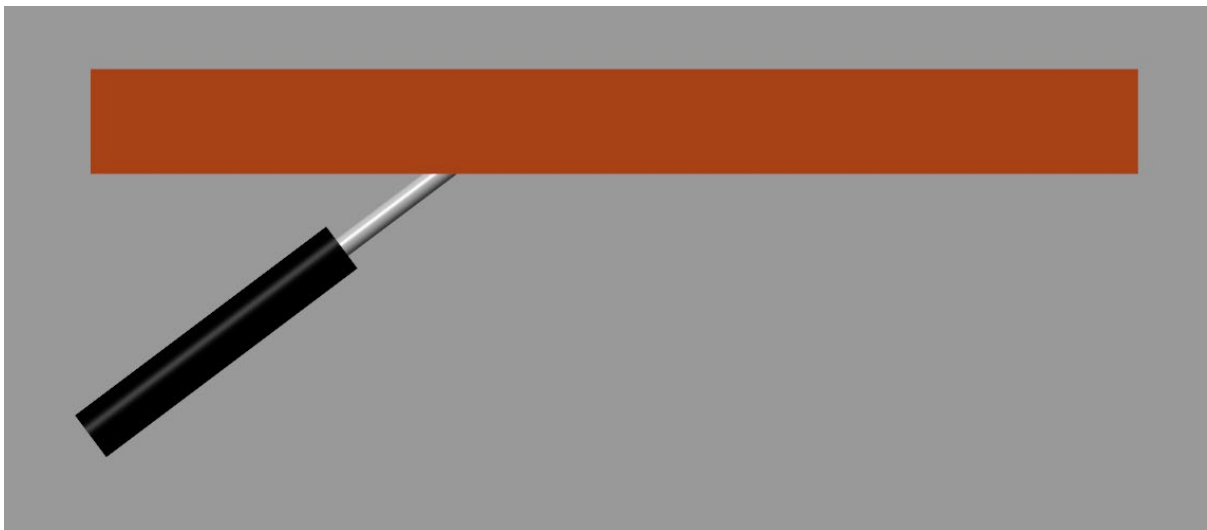


Multibody Dynamics

Introduction to 3D mechanics with MATLAB/Simulink and Simscape Multibody

Morten H. Rudolfson

This tutorial provides an introduction to assembling a simple 3D mechanical system using Simscape Multibody. The system comprises two components: a straight boom and a cylinder.



Ultimately, this tutorial will connect the mechanical system to the simulated hydraulic system from the Simscape Fluids tutorial, enabling control of mechanical movements.

Contents

Initialization	3
Pendulum Modeling	5
Revolute Joint.....	5
Connecting a Mass	6
Mass Property Adjustments.....	7
Intuitive Connections	8
Running the Simulation.....	9
Setting Initial States	10
Result.....	10
Mechanical Cylinder	11
Prismatic Joint.....	11
Settings for Actuation:.....	11
Settings for Sensing:	12
Prismatic Joint with Revolute Joints	12
Graphics and Frames of the Cylinder	12
Setting Initial States	13
Creating a Sub System for the Cylinder.....	13
Moving the contact points of the cylinders	14
Controlling the Piston Position	14
Closed Loop Force Control.....	15
Changing the Control Signal and Feedback of the Cylinder.....	15
Sensing the Boom Angle	15
Closed Loop Control.....	16
Using a Signal Generator	16
Connecting Hydraulics	17
Parameters of the Cylinder.....	17
Connecting the Hydraulic System	18
Initial Condition.....	18
Hydraulic Closed Loop Control	19
Results.....	19

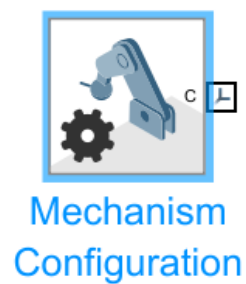
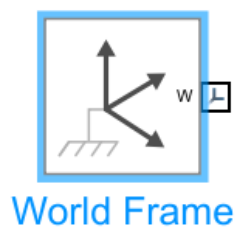
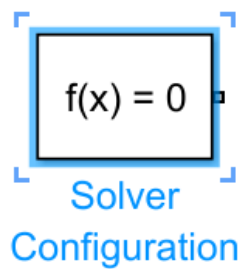
Initialization

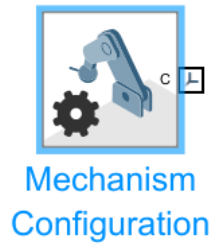
This tutorial is made with MATLAB R2024b, and with the following packages.

- Simulink
- Simscape Multibody
- Simscape Fluids

When a new Simulink model has been opened, locate the following blocks in the library browser, or using the search function.

- **Solver Configuration** - Solver for the simulation
- **World Frame** – a reference point to build from
- **Mechanism Configuration** – Settings for gravitational force and other mechanism settings





Block Parameters: Mechanism Configuration

Mechanism Configuration ☒ Auto Apply

Settings Description

NAME	VALUE
Uniform Gravity	Constant
Gravity	[0 0 -9.81] m/s^2 Compile-time
Linearization Delta	0.001
Joint Mode Transition	

Pendulum Modeling

We begin by modeling a pendulum swinging in 2D. This involves:

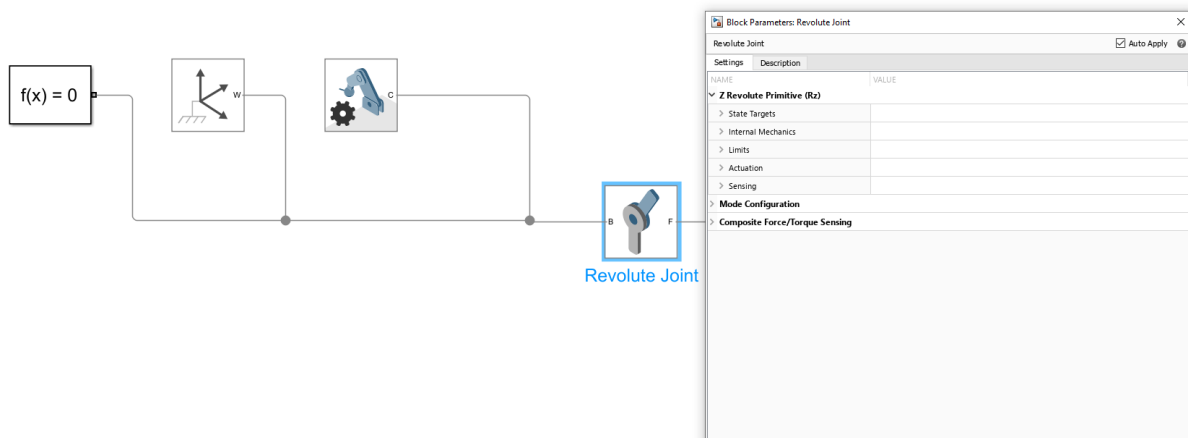
- Adding a Revolute Joint to allow rotation around the z-axis.
- Adjusting the rotation angles to simulate a 90-degree motion around the x-axis and back to the starting position.

