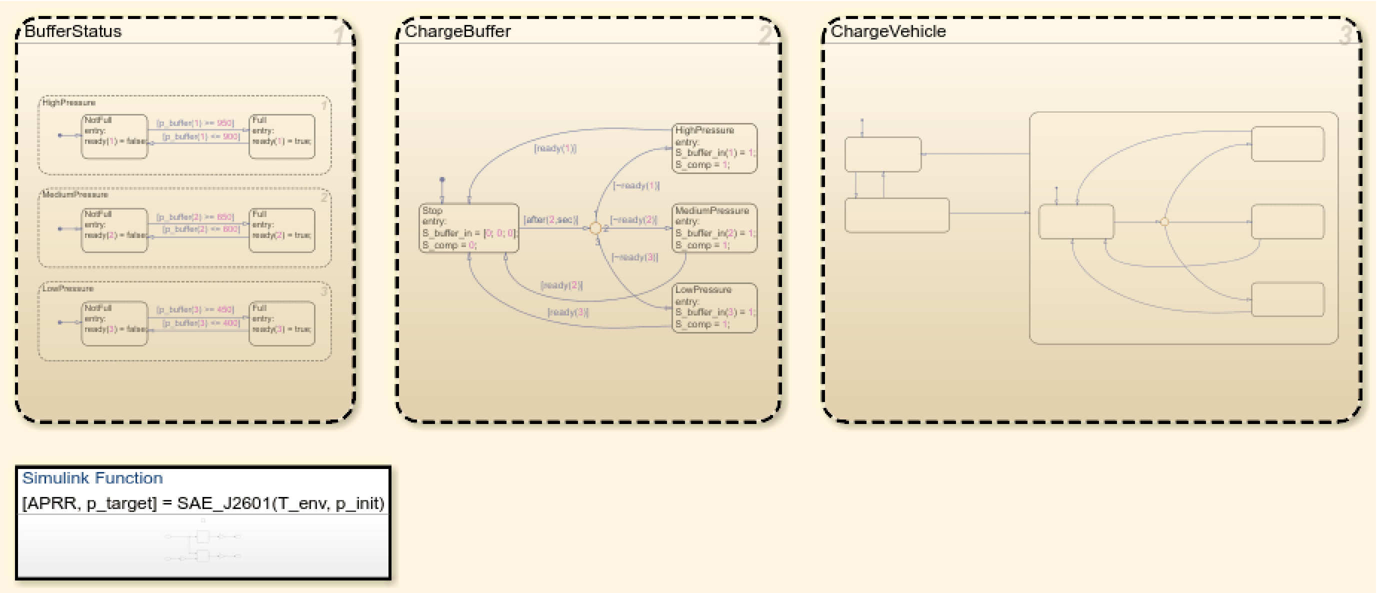
Vehicle Subsystem



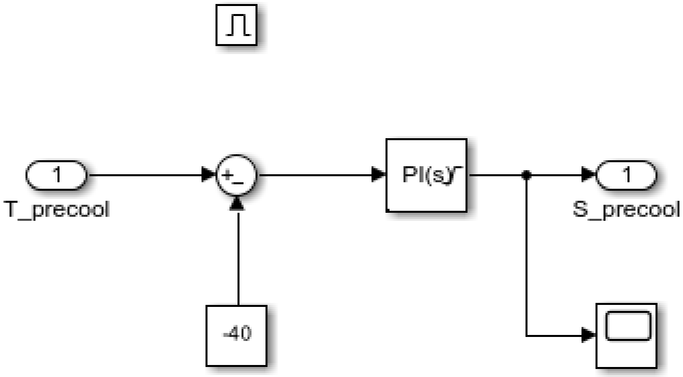
Control Logic Subsystem

The Stateflow chart is divided into 3 subcharts. The BufferStatus subchart reports the readiness of high-pressure, medium-pressure, and low-pressure buffer tanks depending on whether they have sufficient pressure for dispatching. The ChargeBuffer subcharts turns on the compressor to refill the buffer whenever it is not ready. It prioritizes filling the higher pressure tank over the lower pressure tank.

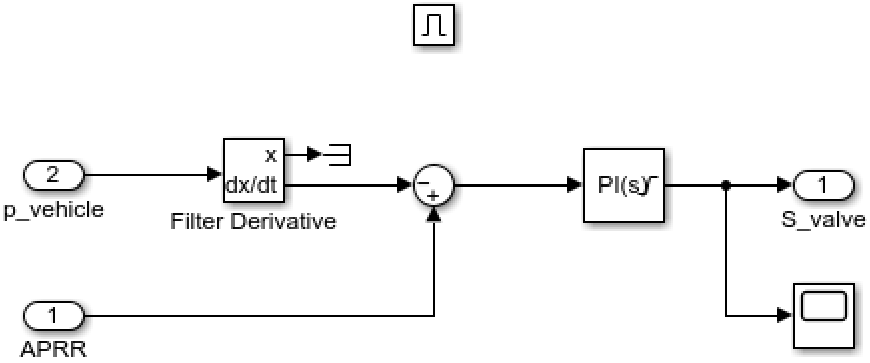
When the dispenser trigger is on, the Simulink Function is called to obtain the Average Pressure Ramp Rate (APRR) and the target pressure based on the tabulated data from SAE J2601. The ChargeVehicle subchart then turns on the reduction valve controller and the precooler to dispense hydrogen. It prioritizes dispatching from the lower pressure buffer tank before switching to the higher pressure buffer tank.



Precooler Controller Subsystem



Valve Controller Subsystem



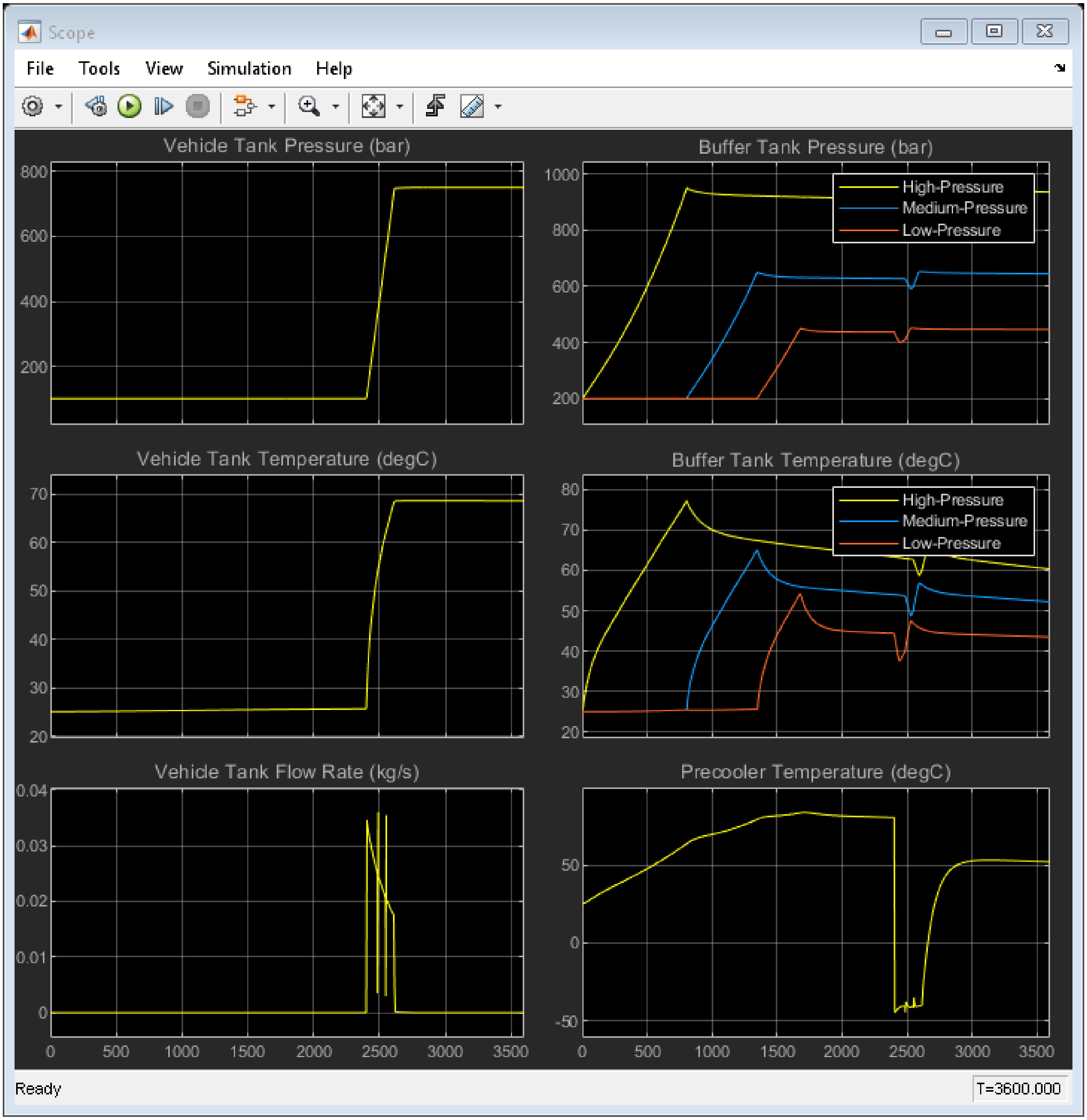
Simulation Results from Scopes

The state of the hydrogen fuel in the vehicle tank is shown in the left column of the scope plots. At t = 2400 s, the dispenser is triggered and filling starts. The increase in pressure is maintained at a constant slope, called the Average Pressure Ramp Rate (APRR), by the Valve Controller. Filling stops once the target pressure is reached. The APRR and target pressure come from the SAE J2601 fueling protocols and depend on the environment temperature and initial pressure in the vehicle. In this example, they are 18.5 MPa/min and 74.5 MPa, respectively.

While filling, the temperature increases significantly due to the amount of compression that occurs in the vehicle tank. SAE J2601 sets a maximum temperature limit of 85 degC. In this example, the temperature reaches 68.6 degC when full. This is achieved by precooling the hydrogen to -40 degC before it enters the vehicle, as shown in the bottom right plot.

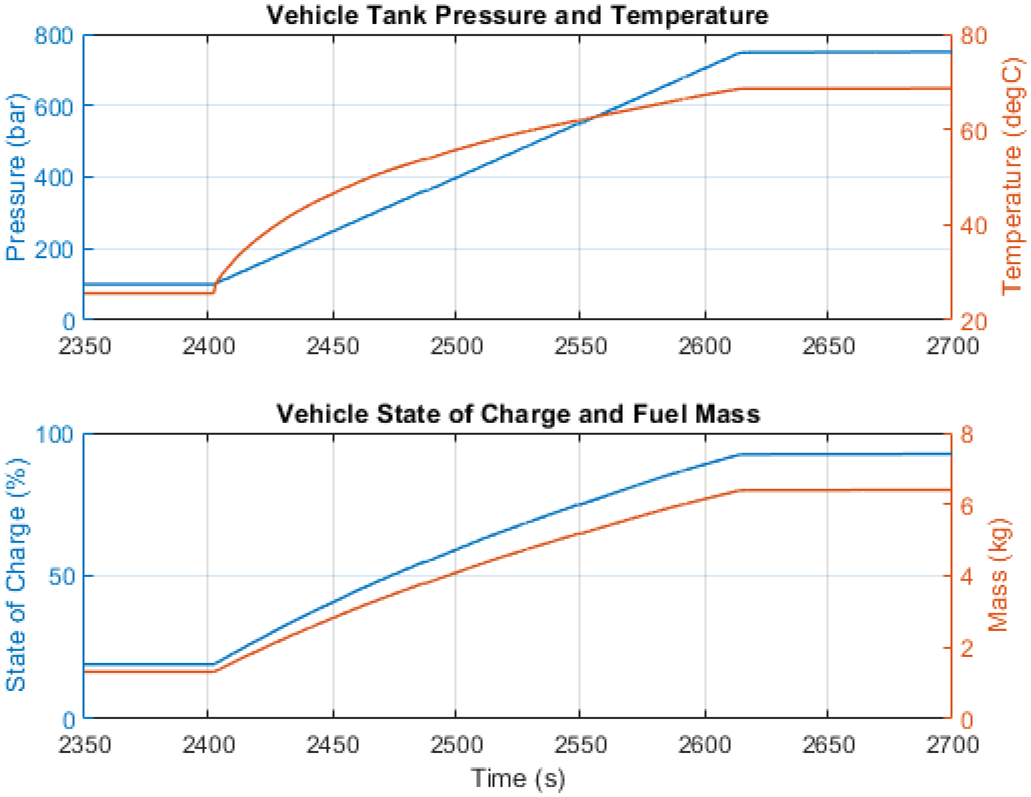
The bottom left plots shows the flow rate of hydrogen into the vehicle tank. Because the controller maintains an APRR, the mass flow rate decreases as the pressure increases. The two spikes in the flow rate occur when it switches from the low-pressure buffer to the medium-pressure buffer and from the medium-pressure buffer to the high-pressure buffer. Further adjustments of the valve timing while switching buffers may reduce these spikes. Nevertheless, the flow rate is below the maximum limit of 0.06 kg/s set by SAE J2601.

The top two plots on the right column shows the state of the cascade buffer storage. At the start of the simulation, the compressor is turned on to fill the buffer in order from the high-pressure tank to the low-pressure tank. Dispatching hydrogen to the vehicle causes a drop in the buffer pressure. Therefore, the compressor turns on again to refill the buffer tanks and maintain buffer pressure.



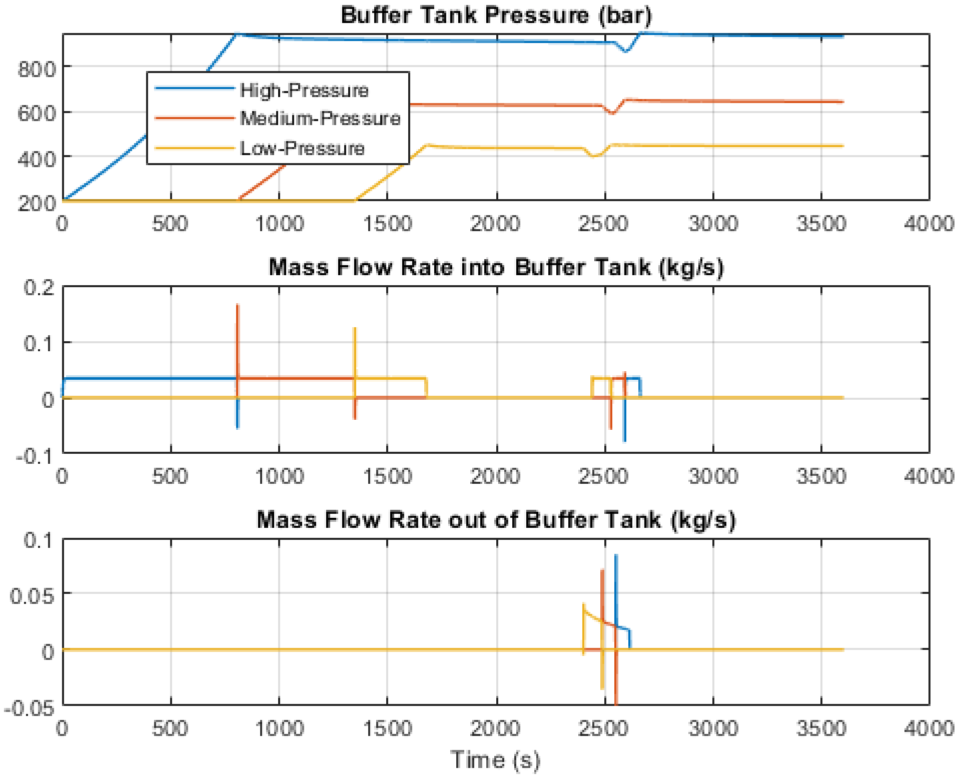
Simulation Results from Simscape Logging

This plot shows a closer view of the state of the hydrogen fuel in the vehicle tank while it is being filled. It also shows the state of charge (SOC) and the total mass of hydrogen in the vehicle. The SOC is defined as the density of hydrogen in the vehicle divided by the density of hydrogen at nominal working pressure (NWP) and 15 degC. The SOC must never exceed 100%.



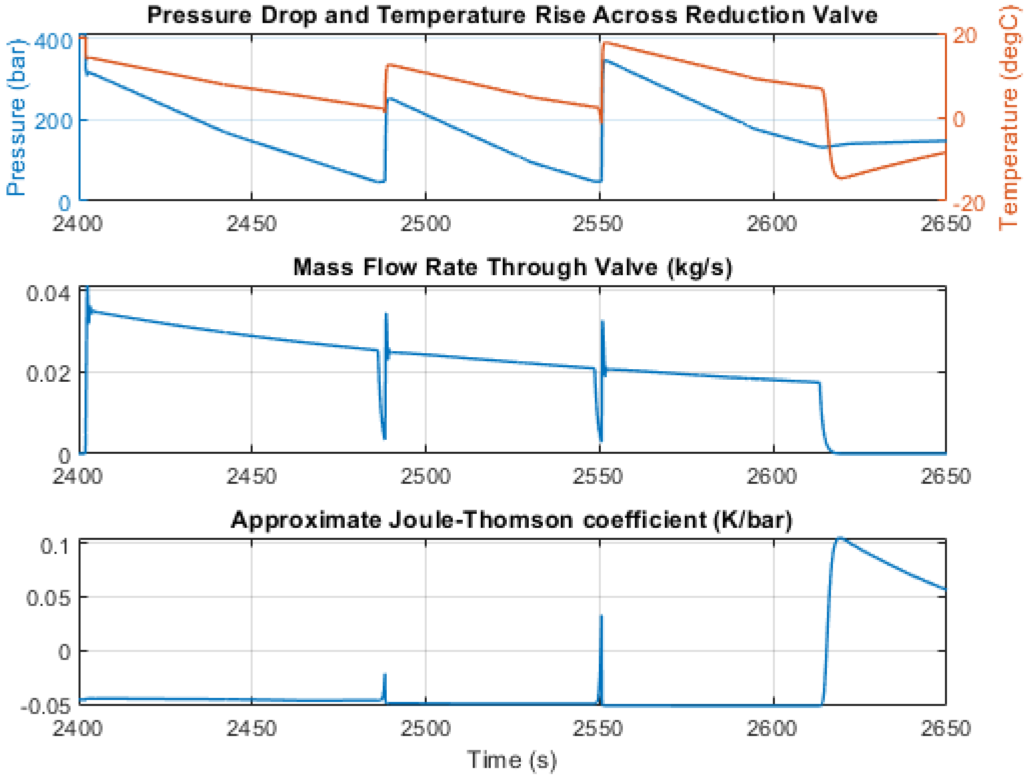
This plot shows the pressures and flow rates in the cascade buffer storage tanks. At the start of the simulation, the high-pressure tank, medium-pressure tank, and low-pressure tank are filled at a rate of about 0.034 kg/s. While switching from one buffer tank to another, some hydrogen from the higher pressure tank leaks to the lower pressure tank, resulting in a spike in flow rate. This can be reduced by adjusting the timing of the valve switching.

At t = 2400 s, hydrogen is dispatched to the vehicle at a rate that varies from 0.035 kg/s to 0.018 kg/s. The hydrogen initially comes from the low-pressure buffer tank. As the vehicle tank pressure increases, it switches to the medium-pressure buffer tank, and then finally the high-pressure buffer tank. As the vehicle tank is filled, the buffer pressure drops, so it needs to be refilled for the next vehicle. A larger buffer will allow more vehicles to be filled before the buffer needs to be refiled.



This plot shows pressure drop and temperature rise across the reduction valve while the vehicle is being filled. Because the buffer is divided into high-pressure, medium-pressure, and low-pressure tanks, the pressure drop curve is divided into three segments and the maximum pressure drop seen by the valve is reduced.

The temperature increases across the valve because the hydrogen is above the Joule-Thomson inversion temperature. The bottom plot approximates the Joule-Thomson coefficient by taking the ratio of the temperature difference to the pressure difference across the valve. The coefficient is negative, meaning that the gas is warmed by the pressure drop.


Pressure Drop and Temperature Rise Across Reduction Valve
Pressure (bar)
400
200
0
Temperature (degC)
20
0
-20
2400
2450
2500
2550
2600
2650
Mass Flow Rate Through Valve (kg/s)
0.04
0.02
0
2400
2450
2500
2550
2600
2650
Approximate Joule-Thomson coefficient (K/bar)
0.1
0.05
0
-0.05
2400
2450
2500
2550
2600
2650
Time (s)

Vehicle HVAC System

This example models the heating and cooling system of a passenger car. The cabin is represented as a volume of moist air exchanging heat with the external environment. The blower drives moist air through the evaporator, blend door, and heater core before returning to the cabin. The blend door controls the amount of air flow through the heater core. The recirculation door controls whether air is brought in from the external environment or from within the cabin.

The evaporator and heater core are part of the refrigerant loop and coolant loop, respectively. The refrigerant loop is a typical R-134a vapor-compression refrigeration cycle with a compressor, a condenser, a liquid receiver, a thermostatic expansion valve, and an evaporator. The P-H Diagram block shows the thermodynamic states of the cycle during simulation.

The coolant loop consists of an engine-driven pump which drives coolant through the coolant jacket to absorb heat and then through the radiator and/or the heater core to reject heat. The thermostat controls the amount of flow through the radiator to maintain the desired engine temperature. The heater control valve controls the amount of flow through the heater core to maintain the desired cabin heat. The expansion tank provides the volume for the coolant to thermally expand and contract.

The refrigerant loop and coolant loop are modeled in separate files via subsystem reference blocks. This makes it easier for different users to work on the refrigerant loop, the coolant loop, and the overall vehicle HVAC model independently. The refrigerant loop and coolant loop referenced subsystems include their own test harness models (requires Simulink Test™) so that they can be simulated independently of the main vehicle HVAC model. For more information, see “Subsystem Reference”.

The vehicle HVAC model can be simulated in two modes: predefined inputs or manual inputs. When using predefined inputs, the HVAC settings and environment conditions are specified by Simulink® source blocks. For the first 1800 s, the environment temperature is 30 degC and air conditioning is on. For the next 1800 s, the environment temperature is reduced to 10 degC, the air conditioning turns off, and the heater turns on. When using manual inputs, the HVAC settings and environment conditions can be adjusted at run time using the dashboard controls.

Model

Manual Inputs

Cabin Subsystem

Heat Transfer Subsystem

Passengers Subsystem

Coolant Loop Subsystem

Engine Block Subsystem

Intake Subsystem

Refrigerant Loop Subsystem

Simulation Results from Scopes

Simulation Results from Simscape Logging

This figure shows a psychrometric chart for the moist air inside the cabin. It is based on the ASHRAE Psychrometric Chart No. 1 and is used to convey thermodynamic properties of moist air, such as dry-bulb temperature, web-bulb temperature, relative humidity, humidity ratio, and enthalpy. When using predefined inputs, the trajectory of the state of the cabin moist air during simulation is plotted on the chart. When using manual inputs, the state of the cabin moist air is continuously updated during run-time.

This plot shows rate of condensation in the evaporator when the air conditioning is turned on.

This plot shows the contributions to power consumption in the vehicle HVAC system, as well as the coefficient of performance of the refrigeration loop. Note that the air conditioning is turned off at 1800 seconds.

Cylinder Design Parameter Sensitivity Analysis

This example shows how to perform a sensitivity analysis for tandem primary cylinder model in Simscape™. The example requires the Simulink Design Optimization™ to be installed and licensed to run.

Introduction

Sensitivity Analysis: Sensitivity analysis can be used to study how the performance of a system depends on key design parameters. A few of the reasons to do this include:

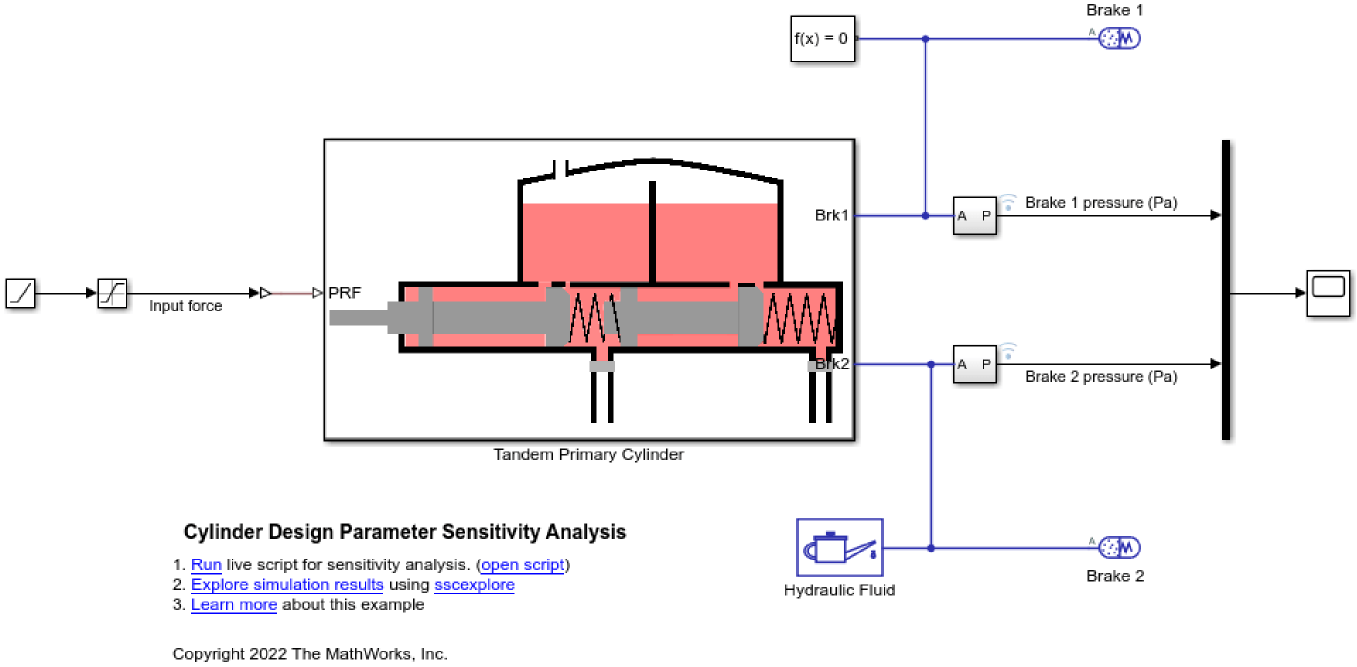
1. To identify the design parameters that have significant impact on the model behavior.
2. To determine if the system displays the expected dependency on input parameters.
3. To determine permissible working range for key design parameters while still delivering the required system behavior and/or performance.
4. To identify if there are common sets of design parameters that influence system behavior in a similar way.
5. To determine requisite manufacturing tolerances to ensure system performance requirements are always met.

There are many approaches to sensitivity analysis including correlation and regression methods which are applied here to a tandem primary cylinder for use in automotive braking systems. In this example you will see functions from Sensitivity Analyzer app are used to perform this analysis.

Tandem Primary Cylinder: The tandem primary cylinder converts the applied mechanical force to the hydraulic force which is transferred to the wheels through the two pressure lines. For more information, see the Tandem Primary Cylinder example in MATLAB.

Procedure for Sensitivity Analysis

Decide parameters for Sensitivity analysis: A change in a design parameter or application of a tolerance to a nominal design parameter value will affect the output of the system. There is a subset of parameters for which the output is more strongly dependent, and the first step is identify this subset via a sensitivity analysis.



Cylinder Design Parameter Sensitivity Analysis

1. Run live script for sensitivity analysis. (open script)
2. Explore simulation results using sscexplore
3. Learn more about this example

For the sensitivity analysis we will consider following parameters:

1 Spring 1 stiffness coefficient. (tpcStiffness1Coeff)

2 Spring 2 stiffness coefficient. (tpcStiffness2Coeff)

3 Spring 1 initial deformation. (tpcSpring1Comp)

4 Spring 2 initial deformation. (tpcSpring2Comp)

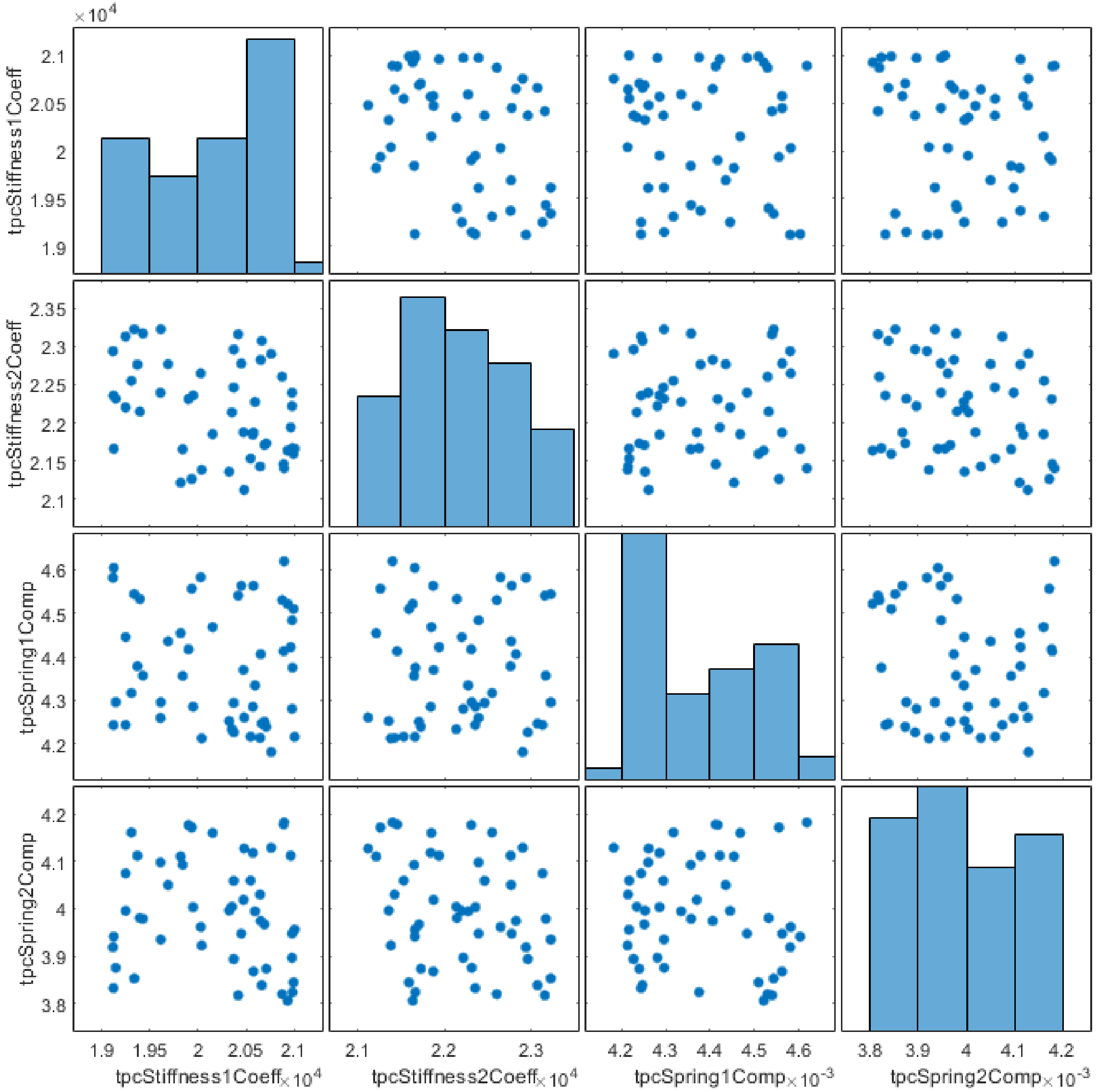
Determine the range of the parameters: The variation in a property of an assembled part is a function of the manufacturing tolerances achieved. For example, the variation in the wire diameter of the spring will result in variation in the spring constant. For this example, we will assume 10% variation in the parameters listed above. It is assumed that the variation follows the uniform distribution.

We will be using Sensitivity Analyzer app to perform the sensitivity analysis.

Collect these *param.Continuous* object instances to a vector.

Sample the parameter space: In this section we generate 50 parameter combinations that span the parameter space. To generate the parameter combination instances the code below can be used. You can use higher number of samples for higher confidence.

You can observe the generated values using a scatter plot.



Define a system performance metric: The next step is to define requirements function that returns a metric of system performance. This metric can then be used to determine if the system is compliant with requirements, or not, for a given set of design parameter values. The pressure generated at Brake Pad 1 and Brake Pad 2 is a good metric for the efficacy of a tandem primary cylinder and hence these two signals are selected.

Next we create the experiment object and specify the input data.

Associate brake pressure 1 data with model output.

Associate brake pressure 2 data with model output.

Add brake pressure data to the sdo.experiment object.

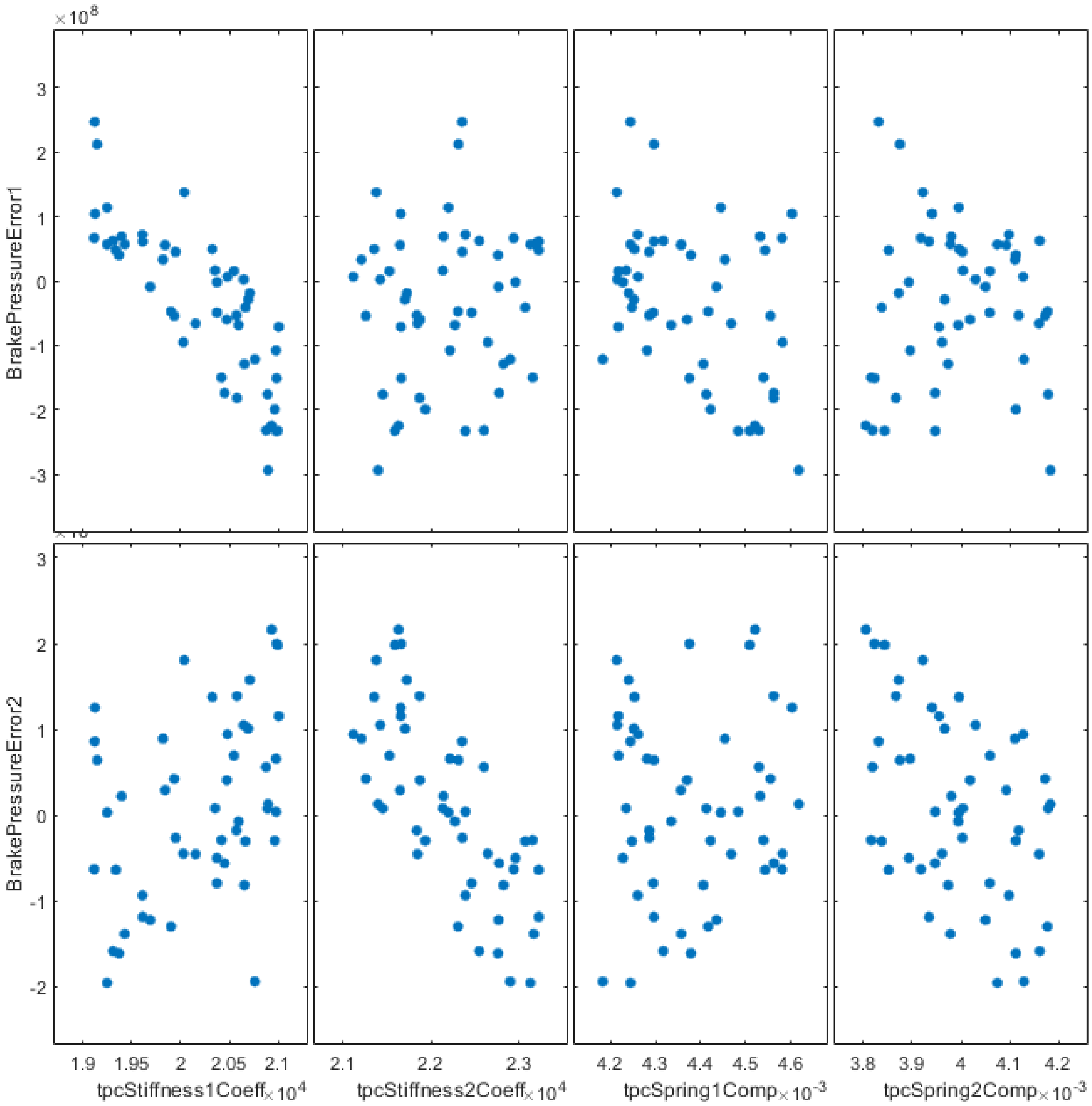
Setup the Objective function: There are many options available in the Sensitivity Analyzer app depending on the objective of the analysis. Here a custom objective function is created. An alternative would be to use the Signal tracking objective from the app. The objective function compares the simulated output to the baseline provided to compute the cumulative error. In general, the objective function for the design parameter sensitivity analysis must be matched to a metric associated with the design functional requirement. Here the assumption is that the braking behavior using nominal design parameters is optimal, and the objective is to reduce variation from this behavior due to manufacturing tolerances in key design parameters.

Evaluate the experiment setup: Now that the setup is completed, you can evaluate the experiment.

```
Model evaluated at 50 samples.
```

Results from Simscape Example

Plot the scatter plot of objectives versus each parameter under consideration.



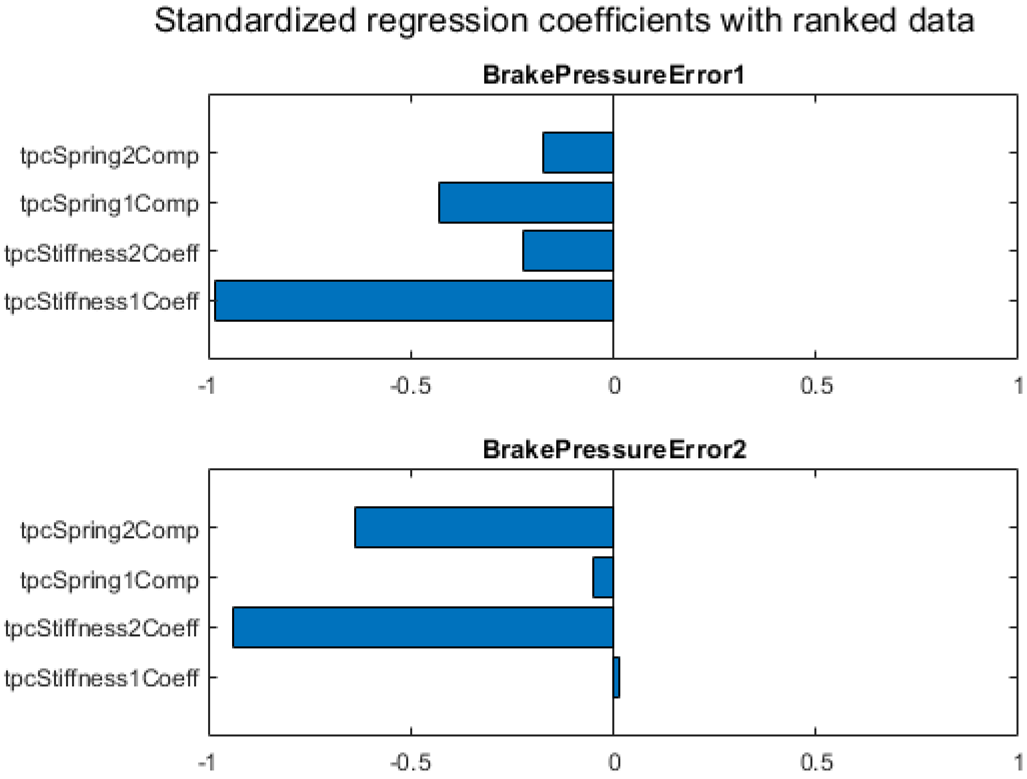
Also, plot the sensitivity of each parameter using Ranked Standardized regression method.

```
                        BrakePressureError1    BrakePressureError2
                        ___________________    ___________________

tpcStiffness1Coeff            -0.98208               0.016321
tpcStiffness2Coeff            -0.22383               -0.93928
tpcSpring1Comp                -0.43096              -0.047329
tpcSpring2Comp                -0.17068               -0.63816
```

The higher value of sensitivity indicates that the design is dependent on the corresponding parameters and vice versa. A positive sign for the sensitivity indicates the objective function is directly proportional to the parameters under consideration whilst a negative sign indicates that the objective function is inversely proportional to the respective parameter.

From the table, we can infer that Port 1 brake pressure is highly dependent on Spring 1 stiffness and Spring 1 initial compression. This is expected because the pressure generated in Line 1 is directly associated with Spring 1. The stiffness and initial compression of Spring 2 also affects the error function by a small amount as it has comparatively low sensitivities. Similarly, Port 2 brake pressure is highly dependent on Spring 2 stiffness and Spring 2 initial compression and it is not affected by changes in Spring 1 stiffness and initial compression.

A Tornado plot gives a graphical representation of sensitivities of parameters for each objective function. In this case there are two objective functions, hence there are two subplots in the Tornado plot.

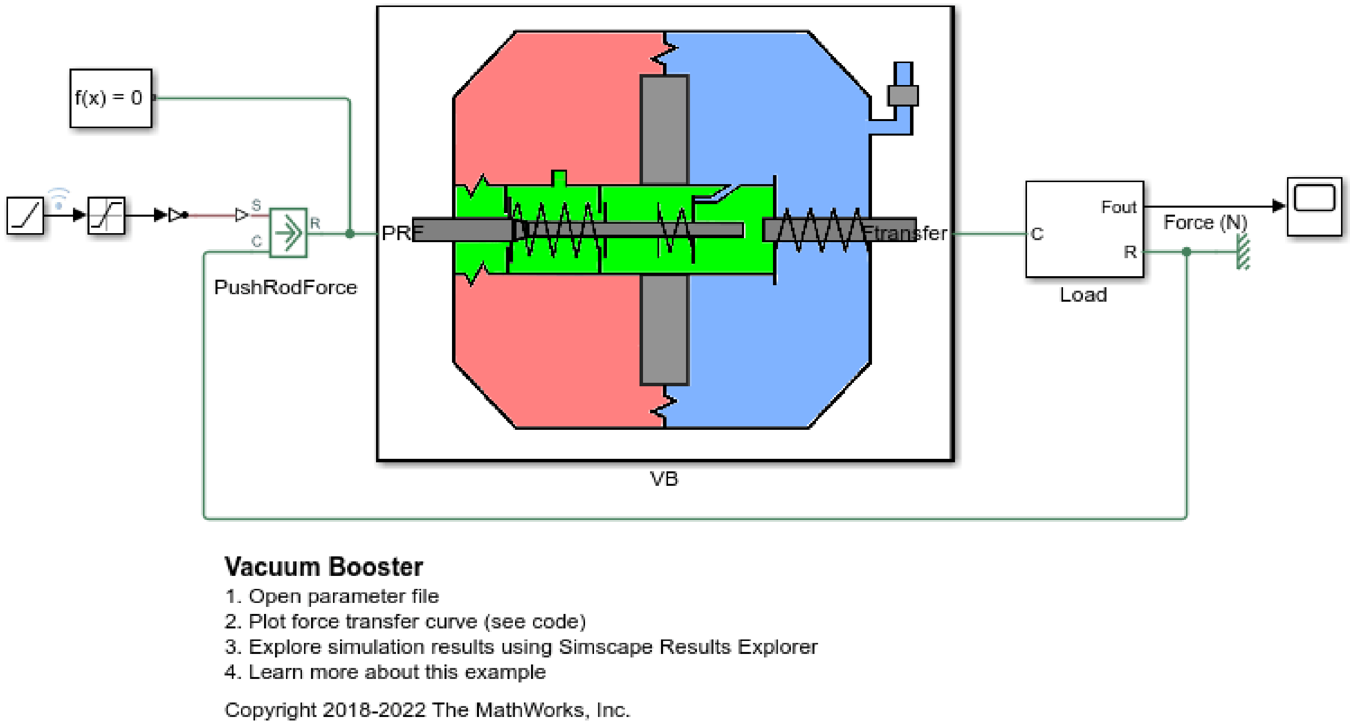


Vacuum Booster

This example shows how to model, parameterize and test a vacuum booster. The model is used to generate the plot between applied and generated forces for the brake vacuum booster.

Model

The following figure shows the model of a vacuum booster. The output of the model is the plot between applied and generated forces for the brake vacuum booster.



Vacuum Booster

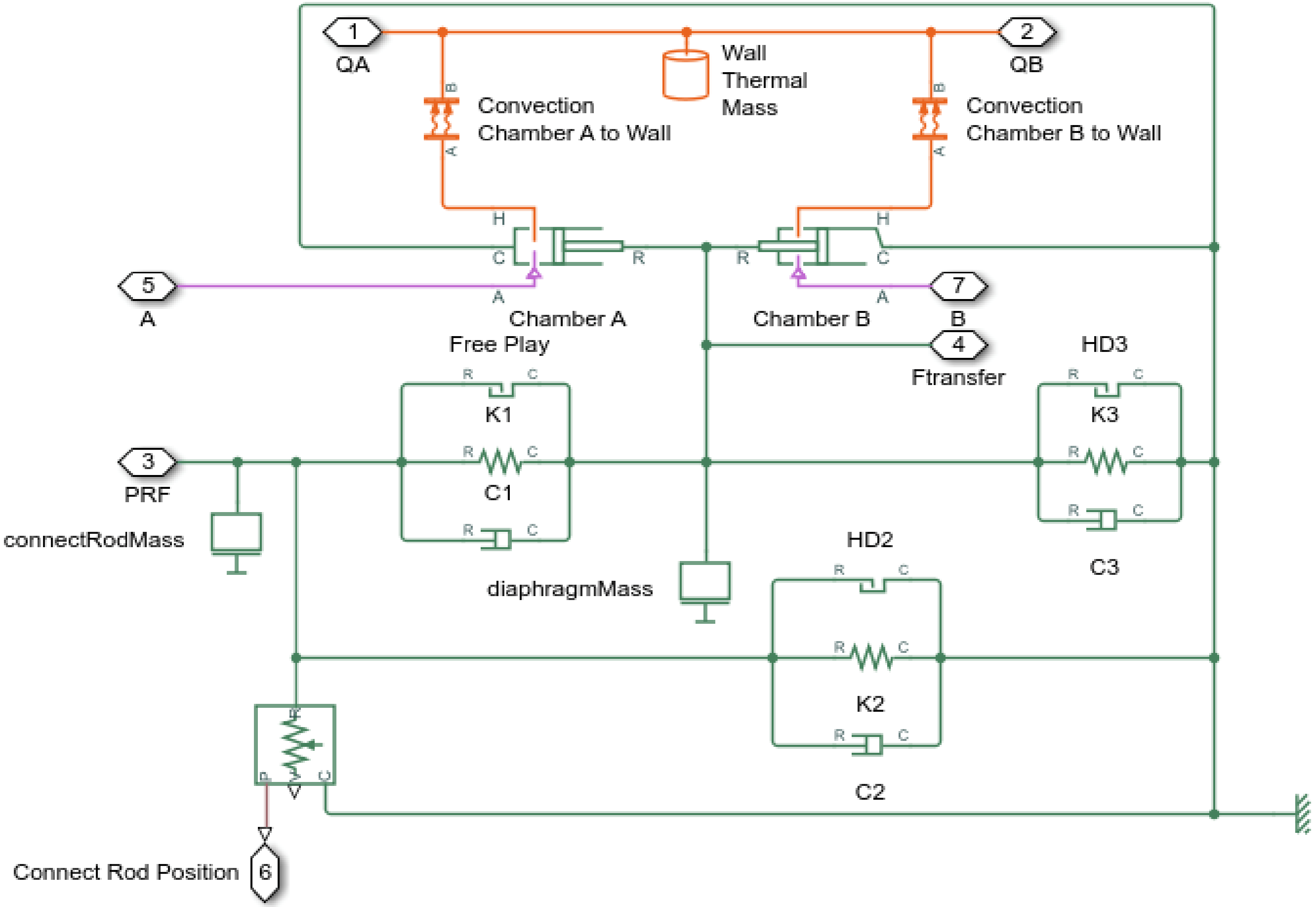
1. Open parameter file
2. Plot force transfer curve (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2018-2022 The MathWorks, Inc.

Diaphragm Chamber Subsystem

This subsystem shows the modeling of the diaphragm mass dynamics. The diaphragm has two sides. One side (chamber B) is permanently connected to the engine manifold, which experiences the vacuum pressure throughout the operation. However the other side (chamber A) is initially connected to the chamber B but after the rod moves a certain distance, based on design it gets disconnected from the chamber B and gets connected with the atmospheric pressure.

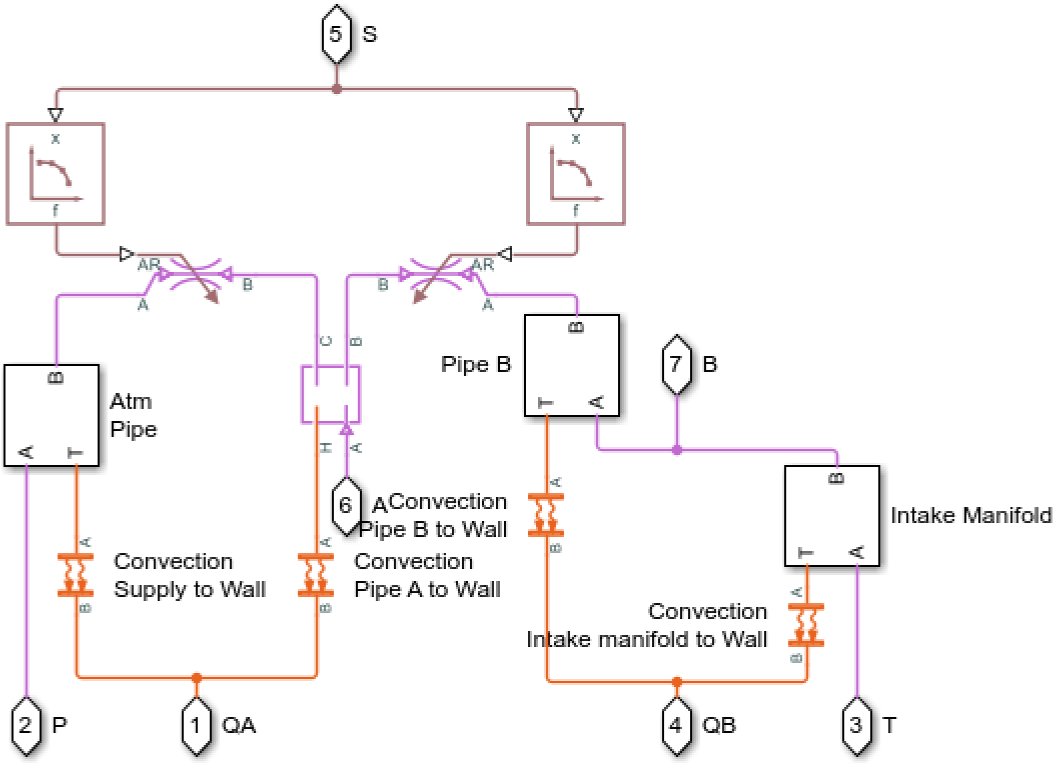
Open Subsystem



Manifold Subsystem

This subsystem shows the modeling of the gas dynamics.

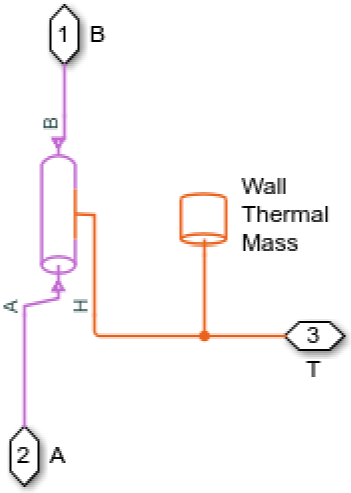
Open Subsystem



Pipe Subsystem

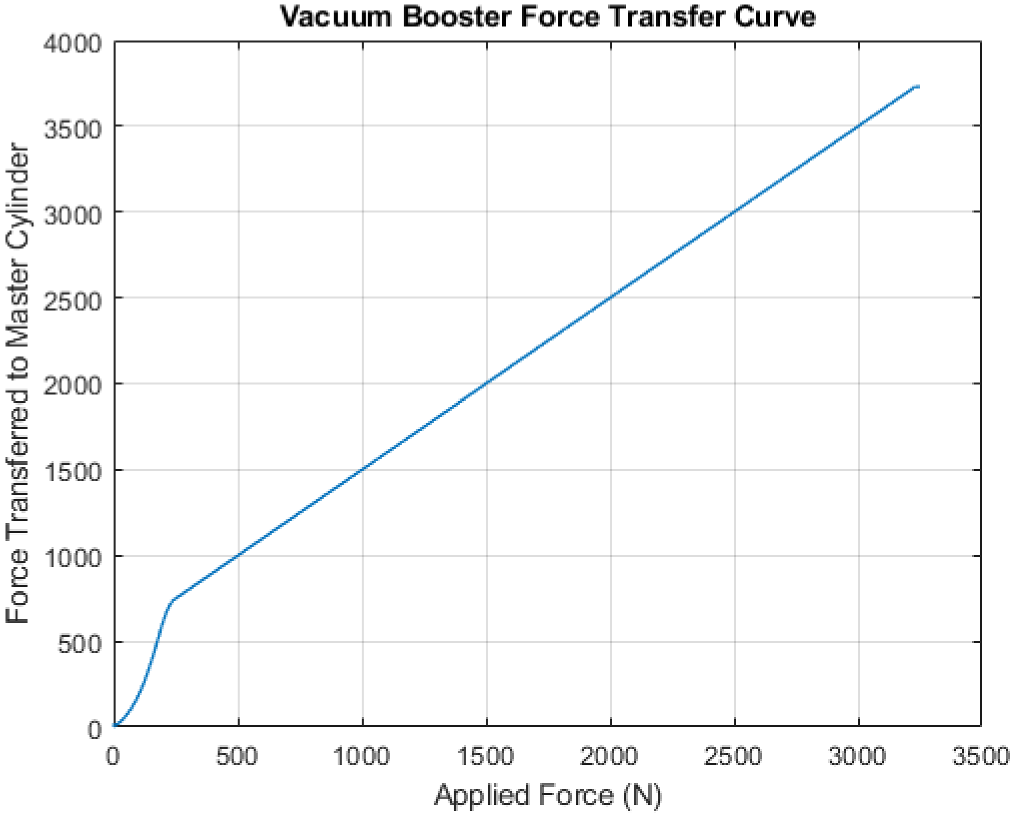
This subsystem shows the modeling of the various connection pipes in the vacuum booster.

Open Subsystem



Simulation Results from Simscape Logging

This script generates a plot between applied and generated forces for the brake vacuum booster.

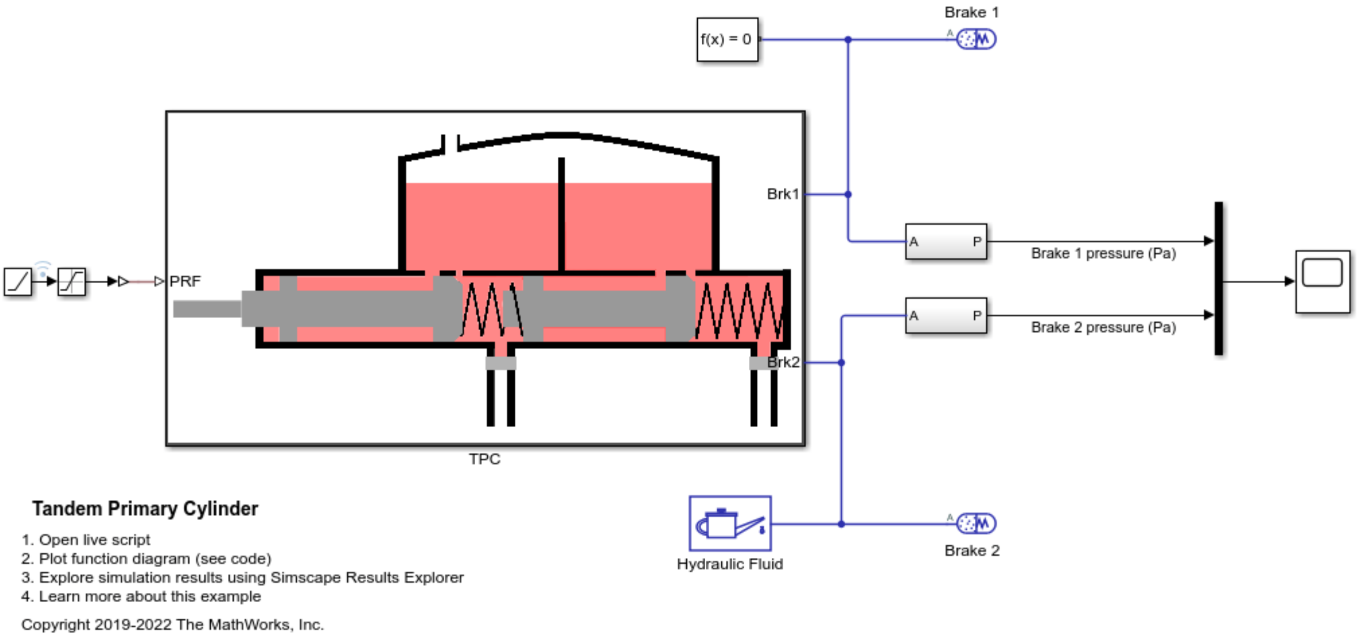

Vacuum Booster Force Transfer Curve
Force Transferred to Master Cylinder
Applied Force (N)
0
500
1000
1500
2000
2500
3000
3500
4000
0
500
1000
1500
2000
2500
3000
3500

Tandem Primary Cylinder

This example shows how to model, parameterize and test a tandem primary cylinder starting from manufacturer datasheet information. First, there is a brief discussion on the mathematical modeling of the system. Given the numerical data extracted from the datasheet, optimization is then used to determine remaining unknown parameters. The model is then simulated and the resulting push rod force - brake pressure relationship curve is compared with the curve provided on the manufacturer datasheet. Understanding the behavior of the tandem primary cylinder is an important prerequisite to selection of other braking system components.

Model

The following figure shows the tandem primary cylinder test harness. Spring-loaded accumulators are used for loading in the 1^{st} and the 2^{nd} circuits in place of disk or drum brakes. The input to the model is the push rod force coming from the lever mechanism or the brake booster. Output of the model is the pressures achieved in the brake circuits 1 and 2 respectively.



Tandem Primary Cylinder

1. Open live script
2. Plot function diagram (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2019-2022 The MathWorks, Inc.

Manufacturer datasheet data

The supplier has provided the following data in the datasheet.

1 *dp* : first and second circuit piston diameter.

2 *stroke1* : first circuit piston stroke.

3 *stroke2* : second circuit piston stroke.

4 *totalStroke* : overall stroke.

5 *disp1* : first circuit displacement.

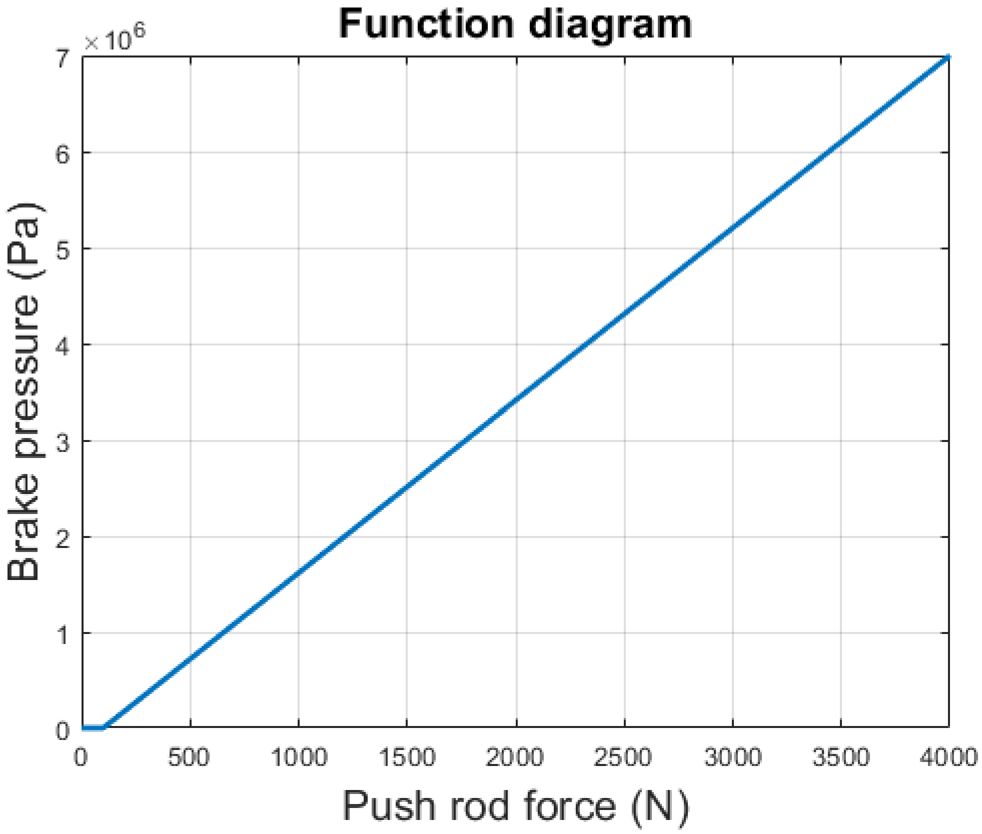
6 *disp2* : second circuit displacement.

7 *totalDisp* : net displacement.

8 *maxPress* : max pressure.

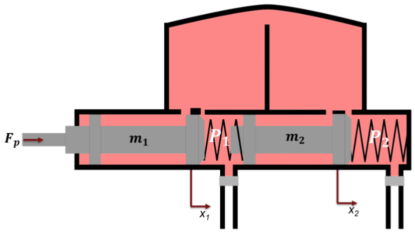
Pressure development characteristics

The supplier datasheet gives the following function diagram or the push rod force-brake pressure curve for the tandem primary cylinder.



Equations of motion (EoM) of system

The following figure shows the tandem primary cylinder. Mass 1 and mass 2 use the coordinate frames shown below. This description only includes the mechanical EoM of the system. The fluid dynamics equations are not shown.



$$m_1\frac{d^2x_1}{dt^2}+c_1\left(\frac{dx_1}{dt}-\frac{dx_2}{dt}\right)+k_1(x_1-x_2)=F_p-p_1A-Preload_1+R_1,\ \text{when}\ x_1=0$$

$$m_2\frac{d^2x_2}{dt^2}+c_1(\frac{dx_2}{dt}-\frac{dx_1}{dt})+c_2\frac{dx_2}{dt}+k_1(x_2-x_1)+k_2x_2=p_1A-p_2A+Preload_1-Preload_2+R_2,\text{ when }x_2=0$$

$$m_1\frac{d^2x_1}{dt^2}+c_1(\frac{dx_1}{dt}-\frac{dx_2}{dt})+k_1(x_1-x_2)=F_p-p_1A-Preload_1,\text{ when }x_1>0$$

$$m_2\frac{d^2x_2}{dt^2}+c_1(\frac{dx_2}{dt}-\frac{dx_1}{dt})+c_2\frac{dx_2}{dt}+k_1(x_2-x_1)+k_2x_2=p_1A-p_2A+Preload_1-Preload_2,\text{ when }x_2>0$$

where,

x_1 and x_2 are positions of piston mass 1 and mass 2 from the left side assumed hard stop.

m_1 and m_2 are masses of piston 1 and 2 respectively.

c_1 and c_2 are damping coefficients of piston 1 and 2 respectively.

k_1 and k_2 are stiffness coefficients of left (Spring 1) and right (spring 2) respectively.

F_p is force applied on push rod of primary cylinder.

P_1 and P_2 are pressures in brake circuit 1 and 2 respectively.

R_1 and R_2 are reaction forces on mass 1 and 2 respectively when both masses are at the leftmost position.

$Preload_1$ and $Preload_2$ are preloads on left and right side springs respectively.

Parameter estimation method

Some parameters, which are important for functionality assessment, are not provided in the manufacturer's datasheet, therefore to simulate the model it is pertinent to estimate these parameters. The EoM are converted to steady state equations to estimate unknown parameters.

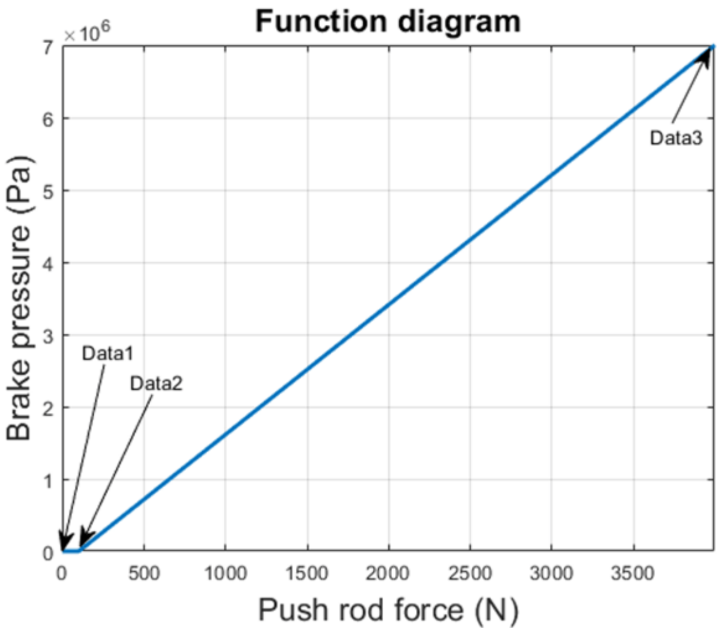
$$k_1(x_1-x_2)=F_p-p_1A-Preload_1+R_1,\text{ when }x_1=0$$

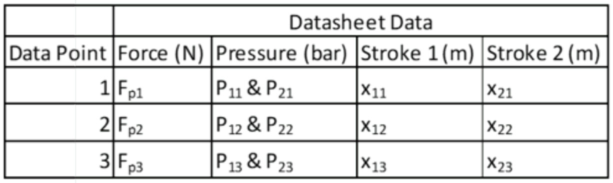
$$k_1(x_2-x_1)+k_2x_2=p_1A-p_2A+Preload_1-Preload_2+R_2,\text{ when }x_2=0$$

$$k_1(x_1-x_2)=F_p-p_1A-Preload_1,\text{ when }x_1>0$$

$$k_1(x_2-x_1)+k_2x_2=p_1A-p_2A+Preload_1-Preload_2,\text{ when }x_2>0$$

Data for the function diagram is explained with following figure.





	Datasheet Data			
Data Point	Force (N)	Pressure (bar)	Stroke 1 (m)	Stroke 2 (m)
1	F_{p1}	P_{11} & P_{21}	x_{11}	x_{21}
2	F_{p2}	P_{12} & P_{22}	x_{12}	x_{22}
3	F_{p3}	P_{13} & P_{23}	x_{13}	x_{23}

where,

F_{p1}, F_{p2} and F_{p3} are the push rod forces for points 1,2, and 3 respectively.

$P_{1*} = P_{2*}$,Here $*$ is $1, 2, 3$ for the brake circuit pressures at points 1,2, and 3 respectively.

x_{11} and $x_{21} = 0$ represent the leftmost positions for piston 1 and 2 respectively.

$x_{13} = TotalStroke$ and $x_{23} = Stroke2$ as per the manufacturer datasheet data.

x_{12} and x_{22} are piston positions at data point 2. This point needs to be iterated on during optimisation to estimate the unknown parameters.

The estimation scheme is as follows:

$$\begin{bmatrix} x_{11} - x_{21} & 0 & 1 & 0 & -1 & 0 \\ x_{21} - x_{11} & x_{21} & -1 & 1 & 0 & -1 \\ x_{12} - x_{22} & 0 & 1 & 0 & 0 & 0 \\ x_{22} - x_{12} & x_{22} & -1 & 1 & 0 & 0 \\ x_{13} - x_{23} & 0 & 1 & 0 & 0 & 0 \\ x_{23} - x_{13} & x_{23} & -1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ Preload_1 \\ Preload_2 \\ R_1 \\ R_2 \end{bmatrix} = \begin{bmatrix} F_{p1} - p_{11}A \\ p_{11}A - p_{21}A \\ F_{p2} - p_{12}A \\ p_{12}A - p_{22}A \\ F_{p3} - p_{13}A \\ p_{13}A - p_{23}A \end{bmatrix}$$

The estimation should yield results with the following conditions met:

1. R_1 and R_2 should be greater than or equal to 0.
2. k_1 and k_2 should be greater than 0.

Assumed parameters

Some parameters are assumed as they are neither given in the datasheet nor can they be estimated. The following is the list:

1. *c1* is damping coefficient for piston 1.
2. *c2* is damping coefficient for piston 2.
3. *mass1* is piston 1 mass.
4. *mass2* is piston 2 mass.
5. *v2D* is brake circuit 1 dead volume.
6. *v2M* is brake circuit 1 maximum volume.
7. *v4D* is brake circuit 2 dead volume.
8. *v4M* is brake circuit 2 maximum volume.
9. *a1Ori* is compensating orifice area.
10. *a2Ori* is brake circuit orifice area

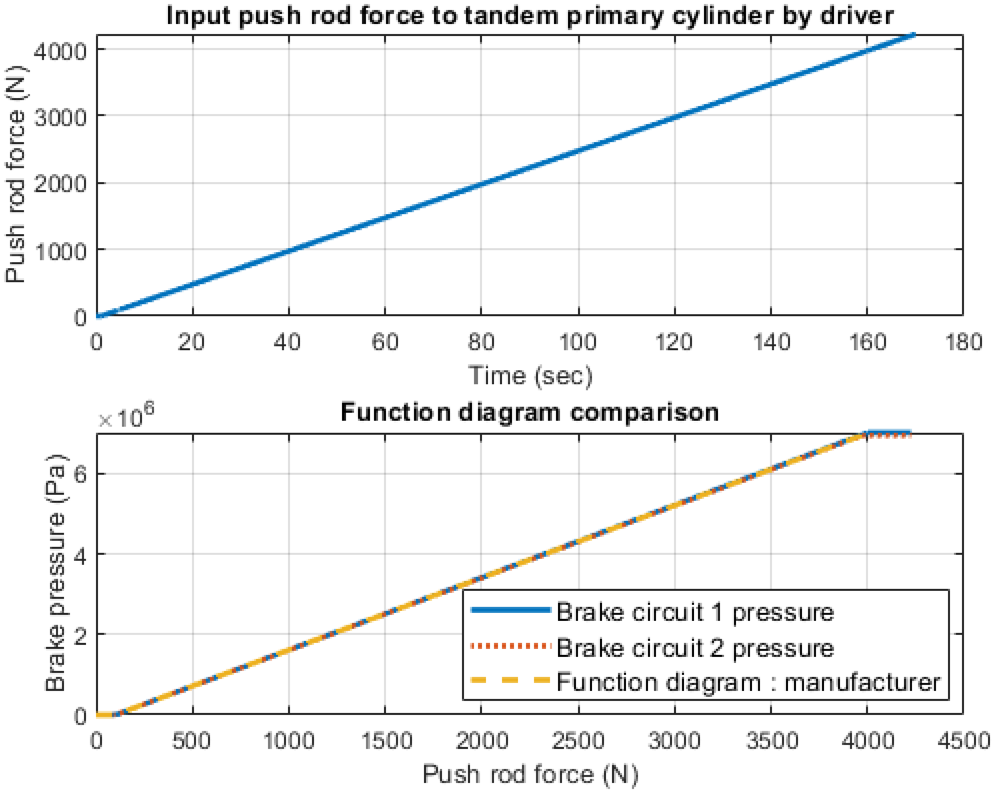
Load circuit

It is important to test the system with the correct load. In the test model, loading is done on the tandem primary cylinder using spring-based accumulators. Setting up correct parameters in the accumulators is important. From the datasheet it is known what should be the stroke values for brake circuits 1 and 2 respectively. Iterated parameters x_{12} and x_{22} are important for deciding the fluid chamber capacity of the accumulators. Assumed or estimated parameters for the load circuit accumulators are:

1. vol1 is fluid chamber capacity for brake circuit 1.
2. vol2 is fluid chamber capacity for brake circuit 2.
3. loadP1 is pressure at full capacity for brake circuit 1.
4. loadP2 is pressure at full capacity for brake circuit 2.

Simulation results from simscape logging

The model generates push rod force-brake pressure plots for selected manufacturer designs. The applied push rod force to the tandem primary cylinder is ramped at a 25 N/sec rate in the simulation .

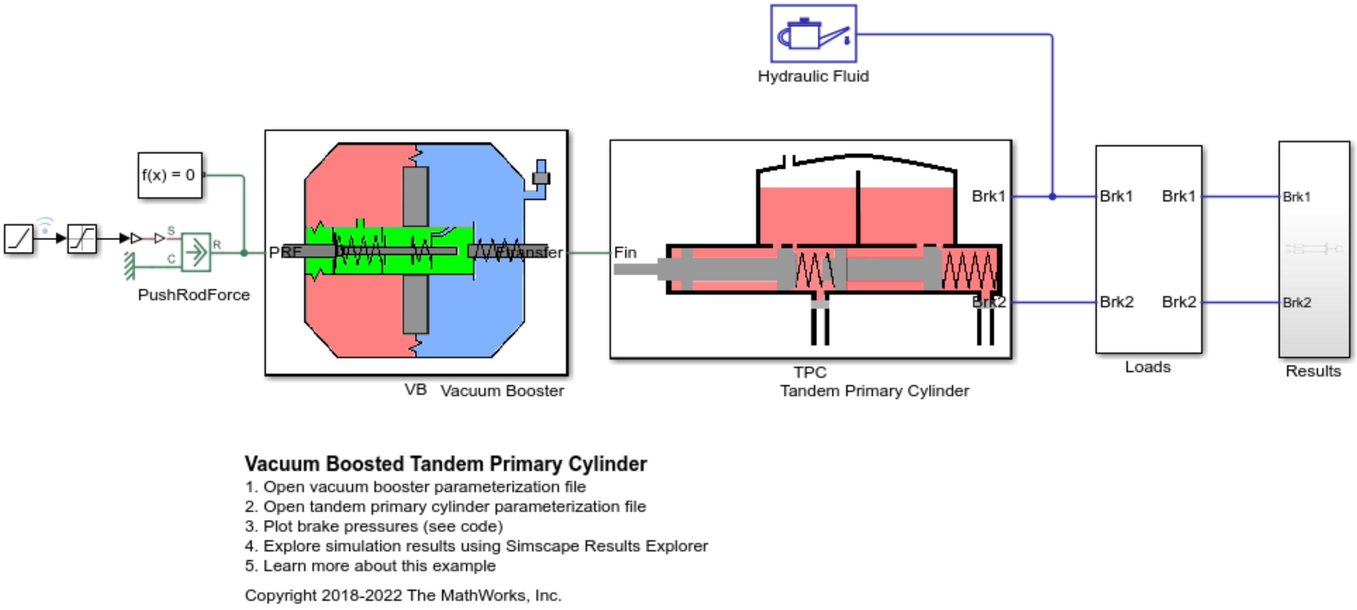

Input push rod force to tandem primary cylinder by driver
Push rod force (N)
0
1000
2000
3000
4000
0
20
40
60
80
100
120
140
160
180
Time (sec)
Function diagram comparison
×10^6
Brake pressure (Pa)
0
2
4
6
Brake circuit 1 pressure
Brake circuit 2 pressure
Function diagram : manufacturer
0
500
1000
1500
2000
2500
3000
3500
4000
4500
Push rod force (N)

Vacuum Boosted Tandem Primary Cylinder

This example shows how to model, parameterize and test a vacuum boosted tandem primary cylinder. A model is used to generate the plot between the applied force and the generated pressures in the brake lines.

Model

The following figure shows the model of a vacuum boosted tandem primary cylinder. The model has push rod force as input and it generates the pressures in the brake lines as output.



Vacuum Boosted Tandem Primary Cylinder

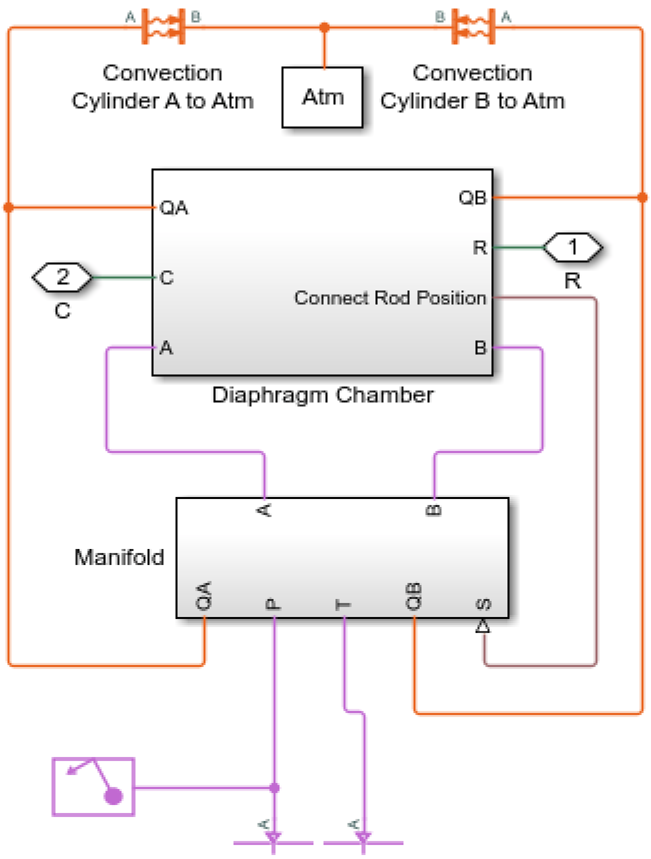
1. Open vacuum booster parameterization file
2. Open tandem primary cylinder parameterization file
3. Plot brake pressures (see code)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example

Copyright 2018-2022 The MathWorks, Inc.

Vacuum Booster Subsystem

This subsystem shows the modeling of the vacuum booster. A vacuum booster boosts the force applied by a driver to the tandem primary cylinder. The boosted force depends on the diameter of the diaphragm in the vacuum booster and the vacuum generated due to the air suction through the intake manifold of the engine.

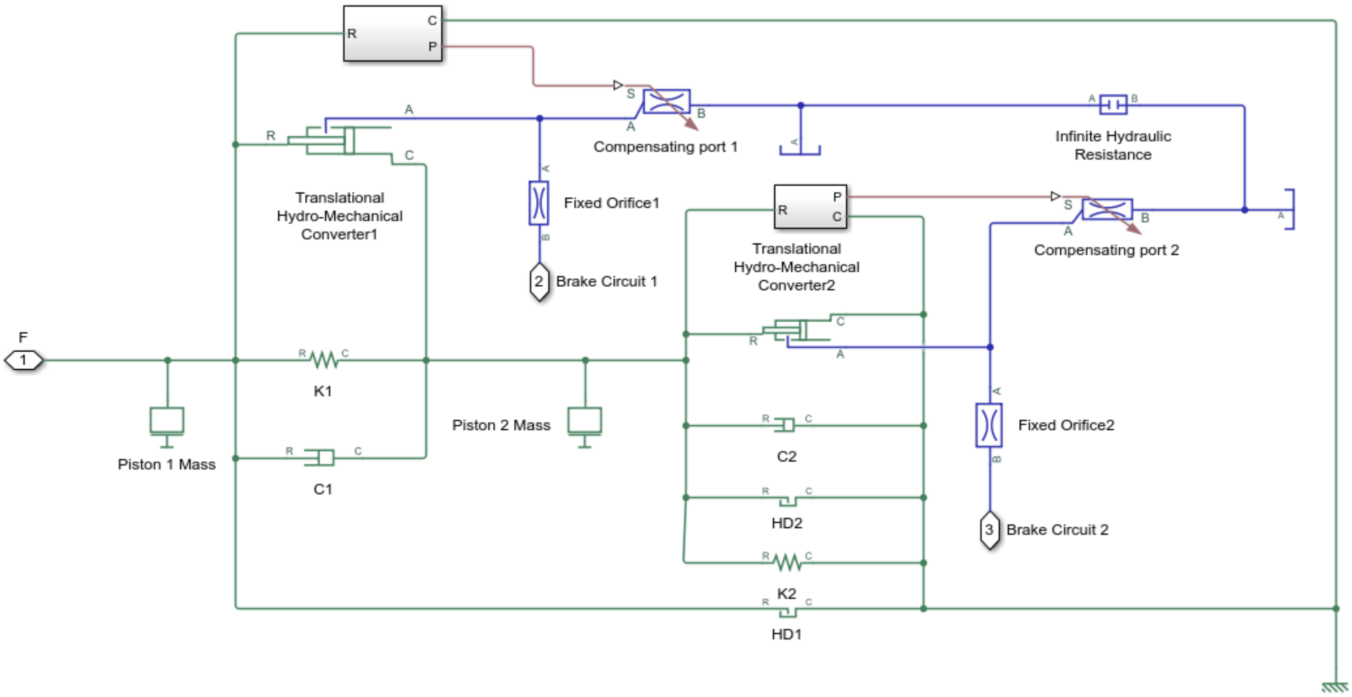
Open Subsystem



Tandem Primary Cylinder Subsystem

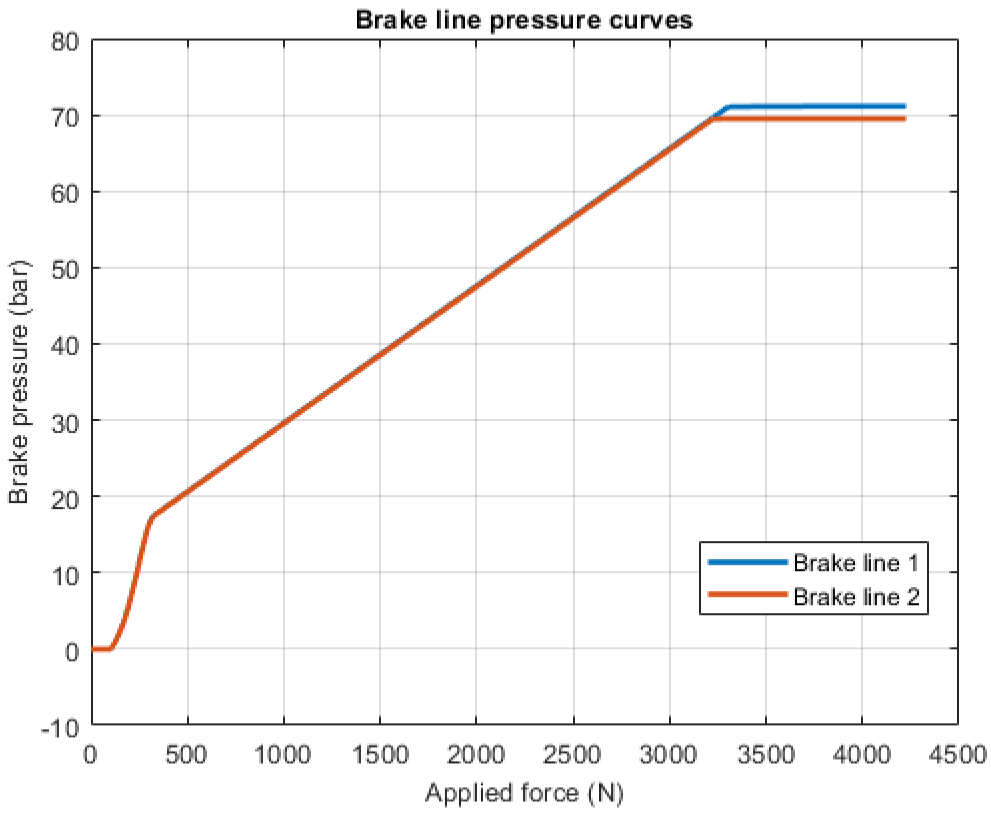
This subsystem shows the modeling of the tandem primary cylinder. A tandem primary cylinder pushes the hydraulic fluid to the caliper disk or the drum brakes through the hydraulic connections or pipes in an automotive vehicle. Pressure is generated in the hydraulic chambers of the brakes due to the fluid flow from the tandem primary cylinder.

Open Subsystem



Simulation Results from Simscape™ Logging

The model generates the relationship between the applied force and the generated pressures in the brake lines as shown in the plot below.

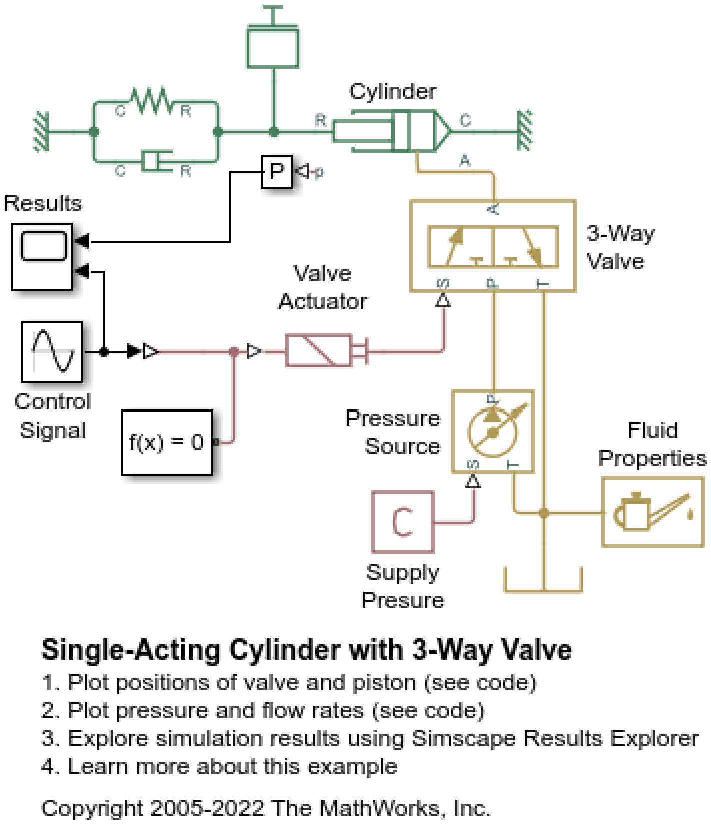

Brake line pressure curves
Brake pressure (bar)
80
70
60
50
40
30
20
10
0
-10
0
500
1000
1500
2000
2500
3000
3500
4000
4500
Applied force (N)
Brake line 1
Brake line 2

Single-Acting Cylinder with 3-Way Valve

This example shows a single-acting hydraulic cylinder controlled by an 3-way directional valve. It drives a load consisting of a mass, viscous friction, and preloaded spring. The pump unit is assumed to be powerful enough to maintain constant pressure at the valve inlet.

The valve is modeled by a 2-Position Valve Actuator block (Valve Actuator) and a 3-Way Directional Valve block (3-Way Valve). At the start of simulation, valve orifices A and T are connected and the cylinder chamber is connected to the tank. When the control signal for the valve exceeds 12, the valve spool moves, connecting valve ports P and A and closing the path between valve ports A and T. As a result, the cylinder chamber is connected to the pressure source and moves the load.

Model

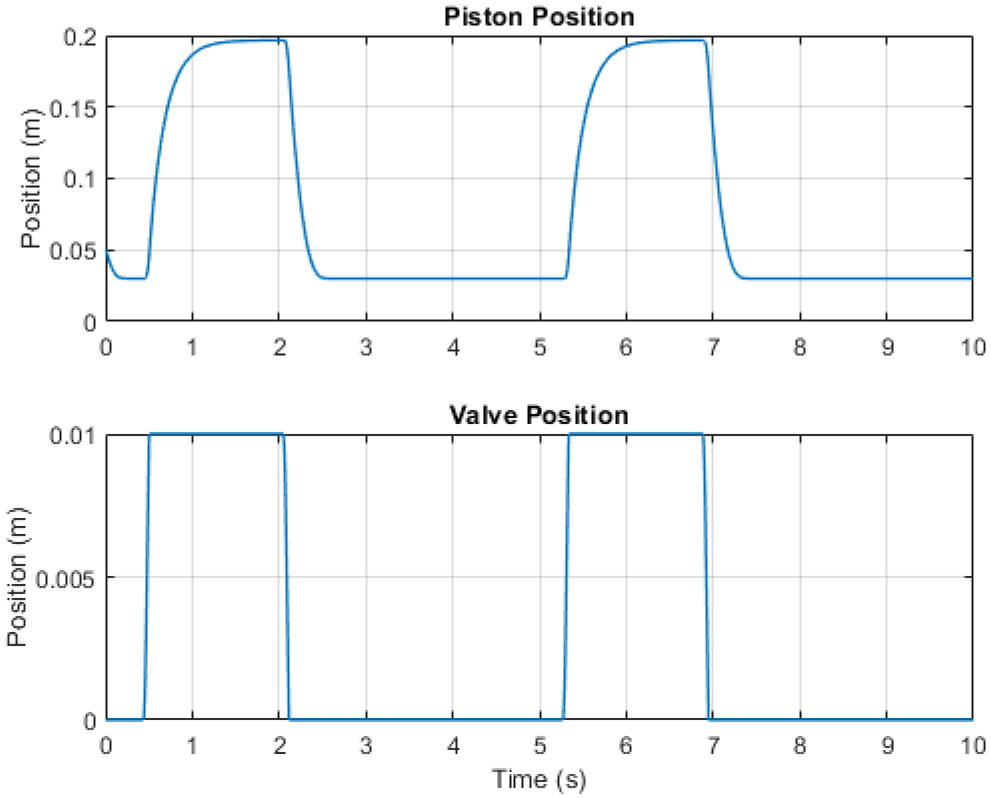


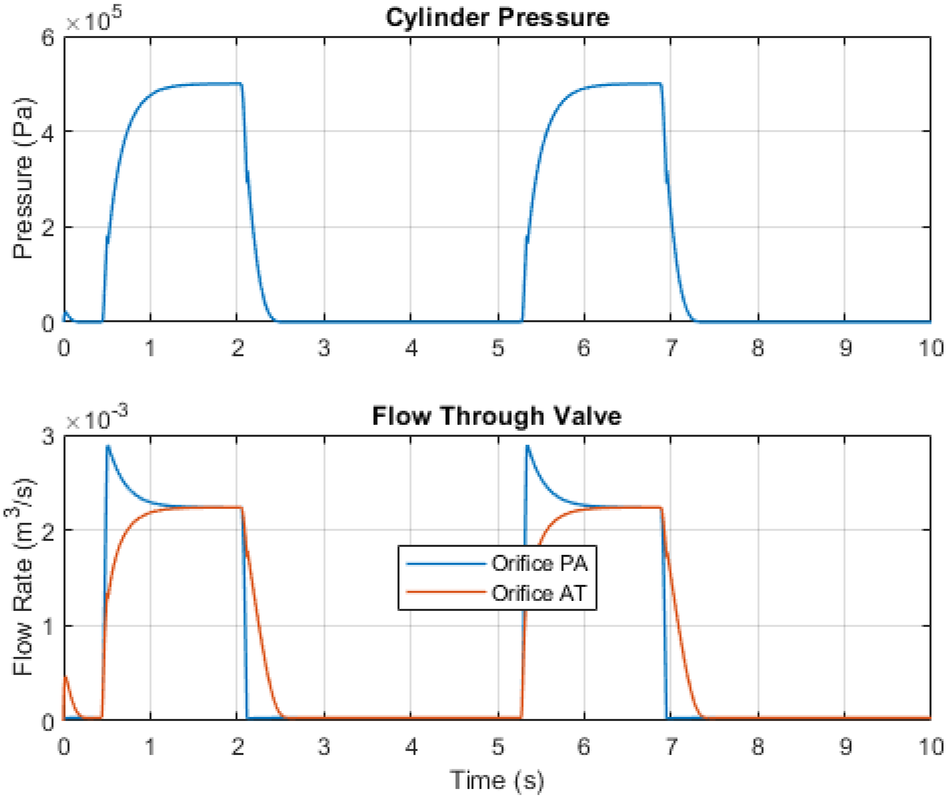
Single-Acting Cylinder with 3-Way Valve

1. Plot positions of valve and piston (see code)
2. Plot pressure and flow rates (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2005-2022 The MathWorks, Inc.

Simulation Results from Simscape Logging

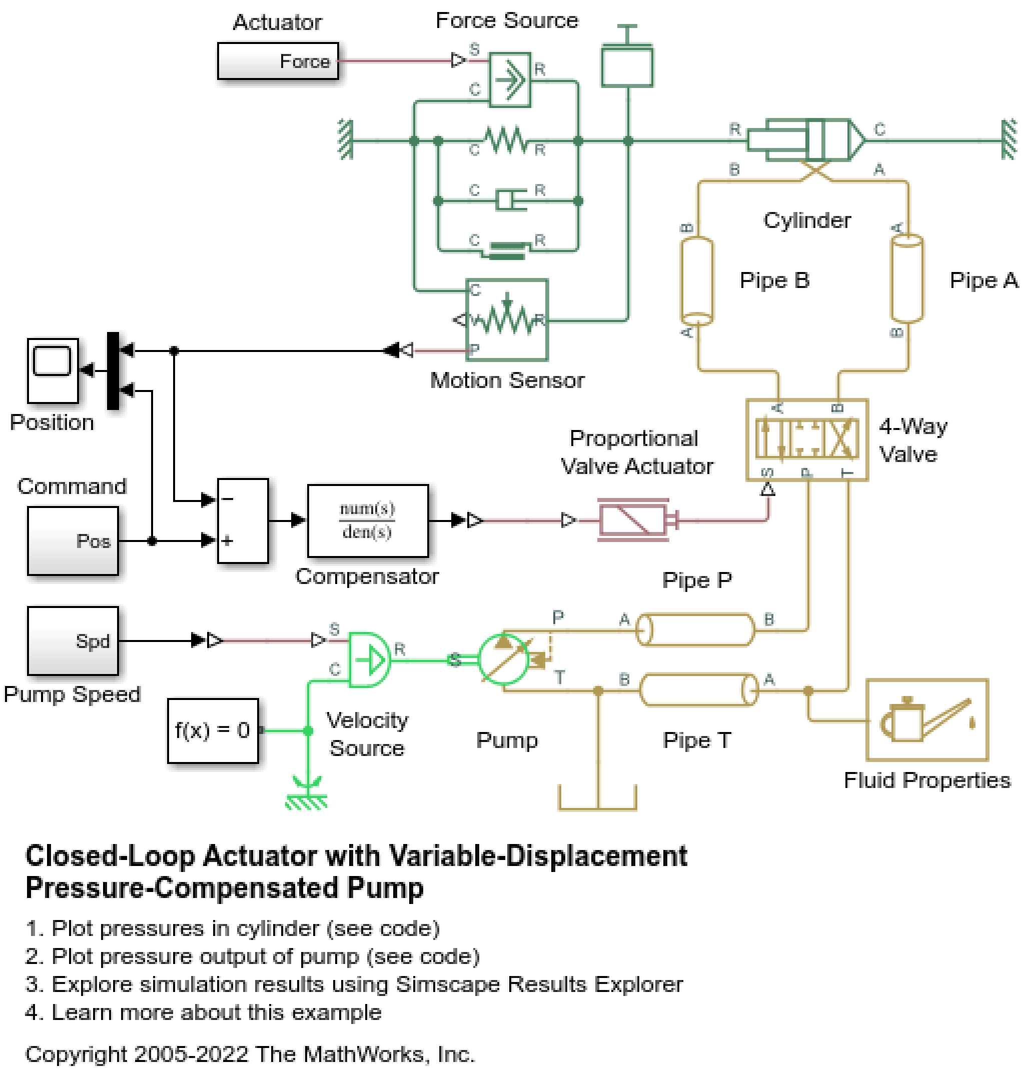



Cylinder Pressure
×10^5
Pressure (Pa)
6
4
2
0
0 1 2 3 4 5 6 7 8 9 10
Flow Through Valve
×10^-3
Flow Rate (m^3/s)
3
2
1
0
Orifice PA
Orifice AT
0 1 2 3 4 5 6 7 8 9 10
Time (s)

Closed-Loop Actuator with Variable-Displacement Pressure-Compensated Pump

This example shows a closed-loop actuator that consists of a proportional 4-way directional valve driving a double-acting hydraulic cylinder. The cylinder drives a load consisting of a mass, viscous and Coulomb friction, constant force, and a spring. The actuator is powered by a variable-displacement, pressure-compensated pump, driven by a constant velocity motor. Pipelines between the valve, cylinder, pump, and the tank are modeled using the Hydraulic Pipeline blocks.

Model

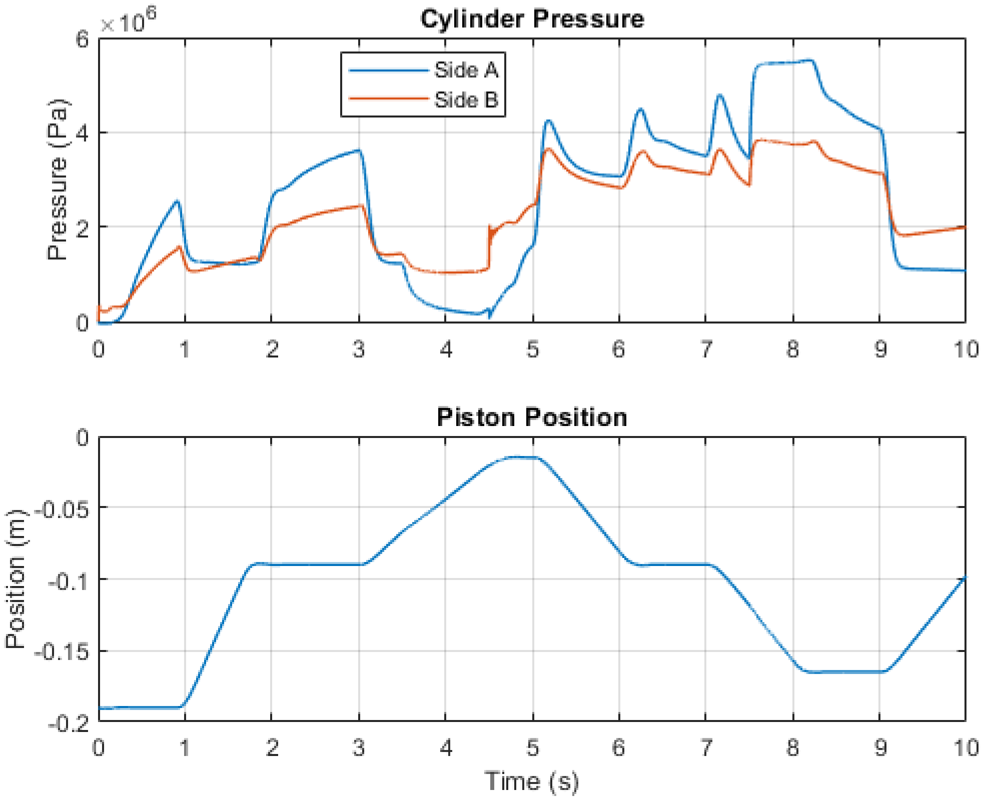


Closed-Loop Actuator with Variable-Displacement Pressure-Compensated Pump

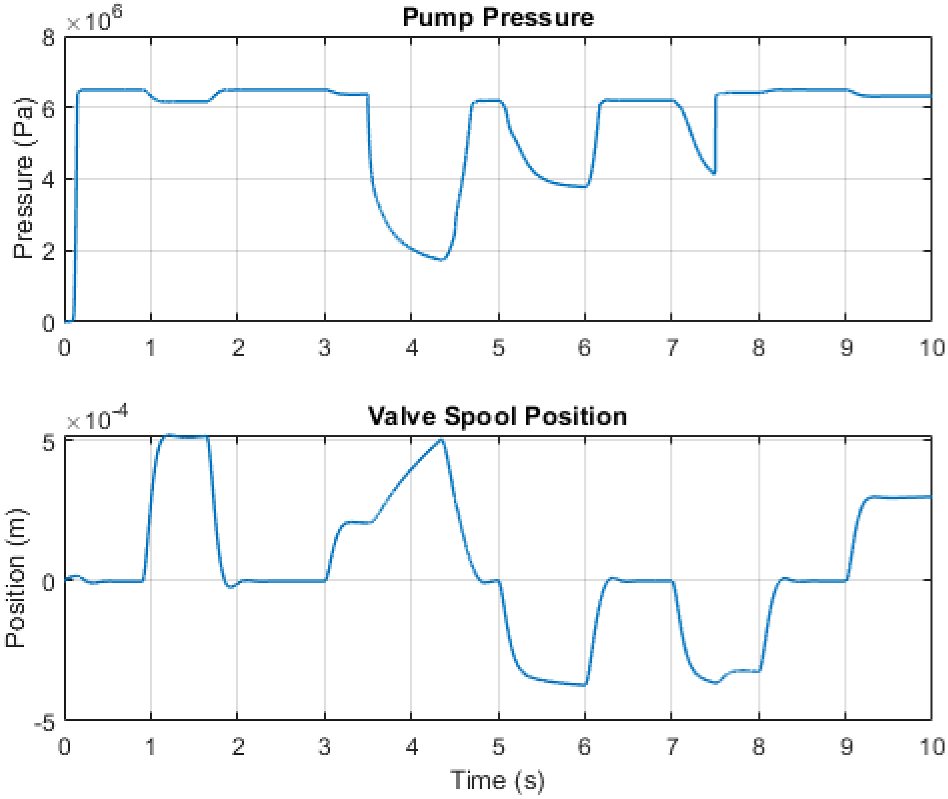
1. Plot pressures in cylinder (see code)
2. Plot pressure output of pump (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Simulation Results from Simscape Logging

The plots below show the pressures in the cylinder and the position of the piston.



