Exercises: Simscape™ Fundamentals

Student Competition - Physical Modeling Training

© 2017 by MathWorks, Inc.

Spring-Mass-Damper System

In this exercise, you will model a mass-spring-damper system and determine the power used in each component over time.

1. Create a new model based on the diagram above. Simulate the model for 4 seconds, and plot the mass position over time on a Simulink scope.
2. Enable signal logging for the model, and then simulate the model again.
3. For both springs and the damper, plot the power within the spring and dampers over time. To do this, use the logged signal data.

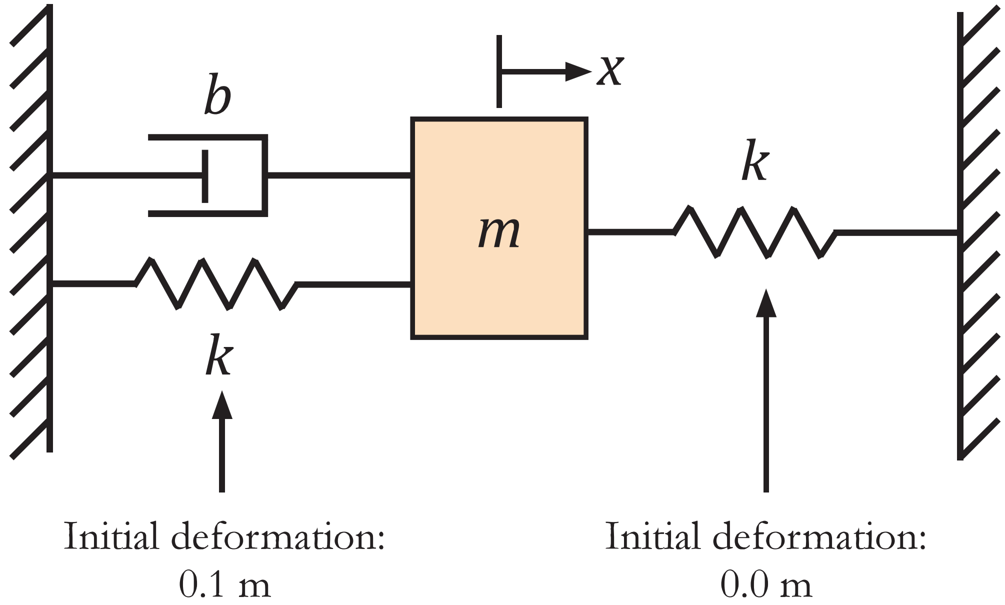
   A. Create a variable representing the logged data for a single component.
   ```
   >> cmp = simlog.Translational_Spring
   ```
   B. Create variables representing the time, force, and velocity
   ```
   >> t = cmp.v.series.time;
   >> vel = cmp.v.series.values;
   >> force = cmp.f.series.values;
   ```
   C. Plot the power over time using the equation $P = F \cdot v$.
   ```
   >> plot(t, force.*vel);
   ```

Parameters

Physical parameters
m = 1 kg
k = 50 N/m
b = 1 N/(m/s)

Initial conditions
Mass initial velocity = 0 m/s
Left spring initial deformation = 0.1 m
Right spring initial deformation = 0.0 m