Revolute Joint

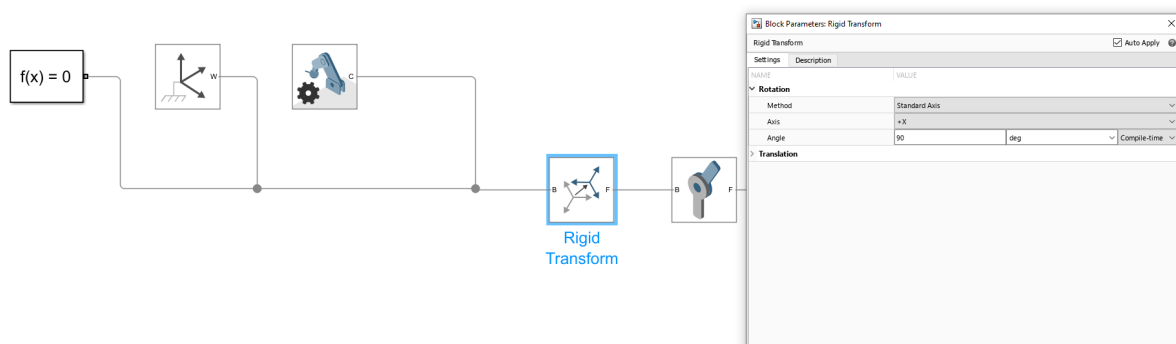
In Simscape Multibody, every joint is defined relative to a local coordinate system that is attached to the connected components.

By convention, the z-axis is used as the axis of motion because it aligns with the principal axis of rotation or translation in the underlying physics engine. This ensures consistency across simulations.

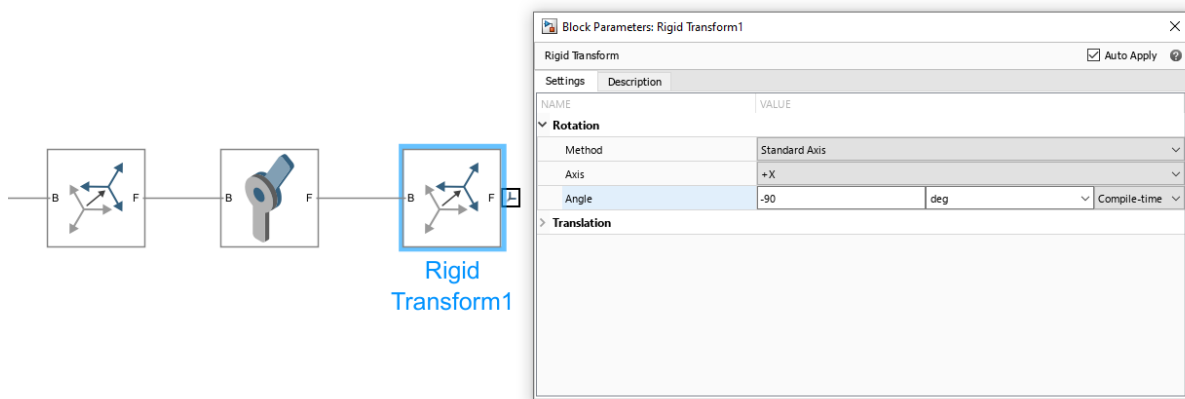
By convention, Simscape Multibody follows a **right-handed coordinate system** where the z-axis is designated for dynamic movements. This provides a standardized approach for integrating multiple components without conflicts in directionality.



90 degree rotation about the x-axis



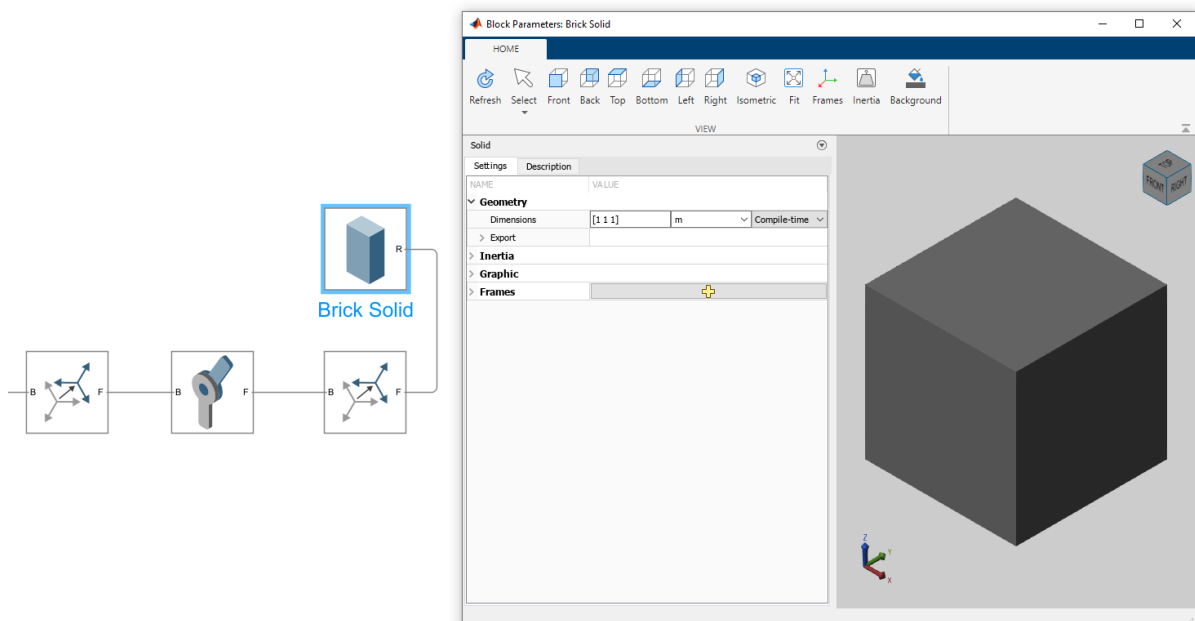
And 90 degrees back again to the original orientation.



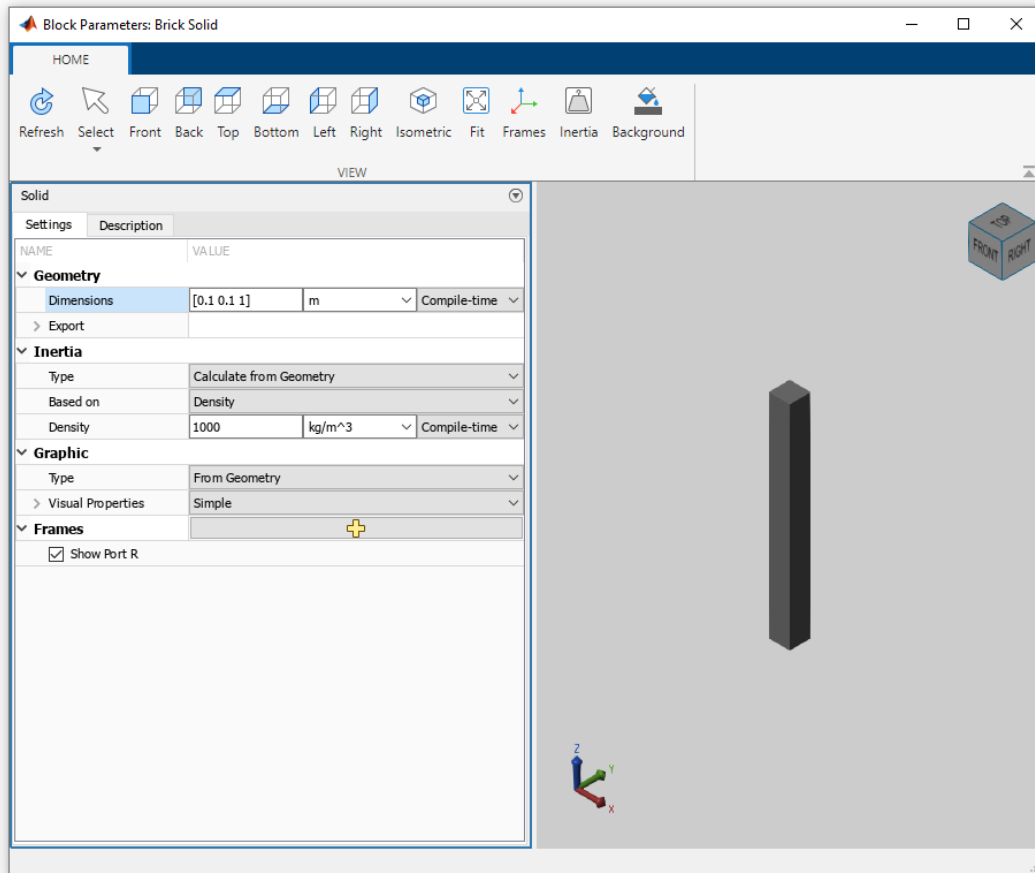
Connecting a Mass

The next step is to attach a mass to the pendulum. For this:

- Insert a rigid body.
- Adjust the mass properties (density, volume, and geometry).
- Make connections intuitive using Simscape Multibody's drag-and-drop functionality.