The plots below show the pressure provided by the pump and the position of the spool in the 4-Way Valve. As the spool position changes the load on the pump increases, causing fluctuations in the pressure at the output of the pump.

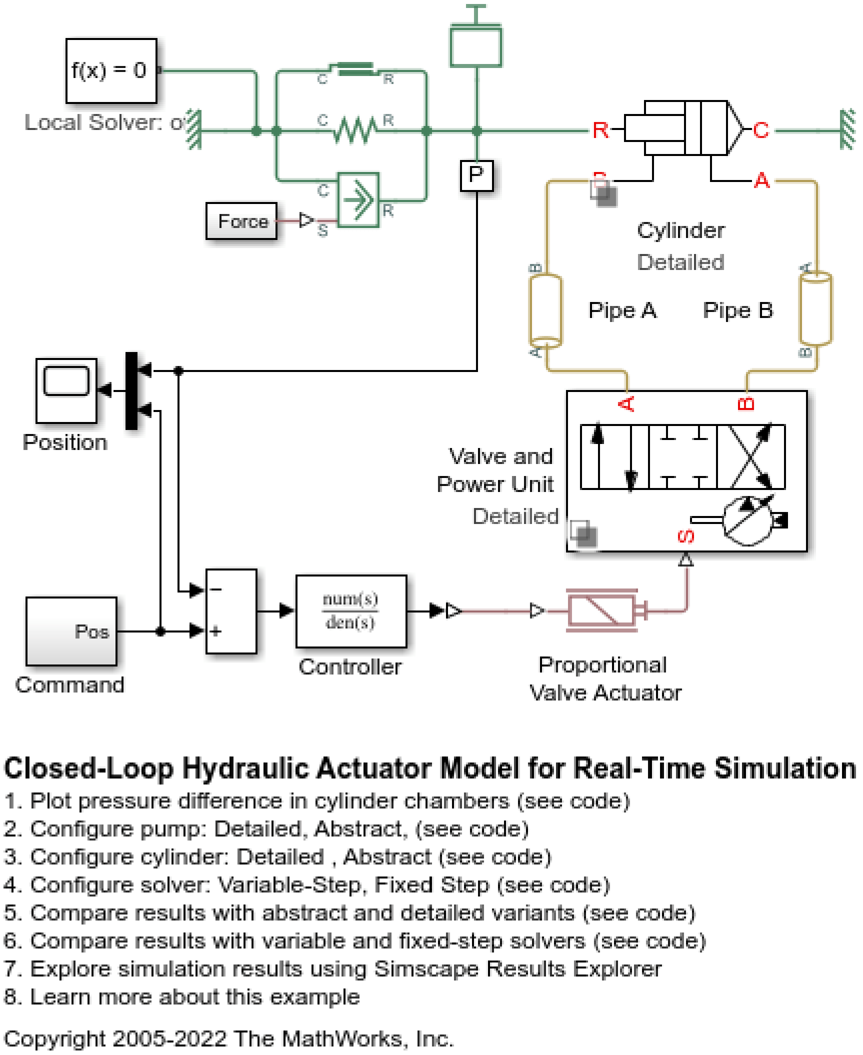

Pump Pressure
×10^6
Pressure (Pa)
8
6
4
2
0
0
1
2
3
4
5
6
7
8
9
10
Valve Spool Position
×10^-4
Position (m)
5
0
-5
0
1
2
3
4
5
6
7
8
9
10
Time (s)

Closed-Loop Hydraulic Actuator Model for Real-Time Simulation

The actuator consists of a proportional 4-way directional valve driving a double-acting hydraulic cylinder. The cylinder drives a load consisting of a mass, viscous and Coulomb friction, constant force, and a spring. The actuator is powered by a variable-displacement, pressure-compensated pump, driven by a constant velocity motor. Pipelines between the valve, cylinder, pump, and the tank are simulated with the Hydraulic Pipeline blocks.

Abstract variants of the hydraulic cylinder and of the valve and pump unit are also included in the model. The model can be configured to use these abstract variants, which neglect effects such as compressibility and use simpler equations that are numerically more efficient. The model can also be configured for simulation with a fixed-step solver, as would be used in hardware-in-the-loop testing.

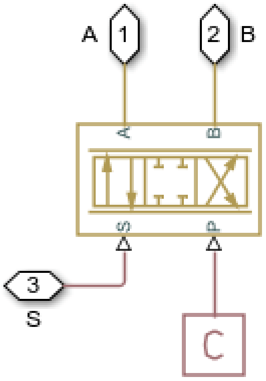
Model



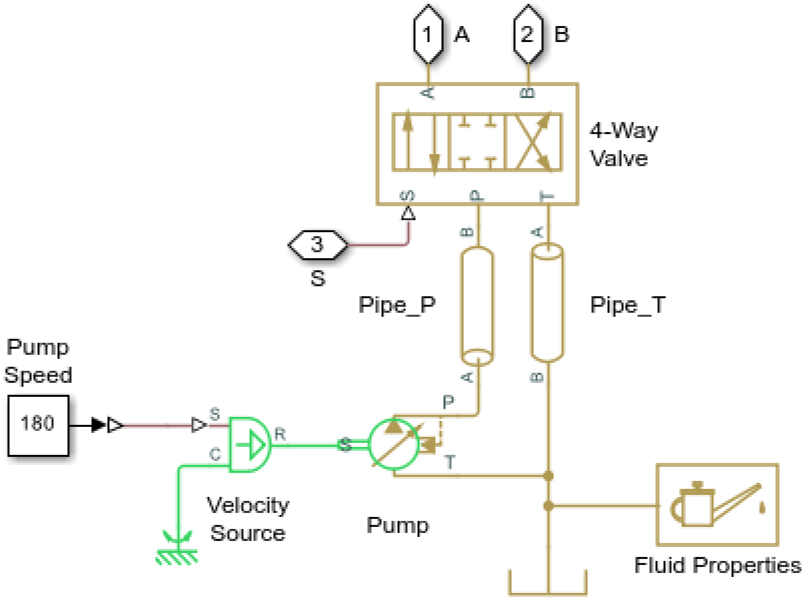
Closed-Loop Hydraulic Actuator Model for Real-Time Simulation

1. Plot pressure difference in cylinder chambers (see code)
2. Configure pump: Detailed, Abstract, (see code)
3. Configure cylinder: Detailed , Abstract (see code)
4. Configure solver: Variable-Step, Fixed Step (see code)
5. Compare results with abstract and detailed variants (see code)
6. Compare results with variable and fixed-step solvers (see code)
7. Explore simulation results using Simscape Results Explorer
8. Learn more about this example

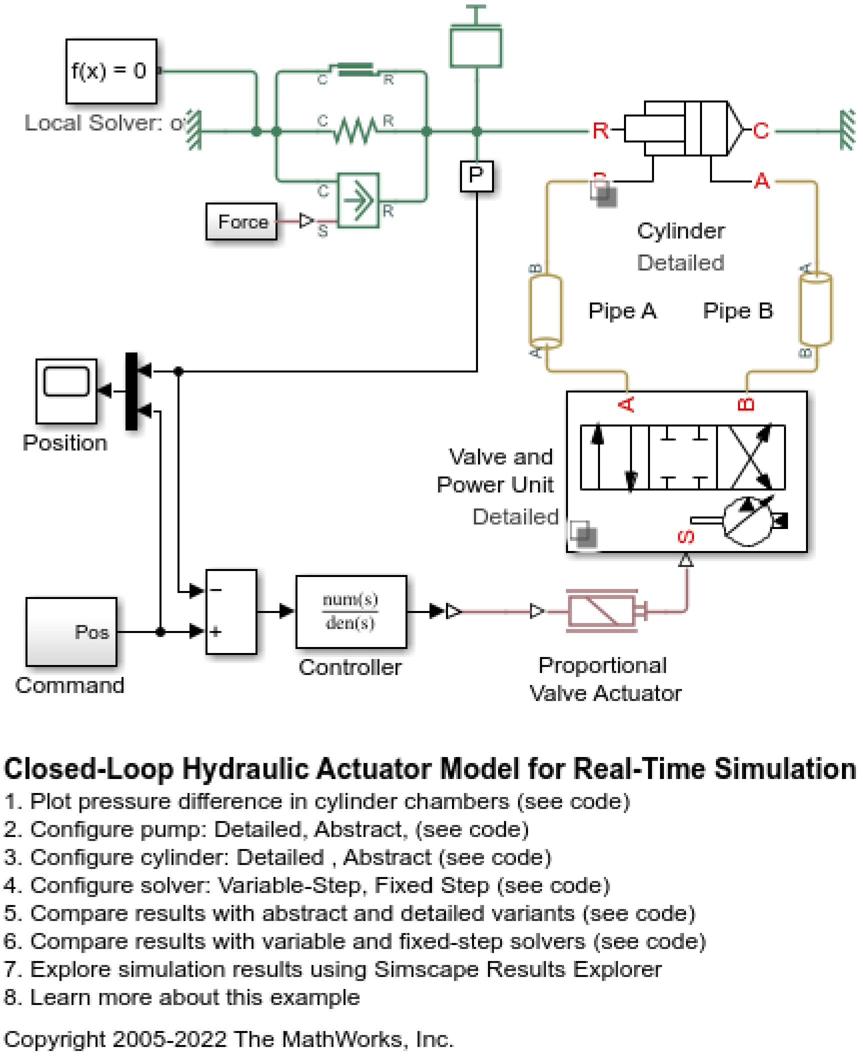
Valve and Power Unit, Abstract Variant



Valve and Power Unit, Detailed Variant



Simulation Results from Simscape Logging

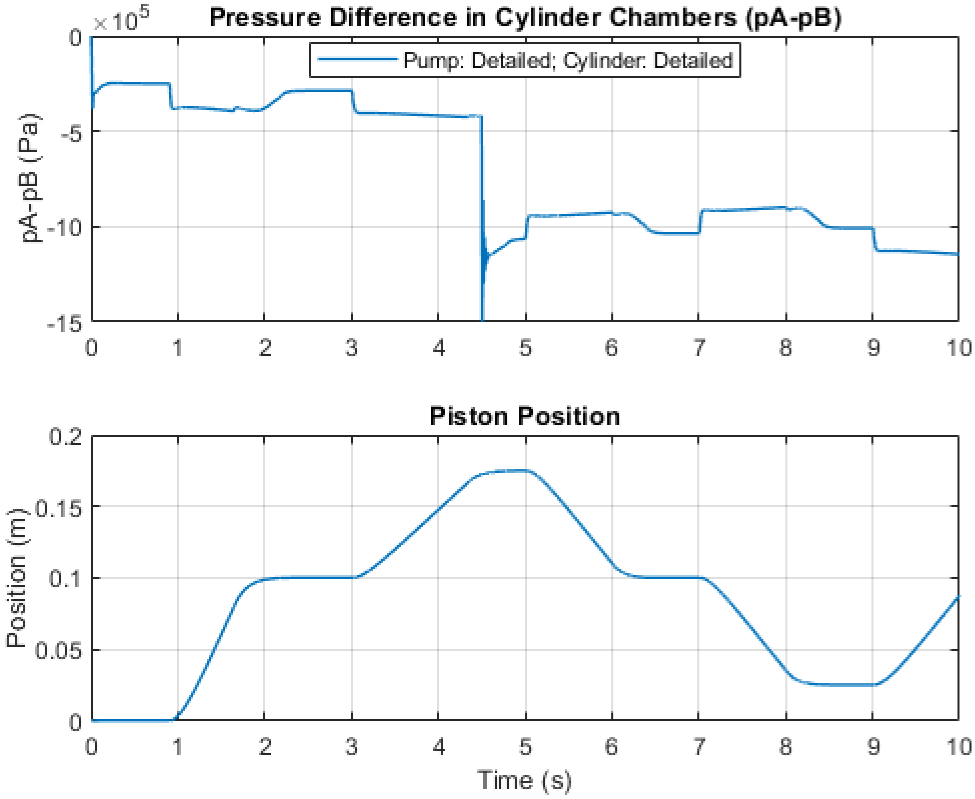


Closed-Loop Hydraulic Actuator Model for Real-Time Simulation

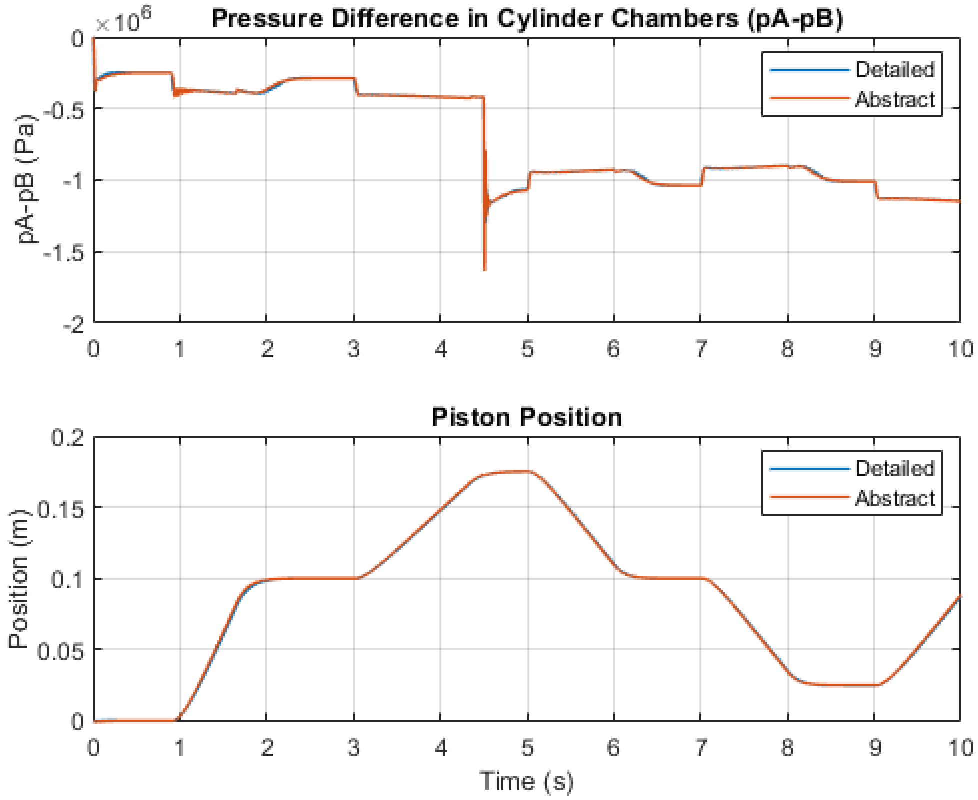
1. Plot pressure difference in cylinder chambers (see code)
2. Configure pump: Detailed, Abstract, (see code)
3. Configure cylinder: Detailed , Abstract (see code)
4. Configure solver: Variable-Step, Fixed Step (see code)
5. Compare results with abstract and detailed variants (see code)
6. Compare results with variable and fixed-step solvers (see code)
7. Explore simulation results using Simscape Results Explorer
8. Learn more about this example

Copyright 2005-2022 The MathWorks, Inc.

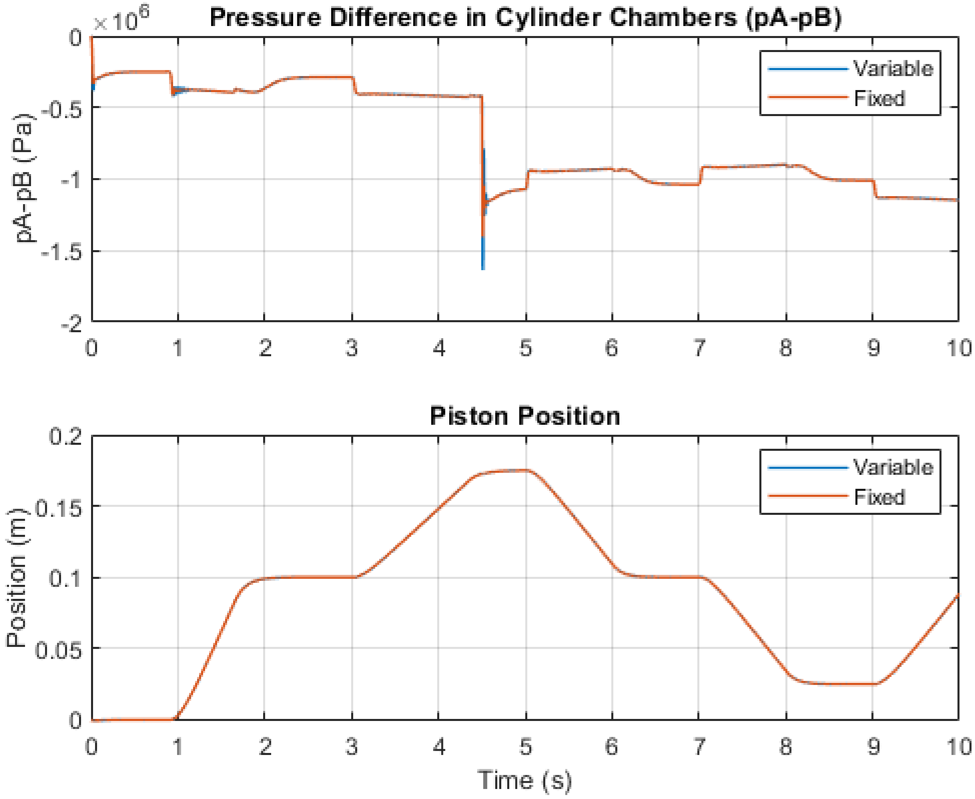
The plots below show the pressure difference between cylinder chambers A and B, as well as the position of the piston.



The plots below show the behavior of the abstract models and the detailed model. The results are very similar, indicating that the abstract models can be used without a significant loss of accuracy.



The plots below show the behavior during a simulation with a variable-step solver (Simscape™ local solver off) and a fixed-step solver (Simscape local solver on). The results are very similar, indicating that a fixed-step solver can be used without a significant loss of accuracy, and that the models can be used for hardware-in-the loop simulation.

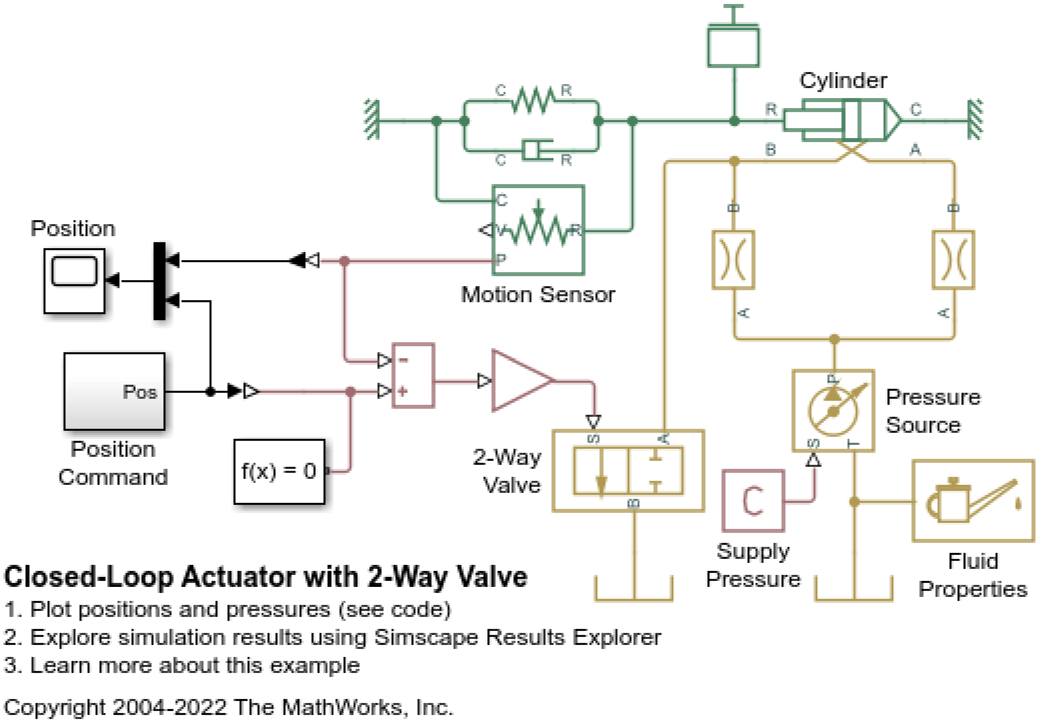

Pressure Difference in Cylinder Chambers (pA-pB)
×10^6
pA-pB (Pa)
0
-0.5
-1
-1.5
-2
Variable
Fixed
0 1 2 3 4 5 6 7 8 9 10
Piston Position
Position (m)
0.2
0.15
0.1
0.05
0
Variable
Fixed
0 1 2 3 4 5 6 7 8 9 10
Time (s)

Closed-Loop Actuator with 2-Way Valve

This example shows the use of a 2-way valve in a closed-loop circuit together with a double-acting hydraulic cylinder, fixed orifice, mass, spring, damper, and control blocks.

The valve is initially opened by 0.42 mm to assure that the cylinder force is close to zero at the start of simulation. The PS Gain block represents a lever through which a control signal is applied to the valve.

Model



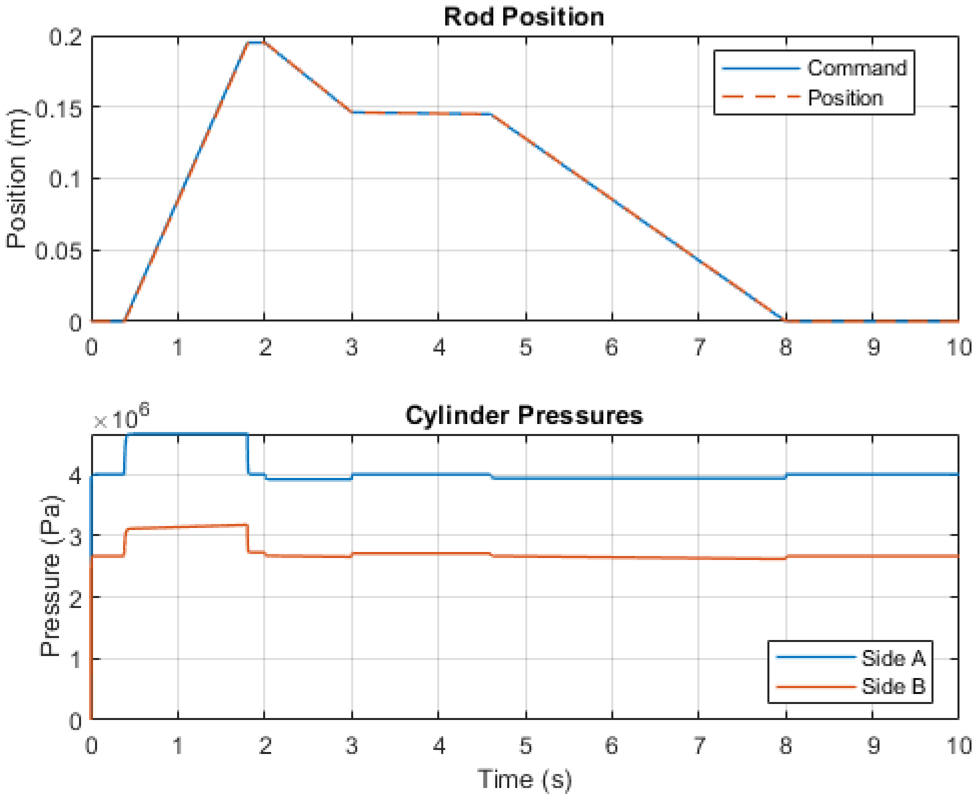
Closed-Loop Actuator with 2-Way Valve

1. Plot positions and pressures (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2004-2022 The MathWorks, Inc.

Simulation Results from Simscape Logging

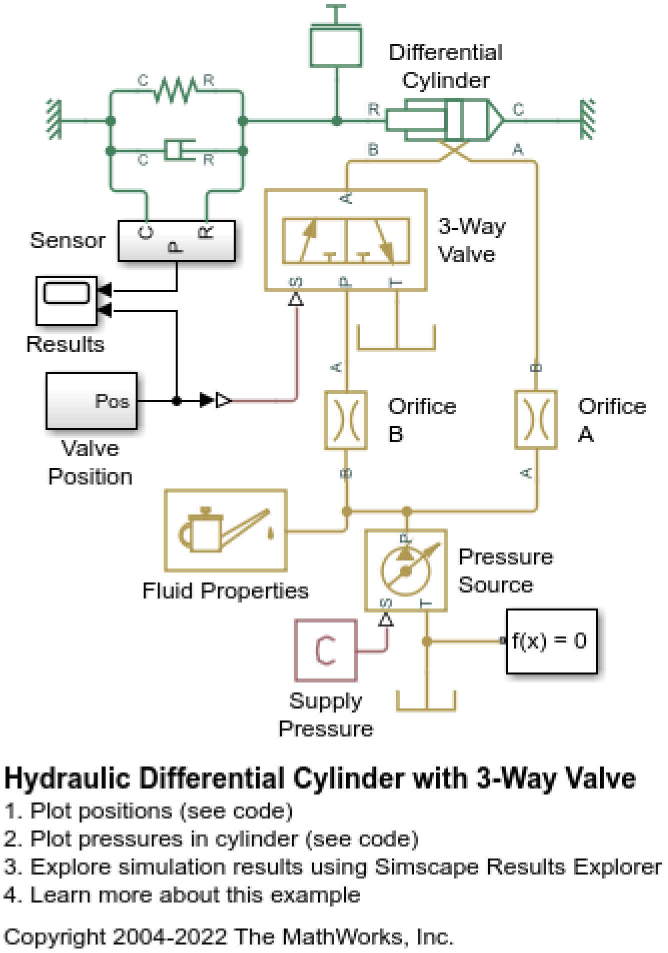
The plot below shows the measured and commanded position of the piston as well as the pressures in the hydraulic cylinder.


Rod Position
Position (m)
0.2
0.15
0.1
0.05
0
0
1
2
3
4
5
6
7
8
9
10
Command
Position
Cylinder Pressures
$\times 10^6$
Pressure (Pa)
4
3
2
1
0
0
1
2
3
4
5
6
7
8
9
10
Time (s)
Side A
Side B

Hydraulic Differential Cylinder with 3-Way Valve

This example shows the use of a 3-way valve to control the motion of a hydraulic differential cylinder. The valve connects cylinder port B either to tank or to cylinder chamber A through the fixed orifice. The cylinder extends when its ports are both connected to the pressure source due to difference in piston areas. Orifice B controls the extension speed. The cylinder retracts if chamber B is connected to tank.

Model

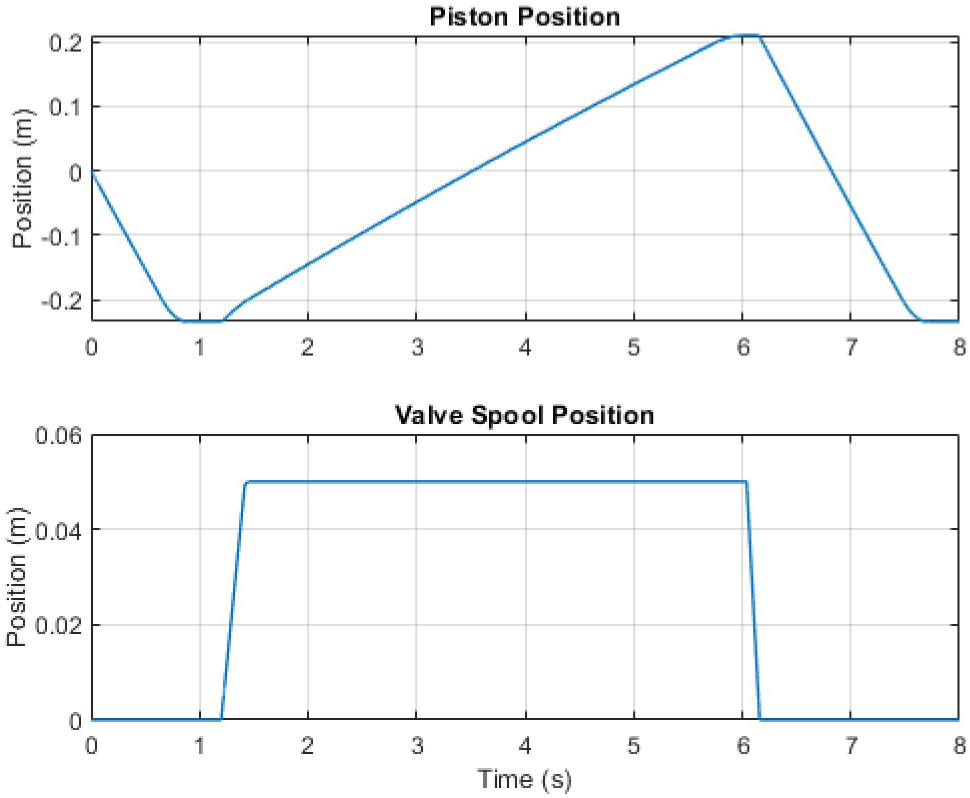


Hydraulic Differential Cylinder with 3-Way Valve

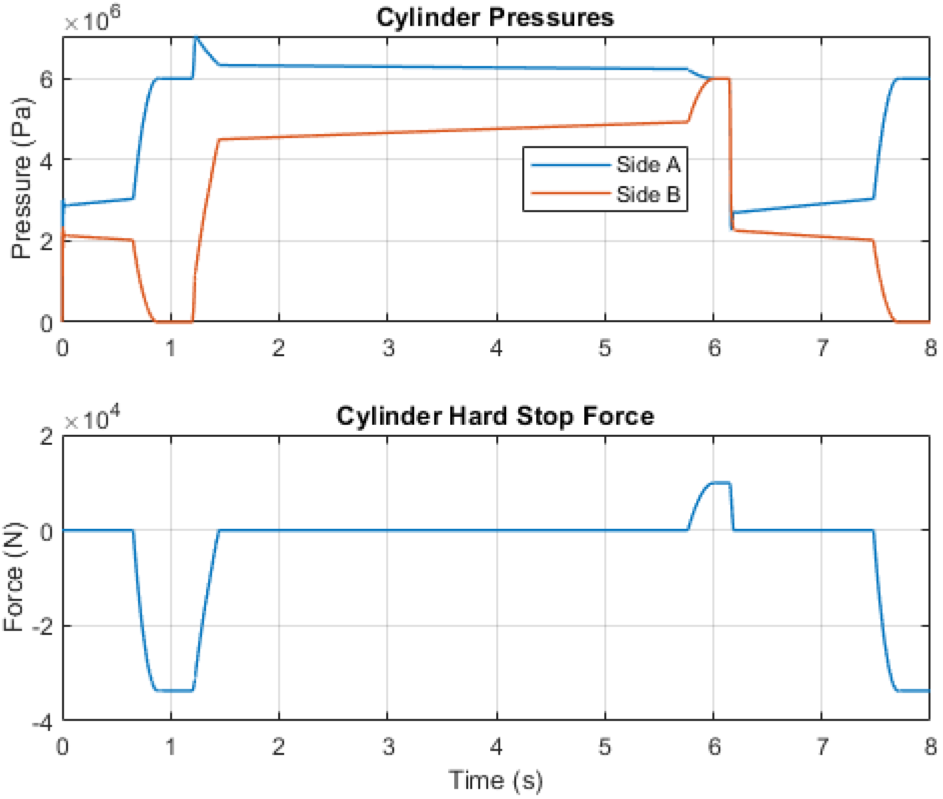
1. Plot positions (see code)
2. Plot pressures in cylinder (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Simulation Results from Simscape Logging

This plot shows the position of the piston (system output) and the position of the valve spool (system input).



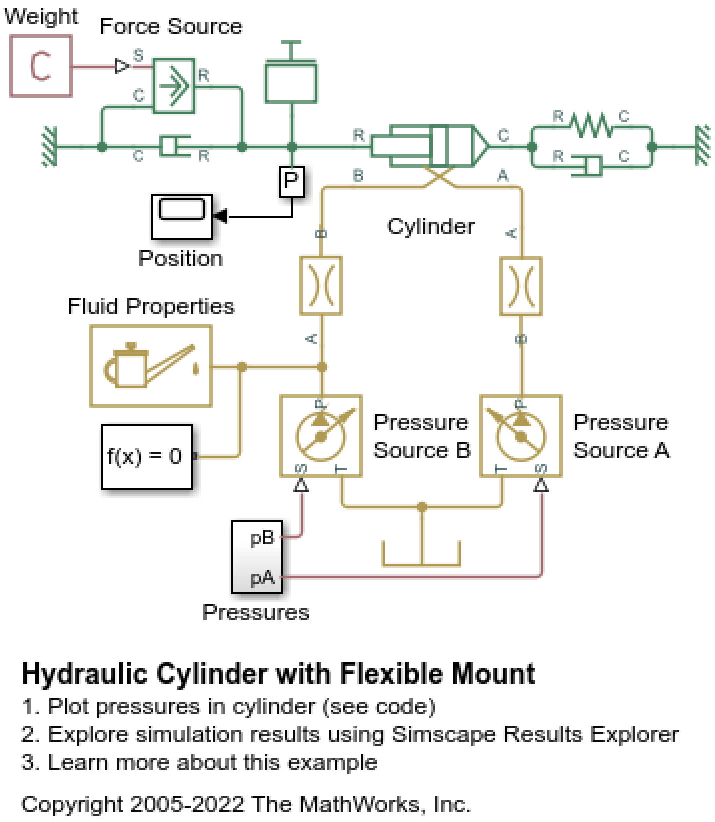
These plots show the pressure inside the cylinder, as well as the force between the piston and the hard stop inside the cylinder.


Cylinder Pressures
×10^6
Pressure (Pa)
0
2
4
6
0
1
2
3
4
5
6
7
8
Side A
Side B
Cylinder Hard Stop Force
×10^4
Force (N)
2
0
-2
-4
0
1
2
3
4
5
6
7
8
Time (s)

Hydraulic Cylinder with Flexible Mount

This example shows the use of a double-acting hydraulic cylinder. The clamping structure is simulated with the spring and damper installed between the cylinder case and the reference. The cylinder performs forward and return strokes caused by the change of pressures at ports A and B. The cylinder load consists of inertia, viscous friction, and constant opposing load of 400 N.

Model



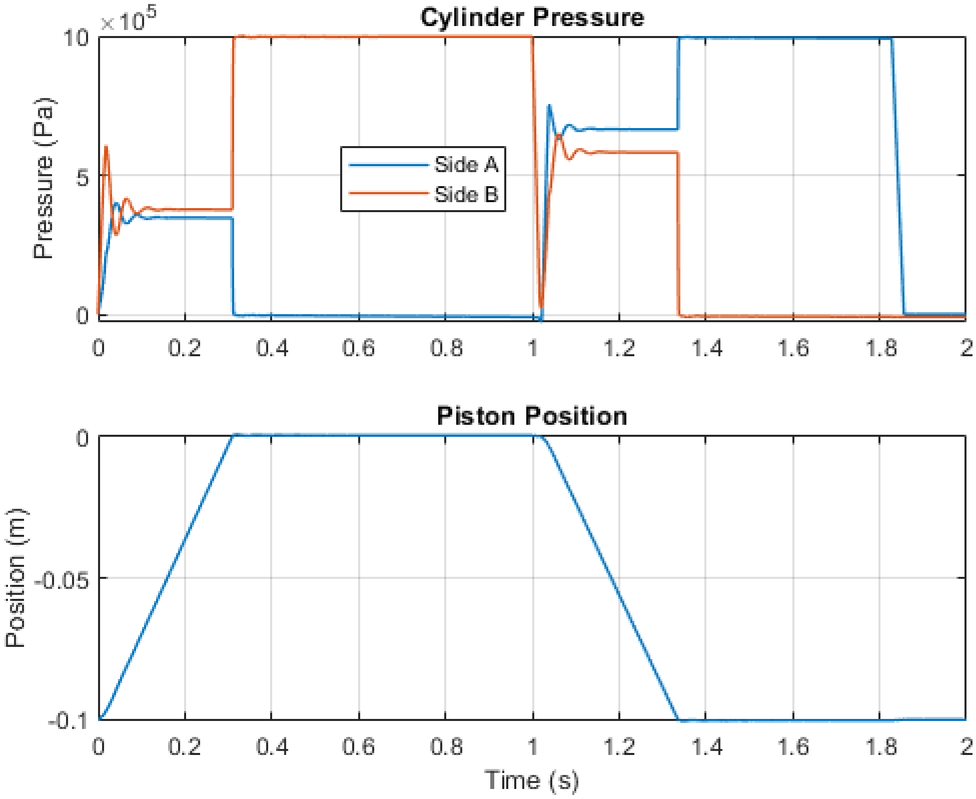
Hydraulic Cylinder with Flexible Mount

1. Plot pressures in cylinder (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2005-2022 The MathWorks, Inc.

Simulation Results from Simscape Logging

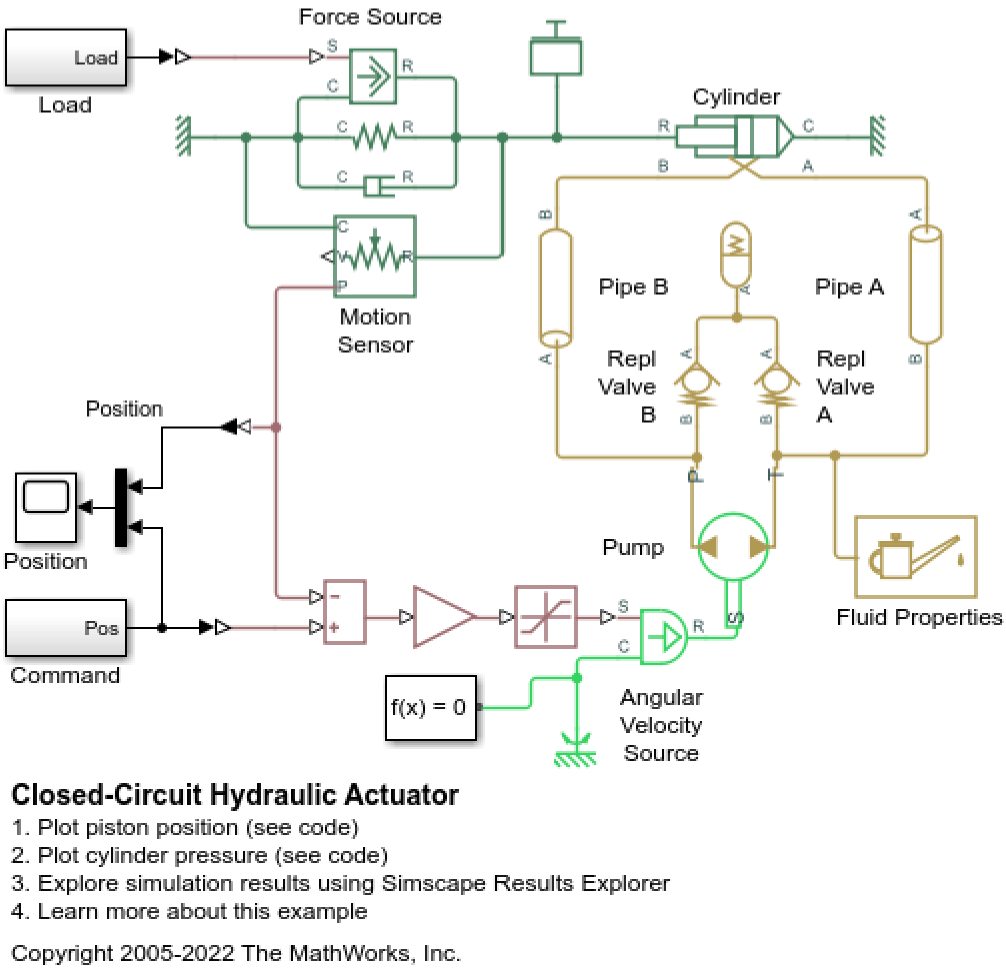
The plots below show the pressures in the cylinder and the position of the piston.


Cylinder Pressure
×10^5
10
5
0
Pressure (Pa)
Side A
Side B
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Piston Position
0
-0.05
-0.1
Position (m)
Time (s)

Closed-Circuit Hydraulic Actuator

This example shows a closed-circuit hydraulic actuator driven by a variable-speed motor. The actuator is arranged as a closed system with two replenishment valves (check valves) and a spring accumulator serving as a replenishment reservoir. The motor velocity is controlled by the difference between the commanded and actual actuator position. The actuator acts against a spring, damper, and a time-varying load.

Model



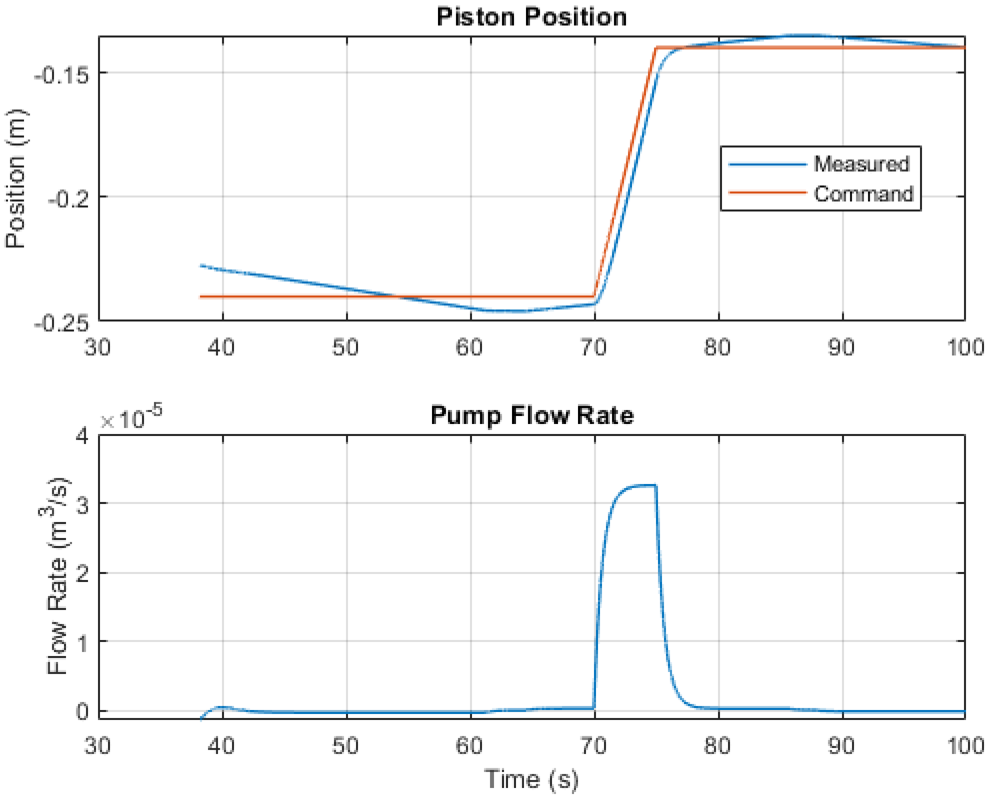
Closed-Circuit Hydraulic Actuator

1. Plot piston position (see code)
2. Plot cylinder pressure (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

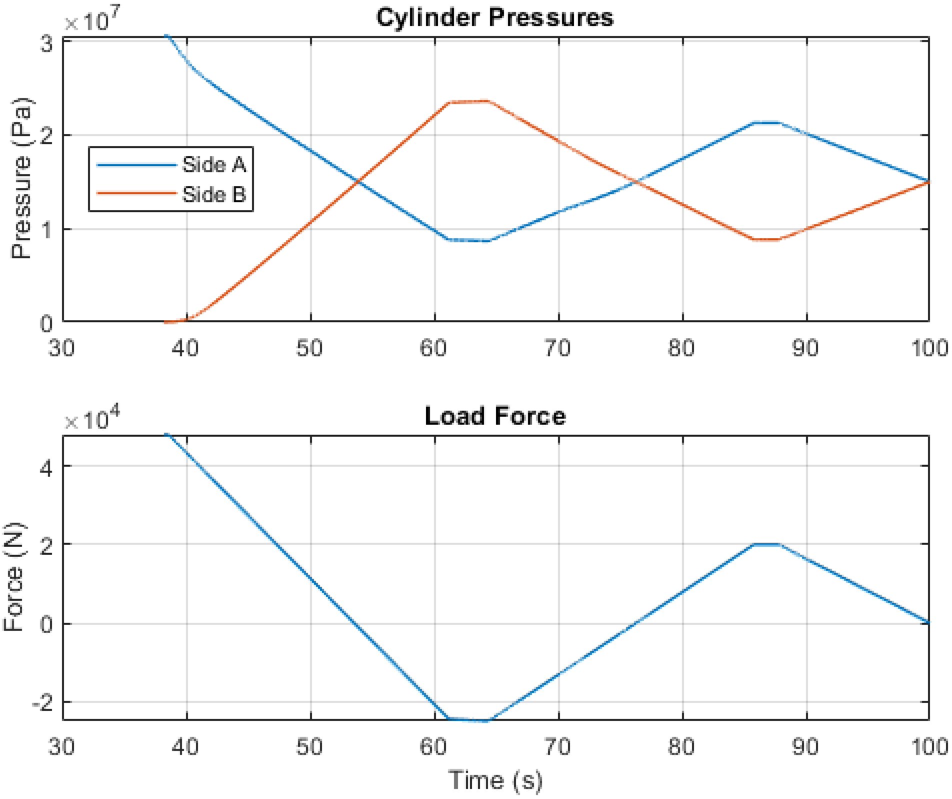
Copyright 2005-2022 The MathWorks, Inc.

Simulation Results from Simscape Logging

This plot shows the position of the piston and the flow rate through the pump.



This plot shows the pressures inside the cylinder and the time-varying load.


Cylinder Pressures
×10^7
3
2
1
0
Pressure (Pa)
Side A
Side B
30
40
50
60
70
80
90
100
Load Force
×10^4
4
2
0
-2
Force (N)
30
40
50
60
70
80
90
100
Time (s)

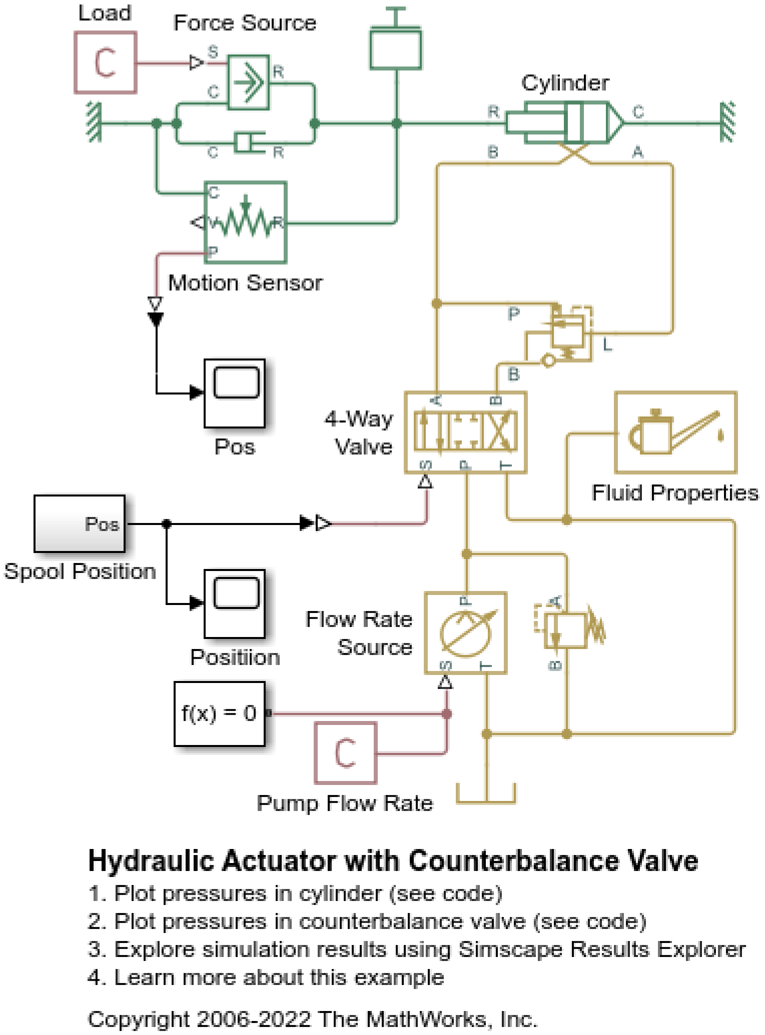
Hydraulic Actuator with Counterbalance Valve

The model consists of a double-acting hydraulic cylinder controlled by an open-center 4-way directional valve and a power unit built of a flow rate source and a pressure-relief valve. The cylinder is loaded with an active force in 10000 N trying to extend the rod. In other words, the cylinder experiences an overriding load while extending. To hold the rod in place while the directional valve is in the neutral position and prevent losing control of the cylinder extension, the counterbalance valve is used. The valve controls the return flow from port A. The valve pilot port P is connected to cylinder port B, thus making it impossible for the cylinder to move until pressure at port B builds up to certain level.

To hold the load, the counterbalance valve must be set to 1.3 * 10000/ 5e-3 = 26e5 Pa, where 1.3 is the safety coefficient, 10000 N is the load, 5e-5 m^2 is the effective piston area in the rod chamber. The cylinder starts moving as the pilot pressure builds up to approximately p = (p_s - p_L) / (k_p + k_c), where p_s is the valve setting pressure, p_L is the load pressure (p_L = L/A_r), L is the load, A_r is the rod chamber effective area (piston side B), k_p is the pilot ratio (k_p is set to 3 in the model), and k_c is the ratio between the cylinder blank chamber area to the rod chamber area ((piston area side B/piston area side A) is set to 2 in the model). As a result, the load starts moving at p = (26e5 - 100000/5e-3) / (3 + 2) = 1.2e5 Pa.

The system is simulated during a 10 second cycle, within which the rod extends by about 0.06 m and then returns to initial position. The counterbalance valve maintains enough pressure in the cylinder chamber to balance the load while the directional valve is in the neutral position.

Model



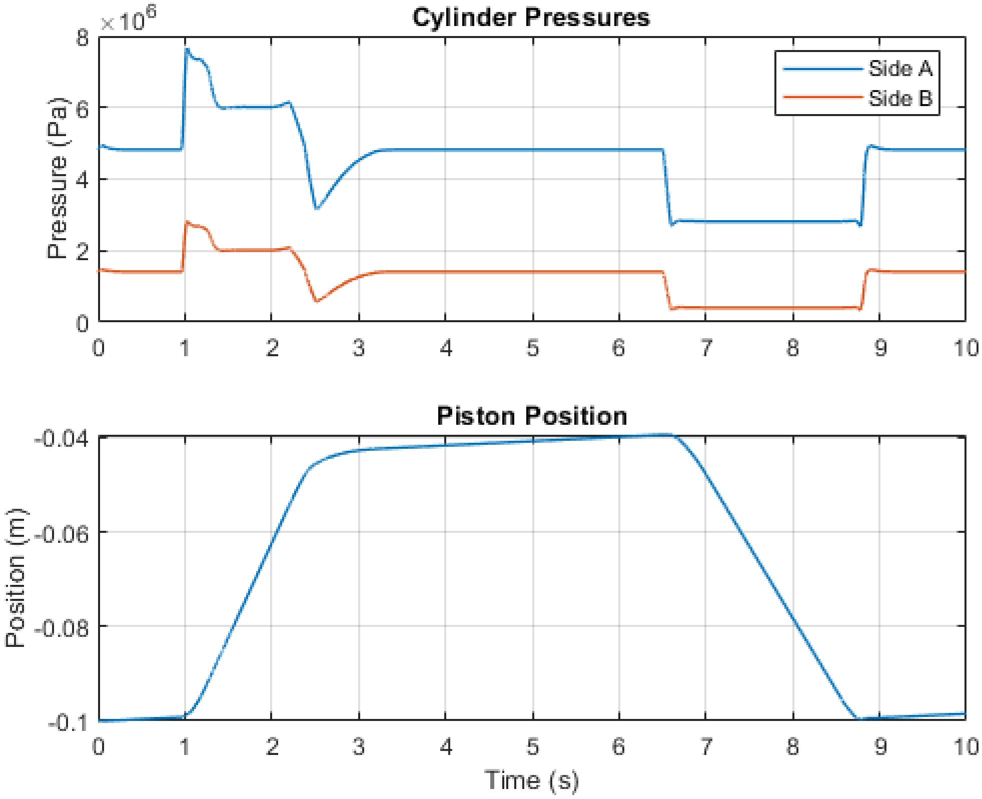
Hydraulic Actuator with Counterbalance Valve

1. Plot pressures in cylinder (see code)
2. Plot pressures in counterbalance valve (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

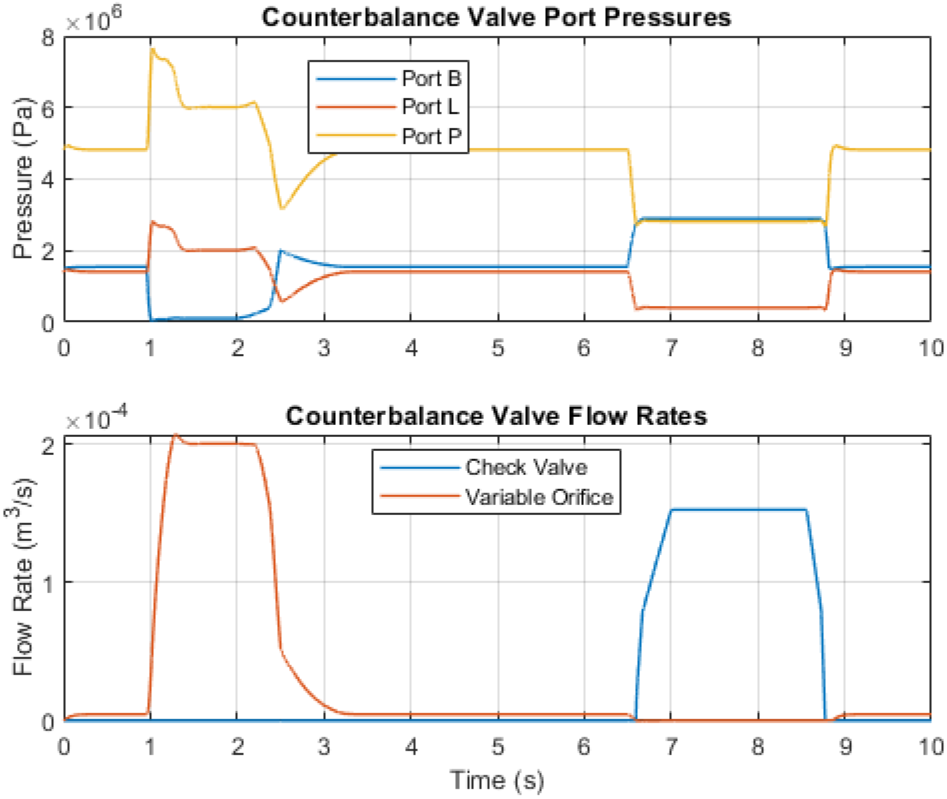
Copyright 2006-2022 The MathWorks, Inc.

Simulation Results from Simscape Logging

The plots below show the pressures in the cylinder and the position of the piston.



The plots below show the port pressures and flow rates for the counterbalance valve. During the first part of the cycle, flow rate through the variable orifice is observed, indicating that the pressures at ports P and L create enough force to overcome the spring force and open the valve. In the second part of the cycle, flow is observed through the check valve in the counterbalance valve.


Counterbalance Valve Port Pressures
×10^6
Pressure (Pa)
0
2
4
6
8
Port B
Port L
Port P
0
1
2
3
4
5
6
7
8
9
10
Counterbalance Valve Flow Rates
×10^-4
Flow Rate (m^3/s)
0
1
2
Check Valve
Variable Orifice
0
1
2
3
4
5
6
7
8
9
10
Time (s)

Hydraulic Actuator with Dual Counterbalance Valves

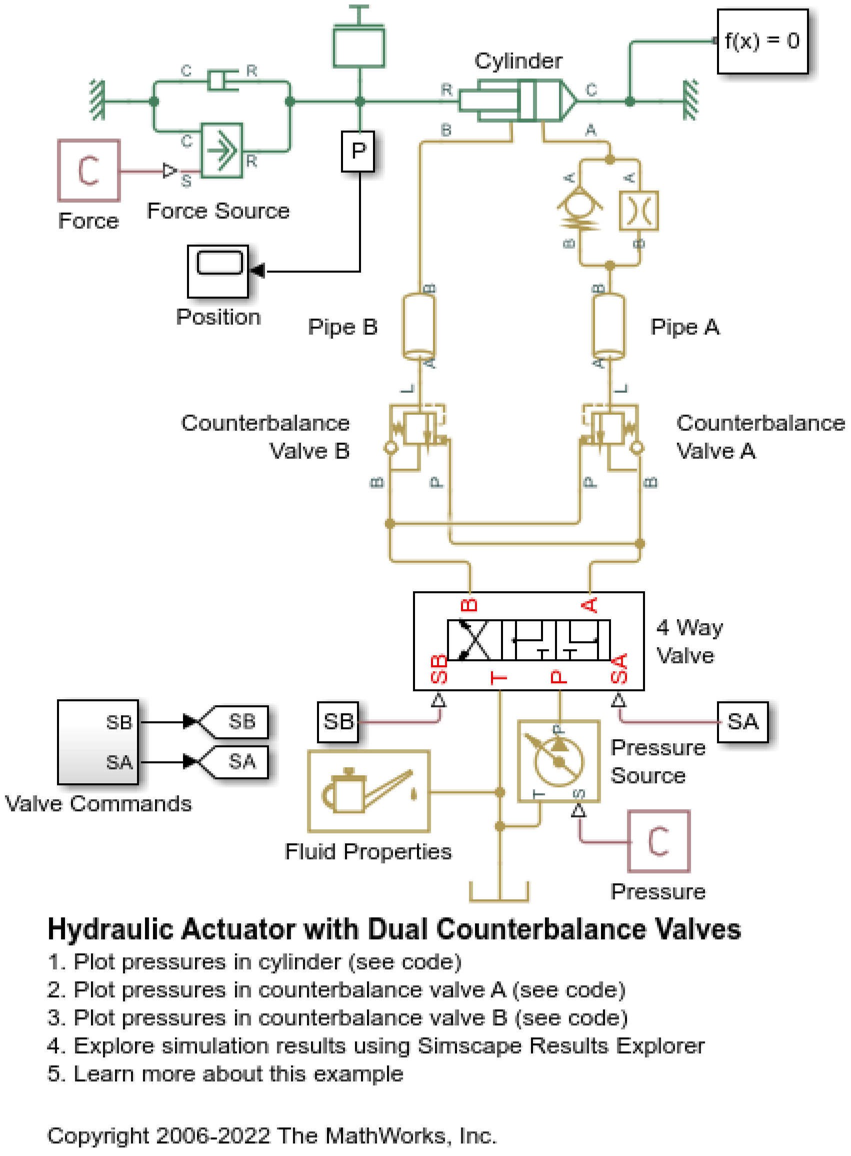
This example shows an actuator built of a double-acting hydraulic cylinder, directional valve, flow control, block of counterbalance valves, power unit, and a control unit.

The cylinder is loaded with an overriding load, which requires the use of counterbalance valves to prevent the load from creeping when the directional valve is in the neutral position. The actuator is equipped with a custom model of a 3-position, 4-way directional valve. The valve connects actuator ports A and B to tank in neutral position, while blocking the pressure port P. Ports A and B are connected to port P if electromagnet A is energized. This causes the cylinder to extend due to the difference in the cylinder effective areas. The piston retracts as electromagnet B is energized.

The counterbalance valves allows free flow from port B to port L through the check valve, but blocks the flow in the opposite direction unless pressure is applied to port P. When sufficient pressure is applied, a variable orifice opens.

The simulated actuator cycle starts with the piston rod at initial position. At 0.65 seconds electromagnet A is energized, causing the piston to extend until it reaches the end of the stroke. On its way forward, both bypass orifices are open and the cylinder differential connection is used to utilize the difference between piston areas. At 11.7 seconds electromagnet B is energized and the piston retracts to its initial position.

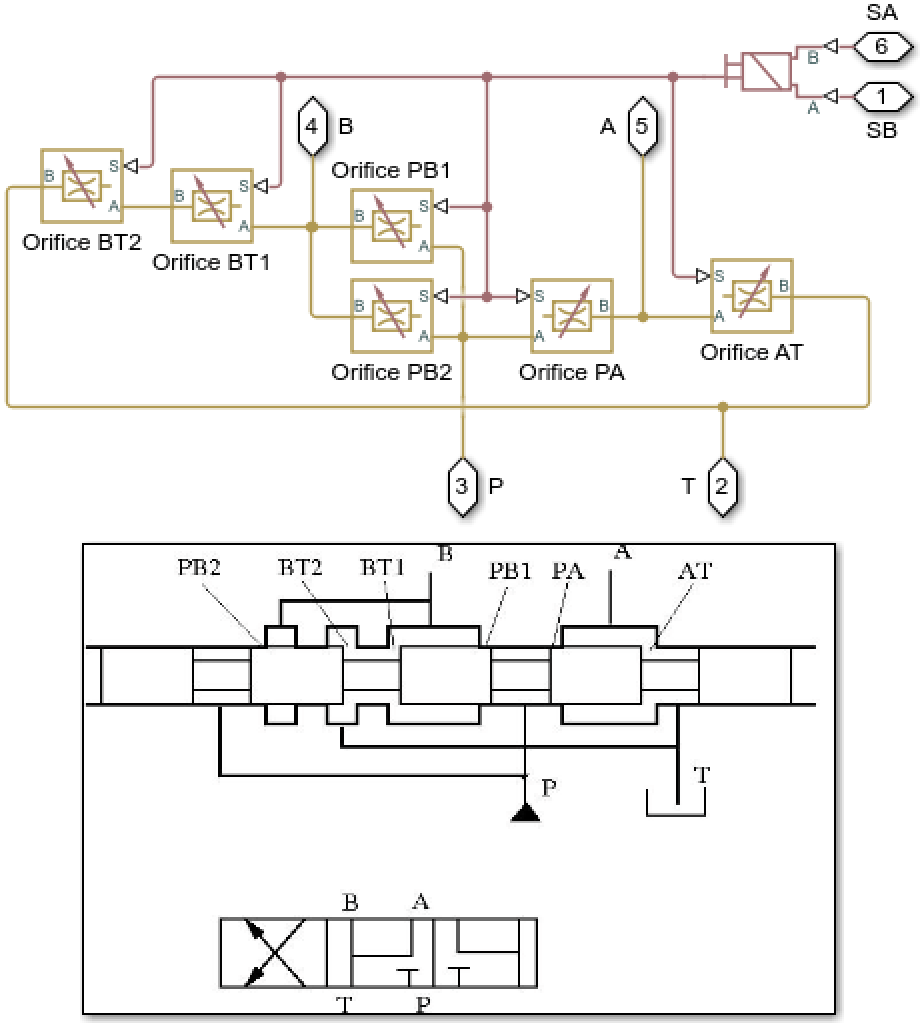
Model



Hydraulic Actuator with Dual Counterbalance Valves

1. Plot pressures in cylinder (see code)
2. Plot pressures in counterbalance valve A (see code)
3. Plot pressures in counterbalance valve B (see code)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example

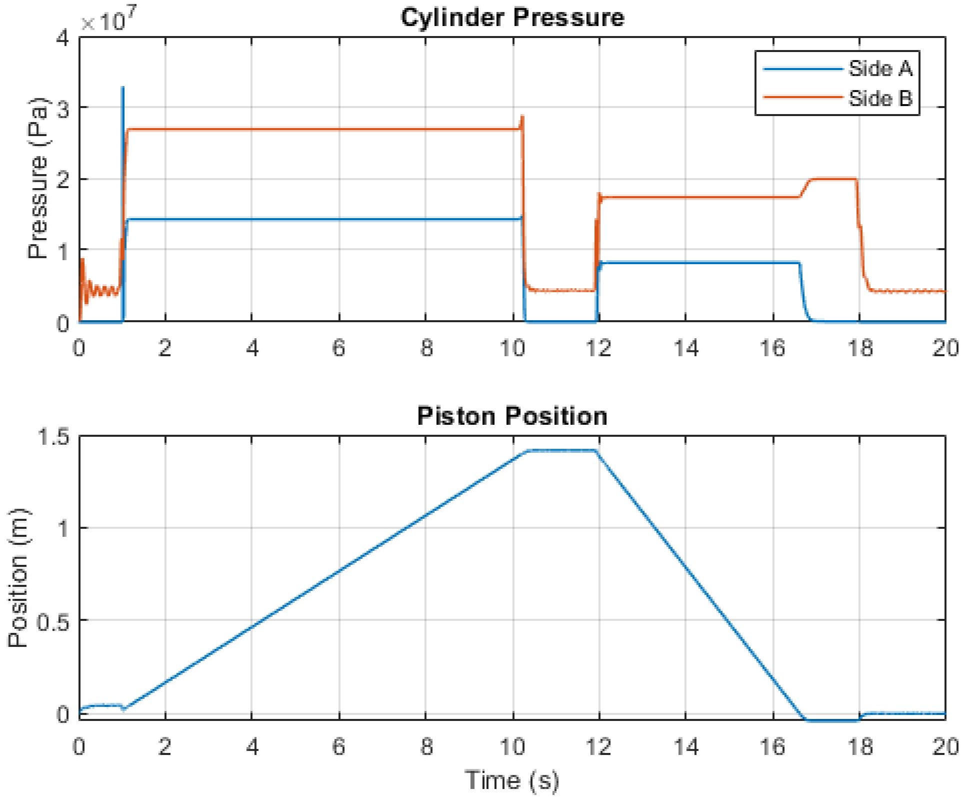
4 Way Valve Subsystem



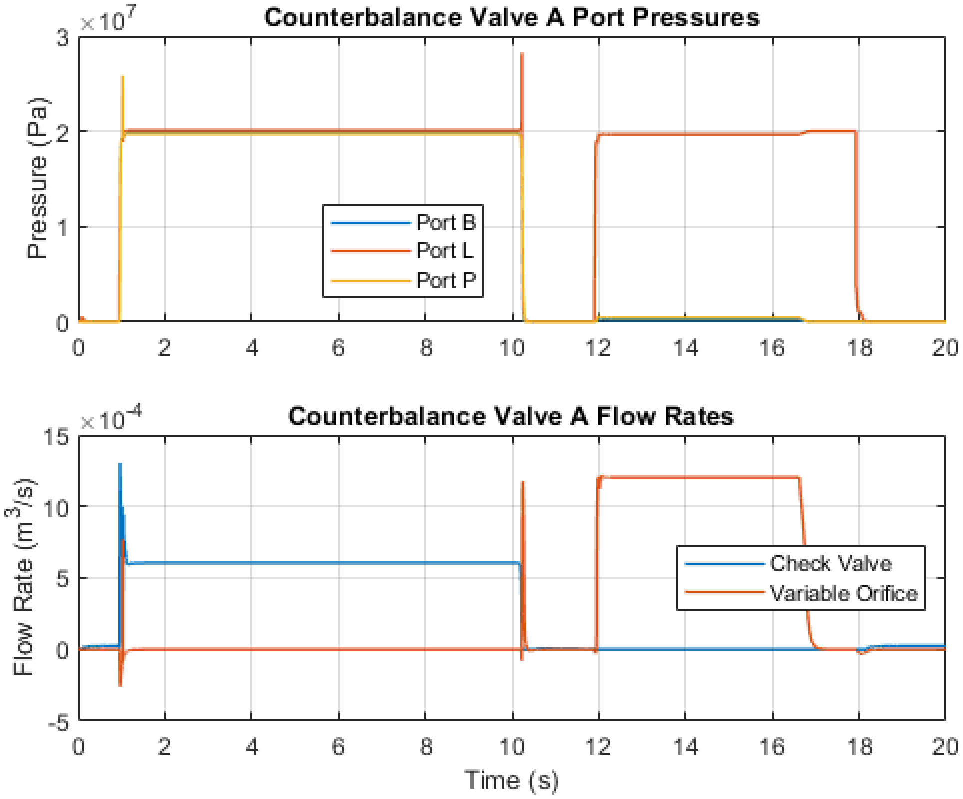
Directional Valve Schematic

Simulation Results from Simscape Logging

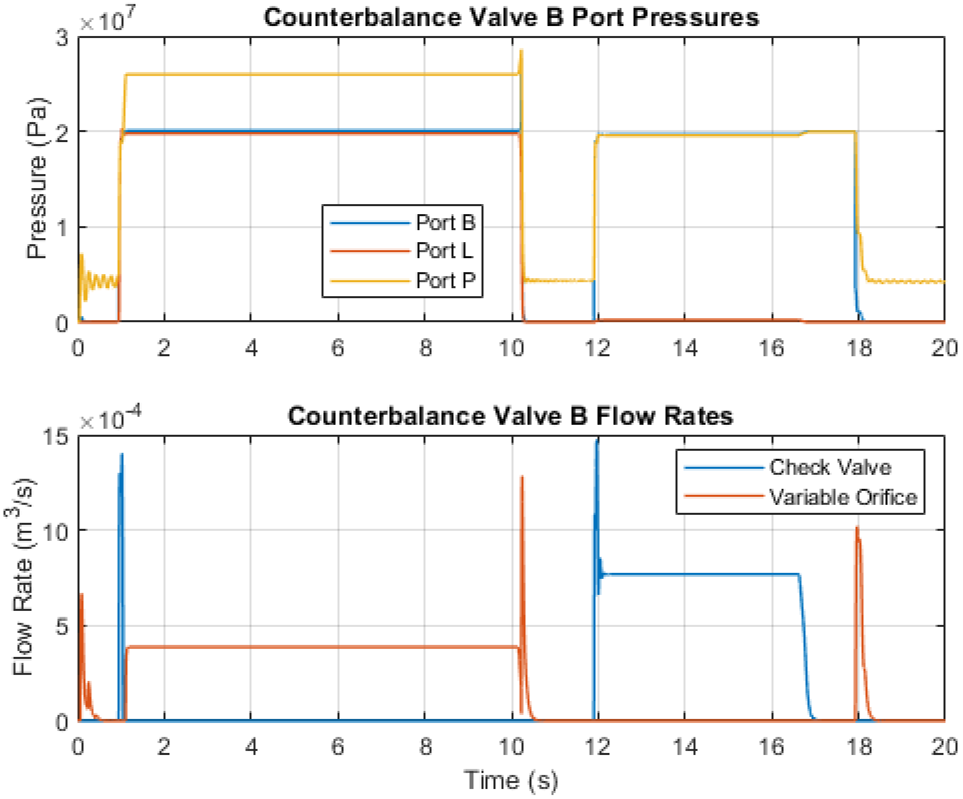
The plots below show the pressures in the cylinder and the position of the piston.



The plots below show the port pressures and flow rates for counterbalance valve A. In the first part of the cycle, flow is observed through the check valve in the counterbalance valve. During the second part of the cycle, flow rate through the variable orifice is observed, indicating that the ports at P and L create enough force to overcome the spring force and open the valve.



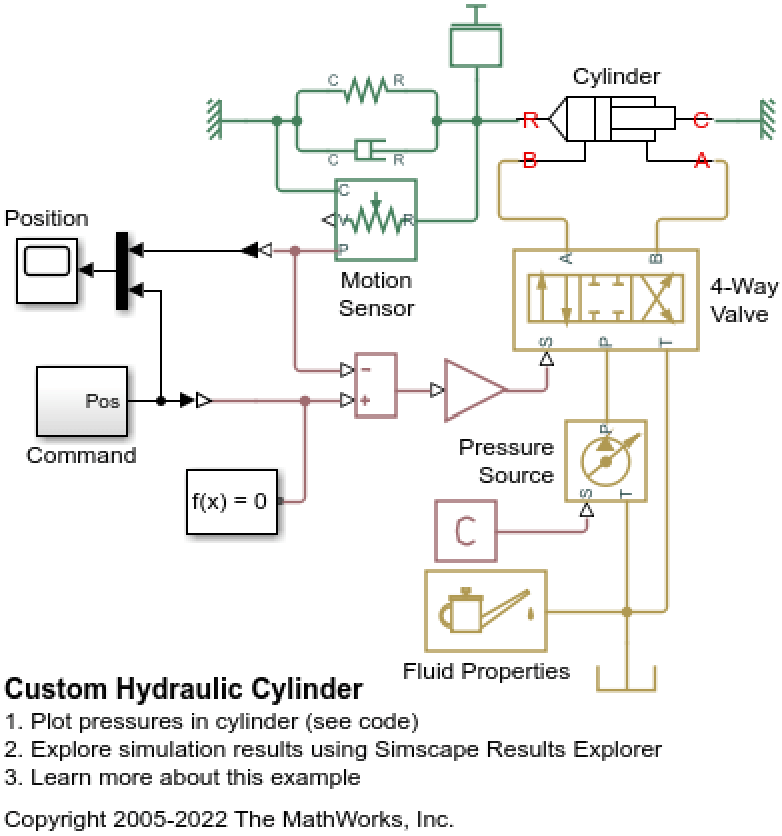
The plots below show the port pressures and flow rates for counterbalance valve B. During the first part of the cycle, flow rate through the variable orifice is observed, indicating that the ports at P and L create enough force to overcome the spring force and open the valve. In the second part of the cycle, flow is observed through the check valve in the counterbalance valve.


Counterbalance Valve B Port Pressures
×10^7
Pressure (Pa)
3
2
1
0
0
2
4
6
8
10
12
14
16
18
20
Port B
Port L
Port P
Counterbalance Valve B Flow Rates
×10^-4
Flow Rate (m^3/s)
15
10
5
0
Check Valve
Variable Orifice
Time (s)

Custom Hydraulic Cylinder

This example shows a custom hydraulic cylinder model that omits the effect of fluid compressibility in the cylinder chambers. The double-acting cylinder connected to a 4-way valve to model a simple closed-loop actuator.

Model

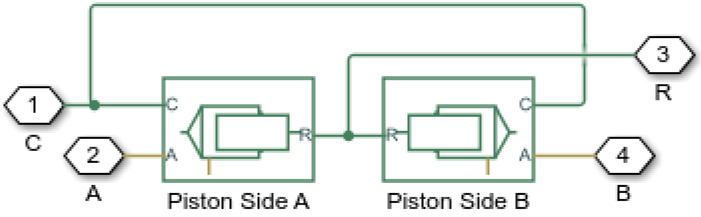


Custom Hydraulic Cylinder

1. Plot pressures in cylinder (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

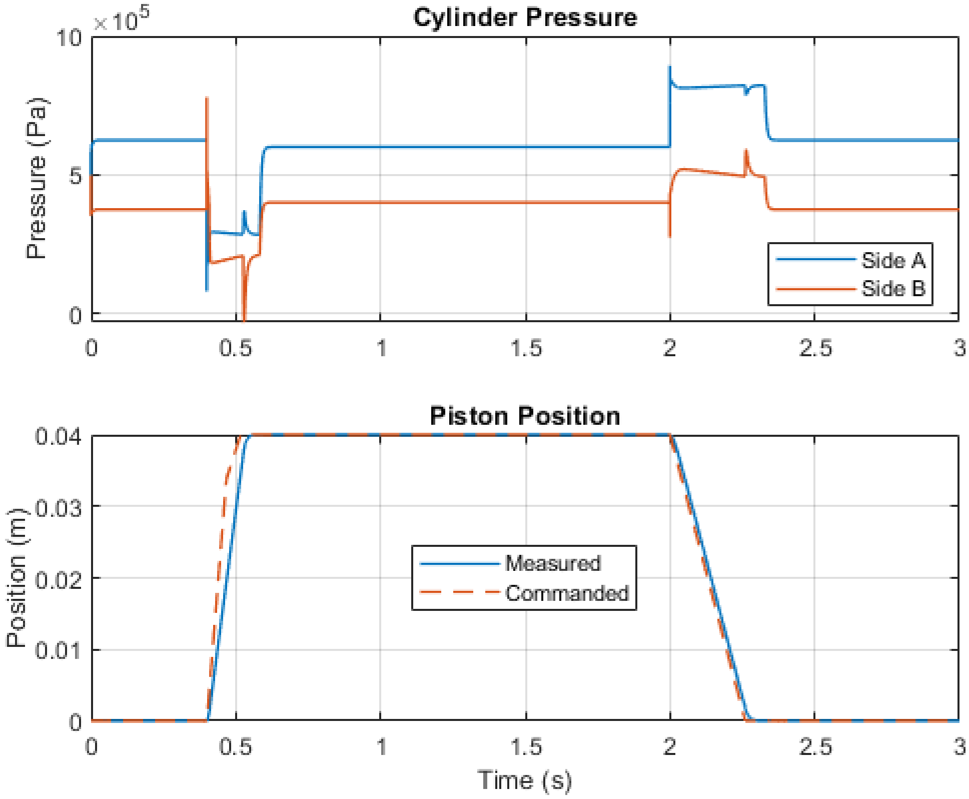
Copyright 2005-2022 The MathWorks, Inc.

Cylinder Subsystem



Simulation Results from Simscape Logging

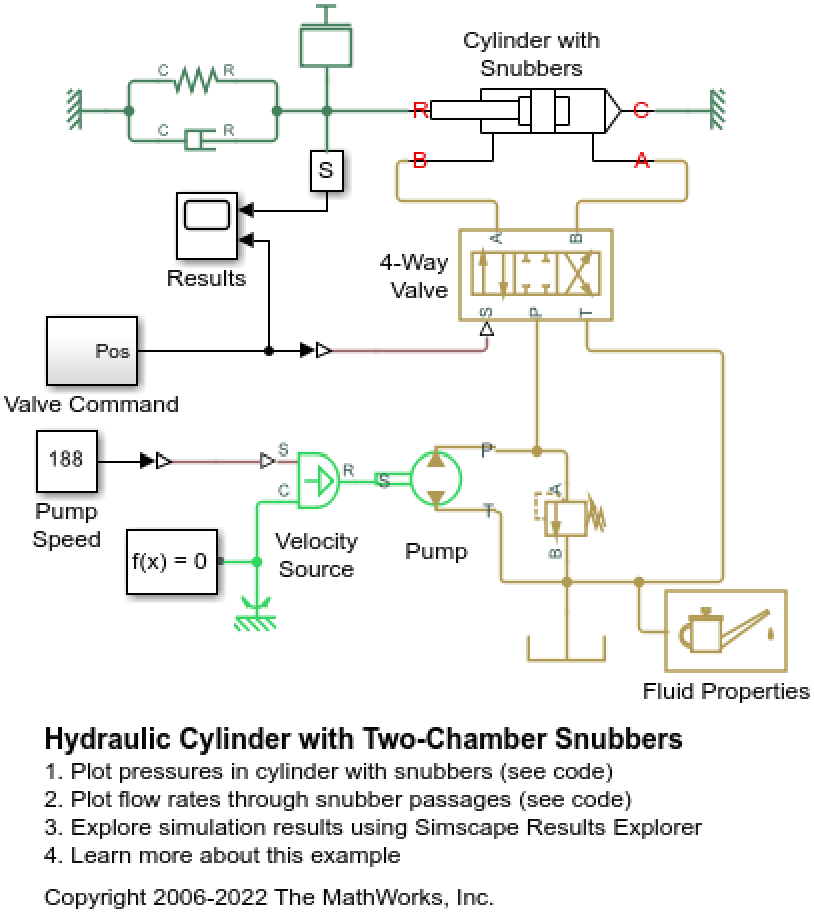
The plots below show the pressures in the cylinder and the position of the piston.


Cylinder Pressure
×10^5
10
5
0
Pressure (Pa)
0
0.5
1
1.5
2
2.5
3
Side A
Side B
Piston Position
0.04
0.03
0.02
0.01
0
Position (m)
Measured
Commanded
Time (s)

Hydraulic Cylinder with Two-Chamber Snubbers

This example shows a hydraulic cylinder equipped with a custom model of snubbers (cushions) on both sides of the cylinder. The snubbers are implemented as two-chamber cushions separated by the cushioning bush as the piston comes close to the end of the stroke. The cushioning provides hydraulic braking as the cylinder reaches the end of its stroke, absorbing some of the kinetic energy of the system before the piston reaches the end stop.

Model

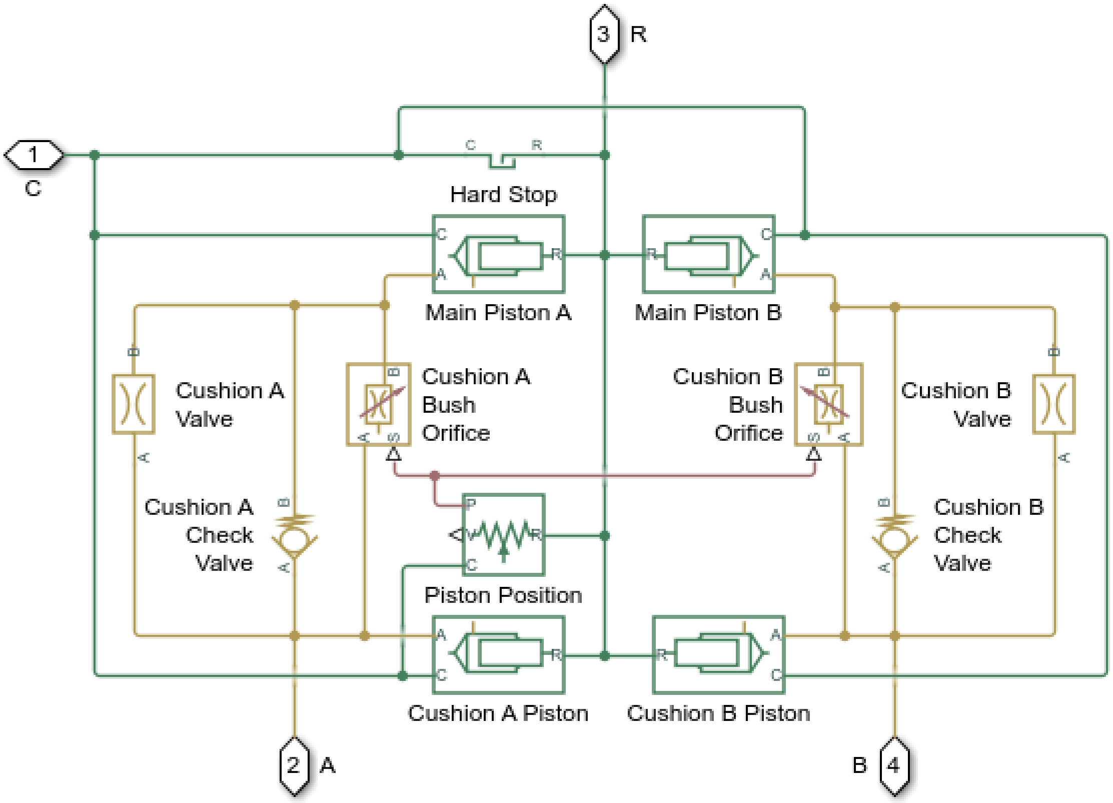


Hydraulic Cylinder with Two-Chamber Snubbers

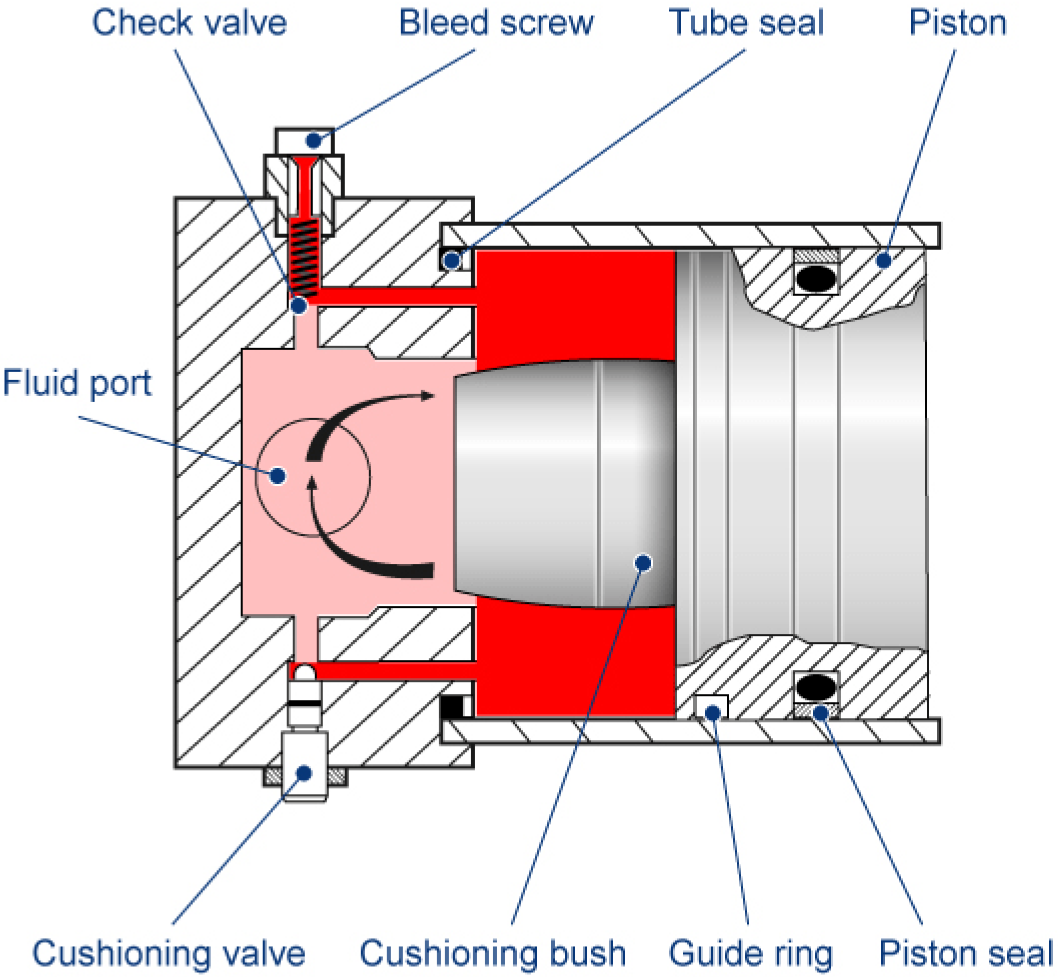
1. Plot pressures in cylinder with snubbers (see code)
2. Plot flow rates through snubber passages (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2006-2022 The MathWorks, Inc.

Cylinder with Snubbers Subsystem

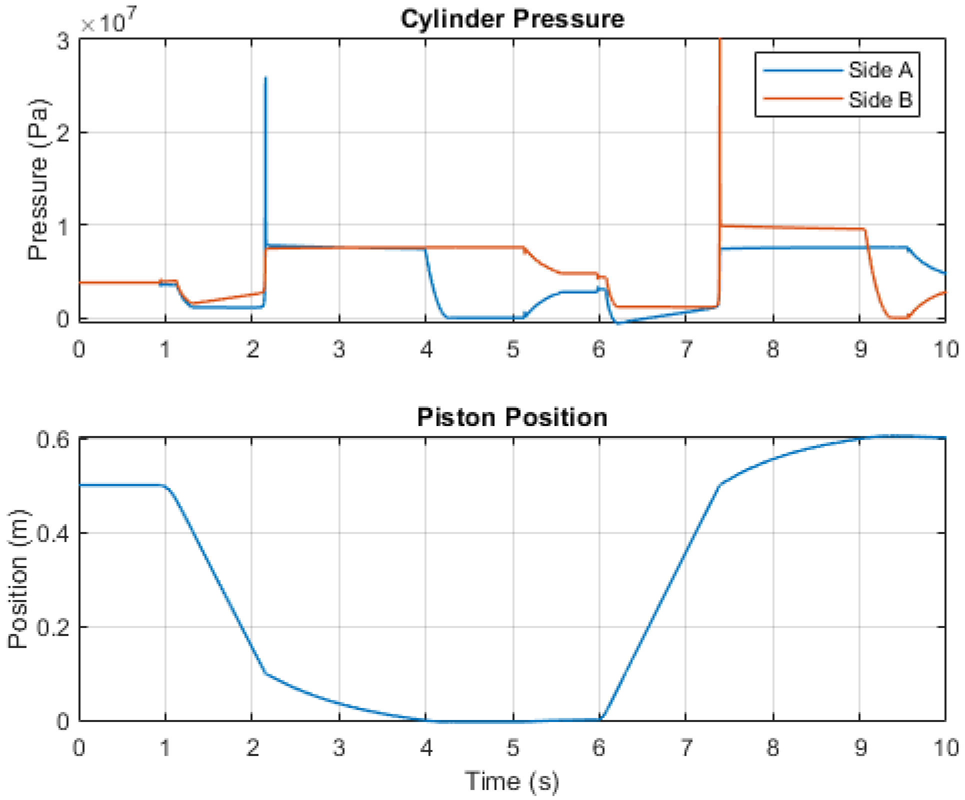


The following figure shows the schematic diagram of one end of the hydraulic cylinder with the snubber passages:

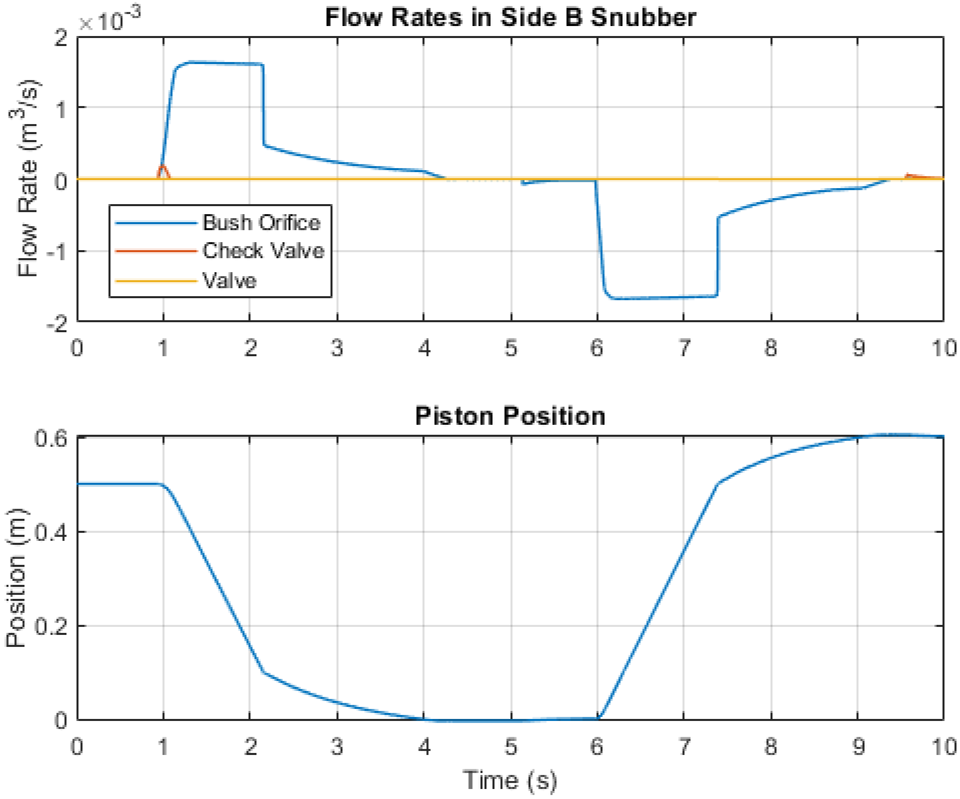


Simulation Results from Simscape Logging

The plots below show the pressures in the cylinder and the position of the piston.



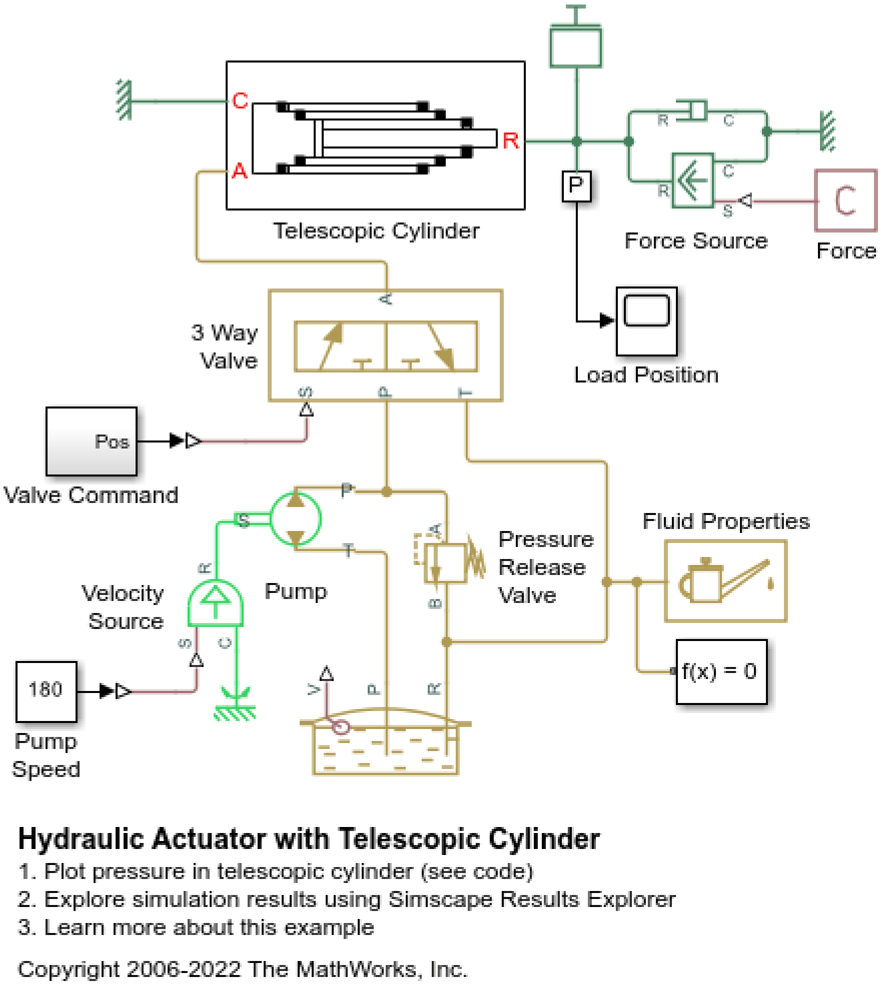
The plots below show the flow rate in the fluid passages that make up the snubber on side B of the cylinder. As the four-way valve connects the pump to side B, the check valve opens to permit the hydraulic fluid to flow out of the snubber and into the main chamber. The piston moves smoothly away from the hard stop, opening a larger passage between the fluid port and the chamber and the check valve closes. During the second half of the cycle, a very small amount of fluid flow passes through the cushion valve, allowing the piston to coast more smoothly into the hard stop.


Flow Rates in Side B Snubber
×10-3
Flow Rate (m3/s)
2
1
0
-1
-2
0
1
2
3
4
5
6
7
8
9
10
Bush Orifice
Check Valve
Valve
Piston Position
Position (m)
0.6
0.4
0.2
0
Time (s)

Hydraulic Actuator with Telescopic Cylinder

This example shows an actuator built around a telescopic hydraulic cylinder, which is equipped with three rods interacting with each other through hard stops. The effective areas of the rods are set to 20, 16, and 12 cm^2 respectively, which causes the first rod to move first, followed by the second, and then by the third rod. The actuator is controlled by a 2-position 3-way valve, which connects the actuator chamber to the tank at neutral position. As the control signal is applied to the valve, the actuator chamber is connected to the pump and the actuator starts extending. The rods are retracted by external load when the chamber is connected to the tank. The power unit relief valve is set to 50 bar, and maximum load to be handled by the actuator is 2500 N.