Simulation parameters
Stop time = 4 s
Solver = `ode23t`



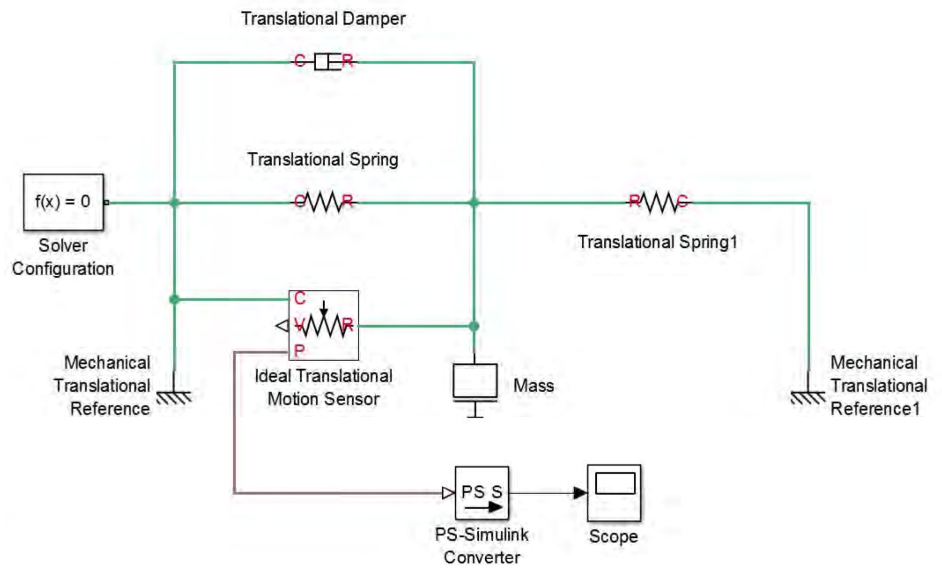
© 2017 by MathWorks, Inc.

Solution: Spring-Mass-Damper System

Model the system as shown in the solution model `twoSpringsSoln`.

Try

```
>> twoSpringsSoln
```

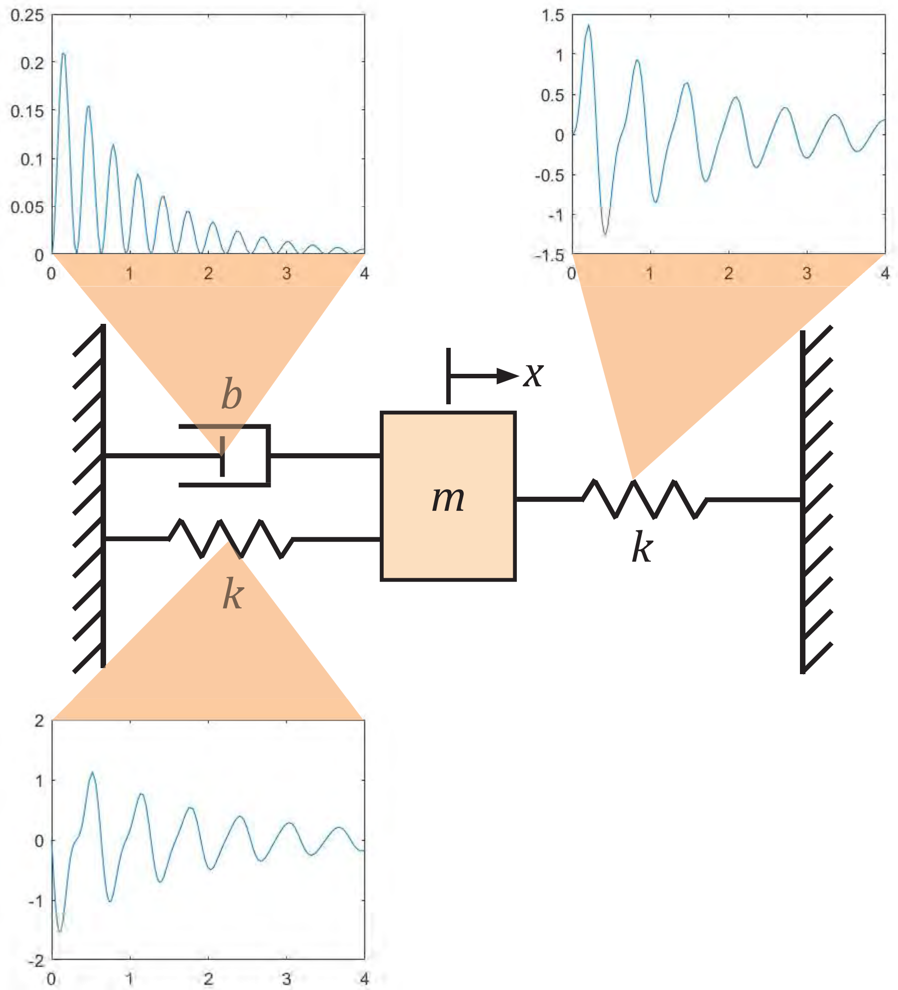


Log signals by setting the **Simulation → Configuration Parameters**. In the **Simscape** section, set the **Log simulation data** option to `All`.