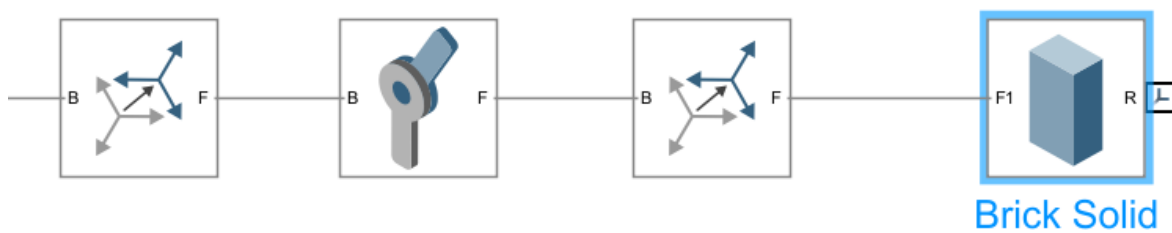
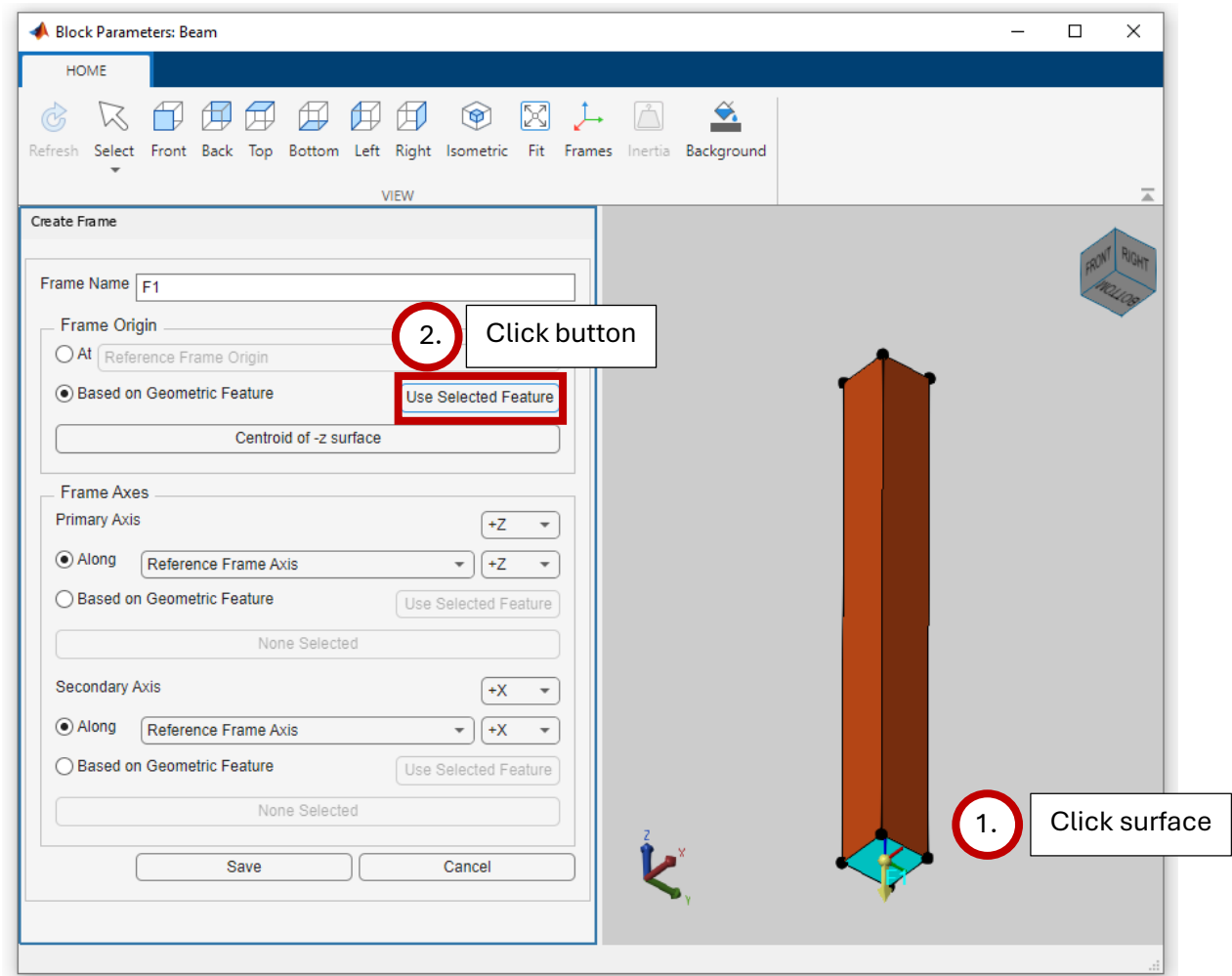


Mass Property Adjustments



Intuitive Connections

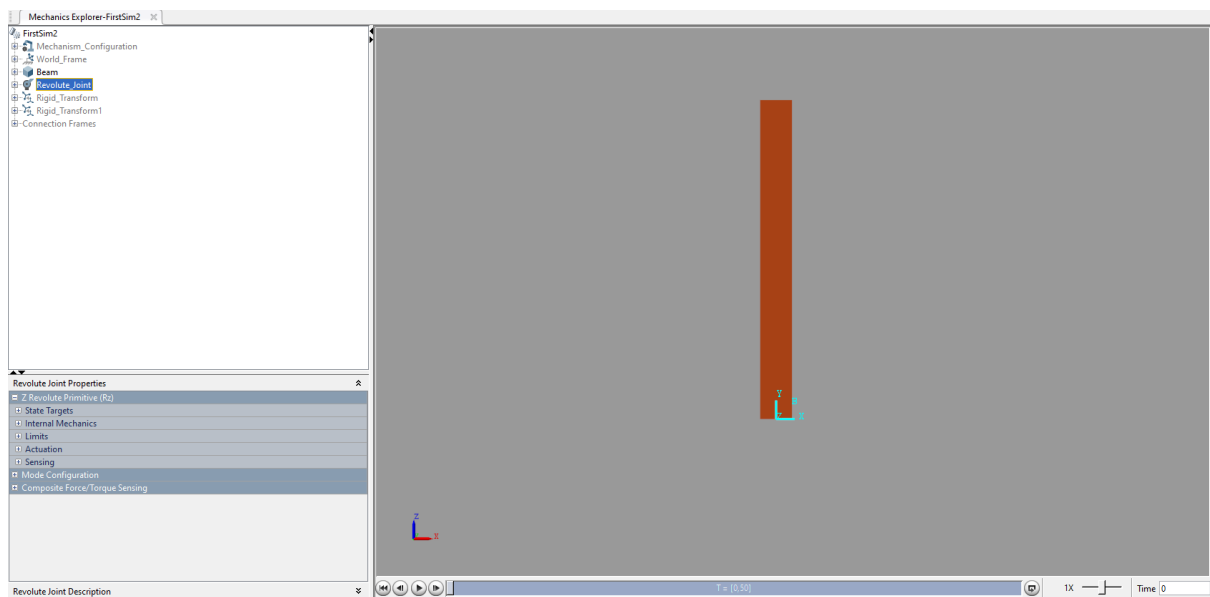
Adding new frames



Running the Simulation

Once the setup is complete, the simulation can be run. Key aspects to observe include:

- Position and orientation of coordinate systems.
- Dynamics of the joints, including rotations and translations.