Model

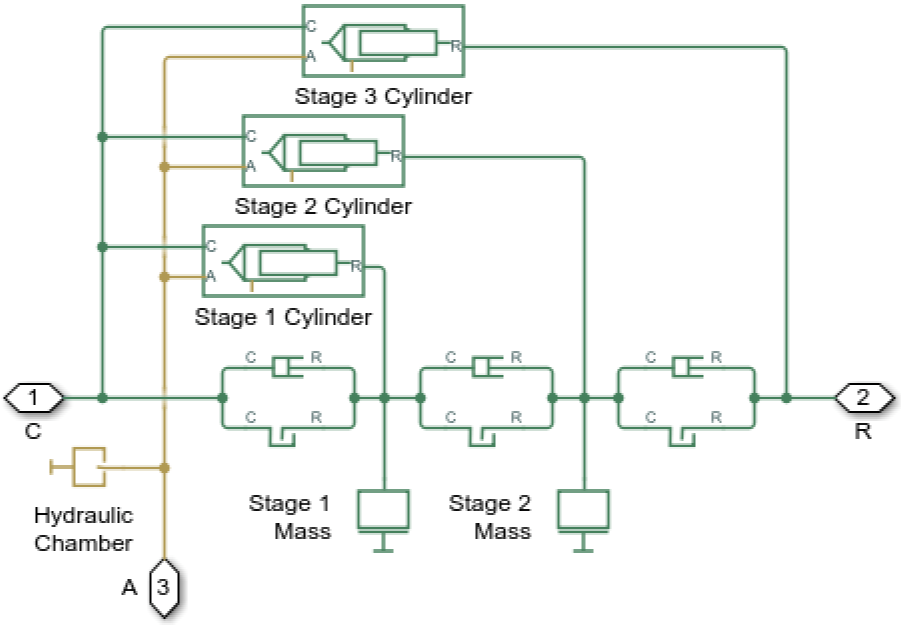


Hydraulic Actuator with Telescopic Cylinder

1. Plot pressure in telescopic cylinder (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

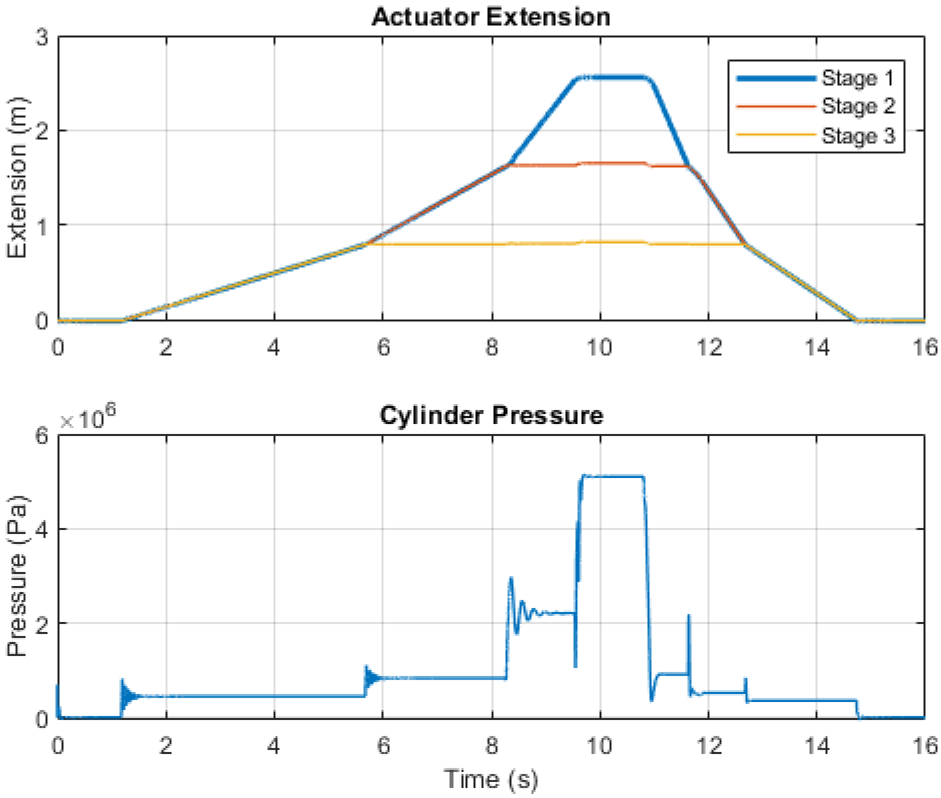
Copyright 2006-2022 The MathWorks, Inc.

Telescopic Cylinder Subsystem



Simulation Results from Simscape Logging

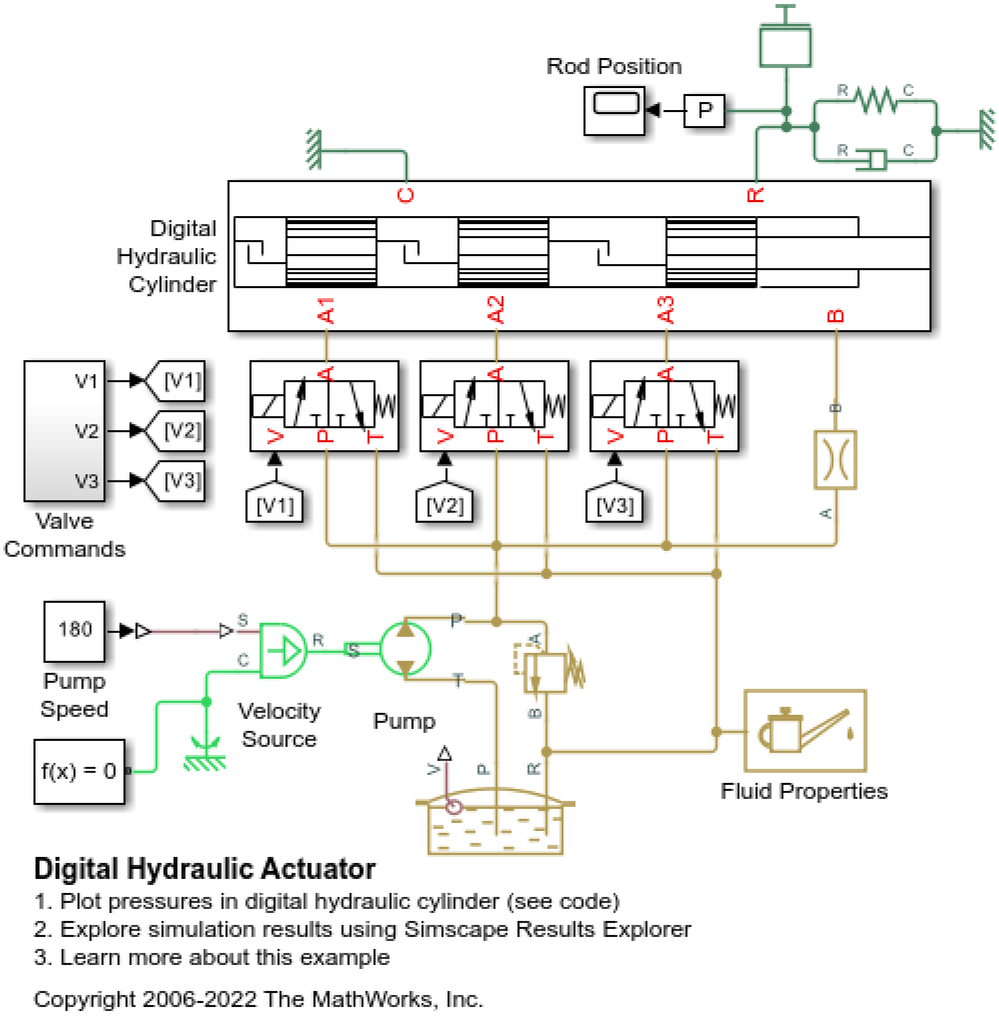
The plots below show the pressures in the cylinder and the extensions of each stage.


Actuator Extension
Extension (m)
3
2
1
0
0
2
4
6
8
10
12
14
16
Stage 1
Stage 2
Stage 3
Cylinder Pressure
×10^6
Pressure (Pa)
6
4
2
0
0
2
4
6
8
10
12
14
16
Time (s)

Digital Hydraulic Actuator

This example shows a digital hydraulic actuator that consists of 3 double-acting cylinders mounted in the same shell and interconnected through the hard stops. The gap between the leftmost piston and the shell is equal to 2^0*base_length, the gap between the first and the second pistons is set to 2^1*base_length, and the gap between the second and the third pistons is set to 2^2*base_length. The piston area in the rightmost chamber is half the size of the areas of the other chambers. The rightmost chamber is permanently connected to the pump while the other three chambers are connected either to pump or tank depending on the 2-position 3-way electro-hydraulic valves. As a result, 2^3 discrete positions can be reached within 2^3*base_length range.

Model

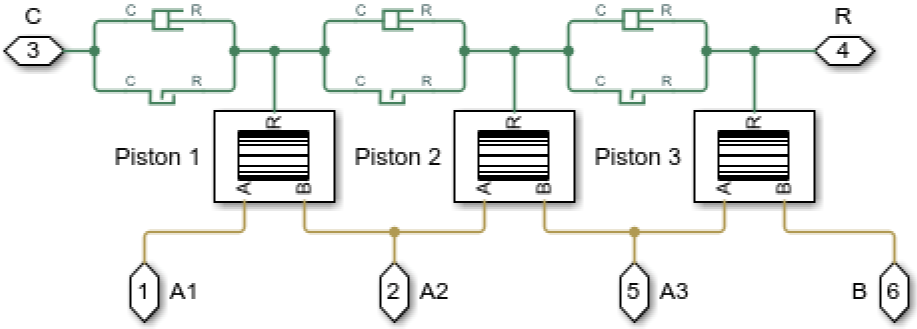


Digital Hydraulic Actuator

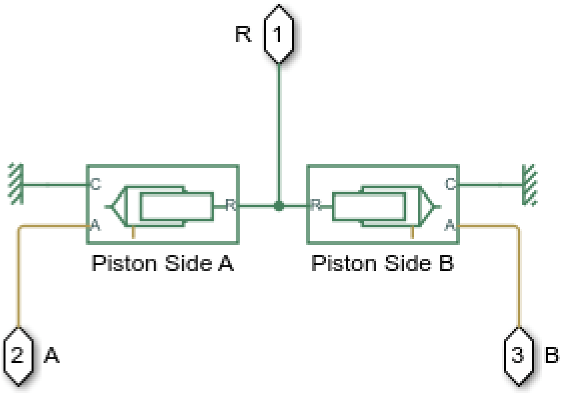
1. Plot pressures in digital hydraulic cylinder (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2006-2022 The MathWorks, Inc.

Digital Hydraulic Cylinder Subsystem

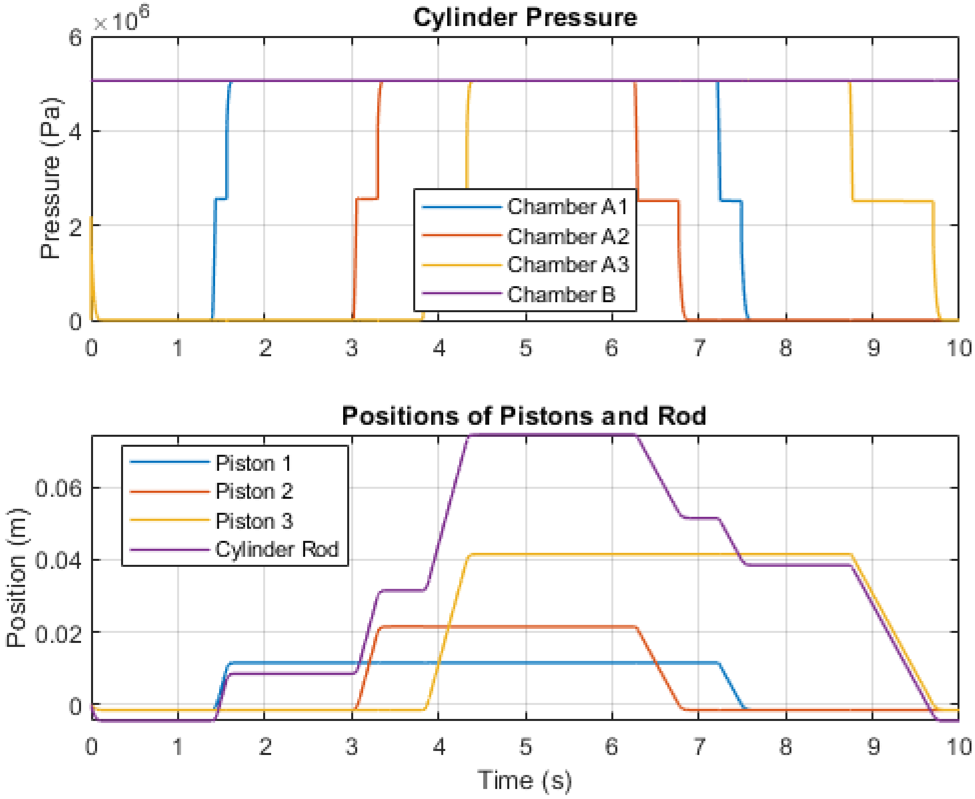


Piston 1 Subsystem



Simulation Results from Simscape Logging

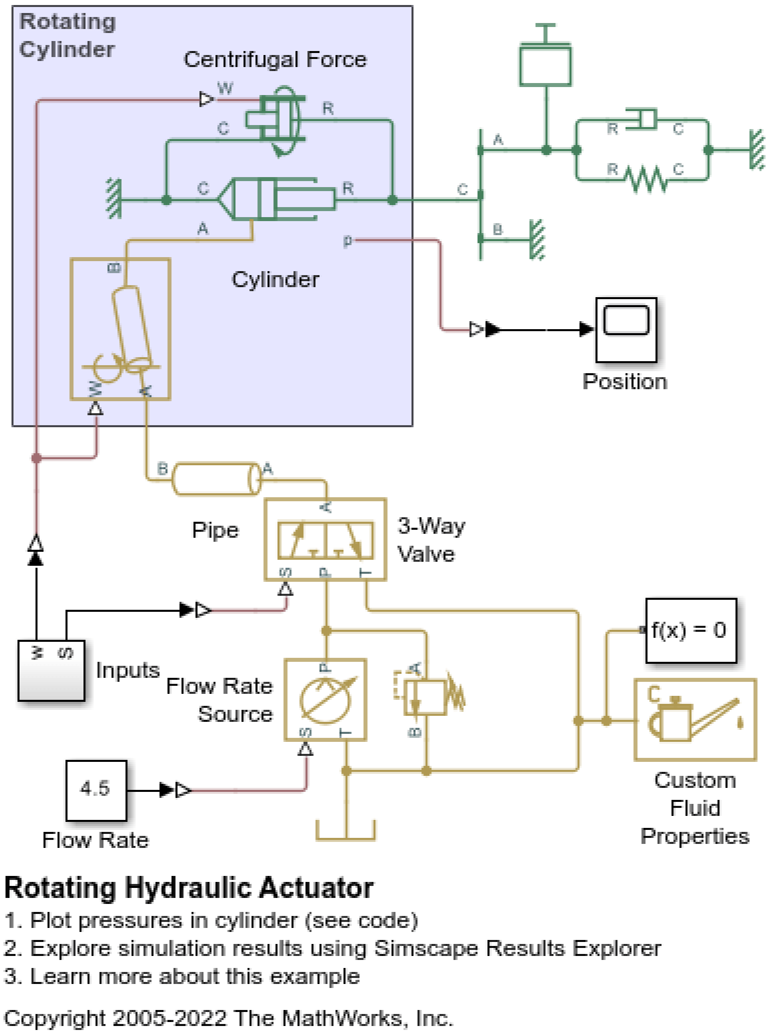
The plots below show the pressures in each cylinder chamber and the extension of each cylinder. Six of the eight possible positions are shown during this cycle.


Cylinder Pressure
×10^6
Pressure (Pa)
6
4
2
0
0
1
2
3
4
5
6
7
8
9
10
Chamber A1
Chamber A2
Chamber A3
Chamber B
Positions of Pistons and Rod
Position (m)
0.06
0.04
0.02
0
Piston 1
Piston 2
Piston 3
Cylinder Rod
0
1
2
3
4
5
6
7
8
9
10
Time (s)

Rotating Hydraulic Actuator

This example shows a typical hydraulic cylinder actuator used to operate friction clutches, brakes and other devices installed on rotating shafts. The key element of the actuator is a piston that moves back and forth under pressure provided through the central drill in the shaft and through channels in the clutch. In the example, the actuator acts against a preloaded spring, which tends to push the piston against the clutch wall.

Model

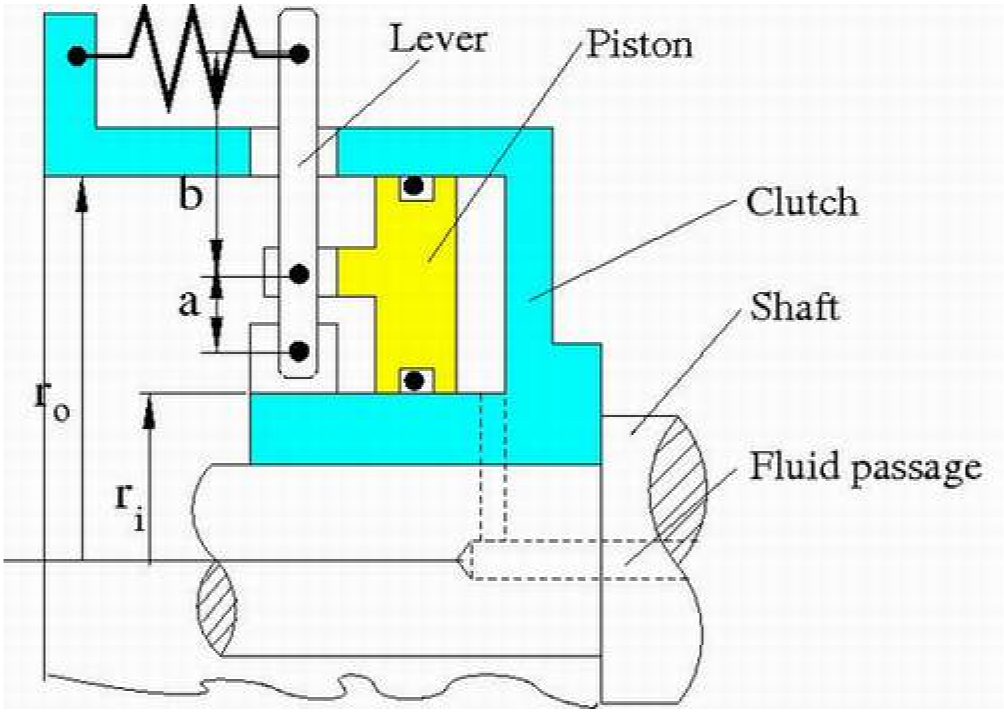


Rotating Hydraulic Actuator

1. Plot pressures in cylinder (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2005-2022 The MathWorks, Inc.

The following figure shows the schematic diagram of the rotating hydraulic actuator:



r_i - Internal piston radius

r_o - External piston radius

a, b - Arm lengths of the lever

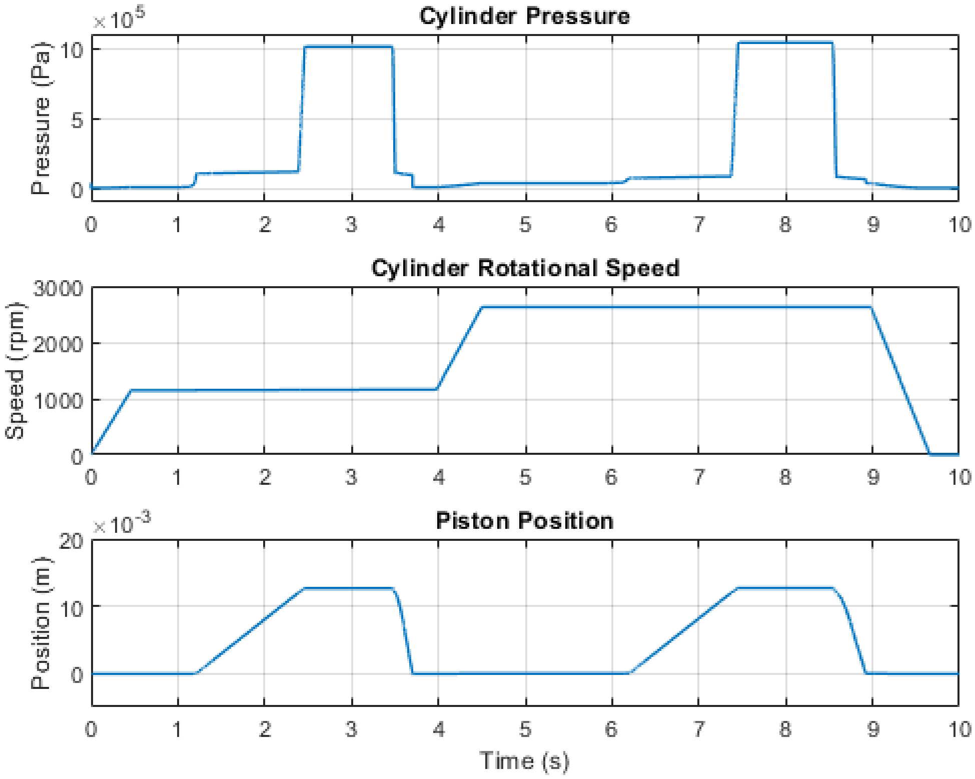
As pressure is applied to the fluid passage in the shaft, static pressure, together with pressure developed in rotating chamber, builds up until it overcomes the preload force and shifts the piston forward.

In the example, the hydraulic system is fed by a pump with delivery of 4.5 lpm. The relief valve is set to 20 bar. The motion control is performed with the 3-way directional valve, which initially connects the clutch chamber with the tank. As control signal is applied, the passage to the tank is closed and the chamber is connected with the pump.

Two actuator cycles are simulated in the example. The first stroke is performed with shaft rotating at angular velocity of 120 rad/s. The angular velocity during the second stroke is 275 rad/s.

Simulation Results from Simscape Logging

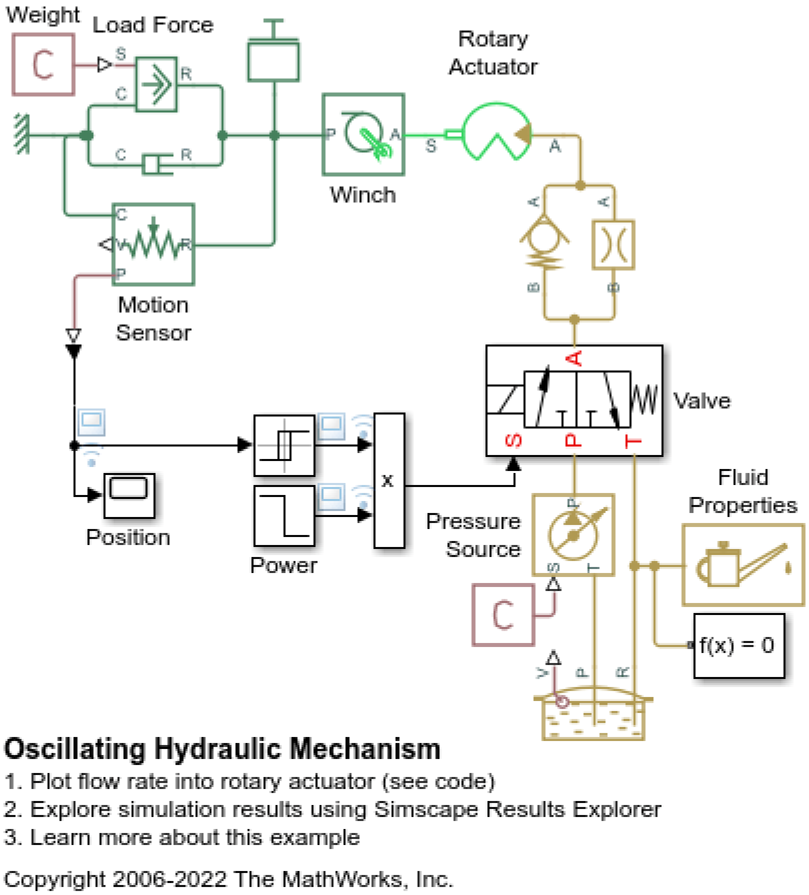
The plots below show the pressure, speed, and extension of the rotating hydraulic cylinder. A slightly higher pressure is seen in the second stroke due to the higher rotational speed of the cylinder.


Cylinder Pressure
×10^5
Pressure (Pa)
10
5
0
0
1
2
3
4
5
6
7
8
9
10
Cylinder Rotational Speed
Speed (rpm)
3000
2000
1000
0
0
1
2
3
4
5
6
7
8
9
10
Piston Position
×10^-3
Position (m)
20
10
0
0
1
2
3
4
5
6
7
8
9
10
Time (s)

Oscillating Hydraulic Mechanism

This example shows an oscillating hydraulic mechanism that consists of a single-acting hydraulic rotary actuator, winch, flow control valve, two-position electro-hydraulic valve, and power and control units. The mechanism maintains oscillating vertical motion of the weight by raising it to 25 cm and then allowing it to fall. To raise the weight, the electro-hydraulic valve is energized and connects the power unit to the hydraulic actuator inlet. As the weight reaches 25 cm the valve is deenergized, connecting the actuator chamber to tank.

Model



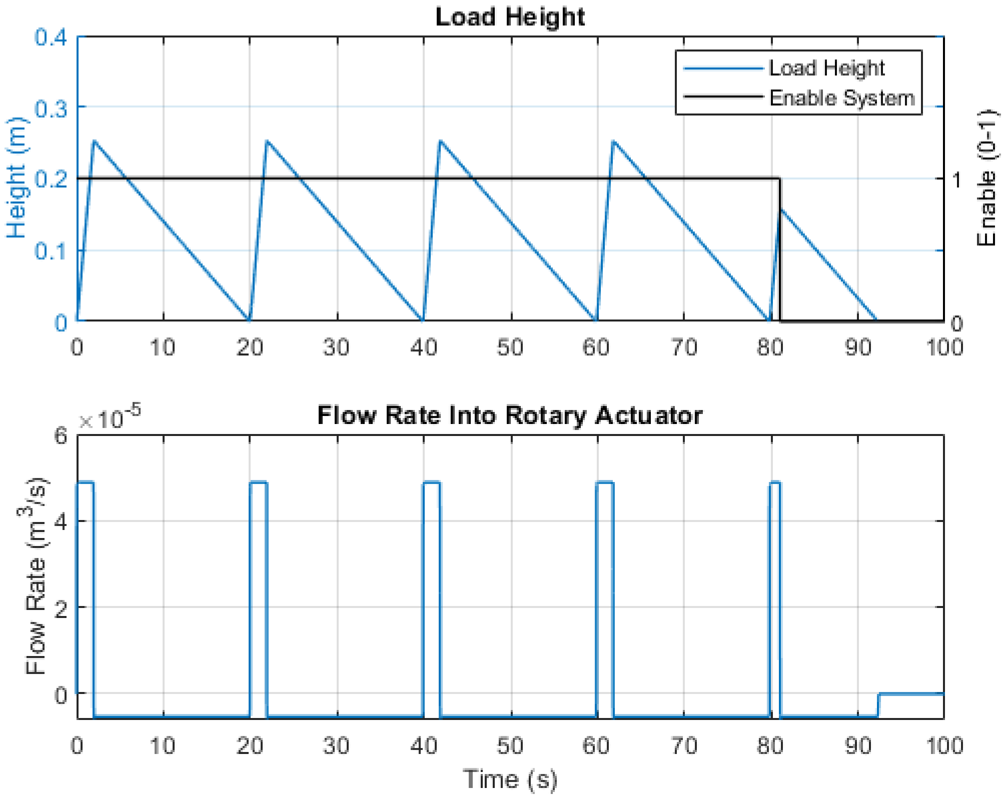
Oscillating Hydraulic Mechanism

1. Plot flow rate into rotary actuator (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2006-2022 The MathWorks, Inc.

Simulation Results from Simscape Logging

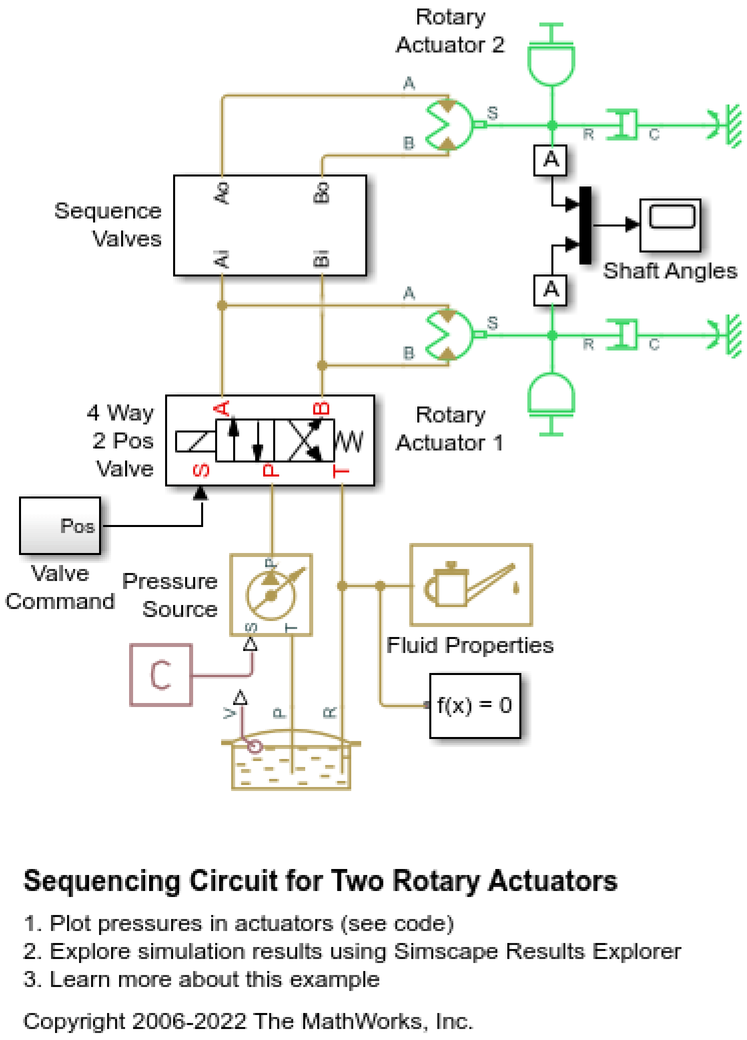
The plots below shows the behavior of the rotary actuator. The load rises much faster than it falls under the influence of gravity. Looking at the flow rate into and out of the rotary actuator, we see that fluid enters much faster than it exits, resulting in different speeds during the cycle. The last cycle is interrupted as the winch is being raised.


Load Height
Height (m)
0.4
0.3
0.2
0.1
0
Load Height
Enable System
Enable (0-1)
1
0
0
10
20
30
40
50
60
70
80
90
100
Flow Rate Into Rotary Actuator
×10^-5
Flow Rate (m^3/s)
6
4
2
0
Time (s)

Sequencing Circuit for Two Rotary Actuators

This example shows a sequence circuit that is based on four check valves installed in both pressure and return lines of the second rotary actuator. The cracking pressure of all the valves is set higher than any load pressure of Rotary Actuator 1, but lower than pressure that develops in its chambers at the end of the stroke when Rotary Actuator 1 reaches its hard stop. As a result, the Rotary Actuator 2 starts moving only after the Rotary Actuator 1 completes its stroke.

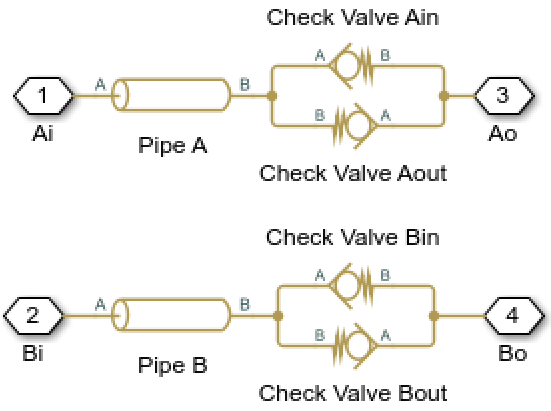
Model



Sequencing Circuit for Two Rotary Actuators

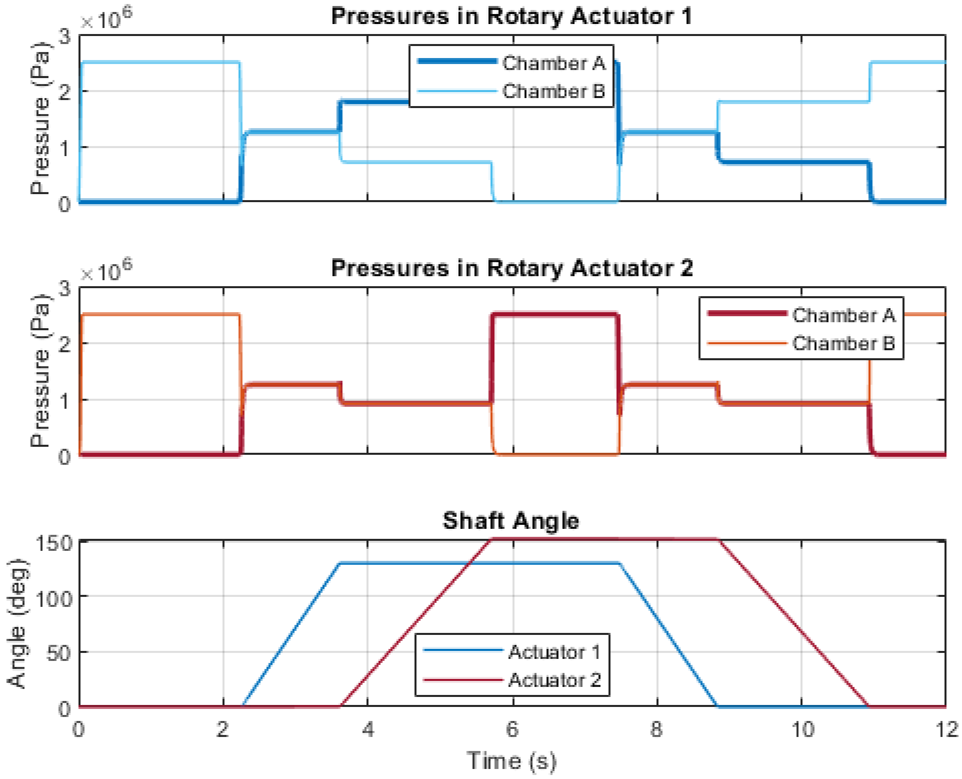
1. Plot pressures in actuators (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Sequence Valves Subsystem



Simulation Results from Simscape Logging

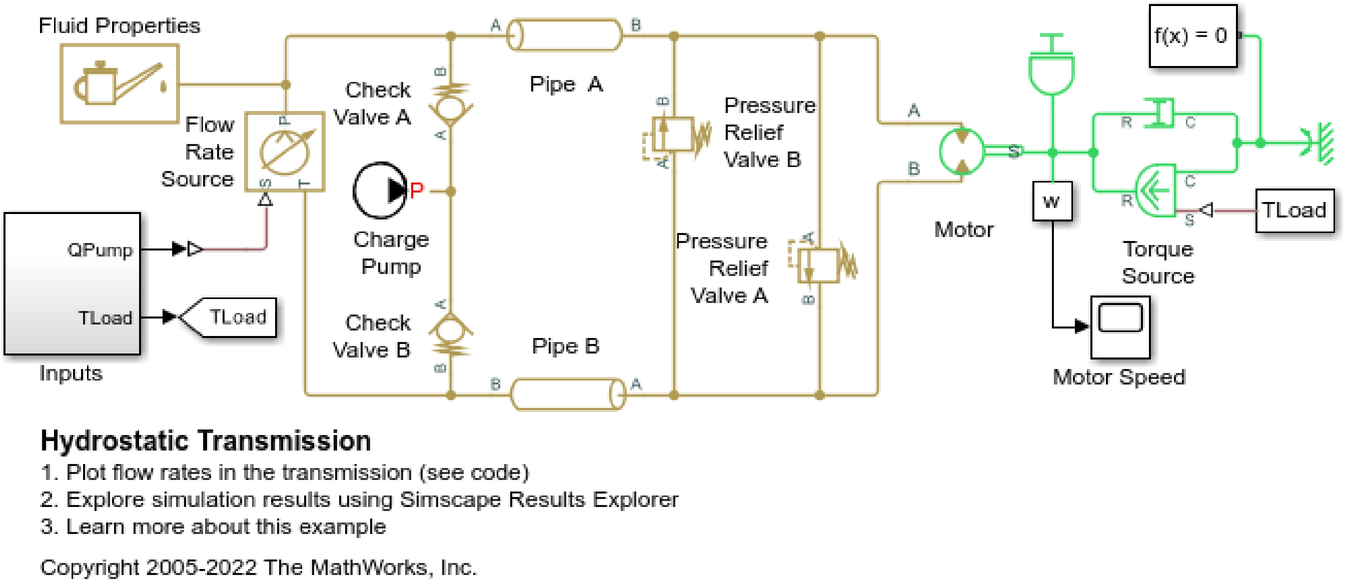
The plots below show the sequence of pressures in the rotary actuator chambers. At the start of the simulation, both actuators are driven against their hard stops at 0 degrees. At 2 seconds, the high pressure line is connected to the opposite chamber of the actuators. The sequencing valves (check valves) prevent flow to actuator 2 until actuator 1 reaches its hard stop. At that point, the pressure rises enough to overcome the check valves and actuator 2 rotates. The opposing set of check valves are similarly tuned to repeat this pattern in the opposite direction.


Pressures in Rotary Actuator 1
×10^6
Pressure (Pa)
3
2
1
0
Chamber A
Chamber B
Pressures in Rotary Actuator 2
×10^6
Pressure (Pa)
3
2
1
0
Chamber A
Chamber B
Shaft Angle
Angle (deg)
150
100
50
0
Actuator 1
Actuator 2
0
2
4
6
8
10
12
Time (s)

Hydrostatic Transmission

This example shows a transmission built of a variable-displacement pump, and a fixed-displacement hydraulic motor. Pipelines between the source and motor are connected by two replenishing valves (check valves) and a charge pump on the pump side and two pressure-relief valves on the motor side. The motor drives a mechanical load consisting of inertia, viscous friction, and time-varying torque. The system is tested with varying pump flow rate and motor torque load.

Model



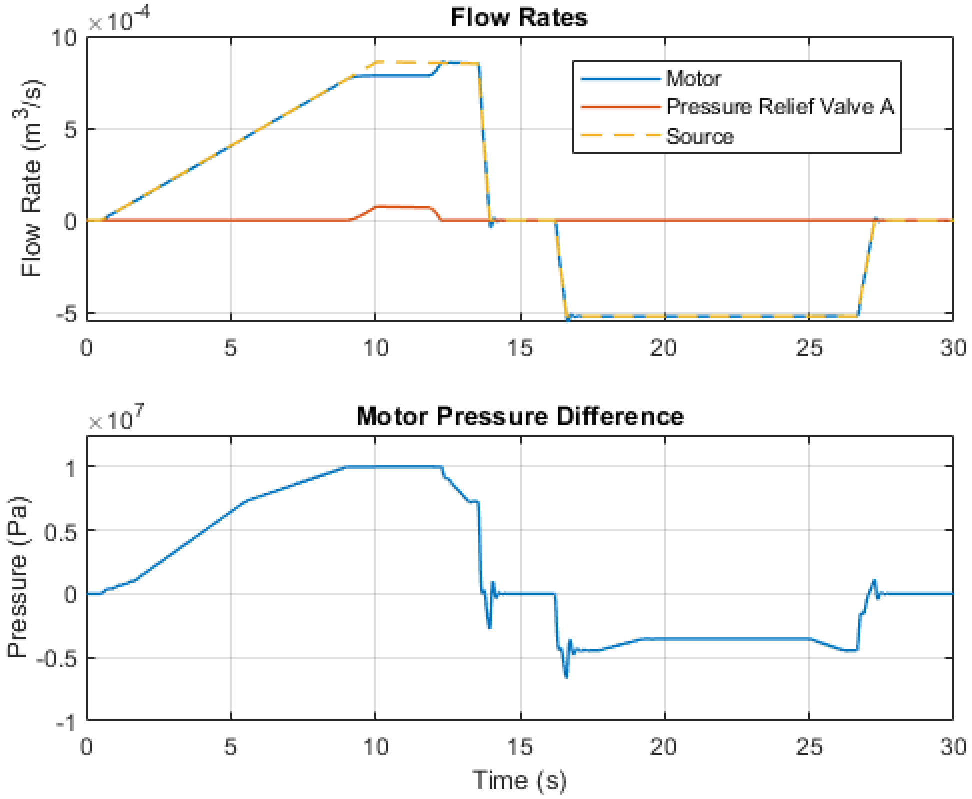
Hydrostatic Transmission

1. Plot flow rates in the transmission (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2005-2022 The MathWorks, Inc.

Simulation Results from Simscape Logging

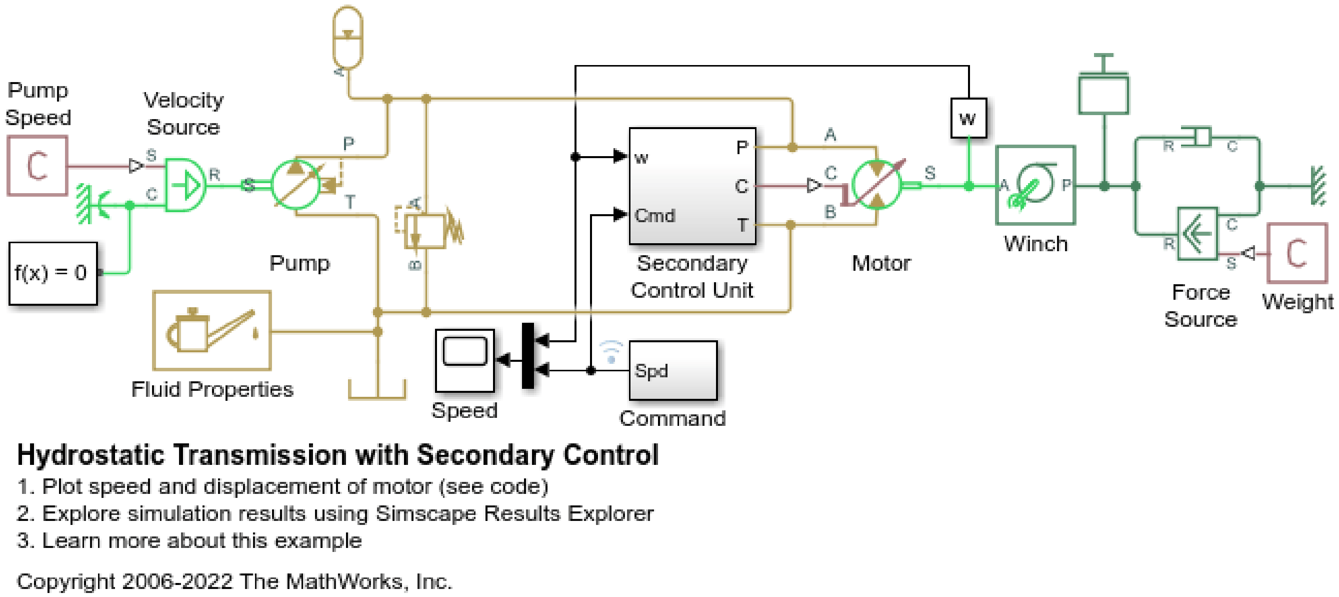
The plots below show the flow rate through the hydrostatic transmission and the pressure drop across the motor. As the motor reaches its nominal pressure, flow is diverted through a pressure relief valve.


Flow Rates
×10^-4
10
5
0
-5
Flow Rate (m^3/s)
Motor
Pressure Relief Valve A
Source
0
5
10
15
20
25
30
Motor Pressure Difference
×10^7
1
0.5
0
-0.5
-1
Pressure (Pa)
0
5
10
15
20
25
30
Time (s)

Hydrostatic Transmission with Secondary Control

This example shows the usage of secondary control in hydrostatic transmissions with a variable-displacement motor. It uses a pressure-compensated variable-displacement pump, a variable-displacement motor, and is controlled by a secondary control unit. The secondary control unit is a servo-cylinder controlled by a proportional 4-way valve. The servo-cylinder drives the control member of the variable-displacement motor, which is represented by a mass, spring, and damper. The hydrostatic transmission is used in a closed-loop control system with an angular velocity feedback. It drives a winch which lifts a mass and acts against a viscous damper. A dynamic compensator is included in the secondary control unit to improve system stability.

Model

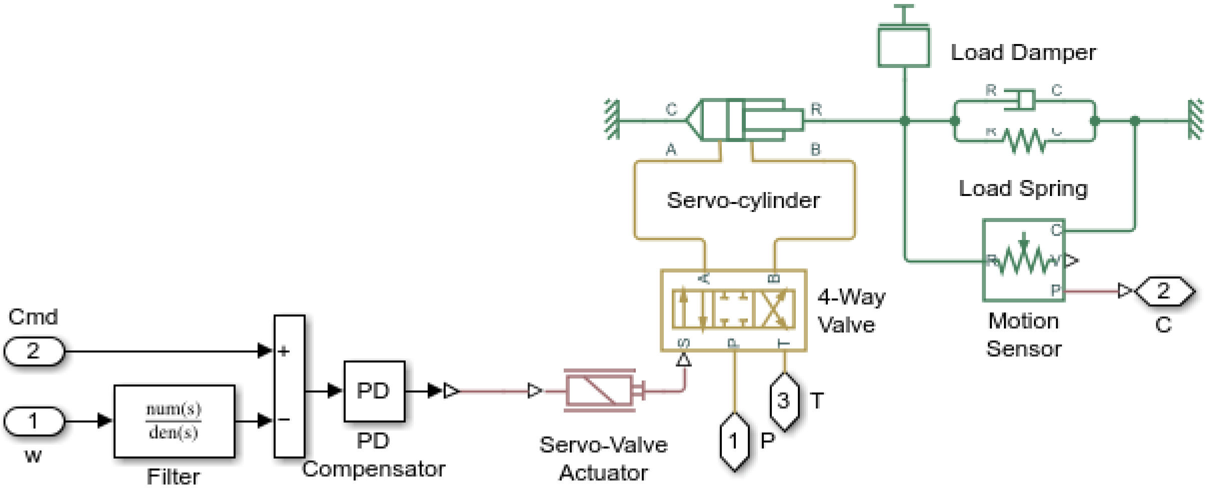


Hydrostatic Transmission with Secondary Control

1. Plot speed and displacement of motor (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

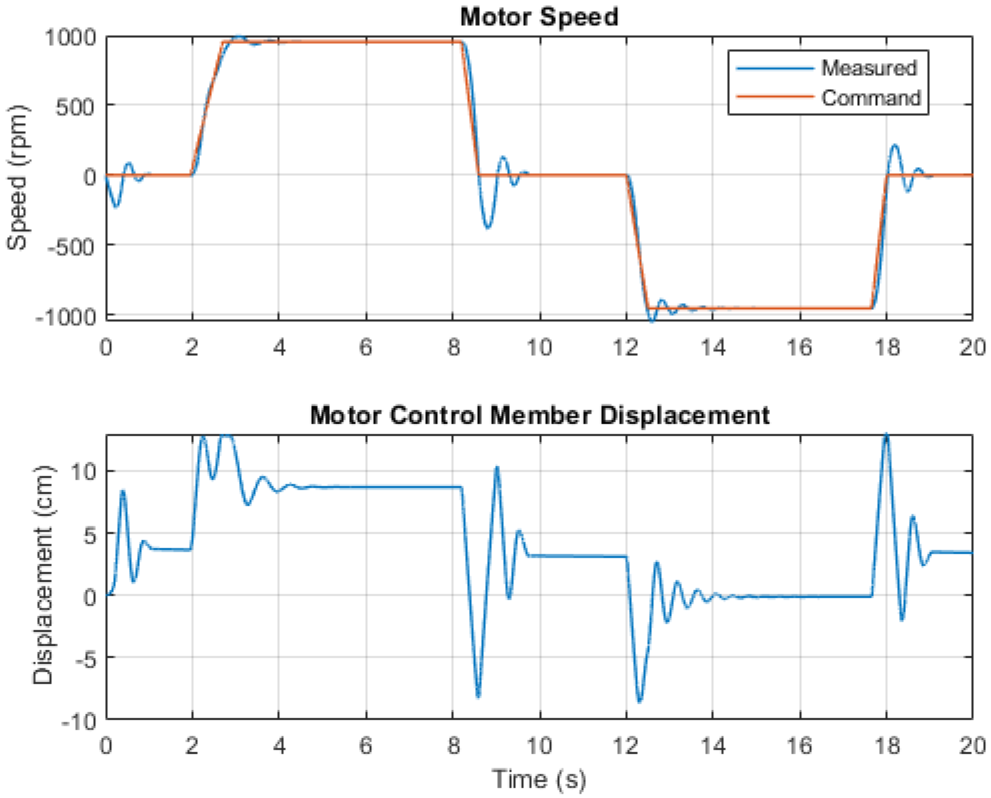
Copyright 2006-2022 The MathWorks, Inc.

Secondary Control Unit Subsystem



Simulation Results from Simscape Logging

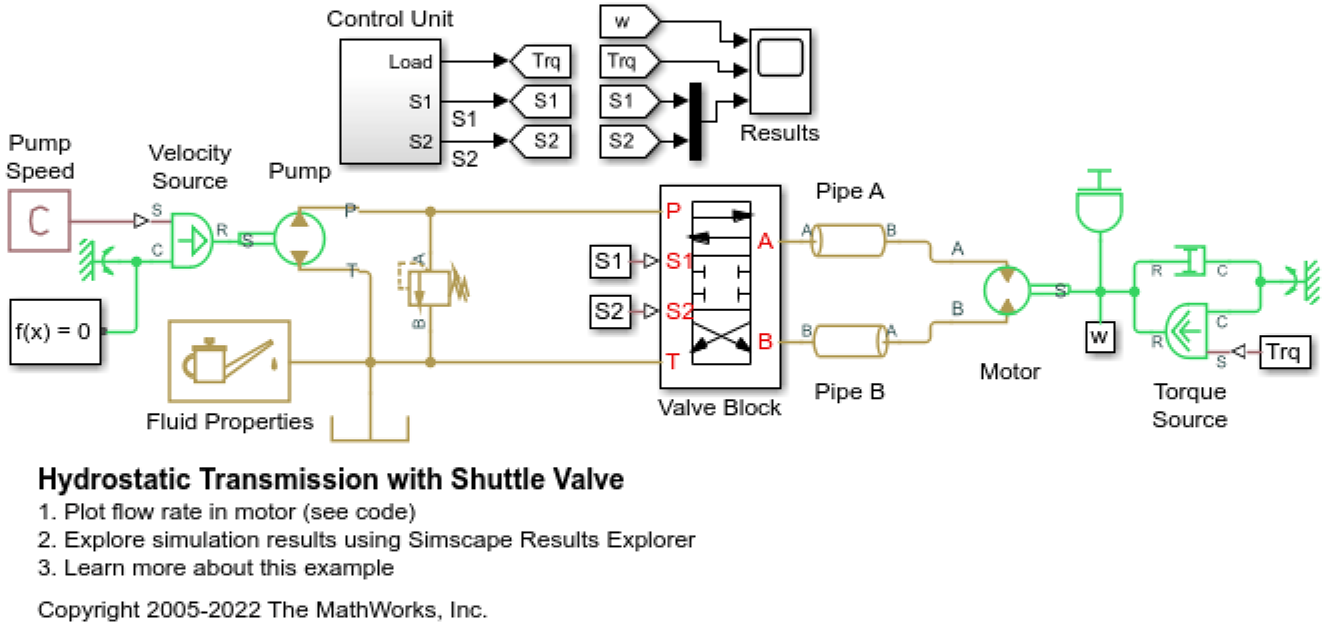
The plots below show the commanded and measured speed of the motor as well as the displacement of the control member which sets the displacement of the motor.



Hydrostatic Transmission with Shuttle Valve

This example shows a hydrostatic transmission with a shuttle valve in the control unit. The 4-way directional valve that controls the motor is actuated by a valve actuator whose input is controlled by two 2-position valves (Shuttle A Valve, Shuttle A1 Valve). The shuttle valve installed between the actuator and the 2-position valves allows the 4-way directional valve to be opened by opening either 2-position valve, and those valves are opened by signals S1 and S2. A fixed orifice is connected to the P port of the two-position valves to decrease flow consumption of the control circuit. Another fixed orifice introduces a small leak from the valve control chamber to avoid this chamber being isolated from the main flow. The motor is connected to pressure line at 1 second and 5 seconds by activating each of the two-position valves.

Model

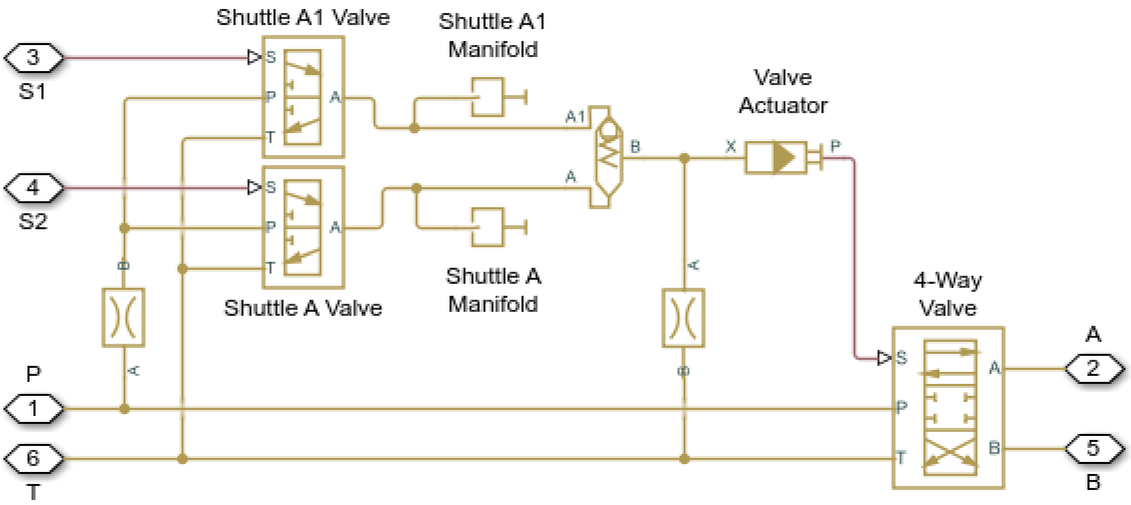


Hydrostatic Transmission with Shuttle Valve

1. Plot flow rate in motor (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

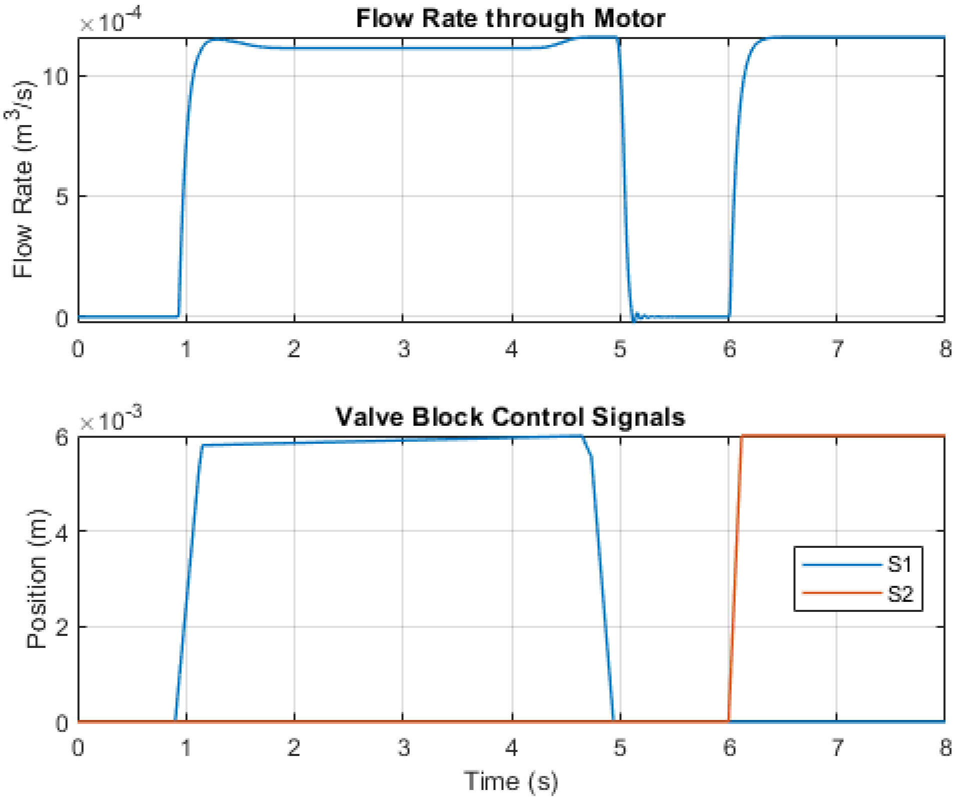
Copyright 2005-2022 The MathWorks, Inc.

Valve Block Subsystem



Simulation Results from Simscape Logging

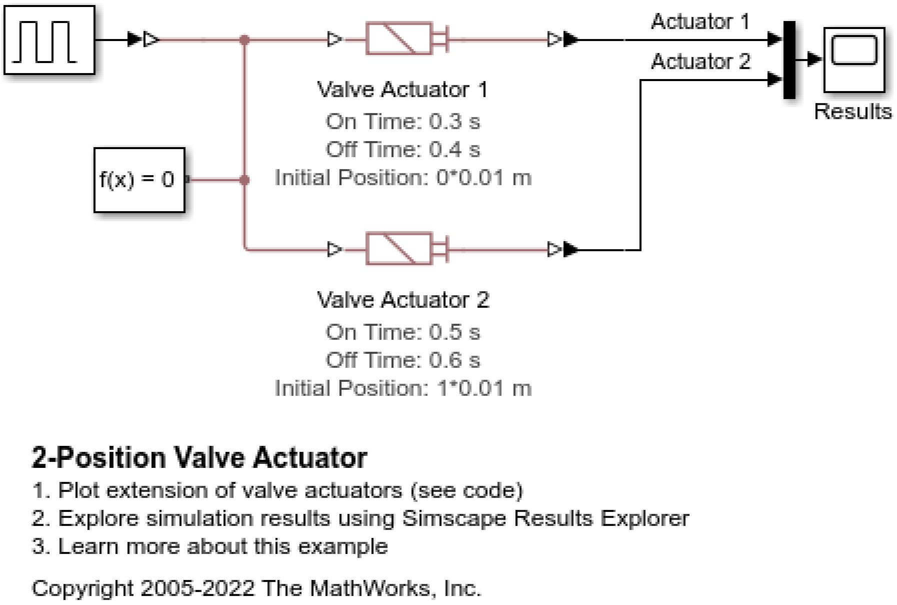
The plots below show the flow rate through the motor and the control signals to valves that connect the motor to the high pressure side of the pump.



2-Position Valve Actuator

This example shows two valve actuators with different values for switching-on time and switching-off time. Valve Actuator 1 is set to start from the retracted position, while Valve Actuator 2 from the extended position. Both actuators are driven with the same pulse signal.

Model



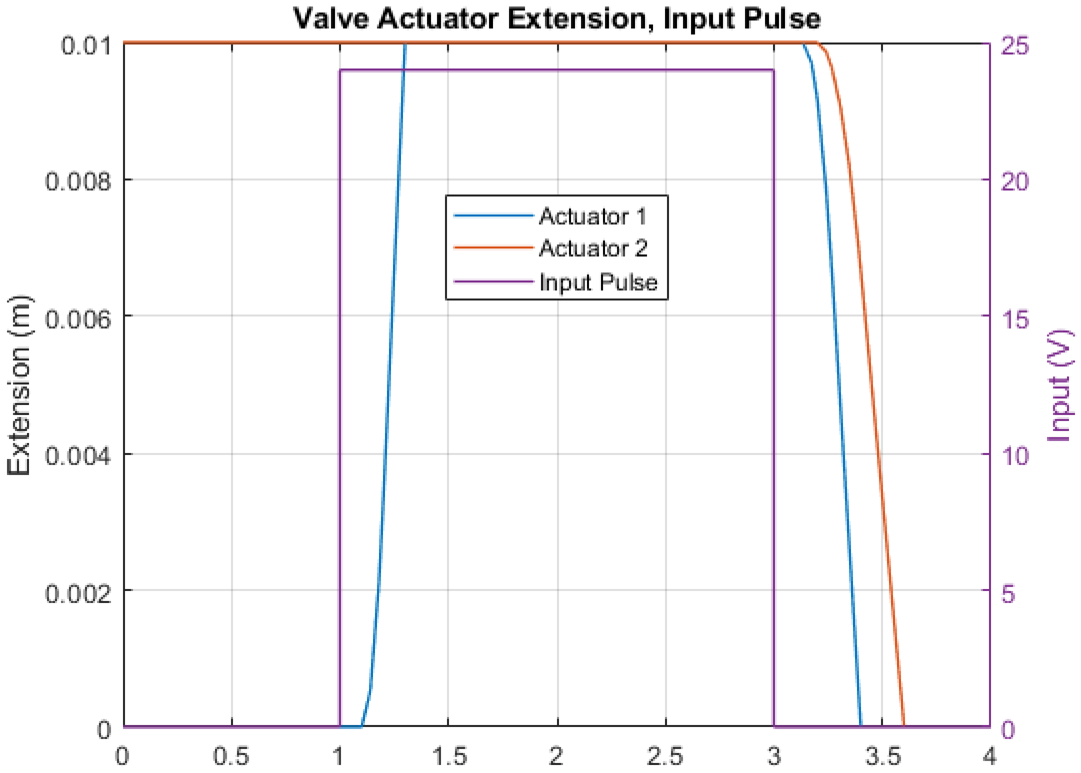
2-Position Valve Actuator

1. Plot extension of valve actuators (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2005-2022 The MathWorks, Inc.

Simulation Results from Simscape Logging

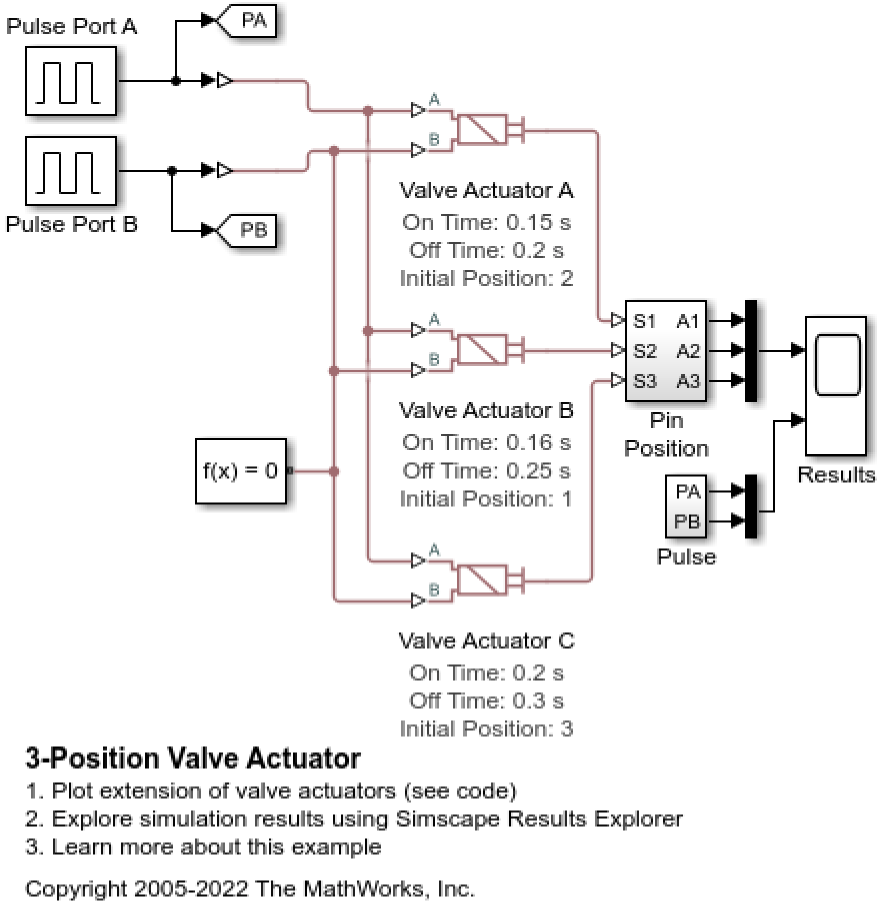
The plot below shows the behavior of two valve actuators with different values for switching-on time and switching-off time.


Valve Actuator Extension, Input Pulse
Extension (m)
0.01
0.008
0.006
0.004
0.002
0
0
0.5
1
1.5
2
2.5
3
3.5
4
Input (V)
25
20
15
10
5
0
Actuator 1
Actuator 2
Input Pulse

3-Position Valve Actuator

This example shows a use of a double-acting valve actuator. All three actuators are driven by the same pulse signals. A pulse is first applied to port A and after 1.5 s delay a pulse is applied to port B. Actuator A is set to start from the "Extended positive" position, while actuator C starts from the "Extended negative" position. As a result, actuators A and C move to the neutral position at the start of simulation and reach this position before the first pulse is applied. The actuators strokes, switch-on, and switch-off times are set to different values to illustrate the effects of these parameters.

Model



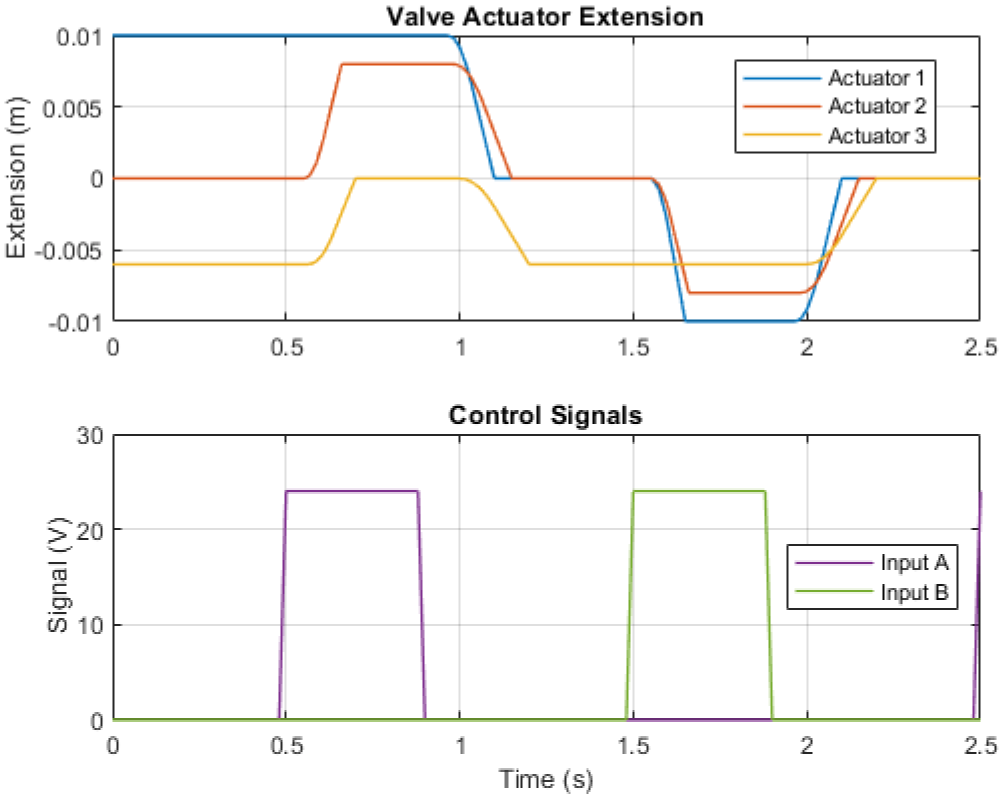
3-Position Valve Actuator

1. Plot extension of valve actuators (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2005-2022 The MathWorks, Inc.

Simulation Results from Simscape Logging

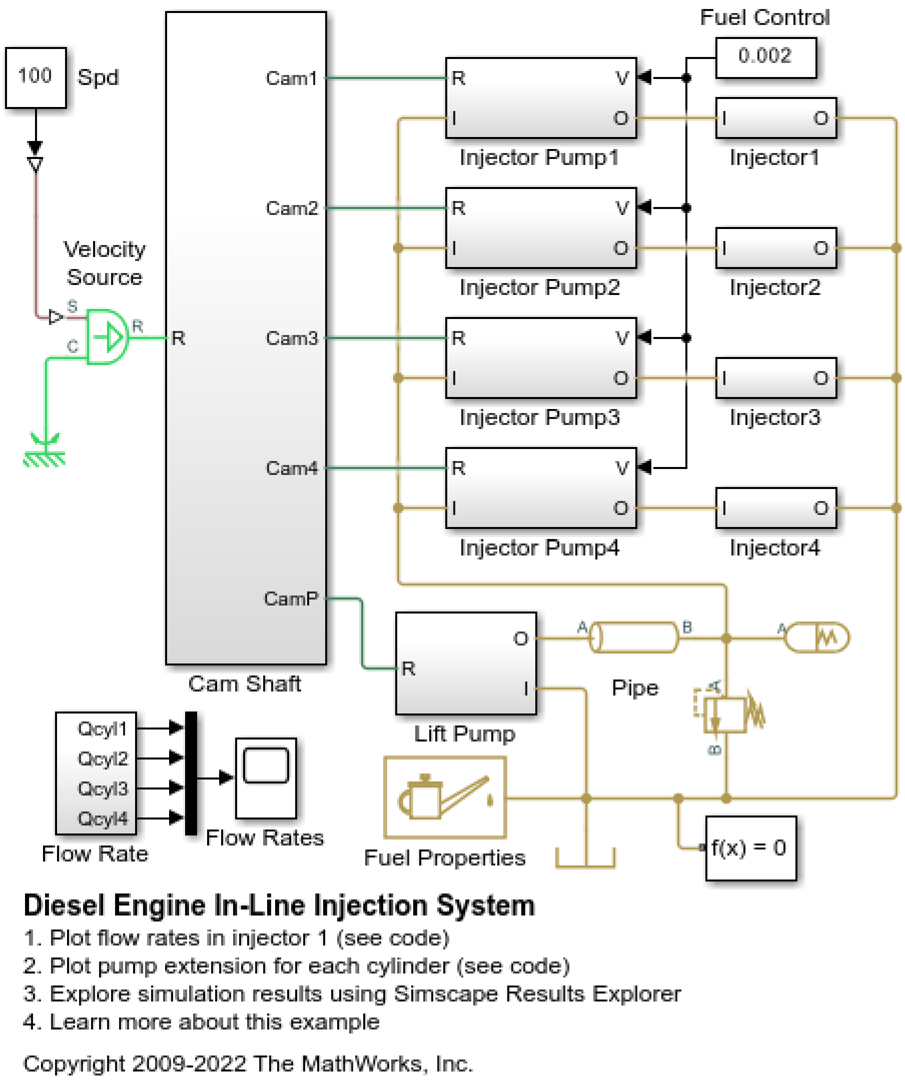
The plots below show the behavior of the 3-Position Valve Actuator block. The same pulses are applied to all three actuators, which each have different values for switching-on time, switching-off time, and stroke.


Valve Actuator Extension
Extension (m)
0.01
0.005
0
-0.005
-0.01
0
0.5
1
1.5
2
2.5
Actuator 1
Actuator 2
Actuator 3
Control Signals
Signal (V)
30
20
10
0
0
0.5
1
1.5
2
2.5
Time (s)
Input A
Input B

Diesel Engine In-Line Injection System

This example shows an in-line multi-element diesel injection system. It contains a cam shaft, lift pump, 4 in-line injector pumps, and 4 injectors.

Model



Diesel Engine In-Line Injection System

1. Plot flow rates in injector 1 (see code)
2. Plot pump extension for each cylinder (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2009-2022 The MathWorks, Inc.

Injection System Description

The diesel injection system simulated by this model is shown in the schematic diagram below.

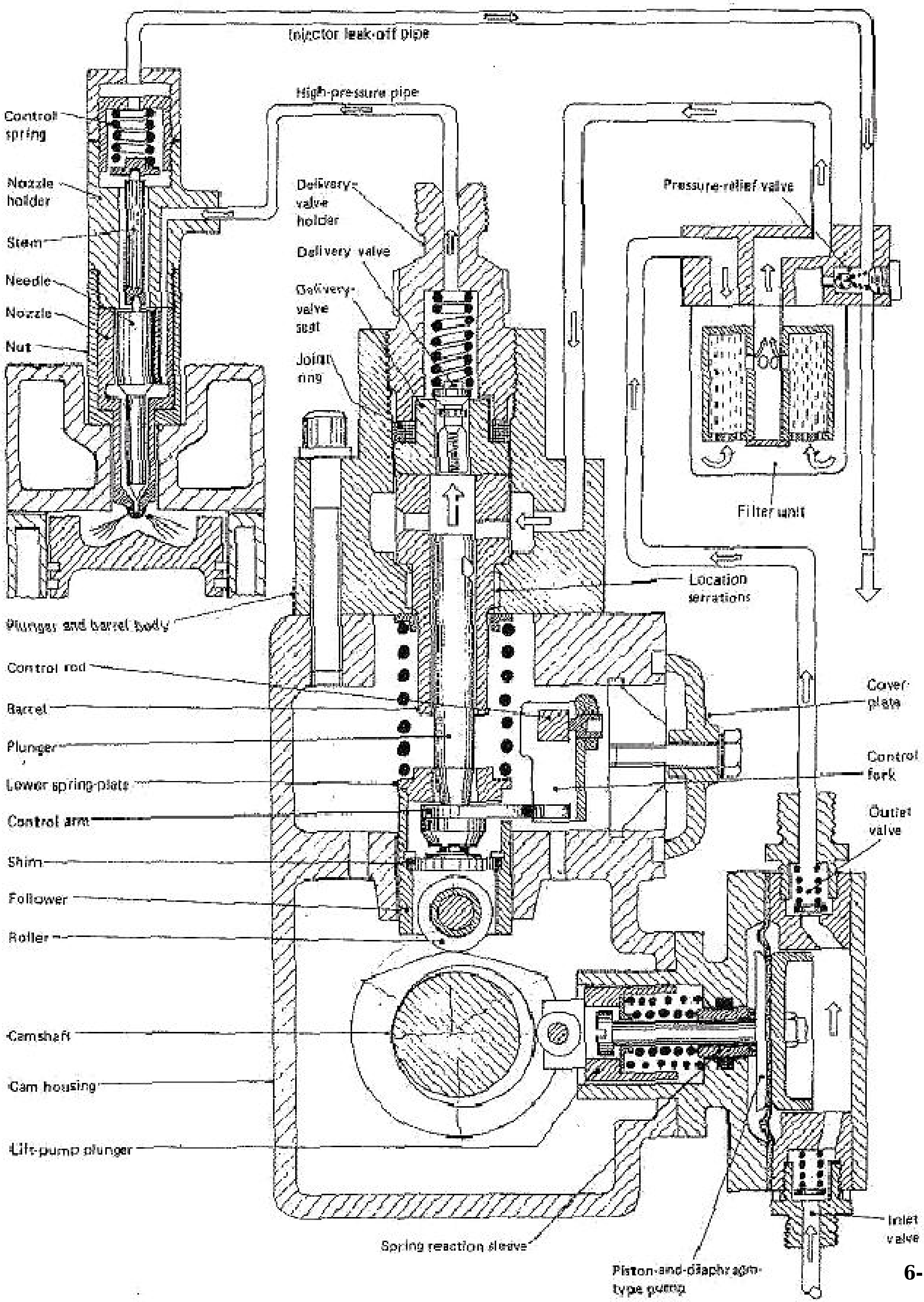

Injector leak-off pipe
High-pressure pipe
Controll spring
Nozzle holder
Stem
Needle
Nozzle
Nut
Delivery-valve holder
Delivery valve
Delivery-valve seat
Joint ring
Pressure-relief valve
Filter unit
Location serrations
Plunger and barrel body
Control rod
Barrel
Plunger
Lower spring-plate
Control arm
Shim
Follower
Roller
Camshaft
Cam housing
Lift-pump plunger
Cover-plate
Control fork
Outlet valve
Inlet valve
Spring reaction sleeve
Piston-and-diaphragm-type pump

Figure 1. Injection System Schematic Diagram

The system structure is reproduced from H. Heisler, *Vehicle and Engine Technology* (second edition), 1999, and is categorized as an in-line multi-element injection system. It consists of the following basic units:

- Cam shaft
- Lift pump
- In-line injector pump with four pumping elements, one element per cylinder
- Injectors

The cam shaft carries five cams. The first one is the eccentric cam to actuate the lift pump. The remaining four are intended to drive plungers of the pump. The cams are installed in such a way that pumping elements deliver fuel in the firing order and at the correct instant in the engine's cycle of operation. The lift pump supplies fluid to the intake of injector pump elements. Each element of the pump consists of a cam-driven plunger, delivery valve, and the governor assembly. The purpose of the governor is to control the volume of the fuel delivered by the plunger to a cylinder. It is attained by rotating the plunger with the helical groove with respect to the spill orifice. All the system units will be described in more details in the following sections.

The objective of the simulation is to investigate the entire system operation. The objective dictates the extent of idealization of every model in the system. If the objective were, for example, delivery valve or the injector investigation, the amount of factors taken into consideration and the scope of the element considered would be different.

Note: The model of the system does not represent any particular injection system. All the parameters have been assigned based on practical considerations, and do not represent any particular manufacturer parameters.

Cam Shaft

The model of a cam shaft is built of five cam models. There are four parabolic profile cams and one eccentric cam. Every cam contains a Simulink® masked subsystem that describes the cam's profile and generates motion profile for the position source, which is built out of Simscape™ blocks.

Cam Profile Simulation

The motion profile is generated as a function of the shaft angle, which is measured with the Angle Sensor block from the Pumps and Motors library. The sensor converts the measured angle to a value in the range from zero to 2*pi. After the cycle angle is determined, it is passed to the Simulink IF subsystem, which computes the profile. The cam that drives the plunger of the pump element is supposed to have a parabolic profile under which the follower moves back and forth at constant acceleration, as follows:

$$x = 2s\cdot\left(\frac{\theta-\theta_E}{\alpha}\right)^2, \qquad if\;\; \theta_E \le \theta < \frac{\alpha}{2}+\theta_E$$

$$x = s(1-2\left(1-\frac{\theta-\theta_E}{\alpha}\right)^2, \qquad if\;\; \frac{\alpha}{2}+\theta_E \le \theta < \alpha+\theta_E$$

$$x = s(1-2\left(\frac{\theta-\theta_R}{\beta}\right)^2, \qquad if\;\; \theta_R \le \theta < \frac{\beta}{2}+\theta_R$$

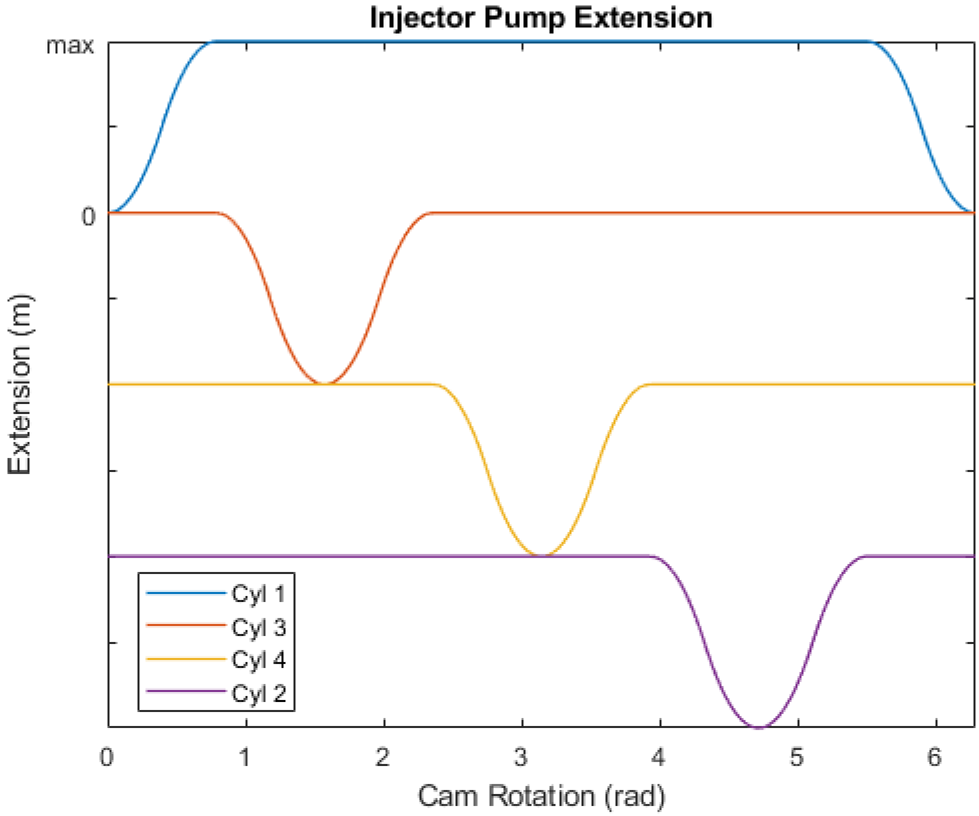
$$x = 2s\left(1-\frac{\theta-\theta_R}{\beta}\right)^2, \qquad if\;\; \frac{\beta}{2}+\theta_R \le \theta < \beta+\theta_R$$

where

x	Follower position
s	Follower stroke
θ	Cam angular position (with phase angle accounted)
θ_E	Start extend angle
α	Extend angle
θ_R	Start retract angle
β	Retract angle

As a result, at *start extend angle* the follower starts moving up and reaches its top position after the shaft turns an additional *extend angle*. The follower starts the return stroke at *start retract angle* and it takes the *retract angle* to complete this motion. The difference between the *start retract angle* and the (*start extend angle* + *extend angle*) sets the *dwell angle* at fully extended position. The profile is implemented in the Simulink IF subsystem.

The firing sequence for the simulated diesel engine is assumed to be 1-3-4-2. The sequence of cam operation is shown in the figure below. The extend and return angles are set to pi/4. The dwell angle with fully extended follower is set to 3*pi/2 rad.



The profile of the eccentric cam is computed with the formula

$$x = e\left(1 - \cos(\theta)\right)$$

where *e* is eccentricity.

Position Source

The position source model, which generates position in mechanical translational motion following the Simulink signal at its input, is built of an Ideal Translational Velocity Source block, a PS Gain block, and a Translational Motion Sensor block installed in the negative feedback. The transfer function of the position source is

$$H(s) = \frac{1}{Ts+1},$$

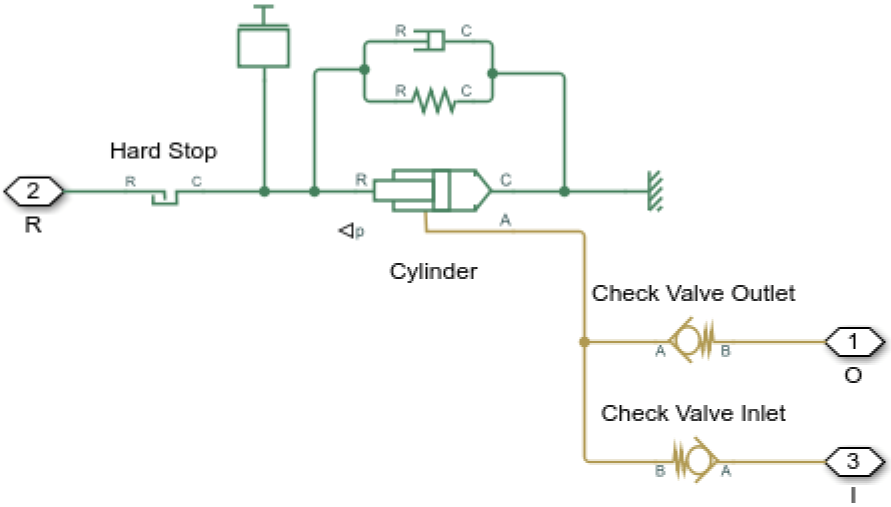
where

T - Time constant, equal 1/Gain,

Gain - Gain of the PS Gain block.

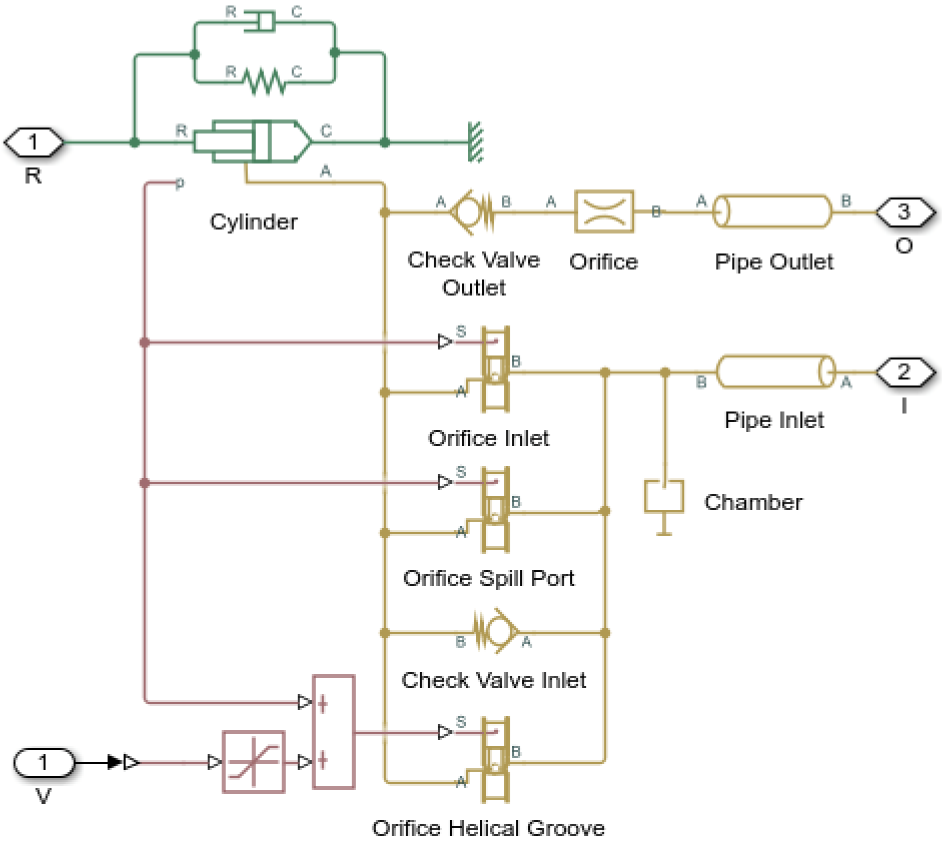
The gain is set to 1e6, which means that signals with frequencies up to 160 kHz are passed practically unaffected.

Lift Pump



The model of the lift pump, which is a piston-and-diaphragm type pump, is built of a Single-Acting Hydraulic Cylinder block and two Check Valve blocks. The check valves simulate inlet and outlet valves installed on both sides of the lift pump (see Figure 1). The contact between the pump rod roller and the cam is represented with the Translational Hard Stop block. The Translational Spring block simulates two springs in the pump that are supposed to maintain permanent contact between the roller and the cam.

Injection Pump



The in-line injection pump is a four-element pumping unit. Each element supplies fuel to its cylinder. All four elements are identical by design and parameters and simulated with the same model called Injection pump element. Each Injection pump element Injection pump element model contains two subsystems, named Pump and Injector, respectively. The Pump represents the pump plunger and the pump control mechanism, while the Injector simulates an injector installed directly on the engine cylinder (see Figure 1).

The pump plunger oscillates inside pump barrel driven by the cam (see Figure 1). The plunger is simulated with the Single-Acting Hydraulic Cylinder block. The Translational Hard Stop and Mass blocks represent the contact between the plunger roller and mass of the plunger, respectively. The contact is maintained by spring TS.

When the plunger moves down, the plunger chamber is filled with fuel under pressure developed by the lift pump. The fluid fills the chamber through two orifices, named Inlet port and Spill port (see Figure 2,a below).

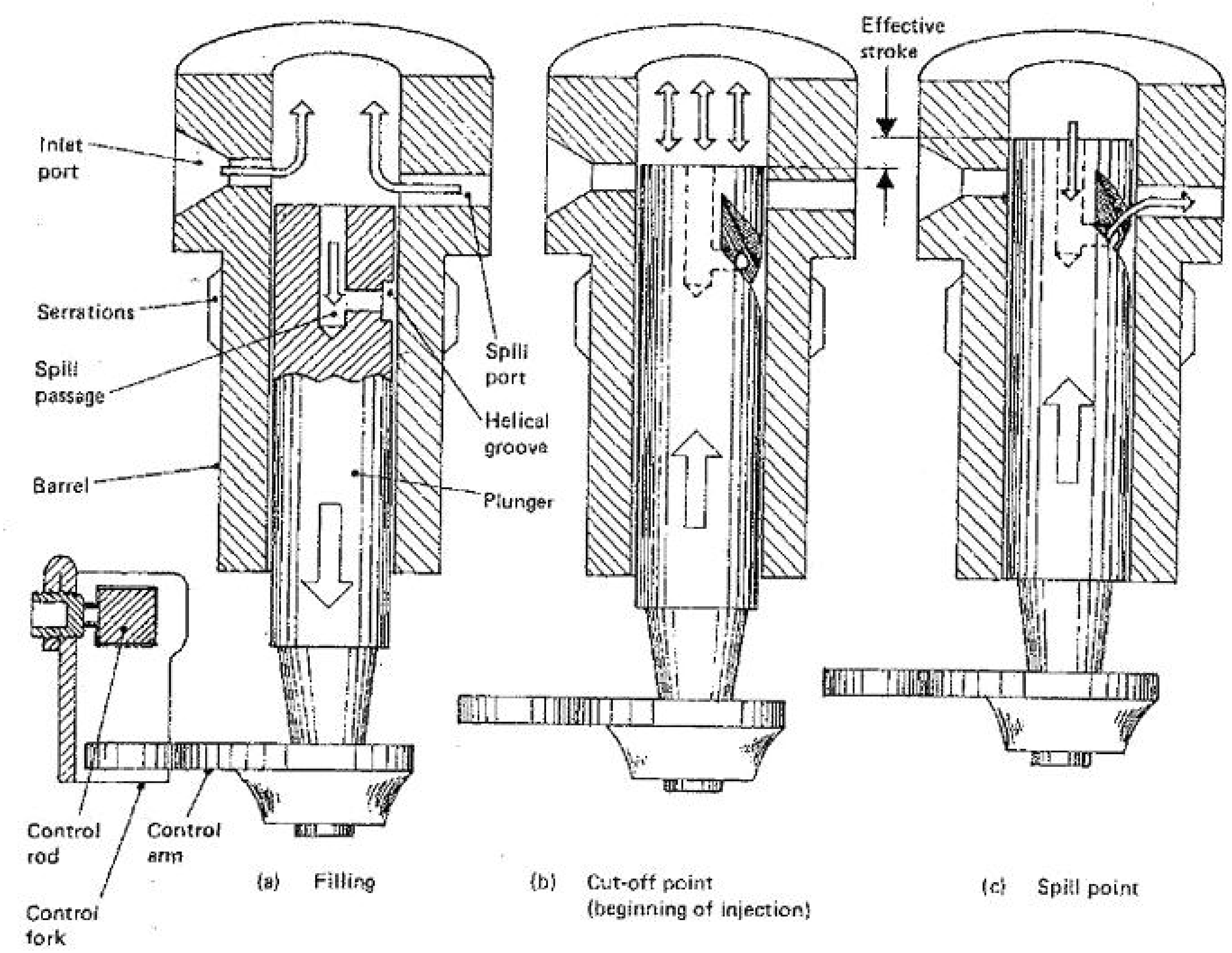


Figure 2. Plunger Interaction with Control Orifices in the Barrel

After the plunger moves towards its top position high enough to cut off both orifices from the inlet chamber, the pressure at the outlet starts building up. At a certain rise the injector at the engine cylinder is forced to open and fuel starts injecting in the cylinder (Figure 2,b).

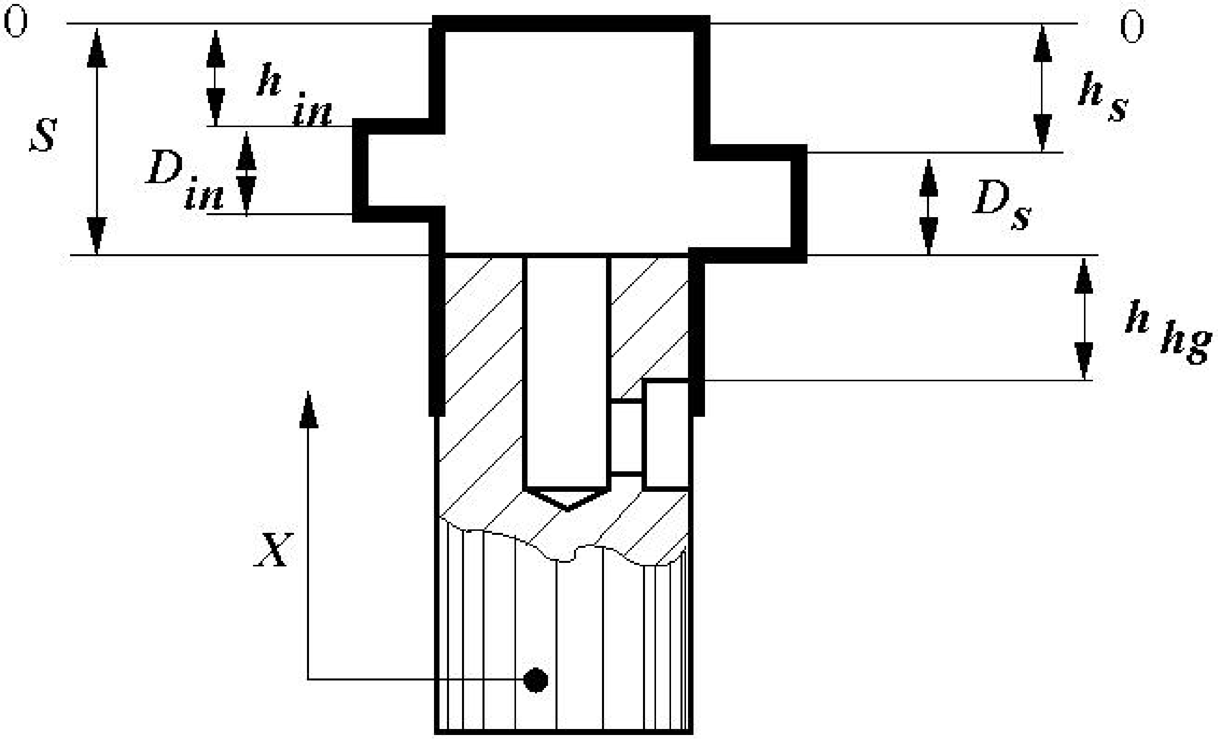
The injection stops as the helical groove formed on the side surface of the plunger reaches the Spill port, which connects the top chamber with the low pressure chamber through the orifice drilled inside the plunger (Figure 2,c). You can control the position of the helical groove with respect to the Spill port by rotating the plunger with the control fork, thus regulating the volume of fuel injected in the cylinder.

The model of the plunger control mechanism is based on the following assumptions:

1. There are three variable orifices in the control circuit: inlet port, spill port, and orifice formed by the helical groove and the spill port. The openings of the inlet and the spill orifices depend on the plunger motion, while the opening of the groove-spill port orifice is a function of the plunger motion and the plunger rotation. For the sake of simplicity, the displacement generated by the plunger

rotation is represented as a source of linear motion which is combined with the plunger displacement.

2. The figure below shows all the dimensions necessary to parameterize the orifices:



D_{in} - Inlet port orifice diameter

D_s - Spill port orifice diameter

S - Plunger stroke

h_{in} - Distance between the inlet orifice and the top plunger position

h_s - Distance between the spill port orifice and the top plunger position

h_{hg} - Distance between the spill port orifice and the upper edge of the helical groove

3. In assigning initial openings and orifice orientations, the plunger *top position is taken as the origin* and the motion in the upward direction is considered as the motion in positive direction. In other words, axis *X* is directed upward. Under these assumptions, the inlet and spill port orifice directions must be set to **Opens in negative direction**, while the groove-spill port orifice must be set to **Opens in positive direction** since it opens when the plunger moves upward. The table below shows the values assigned to initial openings and orifice diameters.

Notation	Name in parameter file	Value	Remarks
S	stroke	0.01 m	

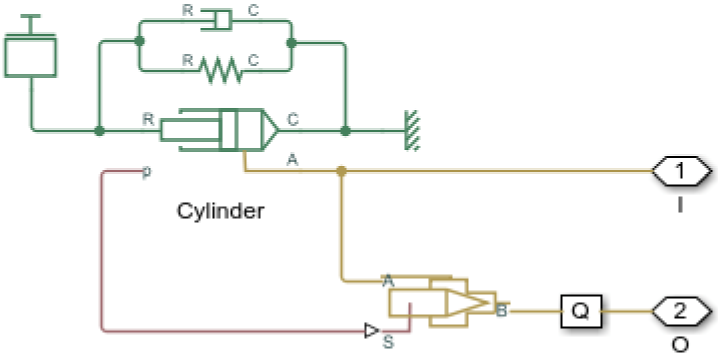
```
D_in      inlet_or_diameter                   0.003 m
D_s       spill_or_diameter                   0.0024 m
h_in      -stroke + inlet_or_diameter + 0.001           The inlet orifice is shifted upward b
h_s       -stroke + spill_or_diameter
h_hg      spill_or_diameter                             The spill orifice is assumed to be fu
```

4. The plunger effective stroke equals to

$$estroke = stroke - D_{in} - 1mm$$

The inlet orifice is generally located higher than the spill orifice. In the example, this distance is set to 1 mm. By rotating the plunger, you change the initial opening of the groove-spill port orifice. Since initial opening is a parameter and cannot be dynamically changed, the shift of the initial opening is simulated by addition of an equivalent linear displacement of the orifice control member. The larger the equivalent signal, the sooner the spill orifice is opened, thus decreasing the volume of fuel delivered to the cylinder. The maximum value of the equivalent signal is equal to the effective stroke. At this value, the spill orifice remains open all the time.

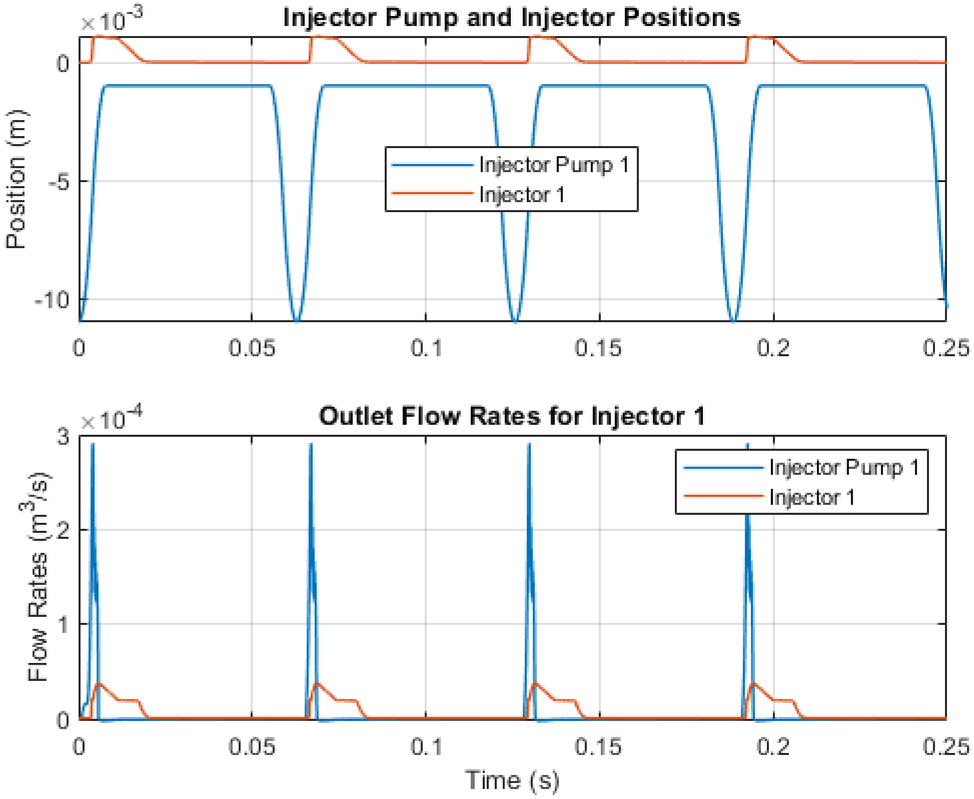
Injector



The model of the injector is based on the Single-Acting Hydraulic Cylinder block and the Needle Valve block. The needle valve is closed at initial position by the force developed by the preloaded spring. As the force developed by the cylinder overcomes the spring force, the injector opens and allows fuel to be injected in the cylinder. In the example, the injector is set to be opened at 1000 bar.