To plot the power in any component, use the MATLAB® commands shown below:

```
>> cmp = simlog.Translational_Spring;
>> t = cmp.v.series.time;
>> vel = cmp.v.series.values;
>> force = cmp.f.series.values;
>> plot(t, force.*vel)
```

As expected, the damper is always dissipating energy. The two springs are at times storing energy and at times releasing energy.



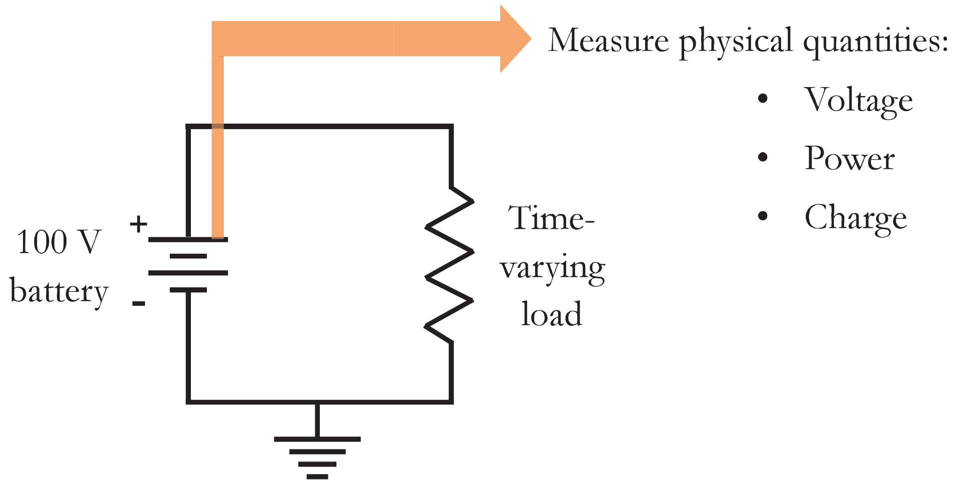
Battery Analysis

In this exercise, you will perform calculations on physical quantities to analyze an electrical circuit containing a battery and a time-varying load.

Starting with the `batteryDischargeStart` model, take the following steps to create a reusable power sensor component.

1. **Measure the battery voltage**.

   Use the appropriate sensor block to measure the battery voltage, in volts. Connect this measurement to the third input port of the Scope block provided.



Try

```
>> batteryDischargeStart
```

2. **Calculate the power supplied by the battery.**

   Recall that electrical power is computed by the expression $P = V \cdot I$. Measure the battery voltage and current and multiply them to give power.

   Connect this measurement to the second input port of the Scope block provided.

3. **Calculate the battery charge in ampere-hours.**

   Recall that charge is found by integrating electric current, so you can use an Integrator block (**Simulink → Continuous** library). Assume that the initial battery charge is 33 ampere-hours. This will be the initial condition of the integrator.