Setting Initial States

Block Parameters: Revolute Joint

Revolute Joint

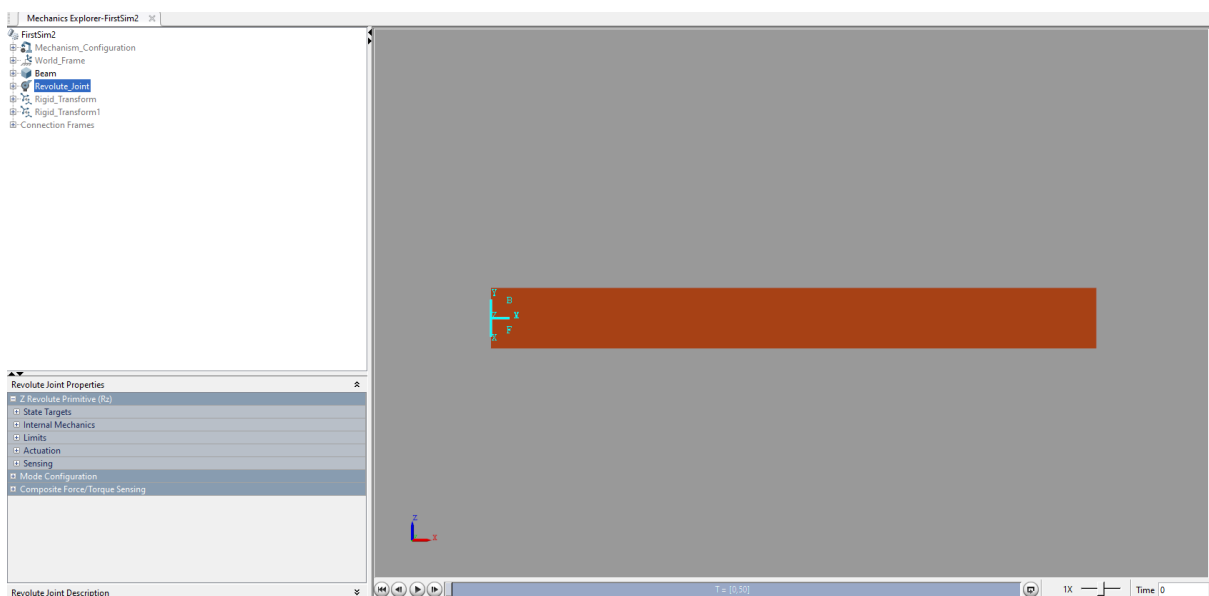
☒ Auto Apply

Settings

Description

NAME	VALUE
▼ Z Revolute Primitive (Rz)	
▼ State Targets	
<input checked="" type="checkbox"/> Specify Position Target	
Priority	High (desired)
Value	-90degCompile-time
<input type="checkbox"/> Specify Velocity Target	
> Internal Mechanics	
> Limits	
> Actuation	
> Sensing	
> Mode Configuration	
> Composite Force/Torque Sensing	

Result



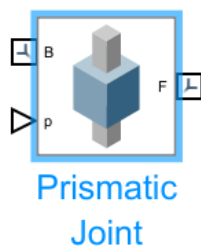
Mechanical Cylinder

Prismatic Joint

We add a cylinder to enable linear motion using:

- Prismatic Joints to allow movement along one axis.
- Actuation and sensing settings to monitor and control position and force.

Settings for Actuation:



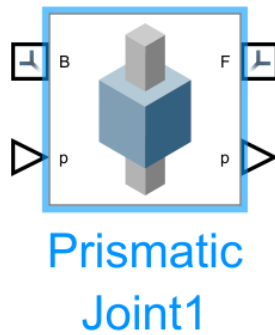
Block Parameters: Prismatic Joint

Prismatic Joint ☒ Auto Apply ?

Settings Description

NAME	VALUE
▼ Z Prismatic Primitive (Pz)	
> State Targets	
> Internal Mechanics	
> Limits	
▼ Actuation	
Force	Automatically Computed
Motion	Provided by Input
> Sensing	
> Mode Configuration	
> Composite Force/Torque Sensing	

Settings for Sensing:



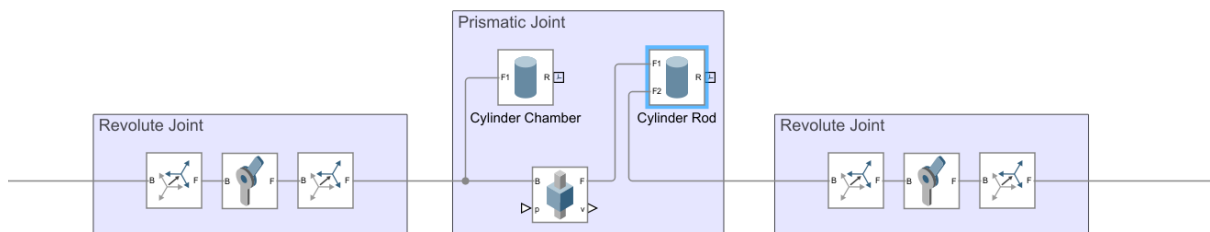
Block Parameters: Prismatic Joint1

Prismatic Joint Auto Apply

Settings Description

NAME	VALUE
Z Prismatic Primitive (Pz)	
> State Targets	
> Internal Mechanics	
> Limits	
> Actuation	
Force	Automatically Computed
Motion	Provided by Input
> Sensing	
<input checked="" type="checkbox"/> Position	
<input type="checkbox"/> Velocity	
<input type="checkbox"/> Acceleration	
<input type="checkbox"/> Actuator Force	
<input type="checkbox"/> Lower-Limit Force	
<input type="checkbox"/> Upper-Limit Force	
> Mode Configuration	
> Composite Force/Torque Sensing	

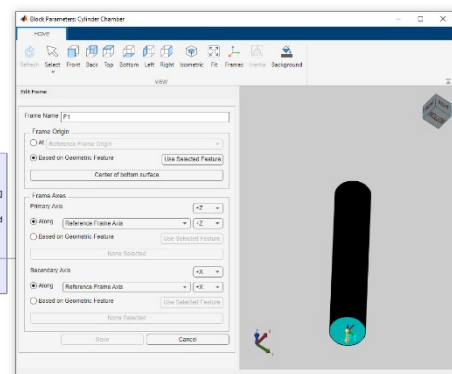
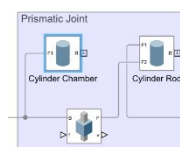
Prismatic Joint with Revolute Joints