Simulation Results from Simscape Logging

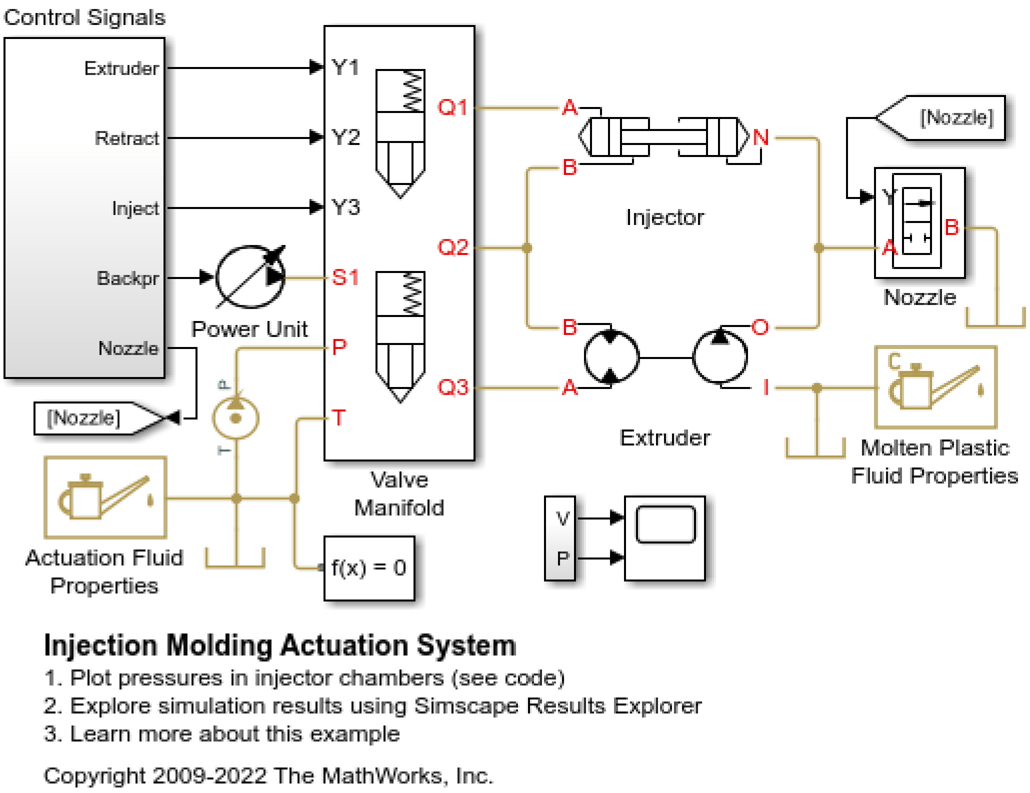
The plots below show the positions and outlet flow rates of injector pump 1 and injector 1. The effect of the cam profile is shown in the displacement of injector pump 1. During the second half of the cam stroke, fuel exits the injector pump and passes into the injector. The fuel exits the injector via a needle valve. The injector has chamber with a preloaded spring which acts to store the fluid from the pump temporarily and push it out of the injector more smoothly.


Injector Pump and Injector Positions
×10^-3
Position (m)
0
-5
-10
0
0.05
0.1
0.15
0.2
0.25
Injector Pump 1
Injector 1
Outlet Flow Rates for Injector 1
×10^-4
Flow Rates (m^3/s)
3
2
1
0
0
0.05
0.1
0.15
0.2
0.25
Time (s)
Injector Pump 1
Injector 1

Injection Molding Actuation System

This example shows an injection molding actuation system. The model contains a set of cartridge valves that control pumps, motors, and cylinders to execute the steps of an injection molding process.

Model



Injection Molding Actuation System

1. Plot pressures in injector chambers (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2009-2022 The MathWorks, Inc.

System Schematic

The system presented in the example is the actuation system of the molding machine injection unit described in full detail in the *Vickers Industrial Hydraulics Manual* (second edition, 1989), and shown in the schematic diagram below.

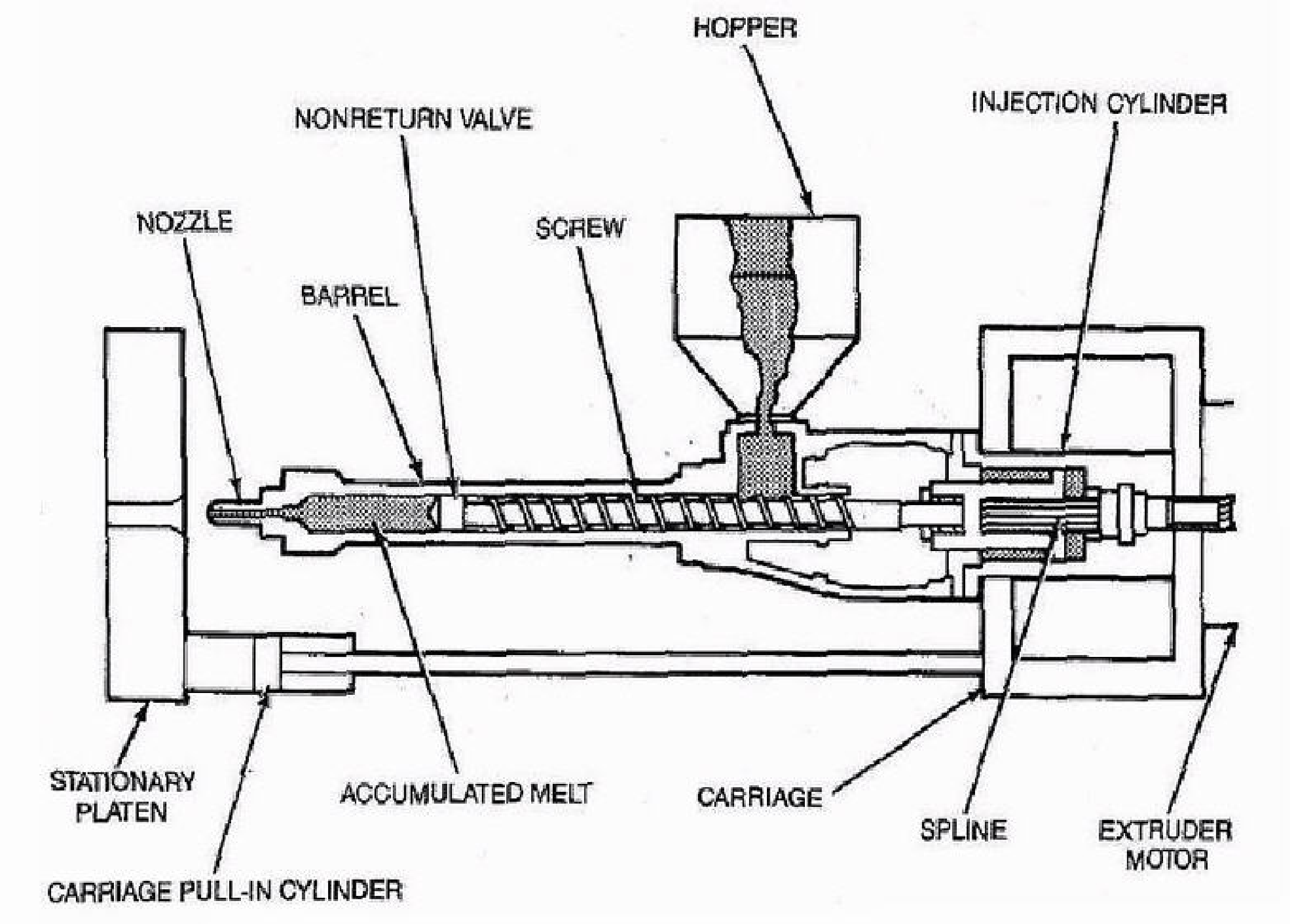


Figure 1. Working Members of the Molding Machine Injection Unit

The unit consists of a screw and barrel, hopper, extruder motor, and injection cylinder, which are all mounted on a carriage. The screw, extruder motor, and injection cylinder are arranged in a single unit that can slide along the carriage to eject the molten plastic from the barrel cylinder into the mold. The cycle considered in the example starts with the extruder rotation, during which molten plastic is pushed forward by the screw through the nonreturn (check) valve. The rotation is stopped when preset volume of molten plastic has been accumulated in the front chamber of the barrel. The injection maintains specified back-pressure while the chamber is filled and is pushed back by the molten plastic.

The next operation following the extruder run is called decompression. During this phase, the active chamber of the injection cylinder is decompressed by connecting it to the tank. The injector unit pauses after decompression for 2 seconds to allow the previously molded and hardened part to be removed from the mold. After the mold is closed, the injection cylinder goes forward and injects molten plastic into the mold.

Hydraulic Schematic

The schematic diagram of the unit hydraulic system is shown in Figure 2.

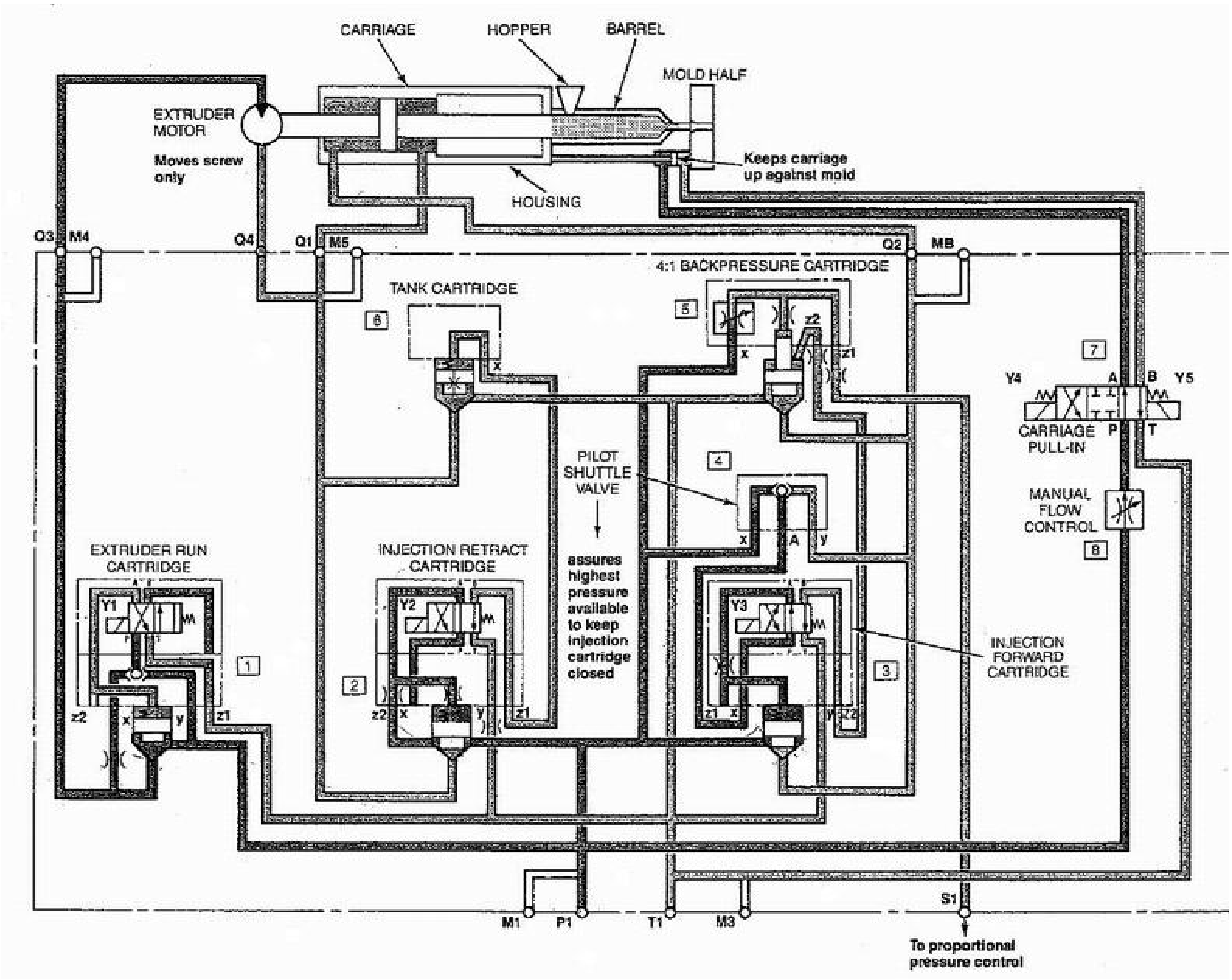
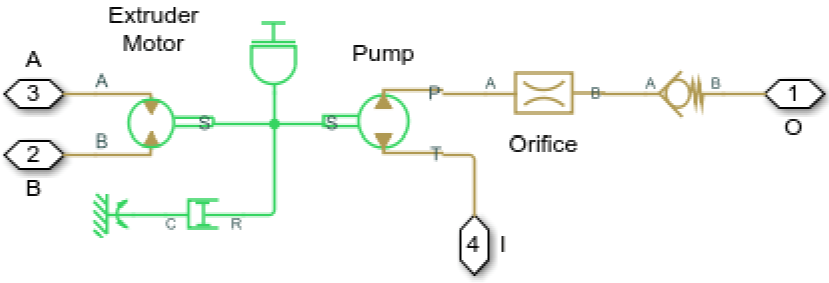
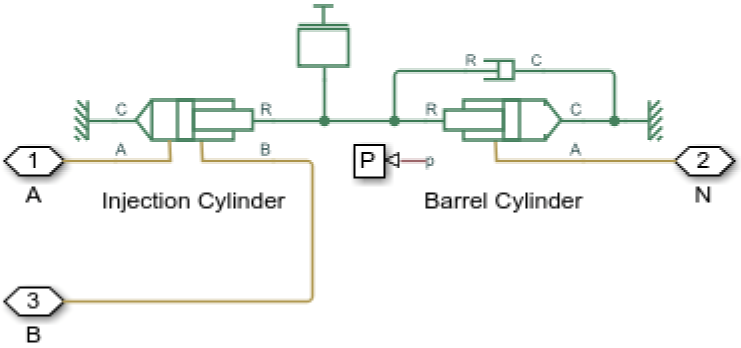


Figure 2. Injection Unit Hydraulic System Schematic Diagram

Extruder Subsystem

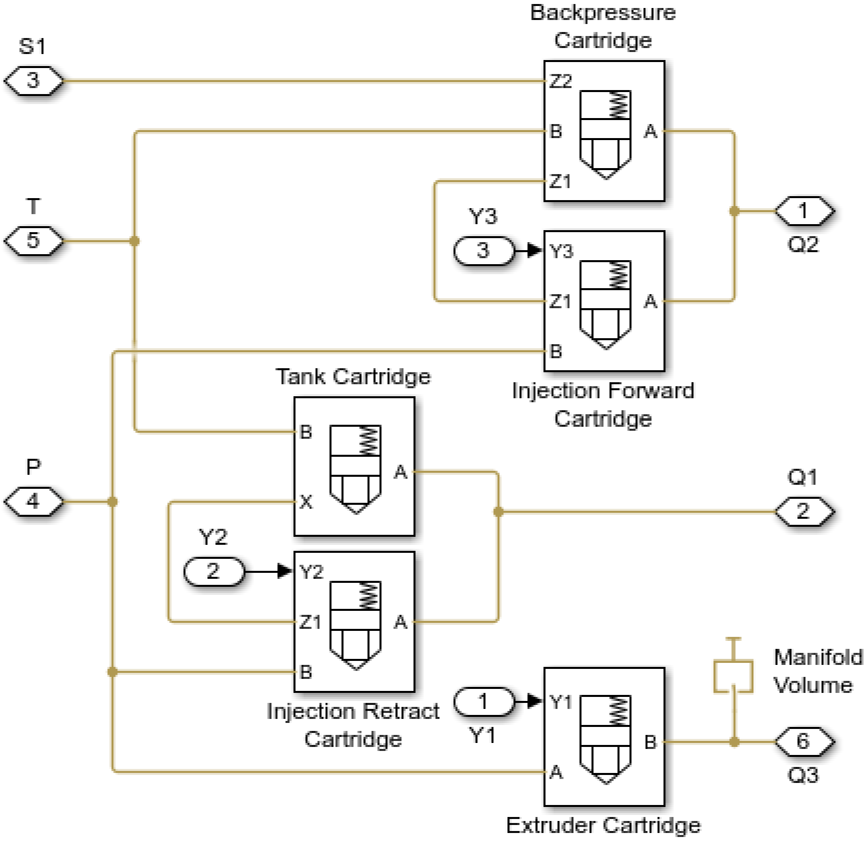


Injector Subsystem



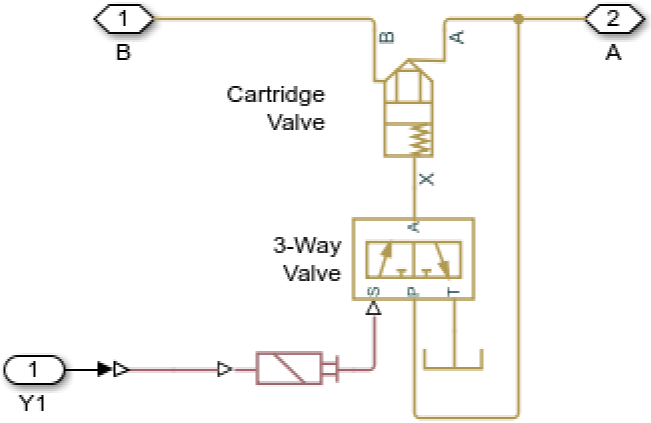
Valve Manifold Subsystem

The system is built of five cartridge valves that are arranged into the injection unit manifold:



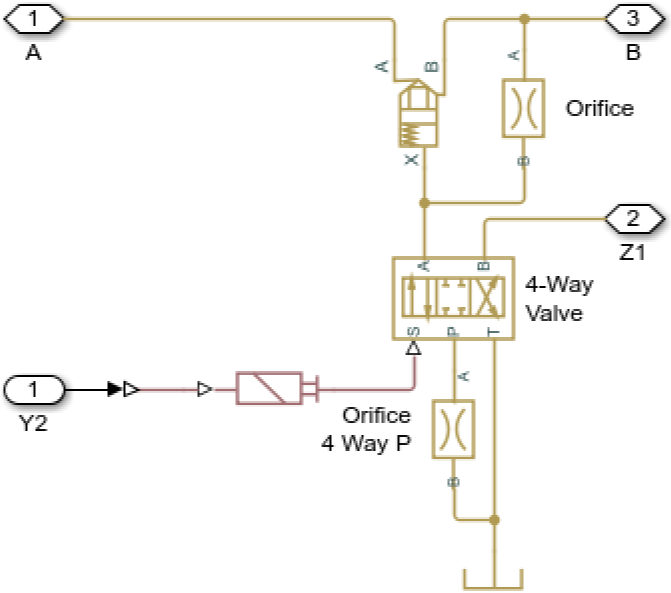
Extruder Cartridge Valve

The extruder cartridge valve controls rotation of the extruder motor. The motor starts rotating as electromagnet Y1 is energized.



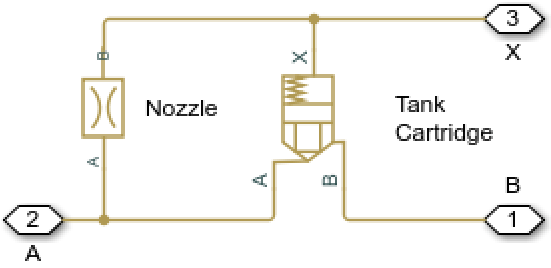
Injection Retract Cartridge Valve

The injection retract cartridge retracts the nozzle from the mold for a very short distance to allow the mold to be opened and the previously molded and hardened part to be removed. The cartridge poppet is opened as electromagnet Y2 is energized. When the electromagnet is deenergized, the poppet is closed and the control chamber of the tank cartridge valve is vented to allow free return flow from the extruder motor.



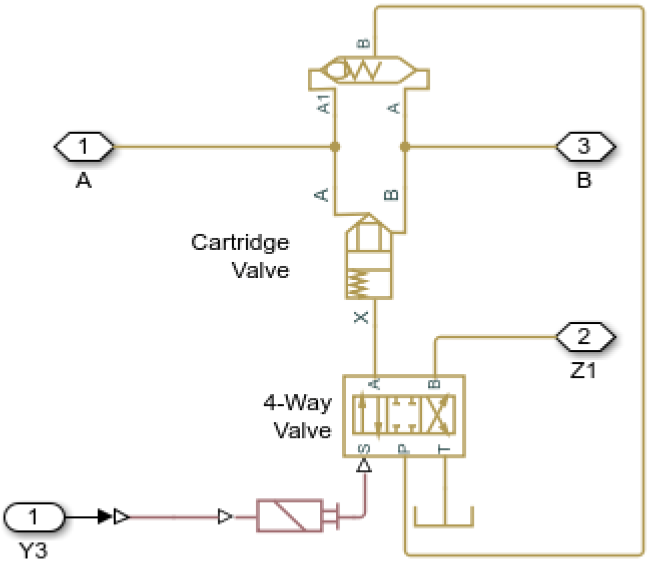
Tank Cartridge Valve

The tank cartridge valve connects extruder motor to the tank during the barrel fill



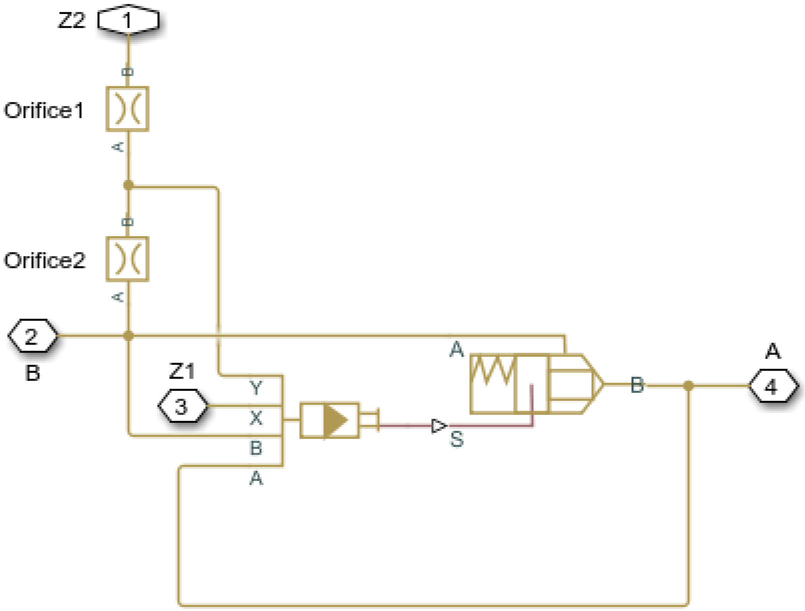
Injection Forward Cartridge Valve

The injection forward cartridge valve pushes the nozzle forward to contact the mold and injects molten plastic into the mold. The cartridge is controlled by electromagnet Y3. At neutral position, one of the control chambers of the back-pressure cartridge is vented to allow this cartridge to work as a pressure relief valve and maintain pressure set by the supply pressure at port Z1 and openings of orifices in the back-pressure cartridge covers.



Backpressure Cartridge Valve

The back-pressure cartridge valve maintains low back-pressure in the injection cylinder chamber to assure required quality of the molten plastic. The cartridge poppet is opened when electromagnet Y3 is deenergized. Simultaneously, the poppet of the back-pressure cartridge is closed by pressurizing the previously vented control chamber.



Control Signals

There are five signals that control this system:

1. Extruder - electromagnet Y1 signal

2. Retract - electromagnet Y2 signal

3. Inject - electromagnet Y3 signal

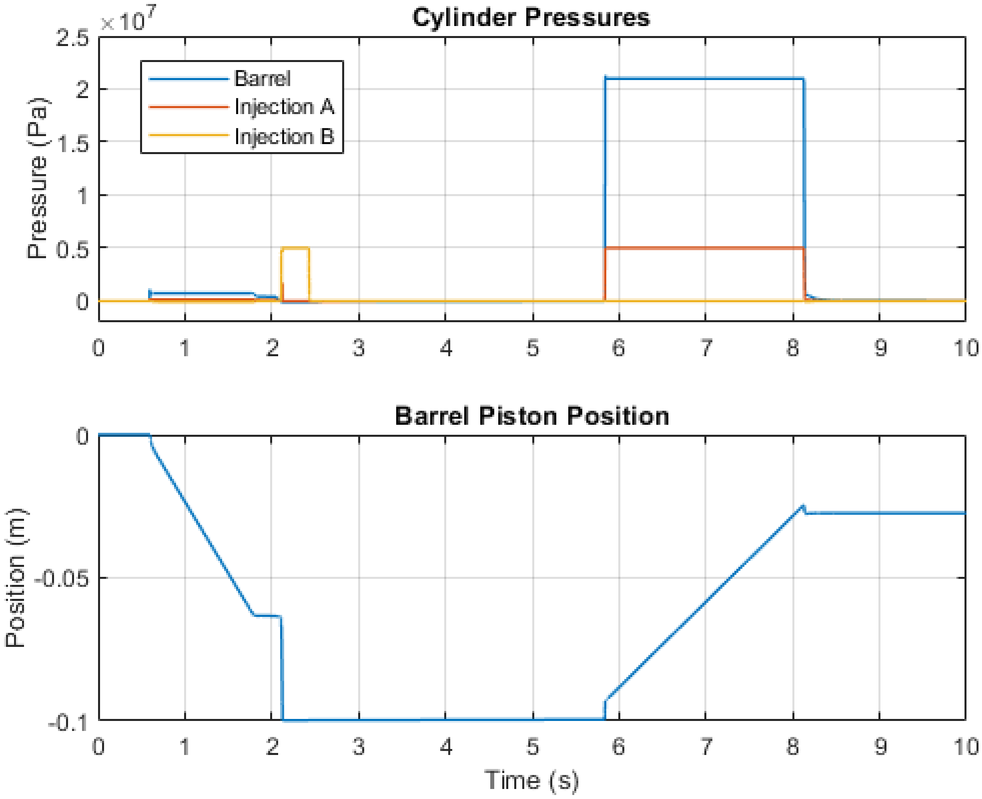
4. Nozzle - simulates the changing gap between the nozzle and the mold as a 2-way valve. In a real machine, the nozzle remains in contact with the mold while the extruder runs and the front portion of the barrel is filled. This contact prevents molten plastic from the spill. Since no mold is considered in the model, the nozzle is kept closed with the 2-way valve controlled by the signal 'Nozzle'. The valve is opened as the nozzle is moved away from the mold to keep things close to the real behavior.

5. Backpr - controls pressure level at port S1 (Figure 2) which establishes the pressure maintained by the back-pressure cartridge.

The purpose of remaining blocks is explained on the model schematic diagram (Figure 1).

Simulation Results from Simscape Logging

The plots below show the pressures in the injector chambers and the position of the barrel piston. The retract, decompression, and injection phases can be identified by the pressures and positions in the system.


Cylinder Pressures
×10^7
2.5
2
1.5
1
0.5
0
Pressure (Pa)
Barrel
Injection A
Injection B
0
1
2
3
4
5
6
7
8
9
10
Barrel Piston Position
0
-0.05
-0.1
Position (m)
Time (s)

Drill-Ream Actuator

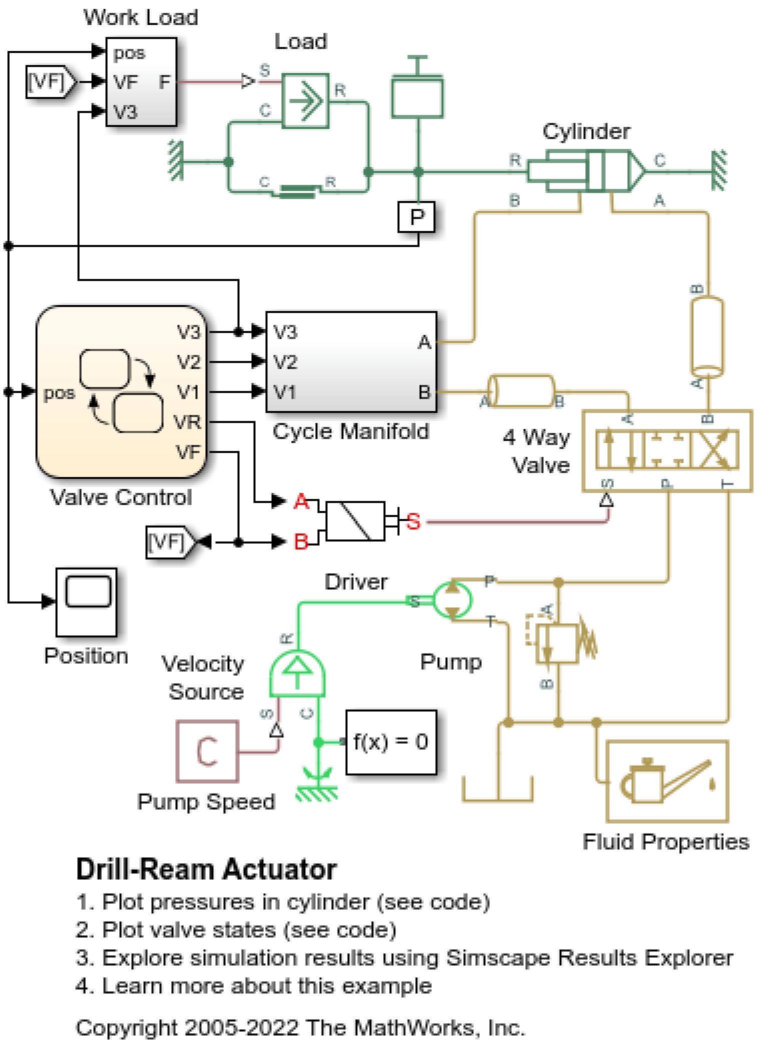
This example shows an actuator that drives a machine tool working unit performing a sequence of three technological operations: coarse drilling, fine drilling, and reaming. The actuator speed is controlled by one of three pressure-compensated flow control valves metering out return flow from the cylinder. The selection of an appropriate flow control is performed by directional valves that are activated by a control unit.

From the Valve Control, signals VF and VR are the signals that activate forward and return motions, respectively, by shifting the main directional valve. Signal V1 activates normally- closed valve Valve1 Retract. When opened, Valve1 Retract bypasses all flow controls and sets the fast traverse speed. V2 controls valve Valve2 Ream. When deenergized, this valve directs return flow through Flow Control Reaming, which is set to reaming speed. Signal V3 directs flow either through Flow Control Fine Drilling or Flow Control Coarse Drilling.

The load is simulated as a position-dependent force using a table lookup block.

Flow control valves are arranged as a subsystem comprised of a fixed orifice and a pressure compensator.

Model

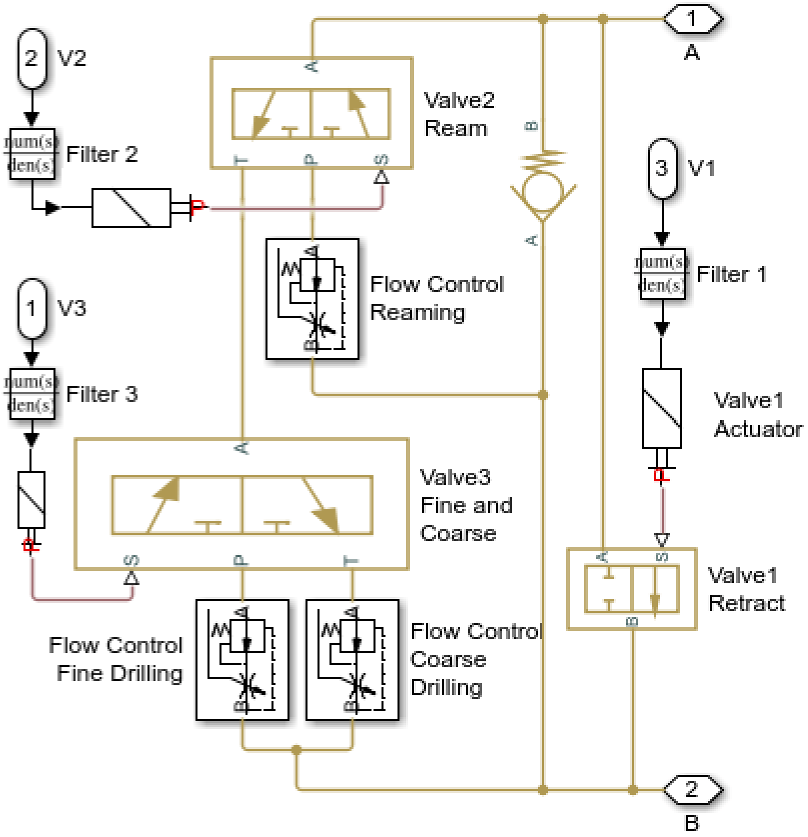


Drill-Ream Actuator

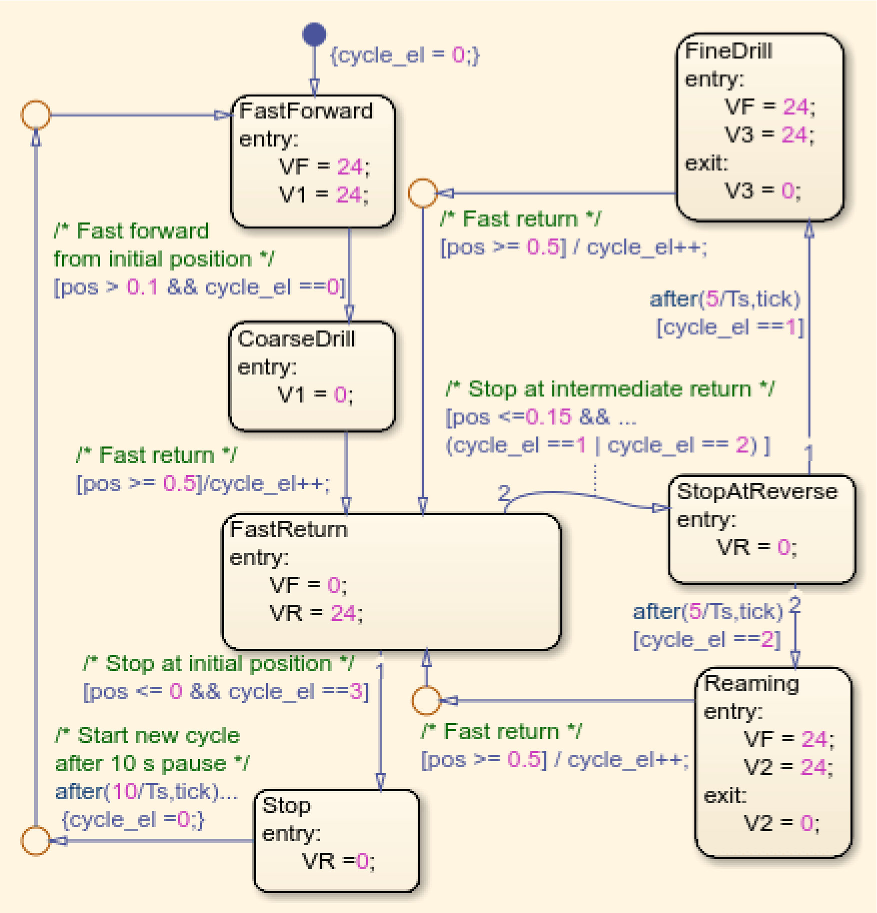
1. Plot pressures in cylinder (see code)
2. Plot valve states (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2005-2022 The MathWorks, Inc.

Cycle Manifold Subsystem

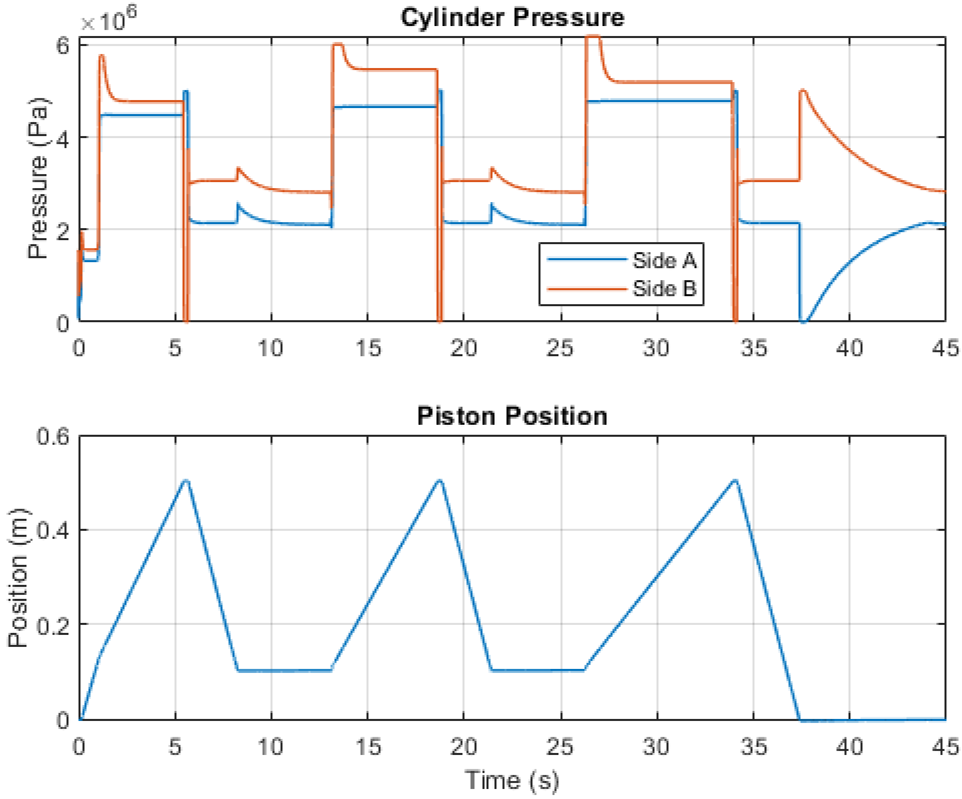


Valve Control Subsystem

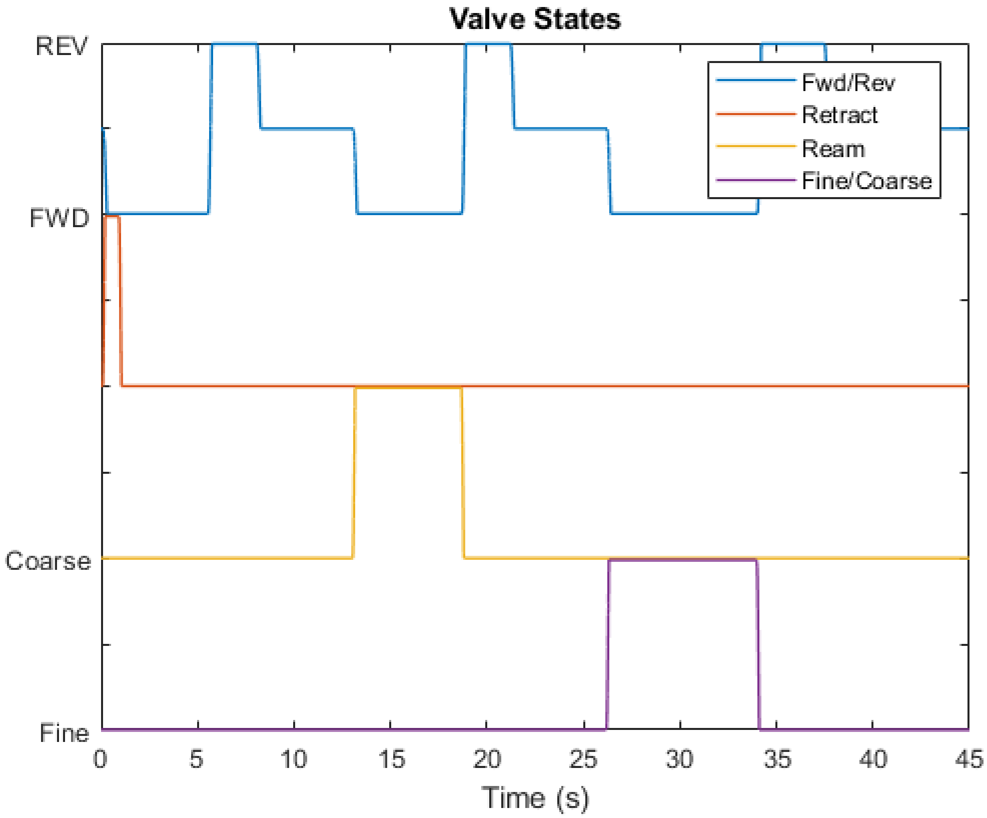


Simulation Results from Simscape Logging

The plots below show the pressures in the cylinder and the position of the piston. Three operating speeds are shown in the plots.



The plots below show the states of the different valves. Forward/Reverse has 3 states (Forward/Locked/Reverse), while the others have two states. Retract and Ream have states of on/off, and Fine/Coarse valve selects between two different flow restrictions.


Valve States
REV
FWD
Coarse
Fine
0
5
10
15
20
25
30
35
40
45
Time (s)
Fwd/Rev
Retract
Ream
Fine/Coarse

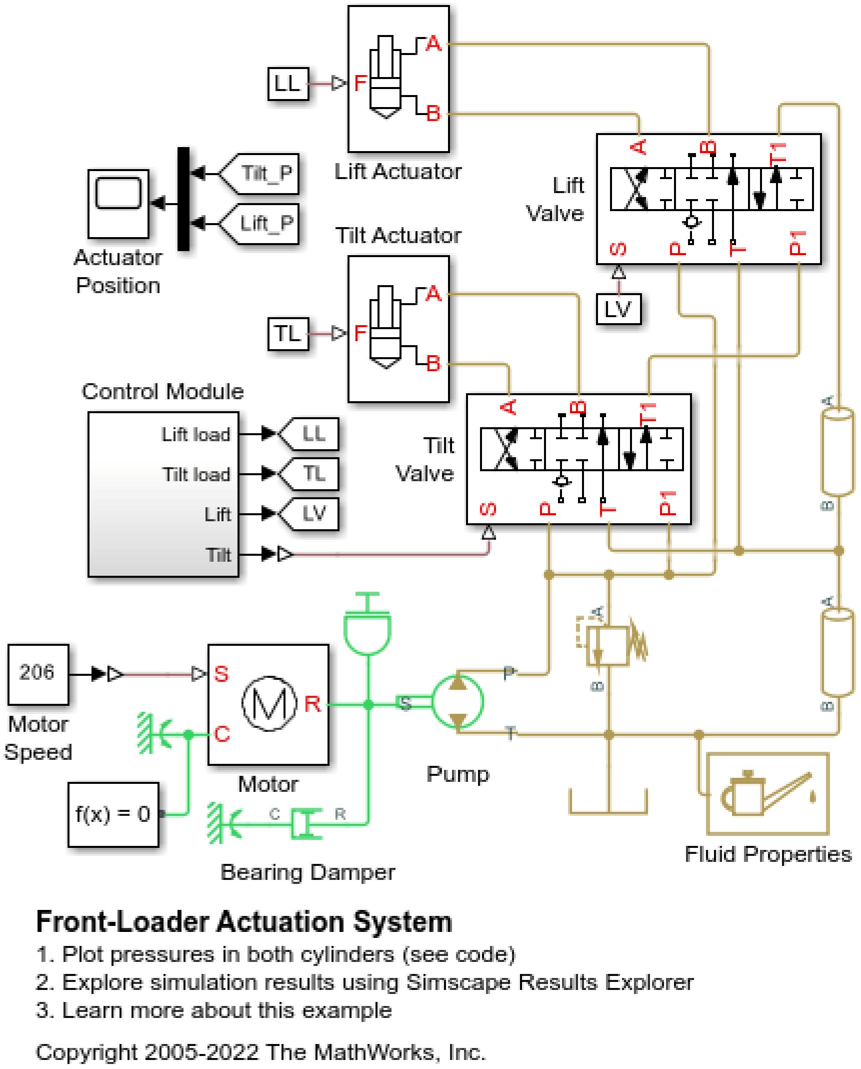
Front-Loader Actuation System

This example shows a simplified version of an actuation system consisting of the lift and tilt cylinders. Each cylinder is controlled by an open center, 6-way, 3-position directional valve. The valves are connected in series through their unloading branch such that the system pump is unloaded when both command levers are in neutral position. If either tilt or lift command is applied, the unloading path is closed.

The pump is driven by a diesel engine simulated with a model that accounts for velocity change caused by the load on the output shaft. The directional valve is a custom model composed of Orifice with Variable Area Round Holes blocks and a Check Valve block.

A typical cycle of a front-loader actuation system is simulated with all the motion and load commands generated as a function of time.

Model

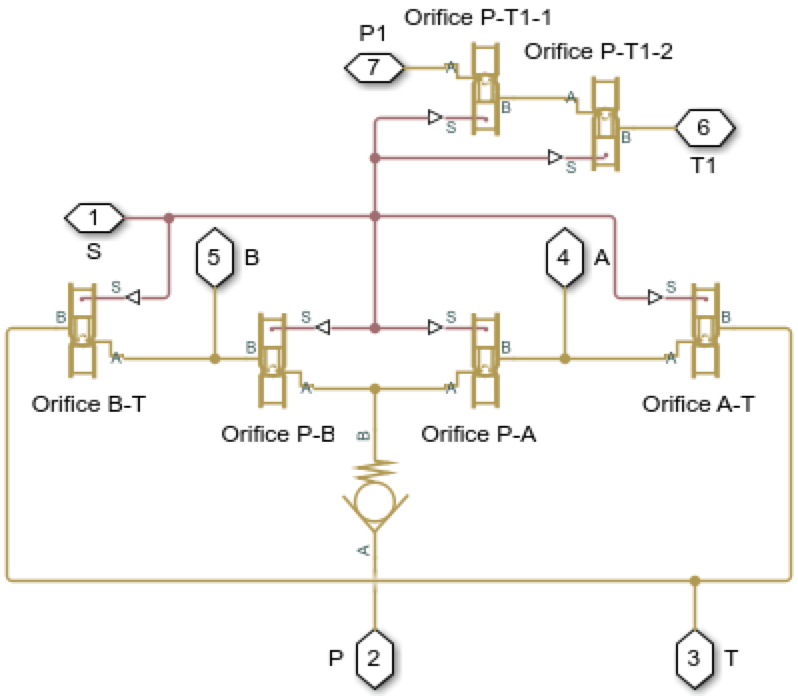


Front-Loader Actuation System

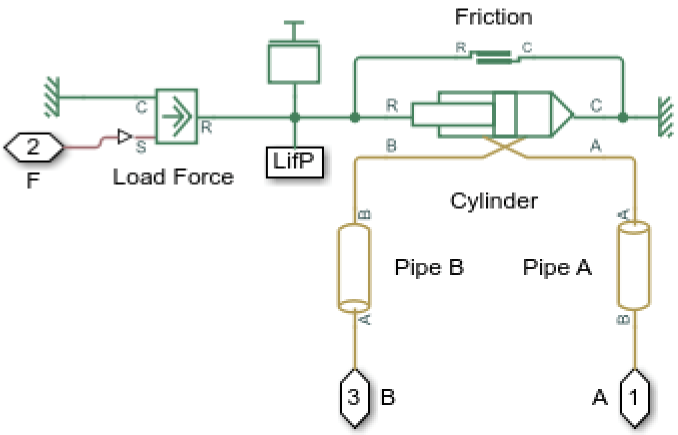
1. Plot pressures in both cylinders (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2005-2022 The MathWorks, Inc.

Lift Valve Subsystem

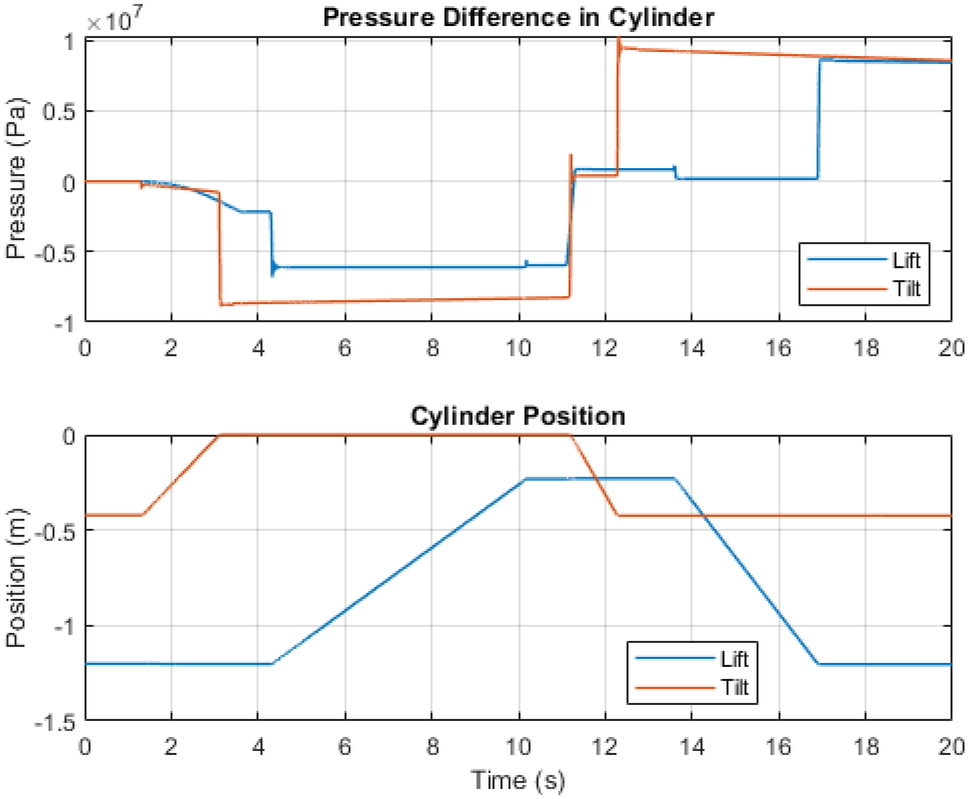


Lift Actuator Subsystem



Simulation Results from Simscape Logging

The plots below show the pressure difference in the lift and tilt cylinders and the position of the in each cylinder.


Pressure Difference in Cylinder
×10^7
1
0.5
0
-0.5
-1
Pressure (Pa)
0
2
4
6
8
10
12
14
16
18
20
Lift
Tilt
Cylinder Position
0
-0.5
-1
-1.5
Position (m)
0
2
4
6
8
10
12
14
16
18
20
Time (s)
Lift
Tilt

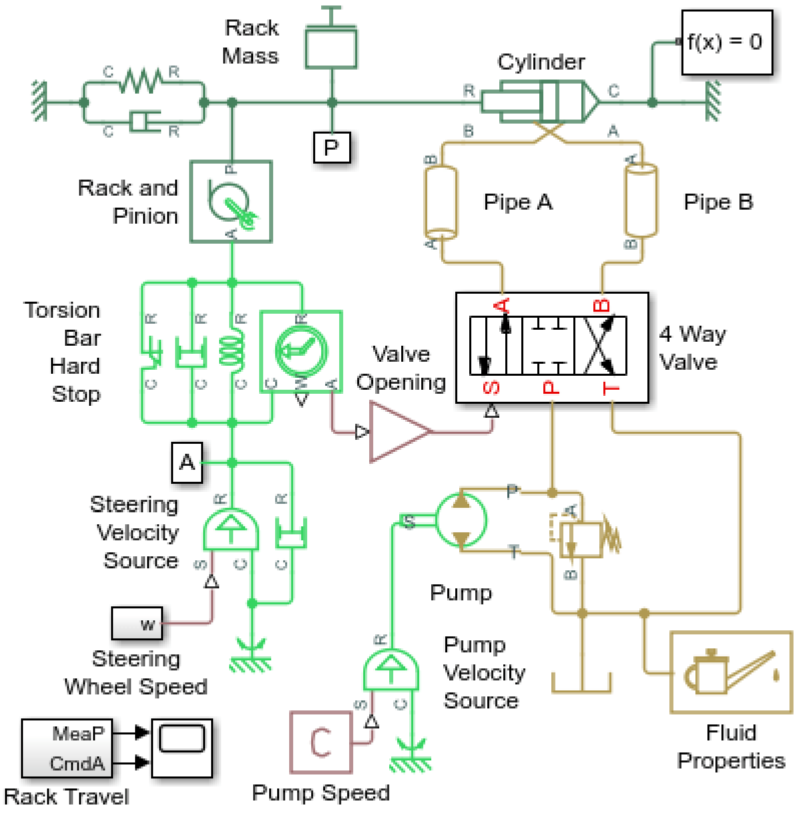
Power-Assisted Steering Mechanism

This example shows a simplified version of a power-assisted steering mechanism. The hydraulic actuation system includes a double-acting hydraulic cylinder, 4-way valve, fixed-displacement pump, and a pressure-relief valve. The steering rack acts against a load modeled by a spring and damper.

The valve opens based on the amount of twist of a torsion bar installed between the steering wheel and the pinion of a rack-and-pinion mechanism. The rotation of the steering wheel causes the torsion bar to twist relative to the pinion. The deformation of the bar is transformed into opening of the 4-way valve, which connects the chambers of the cylinder to pressure or return lines depending upon the direction of rotation. The PS Gain block Valve Opening converts torsion bar twist to the amount of opening for the 4-way valve. Setting the Gain parameter of this block to 0 disables the power steering.

If the torsion bar deformation exceeds 9 degrees the wheel is connected directly to the pinion through the hard stops installed in parallel with the torsion bar. The cylinder moves the steering rack which twists the torsion rod in the opposite direction until the valve is in the neutral position.

Model

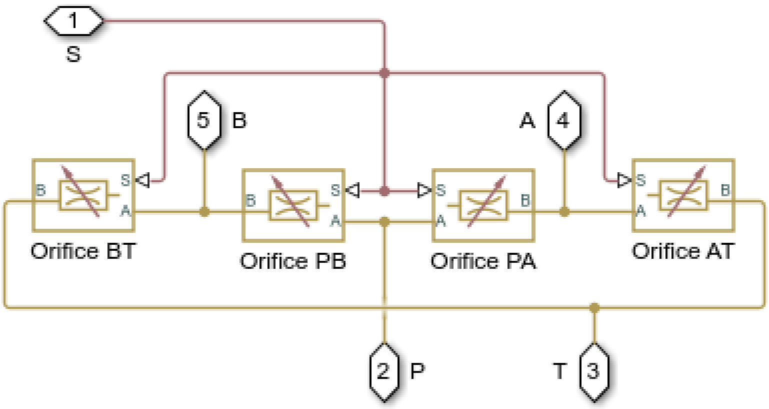


Power-Assisted Steering Mechanism

1. Plot pressures in cylinder (see code)
2. Compare effort with power steering on and off (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

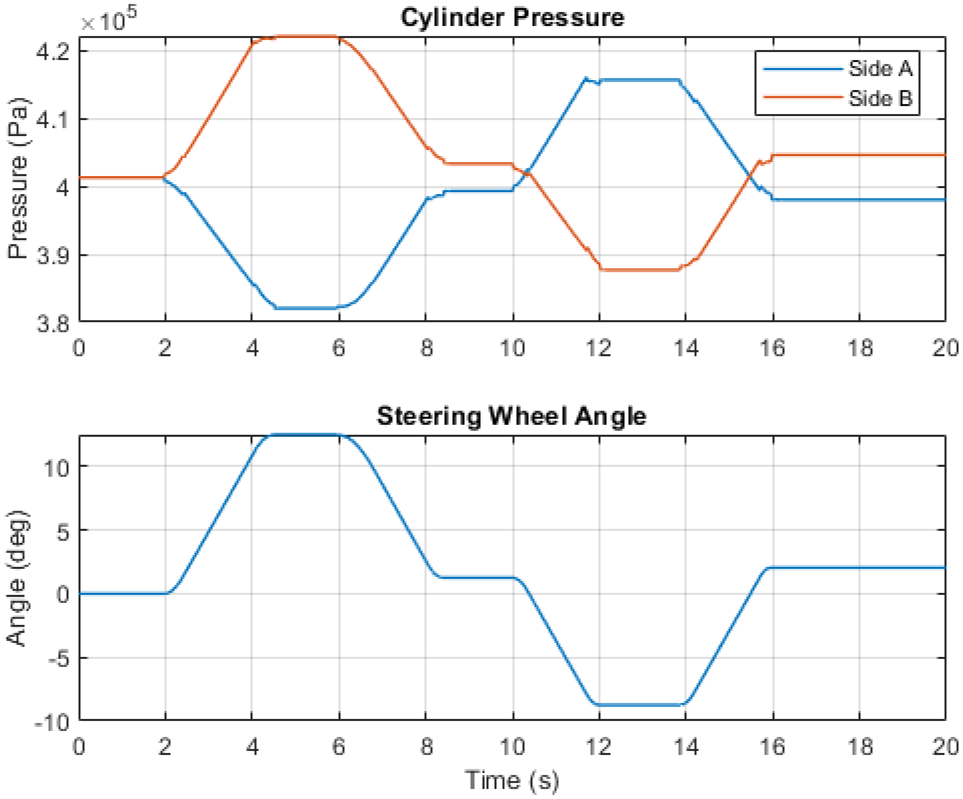
Copyright 2006-2022 The MathWorks, Inc.

4 Way Valve Subsystem

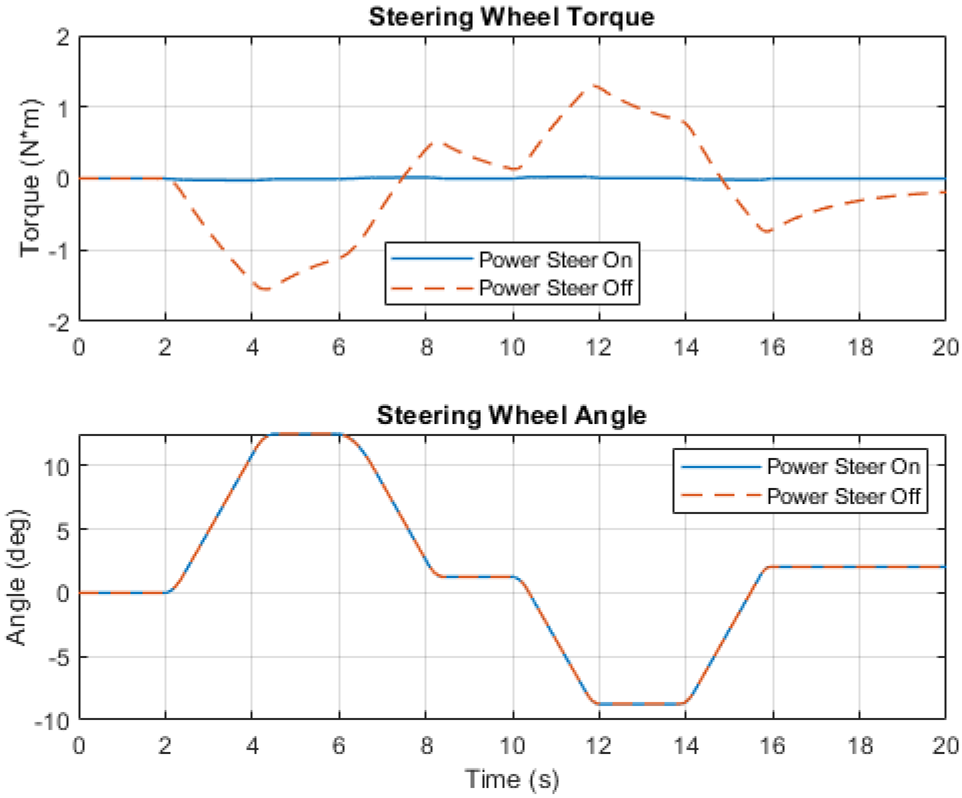


Simulation Results from Simscape Logging

The plots below show the amount of assistance the power steering system provides to the driver as the wheel is turned. The greater the difference in the cylinder chamber pressures, the more assistance that is provided.



The plots below compare the amount of effort required when the power steering is turned on and turned off. For the same steering sequence, the amount of torque required when the power steering is turned off is much higher. The power steering is turned off by setting the gain in Valve Opening to 0.

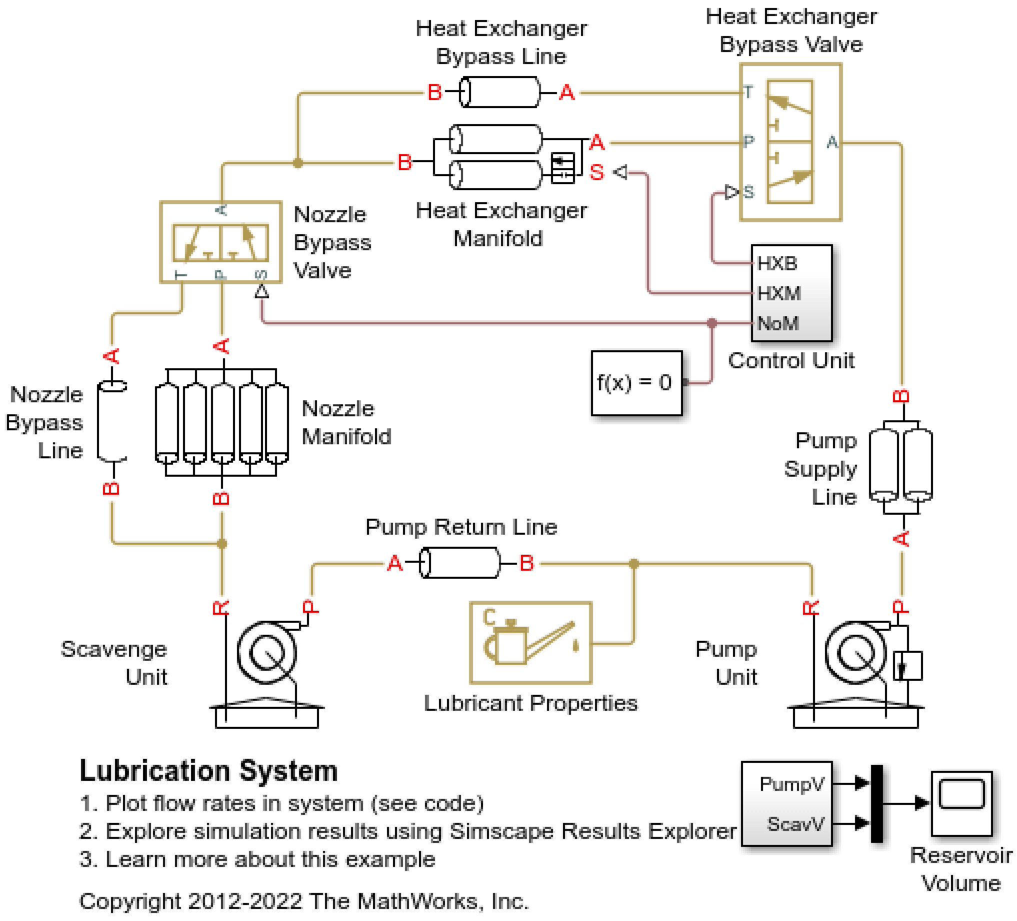

Steering Wheel Torque
Torque (N*m)
2
1
0
-1
-2
0
2
4
6
8
10
12
14
16
18
20
Power Steer On
Power Steer Off
Steering Wheel Angle
Angle (deg)
10
5
0
-5
-10
0
2
4
6
8
10
12
14
16
18
20
Time (s)
Power Steer On
Power Steer Off

Lubrication System

This example shows a simplified version of a lubrication system fed with the centrifugal pump. The system consists of five major units: Pump Unit, Scavenge Unit, Heat Exchanger Manifold, Nozzle Manifold, and Control Unit. Both the pump and the scavenge unit are built around the centrifugal pump. The Scavenge Unit collects fluid discharged by nozzles and pumps it back into the reservoir of the Pump Unit. The Control Unit generates commands to bypass either the heat exchanger, represented as a local resistance, or the nozzles block. In a real system, these commands are generated by temperature sensors installed in lubrication cavities.

The cycle simulated in the example allows you to observe pressures and flow rates at various points of the system. The example also shows the usage of local resistance blocks such as the Elbow, T-junction, and Sudden Area Change.

Model

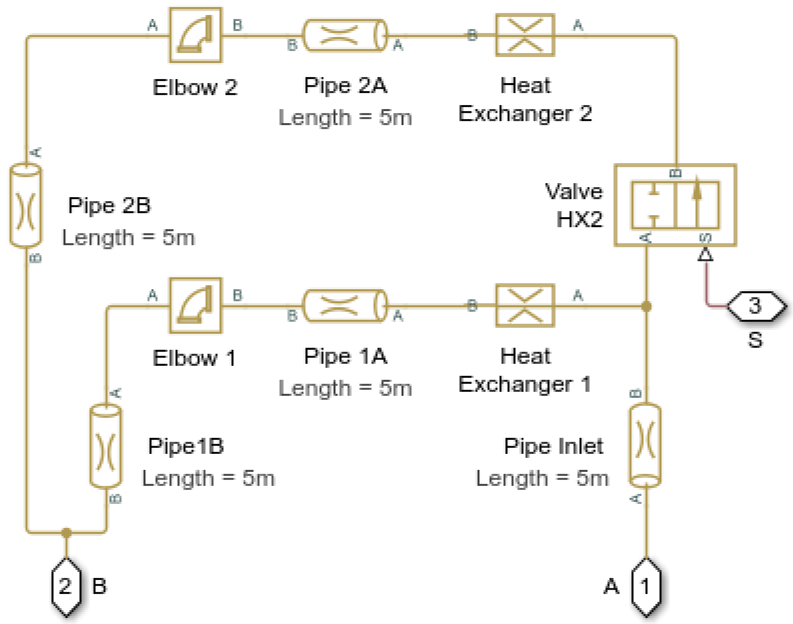


Lubrication System

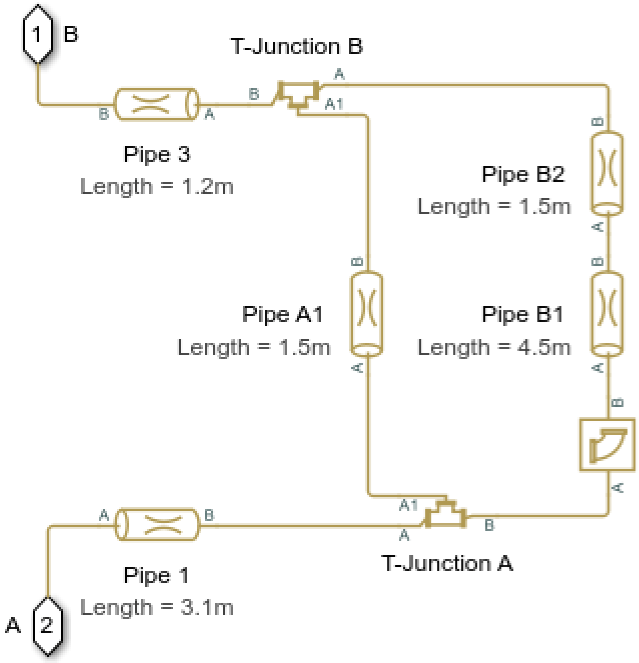
1. Plot flow rates in system (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2012-2022 The MathWorks, Inc.

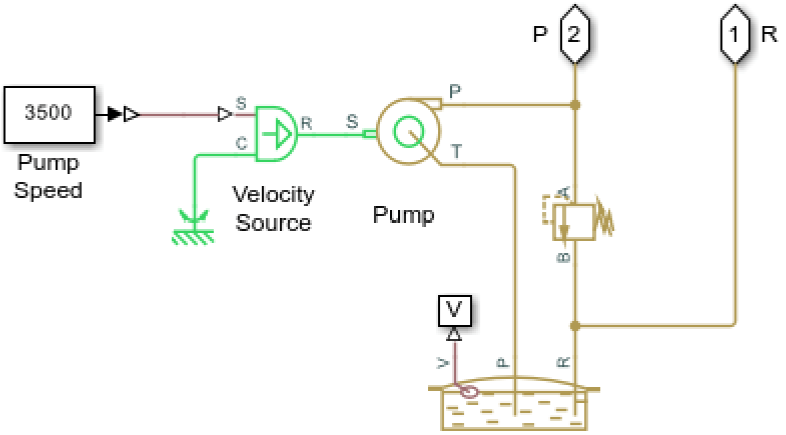
Heat Exchanger Manifold Subsystem



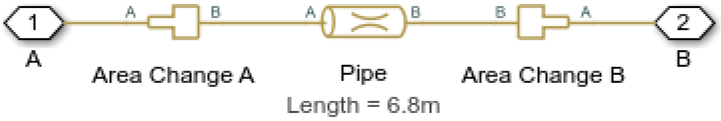
Pump Supply Line Subsystem



Pump Unit Subsystem

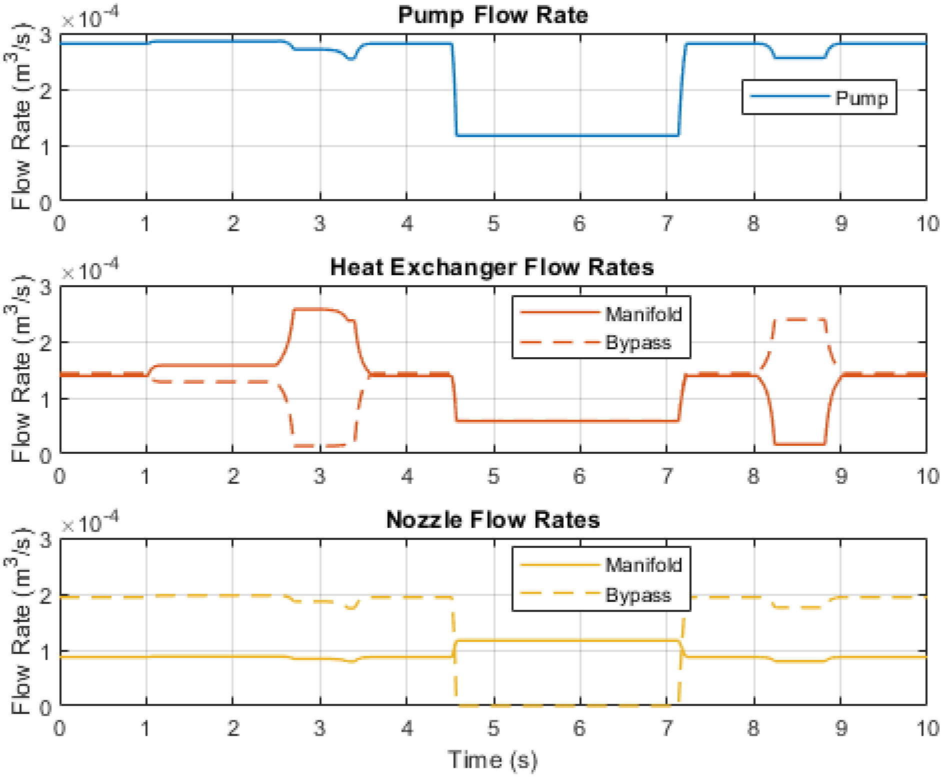


Pump Return Line Subsystem



Simulation Results from Simscape Logging

The plots below show the flow rates through different parts of the lubrication system. As the control member within the different valves are adjusted, flow rates change within the system.


Pump Flow Rate
×10^-4
Flow Rate (m^3/s)
0
1
2
3
Pump
0
1
2
3
4
5
6
7
8
9
10
Heat Exchanger Flow Rates
×10^-4
Flow Rate (m^3/s)
Manifold
Bypass
Nozzle Flow Rates
×10^-4
Flow Rate (m^3/s)
Manifold
Bypass
Time (s)

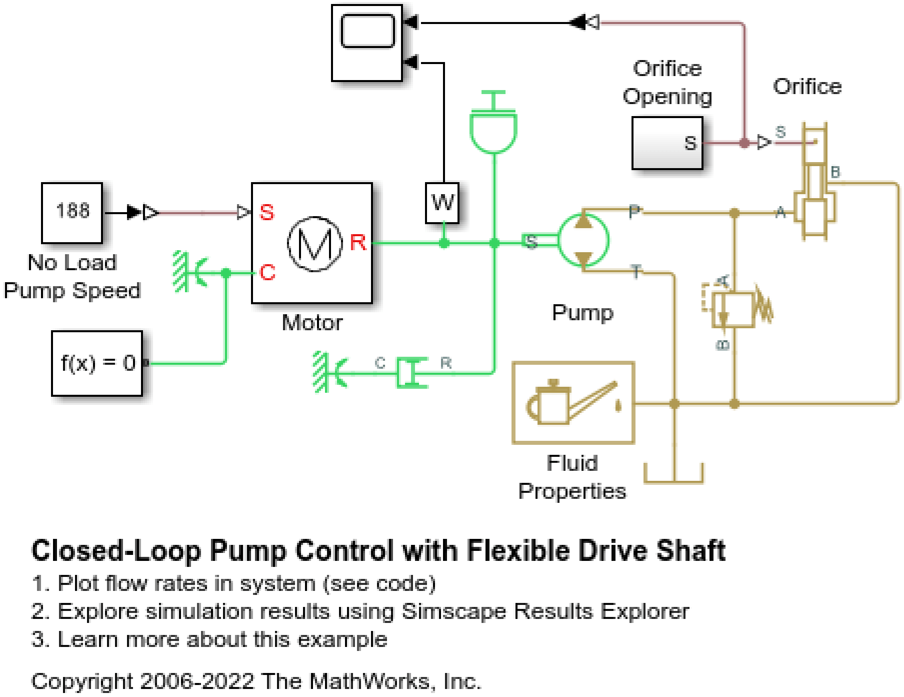
Closed-Loop Pump Control with Flexible Drive Shaft

This example shows a typical power unit consisting of a fixed-displacement pump driven by a motor through a flexible transmission and a pressure-relief valve. A variable orifice serves as a load for the system.

The motor model uses an Ideal Angular Velocity Source block. The load on the shaft decreases the velocity with a slip coefficient of 1.2 (rad/s)/Nm, resulting in a no-load speed for the motor of 188 rad/s. The load on the pump drive shaft is measured with the torque sensor. The shaft between the motor and the pump is assumed to be compliant and is simulated with a rotational spring and damper.

The simulation starts with the variable orifice open, which results in a low output pressure and the maximum flow rate going to the system. The orifice starts closing at 0.5 seconds and is closed completely at about 2.75 seconds. The output pressure builds up until it reaches 75e5 Pa, and is maintained at this level by the pressure-relief valve. Just after 3.2 seconds, the valve starts opening and the system returns to its initial state.

Model

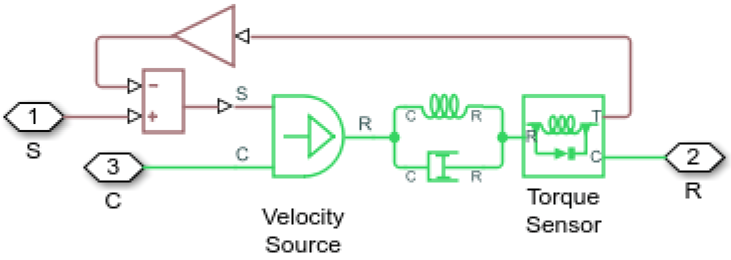


Closed-Loop Pump Control with Flexible Drive Shaft

1. Plot flow rates in system (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

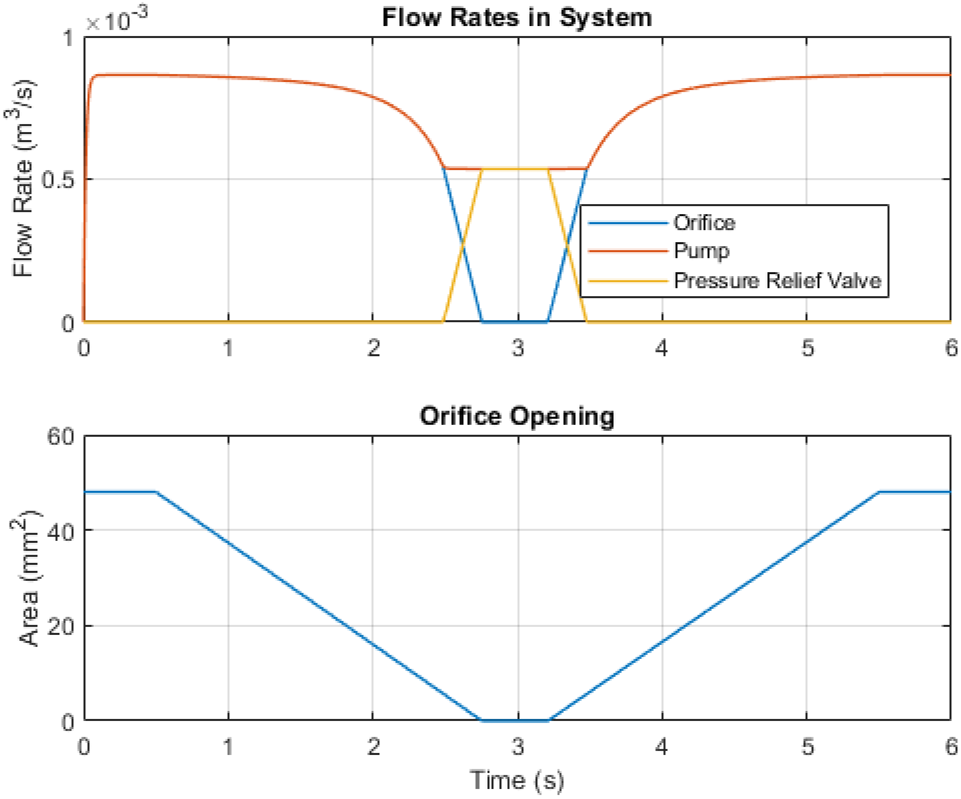
Copyright 2006-2022 The MathWorks, Inc.

Motor Subsystem



Simulation Results from Simscape Logging

The plots below show how flow rates vary in the system as the orifice is closed. All of the fluid exiting the pump goes through the orifice at the start. As the orifice is closed, flow gradually ends up being diverted through the pressure relief valve. Flow resumes through the orifice as it reopens.



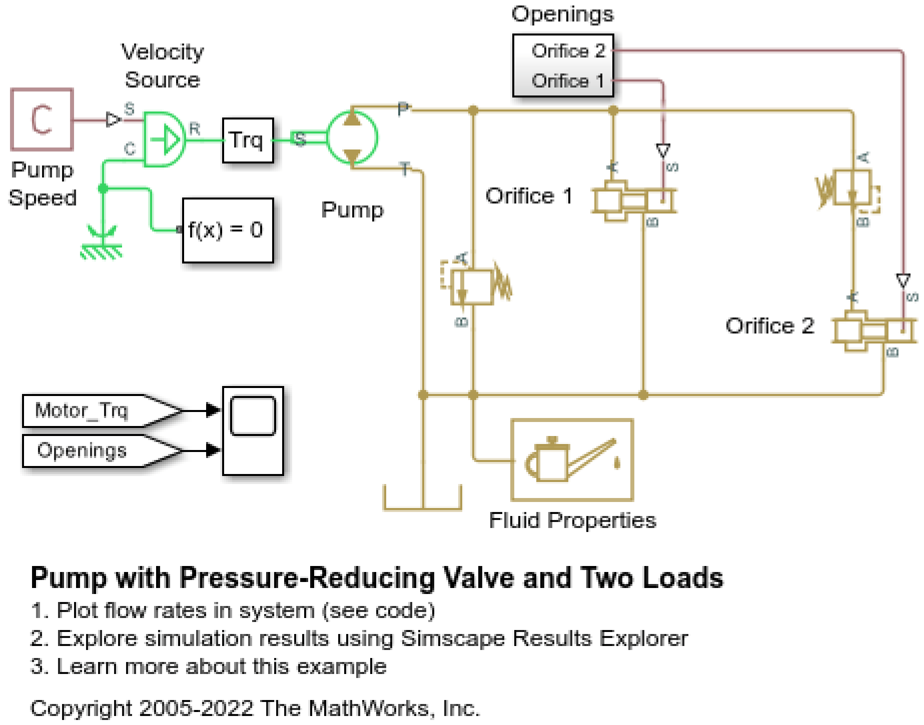
Pump with Pressure-Reducing Valve and Two Loads

This example shows a typical power unit consisting of a fixed-displacement pump driven by an angular velocity source, a pressure-relief valve, a pressure-reducing valve, and two variable orifices.

The variable orifices simulate fluid consumption in the main system and in the reduced pressure branch. The simulation starts with both variable orifices open, which results in a low output pressure and the maximum flow rate going to the system. The orifice for the main branch starts closing at 0.5 seconds and is closed completely at around 1.5 seconds.

The output pressure builds up until it reaches 75e5 Pa, and is maintained at this level by the pressure-relief valve. The pressure reducing valve in the reduced pressure branch is set to 20e5 Pa. It maintains this pressure at its output as long as the pressure upstream of the valve is higher than this setting.

Model

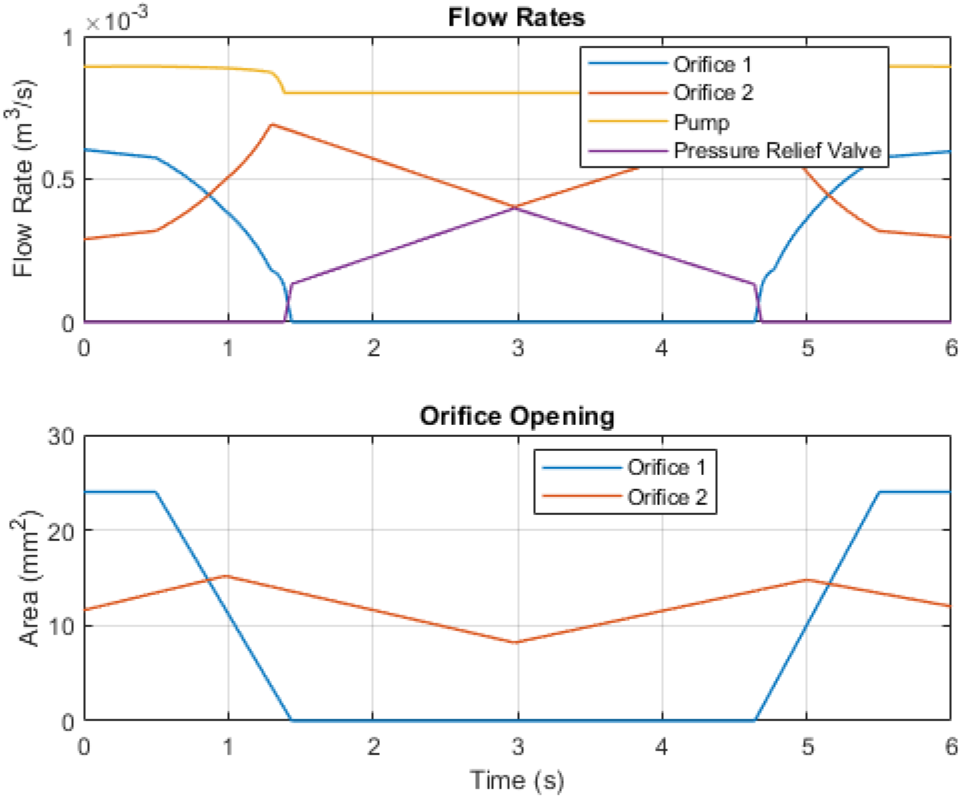


Pump with Pressure-Reducing Valve and Two Loads

1. Plot flow rates in system (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Simulation Results from Simscape Logging

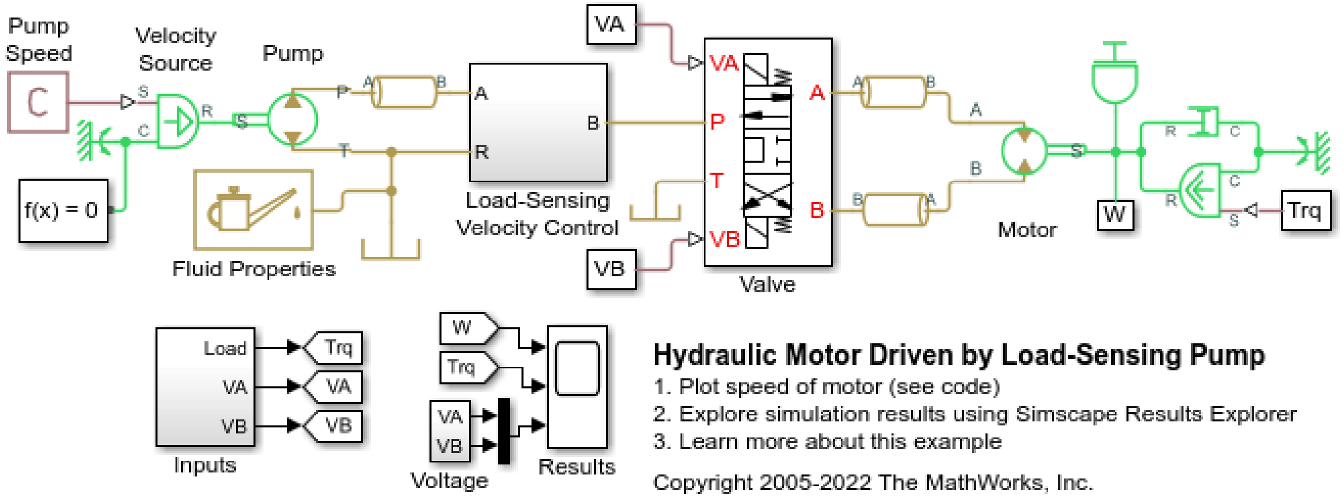
The plots below show how flow rates vary in the system as the orifices open and close. The fluid exiting the pump is divided between the orifices at the start. As the Orifice 1 and 2 is close, flow gradually ends up being diverted through the pressure relief valve.


Flow Rates
×10^-3
Flow Rate (m^3/s)
1
0.5
0
0
1
2
3
4
5
6
Orifice 1
Orifice 2
Pump
Pressure Relief Valve
Orifice Opening
Area (mm^2)
30
20
10
0
0
1
2
3
4
5
6
Orifice 1
Orifice 2
Time (s)

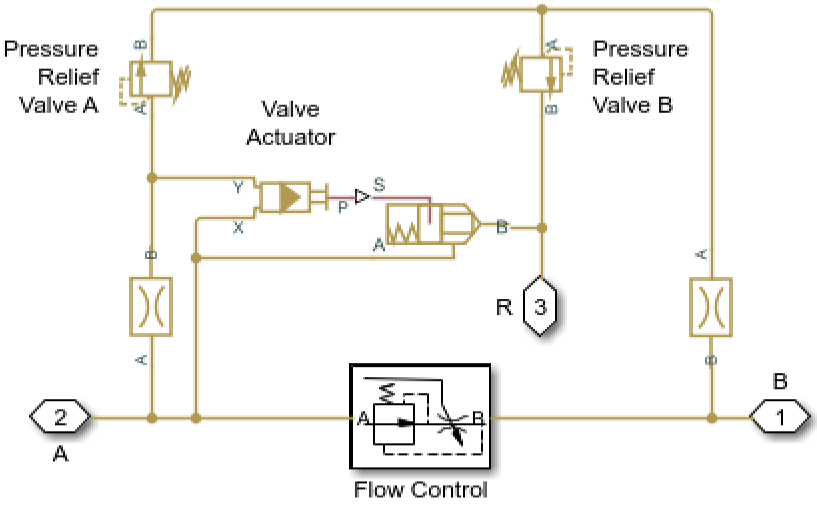
Hydraulic Motor Driven by Load-Sensing Pump

This example shows a circuit that is equipped with a load-sensing velocity regulator installed between the pump and directional valve. Unlike a conventional meter-in velocity control, the load-sensing device automatically adjusts output pressure of the pump in such a way that it equals the sum of the preset pressure drop across the pressure-compensated flow control valve and the pressure induced by load. The pilot-operated pressure-relief valve in the Load-Sensing Velocity Control block is built of the Poppet Valve and the Hydraulic Double-Acting Valve Actuator blocks.

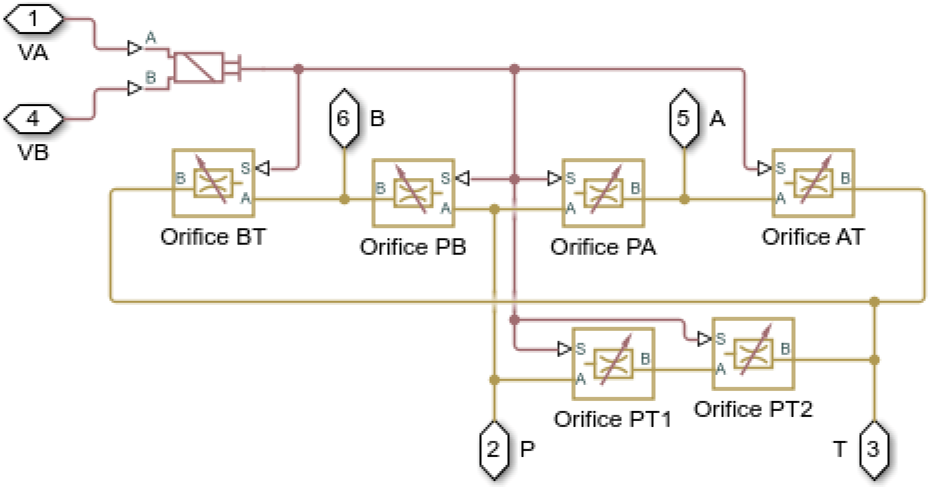
Model


Pump Speed
Velocity Source
Pump
f(x) = 0
Fluid Properties
Load-Sensing Velocity Control
VA
VB
P
T
A
B
R
Valve
Motor
W
Trq
Load
Inputs
Voltage
Results
Hydraulic Motor Driven by Load-Sensing Pump
1. Plot speed of motor (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example
Copyright 2005-2022 The MathWorks, Inc.

Load-Sensing Velocity Control Subsystem

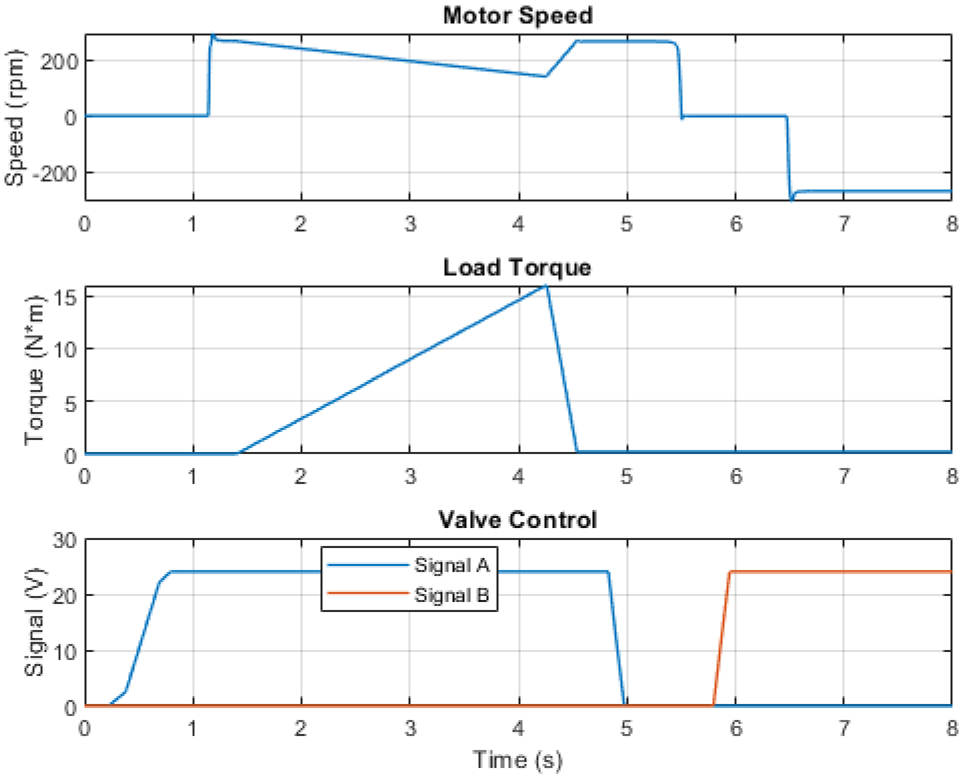


Valve Subsystem



Simulation Results from Simscape Logging

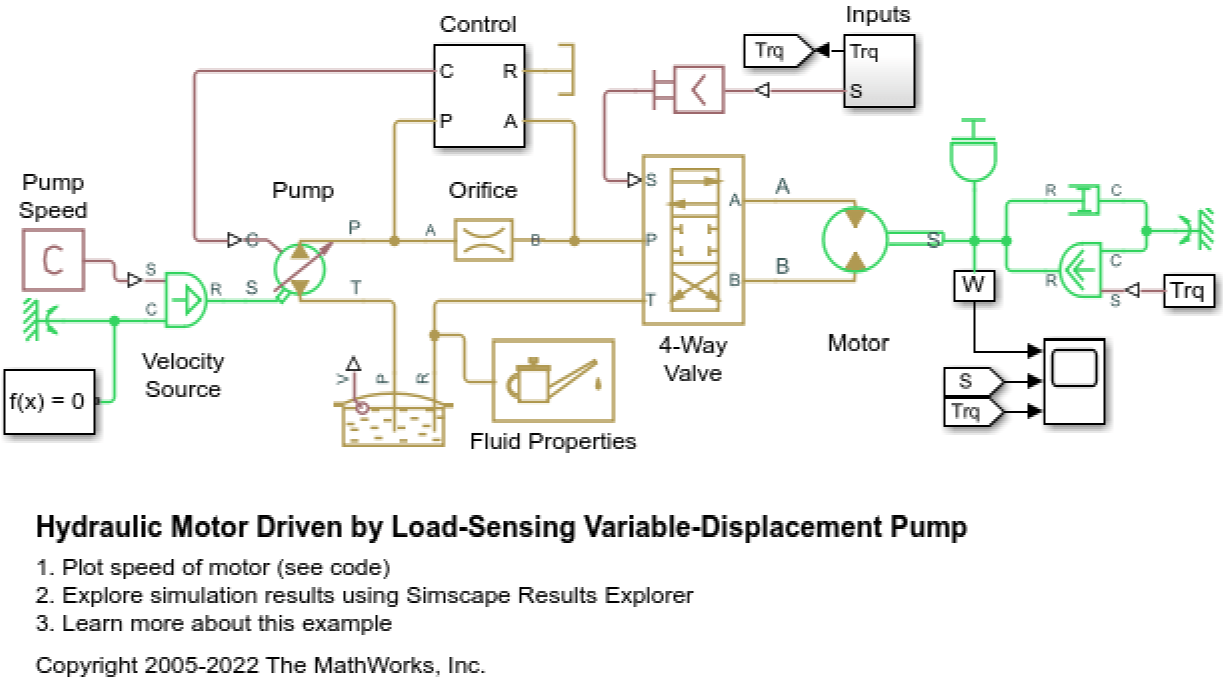
The plots below show the behavior of the motor. The motor speed varies as the load torque is applied. The valve controls pressure to the motor such that it can spin forward, backward, or be locked in place.



Hydraulic Motor Driven by Load-Sensing Variable-Displacement Pump

This example shows a circuit using a load-sensing and pressure-limiting unit in a conventional reciprocal system with variable load on the forward stroke. The unit limits output pressure to 300 bar and maintains a preset pressure drop in 10 bar across velocity control orifice on the discharge port of the pump. The unit is built of two 3-way, 2-position valves, two single-acting hydraulic valve actuators, and one double-acting hydraulic valve actuator.

Model

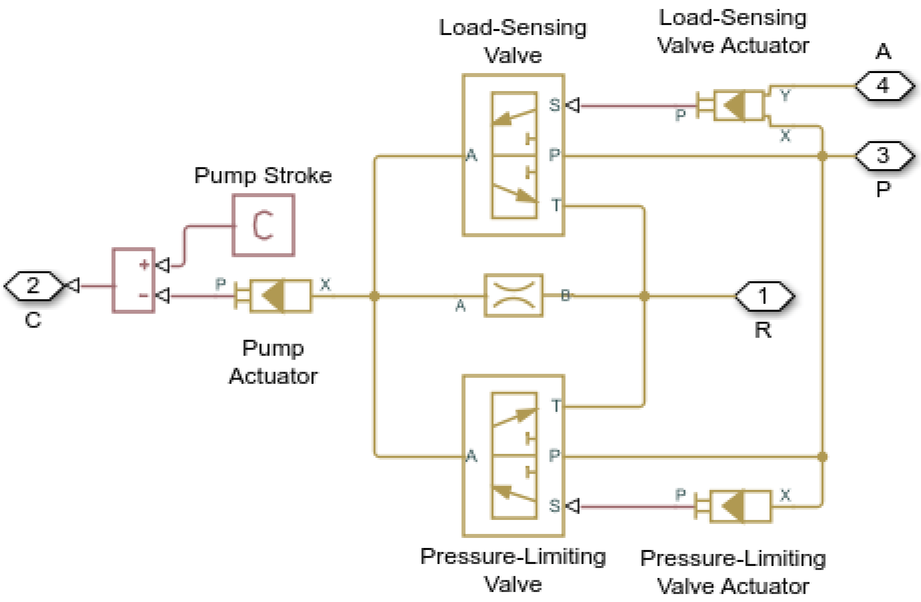


Hydraulic Motor Driven by Load-Sensing Variable-Displacement Pump

1. Plot speed of motor (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

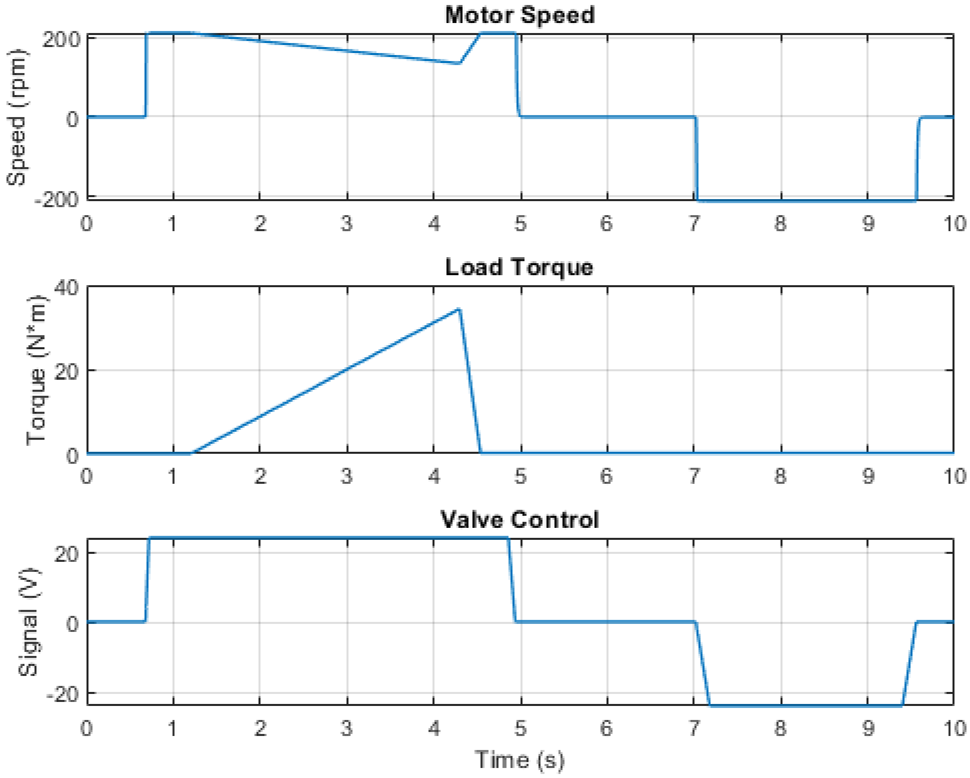
Copyright 2005-2022 The MathWorks, Inc.