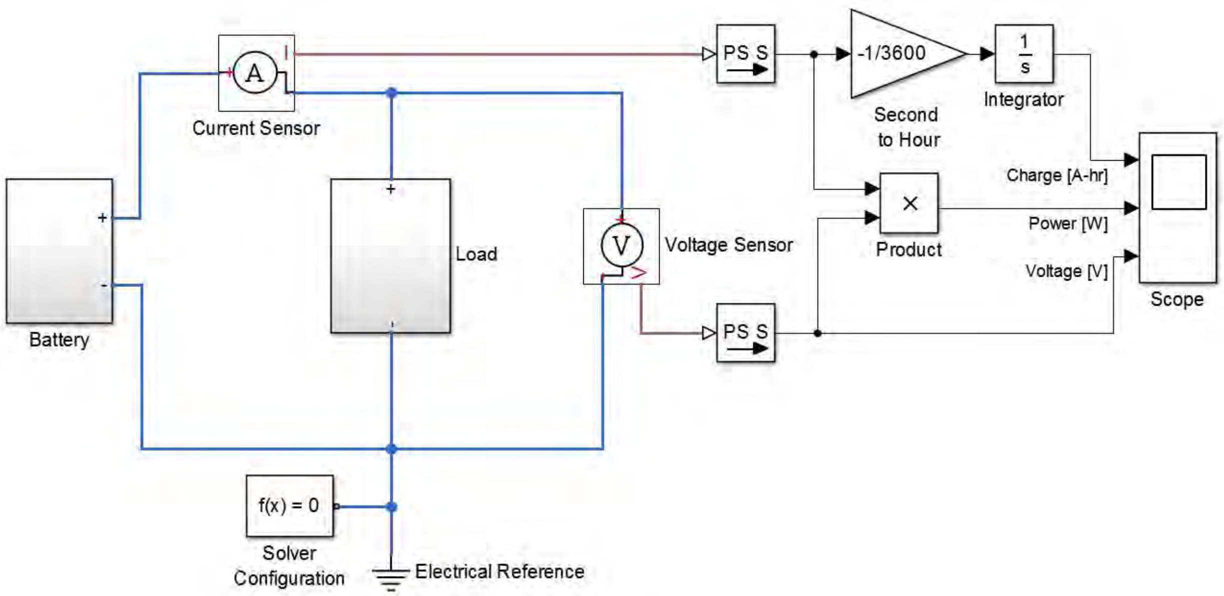
   Hint Time in Simulink is measured in seconds. One ampere-second is equivalent to 1/3600 ampere-hours.

© 2017 by MathWorks, Inc.

Solution: Battery Analysis

1. Connect a Voltage Sensor block in parallel to the battery, and a Current Sensor block in series with the battery. The output of the Voltage Sensor block is the battery voltage.
2. Use a Product block from the **Simulink → Math Operations** library to multiply the outputs of the current and voltage sensors. This will produce the battery power measurement.
3. Use an Integrator block from the **Simulink → Continuous** library to integrate the current measurement for computing battery charge. Set the **Initial condition** parameter of the block to 33.

   Note the factor of -1/3600 to convert from ampere-seconds to ampere-hours. This value is negative because a positive current measured by the sensor means that the battery is discharging.

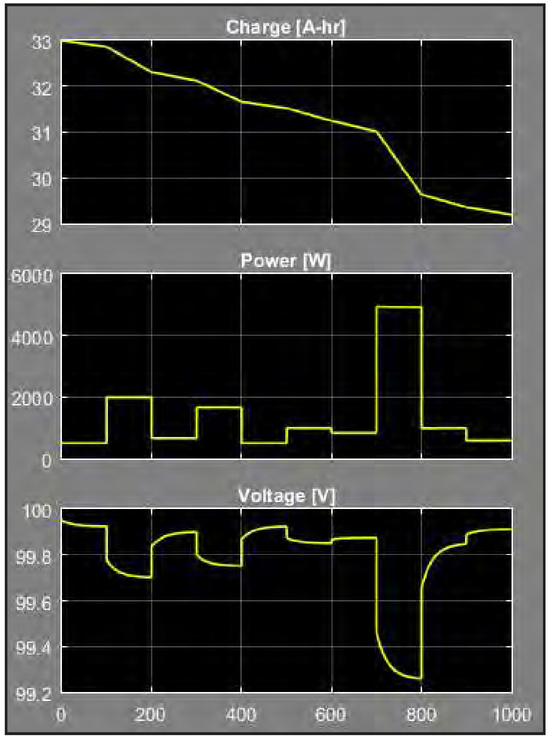


Try

```
>> batteryDischargeSoln
```

As shown in the scope output below, the battery power and voltage varies as the resistive load changes. This changes the rate of battery discharge, which is dictated by the current flowing out of the battery.

At the end of the simulation, the battery charge drops from 33 ampere-hours to about 29 ampere-hours. The battery still has approximately 88 percent charge.



© 2017 by MathWorks, Inc.