Geometry and Frames of the Cylinder

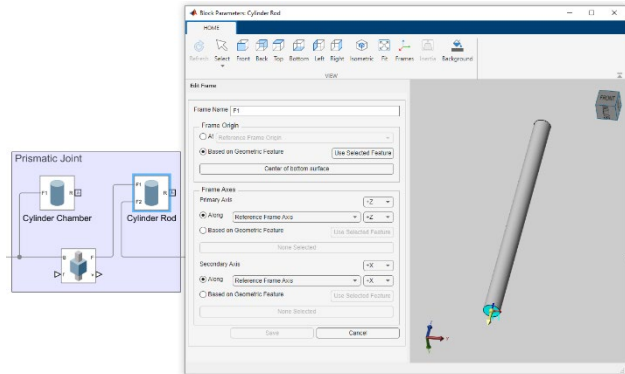
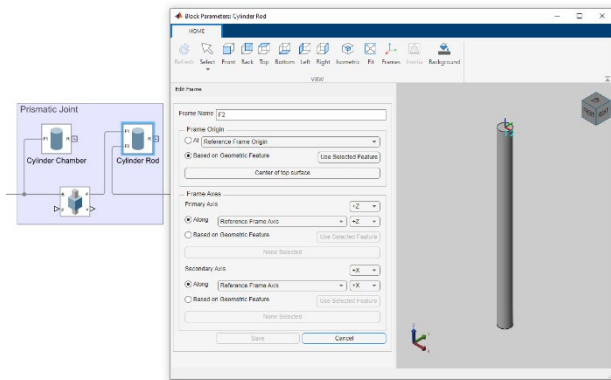
Cylinder Housing

Geometry				
Radius	0.025	m	▼	Compile-time ▼
Length	0.3	m	▼	Compile-time ▼

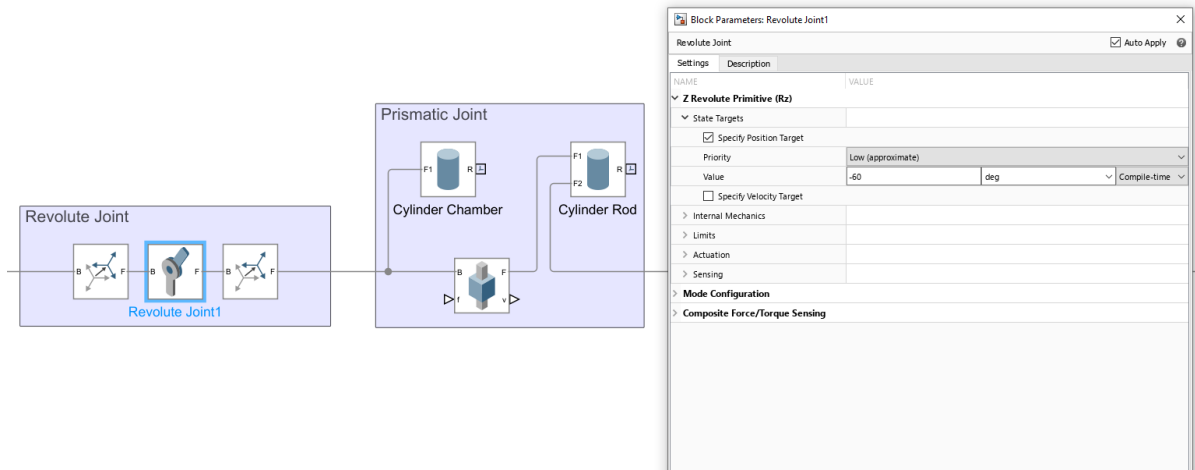


Cylinder Rod

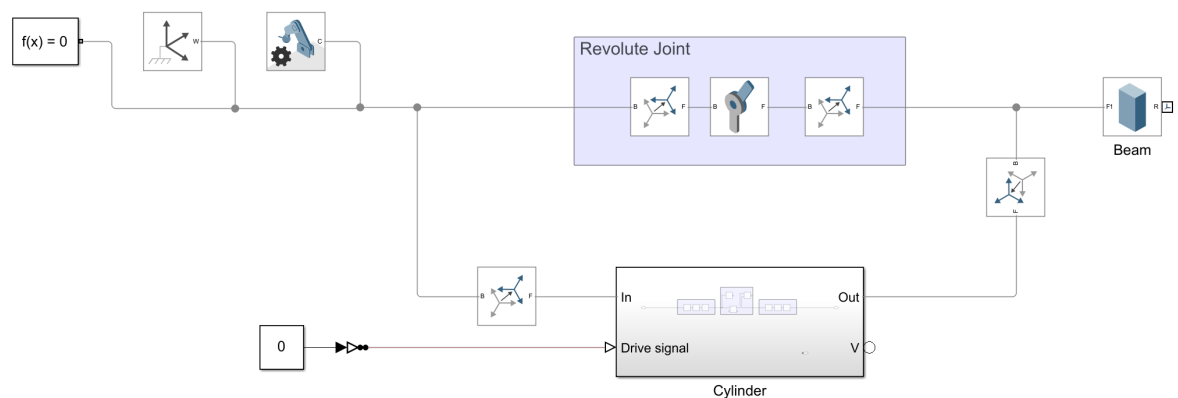
Geometry			
Radius	0.01	m	Compile-time
Length	0.3	m	Compile-time