Control Subsystem



Simulation Results from Simscape Logging

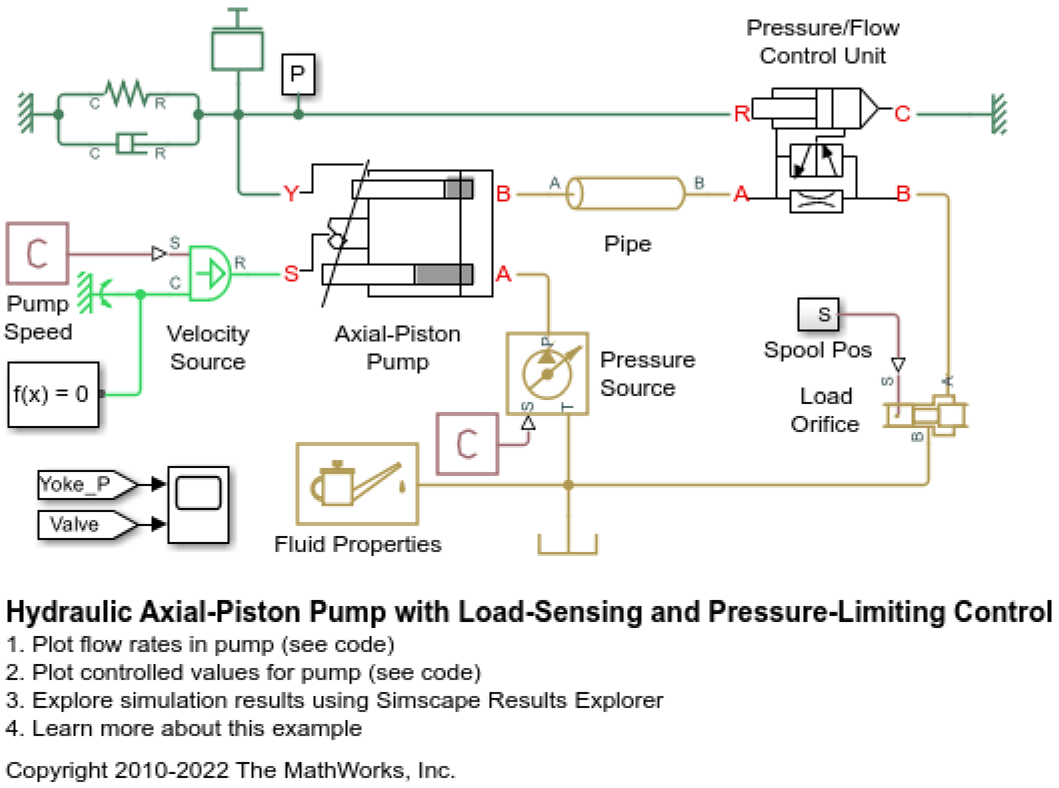
The plots below show the behavior of the motor as the valve inputs and load torque are varied.


Motor Speed
Speed (rpm)
200
0
-200
0
1
2
3
4
5
6
7
8
9
10
Load Torque
Torque (N*m)
40
20
0
Valve Control
Signal (V)
20
0
-20
Time (s)

Hydraulic Axial-Piston Pump with Load-Sensing and Pressure-Limiting Control

This example shows a test rig designed to investigate interaction between an axial-piston pump and a typical control unit, simultaneously performing the load-sensing and pressure-limiting functions. To assure required accuracy, the model of the pump must account for such features as interaction between pistons, swash plate, and porting plate, which makes it necessary to build a detailed pump model.

Model



Hydraulic Axial-Piston Pump with Load-Sensing and Pressure-Limiting Control

1. Plot flow rates in pump (see code)
2. Plot controlled values for pump (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Figure 1. Test Rig Schematic

Test Rig Description

The pump model is represented by the subsystem named Axial-Piston Pump. The prime mover rotating the pump is simulated with the Ideal Angular Velocity source. The pump output passes through a pipeline, control unit, and a variable orifice that acts as a load. To test control unit response to variable load, the orifice changes its area during simulation. The change profile is implemented by the Spool Pos subsystem.

The control unit in the test rig is represented by the subsystem named Pressure/Flow Control Unit. The load-sensing function of the pump control uses a fixed orifice. The control unit keeps the pressure differential across this orifice constant, regardless of pump loading. The control unit receives signals on pump output pressure and load pressure, measured after the flow control valve. Based on these pressures, the unit produces yoke displacement, which affects the angular position of the angled swash plate in the pump. This helps maintain the specified pressure differential across the flow control valve and prevent pump pressure from exceeding the preset value.

Test rig basic parameters:

Pump maximum displacement	7.8877e-6 m^3/rad
Pitch radius	0.04 m
Piston area	1.77e-4 m^2
Number of pistons	5
Maximum piston stroke	0.06 m
Swash plate maximum angle	35 deg (0.6109 rad)
Arm length between the actuator and the swash plate pivoting point	0.055 m
Swash plate actuator stroke	0.04 m
Diameter of the orifice at the bottom of the piston chamber	0.007 m
Pump maximum rated speed	260 rad/s (2482 RPM)
Maximum pressure	270 bar
Rated flow	1.1e-3 m^3/s

All basic parameters are specified through the **Model Properties** > **Callbacks** > **InitFcn** option.

Axial-Piston Pump Model

The pump under investigation is an axial-piston pump with five pistons.

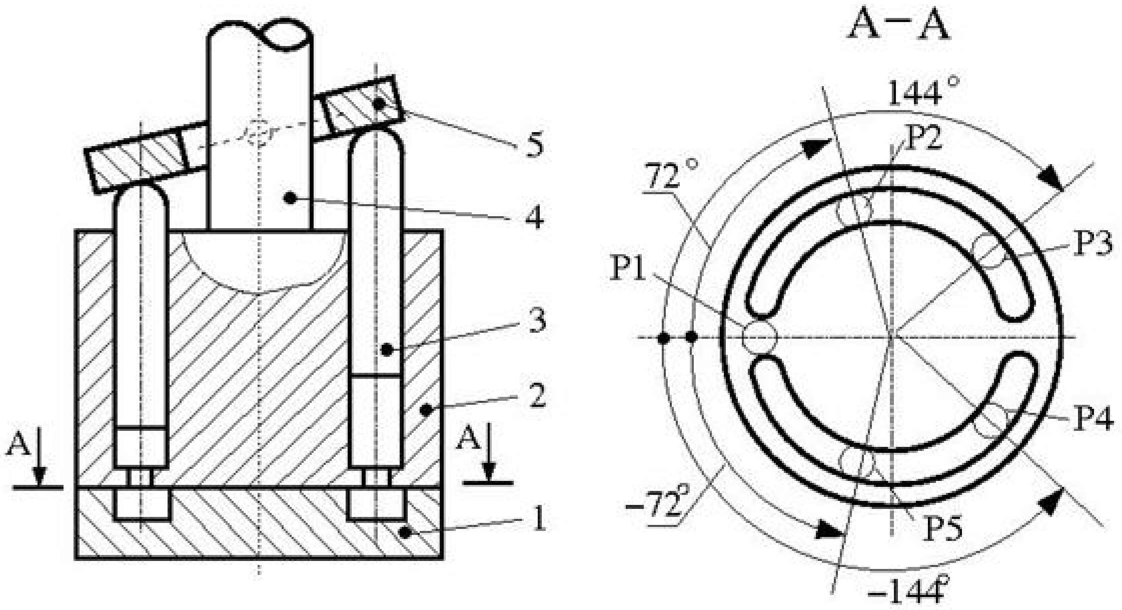


Figure 2. Axial-Piston Pump Schematic

The pump schematic diagram is shown in Figure 2, where:

1 - porting plate

2 - cylinder block (rotor)

3 - piston

4 - driving shaft

5 - swash plate

The block diagram of the pump model is shown in Figure 3.

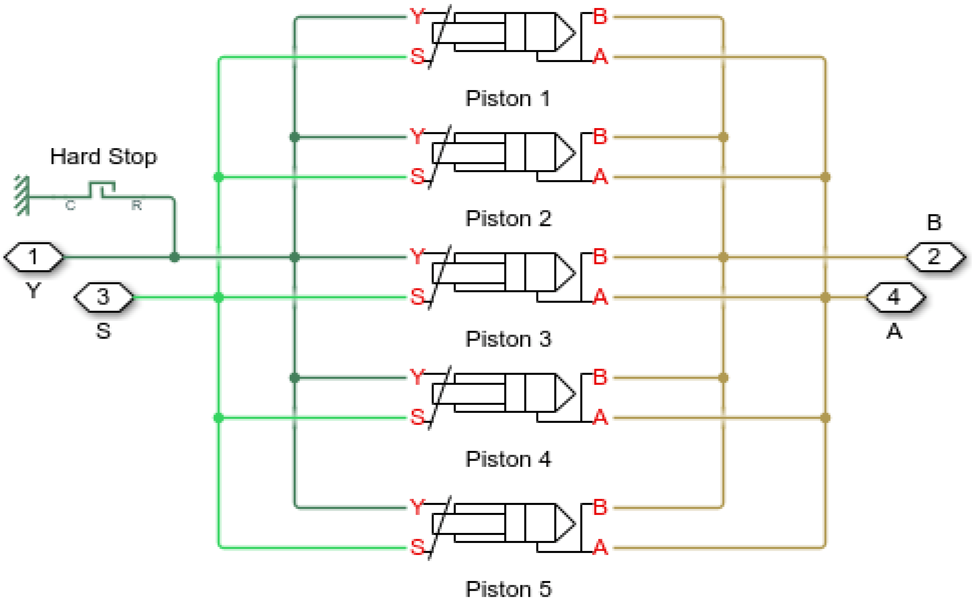


Figure 3. Pump Model

Every piston of the pump is represented by a subsystem called Piston. These subsystems are identical and are connected to the following external ports of the pump model:

- S - pump driving shaft
- Y - yoke connected with the inclined plate of the swash mechanism
- P - pump discharge

The suction ports of all pistons (ports A) are connected to the output of a low pressure booster pump, which is simulated with the Ideal Hydraulic Pressure Source block. The output pressure of the booster pump is set to 5e5 Pa.

The yoke is connected to the Y ports of all pistons, thus acting on the angled plate of the swash mechanism. The displacement of the yoke is limited by the hard stop.

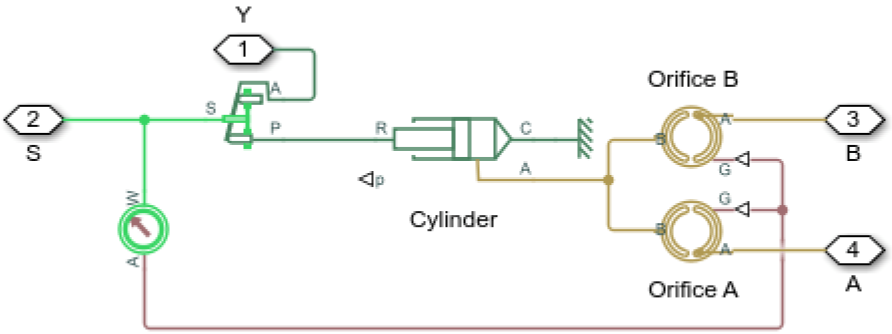


Figure 4. Piston Model

The model of the piston (Figure 4) is based on the Single-Acting Hydraulic Cylinder block, which is mechanically connected to the drive shaft through the Swash Plate block. The cylinder is also hydraulically connected to ports A and B through the Porting Plate Variable Orifice blocks. Ports A and B represent pump discharge and intake ports, respectively.

The port assignment is performed on the basis of the following considerations:

- The pistons are evenly spread along the pitch circle of the piston cylinder, as shown in Figure 2. This makes the angle between pistons 360/5 = 72 degrees.
- Let us assume that the first piston (marked P1 in the schematic) is located exactly at the reference point that corresponds to the lowest piston position. Let us further assume that port A represents the intake outlet of the pump. In other words, the piston moving along slot A in positive direction (clockwise in this case) goes up, and its chamber is filled with fluid by a booster pump. This means that the **Phase angle** parameter of the Porting Plate Variable Orifice A in Piston 1 must be set to zero. The same parameter of the Porting Plate Variable Orifice B in Piston 1 must be set to 180 degrees, because it starts interacting with slot B (pump discharge port) only after rotation by 180 degrees.

In the piston model, parameters **Phase angle** of each Porting Plate Variable Orifice block are denoted as **Phase angle A** and **Phase angle B**, respectively. The values of the phase angle for all five pistons are computed in the initialization section of the Axial-Piston Pump subsystem mask editor. The following table shows their values in degrees, with the corresponding values in radians given in parenthesis:

Piston name in the pump model	Phase angle A	Phase angle B	Phase angle in Swash Plate block
Piston_1	0	180 (pi)	0
Piston_2	72 (1.2566)	-108 (-1.885)	72 (1.2566)
Piston_3	144 (5.5133)	-36 (-0.8029)	144 (5.5133)
Piston_4	-144 (-5.5133)	36 (0.8029)	-144 (-5.5133)
Piston_5	-72 (-1.2566)	108 (1.885)	-72 (-1.2566)

The Swash Plate block in the piston model also requires phase angle to be assigned, to specify the position of a piston with respect to the inclined surface. With the selected reference point, the values of the swash plate phase angle coincide with the Phase angle A values, as shown in the table.

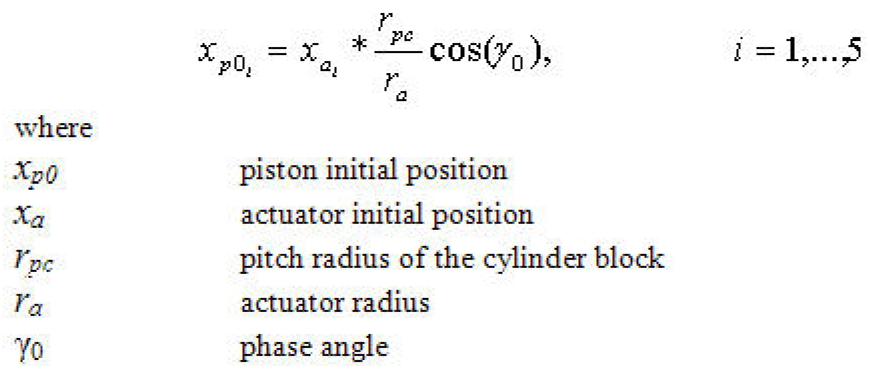
The Porting Plate Variable Orifice blocks require angular position of the respective piston at their input. This function is performed by the Angle Sensor block.

Other important parameters are the stroke of the cylinder and the initial position of the piston with respect to the cylinder cap. The stroke must be large enough to allow the piston to reciprocate even at the maximum angle of the swash plate

Stroke > 2 * *PitchRadius* * `tan`(*MaxAngle*),

where *PitchRadius* is the radius of the cylinder block pitch circle, and *MaxAngle* is the maximum allowed angle of the swash plate.

In the model, the maximum angle is set to 35 degrees (0.6109 rad) and pitch radius is set to 0.04 m, which makes the stroke to be greater than 0.056 m. The stroke is set 0.06 m. The piston initial positions must be equal to half of the stroke at zero initial swash plate angle. But the initial angle changes its value depending on the initial position of the actuator. As a result, the piston initial positions are computed with the equation



$$x_{p0_i} = x_{a_i} * \frac{r_{pc}}{r_a}\cos(\gamma_0), \qquad i = 1,\ldots,5$$

where

x_{p0}	piston initial position
x_a	actuator initial position
r_{pc}	pitch radius of the cylinder block
r_a	actuator radius
γ_0	phase angle

The calculation of the piston initial positions is performed in the initialization section of the subsystem mask editor.

Pressure/Flow Control Unit

The purpose of the control unit is to implement two functions: load sensing and pressure limiting. The load sensing is implemented by maintaining specified pressure differential across the flow control valve. In the model of the test rig (Figure 1), the flow control valve is simulated with the Orifice with Variable Area Slot block. Pressures upstream and downstream of the valve are conveyed to the Pressure/Flow Control unit via ports P and LSP (Figure 5).

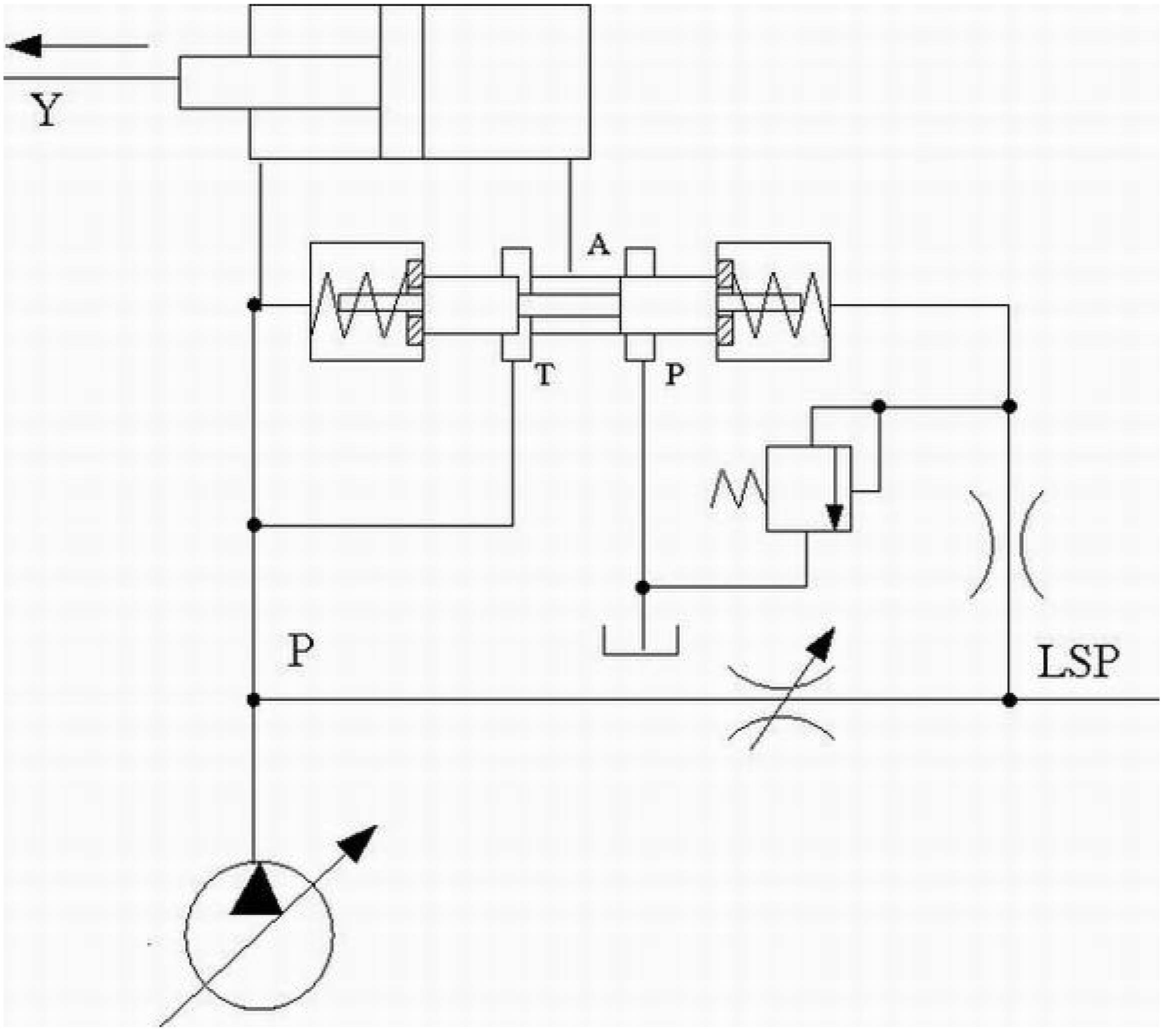


Figure 5. Pressure/Flow Control Unit Schematic

These pressures act on the side faces of the 3-way directional valve and shift the valve proportionally to the pressure difference and setting of the centering springs. The valve connections are selected in such a way that increased pressure differential opens path A-P and closes path A-T. The actuator is arranged as a single-rod differential hydraulic cylinder with rod connected to the pump yoke. The pump displacement is increased if the rod moves in the direction of the arrow shown in the schematic. Due to difference between cylinder effective areas, the displacement is increased if both cylinder chambers are connected to the pump, and decreased if the chamber without rod is connected to tank. As a result, increased pressure differential across the valve causes the pump to decrease its displacement until it returns to preset value. The spring preload of the valve is determined with the equation

$$f_0 \geq \Delta p \cdot A_{LSC}$$

where

f_0 spring preload of the load sensing valve

Δp required pressure differential across the flow control valve

The purpose of the pressure limiting function is to prevent pump pressure from exceeding the preset value. It is implemented with the pressure-relief valve and the orifice in the LSP line. The pressure-relief valve is set to the desired maximum value. When pump pressure builds up to this value, the valve opens and causes pressure in the right chamber of the valve to decrease opening path A-P. The actuator shifts to the right until the pressure returns to the preset value.

The model of a load-sensing valve is built using the 3-Way Directional Valve, Hydraulic Double-Acting Valve Actuator, Pressure-Relief Valve, and Fixed Orifice blocks, as shown in the model diagram (Figures 6 and 7).

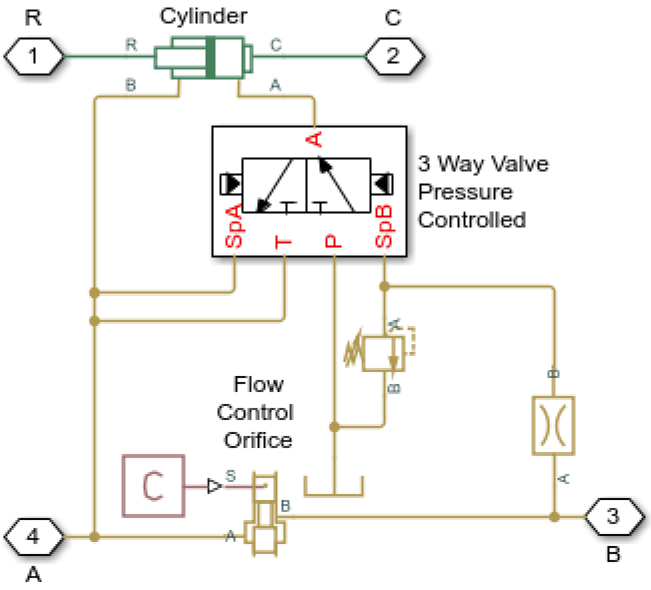


Figure 6. Pressure/Flow Control Unit Model

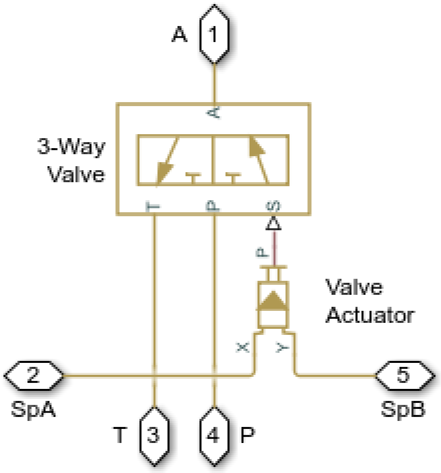


Figure 7. 3 Way Valve Pressure Controlled Model

The pressure differential is set to 20 bar. The 3-Way Directional Valve path A-T must be initially open, to force the pump to increase its displacement at the start of operation. To perform the load-sensing function, pressure increase at the B port (load-sensing port) must open the A-T path and close the A-P path. These are the reasons that determined the valve port connections to the system. The remaining load sensing control valve parameters, such as spring stiffness, valve stroke, valve orifice area, and so on, are tuned in the model to ensure required accuracy, stability, and numerical effectiveness.

The pressure-limiting function is implemented with the combination of the Fixed Orifice and Pressure-Relief Valve blocks. The valve is set to 250 bar. At this pressure, increased flow through the fixed orifice causes pressure at port Y of the Hydraulic Double-Acting Valve Actuator (block Valve Actuator in Figure 7) to drop, which eventually decreases the displacement of the pump.

Cycle Description

The simulated cycle consists of six elements characterized by different load conditions with the Variable Area Slot block.

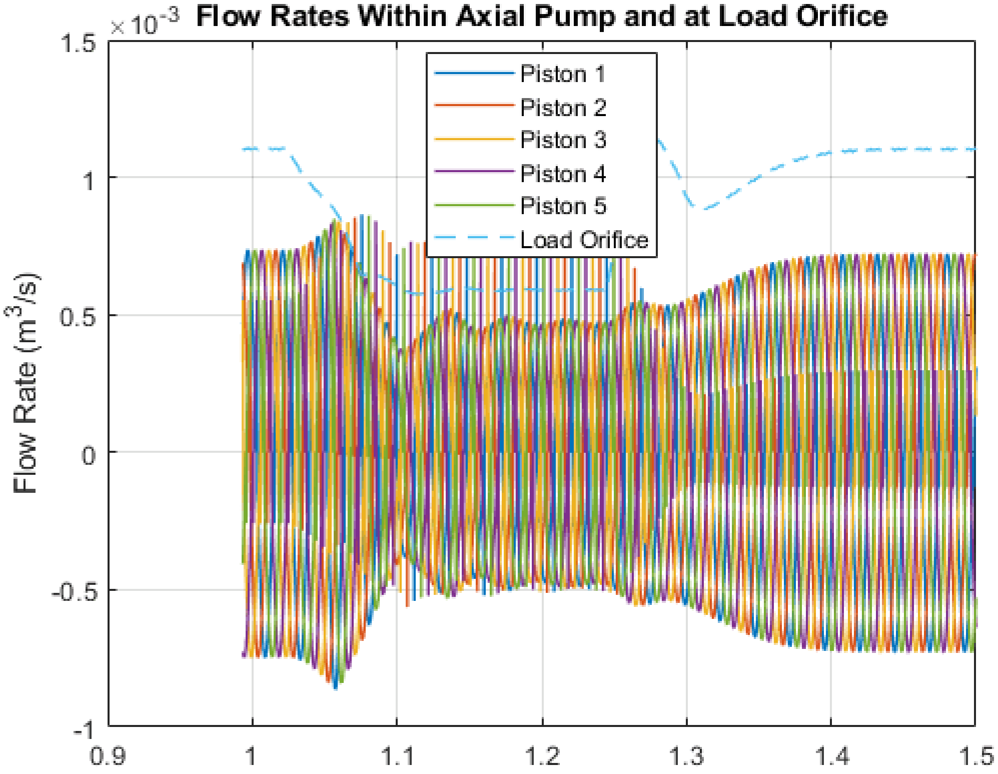
The cycle starts with zero opening signal, followed by opening of 2.8, 5.2, 1, -0.8, and, finally, 2.45 mm. At the start of the cycle, the pump shaft starts rotating at 260 rad/s (~2500 RPM) with the pump yoke initial position set to 5 mm. The servo cylinder starts increasing pump displacement, pump pressure slowly builds up, and the process settles down at ~0.35 s after the pressure differential across the flow control valve becomes close to preset value of 20 bar. The load-sensing valve is opened at this moment by ~1.2 mm.

During next three portions of the cycle, the pump maintains practically the same delivery despite changes of the load valve opening.

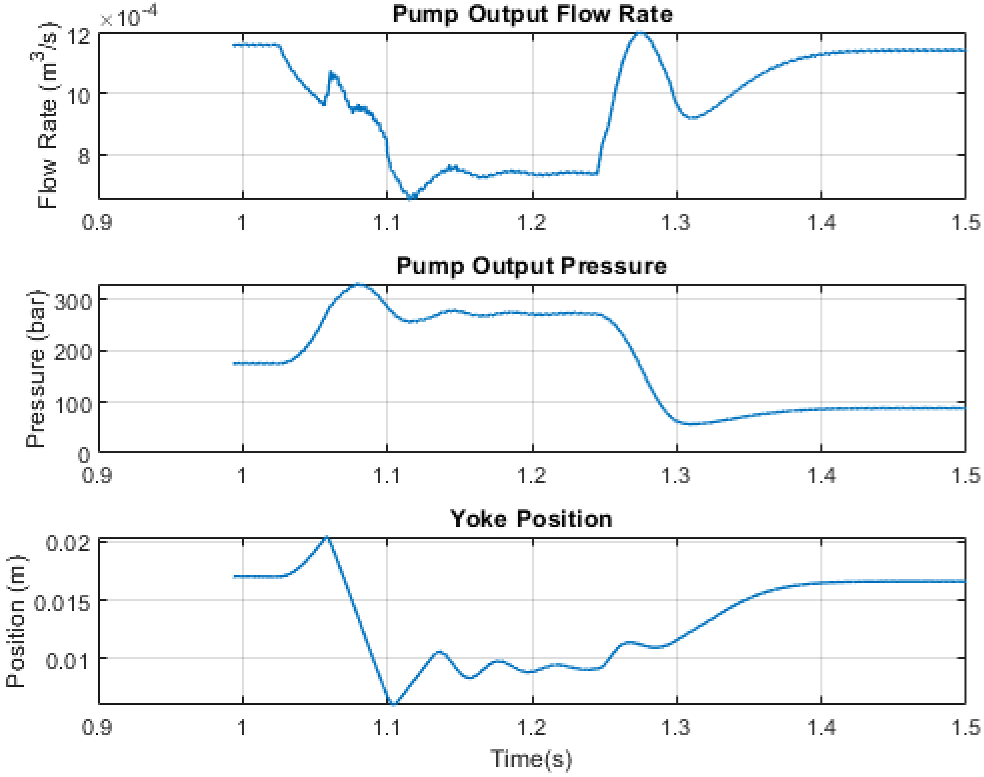
At 1 s, the load valve is practically fully closed, causing pump pressure to rise. The load-limiting function becomes dominant as the pressure reaches 270 bar. The pump returns to the load-sensing mode after the pressure falls below the preset value.

Simulation Results from Simscape Logging

The plot below shows the flow rate within the pump pistons and at the load. The cyclic nature of the piston pressures can be seen, as well as the overall behavior of the pump which stays close to its rated flow rate.



The plots show the load sensing and pressure limiting control. The pump maintains its rated flow rate of 1.1 m^3/s even as the load changes, as shown on the pump output pressure plot. However, as the pump output pressure rises to its maximum rated pressure, the pressure limiting control adjusts the yoke position is and the flow rate drops below its rated flow rate.

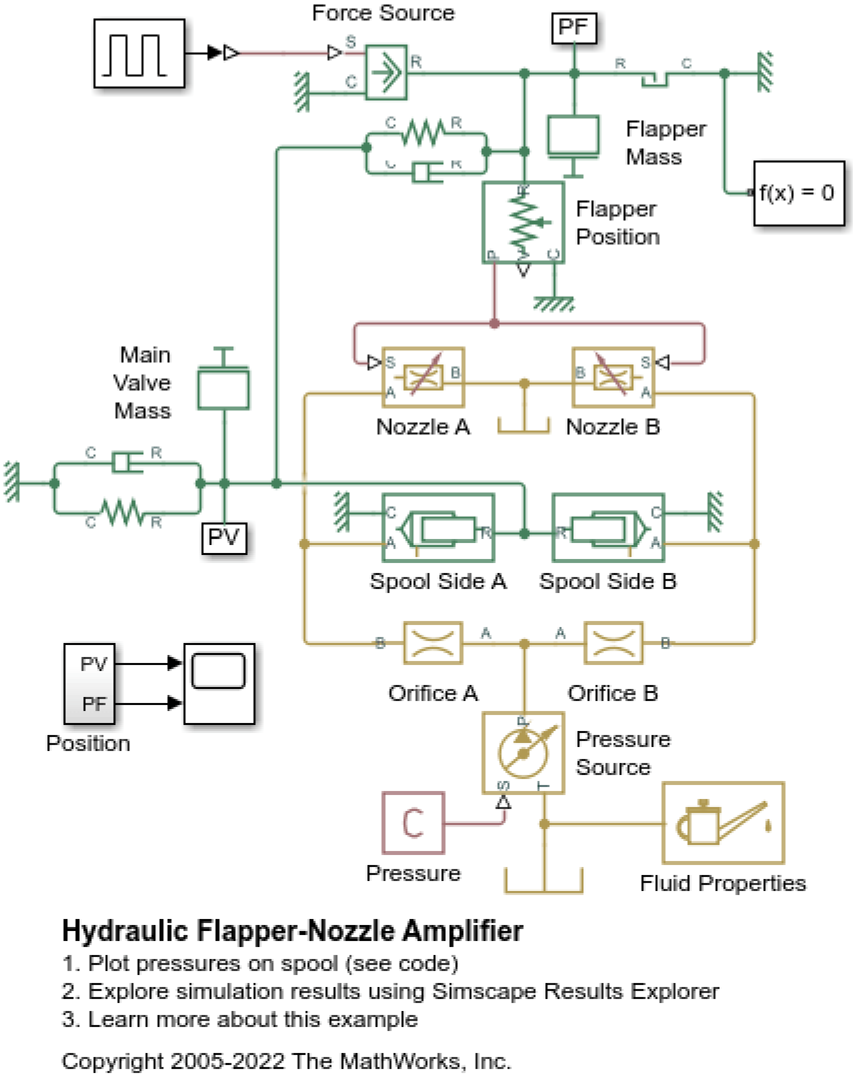

Pump Output Flow Rate
×10^-4
Flow Rate (m^3/s)
12
10
8
0.9
1
1.1
1.2
1.3
1.4
1.5
Pump Output Pressure
Pressure (bar)
300
200
100
0
0.9
1
1.1
1.2
1.3
1.4
1.5
Yoke Position
Position (m)
0.02
0.015
0.01
0.9
1
1.1
1.2
1.3
1.4
1.5
Time(s)

Hydraulic Flapper-Nozzle Amplifier

This example shows an amplifier that consists of two fixed orifices, two variable orifices representing nozzles, flapper, and main valve simulated with mass, viscous friction, and centering spring. Two translational converters represent servo-actuators on both sides of the main valve. A feedback spring connects the flapper and the main valve.

The amplifier is actuated with a force source, which represents the force generated by an electromagnet. The flapper displacement causes pressure difference between converter chambers, which moves the spool. The main valve acts on the flapper in the opposite direction via a feedback spring. The amplifier comes to a steady-state position as the force equilibrium on the flapper is reached.

Model



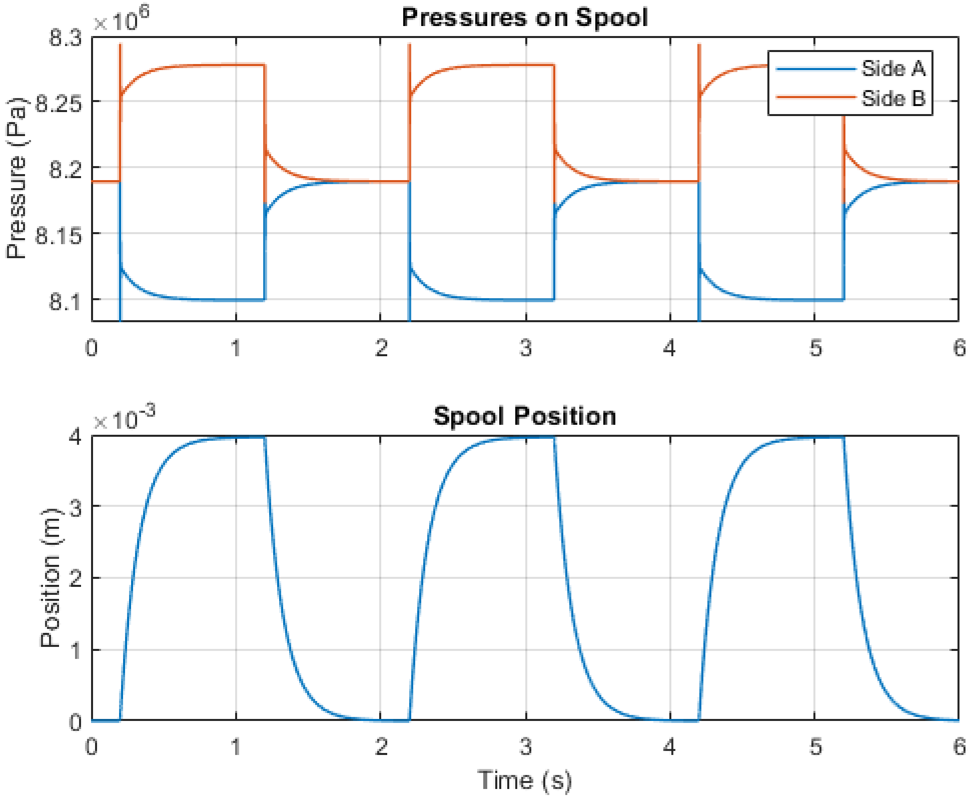
Hydraulic Flapper-Nozzle Amplifier

1. Plot pressures on spool (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2005-2022 The MathWorks, Inc.

Simulation Results from Simscape Logging

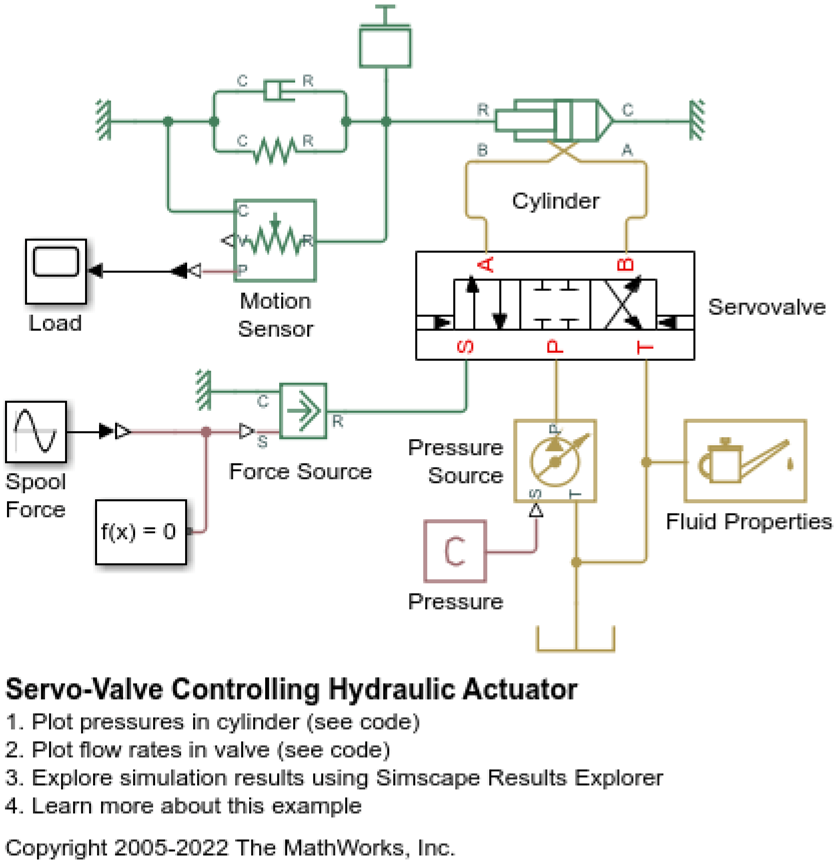
The plots below show the pressures acting on the spool and the position of the spool.



Servo-Valve Controlling Hydraulic Actuator

This example shows a model of a two-stage servo-valve with a 4-way, 3-position spool valve in the power stage and a flapper-nozzle amplifier in the pilot stage. The flapper is connected to the armature of a torque motor, which in the example is represented with an ideal translational force source. The servo-valve shown in the example is equipped with the spring feedback between the flapper and the spool of the main valve. To investigate the behavior of such a valve, axial hydraulic forces on all four spool orifices are accounted for in the model by using Spool Orifice Hydraulic Force blocks. The servo-valve controls a simple double-acting cylinder in an open-loop application.

Model

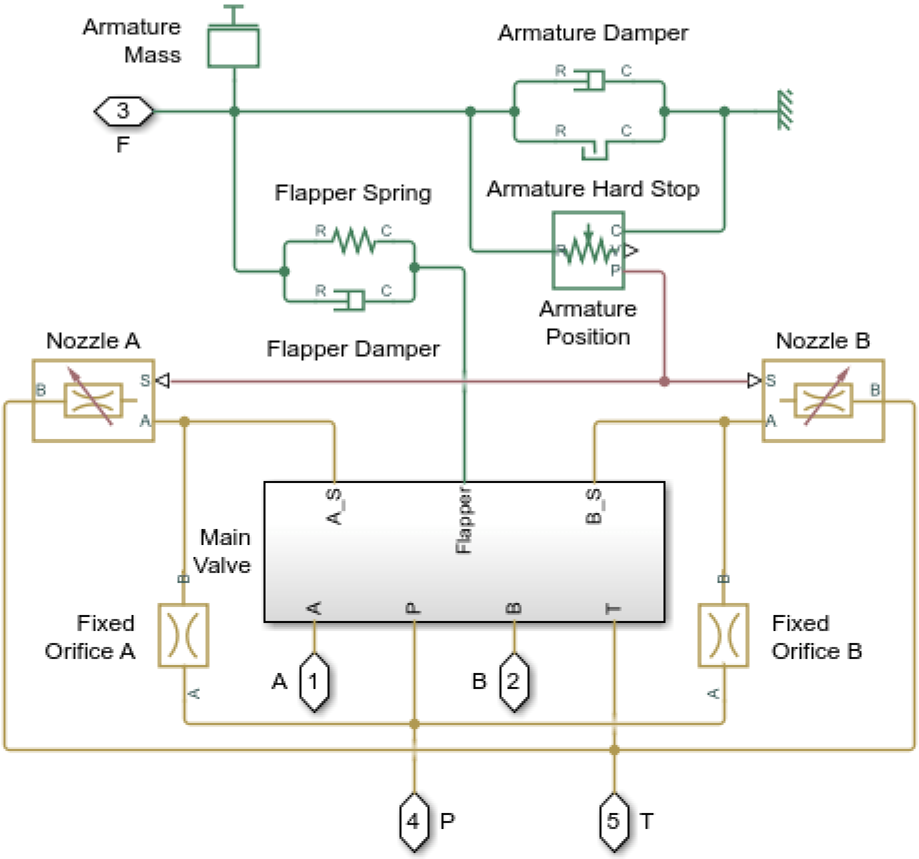


Servo-Valve Controlling Hydraulic Actuator

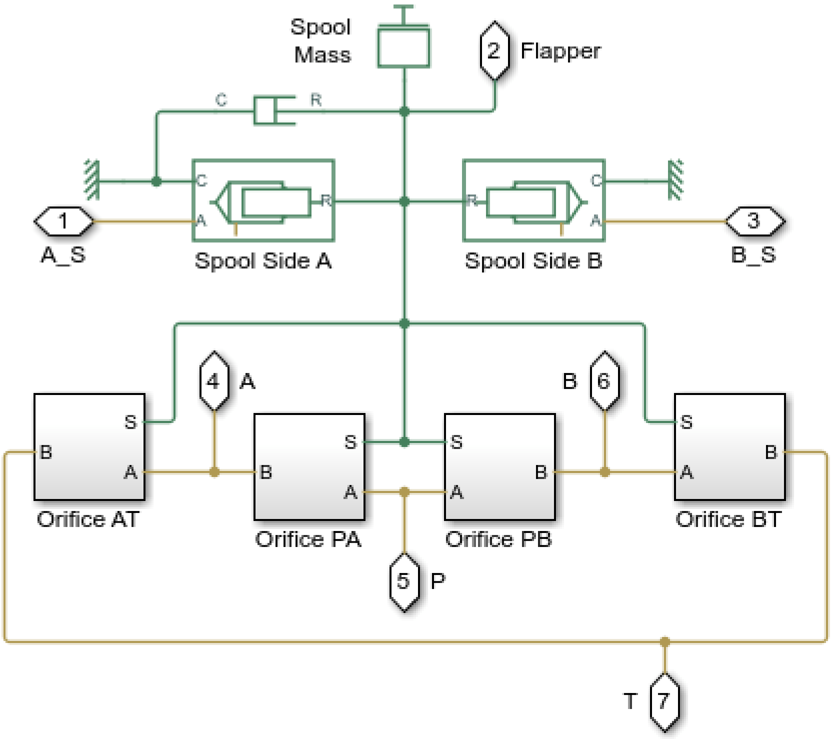
1. Plot pressures in cylinder (see code)
2. Plot flow rates in valve (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2005-2022 The MathWorks, Inc.

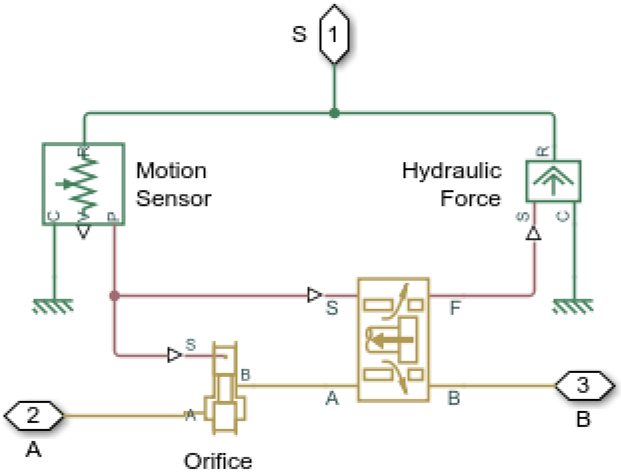
Servovalve Subsystem



Main Valve Subsystem

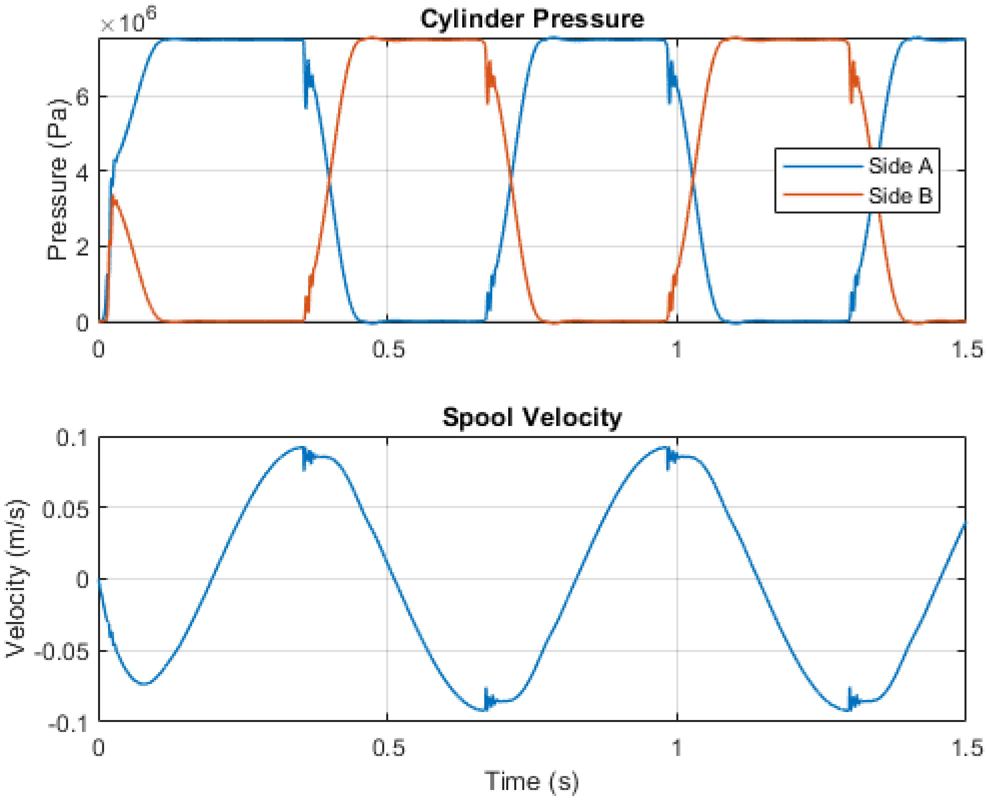


Orifice AT Subsystem

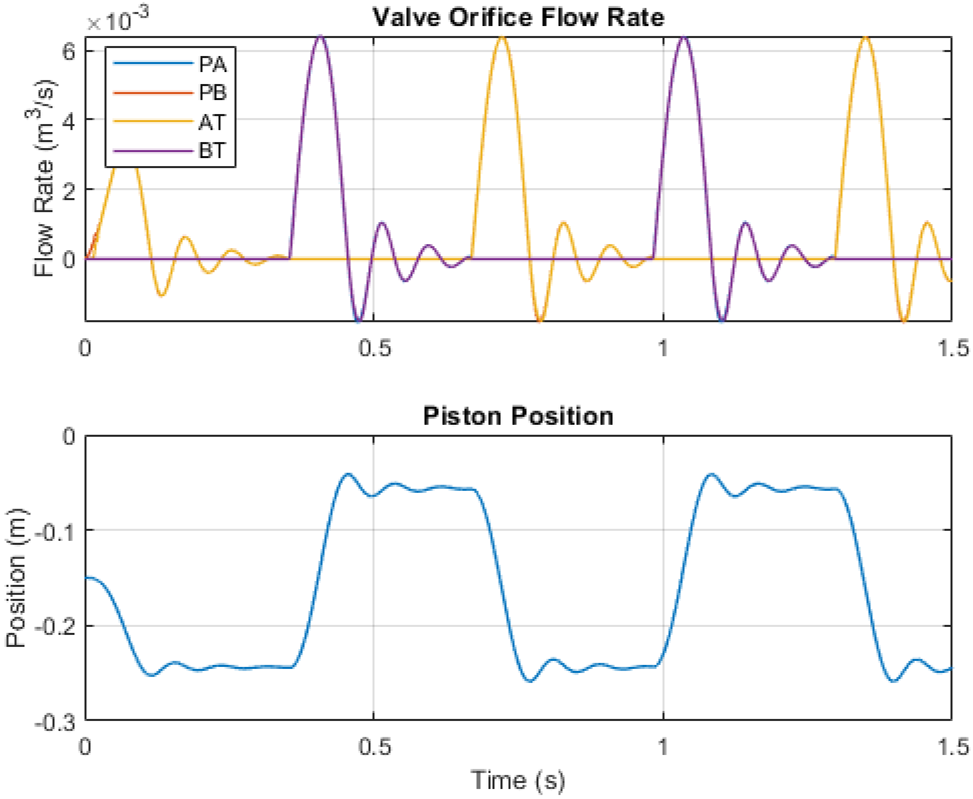


Simulation Results from Simscape Logging

The plots below show the pressures in the cylinder and the velocity of the spool in the pilot stage. The effect of the pilot stage spool oscillations are visible in the cylinder pressures.



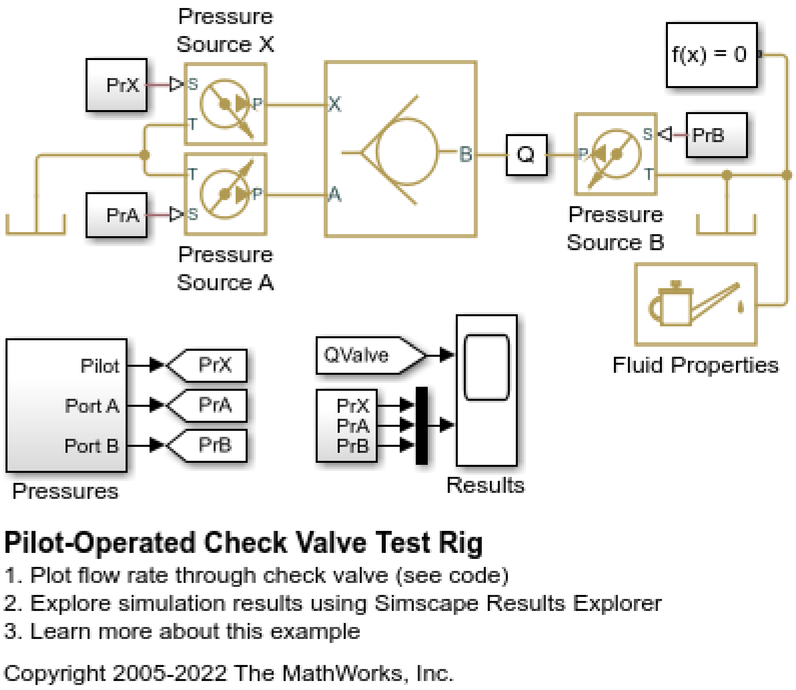
The plots below show the flow rates in the valve and the position of the piston.


Valve Orifice Flow Rate
×10^-3
Flow Rate (m^3/s)
6
4
2
0
0
0.5
1
1.5
PA
PB
AT
BT
Piston Position
Position (m)
0
-0.1
-0.2
-0.3
0
0.5
1
1.5
Time (s)

Pilot-Operated Check Valve Test Rig

This example shows a test circuit built to check the characteristics of a pilot-operated check valve. The valve is loaded with three ideal pressure sources, two of which create pressure differential across the main flow line, while the third applies pressure to the pilot inlet X. The pilot pressure allows flow through the valve even if the main pressure differential is negative.

Model

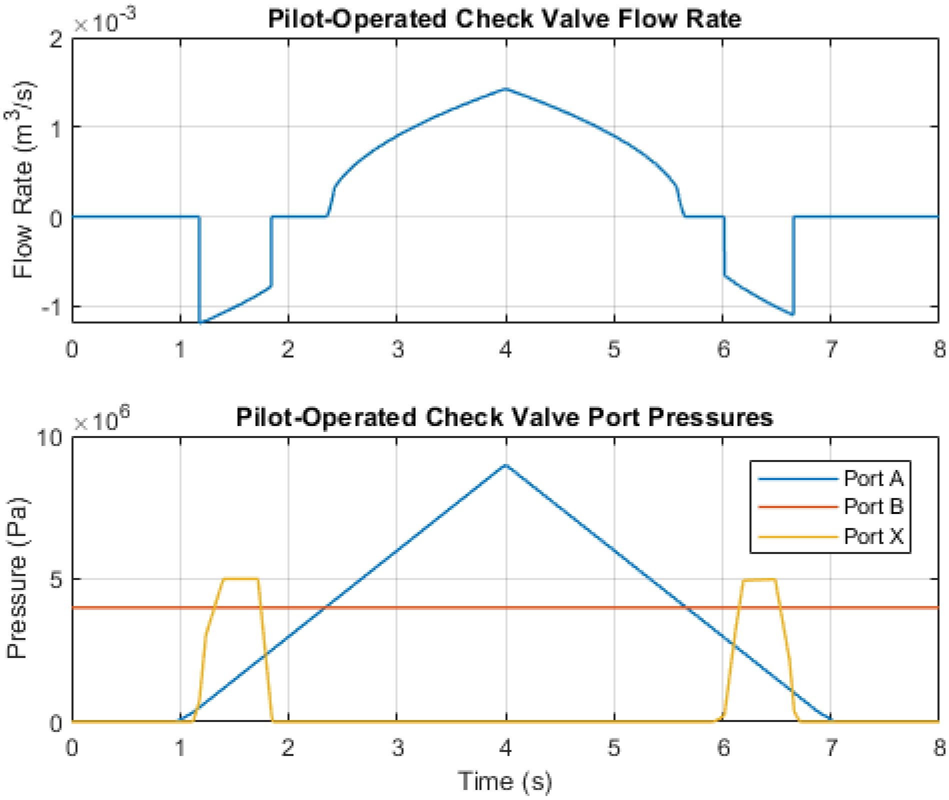


Pilot-Operated Check Valve Test Rig

1. Plot flow rate through check valve (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Simulation Results from Simscape Logging

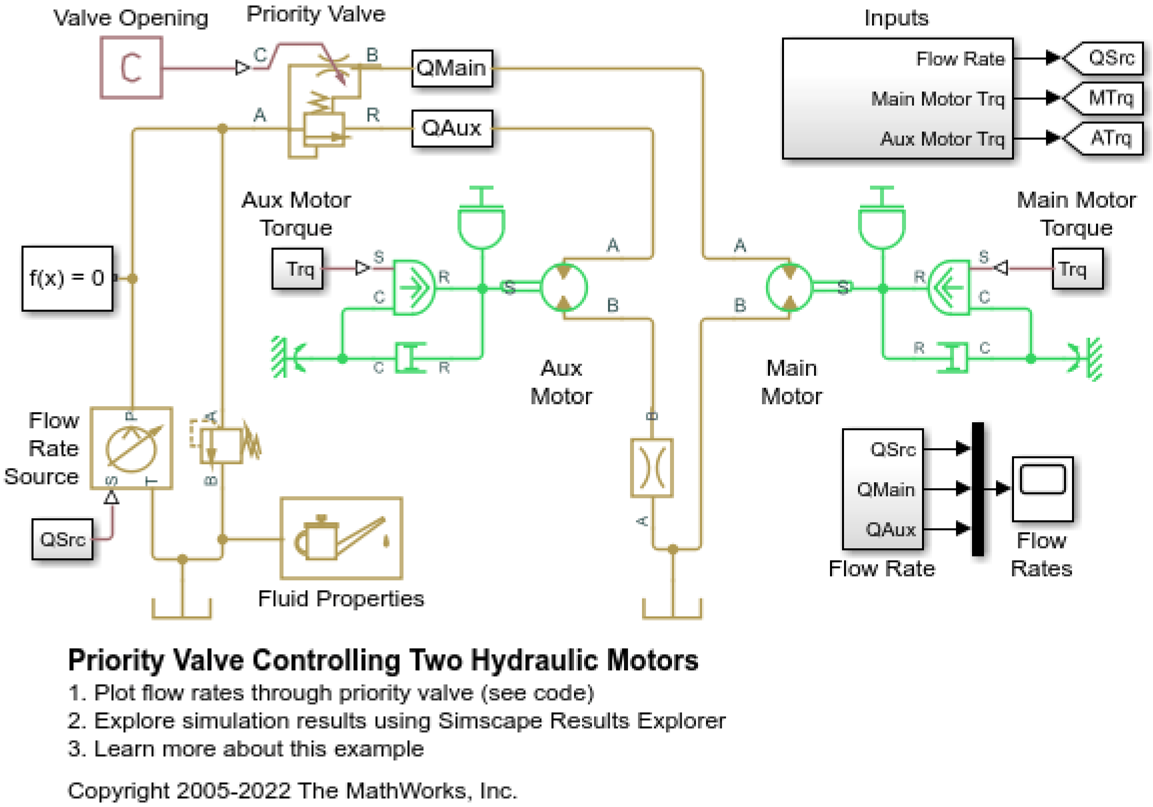
The plots below show the behavior of the Pilot-Operated Check Valve block under varying conditions. Flow is permitted from port A to port B (positive flow rate) when the pressure differential exceeds the cracking pressure. Flow is permitted from port B to port A (negative flow rate) when the pressure at port X is high enough to overcome the pressure differential between port B and port A.


Pilot-Operated Check Valve Flow Rate
×10-3
2
1
0
-1
Flow Rate (m3/s)
0
1
2
3
4
5
6
7
8
Pilot-Operated Check Valve Port Pressures
×106
10
5
0
Pressure (Pa)
Port A
Port B
Port X
0
1
2
3
4
5
6
7
8
Time (s)

Priority Valve Controlling Two Hydraulic Motors

This example shows a pressure-compensated 3-way flow control valve. This valve maintains constant flow rate through the main hydraulic motor, which is connected to the pressure-compensated outlet of the flow control valve. It acts as a priority valve, diverting the excess flow to the auxiliary hydraulic motor if the main hydraulic motor receives enough fluid to maintain a preset angular velocity. The auxiliary motor is shut off completely if there is insufficient flow to power the main hydraulic motor.

Model



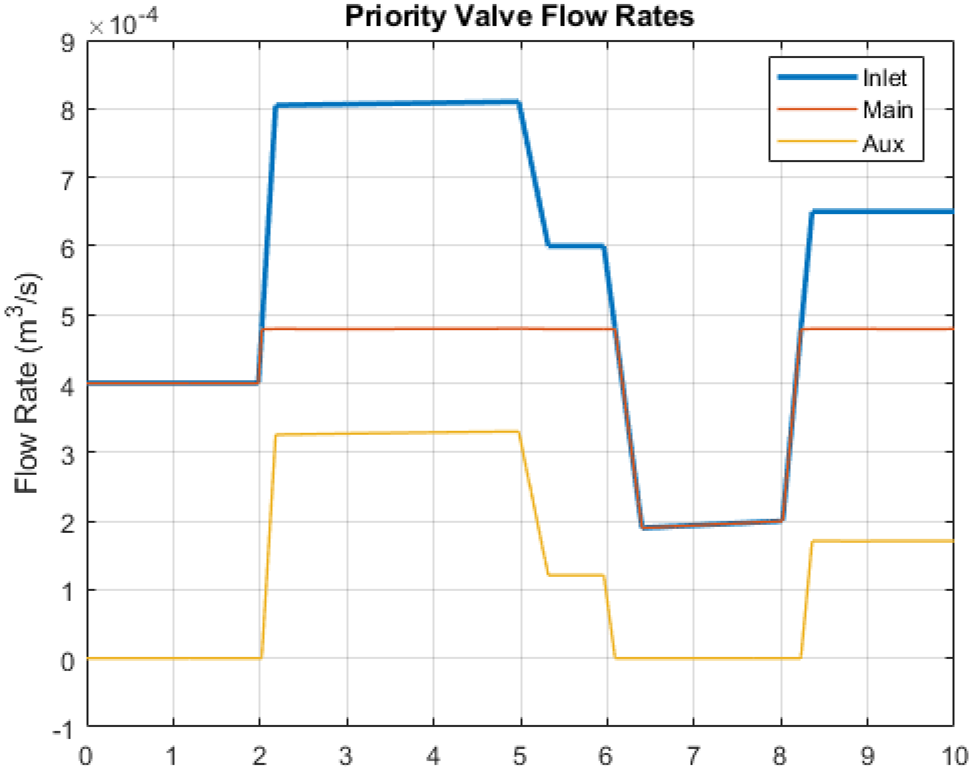
Priority Valve Controlling Two Hydraulic Motors

1. Plot flow rates through priority valve (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2005-2022 The MathWorks, Inc.

Simulation Results from Simscape Logging

The plot below shows the flow rates through the priority valve. As the flow rate at the inlet varies, the main motor is driven at the desired speed and excess flow is directed to the auxiliary hydraulic motor. When the inlet flow drops below the threshold to maintain the speed for the main hydraulic motor, the main motor slows down as well.

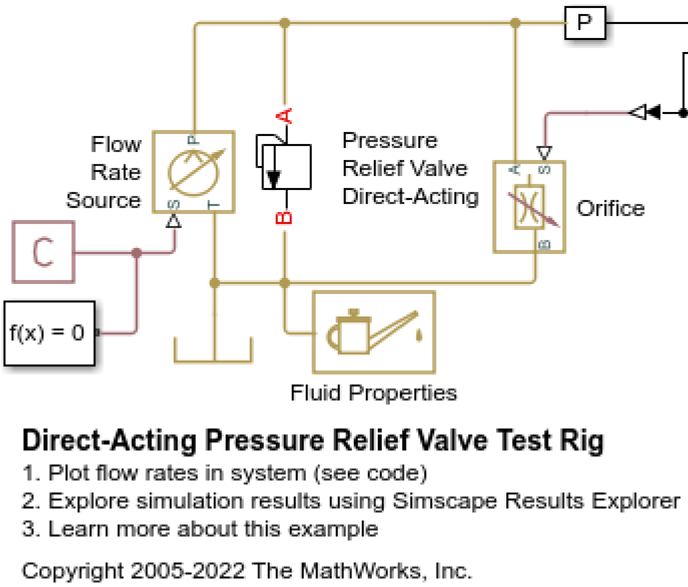

Priority Valve Flow Rates
×10^-4
Flow Rate (m^3/s)
9
8
7
6
5
4
3
2
1
0
-1
0
1
2
3
4
5
6
7
8
9
10
Inlet
Main
Aux

Direct-Acting Pressure Relief Valve Test Rig

This example shows a test circuit built to check the pressure-flow characteristic of a direct-acting pressure-relief valve. The valve model consists of an orifice, a hydro-mechanical converter, a preloaded spring, and a hard stop.

An ideal flow rate source delivers fluid to a system through a variable orifice. As the orifice closes, pressure gradually builds up and eventually reaches the setting of the pressure-relief valve. At this pressure, the valve starts diverting flow to a tank and maintains preset pressure at the pump outlet. As the orifice is opened again, the pressure-relief valve is closed and the entire flow passes through the variable orifice.

Model

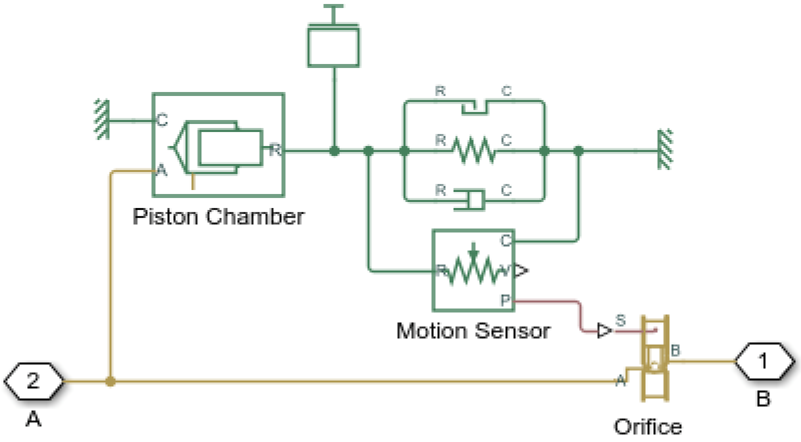


Direct-Acting Pressure Relief Valve Test Rig

1. Plot flow rates in system (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

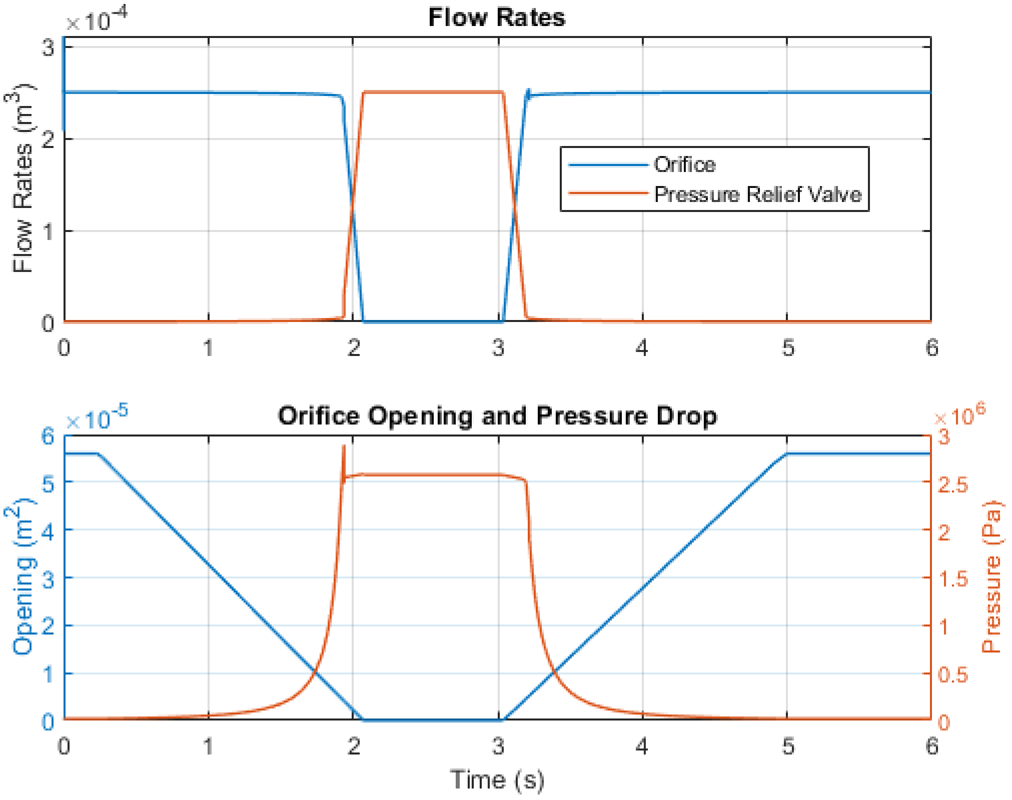
Copyright 2005-2022 The MathWorks, Inc.

Pressure Relief Valve Direct-Acting Subsystem



Simulation Results from Simscape Logging

The plots below show the behavior of the direct-acting pressure relief valve. As the load orifice closes, pressure in the system rises. When the system pressure reaches the pressure setting for the pressure relief valve, the valve opens which diverts flow away from the orifice and allows the system to maintain the set pressure of the pressure relief valve.

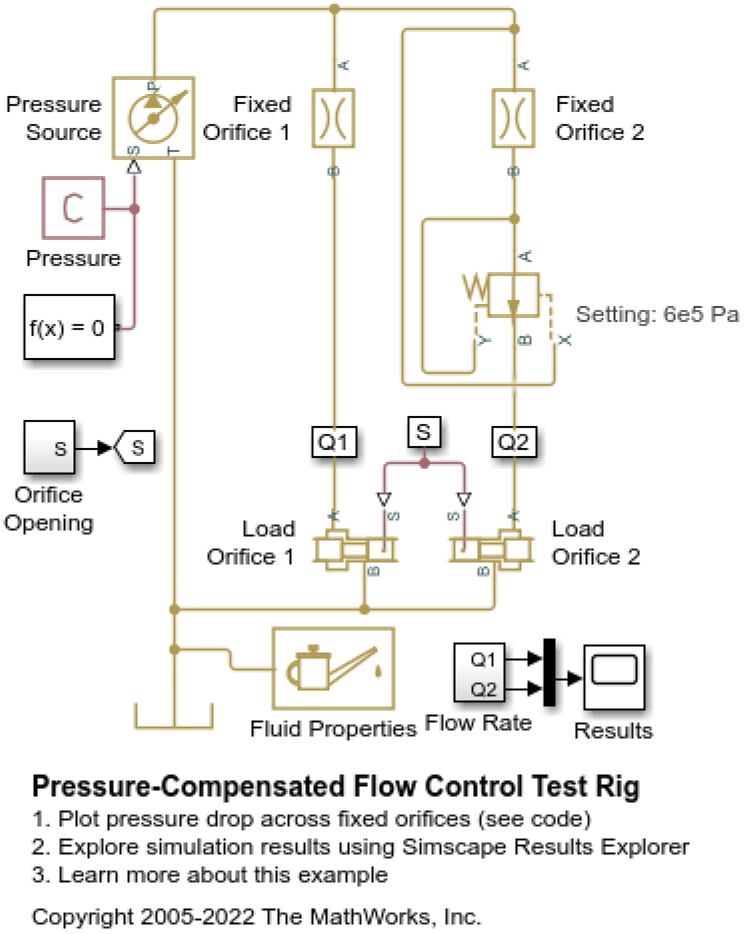


Pressure-Compensated Flow Control Test Rig

This example shows a test circuit built to check the characteristics of a pressure compensator. A pressure source is connected to two flow paths, each with a fixed orifice and a variable orifice that acts as a variable load. Along one flow path a pressure compensator is installed.

The pressure compensator is set to maintain a 6 bar pressure differential across the orifice. As a result, the flow rate along that flow path remains nearly constant as long as the pressure differential across the flow control is bigger than 6 bar. This is not the case for the flow path with no pressure compensator where the flow rate varies with the load.

Model

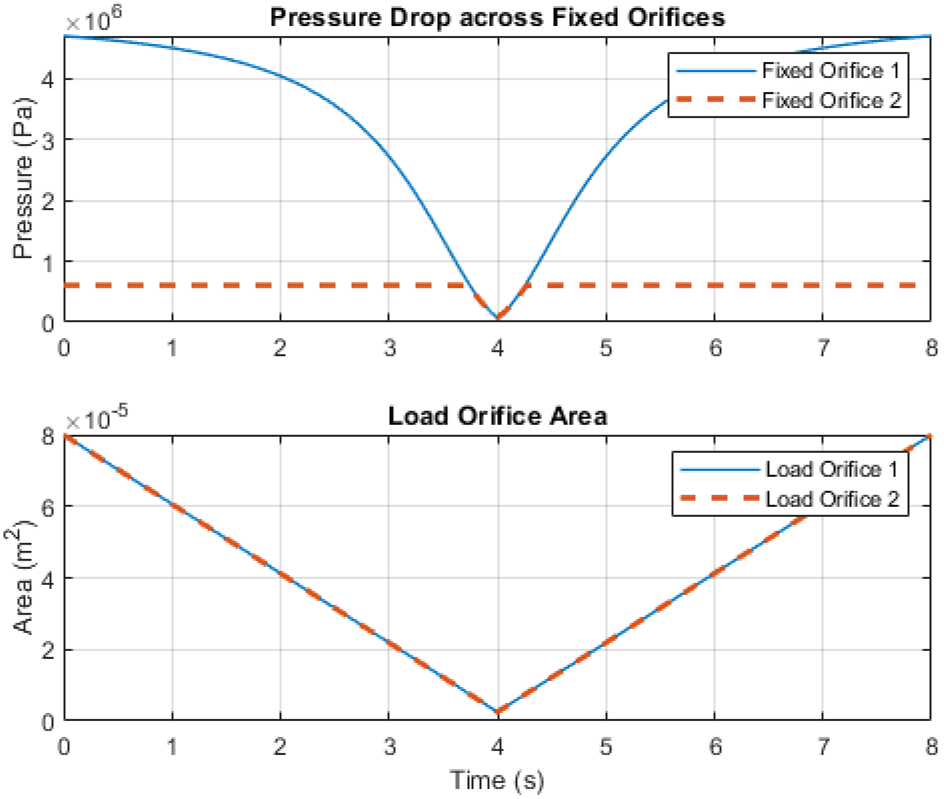


Pressure-Compensated Flow Control Test Rig

1. Plot pressure drop across fixed orifices (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Simulation Results from Simscape Logging

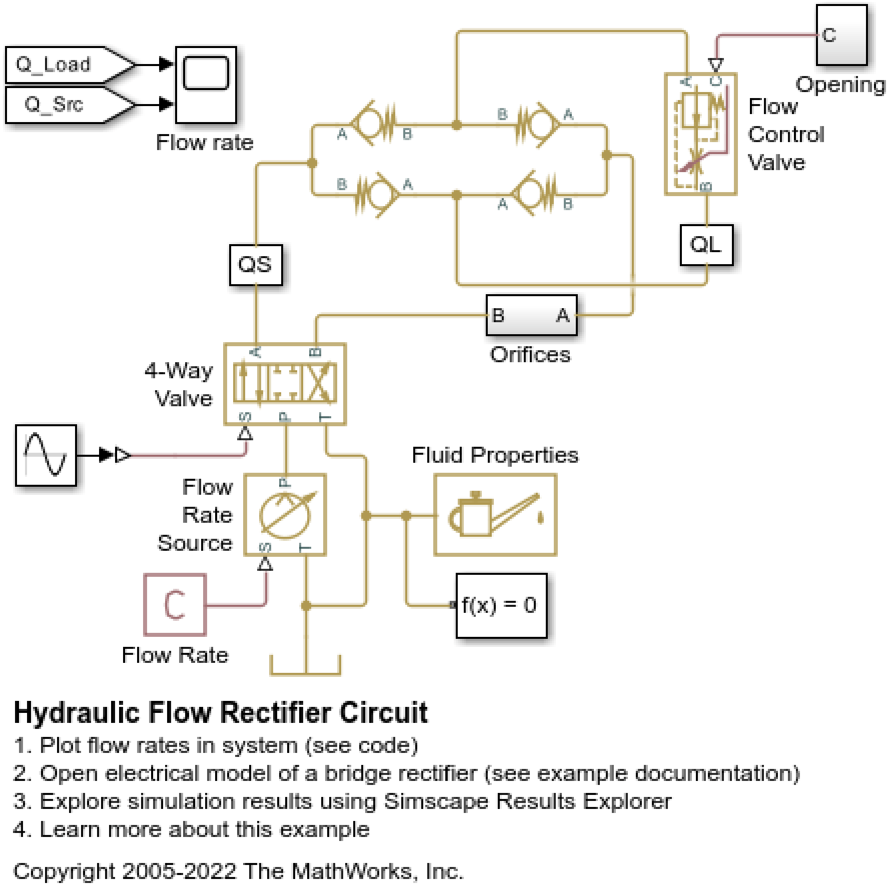
This plot shows the behavior of the pressure compensator, which controls the pressure drop across Fixed Orifice 2. As the downstream loads (variable orifices) are adjusted, the pressure drop across Fixed Orifice 2 stays constant at the set level in the pressure compensator. When the pressure drop across the loads gets too high, the pressure drop across Fixed Orifice 2 then falls below the set level in the pressure compensator.


Pressure Drop across Fixed Orifices
×10^6
Pressure (Pa)
0
1
2
3
4
0 1 2 3 4 5 6 7 8
Fixed Orifice 1
Fixed Orifice 2
Load Orifice Area
×10^-5
Area (m^2)
0
2
4
6
8
0 1 2 3 4 5 6 7 8
Time (s)
Load Orifice 1
Load Orifice 2

Hydraulic Flow Rectifier Circuit

This example shows a flow rectifier circuit, which consists of four check valves and a flow control valve. It is used to control flow rate flowing in both directions with only one flow control valve. Similar to a Graetz circuit implemented with diodes, the check valves are arranged in such a way that flow always passes through the flow control valve in the same direction. Two other check valves, Flow BA Check Valve and Flow AB Check Valve, are used to select an orifice depending on the flow direction.

Model



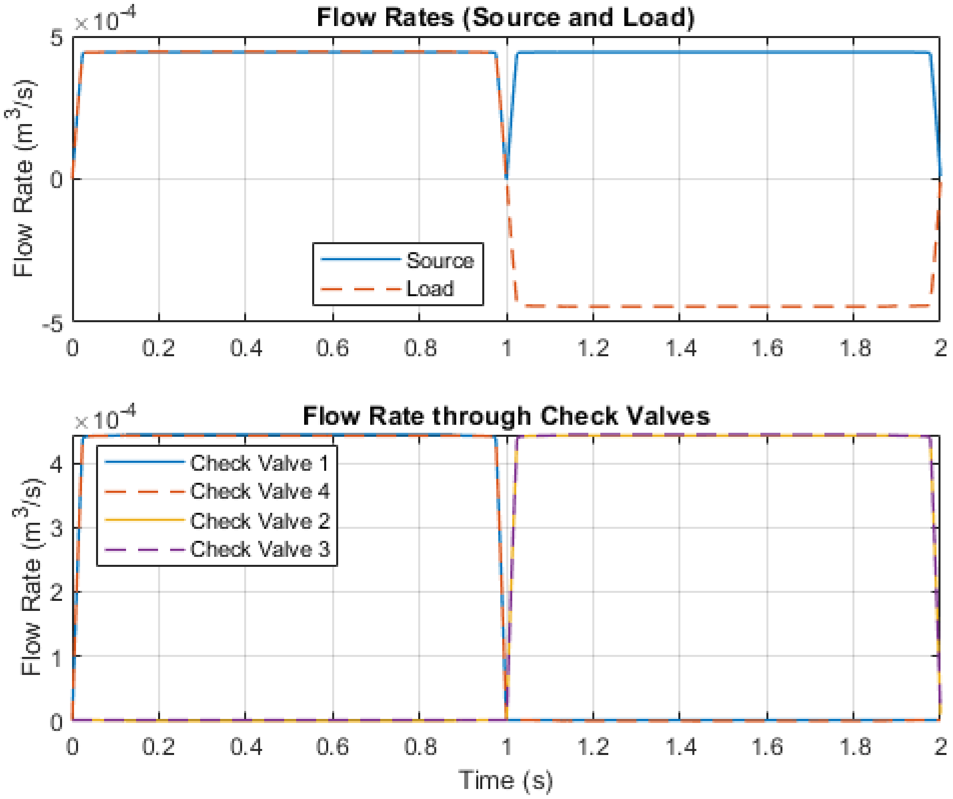
Hydraulic Flow Rectifier Circuit

1. Plot flow rates in system (see code)
2. Open electrical model of a bridge rectifier (see example documentation)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2005-2022 The MathWorks, Inc.

Simulation Results from Simscape Logging

The plots below show the flow rate through various components in the system. Though the flow rate from the source (controlled by the 4-Way Valve) changes direction, the flow rate in the load (Flow Control Valve) remains in the same direction. The flow rates through the check valves are plotted to show how the flow proceeds through the flow rectifier.


Flow Rates (Source and Load)
×10^-4
5
0
-5
Flow Rate (m^3/s)
Source
Load
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Flow Rate through Check Valves
×10^-4
4
3
2
1
0
Flow Rate (m^3/s)
Check Valve 1
Check Valve 4
Check Valve 2
Check Valve 3
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time (s)

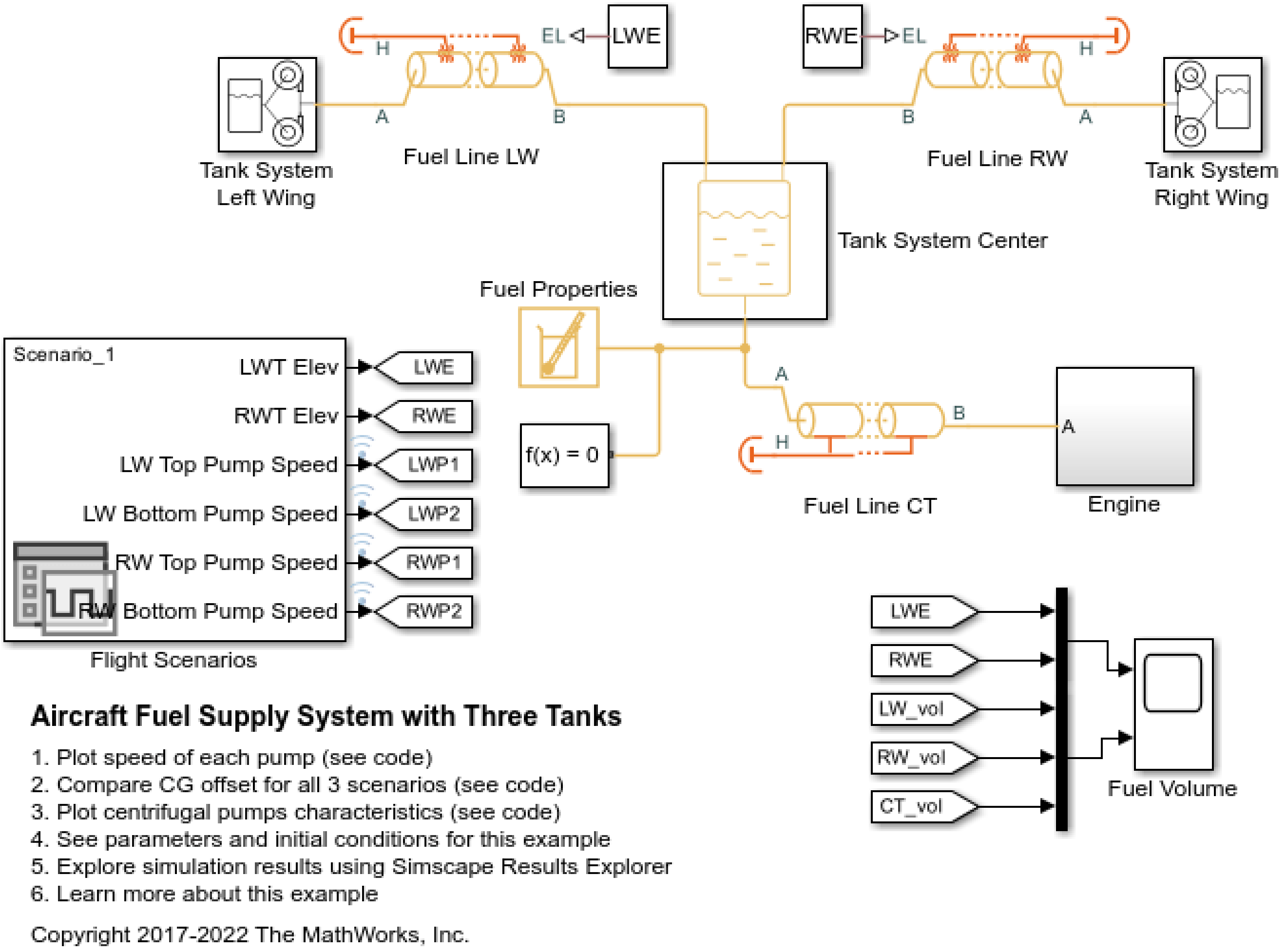
Aircraft Fuel Supply System with Three Tanks

The fuel supply system represented in the example consists of three tanks and an engine. The tanks are connected by the fuel lines and the pressure drop in the fuel lines depends on the bank angle of the aircraft or the relative elevation of the ends of the fuel lines. The central tank bottom is elevated by 36 in with respect to the reference plane, which is drawn horizontally through the entry point of the engine. The side tank bottoms are elevated by 4.2 in each at zero bank angle.

The engine is fed from the central tank, while fuel from the left wing tank and the right wing tank is pumped to the central tank with respective pumping stations. Each pumping station consists of two centrifugal pumps, connected in parallel, with check valves installed in the pump outlets to prevent back flow. The pumps are driven by prime movers at angular velocity of 120 rev/s. The movers are simulated with the ideal torque source. The fuel lines with varying elevation are represented by the Pipe Variable Elevation (TL) block.

In the left wing, right wing, and central tanks, the controllers cut the outlet off if fluid volume in the tank becomes less than preset volume. In the central tank, the controllers also prevent the tank from overflowing. In all tanks, the cutoff is performed with the 2-way valves. The model can be simulated under different flight scenarios including different pump failure.

Model

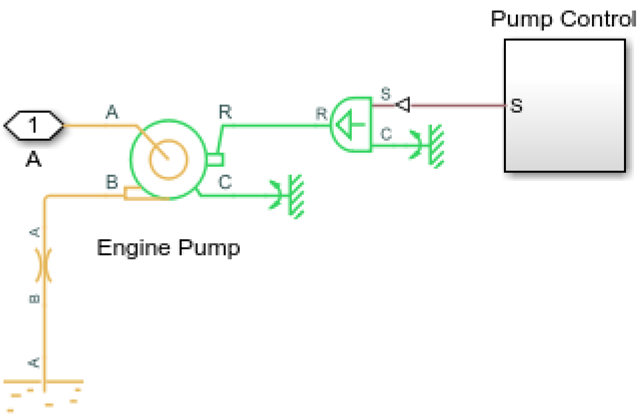


Aircraft Fuel Supply System with Three Tanks

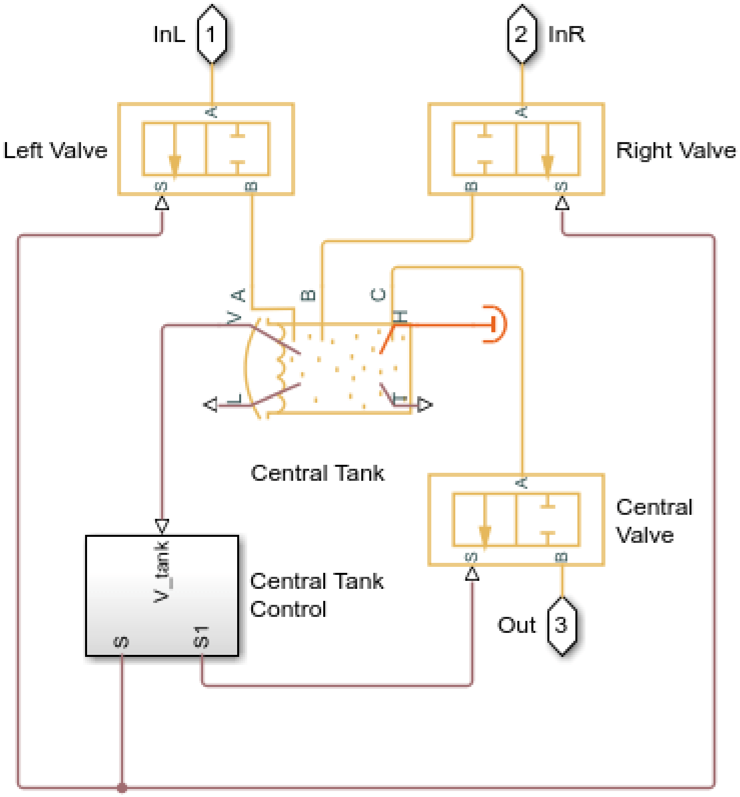
1. Plot speed of each pump (see code)
2. Compare CG offset for all 3 scenarios (see code)
3. Plot centrifugal pumps characteristics (see code)
4. See parameters and initial conditions for this example
5. Explore simulation results using Simscape Results Explorer
6. Learn more about this example

Copyright 2017-2022 The MathWorks, Inc.

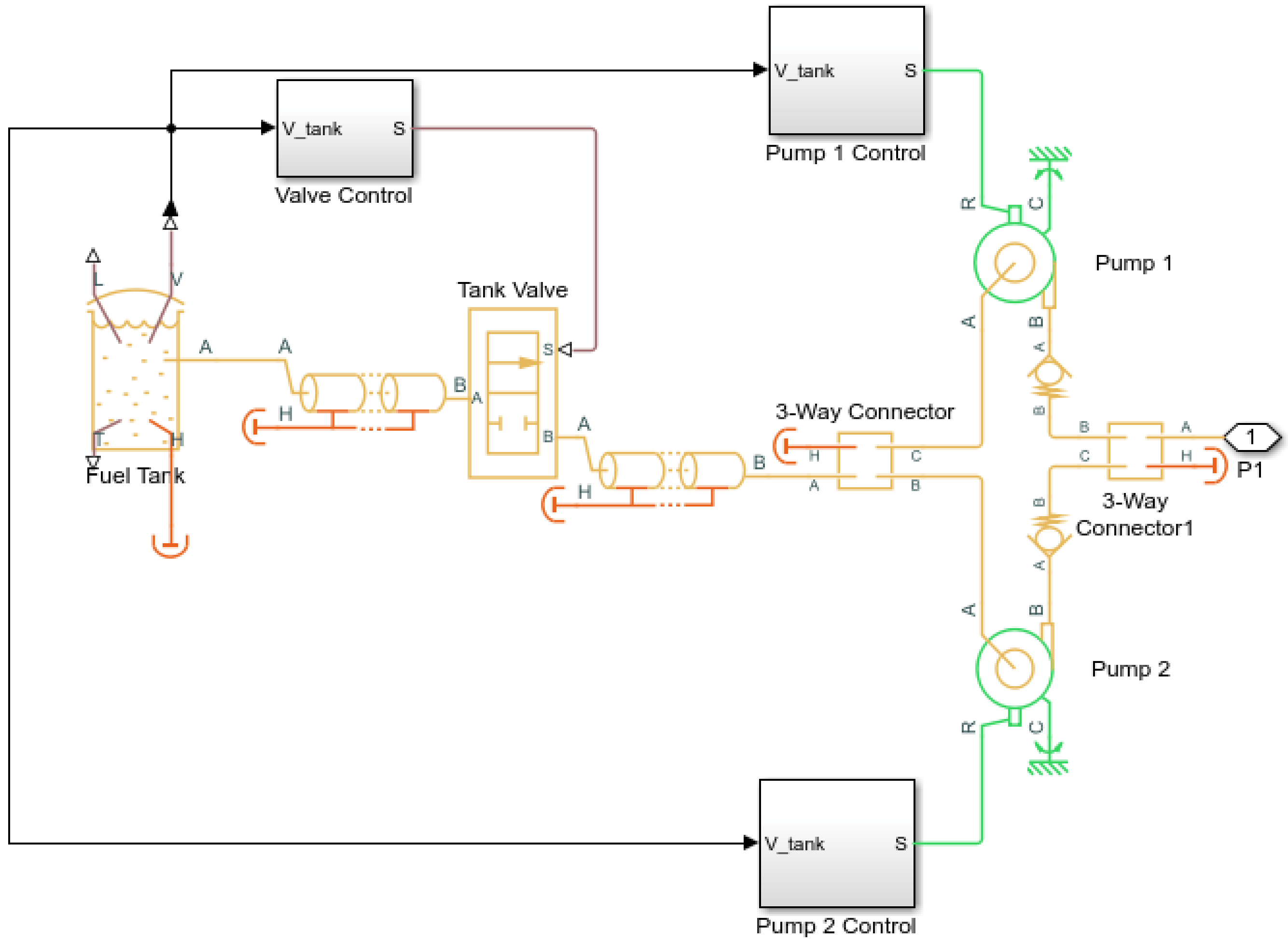
Engine Subsystem



Tank System Center Subsystem

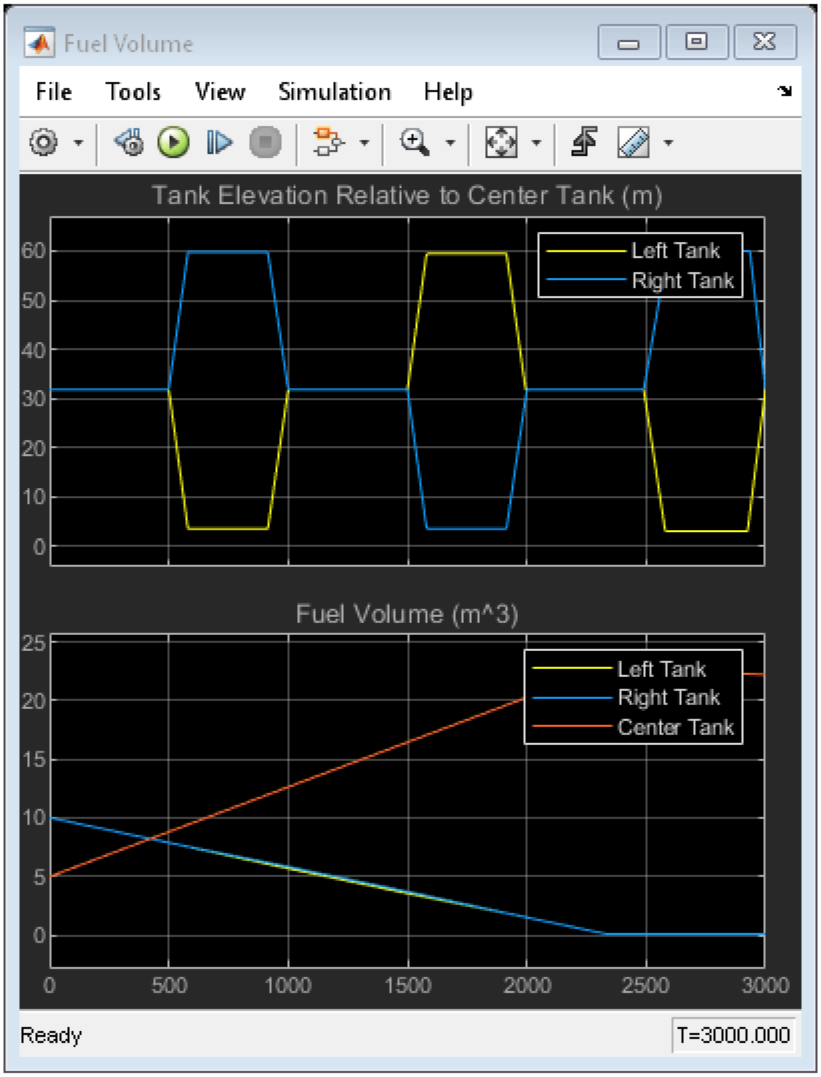


Tank System Left Wing Subsystem



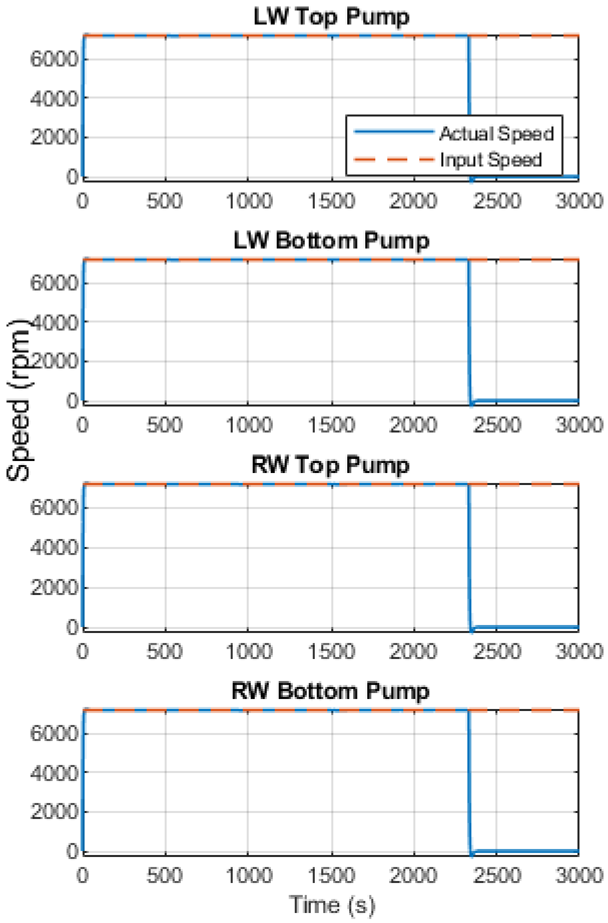
Simulation Results from Scopes

The plots below show the amount of fuel remaining in the tanks. The rate of fuel consumption from each tank is affected by the pressure drop across the fuel lines due to the elevation change relative to the center tank. Since the pump flow rate controllers keep flow rates independent of the aircraft bank angle, the effects of elevation on the tank flow rate will be small.