Water Turbine

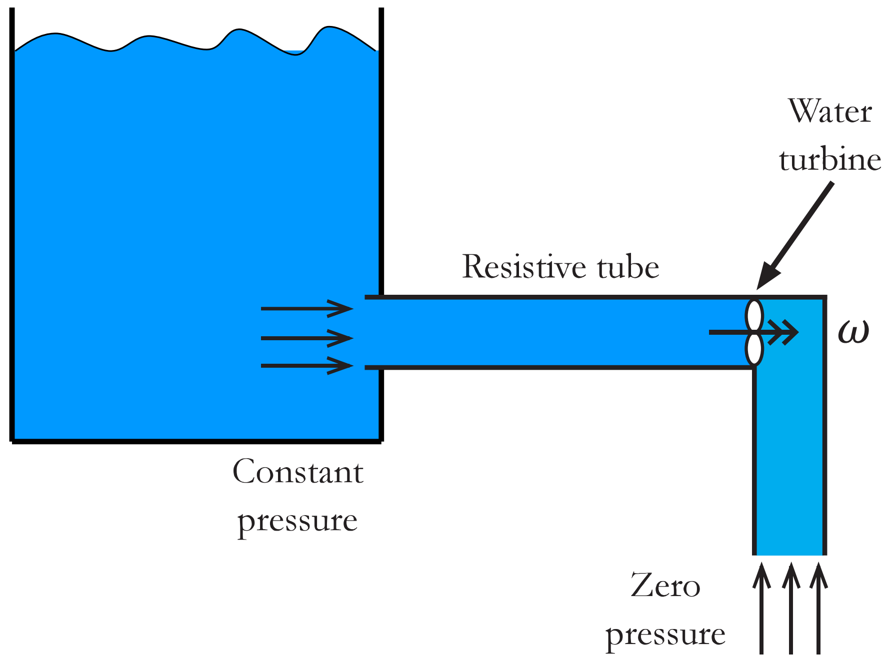
Try

```
>> hydroRotStart
```

In this exercise, you will model the hydromechanical connection to determine the speed at which the water turbine rotates.

In a water turbine, pressure from a water flow is used to rotate the turbine shaft. Starting with the model `hydroRotStart`, take the following steps to create a functional water turbine model.

1. Remove the Ideal Torque Source block and use a Rotational Hydro-Mechanical Converter block to model an ideal hydro-mechanical connection. Keep in mind two things:
   - It takes a volume of 0.0003 m^3 of water to rotate the shaft 1 radian.
   - The turbine case is rigidly connected to the ground, and does not spin at all.
2. Record the maximum flow rate that the water obtains.
3. Make the connection nonideal by using a rotational damper to model viscous friction of the water turbine shaft. Use a damping coefficient of 0.001 N*m/(rad/s).
4. Try setting the initial velocity of the turbine shaft to zero and the initial flow rate of the resistive tube to 0.001 m^3/s. Is there a conflict if both targets are high priority? Why?