Setting Initial States

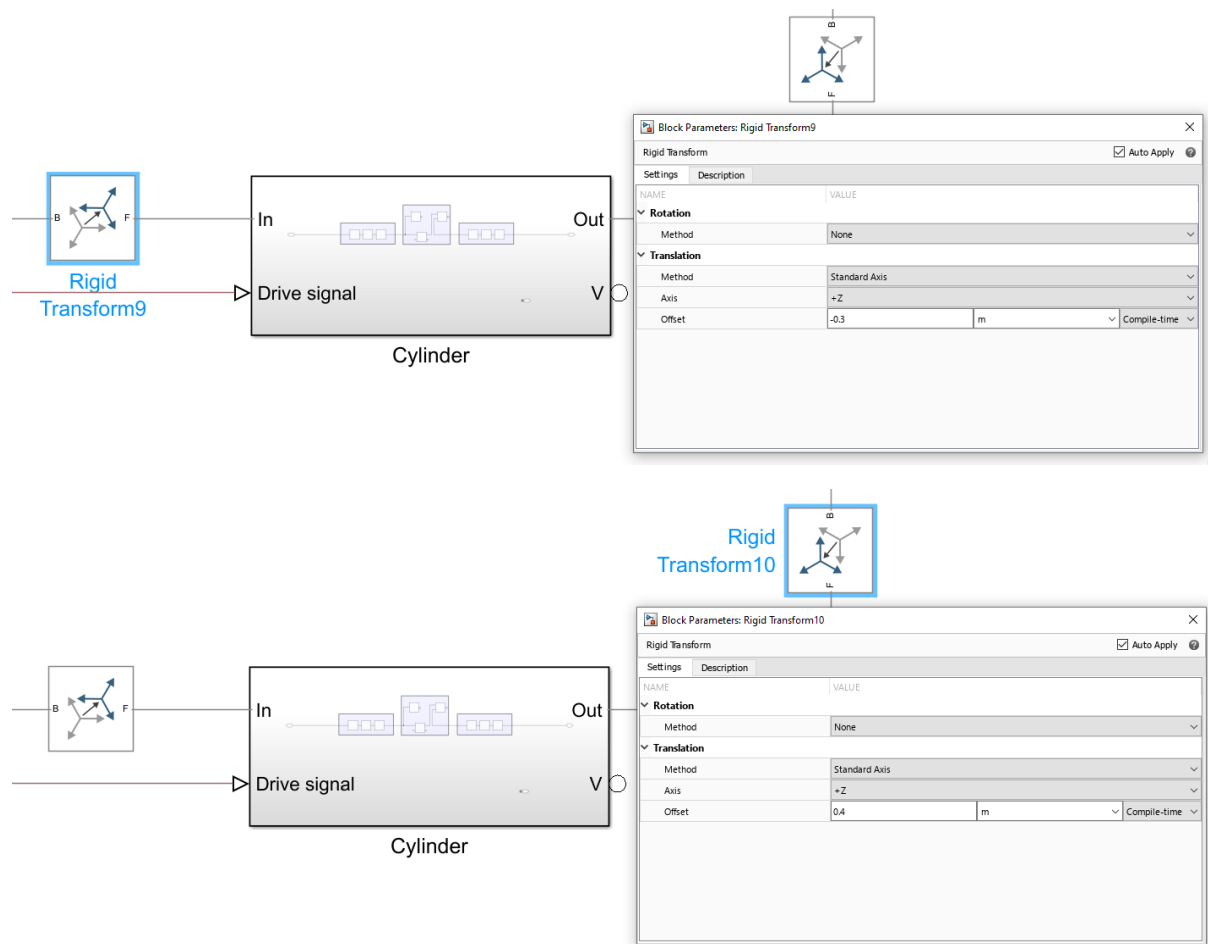


Creating a Sub System for the Cylinder

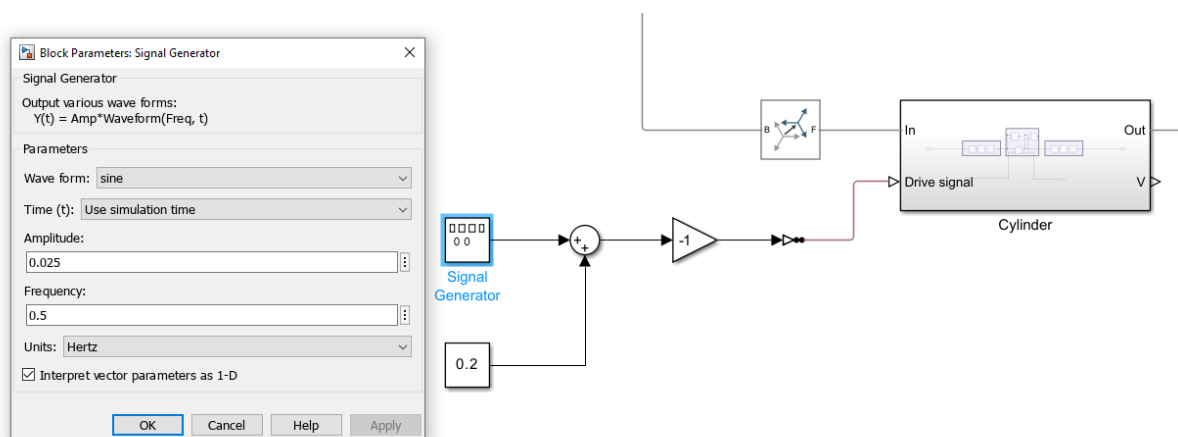


Moving the contact points of the cylinders

The marked transformations in the image above are translations only, moving the endpoints of the cylinder.

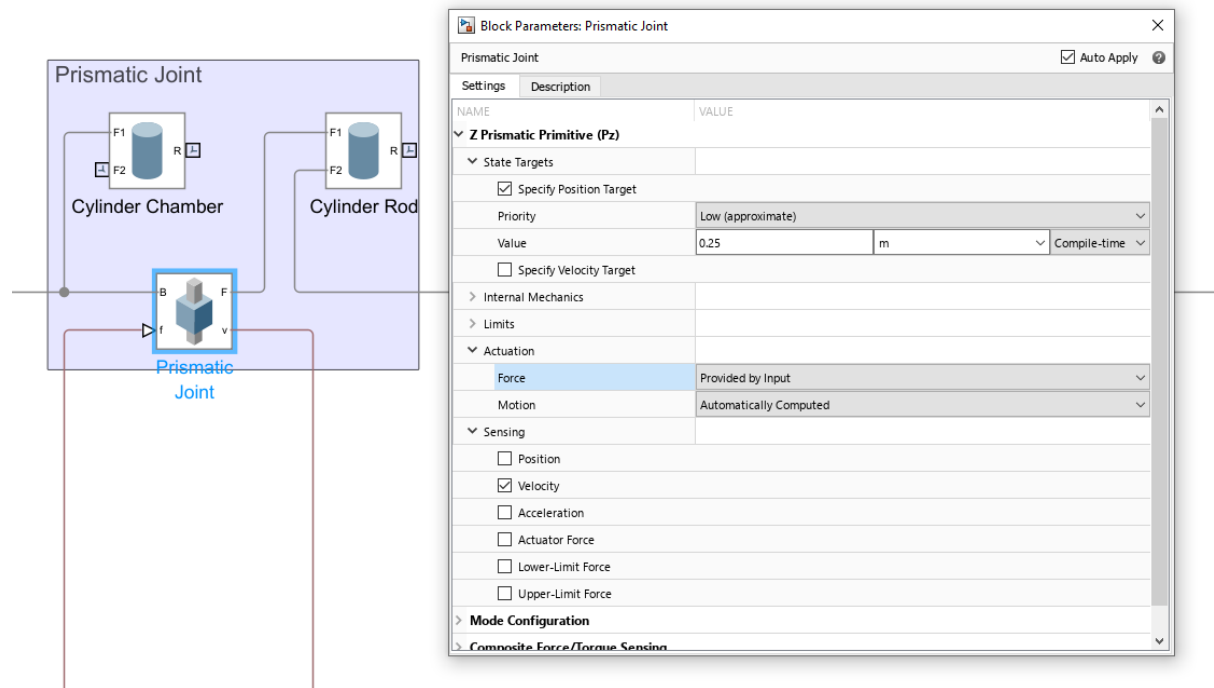


Controlling the Piston Position

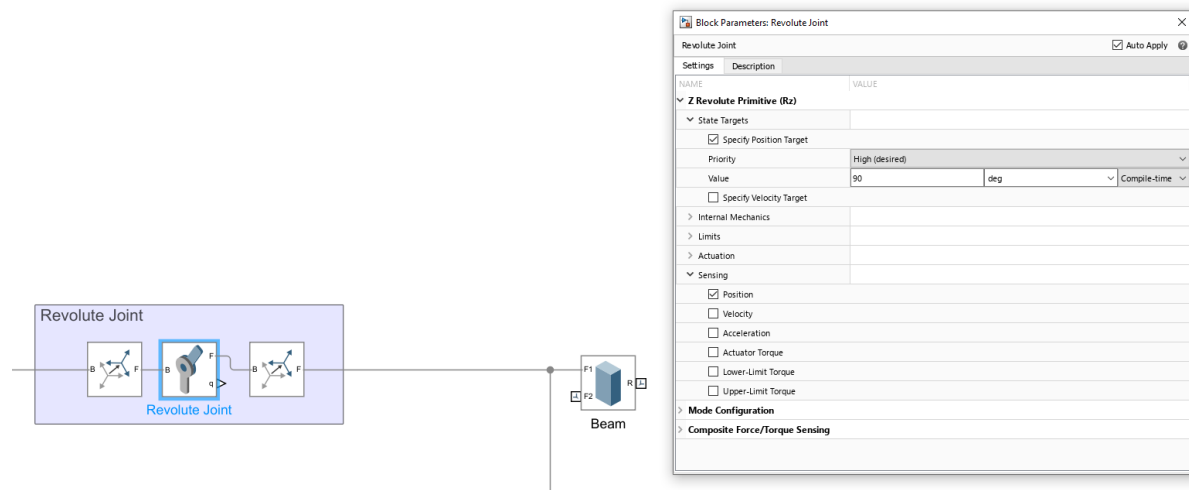


Closed Loop Force Control

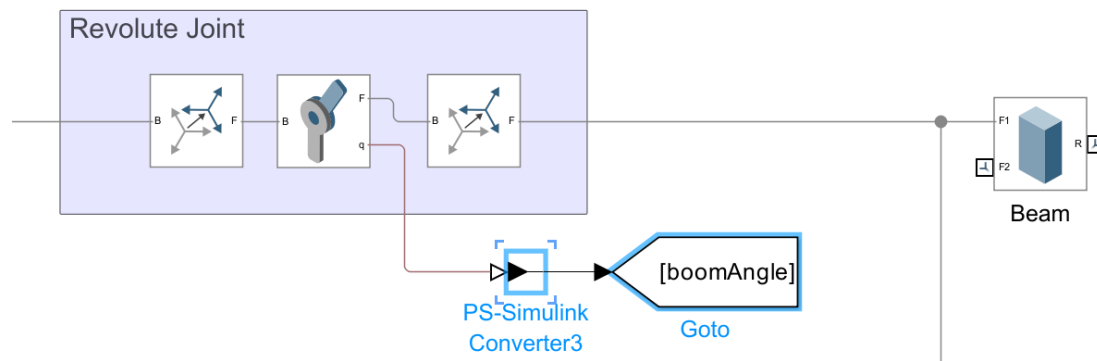
Changing the Control Signal and Feedback of the Cylinder