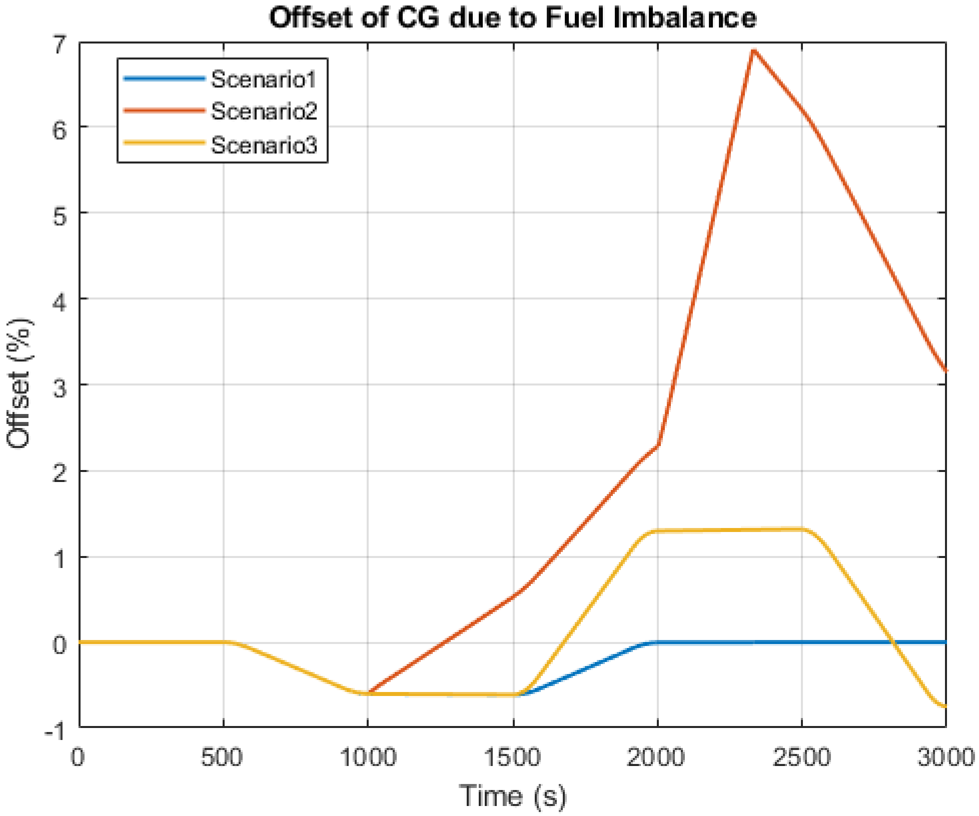


Simulation Results from Simscape Logging

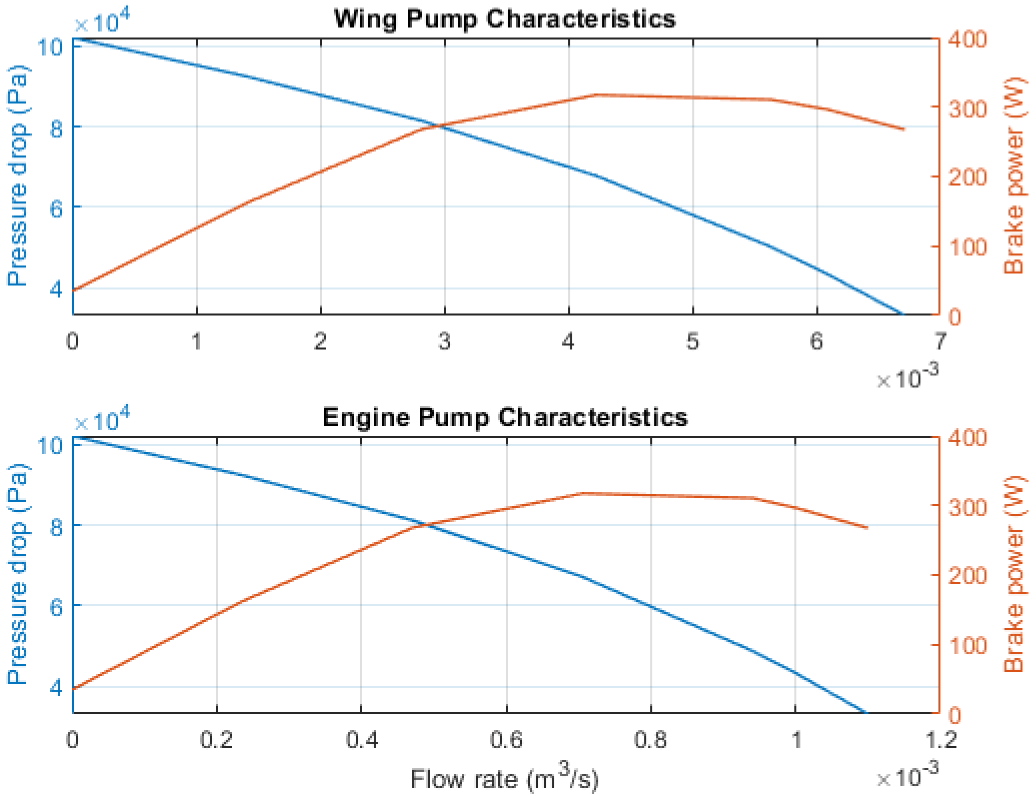
The plots below show the speed for all four pumps in the fuel supply system. For Flight Scenarios 2 and 3, pump speeds drop below the maximum speed to observe the effect on the fuel levels in the tanks.



The plot below shows the CG offset due to fuel distribution as a percentage of the half distance between left and right fuel tanks. The plots show the results for three tests where different combinations of pump failures are triggered.



The plots below show the characteristic curves of centrifugal pumps at reference condition in the fuel supply and engine systems.


Wing Pump Characteristics
×10^4
10
8
6
4
Pressure drop (Pa)
400
300
200
100
0
Brake power (W)
0
1
2
3
4
5
6
7
×10^-3
Engine Pump Characteristics
×10^4
10
8
6
4
Pressure drop (Pa)
400
300
200
100
0
Brake power (W)
0
0.2
0.4
0.6
0.8
1
1.2
×10^-3
Flow rate (m^3/s)

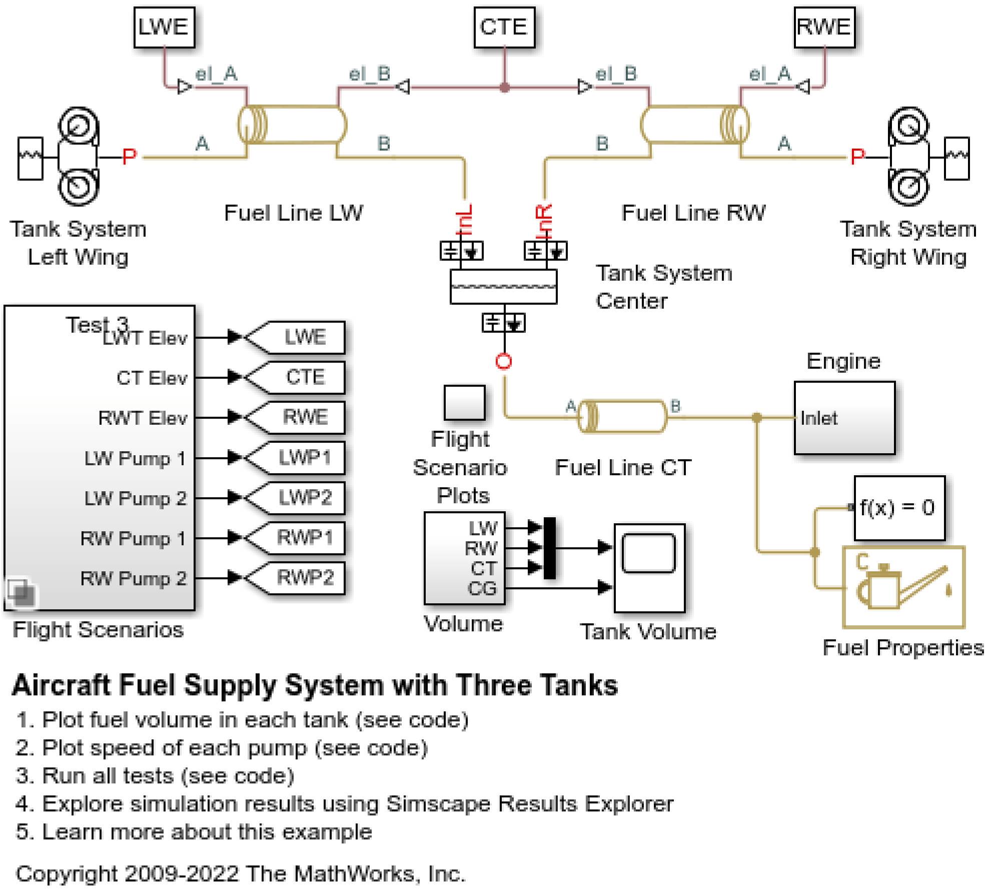
Aircraft Fuel Supply System with Three Tanks

The fuel supply system represented in the example consists of three tanks and an engine. The engine is fed from the central tank, while fuel from the left wing tank and the right wing tank is pumped to the central tank with respective pumping stations. Each pumping station consists of two centrifugal pumps, connected in parallel, with check valves installed in the pump outlets to prevent back flow. The pumps are driven by prime movers at angular velocity of 7200 rpm. The movers are simulated with the ideal angular velocity source.

The central tank bottom is elevated by 36 in with respect to the reference plane, which is drawn horizontally through the entry point of the engine. The side tank bottoms are elevated by 4.2 in each.

The left wing and the right wing tanks are equipped with the volume control block intended to cut the outlet off if fluid volume in the tank becomes less than preset volume. A similar block in the central tank controls both the tank overfill and the exposure of its highest port. In all tanks, the cutoff is performed with the 2-way valves.

Model

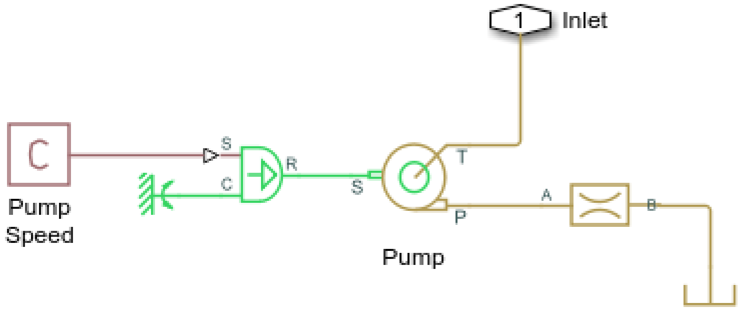


Aircraft Fuel Supply System with Three Tanks

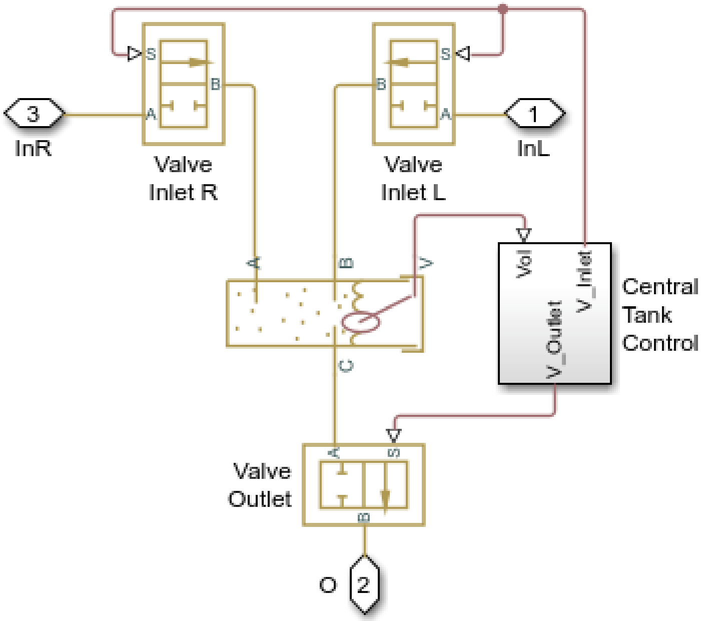
1. Plot fuel volume in each tank (see code)
2. Plot speed of each pump (see code)
3. Run all tests (see code)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example

Copyright 2009-2022 The MathWorks, Inc.

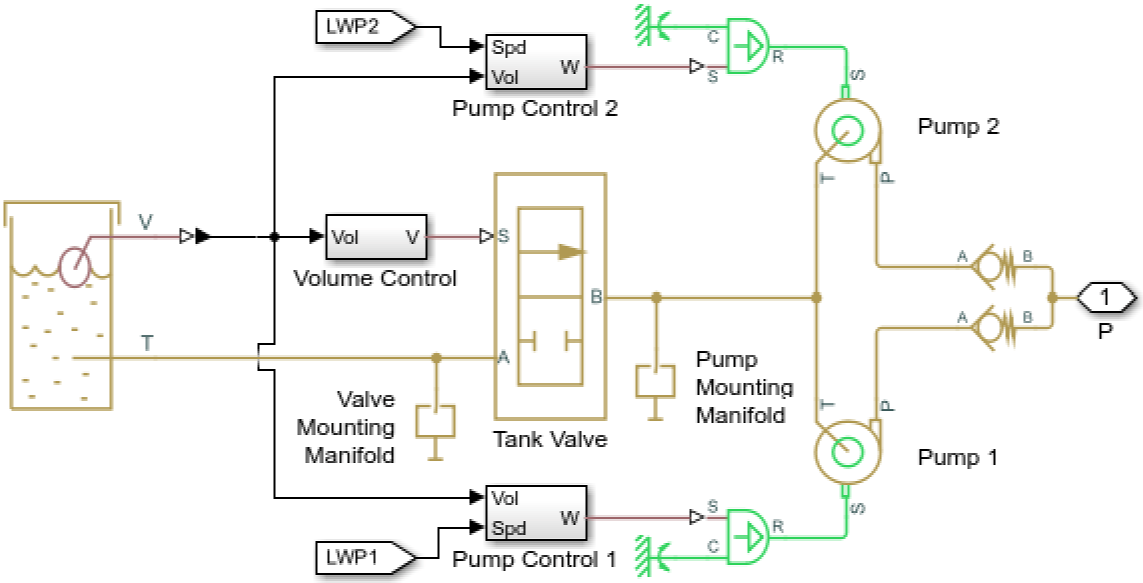
Engine Subsystem



Tank System Center Subsystem

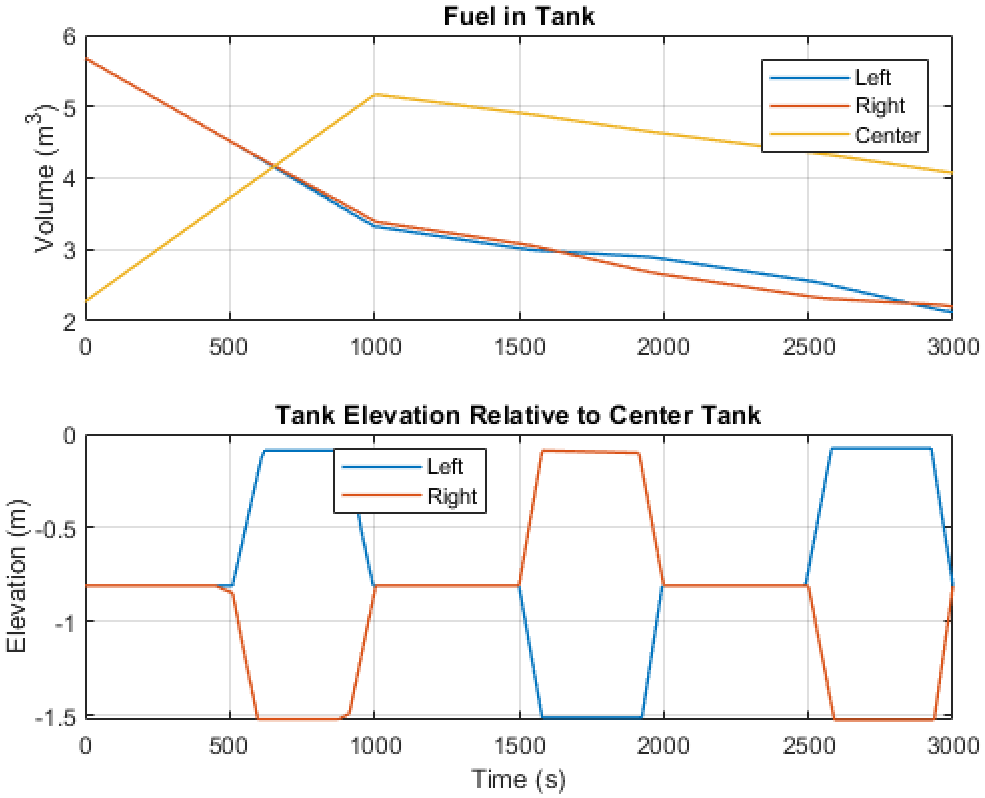


Tank System Left Wing Subsystem

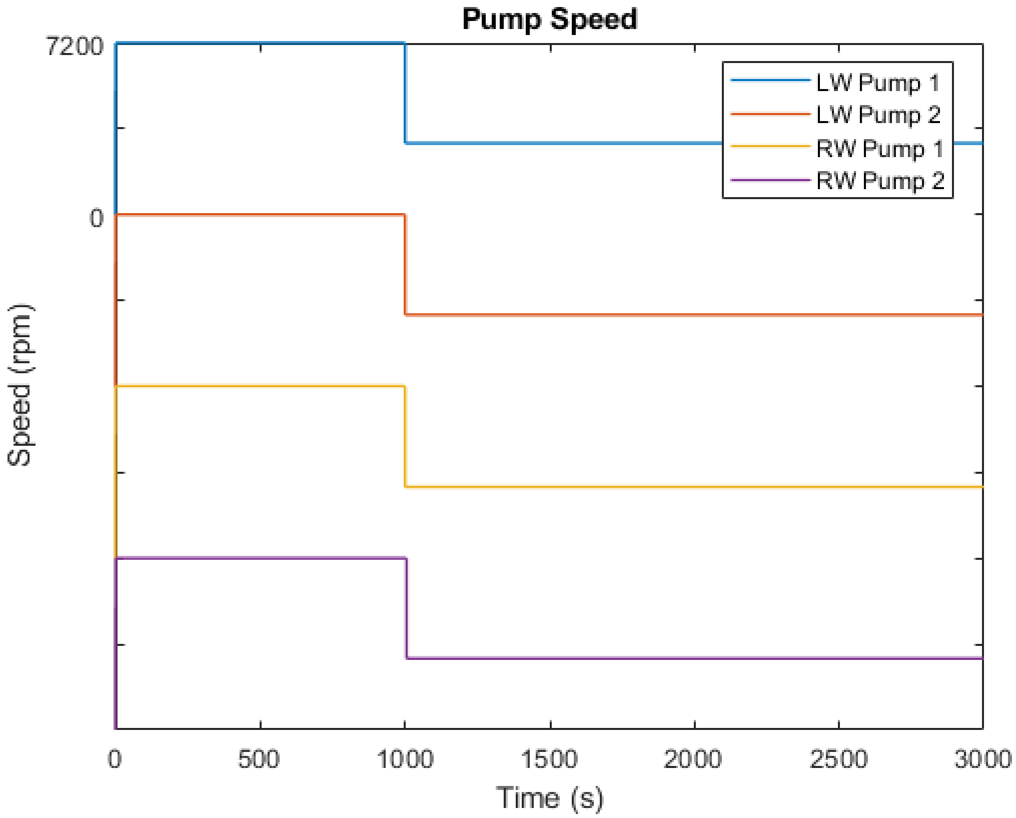


Simulation Results from Simscape Logging

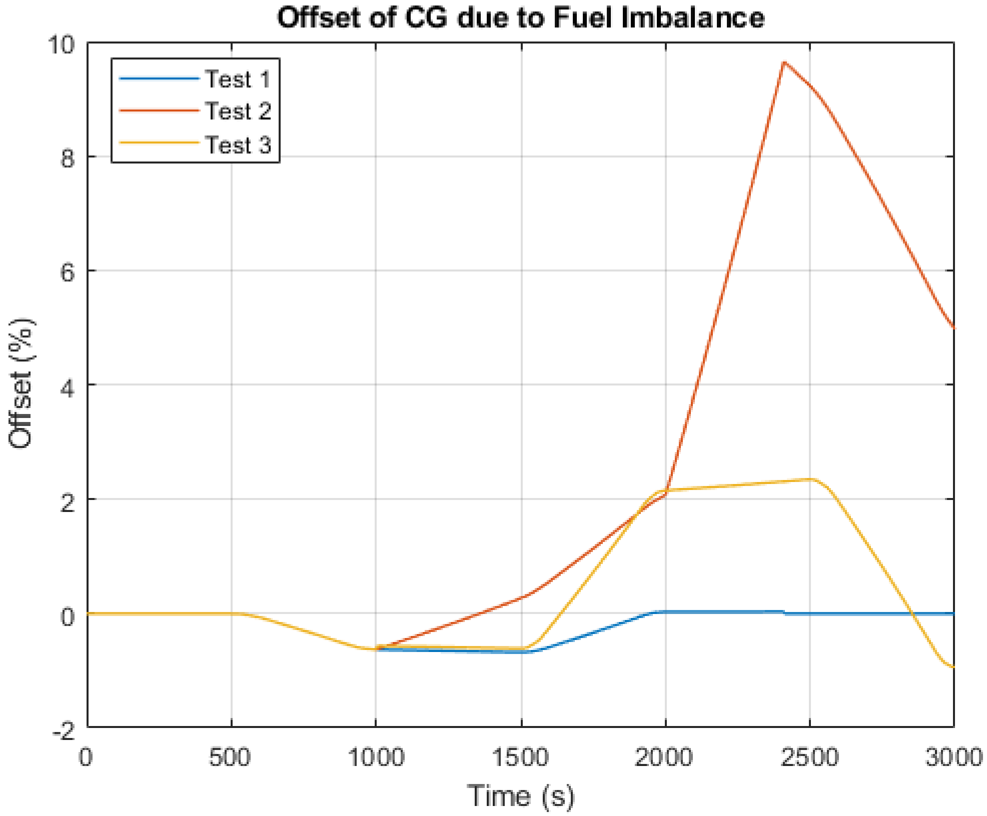
The plots below show the amount of fuel remaining in the tanks. The rate of fuel consumption from each tank is affected the pressure drop across the fuel lines due to the elevation change relative to the center tank.



The plots below show the speed for all four pumps in the fuel supply system. For some test sequences, pump speeds drop below the maximum speed to observe the effect on the fuel levels in the tanks.



The plot below shows the offset of the CG from the fuel on the aircraft as a percentage of the half the distance between the left and right fuel tanks. The plots show the results for three tests where different combinations of pump failures are triggered.

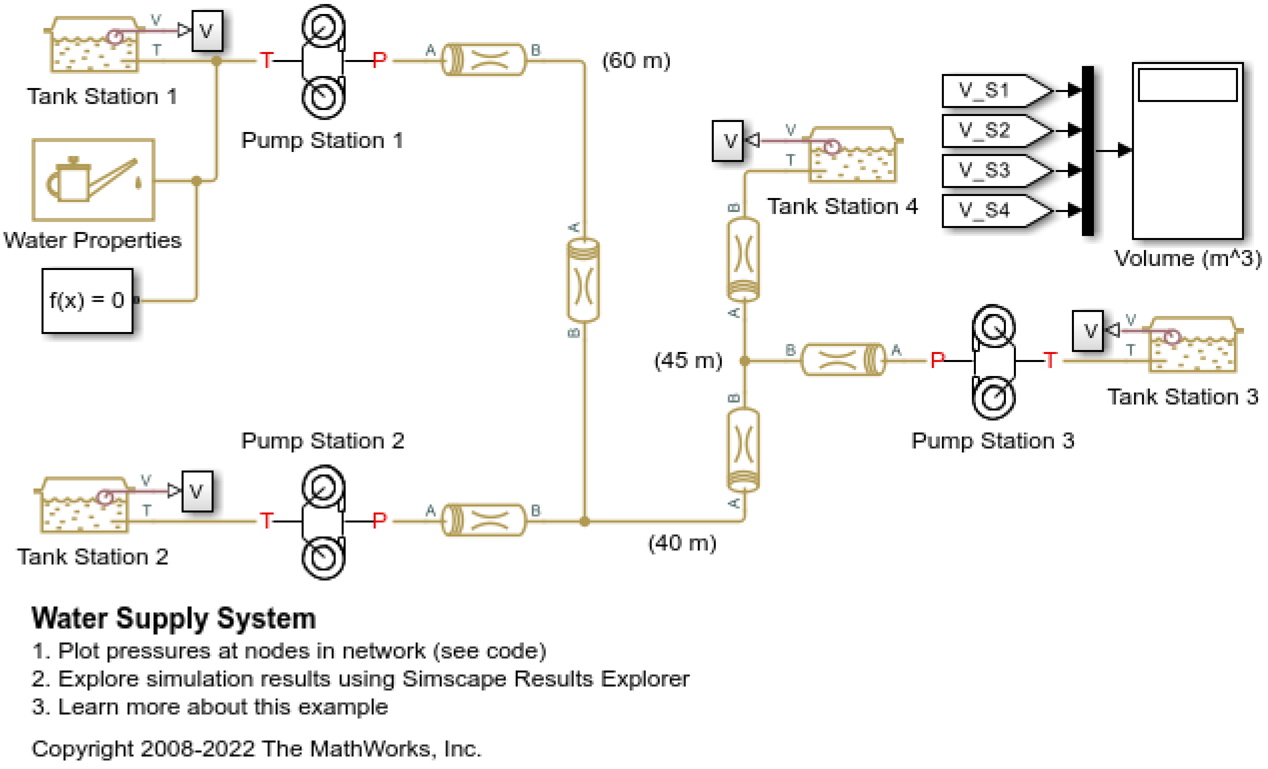

Offset of CG due to Fuel Imbalance
Test 1
Test 2
Test 3
Offset (%)
10
8
6
4
2
0
-2
0
500
1000
1500
2000
2500
3000
Time (s)

Water Supply System

This example shows a water supply system consisting of three pumping stations located at 45, 25, and 30 m with respect to to the reference plane, respectively. All three stations are expected to pump water in a tank located at 61 m. All tanks are large enough to assume that the fluid level remains nearly constant, allowing them to be modeled with the Constant Head Tank block. The initial volume of water in each tank is set to 100 m^3. Each pumping station consists of a tank, two centrifugal pumps installed in parallel, and a prime mover rotating at 1700 rpm. The pump characteristics are specified using lookup tables.

The objective of the simulation is to determine steady-state flow rates and pressures. For this reason, all pipes are simulated with the Resistive Pipe LP block and no system dynamics are observed.

Model

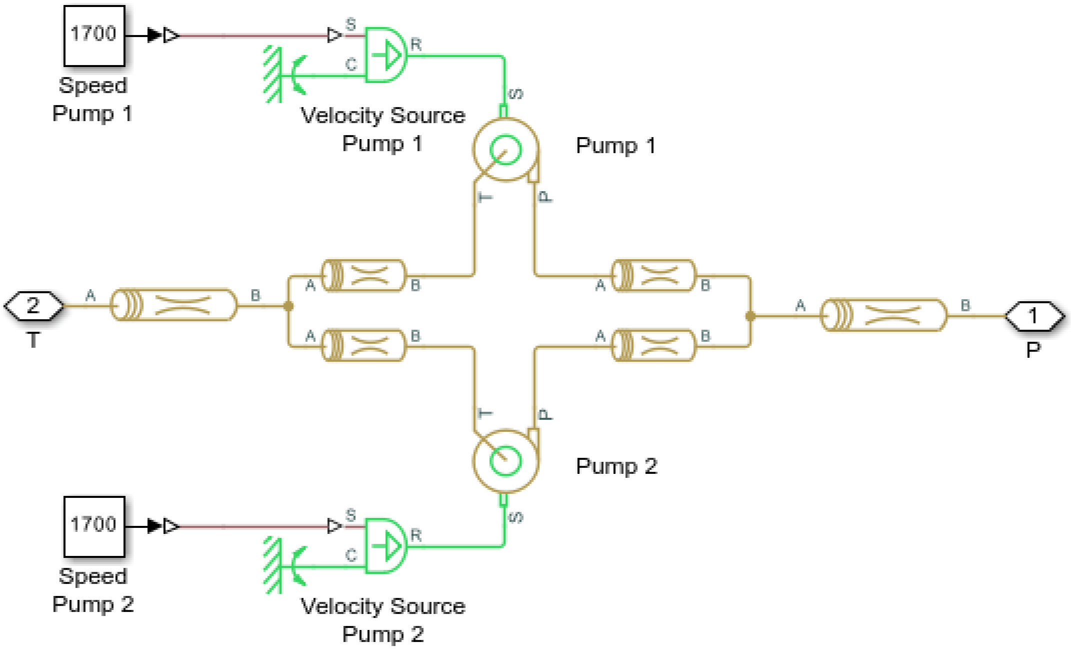


Water Supply System

1. Plot pressures at nodes in network (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

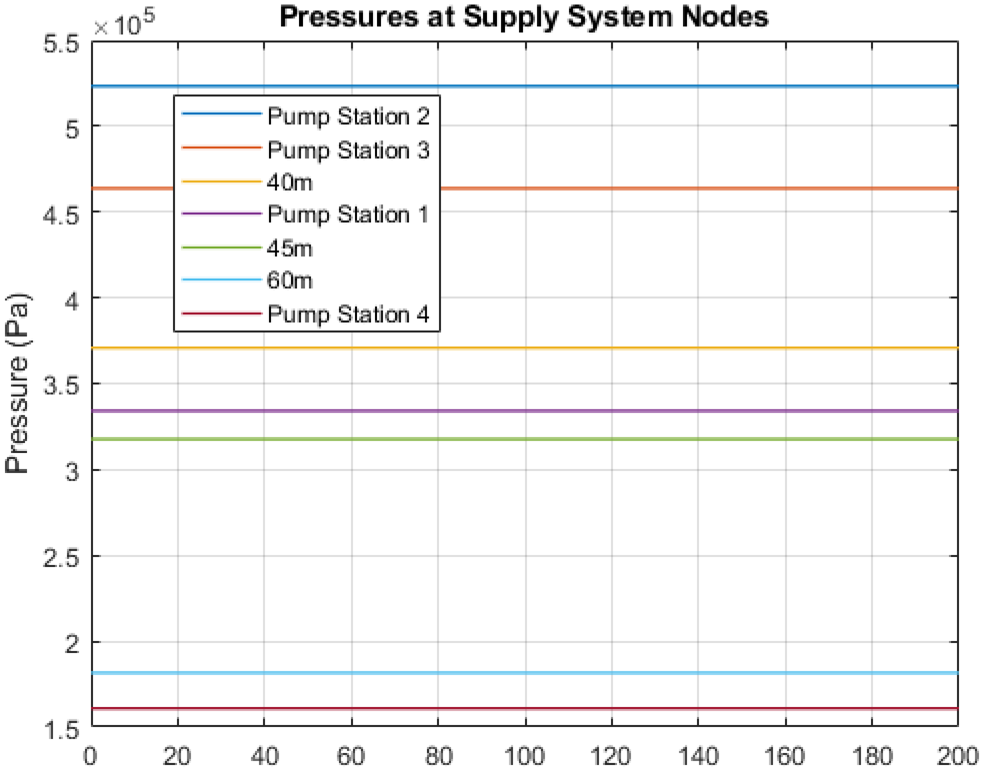
Copyright 2008-2022 The MathWorks, Inc.

Pump Station 1 Subsystem



Simulation Results from Simscape Logging

The plot below shows the pressure at each node in the water supply system.

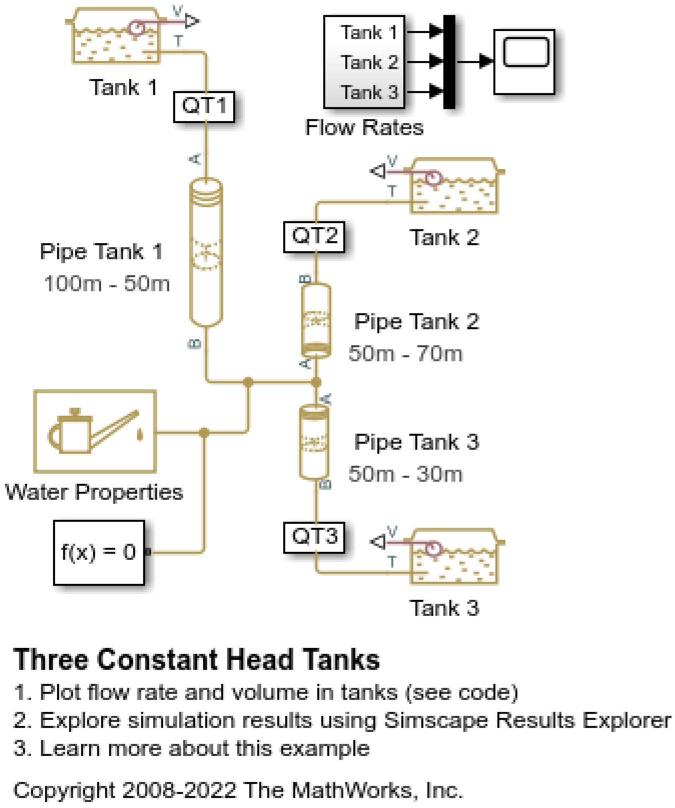

Pressures at Supply System Nodes
× 10^5
Pressure (Pa)
5.5
5
4.5
4
3.5
3
2.5
2
1.5
0
20
40
60
80
100
120
140
160
180
200
Pump Station 2
Pump Station 3
40m
Pump Station 1
45m
60m
Pump Station 4

Three Constant Head Tanks

This example shows a classical problem of fluid transportation: to determine flow rates, pressures, and fluid volumes in a system built of three constant head tanks. The tanks are located at different elevations and connected with pipelines combined together in a common node located at 50 meters with respect to the reference plane. The simulation time is set to 50 seconds, which is enough for system variables to settle down and reach near steady-state values.

The pipelines are simulated with the Segmented Pipe LP block, which accounts for hydraulic losses, fluid inertia, and the head due to different node elevations.

Model



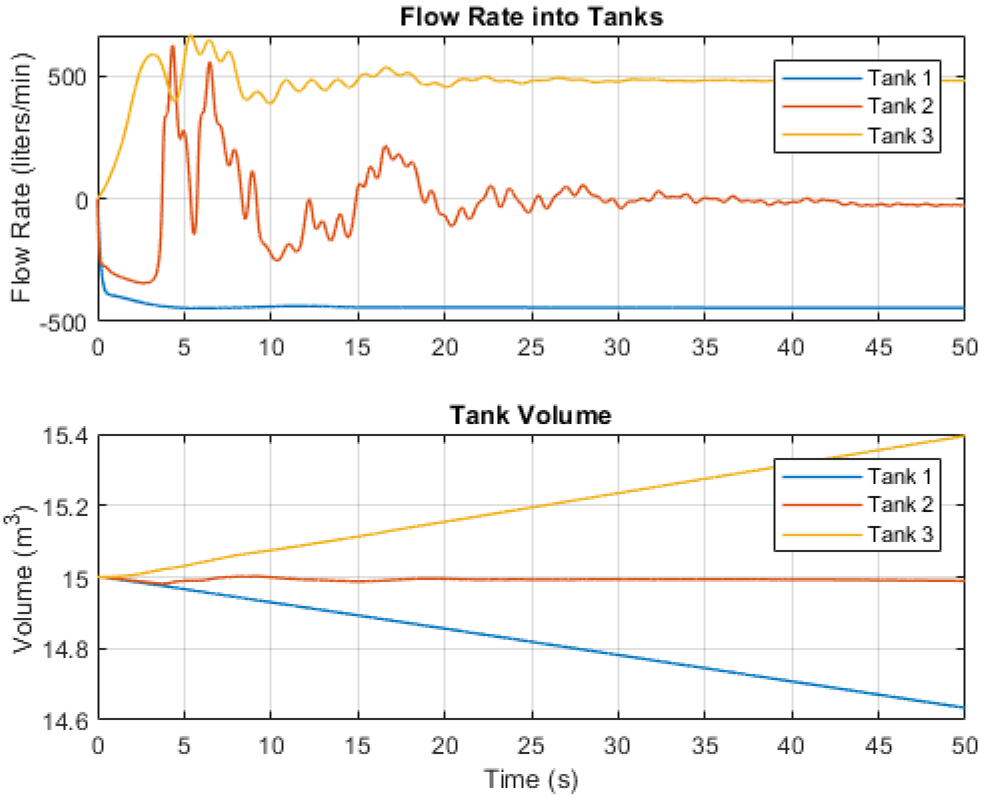
Three Constant Head Tanks

1. Plot flow rate and volume in tanks (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2008-2022 The MathWorks, Inc.

Simulation Results from Simscape Logging

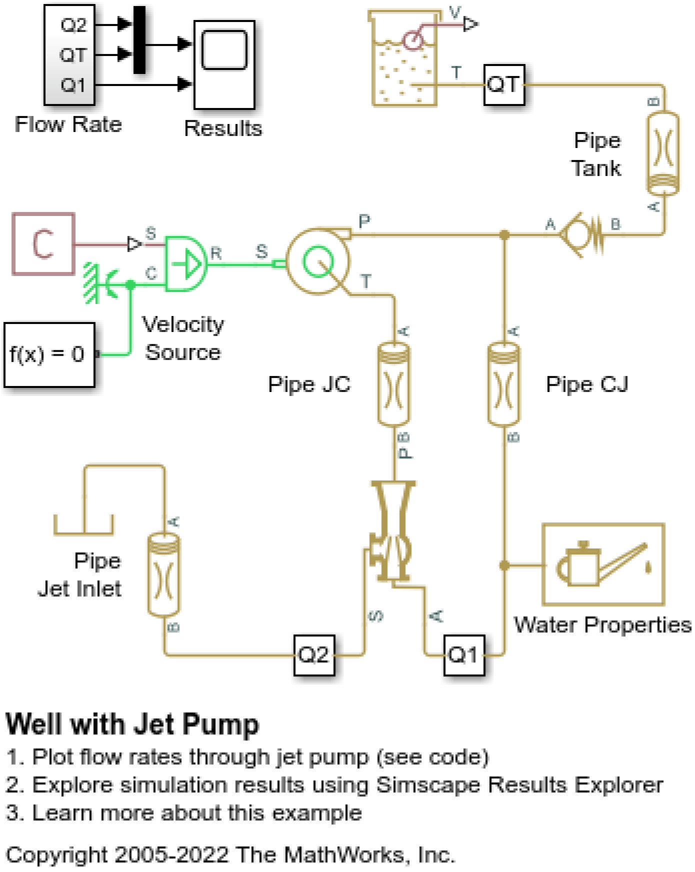
The plots below show the flow rate into and volume of water in each tank. Simulation results show that the pipelines account for fluid inertia, hydraulic losses, and head due to elevation.


Flow Rate into Tanks
Flow Rate (liters/min)
500
0
-500
0
5
10
15
20
25
30
35
40
45
50
Tank 1
Tank 2
Tank 3
Tank Volume
Volume (m^3)
15.4
15.2
15
14.8
14.6
Time (s)

Well with Jet Pump

This example shows a well jet pump installation. The well jet pump installation presented in this example consists of a surface-mounted centrifugal pump and a jet pump installed in the well below the surface of the water.

Model



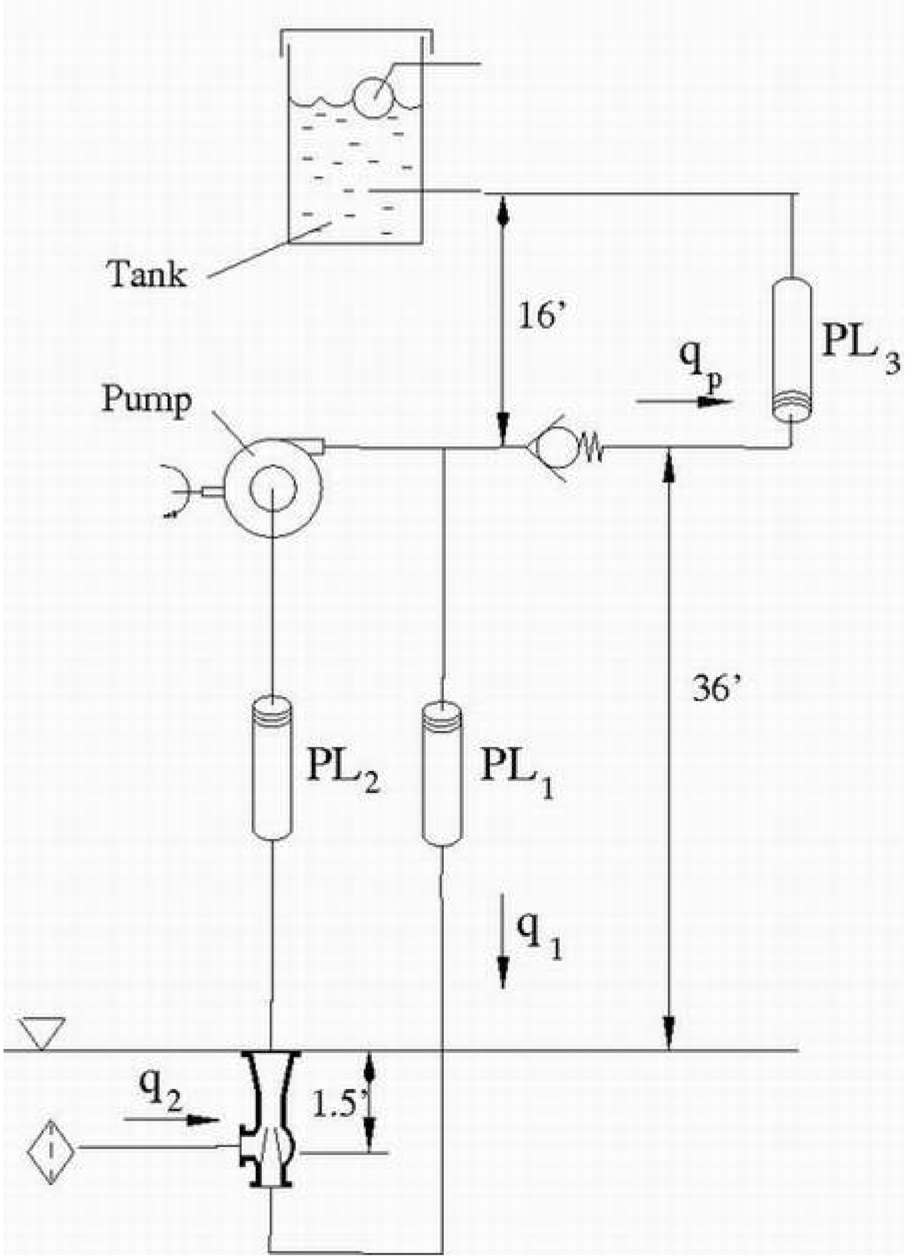
Well with Jet Pump

1. Plot flow rates through jet pump (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2005-2022 The MathWorks, Inc.

System Schematic

In this well jet pump installation, the distance between the centrifugal pump and water level is 36 feet. The jet pump installed in the well 1.5 feet below the surface of the water. See the schematic below.

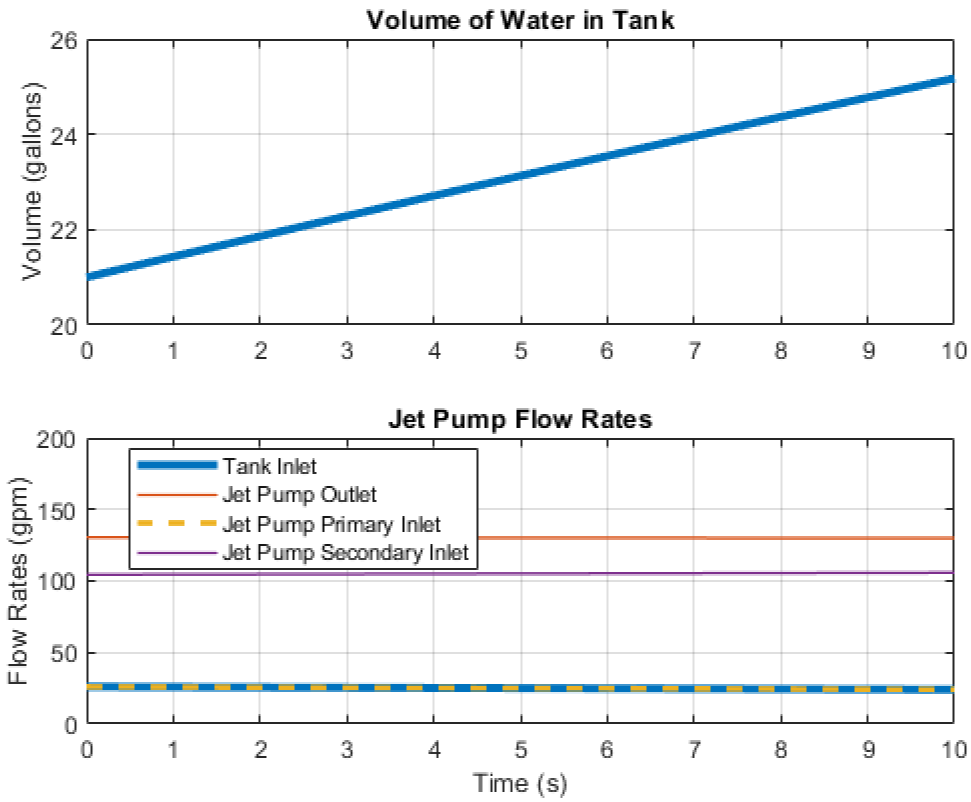


The centrifugal pump discharge pressure is 20 psi and its delivery at nominal conditions is expected to be about 130 gallons/min. The pipe between the centrifugal pump output and the jet pump inlet has a 2 inch diameter and is 37.5 feet long. The diameter of the pipe connecting the jet pump output and centrifugal pump input is 4 inches, and it is 36 feet long. The centrifugal pump output is divided into two parts: one goes through the check valve into the tank, and the rest is diverted to the inlet of the jet pump. The delivery of the pump assembly must be greater than 25 gallons/minute. The centrifugal pump NPSH must be greater than 2 psi.

The jet pump parameters were initially determined based on recommendations provided in the *Pump Handbook* by Karassic, I. J., et al (third edition, 2001), and then modified to simplify the calculations. As a result, the primary flow rate (q_1) of the jet pump is 104.5 gpm, the secondary flow rate (q_2) is 26.5 gpm, and the output of the pump assembly (q_p) is 26.5 gpm.

Simulation Results from Simscape Logging

The plots below show flow rates into and out of the jet pump and the volume of the water tank.

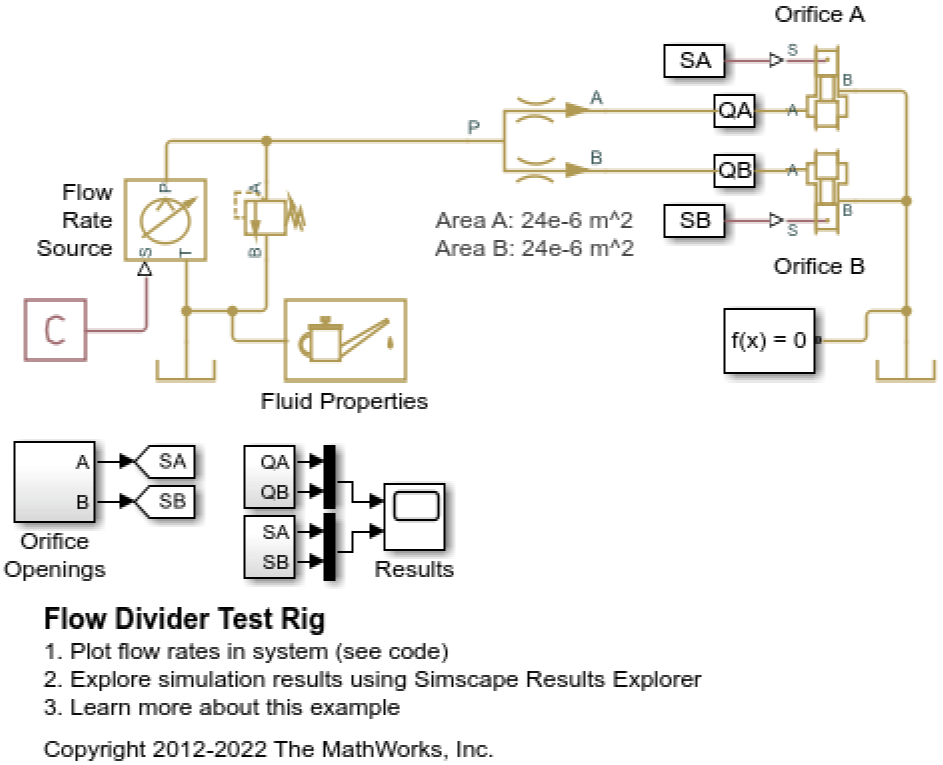


Flow Divider Test Rig

This example shows the behavior of a flow divider subjected to varying loads. The model consists of a power unit, flow divider, and two consumers. The power unit is modeled using a flow rate source and a pressure relief valve. The Flow Divider block splits the flow between the two variable orifices.

To test the ability of the Flow Divider block to split the flow equally between the two consumers, the load in the branches is varied by changing the passage area of the load orifices.

Model



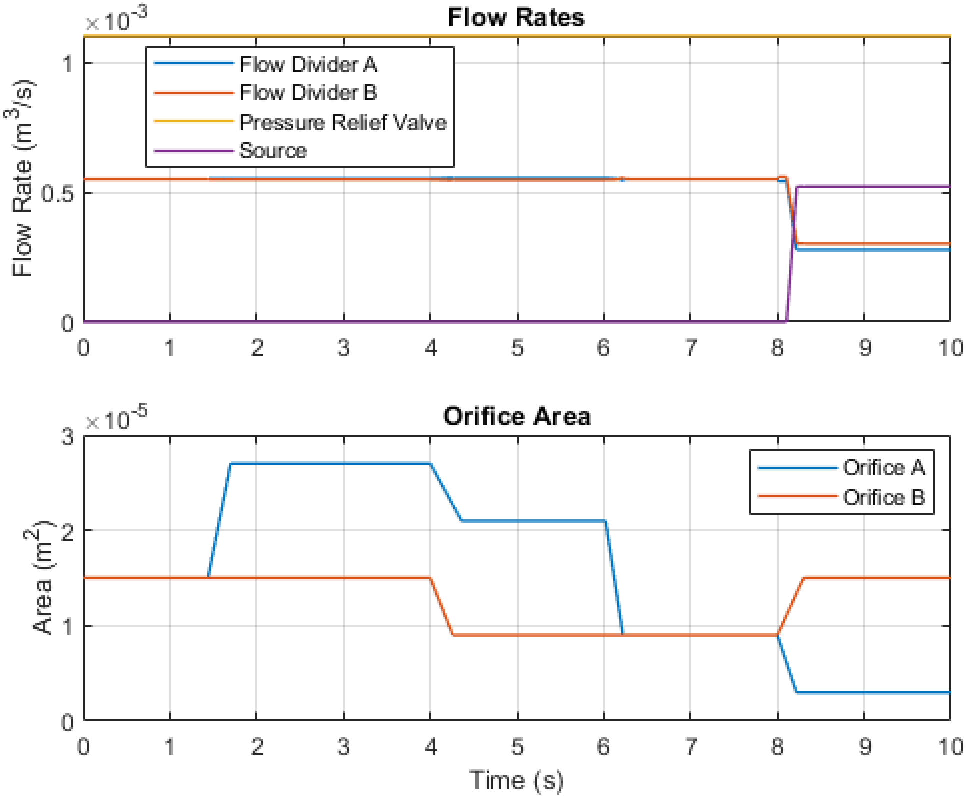
Flow Divider Test Rig

1. Plot flow rates in system (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2012-2022 The MathWorks, Inc.

Simulation Results from Simscape Logging

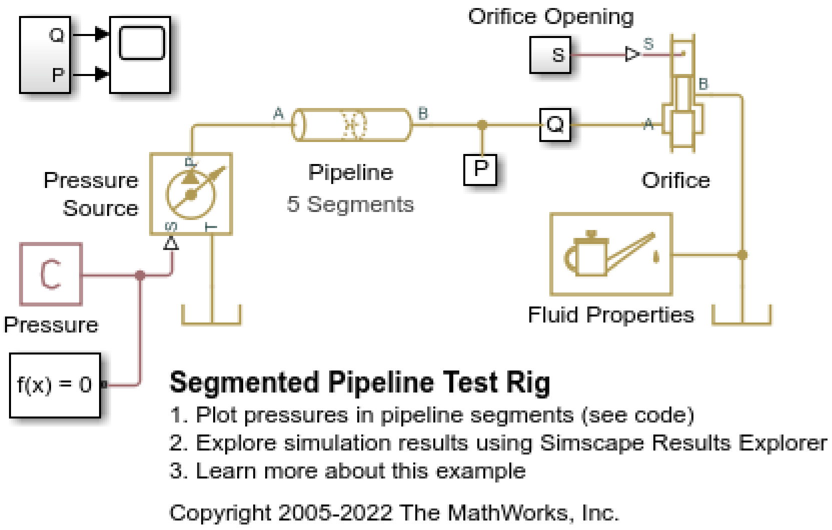
The plots below show the behavior of the flow divider as the load on each branch is varied. The flow divider manages to divide the flow nearly equal to the ratio of the flow divider orifice areas for the entire simulation. Near the end, the restrictions downstream of the flow divider become high enough that flow is diverted through the pressure relief valve.


Flow Rates
×10-3
Flow Rate (m3/s)
1
0.5
0
0
1
2
3
4
5
6
7
8
9
10
Flow Divider A
Flow Divider B
Pressure Relief Valve
Source
Orifice Area
×10-5
Area (m2)
3
2
1
0
Orifice A
Orifice B
Time (s)

Segmented Pipeline Test Rig

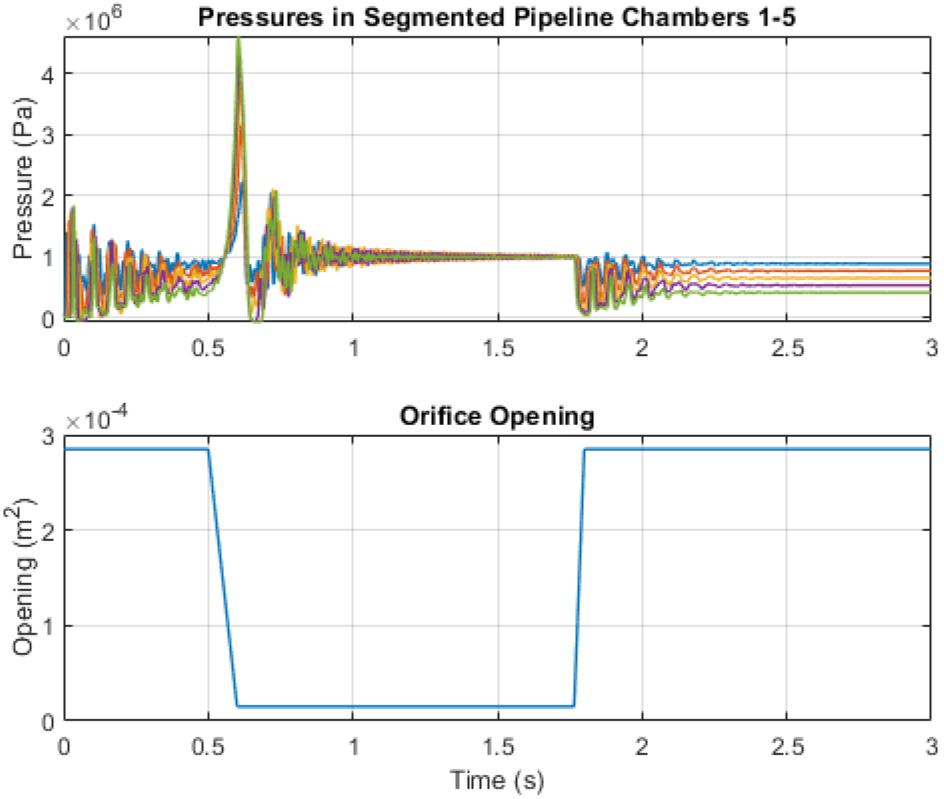
This example shows a test rig used to investigate the water hammer effect, which occurs when a variable orifice is abruptly shut off with full flow rate flowing through the branch. The water hammer behavior of the pipeline can be investigated by changing pipeline dimensions, number of segments in the pipeline model, fluid properties, valve closure speed, and minimum opening.

Model


Q
P
Orifice Opening
S
S
B
A
B
Q
A
Pipeline
5 Segments
P
Orifice
Pressure
Source
P
S
T
C
Pressure
Fluid Properties
f(x) = 0
Segmented Pipeline Test Rig
1. Plot pressures in pipeline segments (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example
Copyright 2005-2022 The MathWorks, Inc.

Simulation Results from Simscape Logging

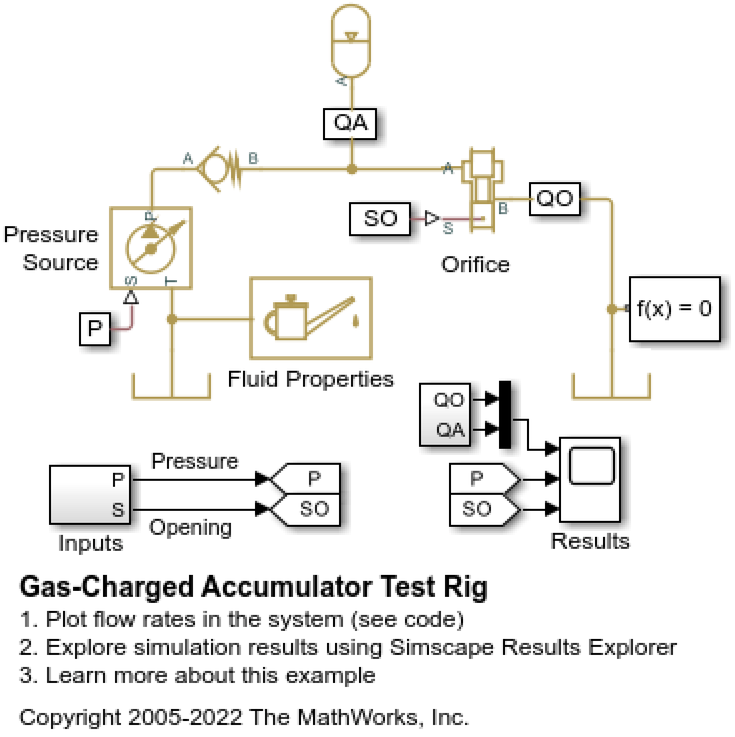
The plot below shows the pressure in each of the chambers in the segmented pipeline. The orifice at the outlet of the pipeline closes suddenly, creating pressure waves within the pipeline (water hammer effect).


Pressures in Segmented Pipeline Chambers 1-5
×10^6
Pressure (Pa)
0
1
2
3
4
0
0.5
1
1.5
2
2.5
3
Orifice Opening
×10^-4
Opening (m^2)
0
1
2
3
0
0.5
1
1.5
2
2.5
3
Time (s)

Gas-Charged Accumulator Test Rig

This example shows a test rig for a gas-charged accumulator. The accumulator is charged by the pressure source while the orifice is closed, held at its charged pressure by the check valve, and discharged as the orifice opens.

Model

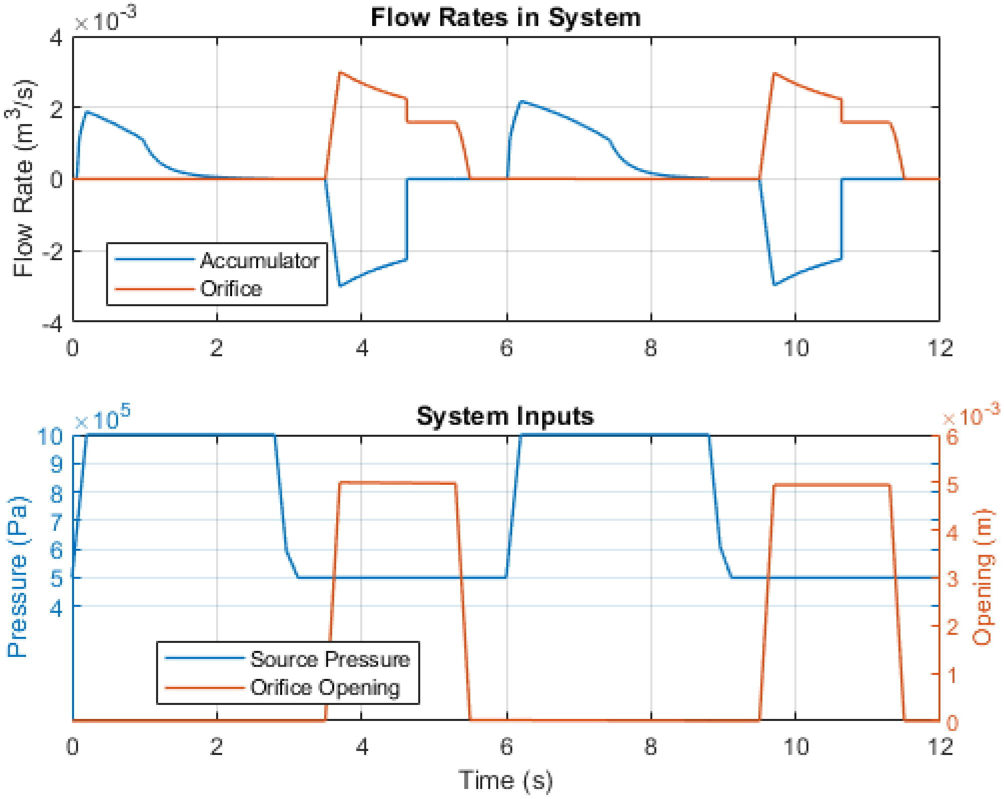


Gas-Charged Accumulator Test Rig

1. Plot flow rates in the system (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Simulation Results from Simscape™ Logging

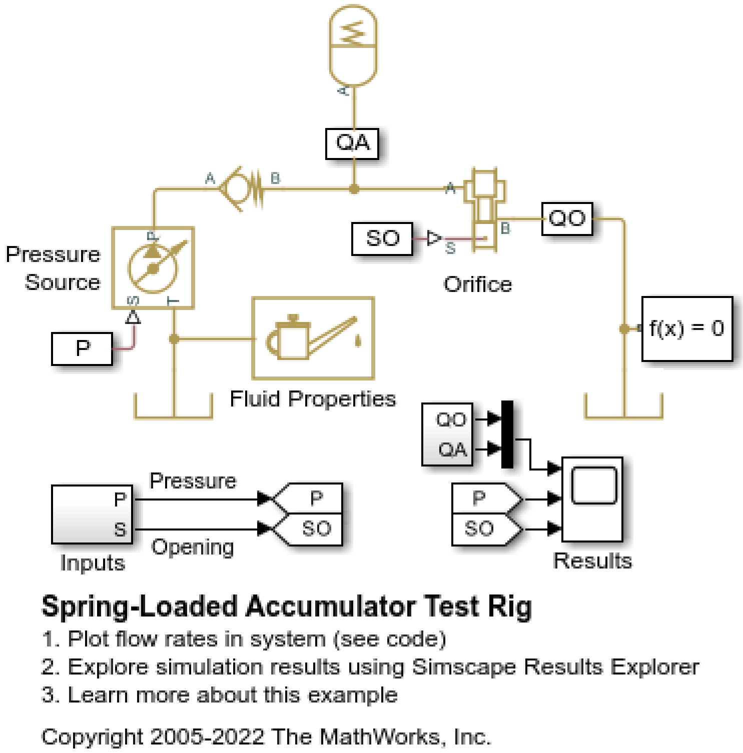
The plots below show two cycles of charging and discharging the accumulator. While the orifice is closed, the supply pressure is increased, charging the accumulator. The check valve holds the pressure in the accumulator as the supply pressure is dropped. The orifice opens, allowing the accumulator to discharge. Once the accumulator is empty, the flow rate through the orifice stays steady at a lower value due to the lower supply pressure. The orifice closes, and the cycle repeats.


Flow Rates in System
×10^-3
Flow Rate (m^3/s)
4
2
0
-2
-4
0
2
4
6
8
10
12
Accumulator
Orifice
System Inputs
×10^5
Pressure (Pa)
10
9
8
7
6
5
4
×10^-3
Opening (m)
6
5
4
3
2
1
0
Source Pressure
Orifice Opening
Time (s)

Spring-Loaded Accumulator Test Rig

This example shows a test rig for a spring-loaded accumulator. The accumulator is charged by the pressure source while the orifice is closed, held at its charged pressure by the check valve, and discharged as the orifice opens.

Model



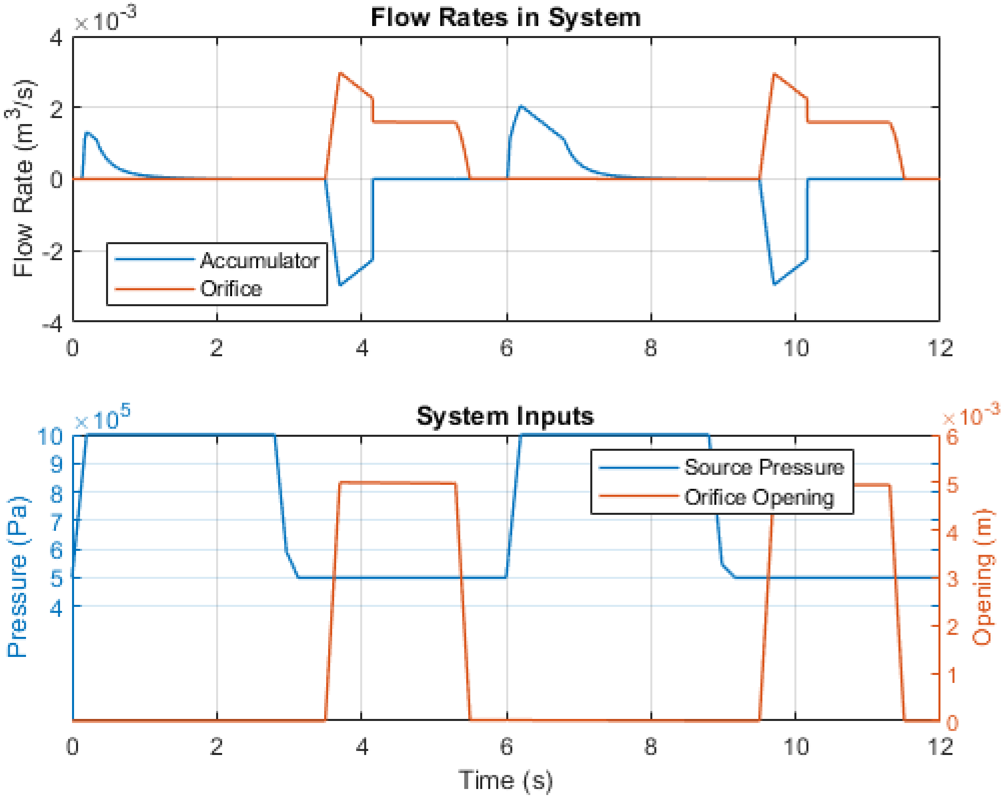
Spring-Loaded Accumulator Test Rig

1. Plot flow rates in system (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2005-2022 The MathWorks, Inc.

Simulation Results from Simscape™ Logging

The plots below show two cycles of charging and discharging the accumulator. While the orifice is closed, the supply pressure is increased, charging the accumulator. The check valve holds the pressure in the accumulator as the supply pressure is dropped. The orifice opens, allowing the accumulator to discharge. Once the accumulator is empty, the flow rate through the orifice stays steady at a lower value due to the lower supply pressure. The orifice closes, and the cycle repeats.

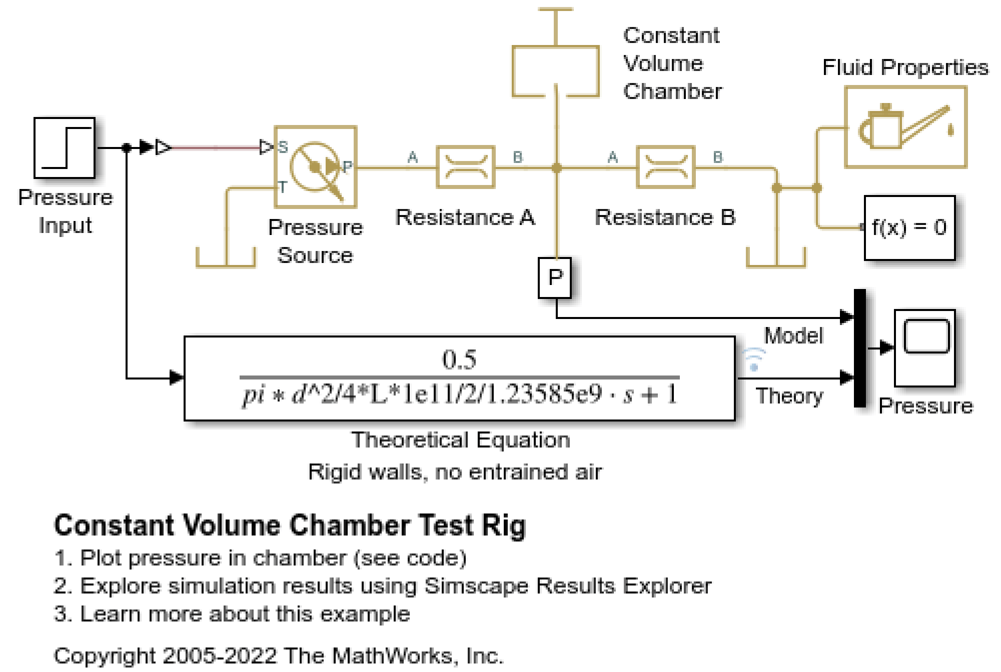

Flow Rates in System
×10-3
Flow Rate (m3/s)
4
2
0
-2
-4
0
2
4
6
8
10
12
Accumulator
Orifice
System Inputs
×105
Pressure (Pa)
10
9
8
7
6
5
4
×10-3
Opening (m)
6
5
4
3
2
1
0
Source Pressure
Orifice Opening
Time (s)

Constant Volume Chamber Test Rig

This example shows the behavior of a constant volume chamber exposed to a step increase in pressure. The chamber is placed between two linear hydraulic resistances and subjected to an abrupt change in supply pressure. The chamber is a piece of a steel cylindrical pipe of 0.03 m internal diameter, 0.036 m external diameter, and 16.5 m length. If the chamber wall is set to rigid and the amount of entrapped air is zero, the pressure change at the chamber inlet can be determined analytically.

The chamber characteristics can be investigated by changing air content in the fluid, by switching from rigid to compliant walls, and by changing wall compliance and pipe viscoelasticity time constant.

Model



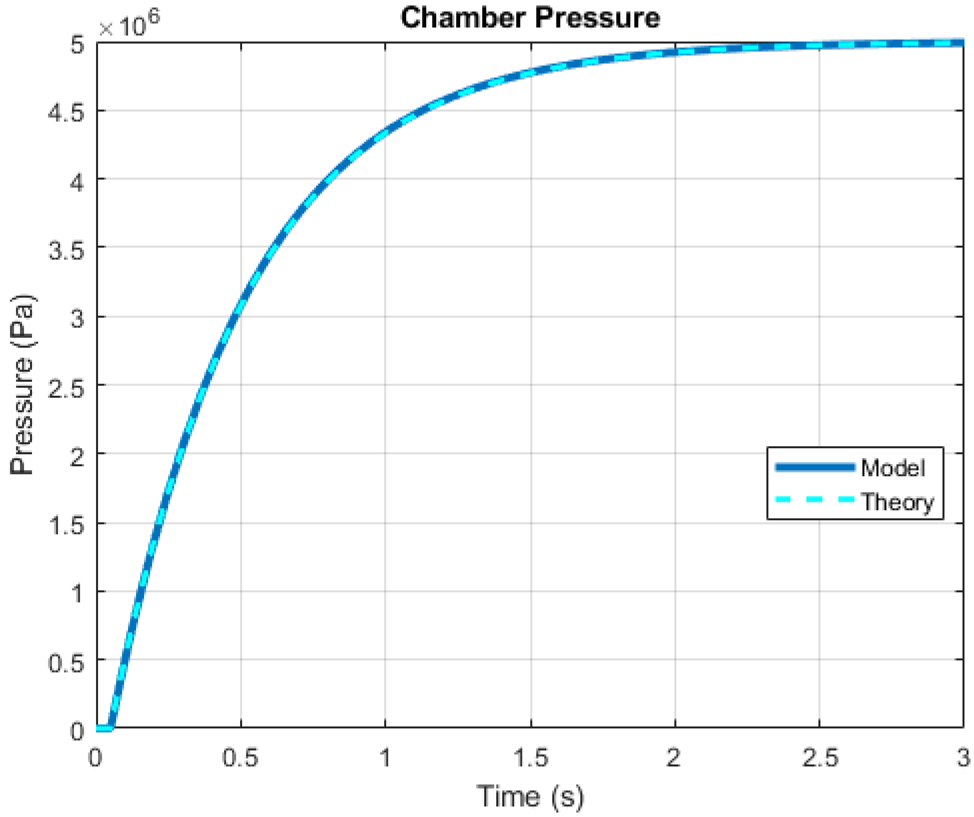
Constant Volume Chamber Test Rig

1. Plot pressure in chamber (see code)
2. Explore simulation results using Simscape Results Explorer
3. Learn more about this example

Copyright 2005-2022 The MathWorks, Inc.

Simulation Results from Simscape Logging

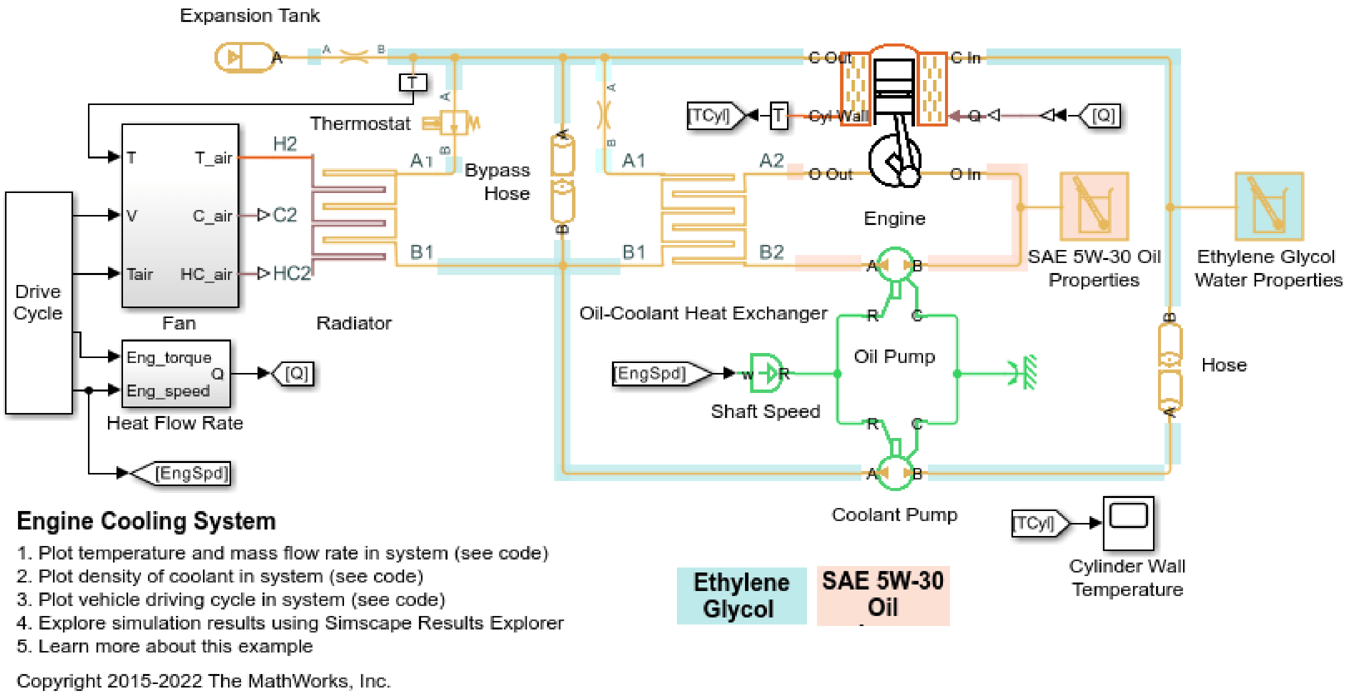
The plot below shows how the pressure in a constant volume chamber varies with time.


Chamber Pressure
$\times 10^6$
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
Pressure (Pa)
0
0.5
1
1.5
2
2.5
3
Time (s)
Model
Theory

Engine Cooling System

This example shows how to model an engine cooling system with an oil cooling circuit using Simscape™ Fluids™ Thermal Liquid blocks. The system includes a coolant circuit and an oil cooling circuit. A fixed-displacement pump drives coolant through the cooling circuit. The main portion of heat from the engine is absorbed by the coolant and dissipated through the radiator. The system temperature is regulated by the thermostat, which diverts flow to the radiator only when the temperature is above a threshold. The oil cooling circuit also absorbs some of the heat from the engine. The heat added to the oil is transferred to the coolant by the oil-coolant heat exchanger. The Radiator is the E-NTU Heat Exchanger (TL) block with the air-side flow controlled by physical signal inputs. The oil-coolant heat exchanger is the E-NTU Heat Exchanger (TL-TL) block. Both coolant pump and oil pump are driven by the engine speed.

Model



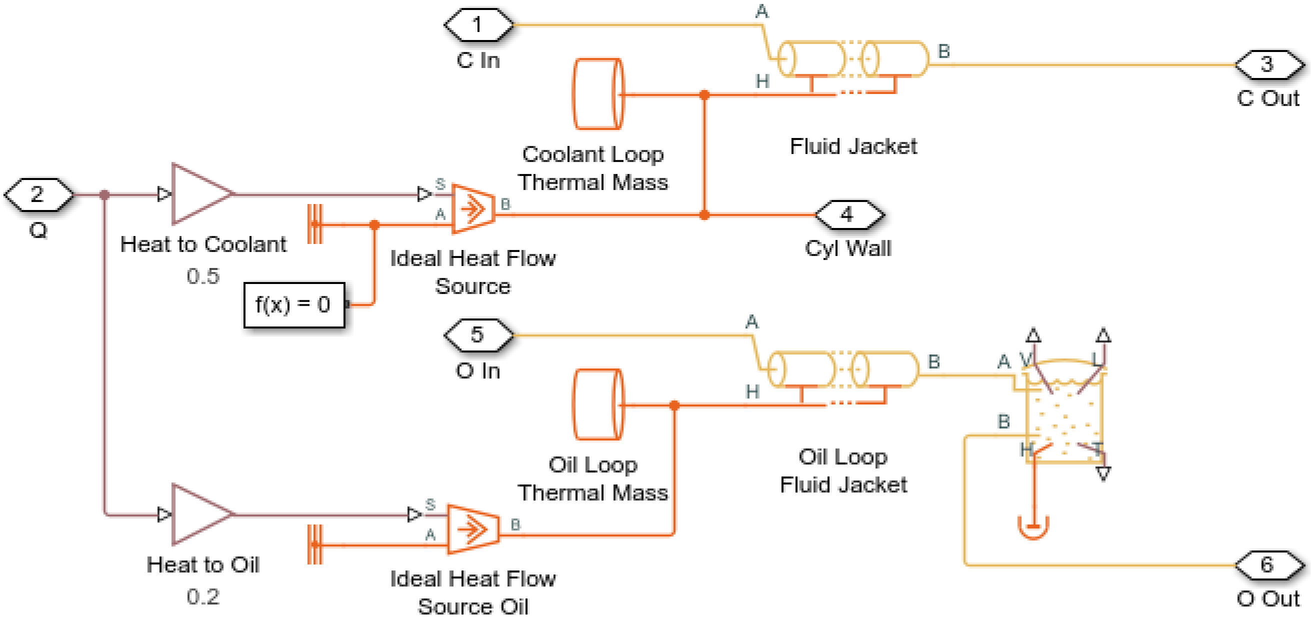
Engine Cooling System

1. Plot temperature and mass flow rate in system (see code)
2. Plot density of coolant in system (see code)
3. Plot vehicle driving cycle in system (see code)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example

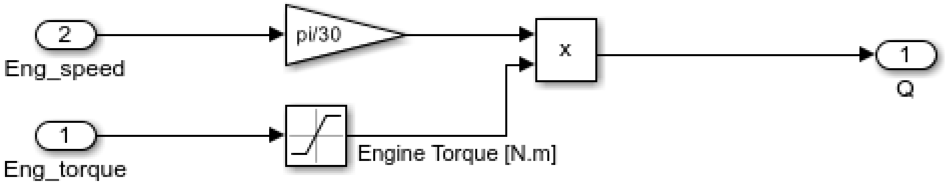
Copyright 2015-2022 The MathWorks, Inc.

Engine Subsystem

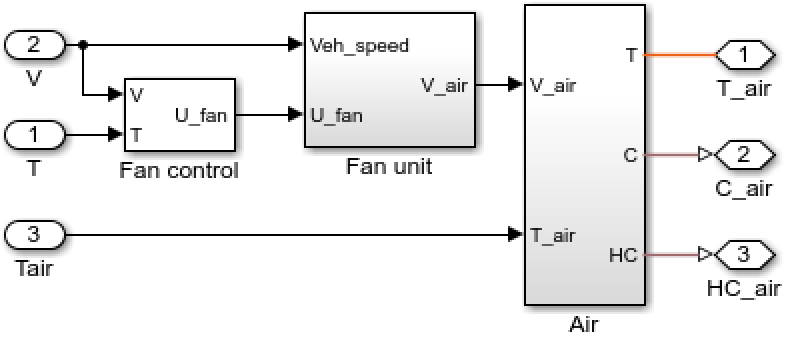
The thermal power generated by the engine is calculated as a function of the instantaneous engine speed and the engine torque. This power is separated in two parts going to the coolant and the oil circuit. It is assumed that 50% of the amount of heat rejected from the engine is added to the coolant and 20% of the heat rejected from the engine is added to the oil.



Heat Flow Rate in Engine Subsystem

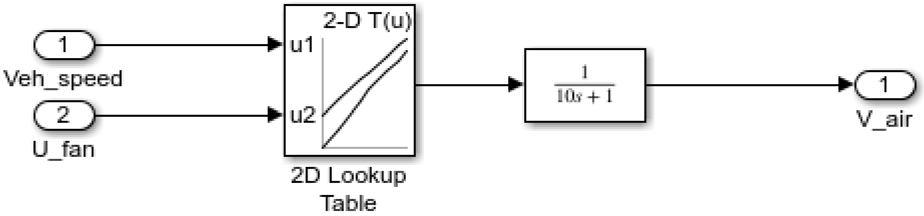


Fan Subsystem



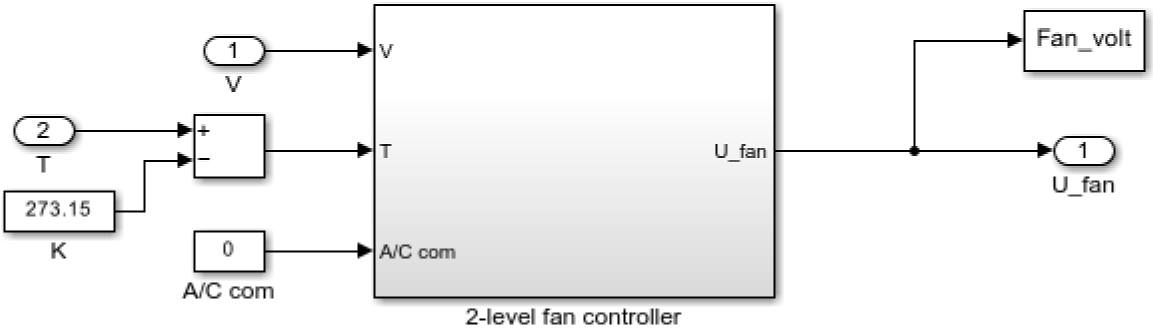
Fan Unit Subsystem

The cooling air speed in the radiator is modeled with a 2D lookup table as a function of instantaneous vehicle speed and the fan controller signal.

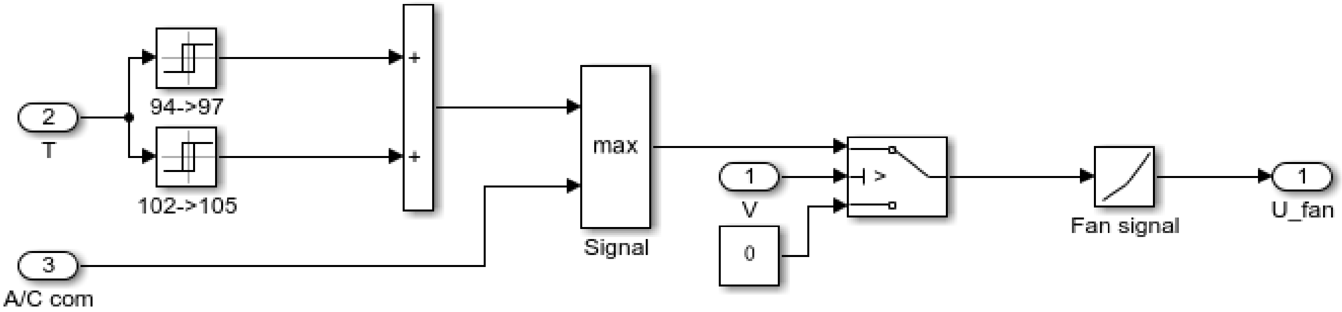


Fan Control Subsystem

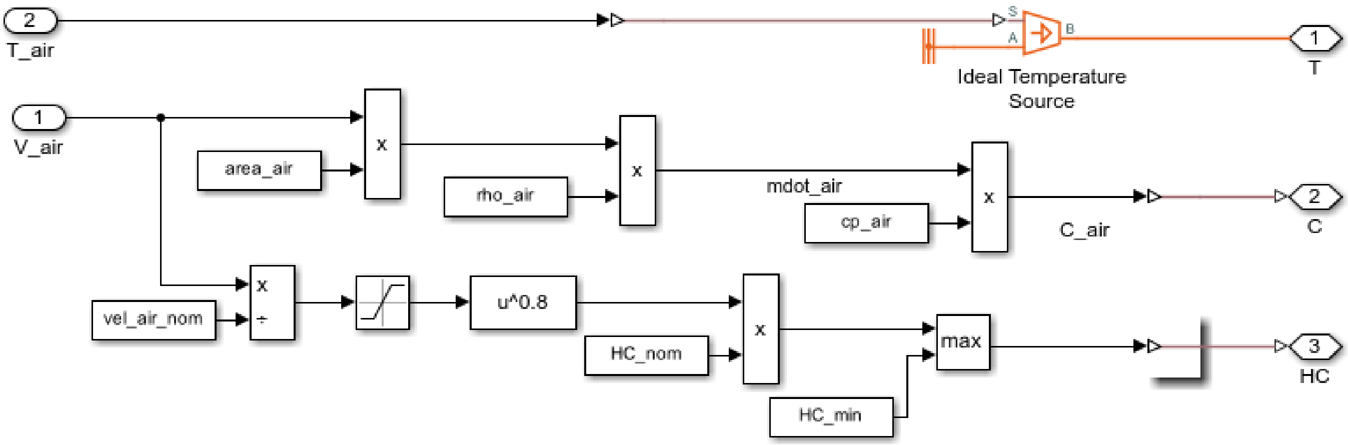
The fan controller unit includes two control levels. The primary level operates at coolant temperatures higher than the primary control target temperature. Once the coolant temperature exceeds a temperature threshold, the secondary level is activated.



2-Level Fan Controller Subsystem

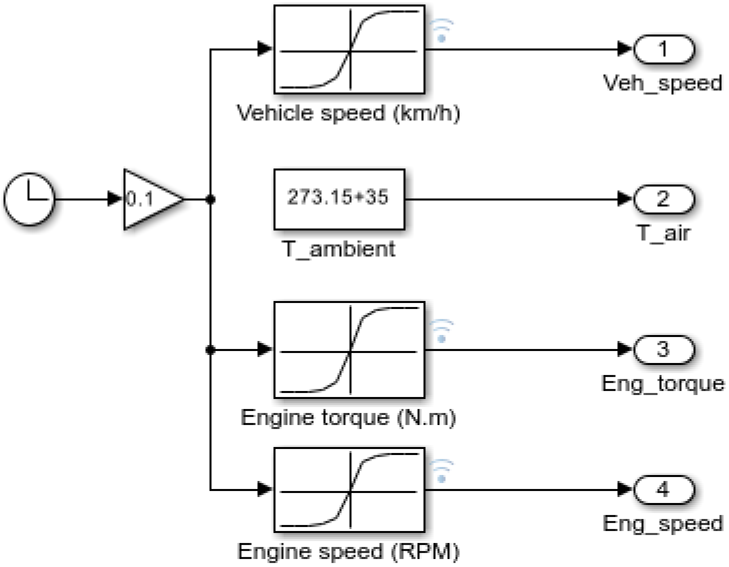


Air Subsystem

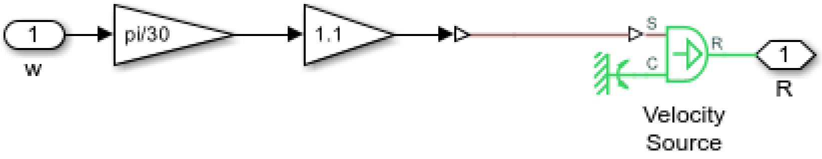


Driving Cycle Subsystem

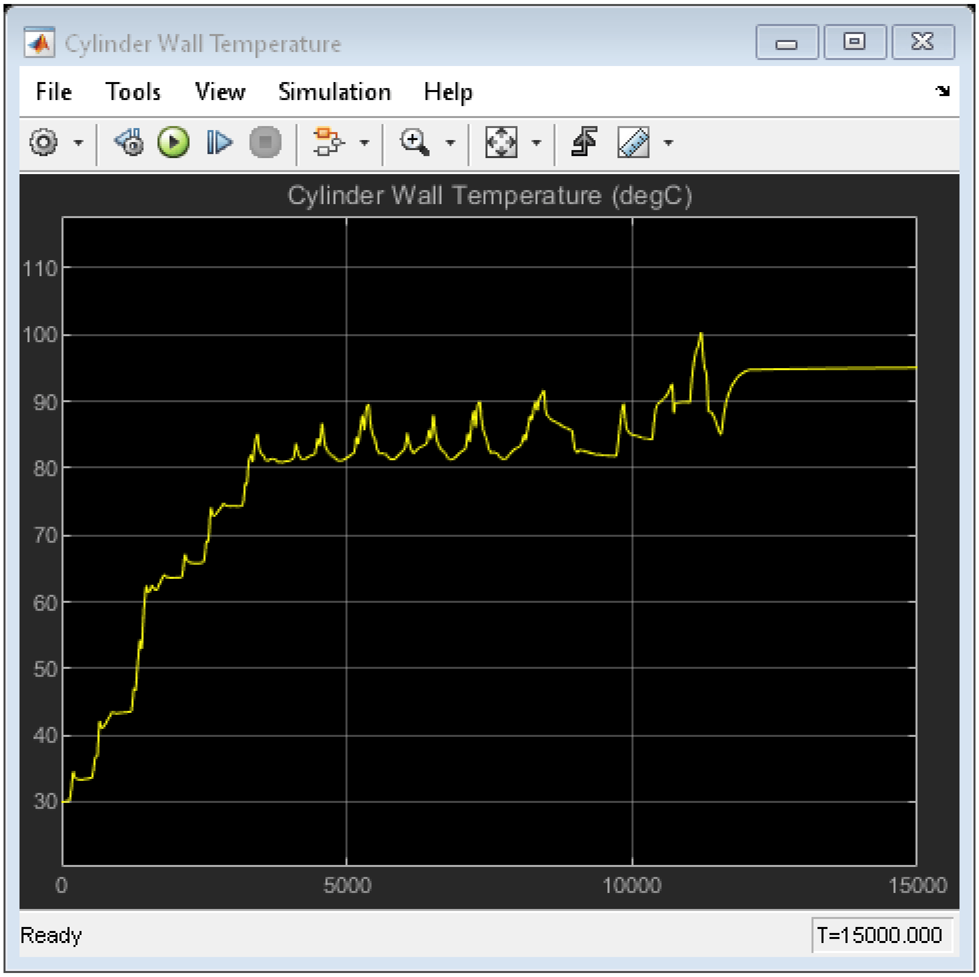
The real vehicle driving cycle is presented based on the instantaneous vehicle speed, the engine speed, and the engine torque inputs.



Shaft Speed Subsystem

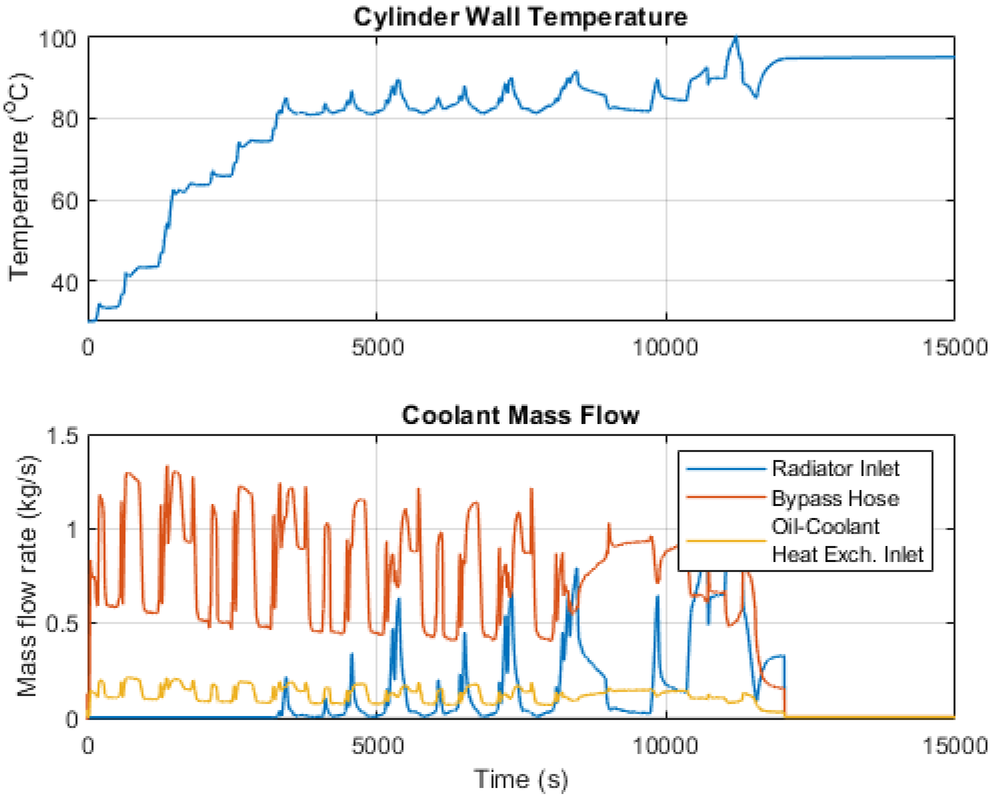


Simulation Results from Scopes

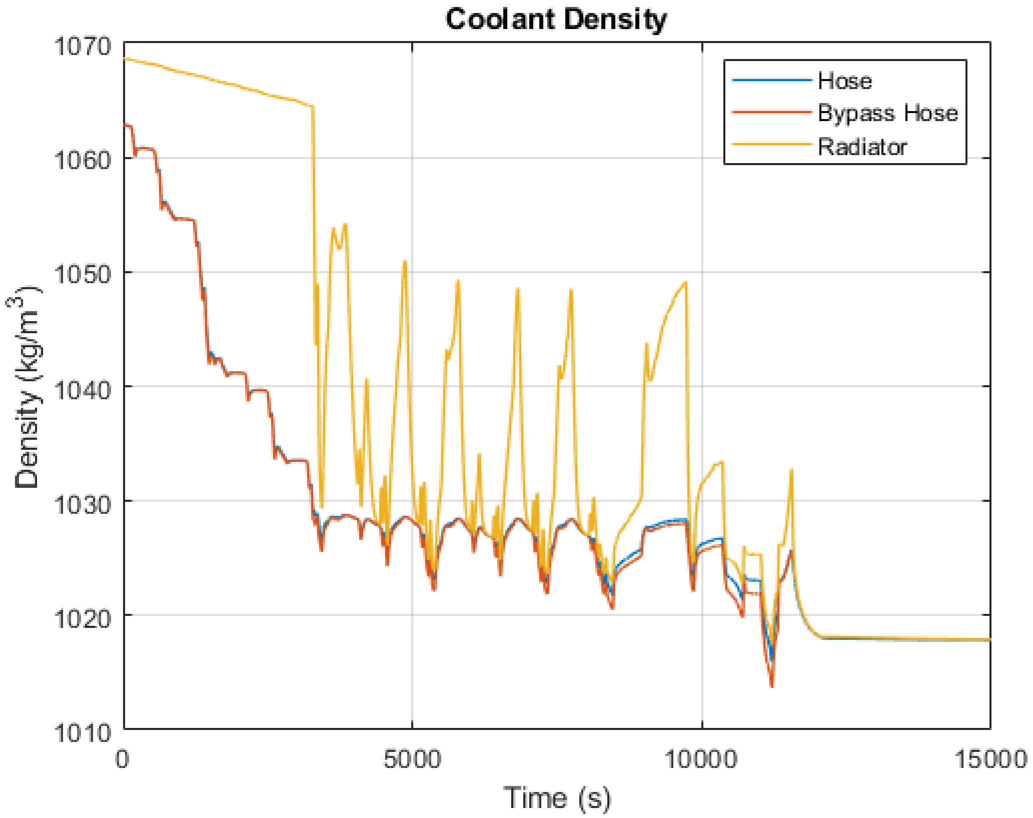

Cylinder Wall Temperature
File Tools View Simulation Help
Cylinder Wall Temperature (degC)
110
100
90
80
70
60
50
40
30
0
5000
10000
15000
Ready
T=15000.000

Simulation Results from Simscape Logging

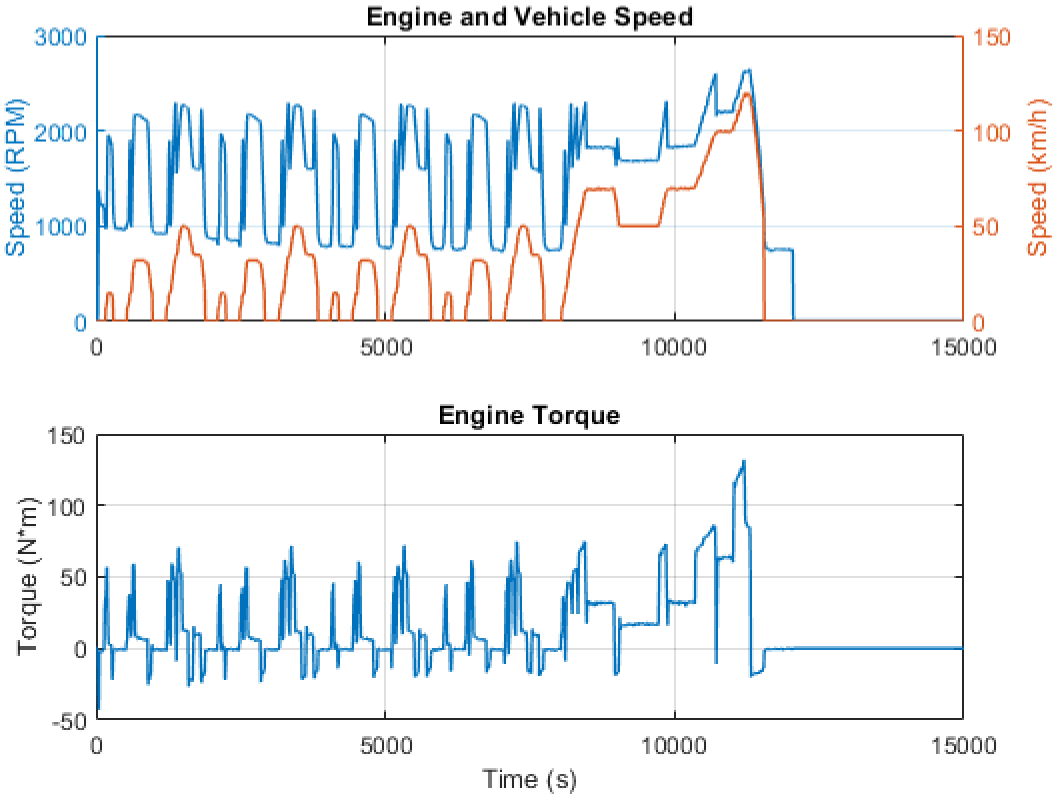
These plots show the effect of opening the thermostat in the engine cooling system. The temperature of the engine block climbs steadily until the thermostat opens. At that point, the flow of coolant through the radiator climbs sharply and the flow of coolant through the bypass hose decreases. Because coolant passing through the radiator releases heat to the atmosphere, the engine block temperature rises more slowly.