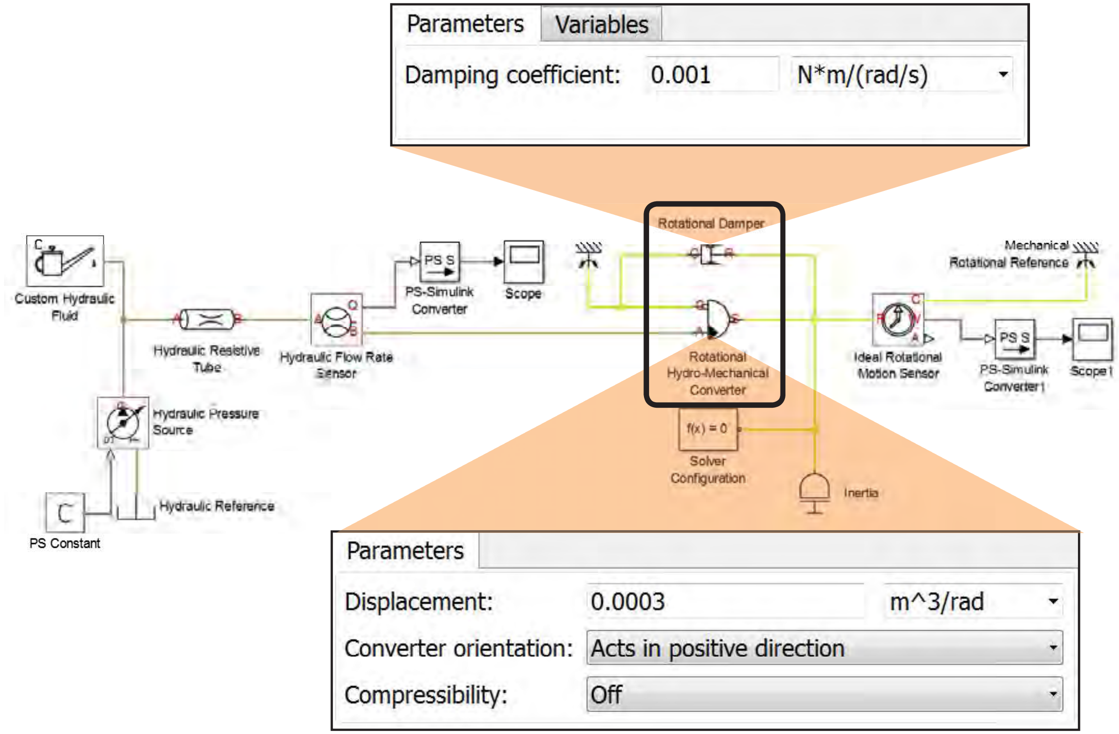
© 2017 by MathWorks, Inc.

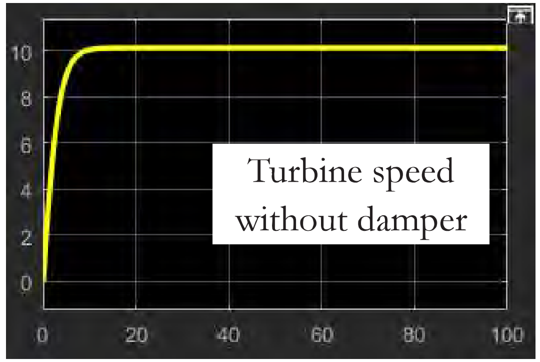
Solution: Water Turbine

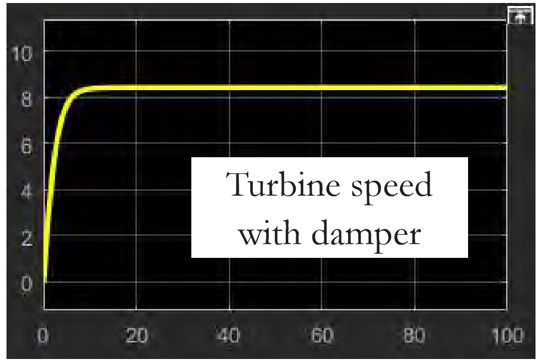
Try

```
>> hydroRotSoln
```

Replace the Ideal Torque Source block with the Rotational Hydro-Mechanical Converter block as shown below. Disconnect the outlet of the Hydraulic Resistive Tube block and the Hydraulic Reference block – if you leave them connected, there is no pressure at the port that can rotate the turbine.







The flow rate in the resistive tube and the velocity of the turbine shaft are coupled by the Rotational Hydro-Mechanical Converter block. Note that the units of the converter displacement are in m^3/rad, which can also be expressed as $(m^3/s)/(rad/s)$.

You can verify the initial conditions using the Variable Viewer (**Analysis → Simscape → Variable Viewer**), as shown to the right.

Name	Status	Priority	Target	Start	Unit
Hydraulic Resistive Tube					
q		High	1000.0	999.9888890123...	cm^3/s
A					
B					
p				1.41203E-4	bar
Inertia					
w		High	0.0	3.333296296707...	rad/s
I					
t				0.0224306	N*m
Custom Hydraulic Fluid					

© 2017 by MathWorks, Inc.