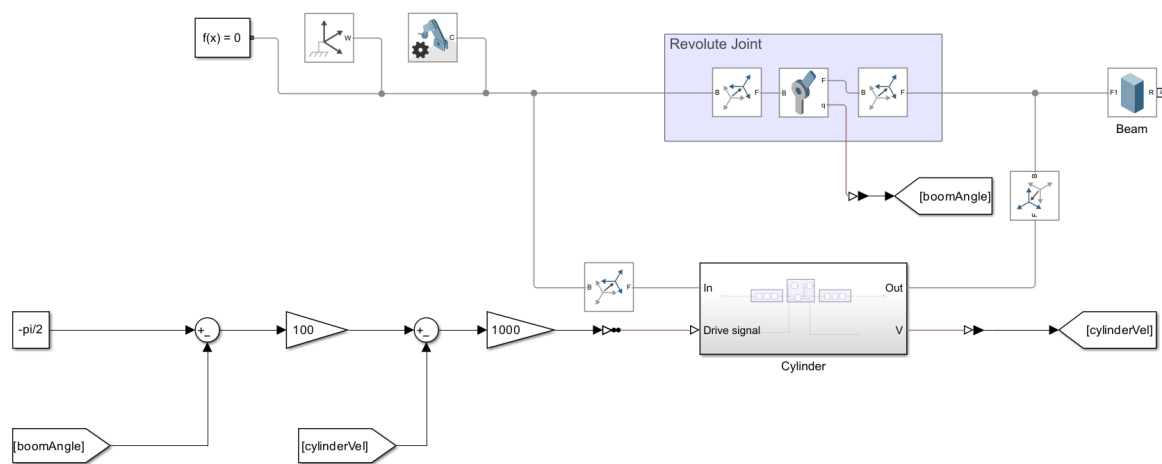
Sensing the Boom Angle



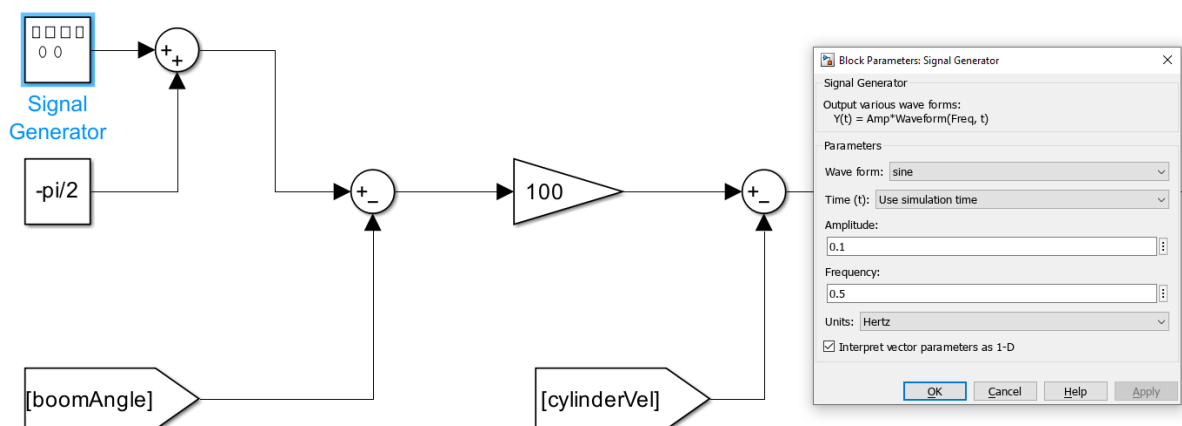
Converting signal and GoTo function to transmit signals