This plot shows the density of the coolant at different locations in the cooling system over time. The density of the coolant varies throughout the network based on the local temperature and pressure.



These plots show the instantaneous vehicle speed, engine speed and torque input profiles. The vehicle starts from rest accelerating to near its maximum speed. Subsequently, the vehicle decelerates till it fully stops.

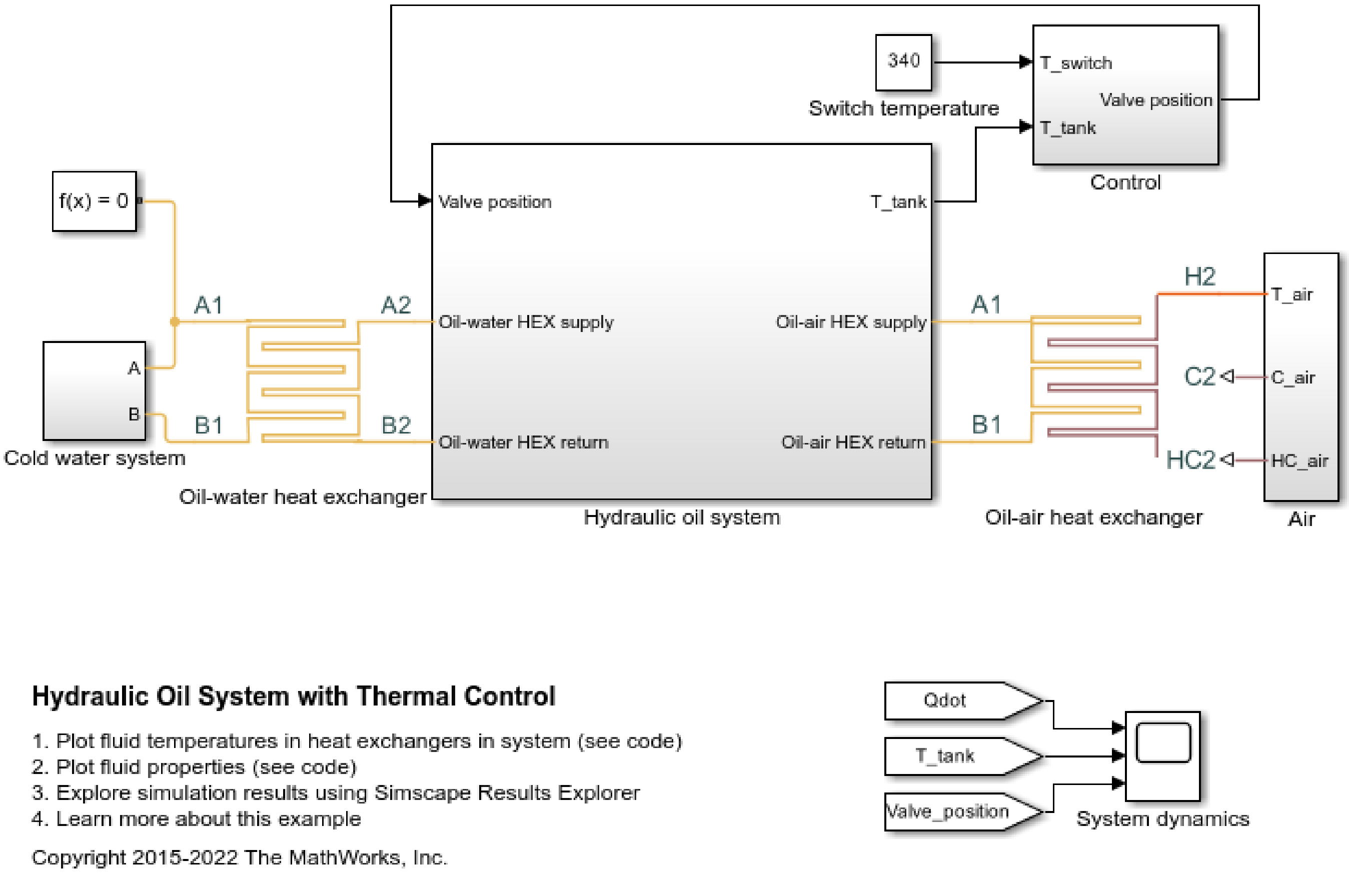

Engine and Vehicle Speed
Speed (RPM)
3000
2000
1000
0
Speed (km/h)
150
100
50
0
0
5000
10000
15000
Engine Torque
Torque (N*m)
150
100
50
0
-50
0
5000
10000
15000
Time (s)

Hydraulic Oil System with Thermal Control

This example shows a hydraulic oil system with a thermal control using Simscape™ Fluids™ Thermal Liquid blocks. The hydraulic oil system consists of an oil storage tank represented by the Tank (TL) block with two inlets, a pump represented by a Mass Flow Rate Source (TL) block, and pipelines represented by Pipe (TL) block.

The system is subjected to intensive heating and as the result, the oil temperature in the tank keeps rising. To prevent the oil from overheating, the primary oil-air heat exchanger absorbs heat from the system. In the case when the oil-air heat exchanger cannot maintain the tank temperature below a threshold, the second oil-water heat exchanger is added to the system to further absorb heat. The oil-air heat exchanger is the E-NTU Heat Exchanger (TL) block with the air-side flow controlled by physical signal inputs. The flow of the oil into each heat exchanger is regulated through the 3-Way Directional Valve (TL) block.

Model

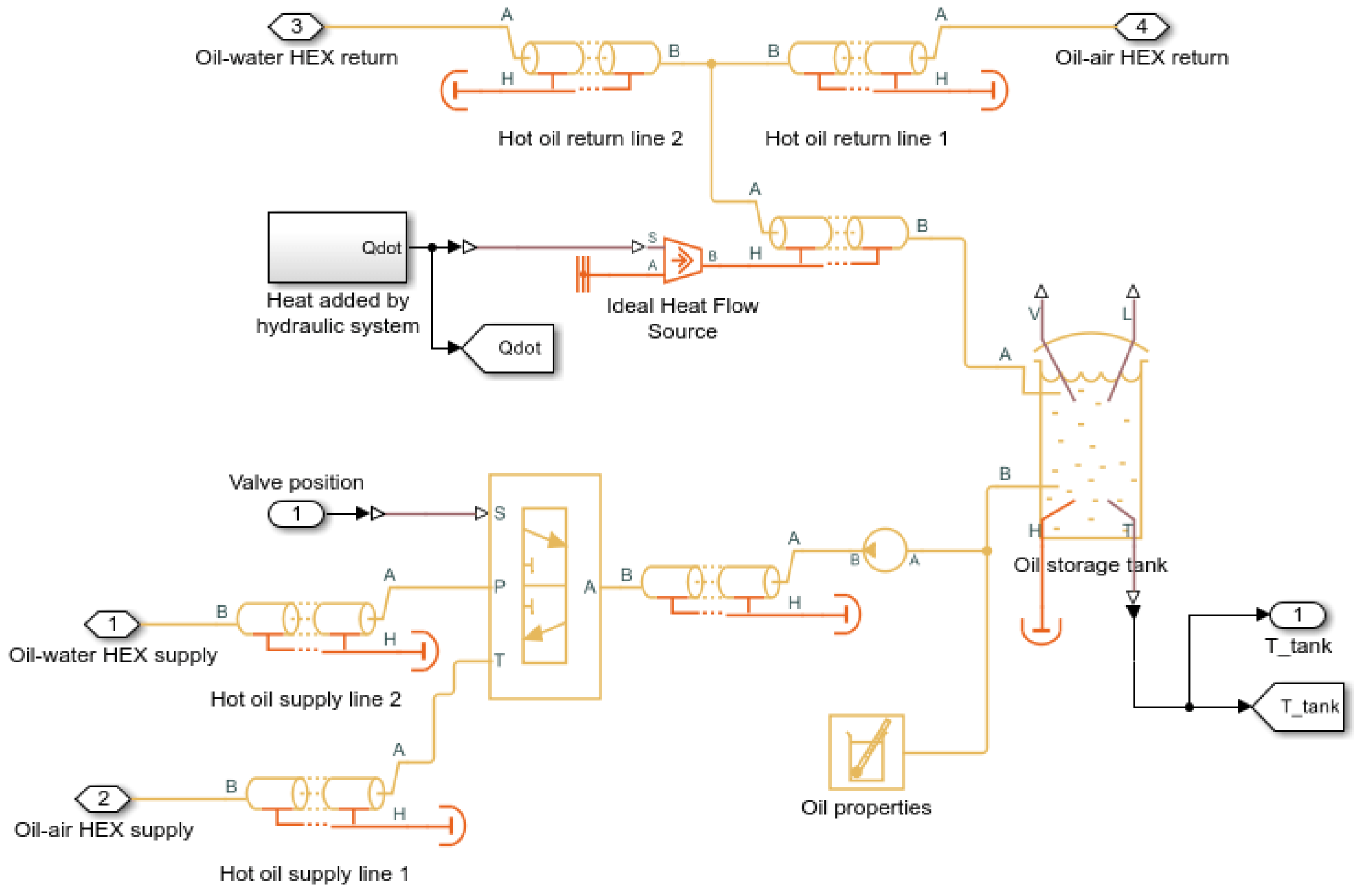


Hydraulic Oil System with Thermal Control

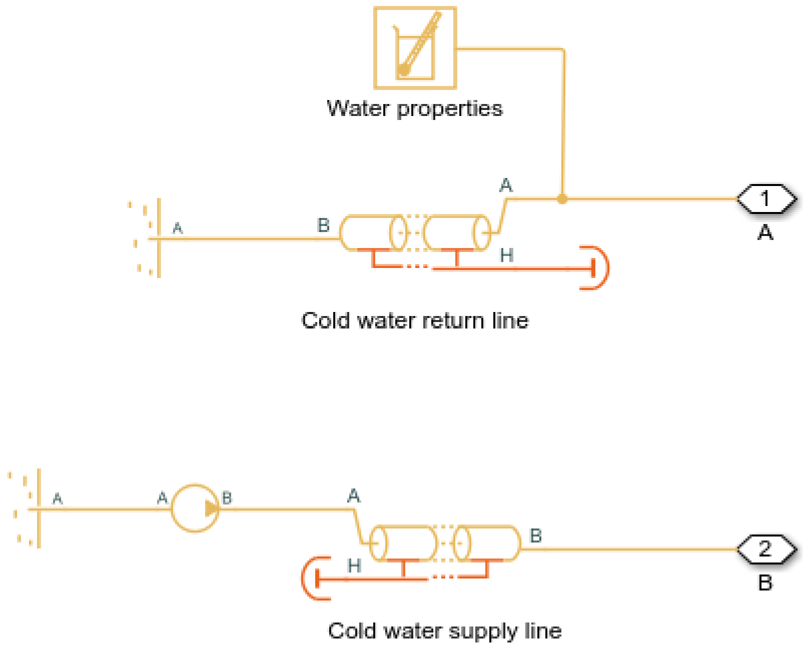
1. Plot fluid temperatures in heat exchangers in system (see code)
2. Plot fluid properties (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2015-2022 The MathWorks, Inc.

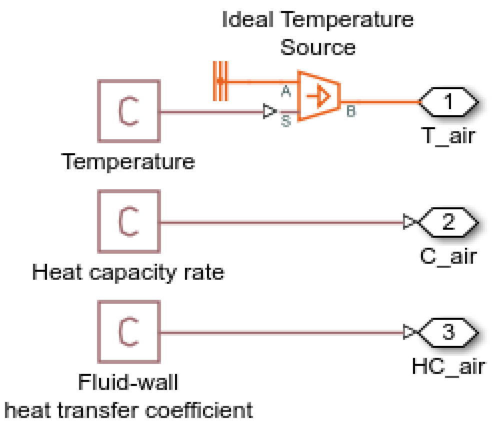
Hydraulic Oil Subsystem



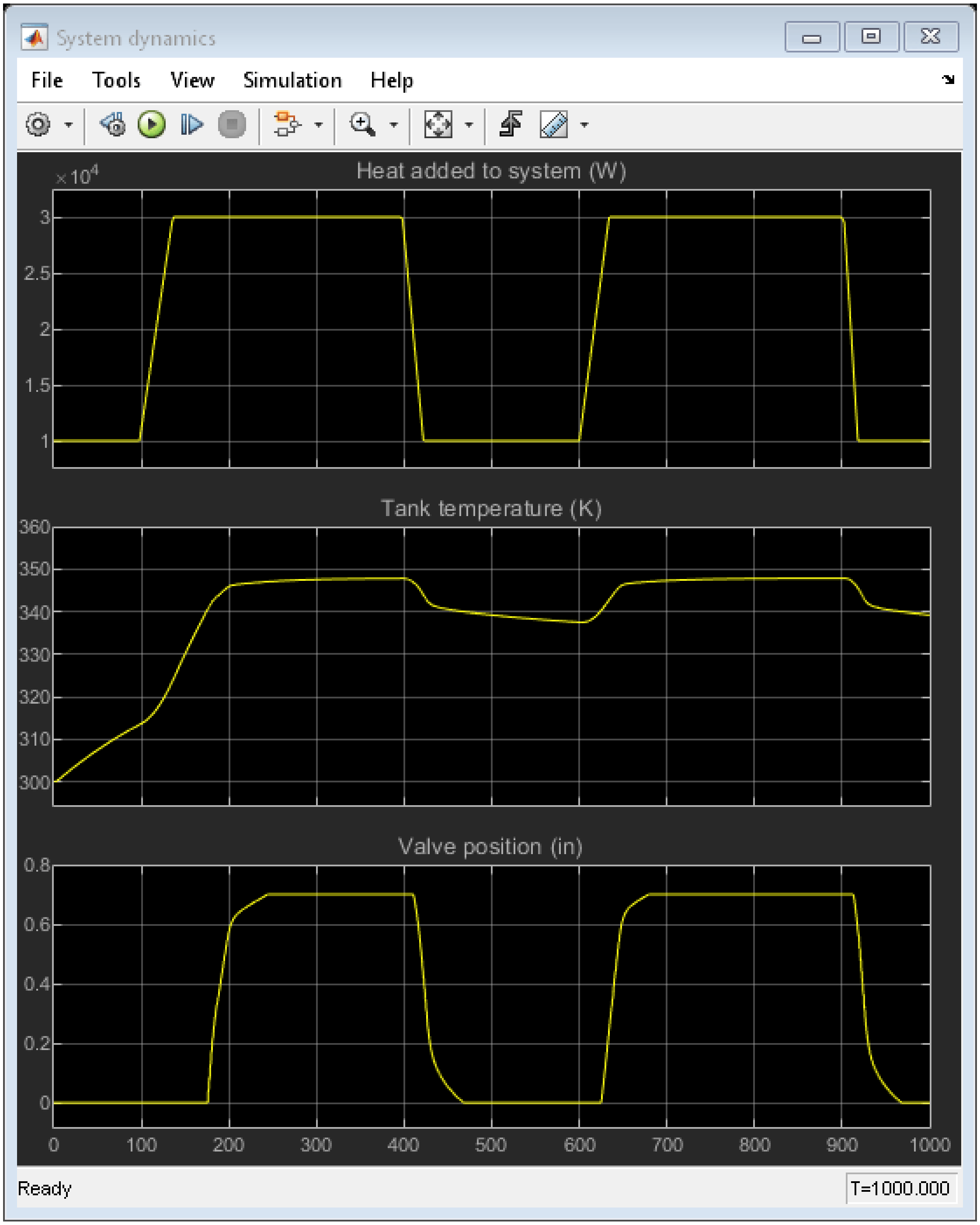
Cold Water Subsystem



Air Subsystem

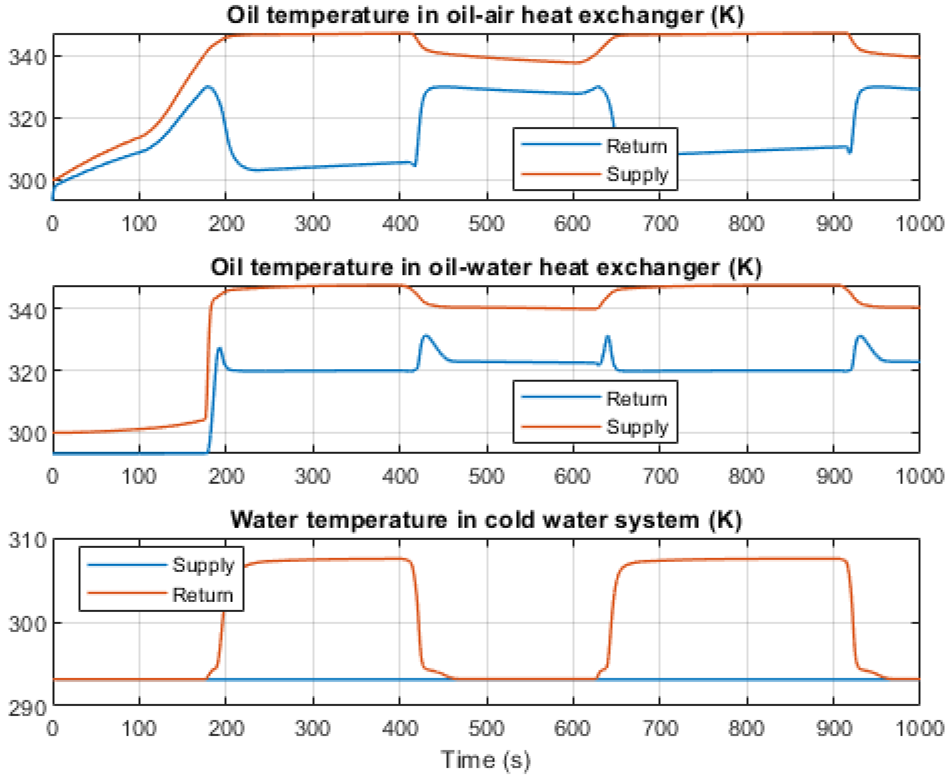


Simulation Results from Scopes



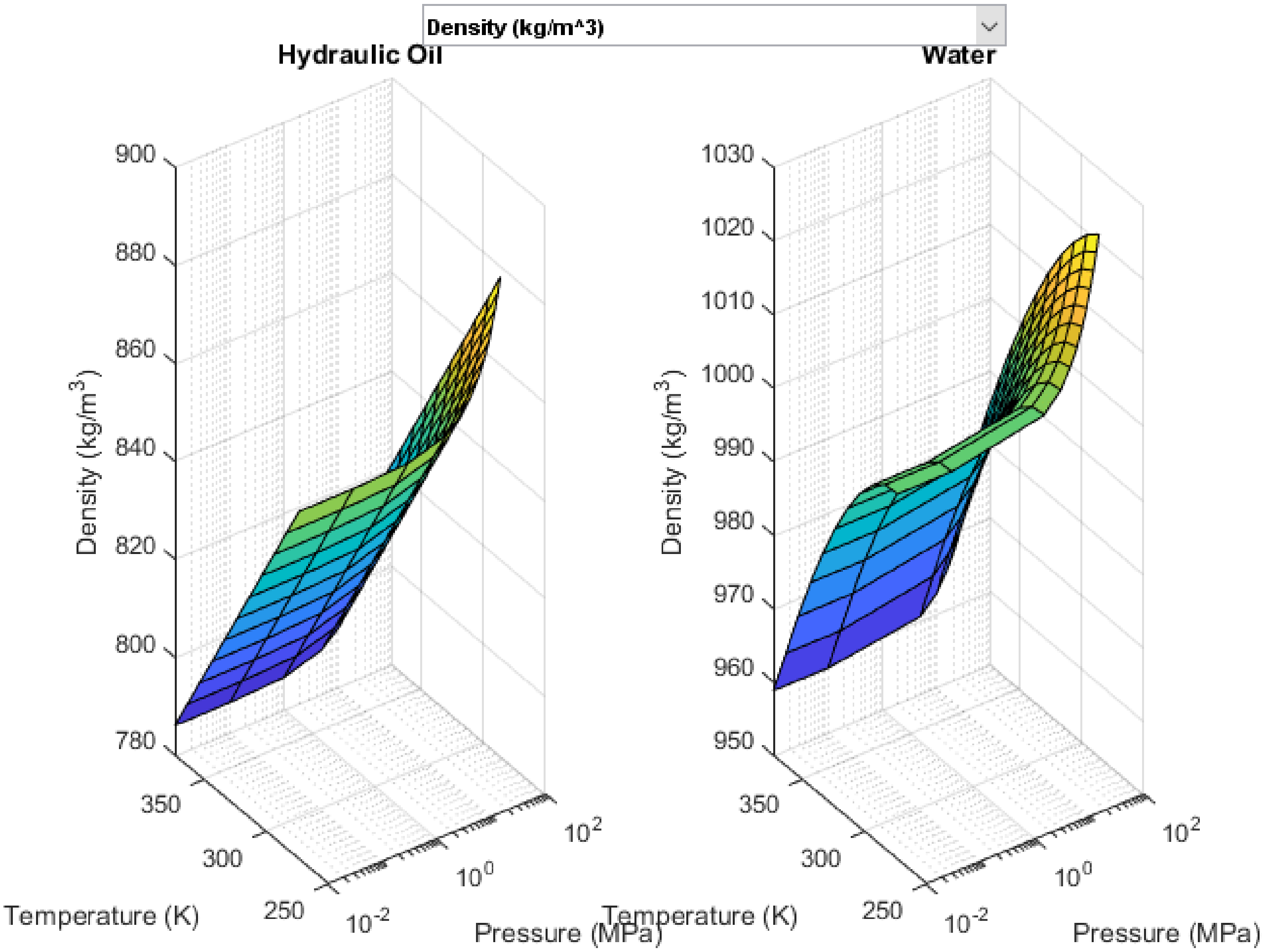
Simulation Results from Simscape Logging

These plots show the temperatures of hydraulic oil and water in the two heat exchangers. The oil-air heat exchanger primarily absorbs heat from the system. In the case when the oil temperature in the tank reaches a threshold, the 3-way directional valve diverts most of the flow to the oil-water heat exchanger to further lower the temperature.



Fluid Properties

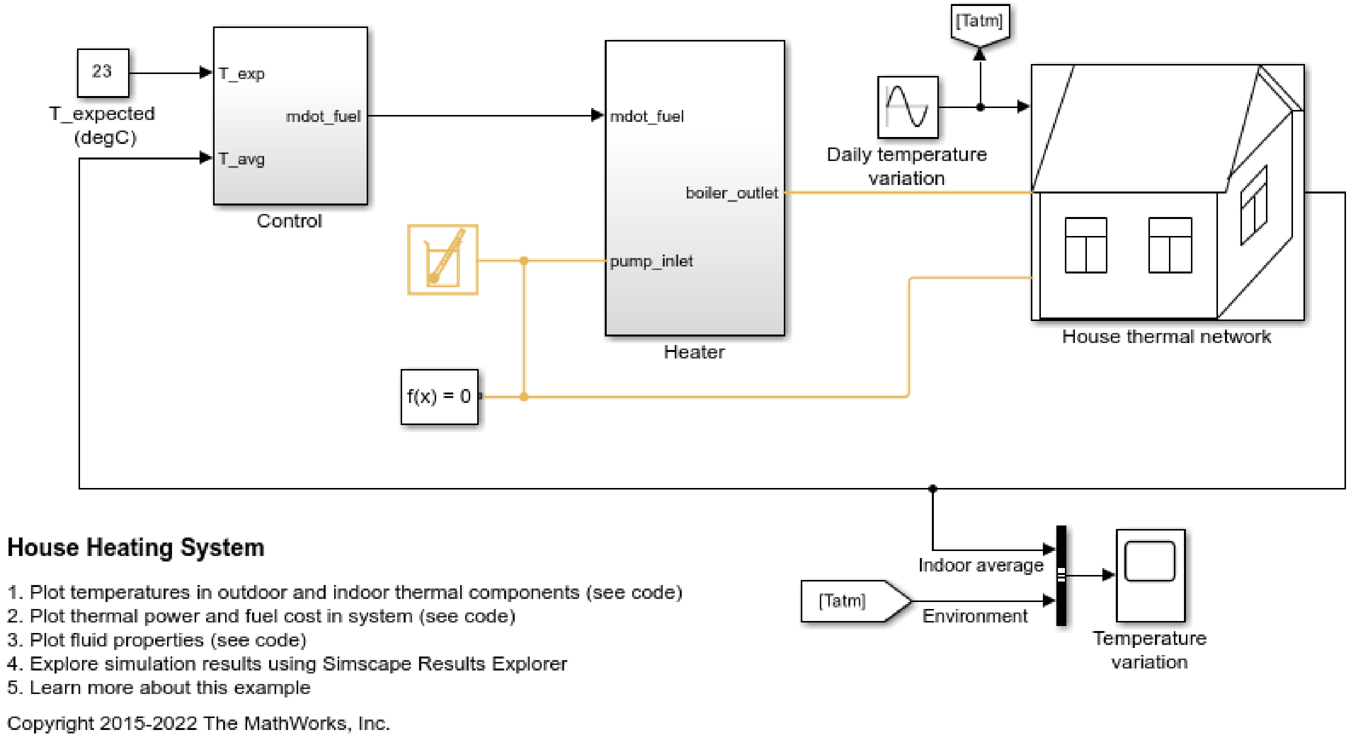
This plots show the fluid properties of hydraulic oil and water as functions of pressure and temperature.


Density (kg/m^3)
Hydraulic Oil
Water
Density (kg/m³)
900
880
860
840
820
800
780
1030
1020
1010
1000
990
980
970
960
950
Temperature (K)
350
300
250
Pressure (MPa)
10⁻²
10⁰
10²

House Heating System

This example shows how to model a simple house heating system. The model contains a heater, a controller, and a house structure with four radiators and four rooms. Each room exchanges heat with the environment through its exterior walls, roof, and windows. Each path is simulated as a combination of a thermal convection, thermal conduction, and the thermal mass. It is assumed that heat is not transferred internally between rooms. The heater consists of a furnace, a boiler, an accumulator, and a pump to circulate hot water in the system. The controller starts admitting fuel into the furnace if the overall average temperature of rooms falls below 21 degree C and it stops if the temperature exceeds 25 degree C. The simulation calculates the heating cost and indoor temperatures.

Model



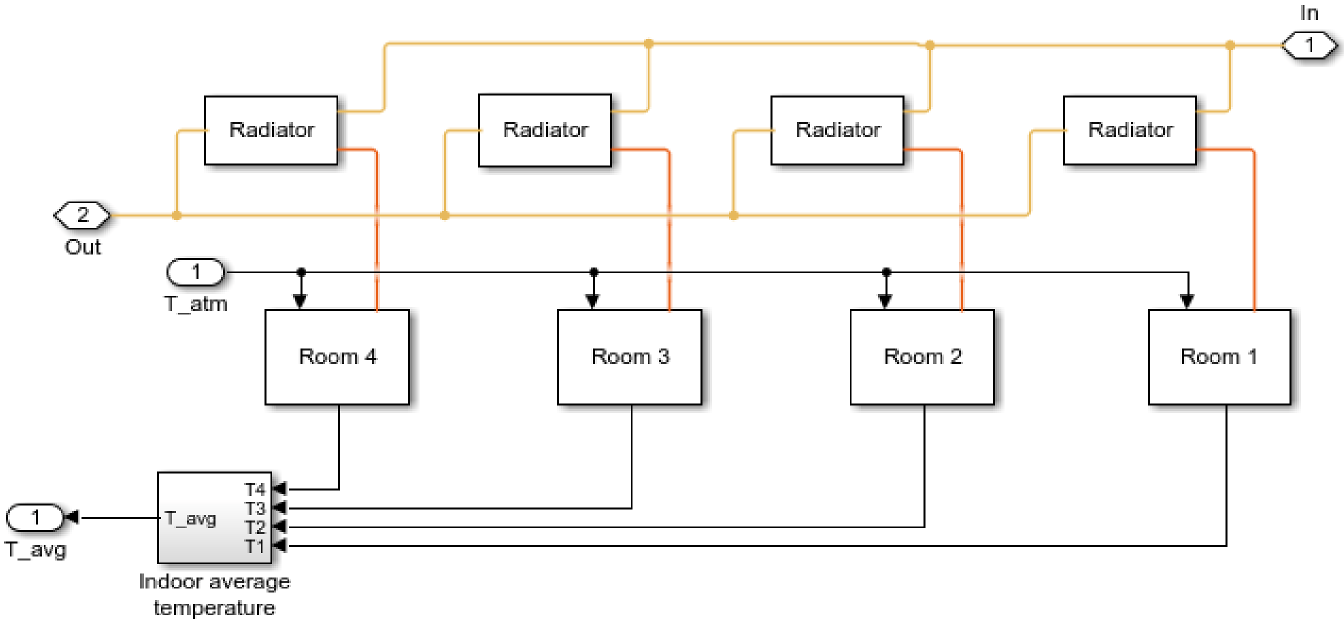
House Heating System

1. Plot temperatures in outdoor and indoor thermal components (see code)
2. Plot thermal power and fuel cost in system (see code)
3. Plot fluid properties (see code)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example

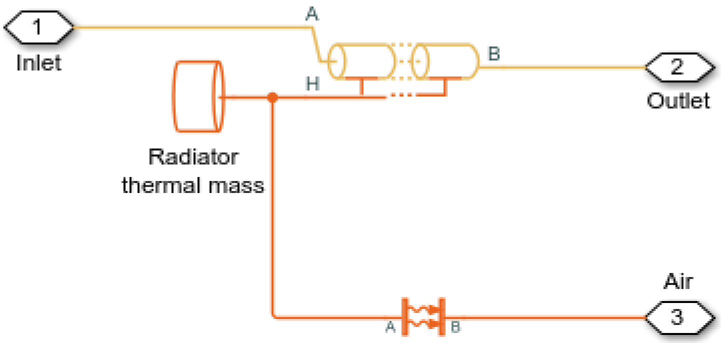
Copyright 2015-2022 The MathWorks, Inc.

House Thermal Network

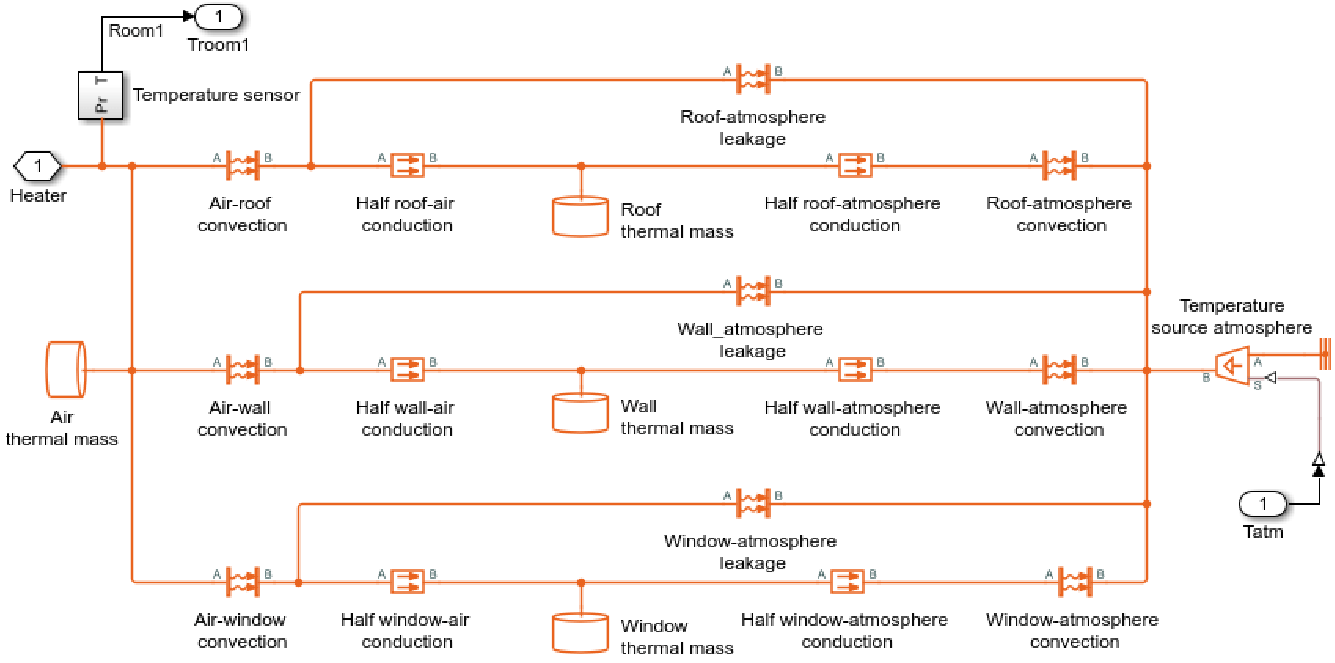
It consists of four radiators and four rooms.



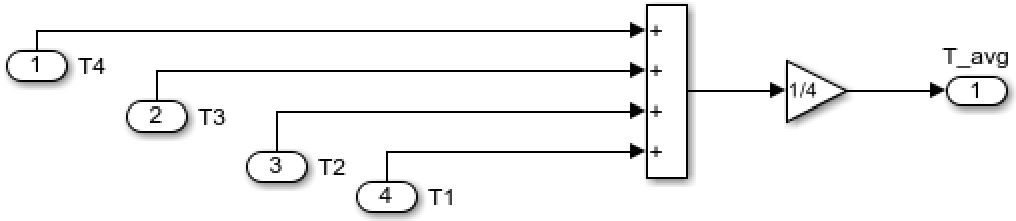
Radiator Subsystem



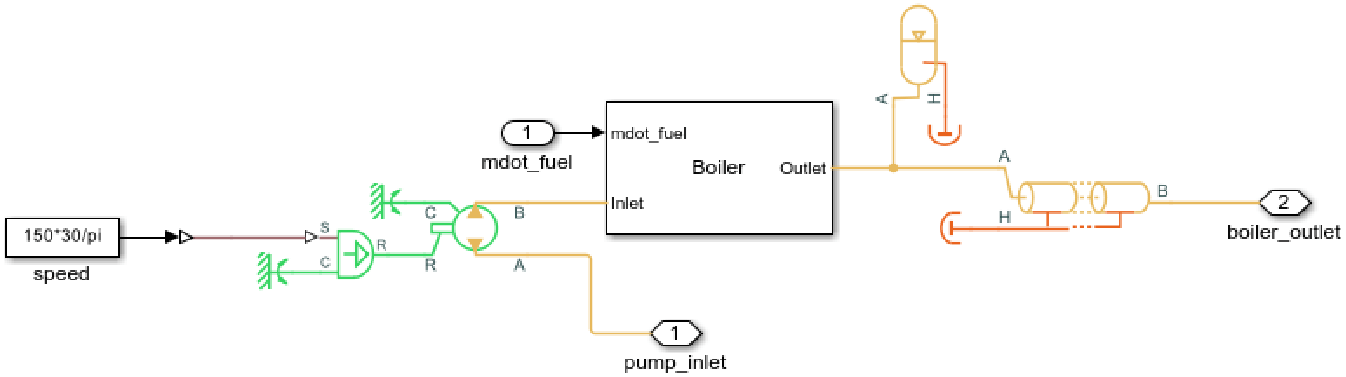
Room Subsystem



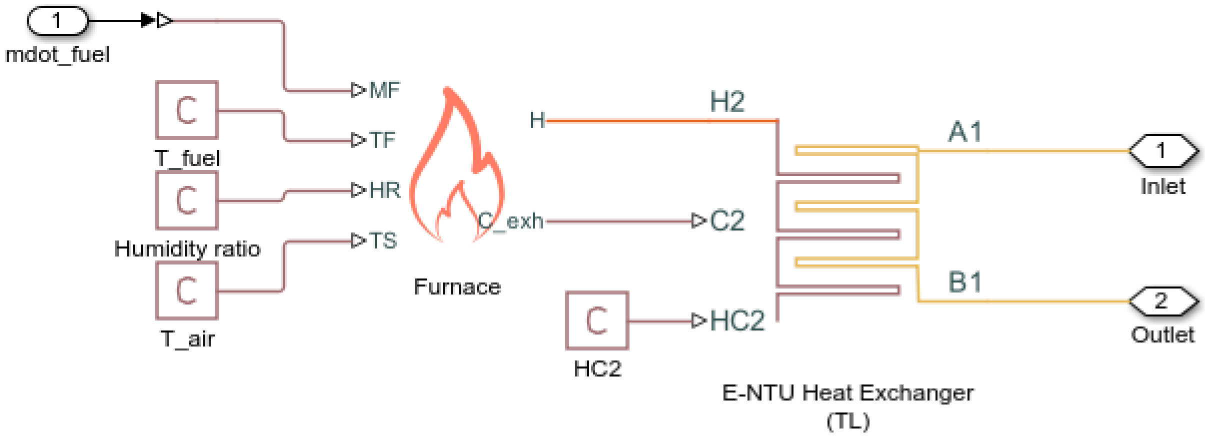
Indoor Average Temperature Subsystem



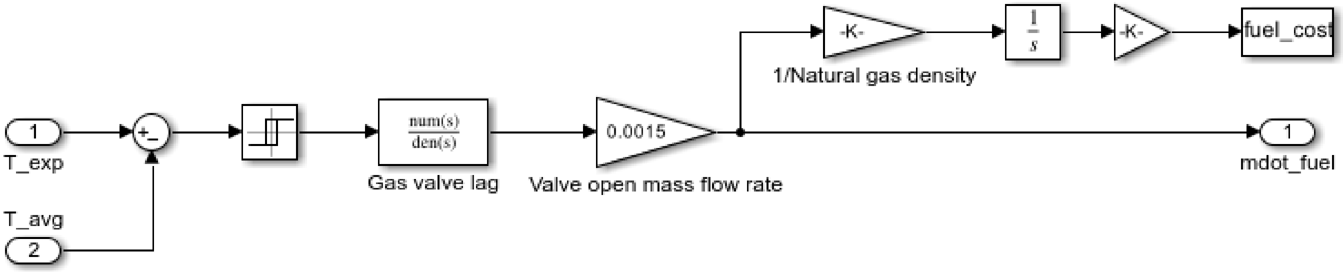
Heater Subsystem



Boiler Subsystem

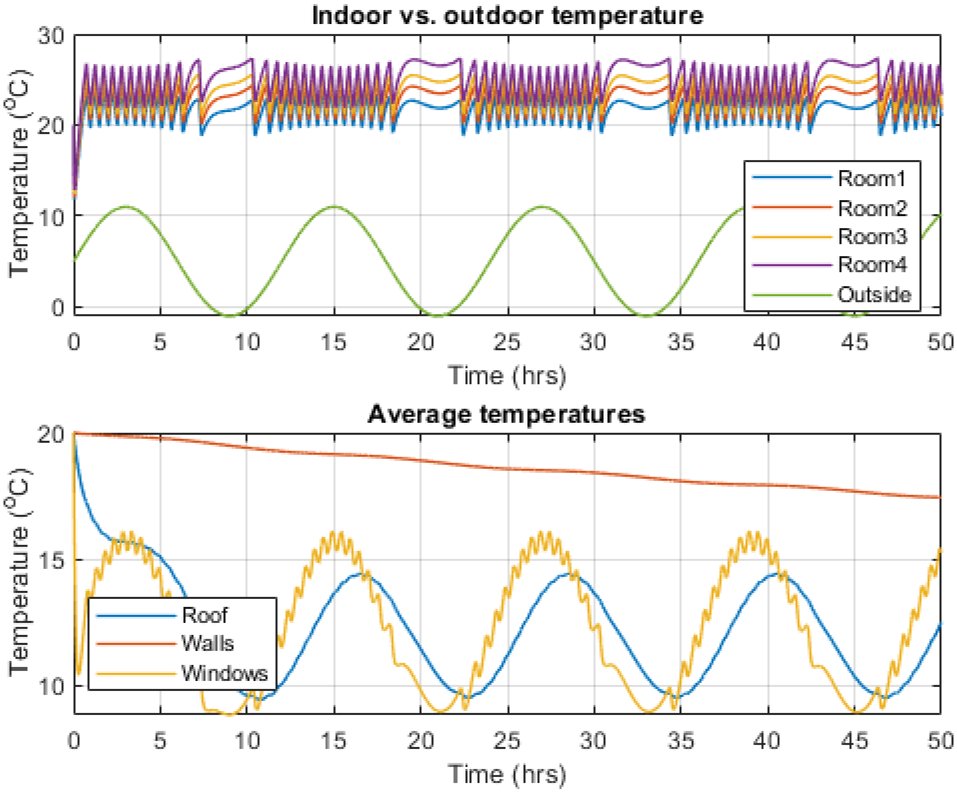


Control Subsystem

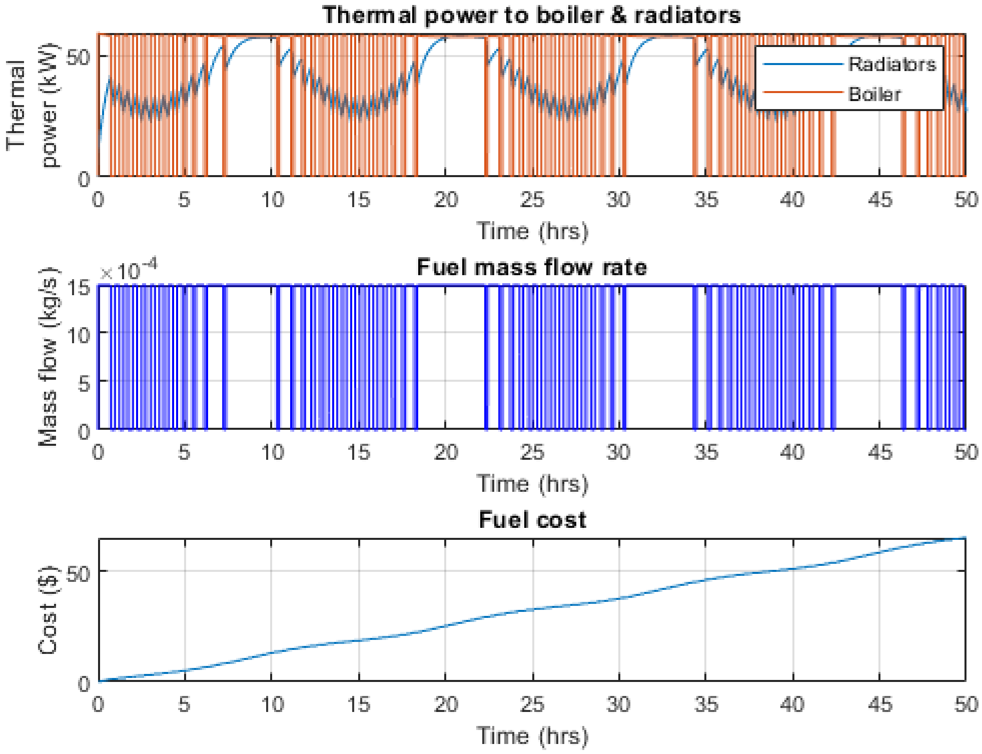


Simulation Results from Simscape Logging

The first plot shows the temperature variation of each room compared to the outside temperature fluctuations. The second plot shows the average overall temperatures of roof, walls, and windows.

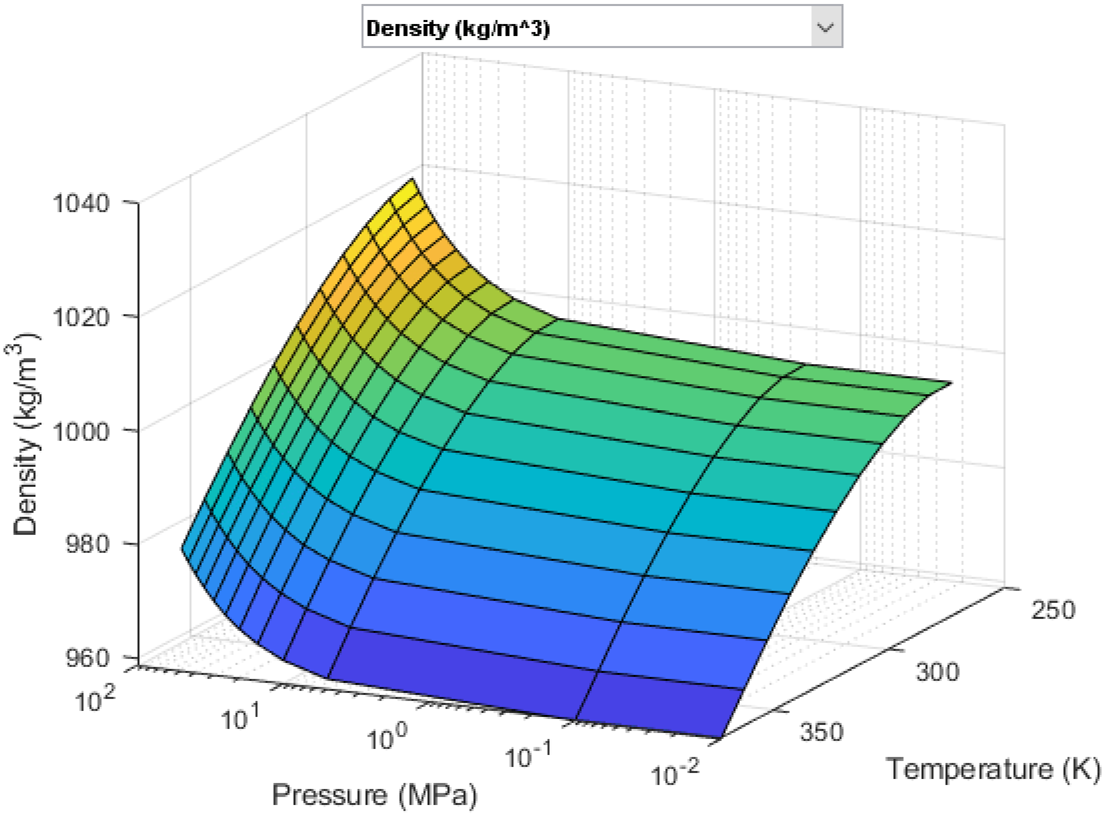


These plots show heat flow rate from furnace to the boiler and to radiators. The cost of fuel is calculated based on the fuel mass flow rate through the furnace. It is assumed that the fuel has the density of natural gas (~ 0.9 kg/m^3) and the cost of natural gas is $8.50 per thousand cubic feet (Mcf): $8.50 per 1000 cf.



Fluid Properties

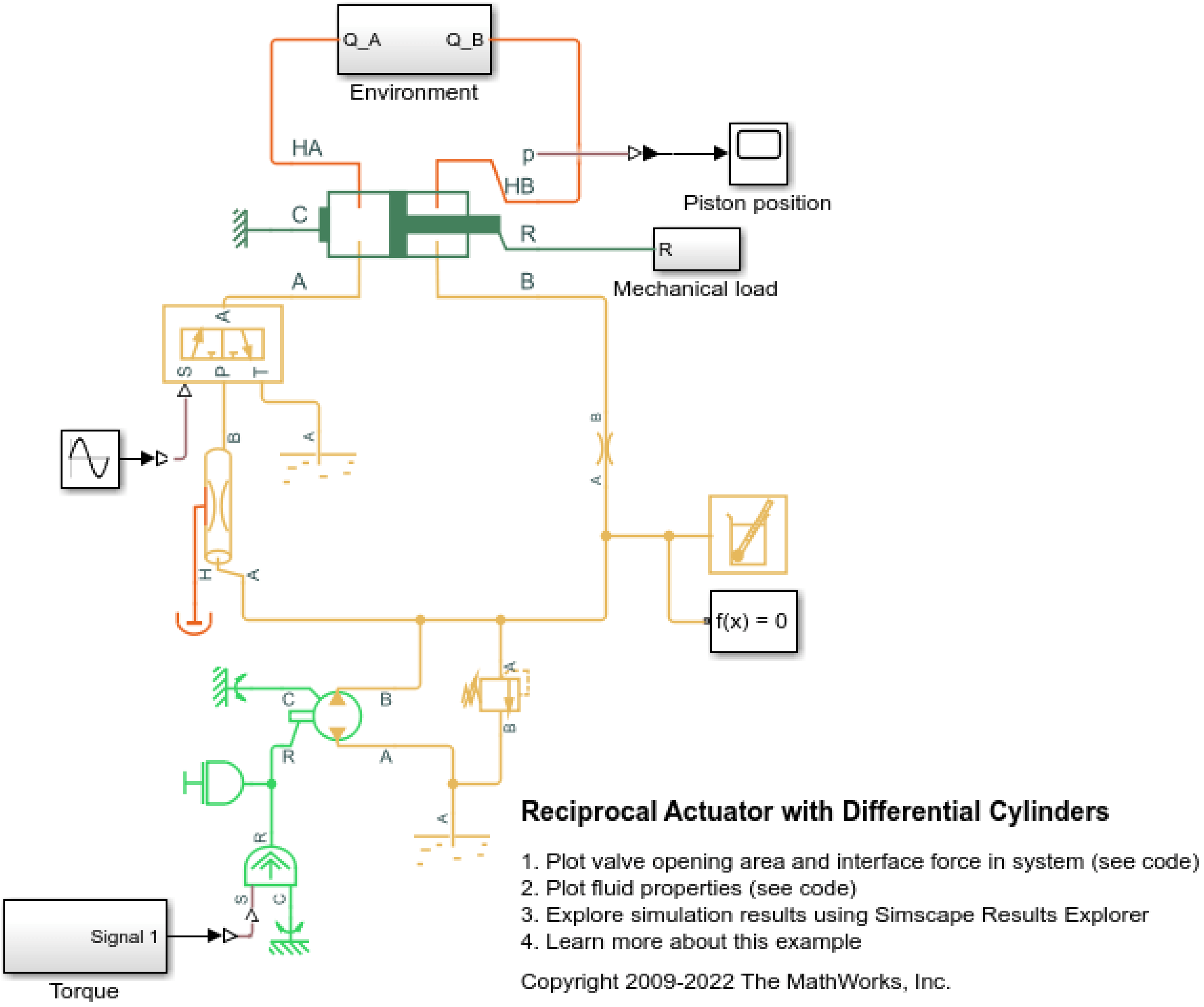
This figure plots the fluid properties of water as functions of pressure and temperature.


Density (kg/m^3)
1040
1020
1000
980
960
Density (kg/m^3)
10^2
10^1
10^0
10^-1
10^-2
Pressure (MPa)
250
300
350
Temperature (K)

Reciprocal Actuator with Differential Cylinders

This example shows a double-acting actuator with differential cylinders. The pump output is connected to cylinder B of the actuator while cylinder A of the actuator can be connected to either the pump or the reservoir through the 3-way directional valve. When cylinder A is connected to the pump, pressures at both cylinders become equal. Because of the larger effective piston area in cylinder A, the interface force in cylinder A is larger than that of in cylinder B which causes the piston to extend. When cylinder A is connected to the reservoir, the piston starts to retract. The 3-way directional valve is controlled by a sinusoidal signal to achieve repeating reciprocal motion in the actuator.

Model

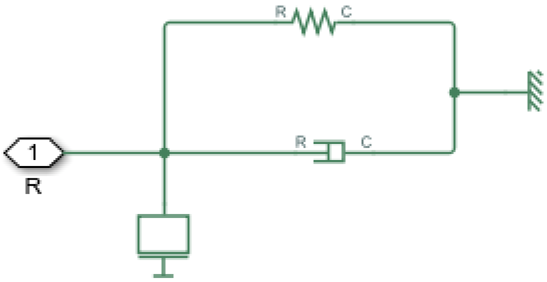


Reciprocal Actuator with Differential Cylinders

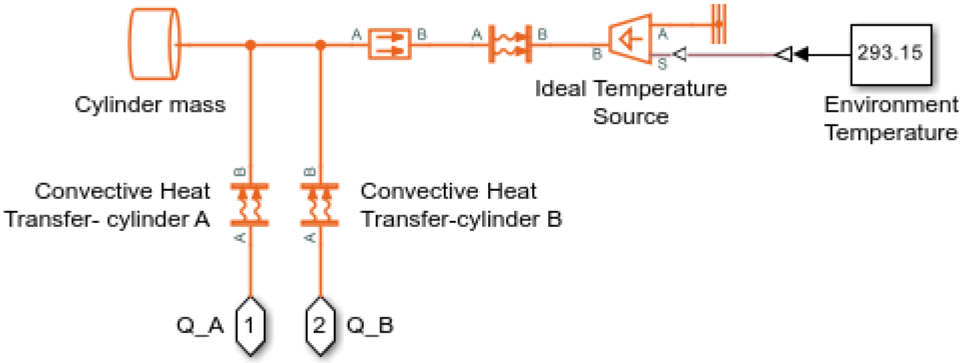
1. Plot valve opening area and interface force in system (see code)
2. Plot fluid properties (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

Copyright 2009-2022 The MathWorks, Inc.

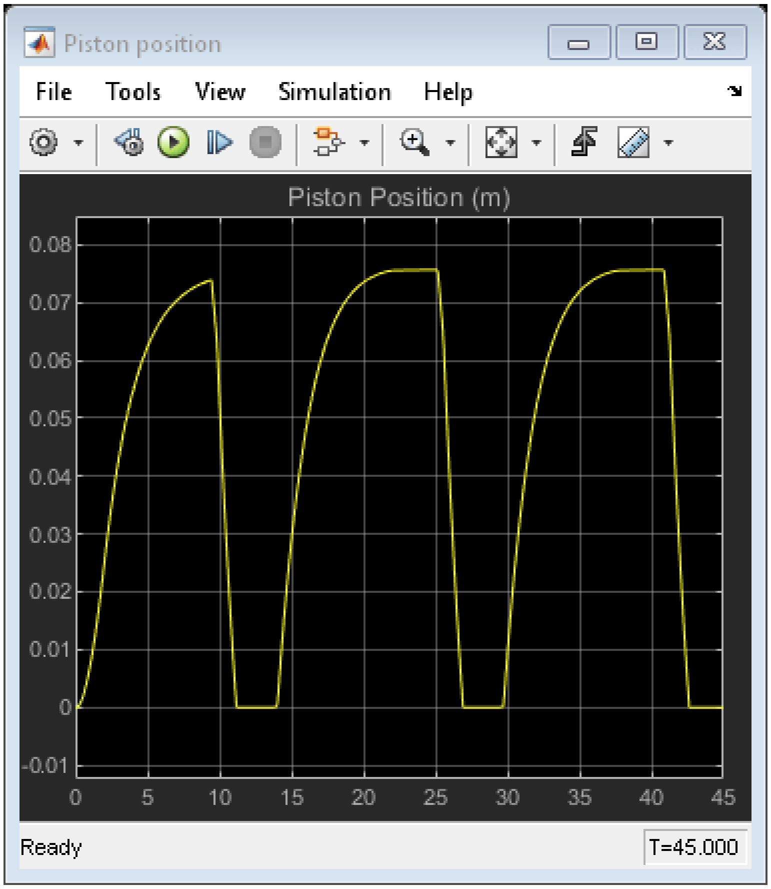
Mechanical Load Subsystem



Environment Subsystem

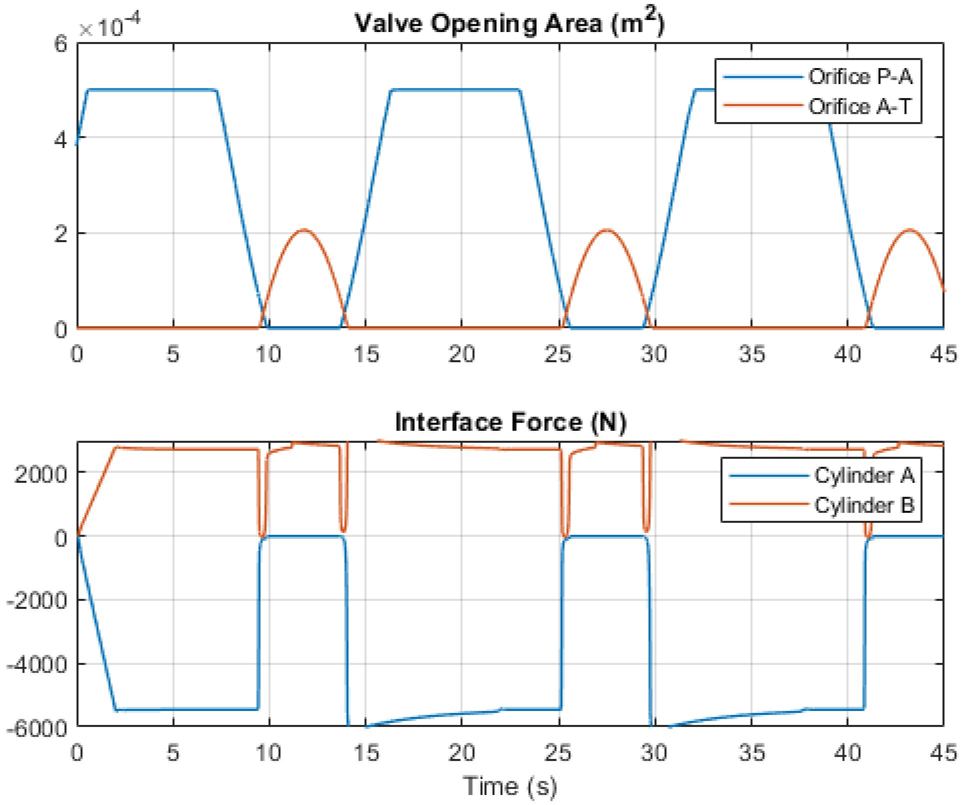


Simulation Results from Scopes

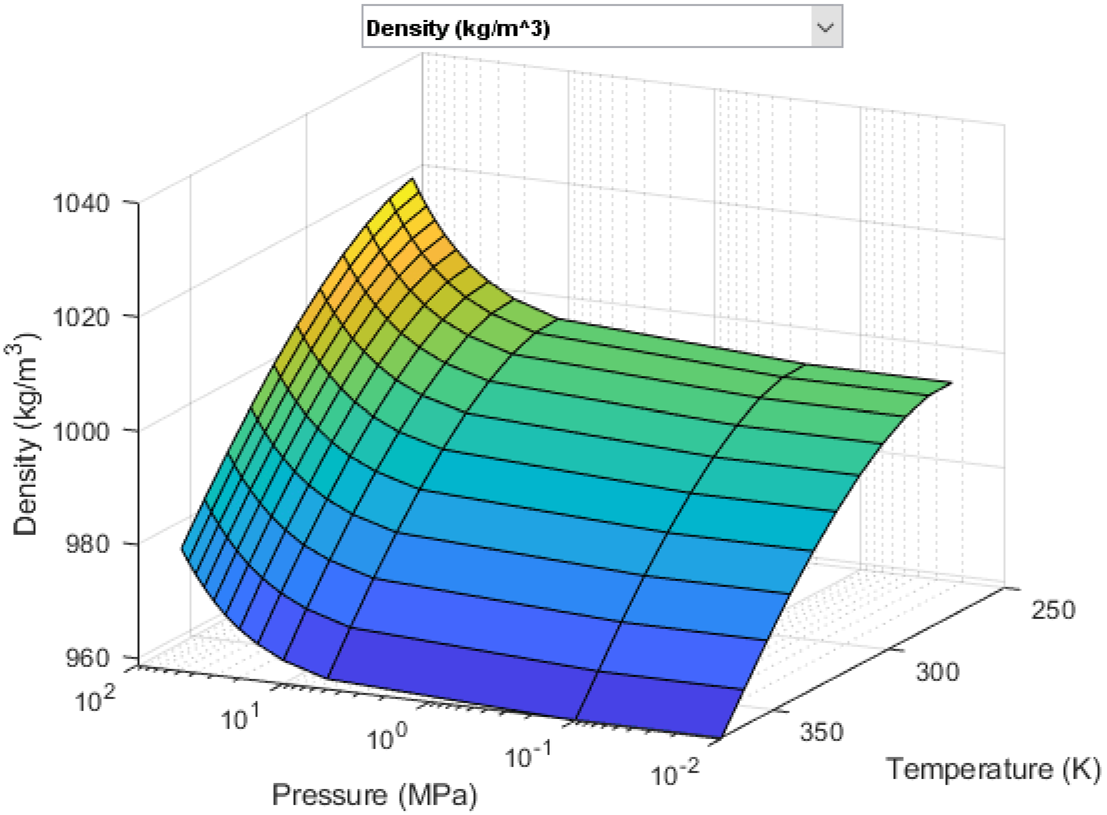


Simulation Results from Simscape Logging

These plots show the 3-way valve opening area and the interface forces generated on each side of the actuator during the reciprocal motion.


Valve Opening Area (m^2)
×10^-4
6
4
2
0
0 5 10 15 20 25 30 35 40 45
Orifice P-A
Orifice A-T
Interface Force (N)
2000
0
-2000
-4000
-6000
0 5 10 15 20 25 30 35 40 45
Time (s)
Cylinder A
Cylinder B

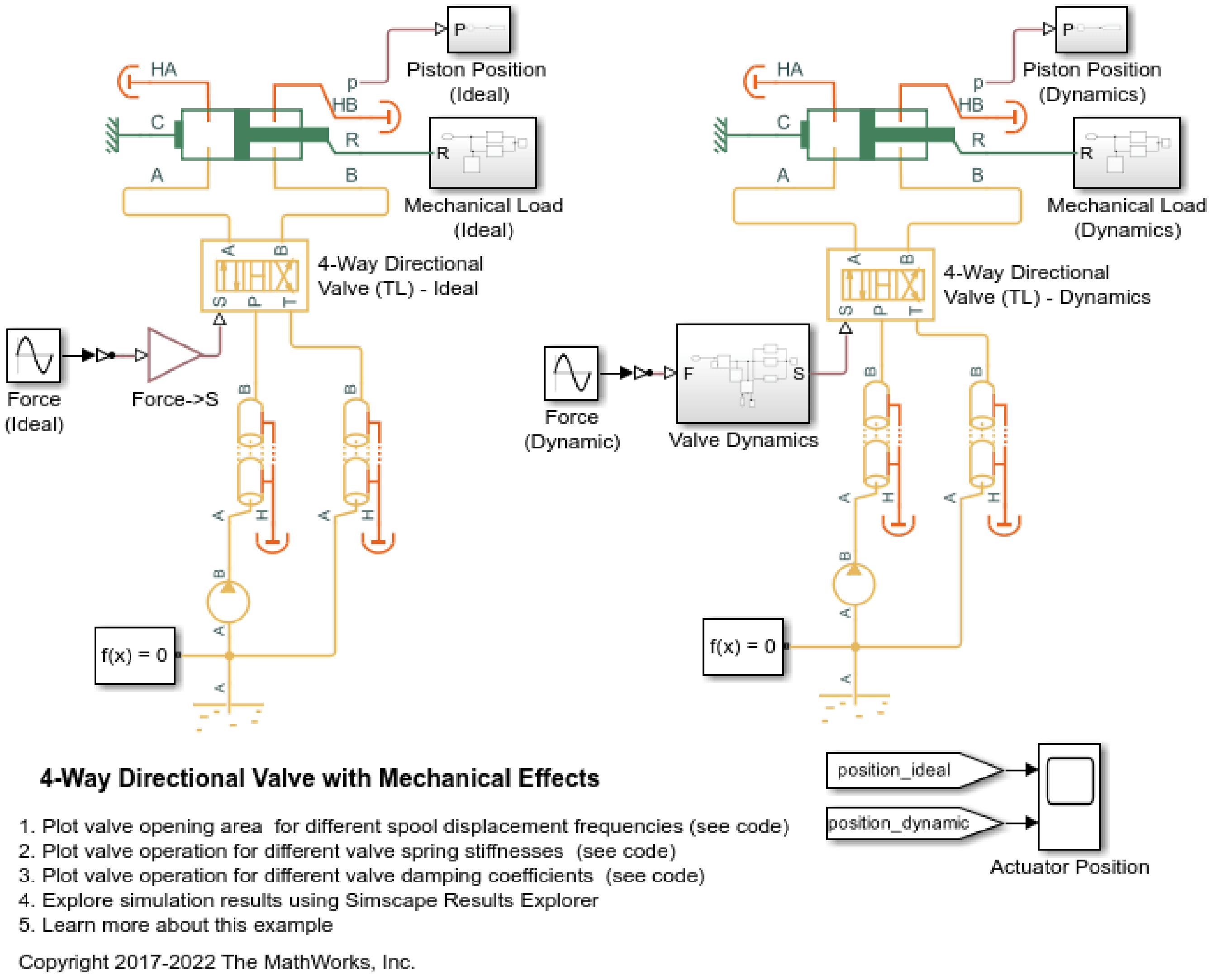
Fluid Properties



4-Way Directional Valve with Mechanical Effects

This example compares two 4-way directional valves, one with mechanical components to account for spool mass, valve spring and damper and one without (ideal valve). The input force to both valves is a sine wave. The operation of two valves are compared for different displacement frequencies, valve spring stiffness and damper coefficients.

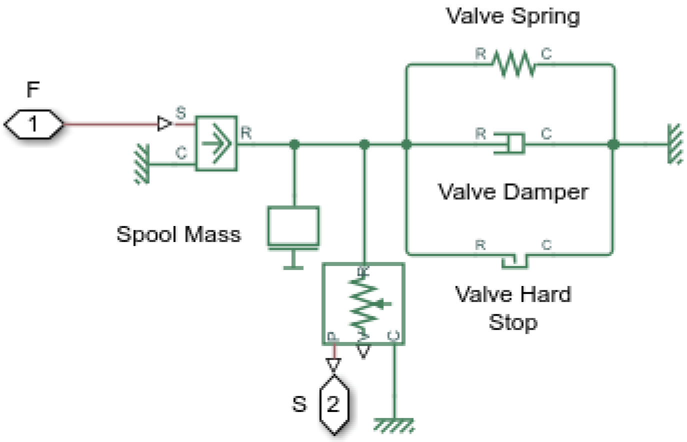
Model



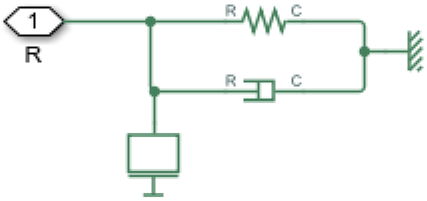
4-Way Directional Valve with Mechanical Effects

1. Plot valve opening area for different spool displacement frequencies (see code)
2. Plot valve operation for different valve spring stiffnesses (see code)
3. Plot valve operation for different valve damping coefficients (see code)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example

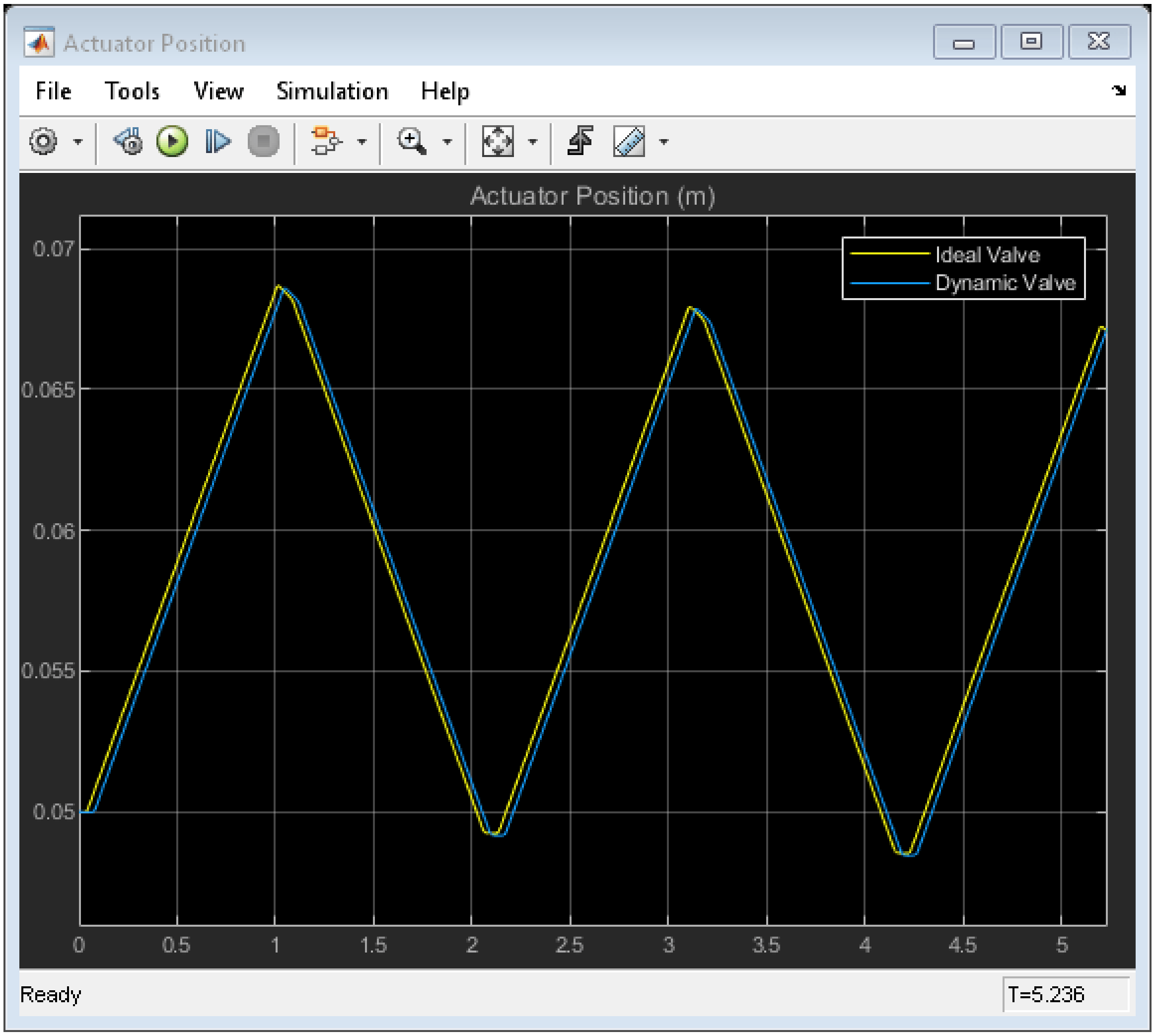
Valve Dynamics Subsystem



Mechanical Load Subsystem

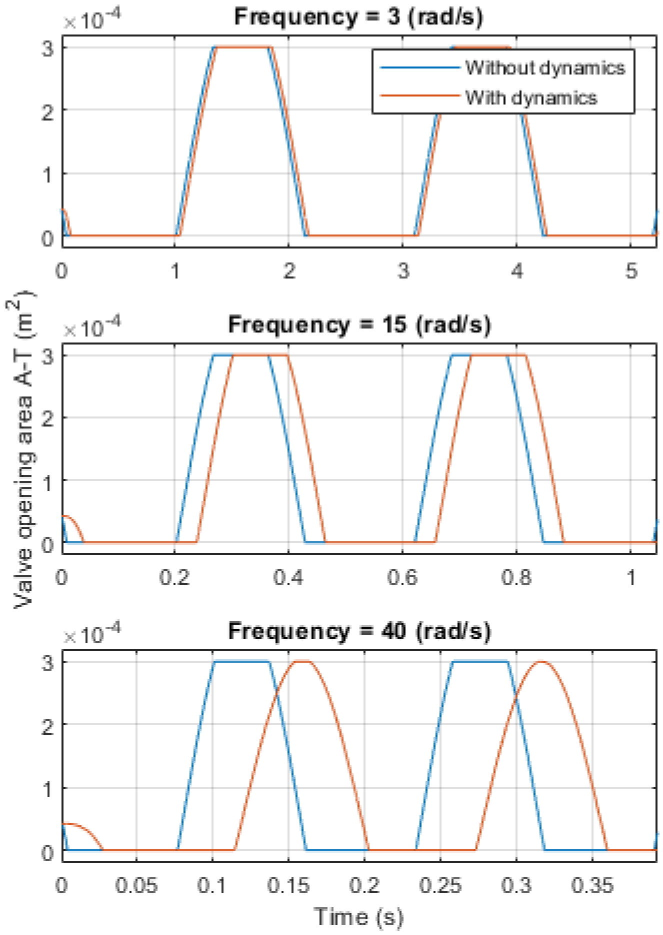


Simulation Results from Scopes

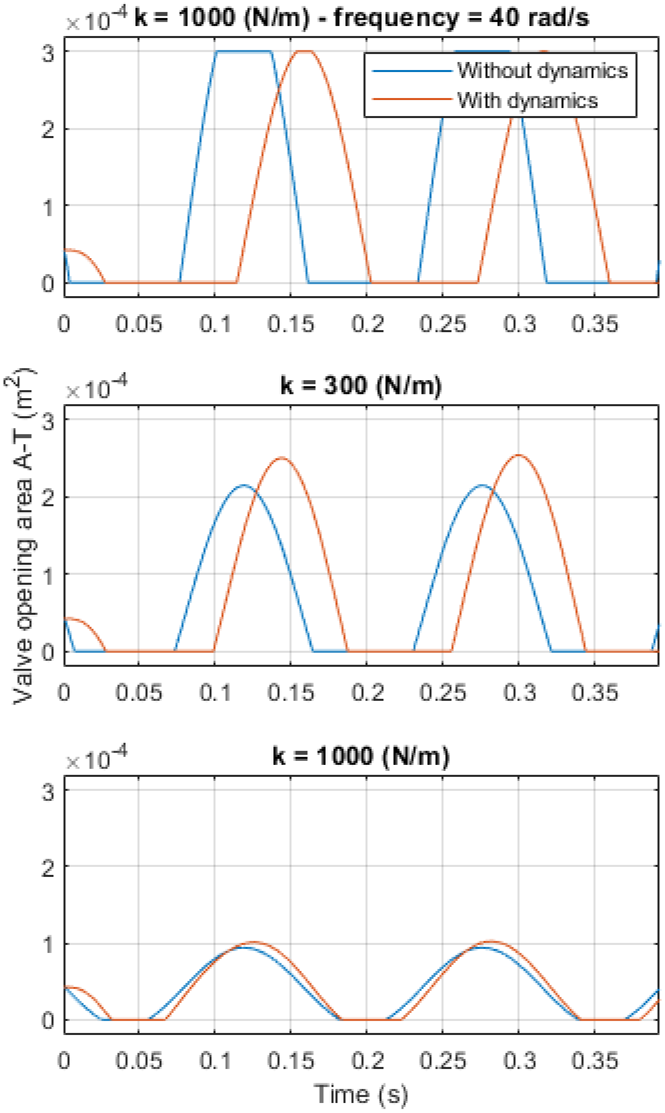


Simulation Results from Simscape Logging

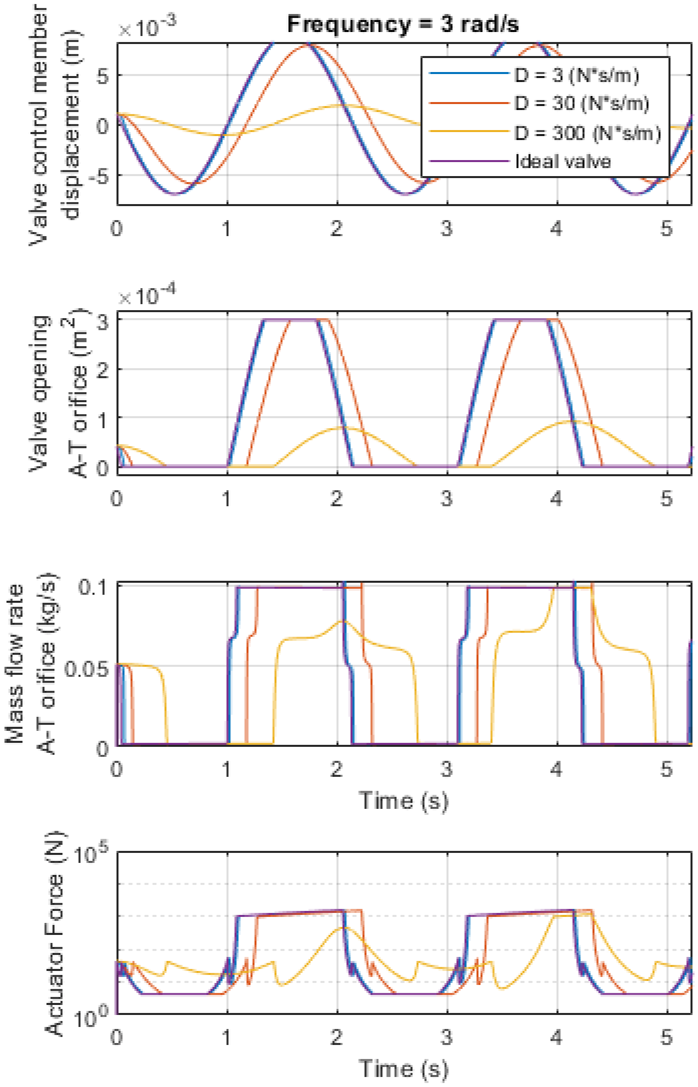
The plots below show the valve opening area (orifice A-T) for ideal valve and valve with dynamics for different input force frequencies.



The plots below show the valve opening area (orifice A-T) for ideal valve and valve with dynamics for different spring stiffnesses.



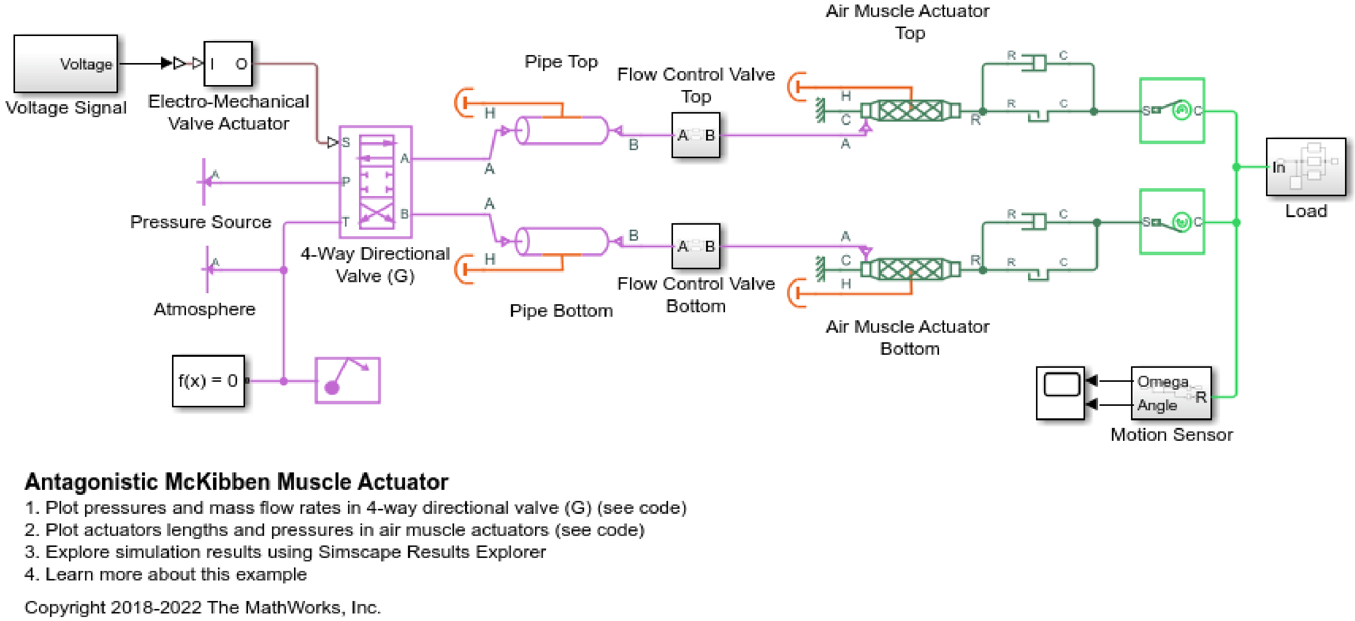
The plots below show the valve opening area (orifice A-T), control member displacement, and mass flow rate for ideal valve and valve with dynamics for different valve damping coefficients.


Frequency = 3 rad/s
×10^-3
Valve control member displacement (m)
5
0
-5
D = 3 (N*s/m)
D = 30 (N*s/m)
D = 300 (N*s/m)
Ideal valve
0
1
2
3
4
5
×10^-4
Valve opening A-T orifice (m^2)
3
2
1
0
Mass flow rate A-T orifice (kg/s)
0.1
0.05
0
Time (s)
Actuator Force (N)
10^5
10^0
Time (s)

Antagonistic McKibben Muscle Actuator

This demo shows a muscle actuation based on two air muscle actuators (or McKibben artificial muscles) in antagonistic connection. The air muscle actuators are connected to the opposite sides of a lever. The 4-way directional valve is controlled by an electro-mechanical valve actuator. In the 4-way directional way when the high-pressure path P-A and return line B-T are open, the top air muscle actuator contracts and forces the bottom air muscle actuator on the opposite side to extend. Similarly, as the high-pressure path P-B and return line A-T open, the bottom air muscle actuator starts to contract and forces the top air muscle actuator to extend. The oscillating motions of the muscles are converted into the angular rotation of the output load connected to the mechanical linkage modeled with the slider-cranks.

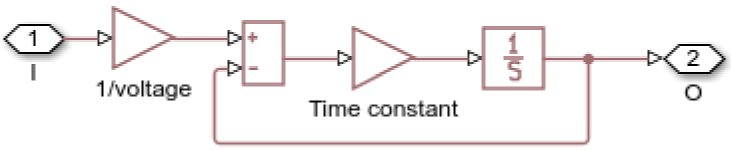
Model



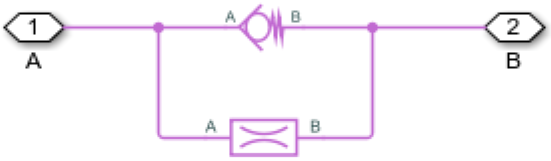
Antagonistic McKibben Muscle Actuator

1. Plot pressures and mass flow rates in 4-way directional valve (G) (see code)
2. Plot actuators lengths and pressures in air muscle actuators (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

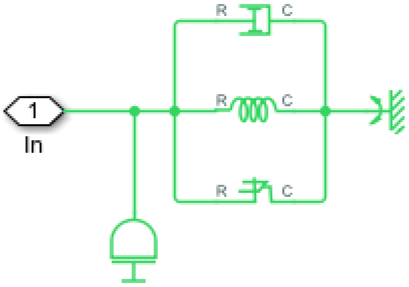
Electro-Mechanical Valve Actuator Subsystem



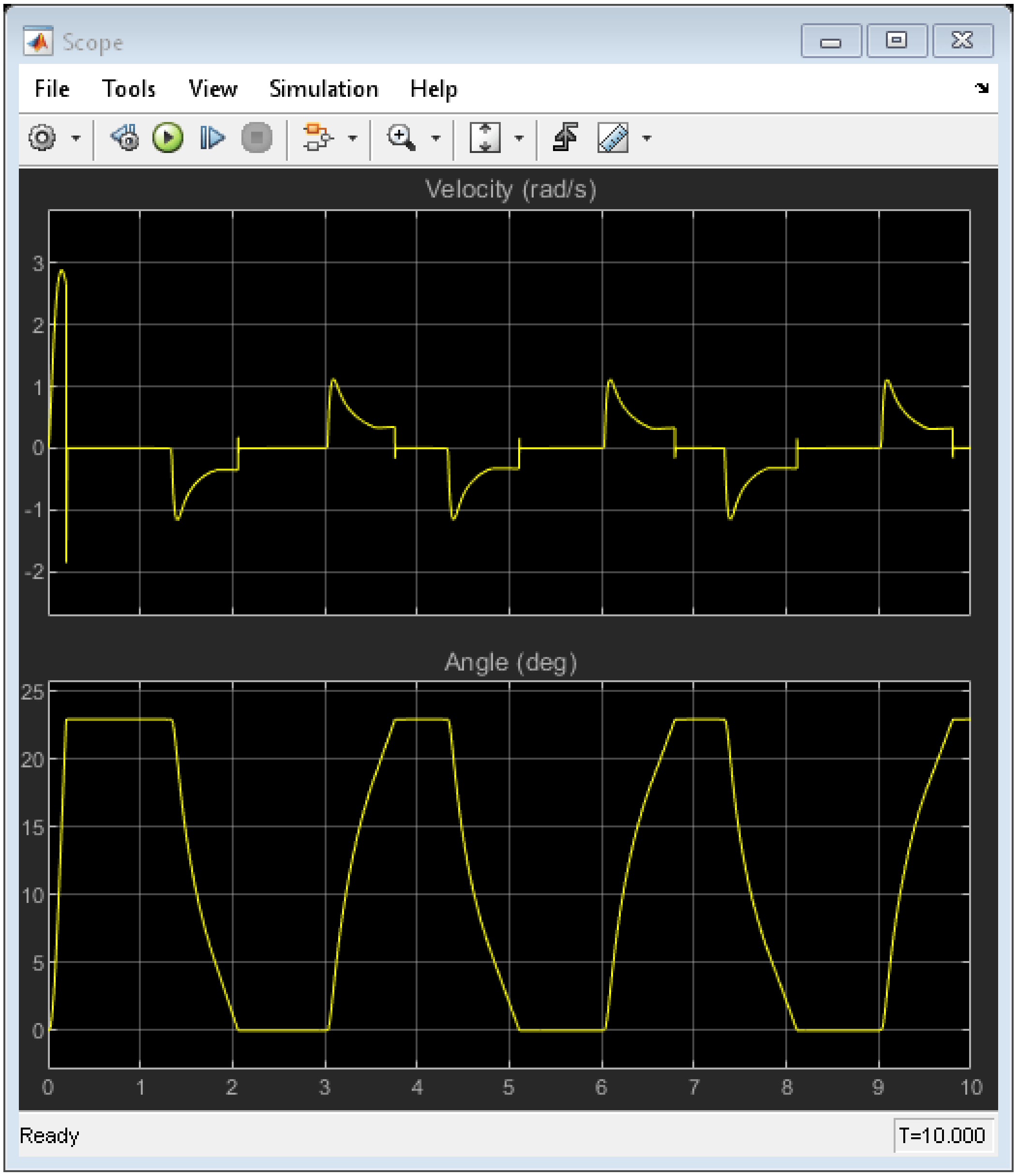
Flow Control Valve Subsystems



Load Subsystem

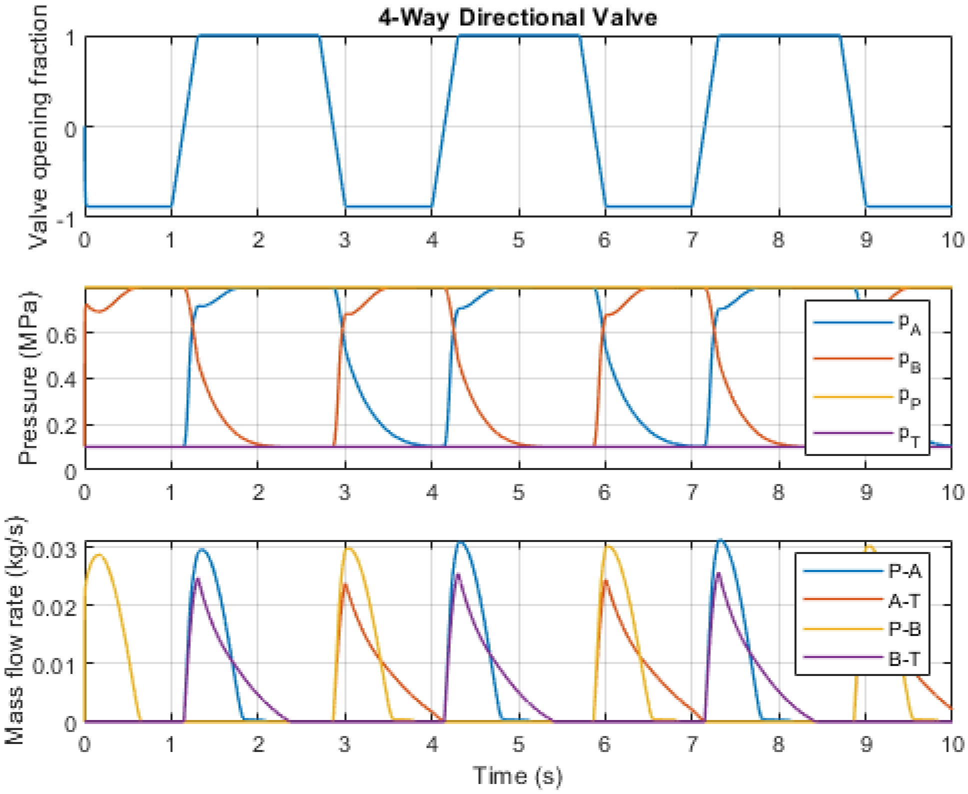


Simulation Results from Scopes

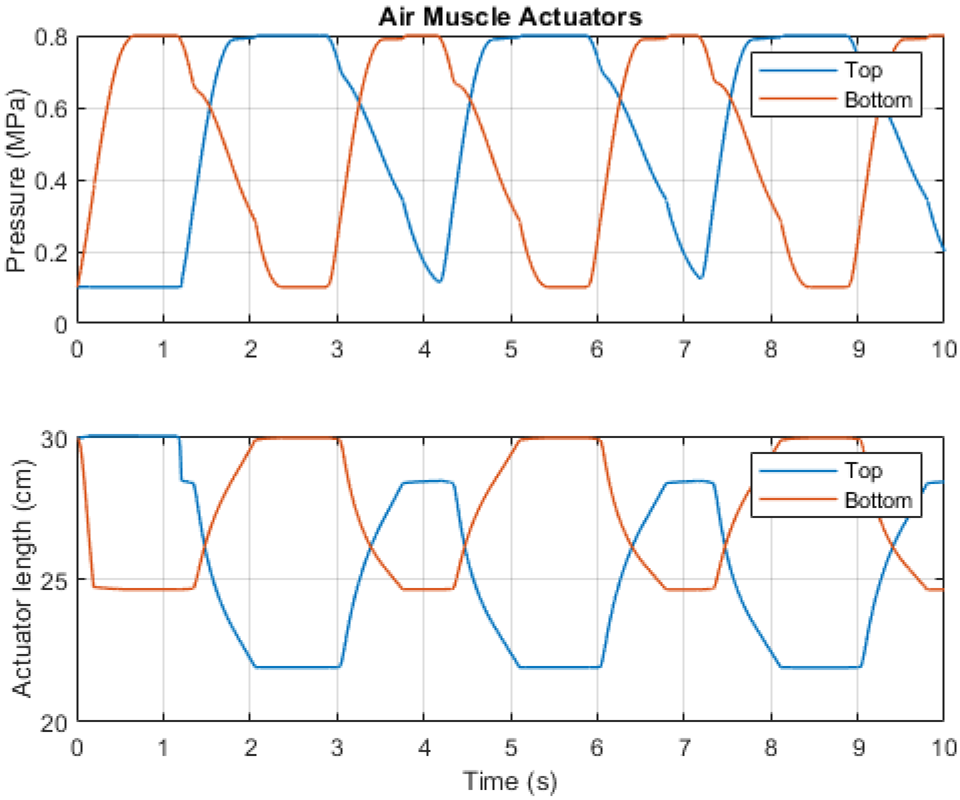


Simulation Results from Simscape Logging

The plots below show the pressures across and mass flow rates through the 4-Way Directional Valve (G) for a given voltage signal.



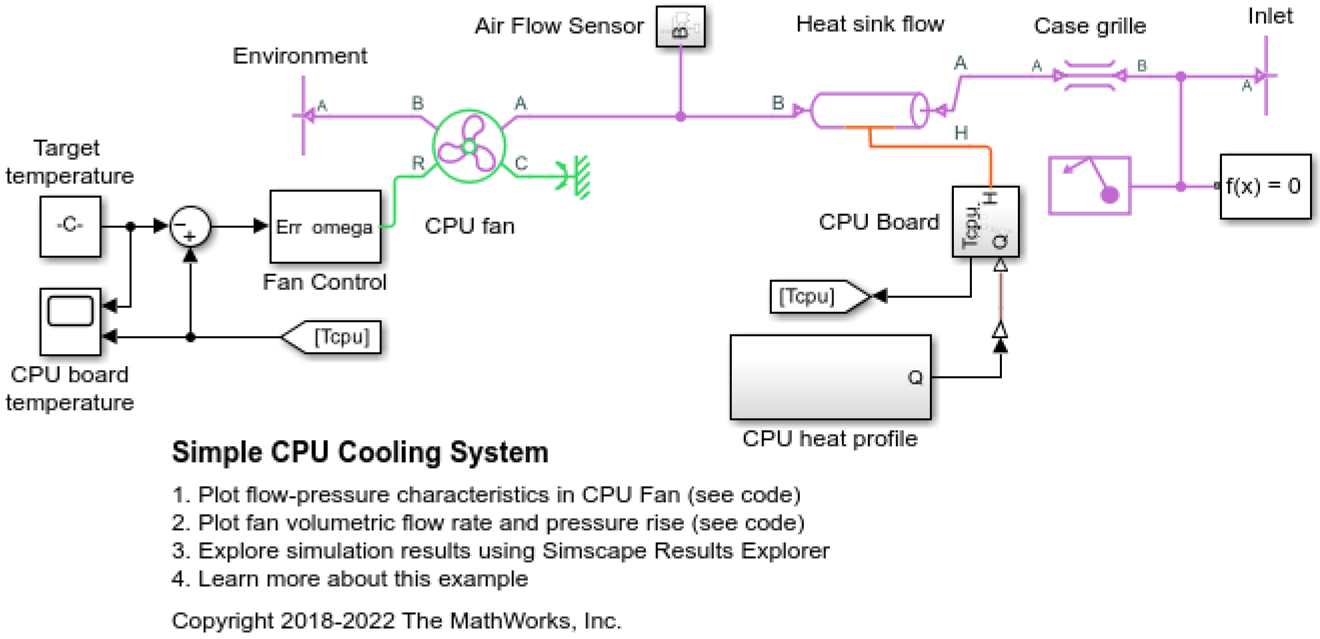
The plot below shows the pressure and mass flow rates in air muscle actuators.


Air Muscle Actuators
Pressure (MPa)
0
0.2
0.4
0.6
0.8
0
1
2
3
4
5
6
7
8
9
10
Top
Bottom
Actuator length (cm)
20
25
30
0
1
2
3
4
5
6
7
8
9
10
Time (s)
Top
Bottom

Simple CPU Cooling System

This example shows a simple CPU cooling system consists of a heat sink, a CPU fan, and fan controllers. The heat generated by the CPU is transferred to the heat sink by conduction and it is dissipated to the cooling air by forced convection mechanism. The heat sink is a parallel plate fin with rectangular fins between the plate fins. The CPU fan moves air across the heat sink by drawing air into the case grille from the outside and ejecting warm air from inside. The fan controller unit includes three control levels, representing a 3-speed switch, according to the CPU temperature. For lower CPU temperature, the fan speed is decreased and for higher temperature it is increased.