Closed Loop Control

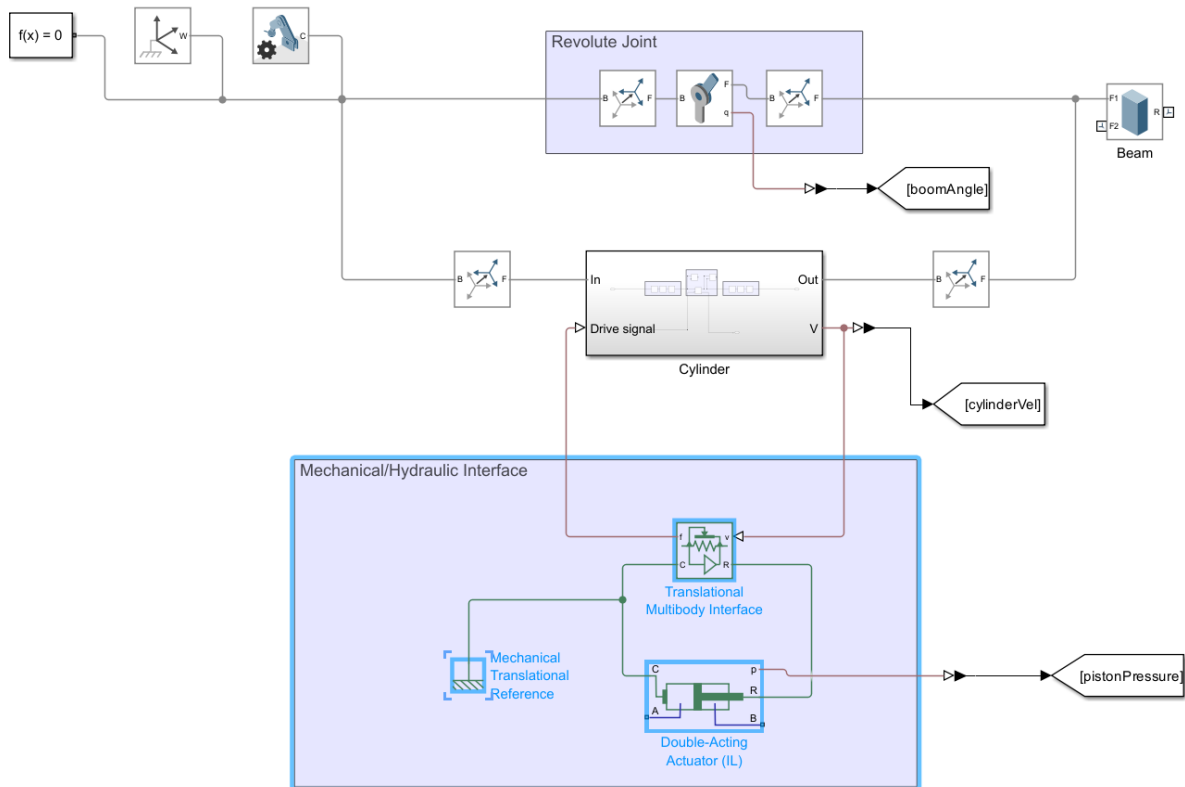


Using a Signal Generator

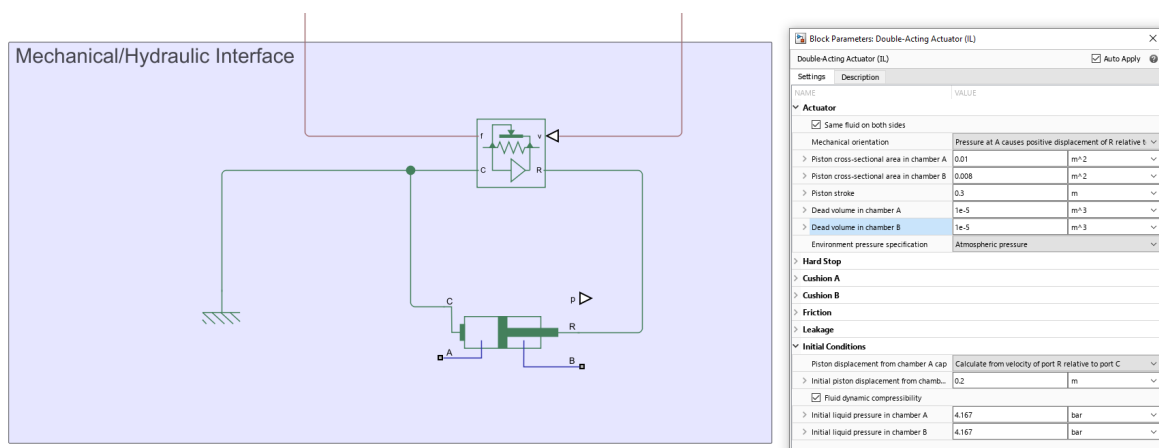


Connecting Hydraulics

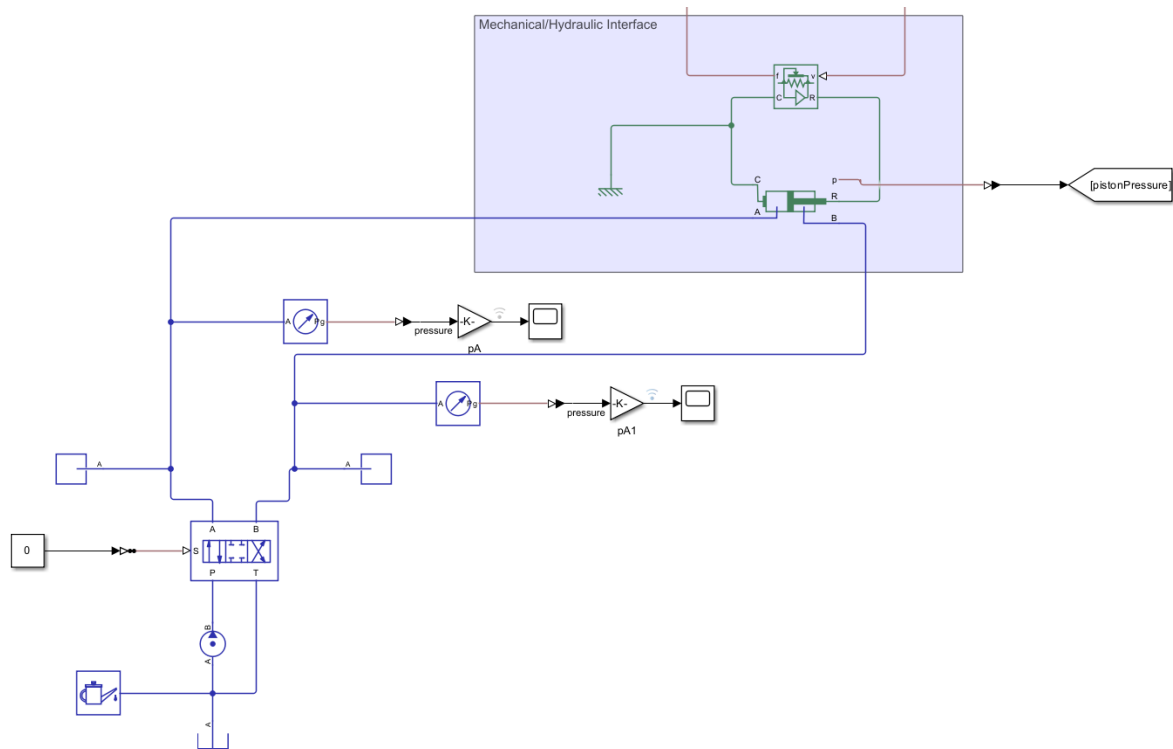
The components we need to connect the mechanical system to the hydraulic system from the Fluids tutorial is listed in the “Mechanical/Hydraulic Interface” section in the figure below.



Parameters of the Cylinder



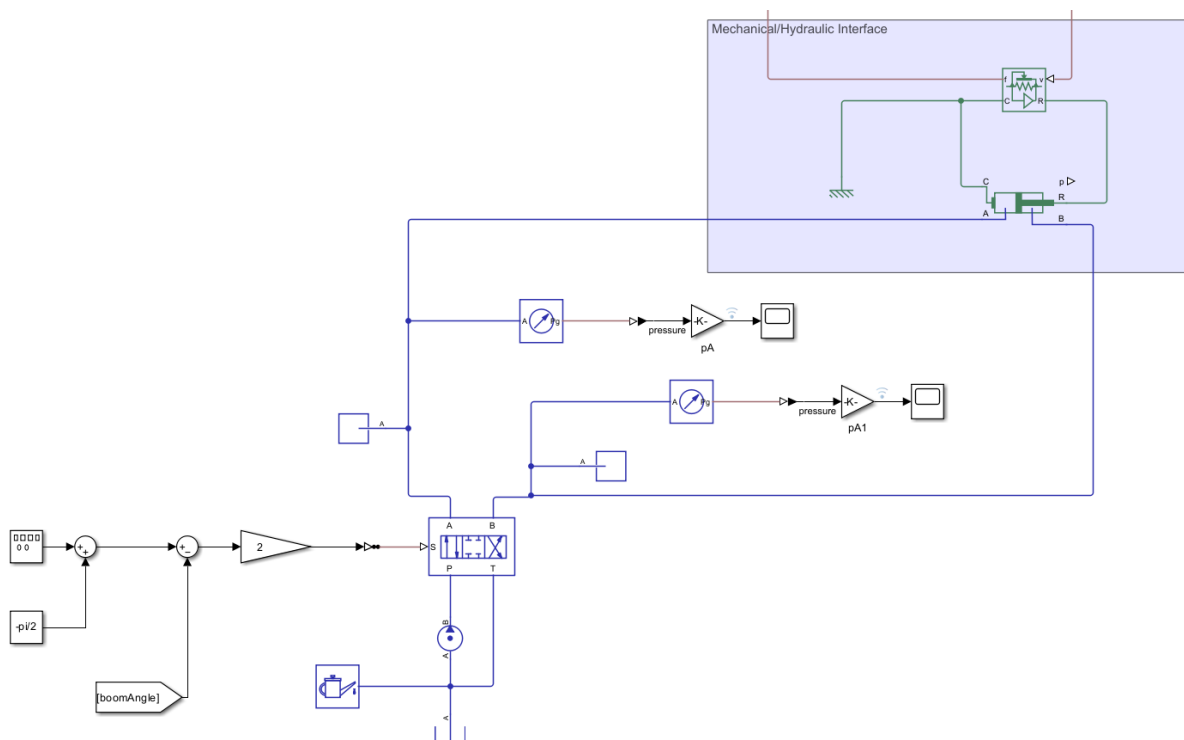
Connecting the Hydraulic System



Initial Condition

Measure the hydraulic pressure at the A port of the cylinder at steady state to set the correct initial pressure in the cylinder.

Hydraulic Closed Loop Control



Results