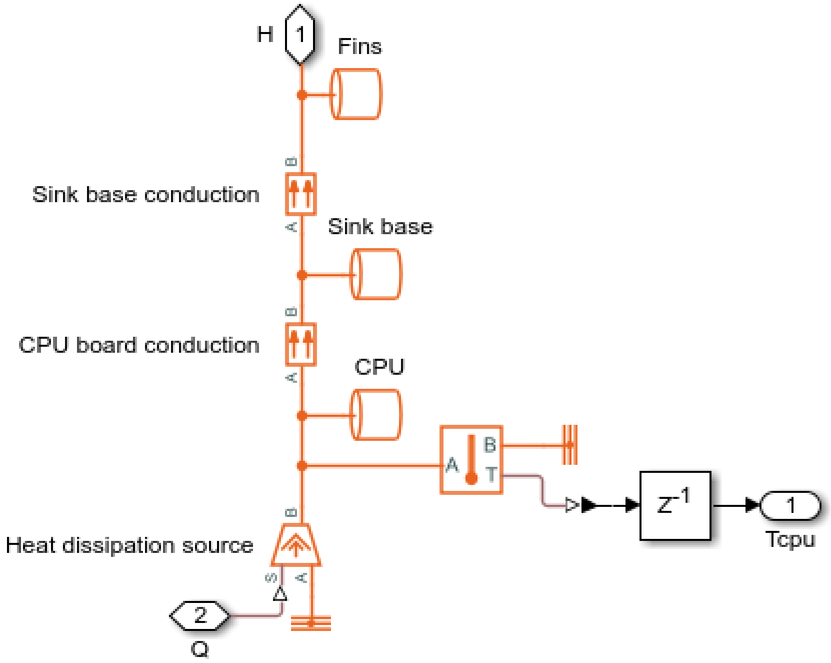
Model



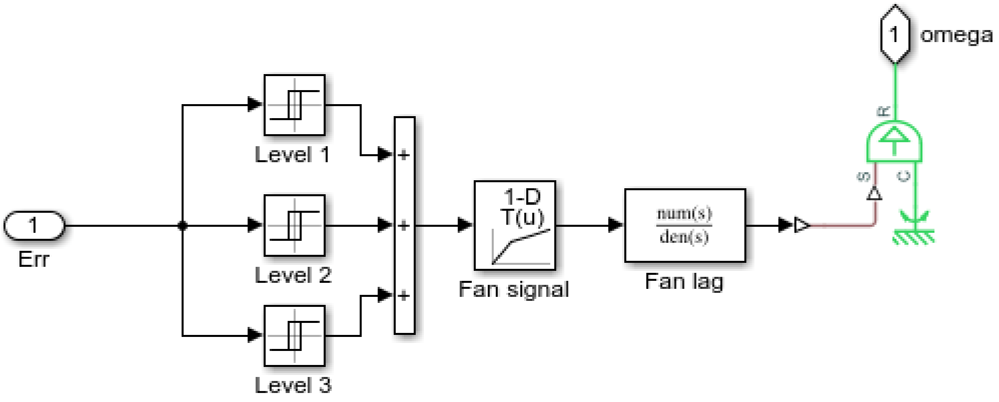
Simple CPU Cooling System

1. Plot flow-pressure characteristics in CPU Fan (see code)
2. Plot fan volumetric flow rate and pressure rise (see code)
3. Explore simulation results using Simscape Results Explorer
4. Learn more about this example

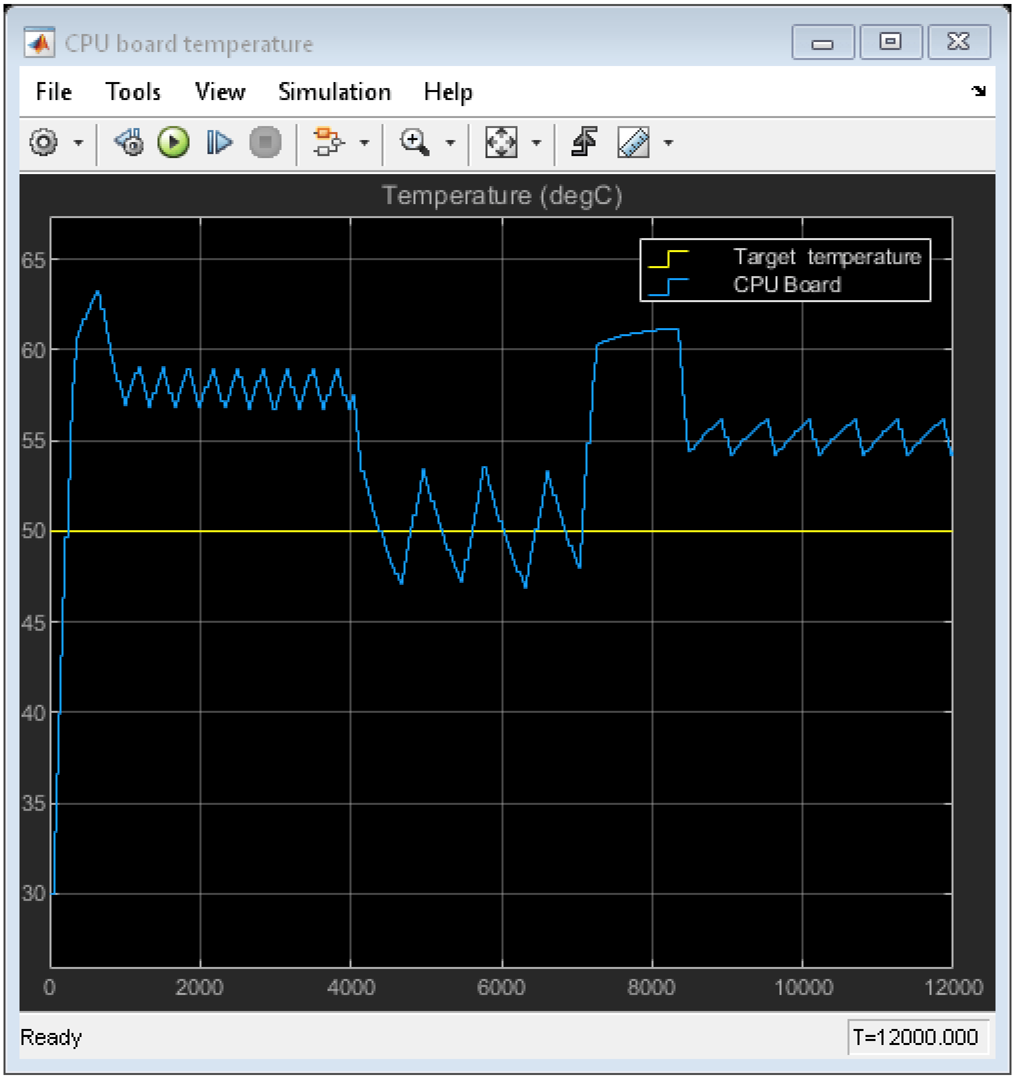
CPU Board Subsystem



Fan Control Subsystem

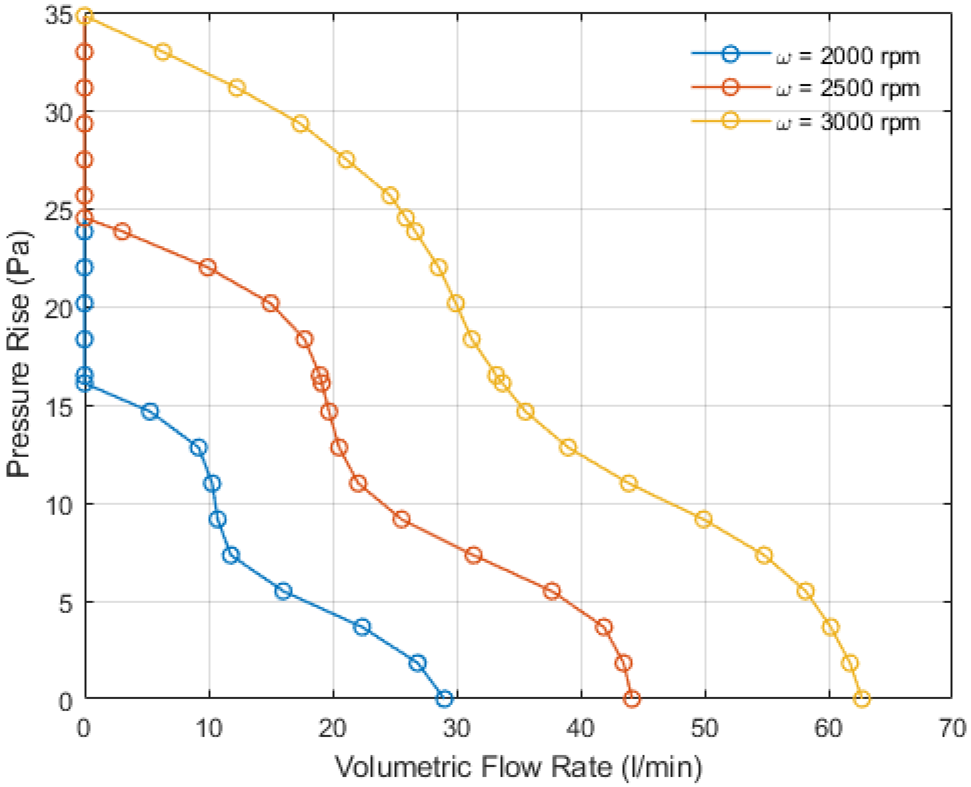


Simulation Results from Scopes

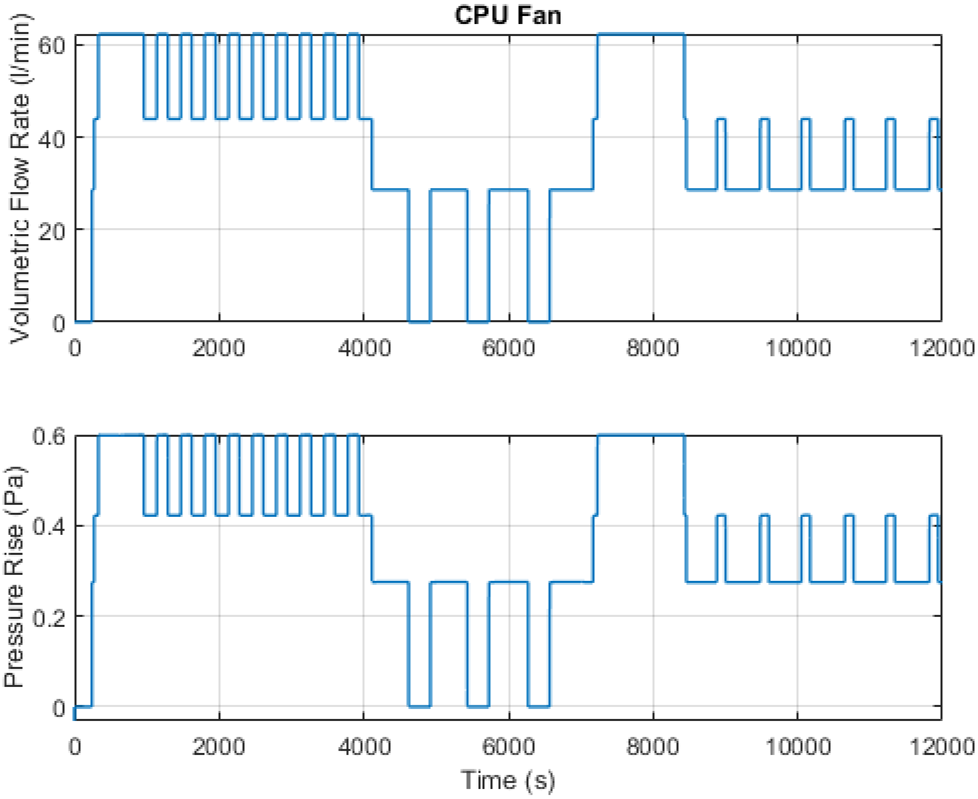


Simulation Results from Simscape Logging

The plot below shows the volumetric flow rate - pressure rise characteristics of CPU fan.



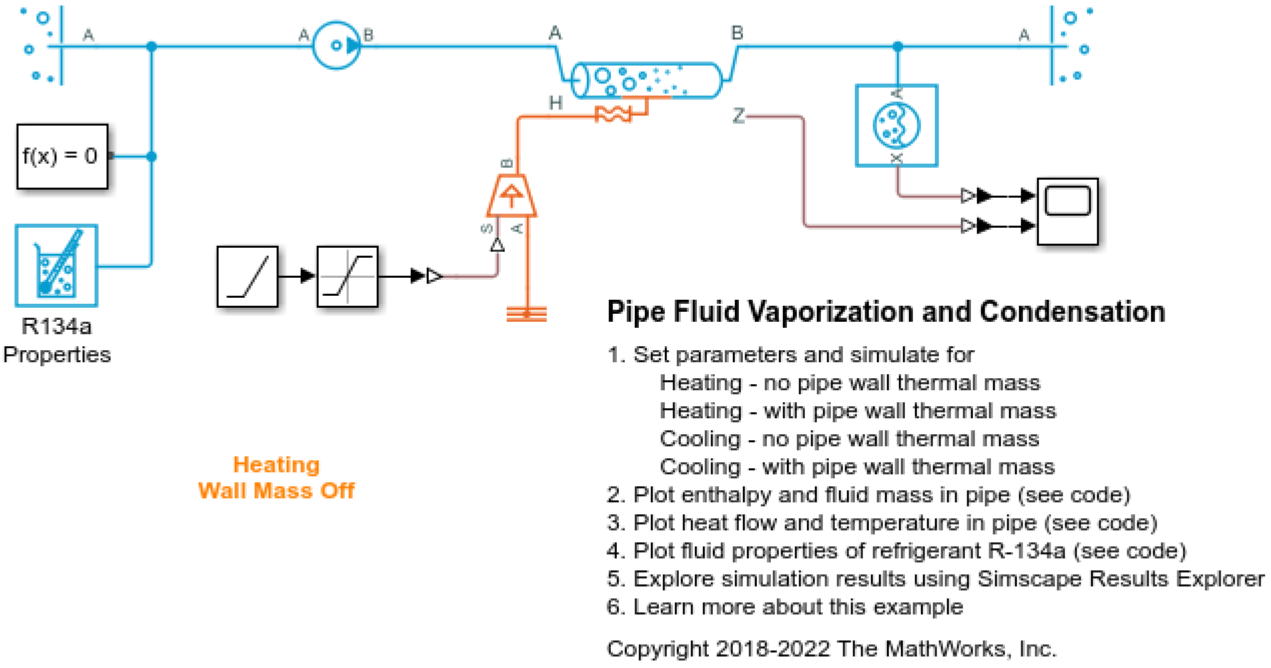
The plots below show the volumetric flow rate through and pressure rise across the CPU fan.


CPU Fan
Volumetric Flow Rate (l/min)
0
20
40
60
0
2000
4000
6000
8000
10000
12000
Pressure Rise (Pa)
0
0.2
0.4
0.6
0
2000
4000
6000
8000
10000
12000
Time (s)

Pipe Fluid Vaporization and Condensation

This example shows the 3-Zone Pipe (2P) block used to model vaporization or condensation of fluid flow in a pipe. The block divides the internal fluid volume into up to three zones: liquid zone, mixture zone, and vapor zone, depending on the state of the fluid along the pipe. As fluid flows through the pipe, heat is transferred between the environment external to the pipe and the fluid inside the pipe, causing it to change from liquid to mixture to vapor for the heating case or from vapor to mixture to liquid for the cooling case. The effect of thermal storage in the pipe wall can be optionally turned on by specifying a nonzero pipe wall thickness.

Model

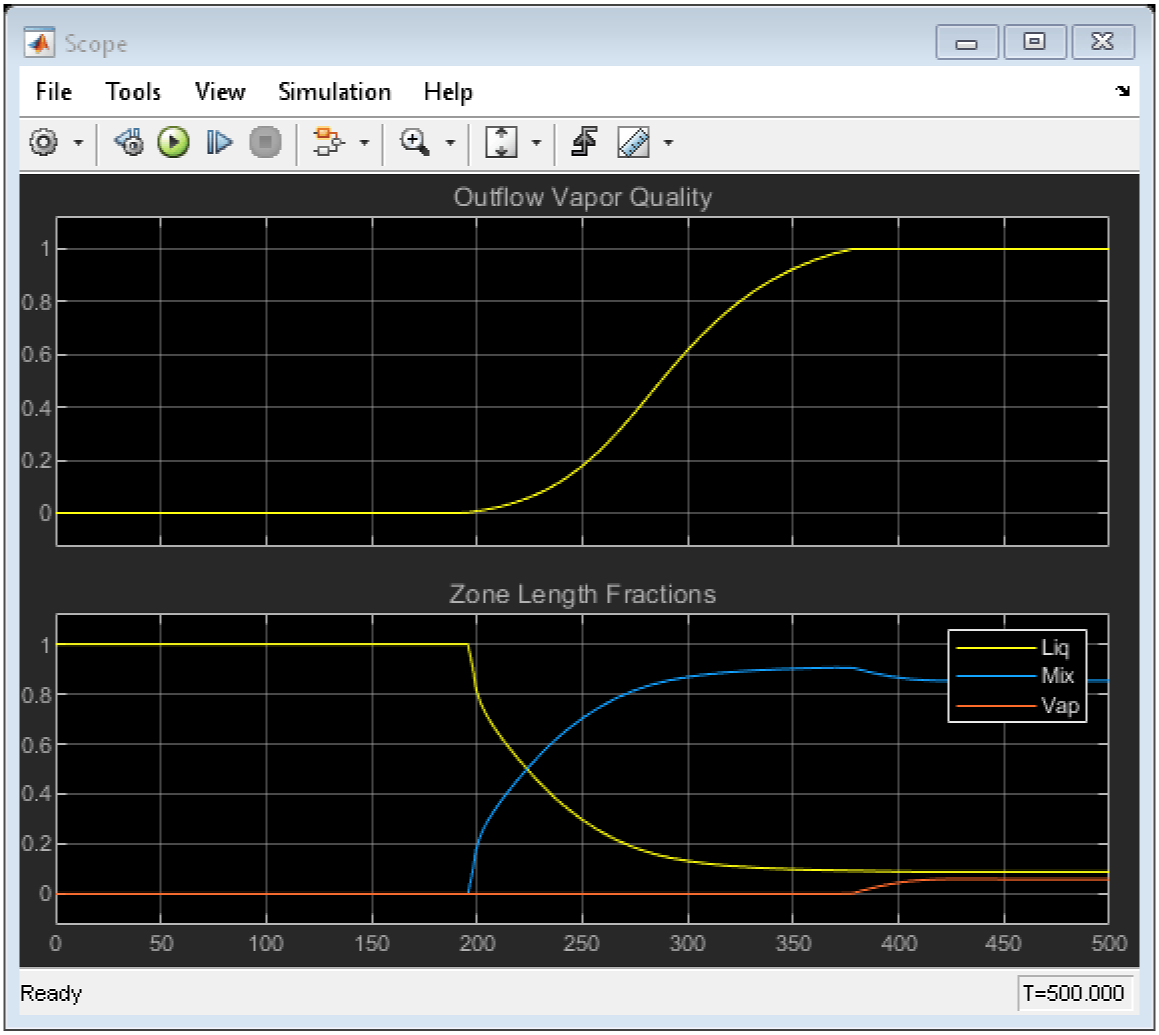


Pipe Fluid Vaporization and Condensation

1. Set parameters and simulate for
   Heating - no pipe wall thermal mass
   Heating - with pipe wall thermal mass
   Cooling - no pipe wall thermal mass
   Cooling - with pipe wall thermal mass
2. Plot enthalpy and fluid mass in pipe (see code)
3. Plot heat flow and temperature in pipe (see code)
4. Plot fluid properties of refrigerant R-134a (see code)
5. Explore simulation results using Simscape Results Explorer
6. Learn more about this example

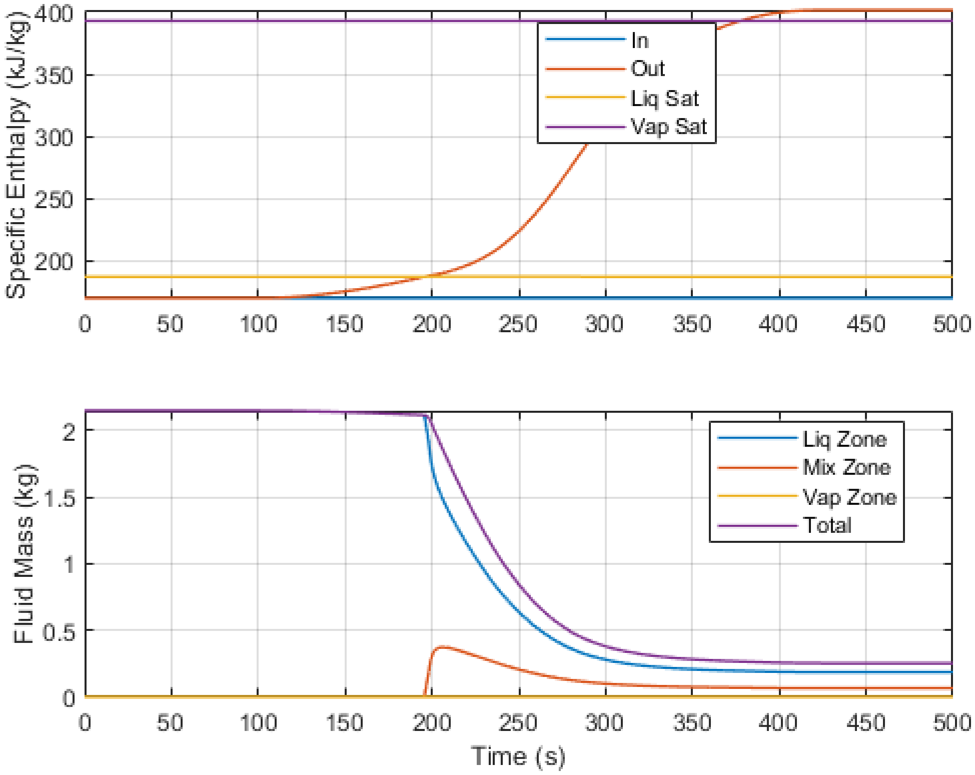
Copyright 2018-2022 The MathWorks, Inc.

Simulation Results from Scopes

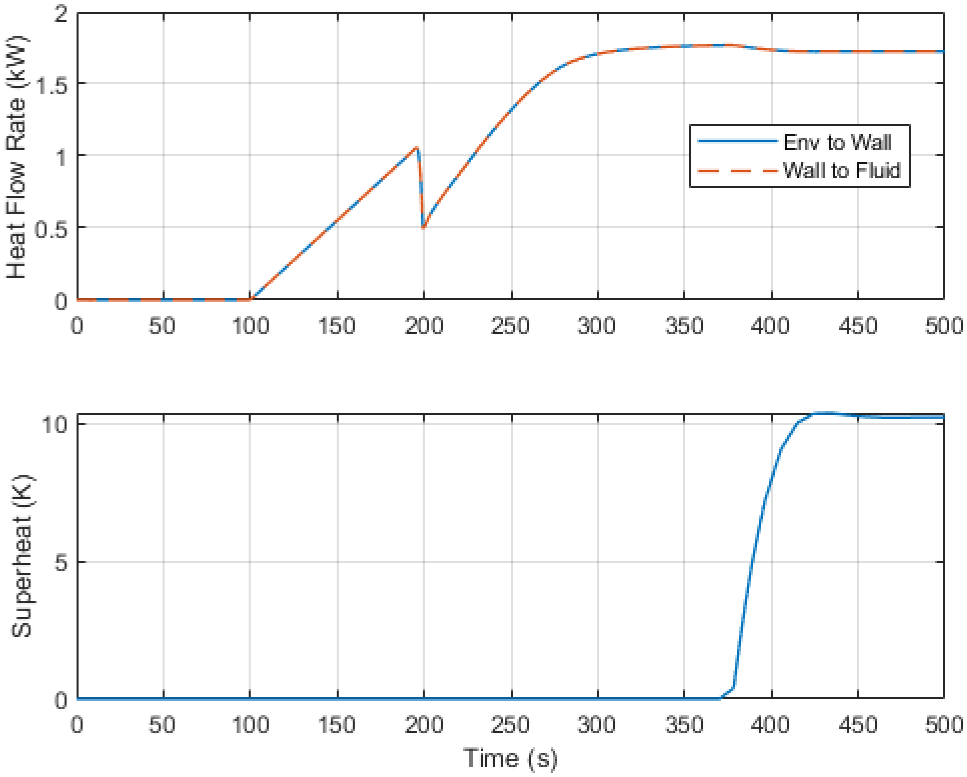


Simulation Results from Simscape Logging

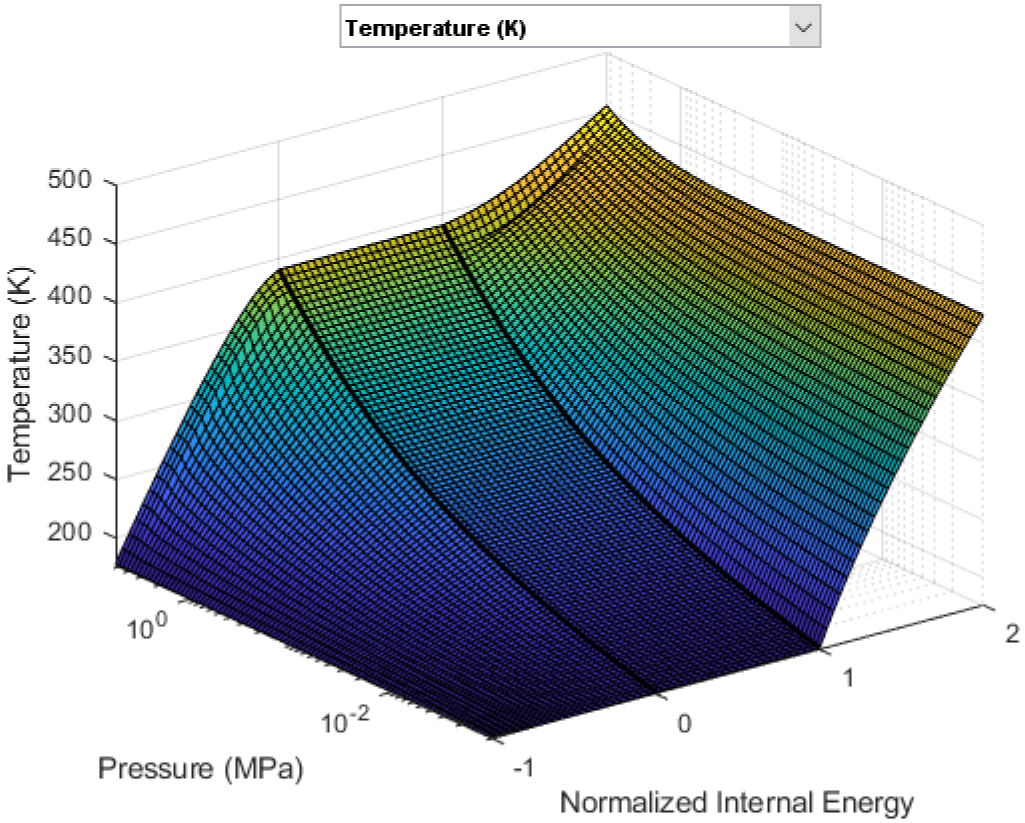
This plot shows specific enthalpy at the beginning and at the end of the pipe. The appearance and disappearance of the liquid, mixture, and vapor zones occurs when the specific enthalpies cross a saturation boundary. The plot also shows the amount of fluid mass within each zone.

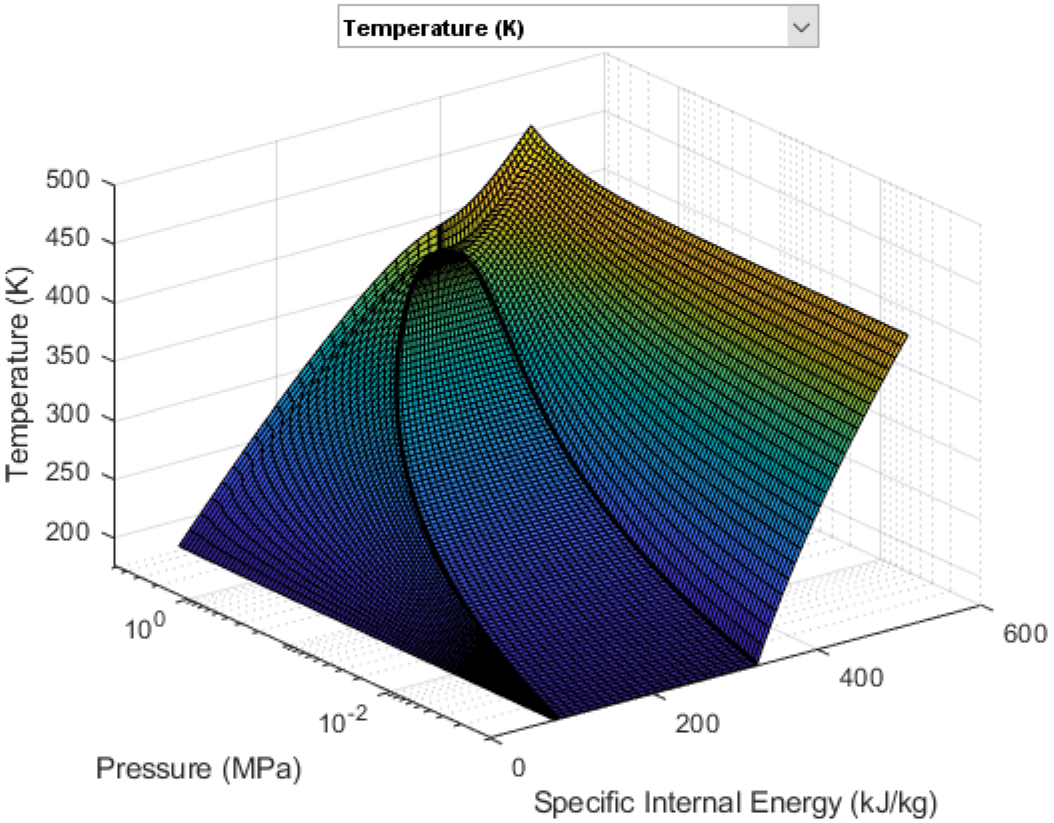


This plot shows the rate of heat transfer, which consists of the heat flow rate from the external environment to the pipe wall and the heat flow rate from the pipe wall to the fluid. When the pipe wall thermal mass is omitted, the two heat flow rates are equal. When there is nonzero pipe wall thermal mass, the two heat flow rates are not equal until steady-state is reached due to thermal storage in the segment of the pipe wall over each zone. The plot also shows the amount of superheat in the outflow for the heating case or the amount of subcooling in the outflow for the cooling case.


Heat Flow Rate (kW)
2
1.5
1
0.5
0
0
50
100
150
200
250
300
350
400
450
500
Env to Wall
Wall to Fluid
Superheat (K)
10
5
0
Time (s)

Fluid Properties

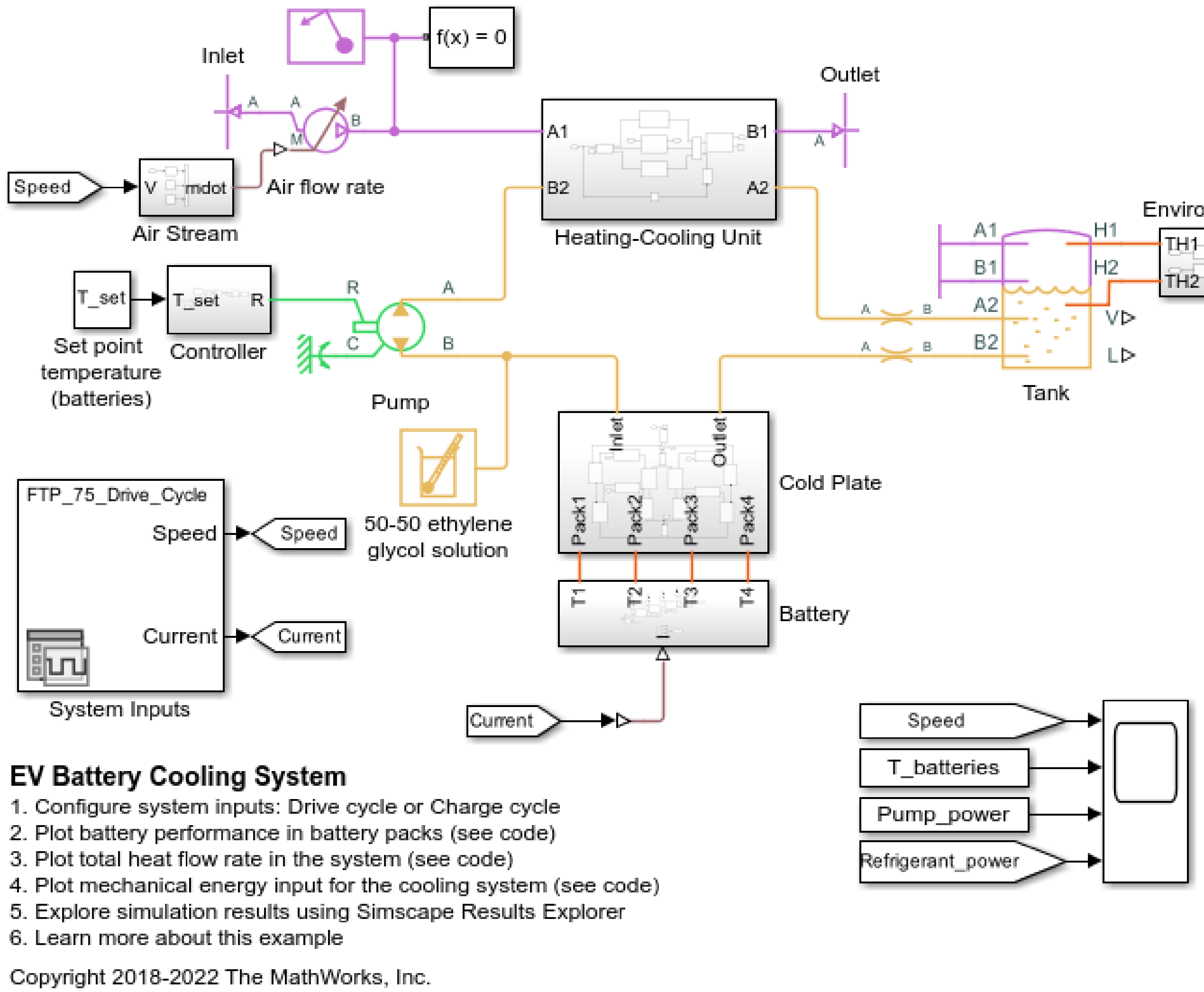



Temperature (K)
Temperature (K)
500
450
400
350
300
250
200
Pressure (MPa)
10^0
10^{-2}
Specific Internal Energy (kJ/kg)
0
200
400
600

EV Battery Cooling System

This demo shows an Electric Vehicle (EV) battery cooling system. The battery packs are located on top of a cold plate which consists of cooling channels to direct the cooling liquid flow below the battery packs. The heat absorbed by the cooling liquid is transported to the Heating-Cooling Unit. The Heating-Cooling Unit consists of three branches to switch operating modes to cool and heat the battery. The Heater represents an electrical heater for fast heating of the batteries under low temperature conditions. The Radiator uses air-cooling and/or heating when the batteries are operated stably. The Refrigerant system is used for cooling the overheated batteries. The refrigeration cycle is represented by the amount of heat flow extracted from the cooling liquid. The system is simulated under either FTP-75 drive cycle or fast charge scenarios with different environment temperatures.

Model

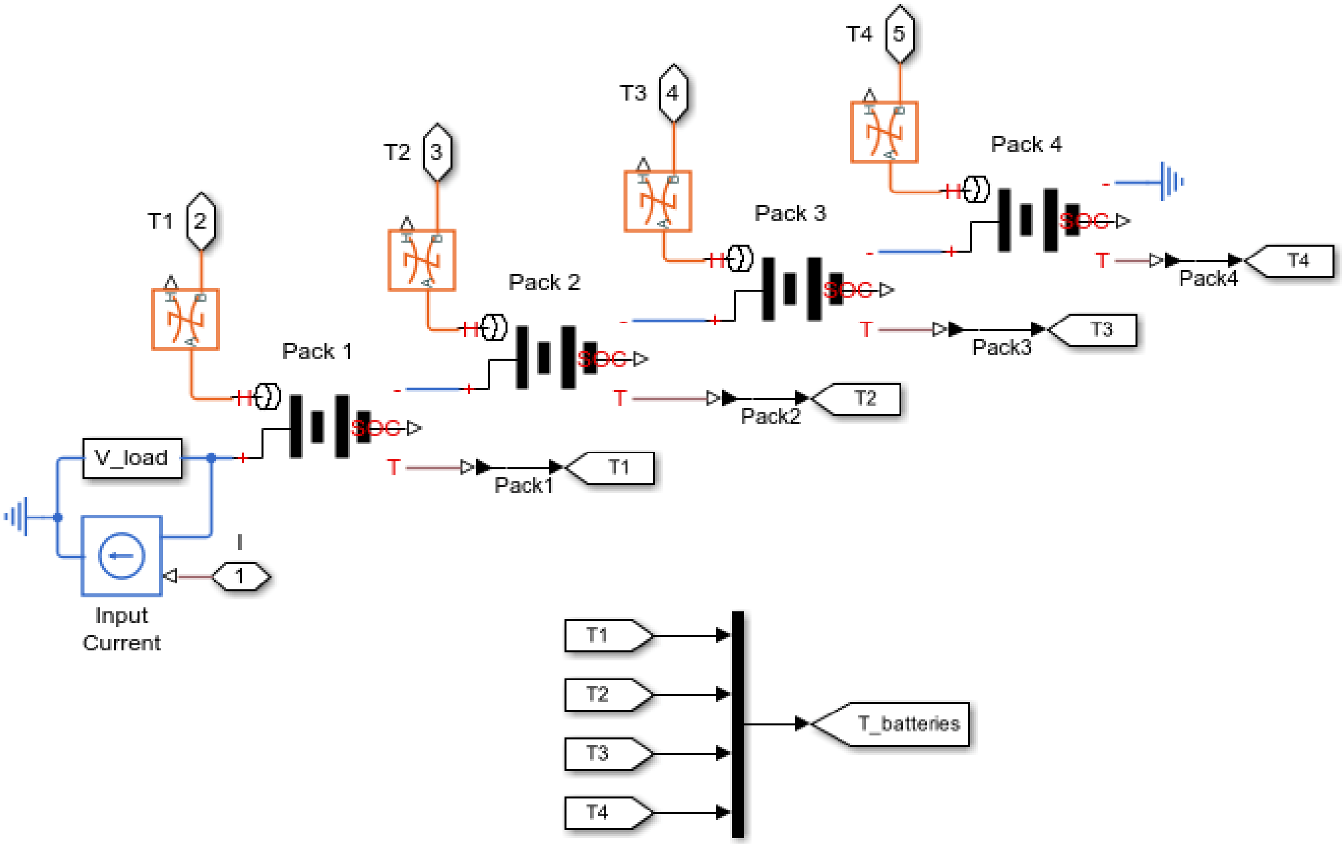


EV Battery Cooling System

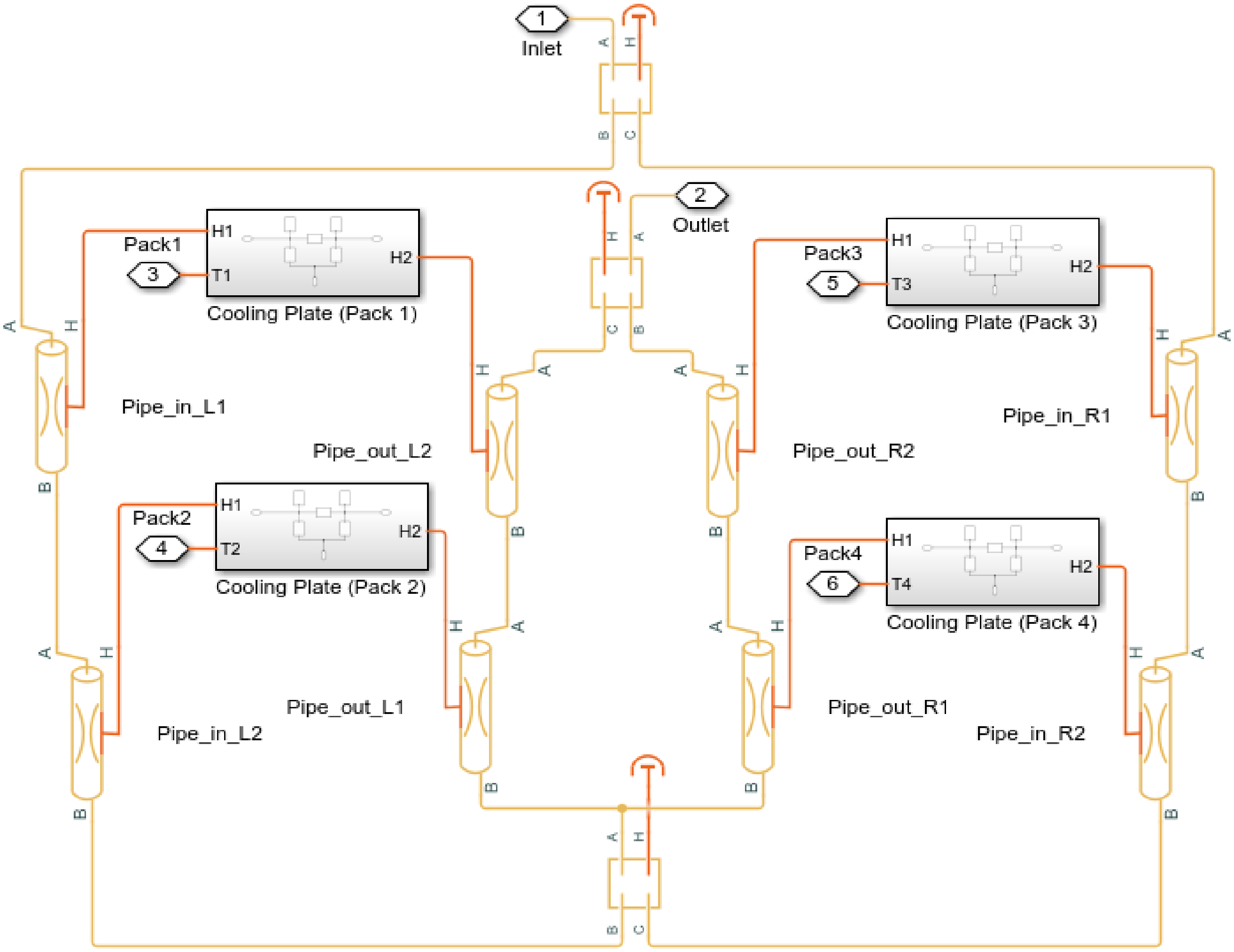
1. Configure system inputs: Drive cycle or Charge cycle
2. Plot battery performance in battery packs (see code)
3. Plot total heat flow rate in the system (see code)
4. Plot mechanical energy input for the cooling system (see code)
5. Explore simulation results using Simscape Results Explorer
6. Learn more about this example

Copyright 2018-2022 The MathWorks, Inc.

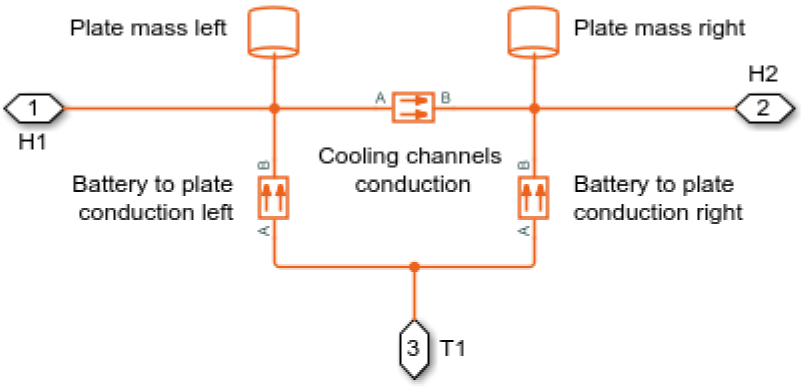
Battery Subsystem



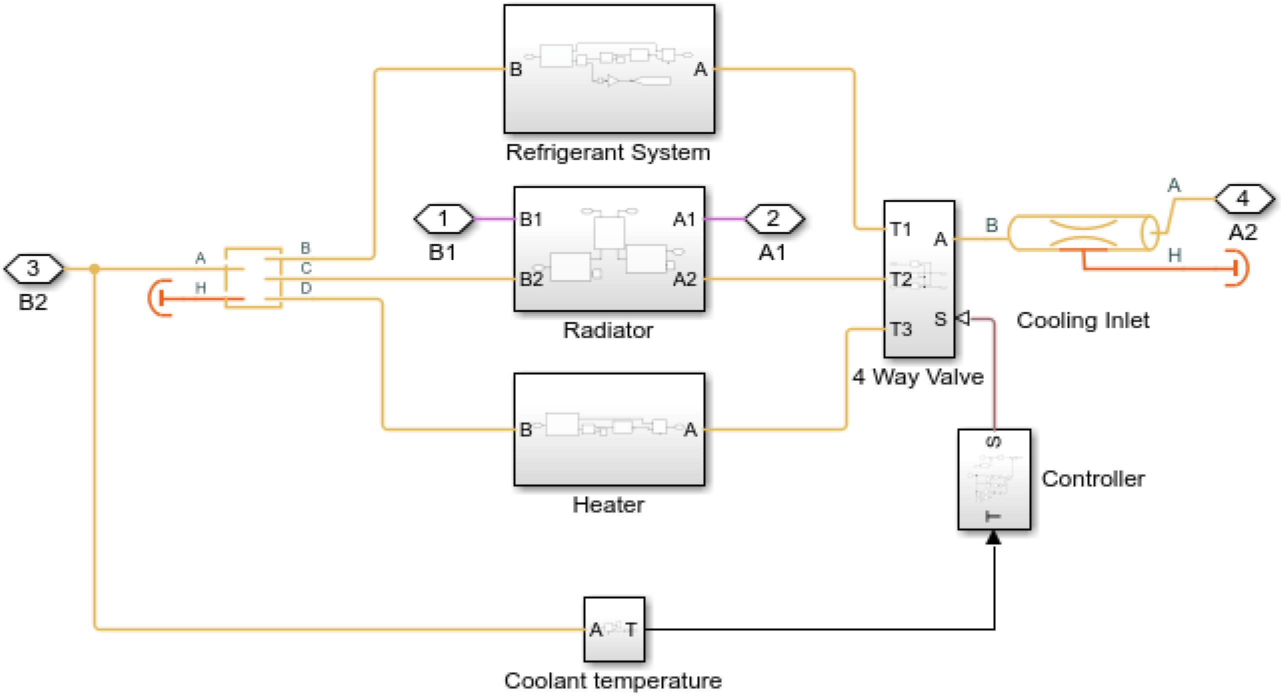
Cold Plate Subsystem



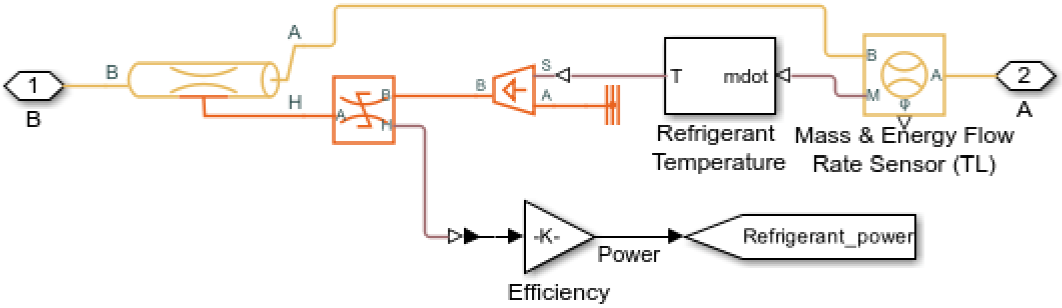
Cooling Plate (Pack 1) Subsystem



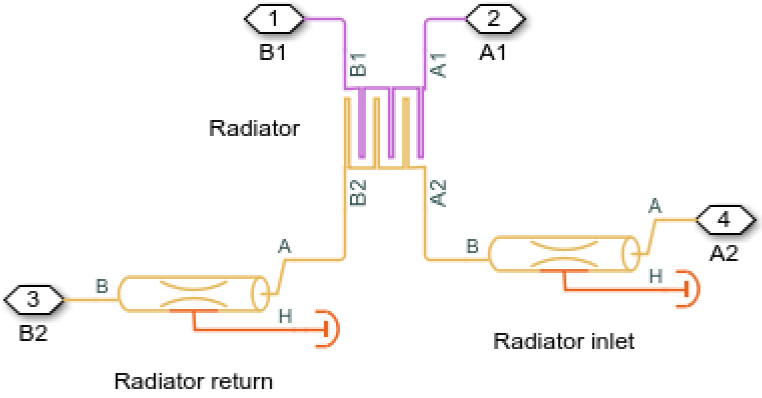
Heating-Cooling Unit Subsystem



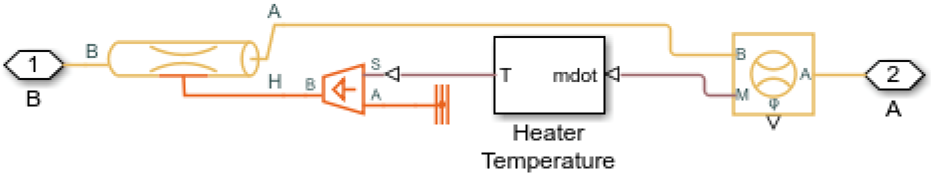
Refrigerant Subsystem



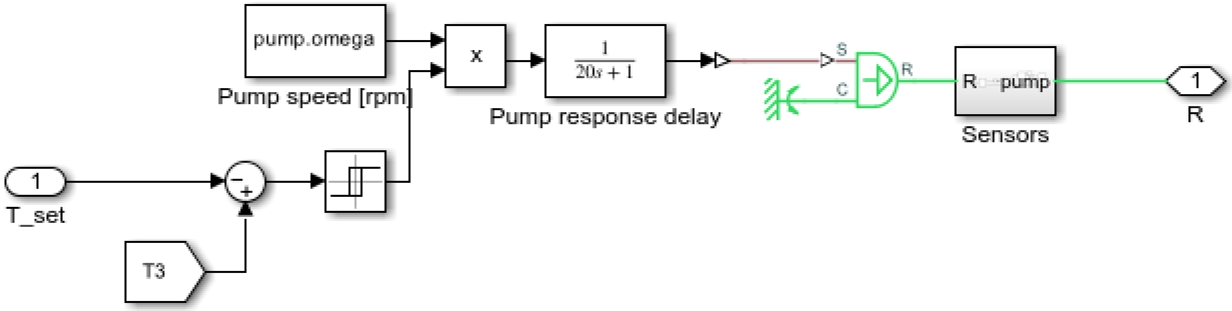
Radiator Subsystem



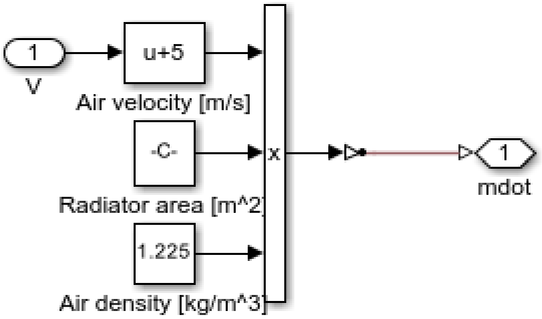
Heater Subsystem



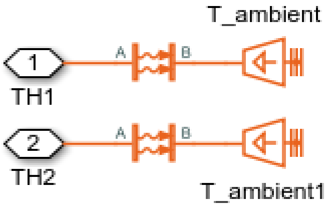
Pump Controller



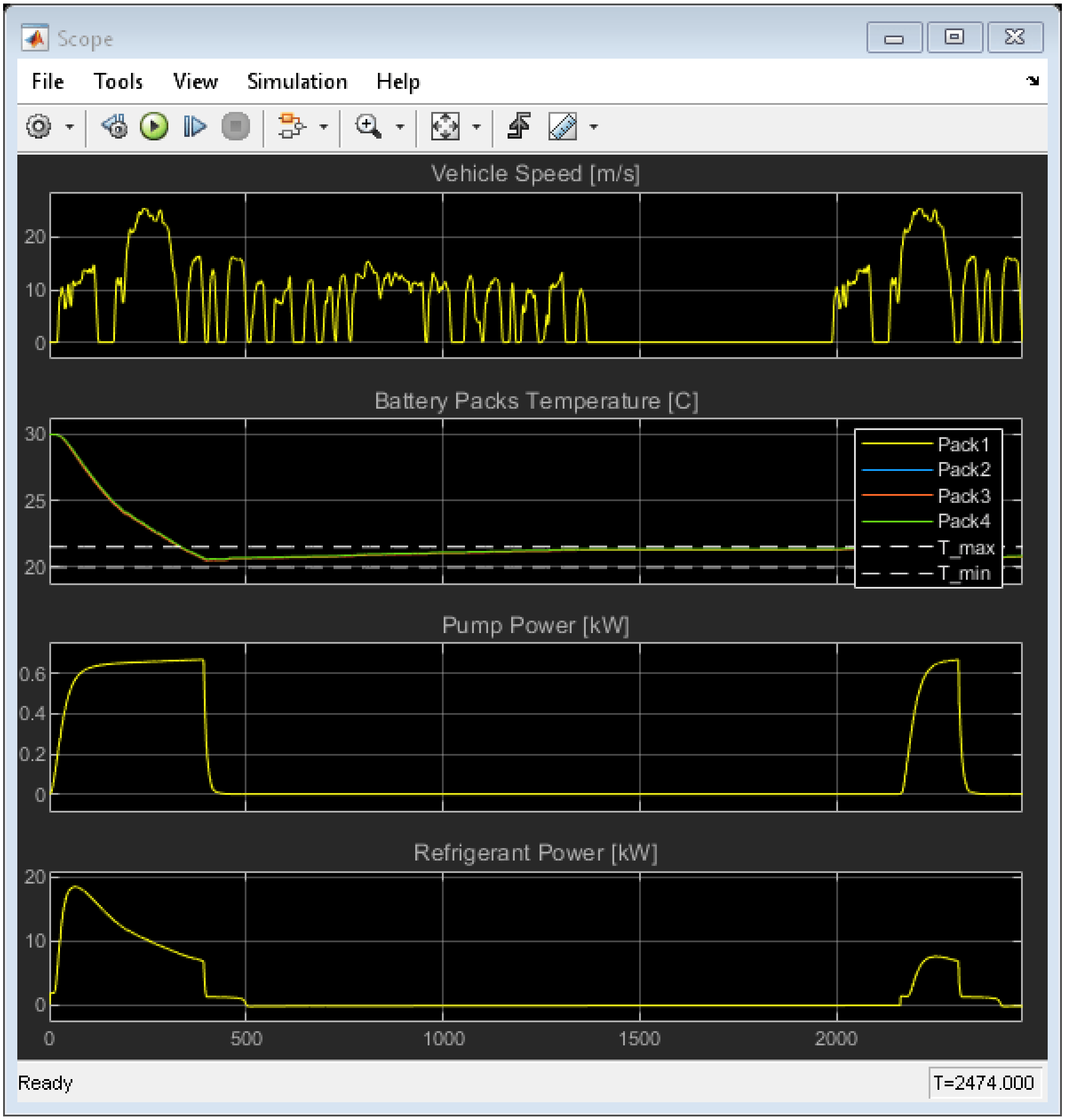
Air Stream



Environment

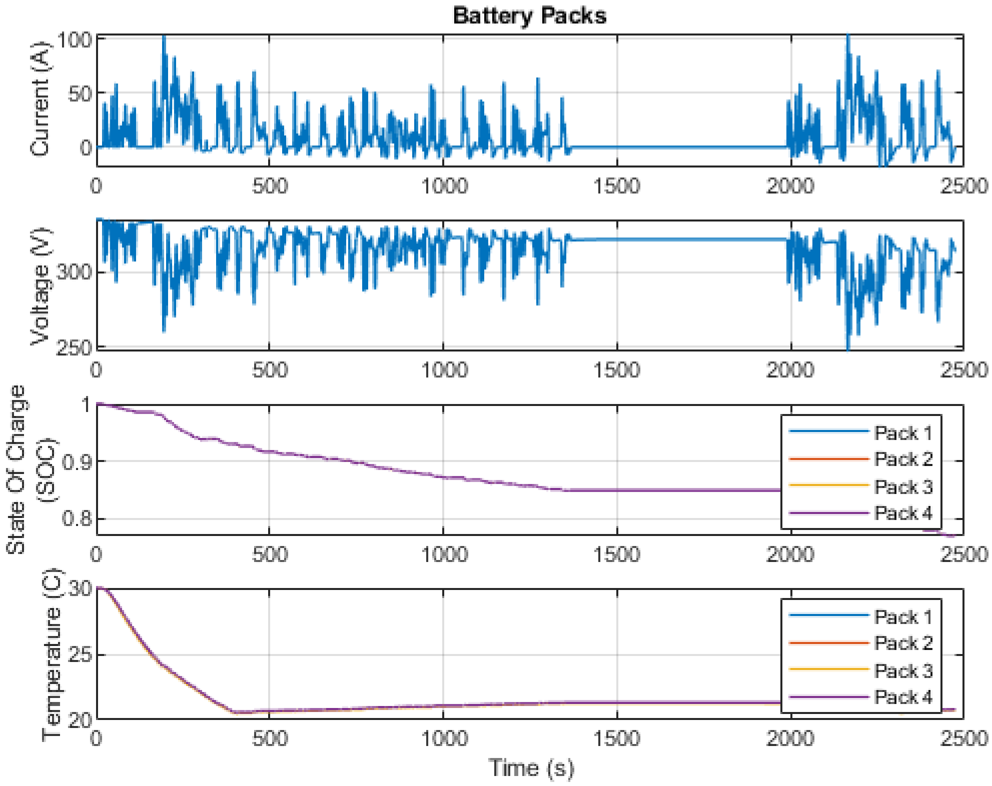


Simulation Results from Scopes

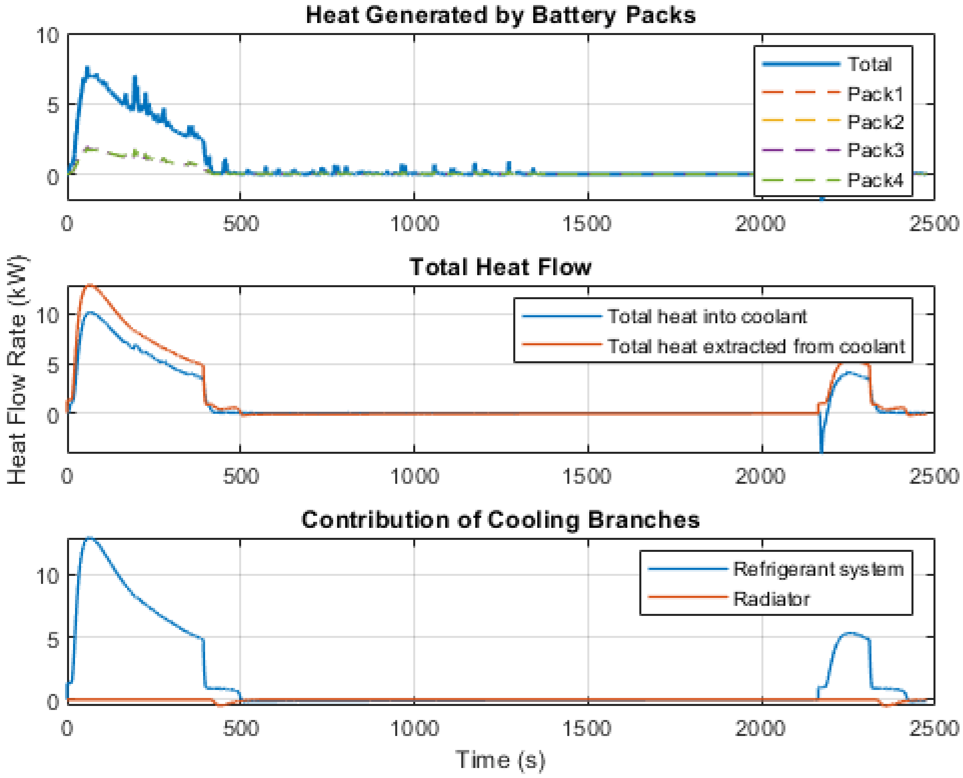


Simulation Results from Simscape Logging

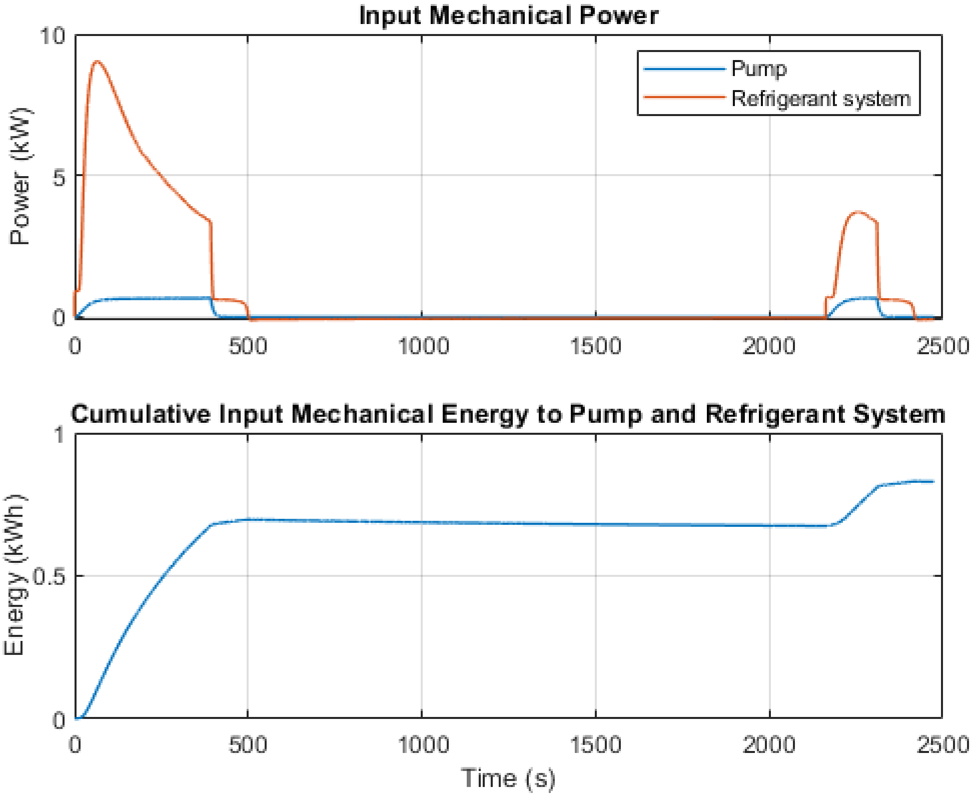
This figure shows the performance of series of four lithium-ion battery packs. The data used for each cell is based on T. Huria, M. Ceraolo, J. Gazzarri,R. Jackey. "High Fidelity Electrical Model with Thermal Dependence for Characterization and Simulation of High Power Lithium Battery Cells," IEEE International Electric Vehicle Conference, March 2012.



This figure shows the total heat flow rate generated by batteries, the total heat flow into to the coolant and the total heat flow extracted from the coolant, and the contribution of each cooling system.



This figure shows the mechanical energy input for the cooling system.

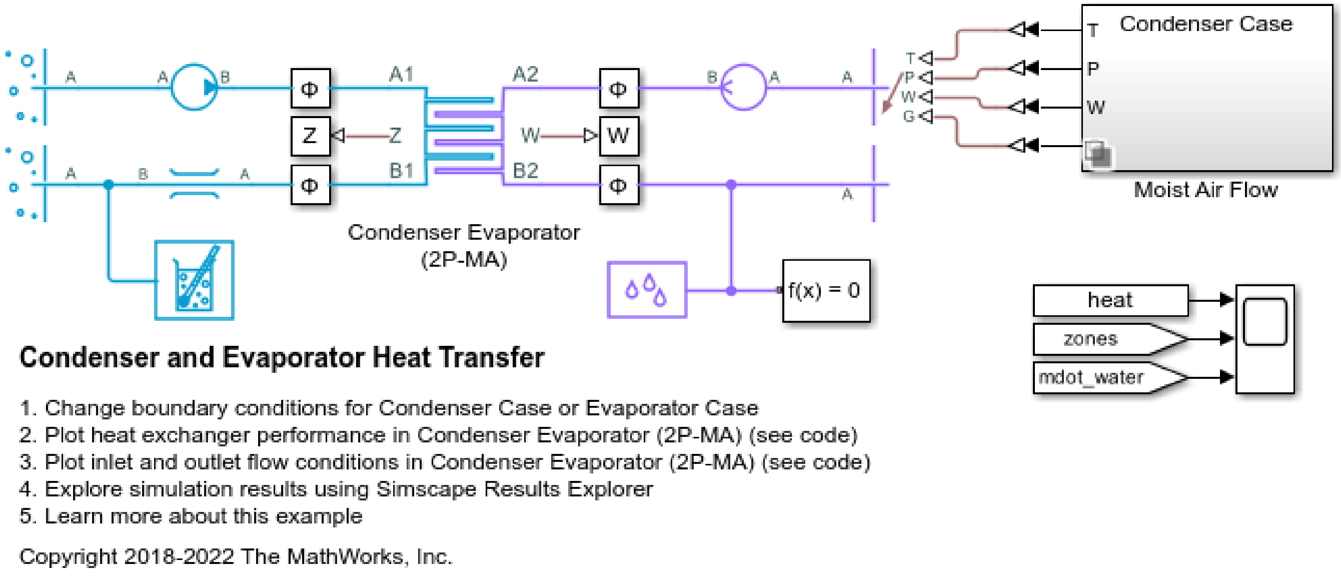

Input Mechanical Power
Power (kW)
10
5
0
0
500
1000
1500
2000
2500
Pump
Refrigerant system
Cumulative Input Mechanical Energy to Pump and Refrigerant System
Energy (kWh)
1
0.5
0
0
500
1000
1500
2000
2500
Time (s)

Condenser and Evaporator Heat Transfer

This example models a condenser or an evaporator in simple test setup with R134a refrigerant on the left side and moist air on the right side. It has a cross flow arrangement with the moist air blowing across tube banks filled with the refrigerant.

The Condenser Evaporator (2P-MA) block can operate as either a condenser or an evaporator. In the condenser case, heat flows from the refrigerant to the moist air. This causes the refrigerant to condense from a superheated vapor to a two-phase mixture to a subcooled liquid, resulting in up to three fluid zones along the length of the condenser tubes. In the evaporator case, heat flows from the moist air to the refrigerant. If the moist air is sufficiently wet, water vapor condenses on the surface and is removed from the rest of the moist air flow.

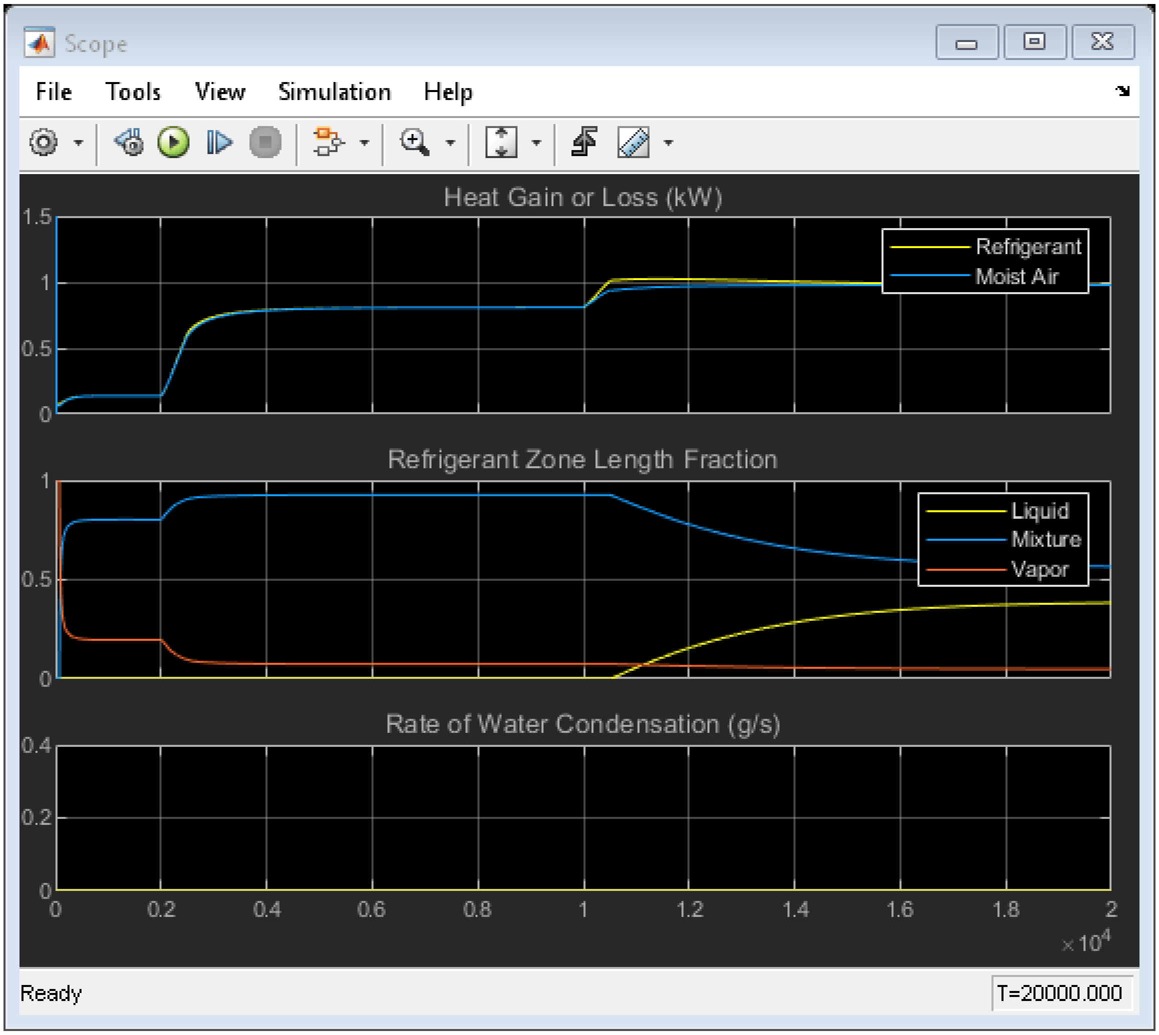
Model



Condenser and Evaporator Heat Transfer

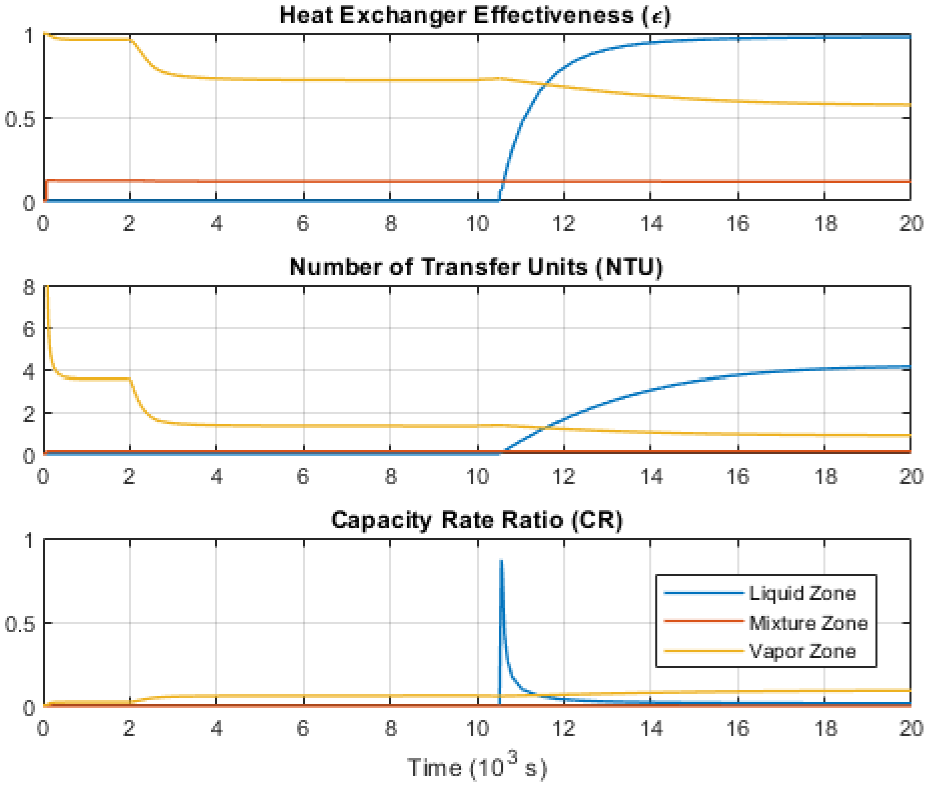
1. Change boundary conditions for Condenser Case or Evaporator Case
2. Plot heat exchanger performance in Condenser Evaporator (2P-MA) (see code)
3. Plot inlet and outlet flow conditions in Condenser Evaporator (2P-MA) (see code)
4. Explore simulation results using Simscape Results Explorer
5. Learn more about this example

Simulation Results from Scopes

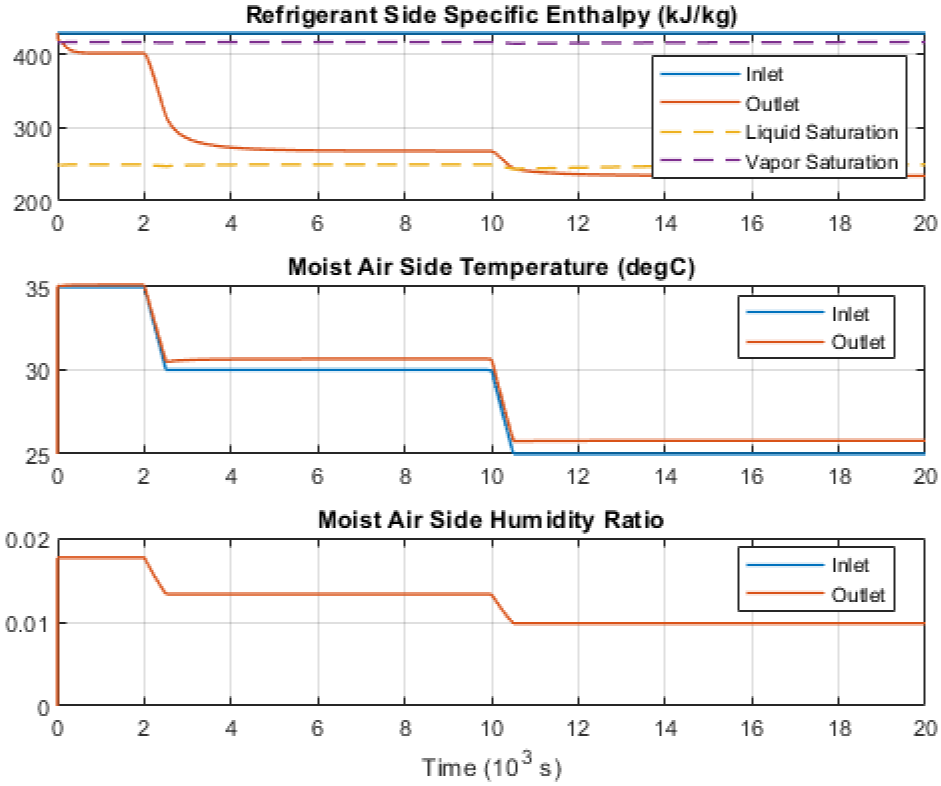


Simulation Results from Simscape Logging

Heat transfer between the refrigerant and the moist air is computed with the ϵ-NTU method. The liquid zone, mixture zone, and vapor zone along the refrigerant tubes have different heat transfer coefficients, heat transfer surface areas, and inlet temperatures. Therefore, the heat exchanger effectiveness, number of transfer units, and the capacity rate ratio are different in each zone. If a zone does not exist, a default value of 0 is shown.



This plot shows the fluid state at the inlets and outlets of the condenser or the evaporator. Liquid, mixture, and vapor zones appears and disappears as the specific enthalpy crosses the saturation boundaries.


Refrigerant Side Specific Enthalpy (kJ/kg)
400
300
200
Inlet
Outlet
Liquid Saturation
Vapor Saturation
0
2
4
6
8
10
12
14
16
18
20
Moist Air Side Temperature (degC)
35
30
25
Inlet
Outlet
Moist Air Side Humidity Ratio
0.02
0.01
0
Inlet
Outlet
Time (10^3 s)

Photovoltaic Thermal (PV/T) Hybrid Solar Panel

This example shows how to model the cogeneration of electrical power and heat using a hybrid PV/T solar panel. The generated heat is transferred to water for household consumption.

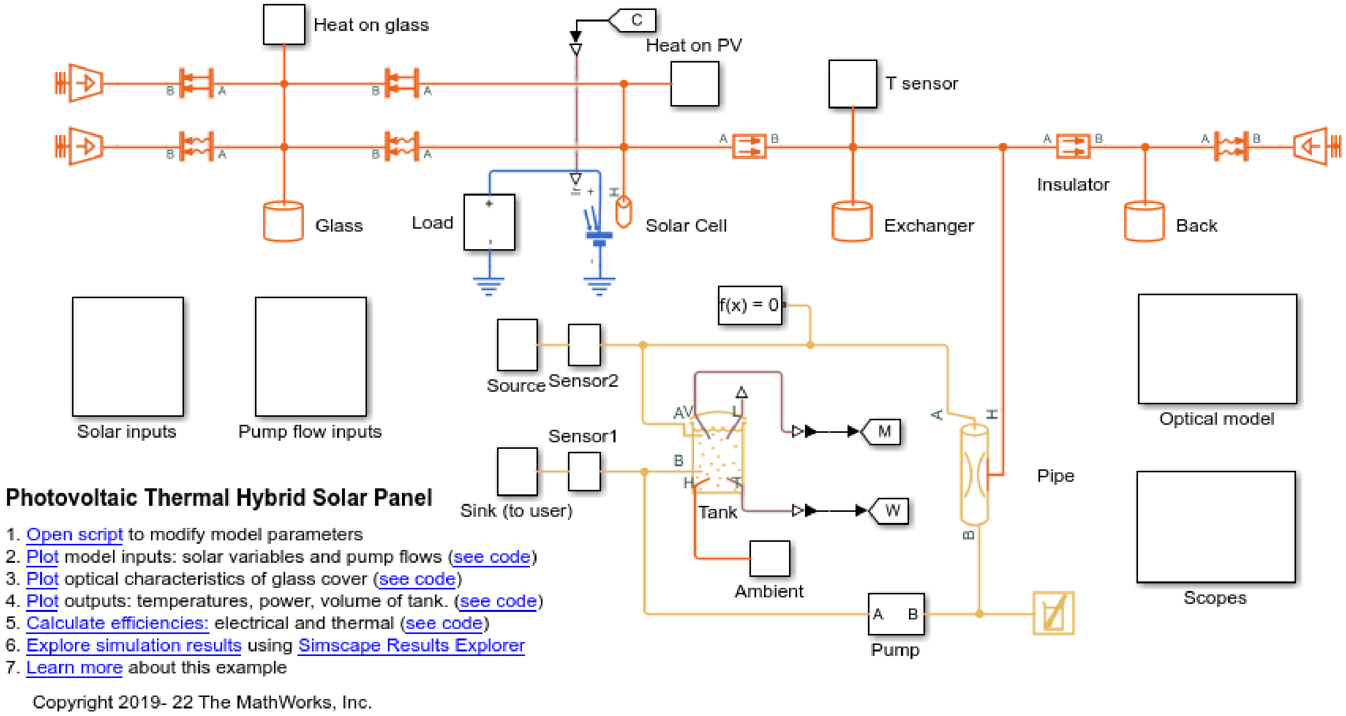
It uses blocks from the Simscape™ Foundation™, Simscape Electrical™, and Simscape Fluids™ libraries. The electrical portion of the network contains a Solar Cell block, which models a set of photovoltaic (PV) cells, and a Load subsystem, which models a resistive load. The thermal network models the heat exchange that occurs between the physical components of the PV panel (glass cover, heat exchanger, back cover) and the environment. Heat is exchanged through conduction, convection, and radiation. The thermal-liquid network contains a pipe, a tank, and pumps. The pumps control the flows of the liquids through the system.

To model the reflection, absorption and transmission of light in the glass cover, an optical model is embedded in a MATLAB® Function block.

Model overview

Open the model to view its structure:

```
open_system('sscv_hybrid_solar_panel');
```



Photovoltaic Thermal Hybrid Solar Panel

1. Open script to modify model parameters
2. Plot model inputs: solar variables and pump flows (see code)
3. Plot optical characteristics of glass cover (see code)
4. Plot outputs: temperatures, power, volume of tank. (see code)
5. Calculate efficiencies: electrical and thermal (see code)
6. Explore simulation results using Simscape Results Explorer
7. Learn more about this example

Copyright 2019- 22 The MathWorks, Inc.

The thermal network is in red, the electrical network in blue and the thermal liquid network in yellow. There are subsystems for the solar and pump inputs. There is also a subsystem that contains scopes for visualizing the simulation results. Another subsystem contains the function for the optical model.

Parameters

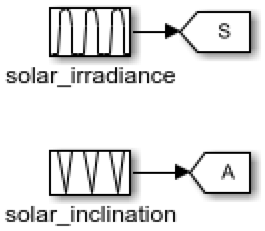
You can use the hybrid_solar_panel_data.m script to change the parameter values that this example uses for components such as the load, solar cell, pipe, and tank.

```
edit sscv_hybrid_solar_panel_data;
```

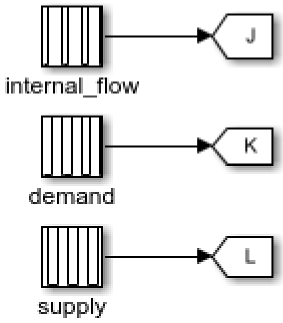
Inputs

The inputs of the model are the pump flows and the solar variables for irradiance and incidence angle. A repeating sequence block is used to define the inputs because they follow a 24-hour periodic cycle.

```
open_system('sscv_hybrid_solar_panel/Solar inputs');
```



```
open_system('sscv_hybrid_solar_panel/Pump flow inputs');
```

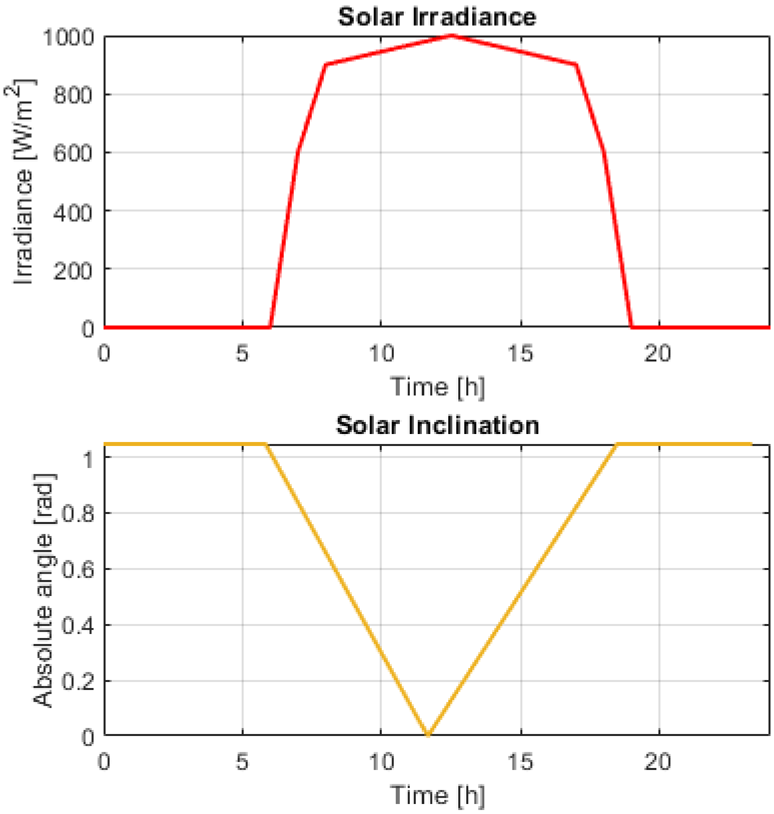


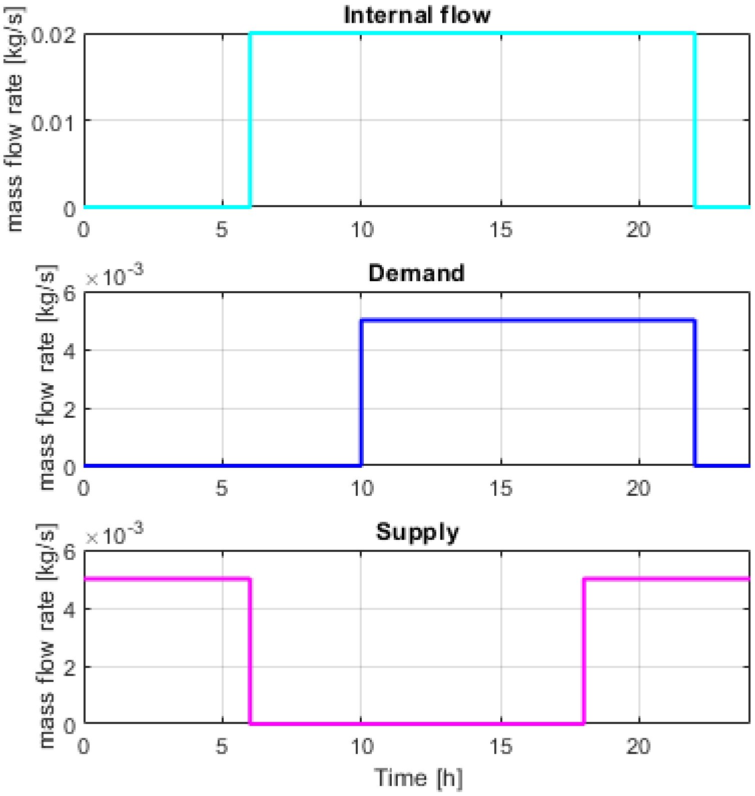
The sun rises at 6:00 and sets at 19:00. The irradiance follows a bell curve that peaks at 12:30. The incidence angle changes from pi/3 to 0.

There are three pumps. One pump models user demand, another models source supply, and a third models internal flow that forces convection in the pipe. The demand is constant and only non-zero from 10:00 to 22:00. The supply is constant and only non-zero from 18:00 to 6:00. The internal flow is also constant and only non-zero from 6:00 to 22:00. This model is used for the internal flow because it is not efficient to force heat exchange during the night when the ambient temperature is low.

You can use the hybrid_solar_panel_plot_inputs.m script to plot the inputs:

```
sscv_hybrid_solar_panel_plot_inputs;
```

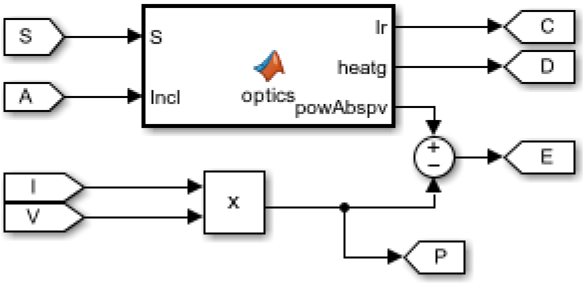

Solar Irradiance
Irradiance [W/m2]
1000
800
600
400
200
0
0
5
10
15
20
Time [h]
Solar Inclination
Absolute angle [rad]
1
0.8
0.6
0.4
0.2
0
0
5
10
15
20
Time [h]



Optical model for the glass cover

The optical model is inside a subsystem:

```
open_system('sscv_hybrid_solar_panel/Optical model');
```



It consists of a MATLAB® Function block, with the 2 solar inputs, and 3 outputs: the transmitted irradiance on the PV cells, the heat absorbed by the glass, and the radiative power absorbed by the PV cells. Part of it will be transformed into electrical power (V*I) and the rest will be heat absorbed by the PV cells.

From an optical point of view, the glass consists of 2 parallel boundaries (air-glass, glass-air), each one of those reflects and transmits light. The reflection coefficient in a boundary is obtained from the

Fresnel equations. r_p is for P-polarization and r_s for S-polarization. The total reflection is the average of both, and the transmittance is $1 - r$ as there is no absorption so far:

$$r_p = \left(\frac{n_{rel}^2 \cos(\theta_i) - \sqrt{n_{rel}^2 - \sin(\theta_i)^2}}{n_{rel}^2 \cos(\theta_i) + \sqrt{n_{rel}^2 - \sin(\theta_i)^2}} \right)^2$$

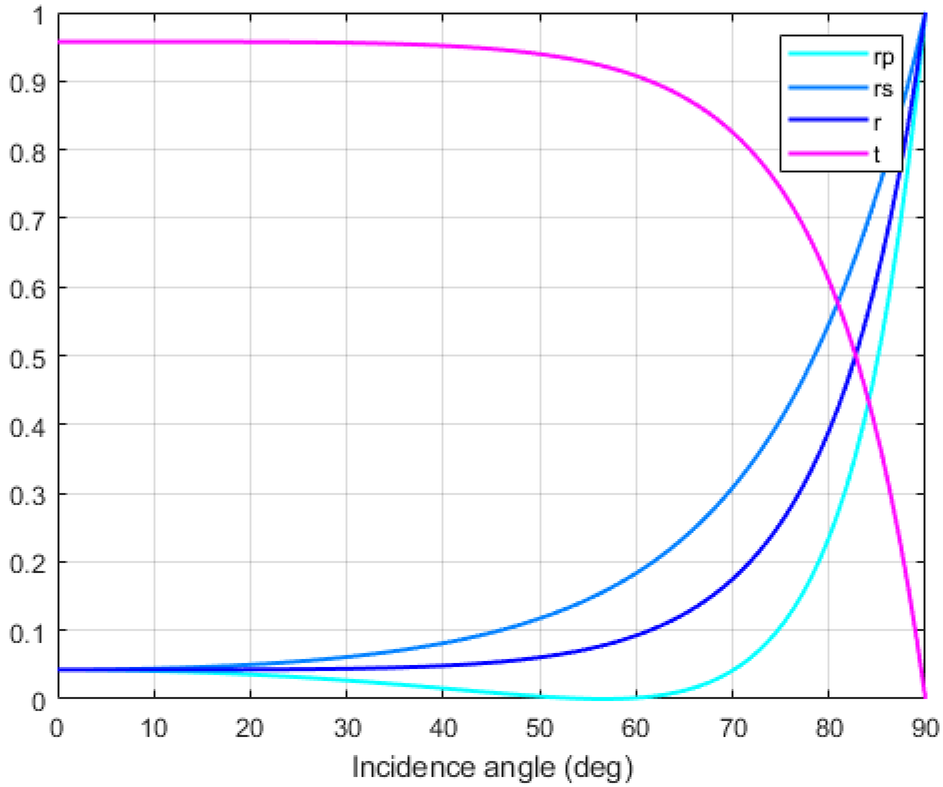
$$r_s = \left(\frac{\cos(\theta_i) - \sqrt{n_{rel}^2 - \sin(\theta_i)^2}}{\cos(\theta_i) + \sqrt{n_{rel}^2 - \sin(\theta_i)^2}} \right)^2$$

$$r = \frac{1}{2}\left(r_p + r_s\right)$$

$$t = 1 - r$$

This is an example of the optical coefficients rp, rs, r and t in function of incidence angle:

```
nrel = 1.52; %Optical index from air to glass
theta = linspace(0, pi/2, 100);
rp = ( nrel^2*cos(theta) - sqrt(nrel^2 - sin(theta).^2) ).^2./...
    ( nrel^2*cos(theta) + sqrt( nrel^2 - sin(theta).^2 ) ).^2 ;
rs = ( cos(theta) - sqrt(nrel^2 - sin(theta).^2) ).^2./...
    ( cos(theta) + sqrt( nrel^2 - sin(theta).^2 ) ).^2 ;
r = 0.5*(rp + rs);
t = 1 - r;

figure();
plot(theta*180/pi, rp, 'Color', [0 1 1], 'LineWidth', 1.5);
hold on
plot(theta*180/pi, rs, 'Color', [0 0.5 1], 'LineWidth', 1.5);
plot(theta*180/pi, r, 'Color', [0 0 1], 'LineWidth', 1.5);
plot(theta*180/pi, t, 'Color', 'm', 'LineWidth', 1.5);
legend('rp','rs','r','t');
xlabel('Incidence angle (deg)');
grid on
box on
```



This is what happens in one boundary, but the glass has 2 parallel boundaries separated by d_g. The angle after the 1st boundary is the incidence angle on the 2nd boundary and is calculated from **Snell's Law**:

$$n_1 \sin(\theta_1) = n_2 \sin(\theta_2)$$

When the light enters the glass, it absorbs part of it with a constant probability per unit length (alpha_g), resulting in an exponential decay from distance travelled for the transmittance coefficient in the glass:

$$\tau_g = \exp\left(\frac{-\alpha_g d_g}{\cos(\theta_2)}\right)$$

Then, when it arrives at the 2nd boundary, it reflects and transmits again with Fresnel equations. The reflected light is trapped inside the glass, reflecting infinite times between the 2 boundaries until completely absorbed. The total reflection and transmission coefficients of the system are then the sum of an infinite geometrical series, for which the result is:

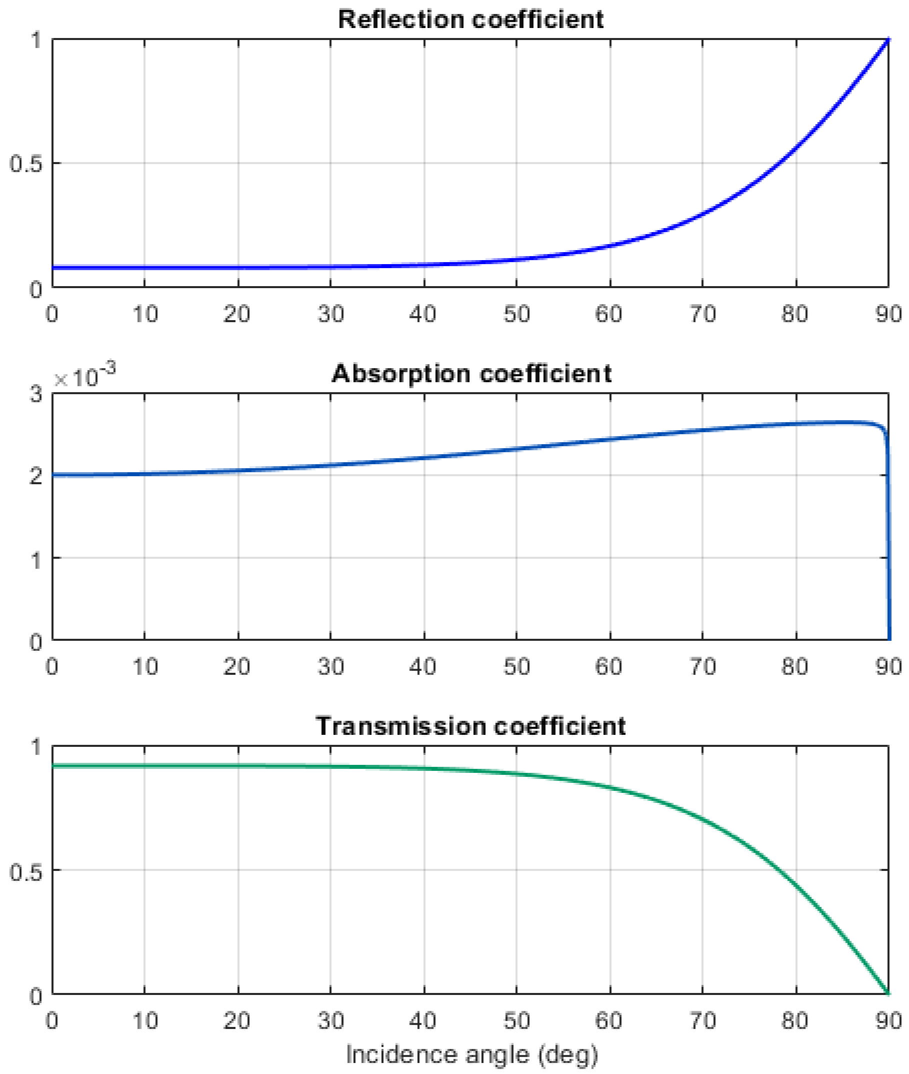
$$T_g = \frac{t_1 \tau_g t_2}{1 - r_1 r_2 \tau_g^2}$$

$$R_g = r_1 + \frac{t_1^2 \tau_g^2 r_2}{1 - r_1 r_2 \tau_g^2}$$

$$A_g = 1 - T_g - R_g$$

Finally, the total optical coefficients for the glass are:

```
sscv_hybrid_solar_panel_plot_optics;
```

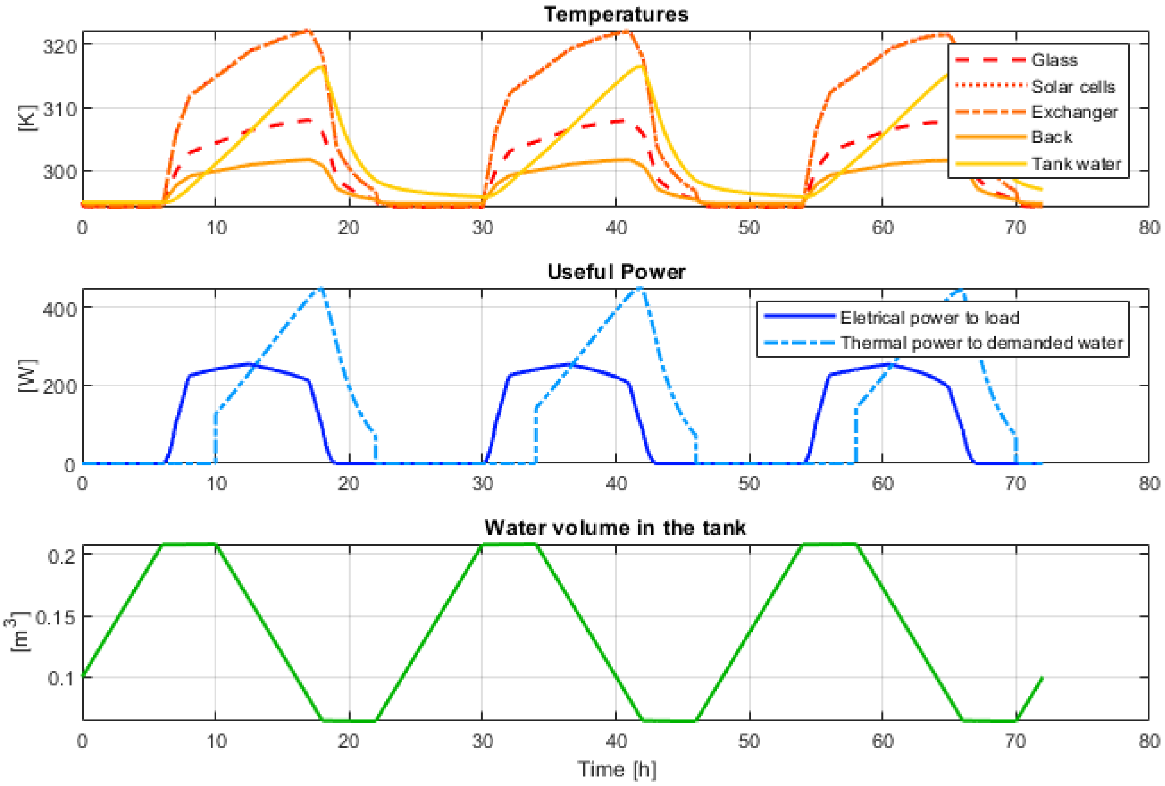


Outputs

The outputs of the model are the temperatures of all components of the panel, the electrical and thermal power, and the volume in the tank.

You can use the script hybrid_solar_panel_plot_outputs to plot the solution:

```
sscv_hybrid_solar_panel_plot_outputs;
```



Efficiency calculation

From the outputs it is possible to calculate the electrical, thermal, and total efficiency of the panel:

```
sscv_hybrid_solar_panel_efficiency;
```

```
****** Efficiency Calculation *********

Total input energy from the sun in the period: 43.784 kWh
Average input energy from the sun per day: 14.5947 kWh/day

Total electrical energy supplied to the load: 7.534 kWh
Average electrical energy supplied per day: 2.5113 kWh/day

Total absolute thermal energy in the water supplied to the user: 26.4353 kWh
Total absolute thermal energy in the water extracted from the source: 16.5049 kWh

Total used thermal energy (sink - source): 9.9303 kWh
Average used thermal energy per day (sink - source): 3.3101 kWh/day

Electrical efficiency: 0.17207
Thermal efficiency: 0.2268
Total efficiency: 0.39888

***************************************
```

The electrical efficiency is on the order of standard PV cells, but adding the thermal efficiency the production of energy is significantly better, with a system efficiency on the order of a cogeneration plant.

A further analysis could use Simulink® Design Optimization™ or other optimization tools to find optimal values for certain parameters eligible for control, maximizing total efficiency.

Another improvement would be the addition of controllers to the pumps and the electrical load, in order to drive the system to different operating points and optimize the performance.