Simscape™

Getting Started Guide

MATLAB® & SIMULINK®

R2015b

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.
3 Apple Hill Drive
Natick, MA 01760-2098

Simscape™ Getting Started Guide

Revision History

September 2007	Online only	New for Version 2.0 (Release 2007b)
March 2008	Online only	Revised for Version 2.1 (Release 2008a)
October 2008	Online only	Revised for Version 3.0 (Release 2008b)
March 2009	Online only	Revised for Version 3.1 (Release 2009a)
September 2009	Online only	Revised for Version 3.2 (Release 2009b)
March 2010	Online only	Revised for Version 3.3 (Release 2010a)
September 2010	Online only	Revised for Version 3.4 (Release 2010b)
April 2011	Online only	Revised for Version 3.5 (Release 2011a)
September 2011	Online only	Revised for Version 3.6 (Release 2011b)
March 2012	Online only	Revised for Version 3.7 (Release 2012a)
September 2012	Online only	Revised for Version 3.8 (Release 2012b)
March 2013	Online only	Revised for Version 3.9 (Release 2013a)
September 2013	Online only	Revised for Version 3.10 (Release 2013b)
March 2014	Online only	Revised for Version 3.11 (Release 2014a)
October 2014	Online only	Revised for Version 3.12 (Release 2014b)
March 2015	Online only	Revised for Version 3.13 (Release 2015a)
September 2015	Online only	Revised for Version 3.14 (Release 2015b)

Contents

1 Product Fundamentals

Product Fundamentals

Simscape Product Description

Model and simulate multidomain physical systems

Simscape provides an environment for modeling and simulating physical systems spanning mechanical, electrical, hydraulic, and other physical domains. It provides fundamental building blocks from these domains that you can assemble into models of physical components, such as electric motors, inverting op-amps, hydraulic valves, and ratchet mechanisms. Because Simscape components use physical connections, your models match the structure of the system you are developing.

Simscape models can be used to develop control systems and test system-level performance. You can extend the libraries using 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 physical system in Simulink®. To deploy your models to other simulation environments, including hardware-in-the-loop (HIL) systems, Simscape supports C-code generation.

Key Features

- Single environment for modeling and simulating mechanical, electrical, hydraulic, thermal, and other multidomain physical systems
- Libraries of physical modeling blocks and mathematical elements for developing custom components
- MATLAB based Simscape language, enabling text-based authoring of physical modeling components, domains, and libraries
- Physical units for parameters and variables, with all unit conversions handled automatically
- Ability to simulate models that include blocks from related physical modeling products without purchasing those products
- Support for C-code generation

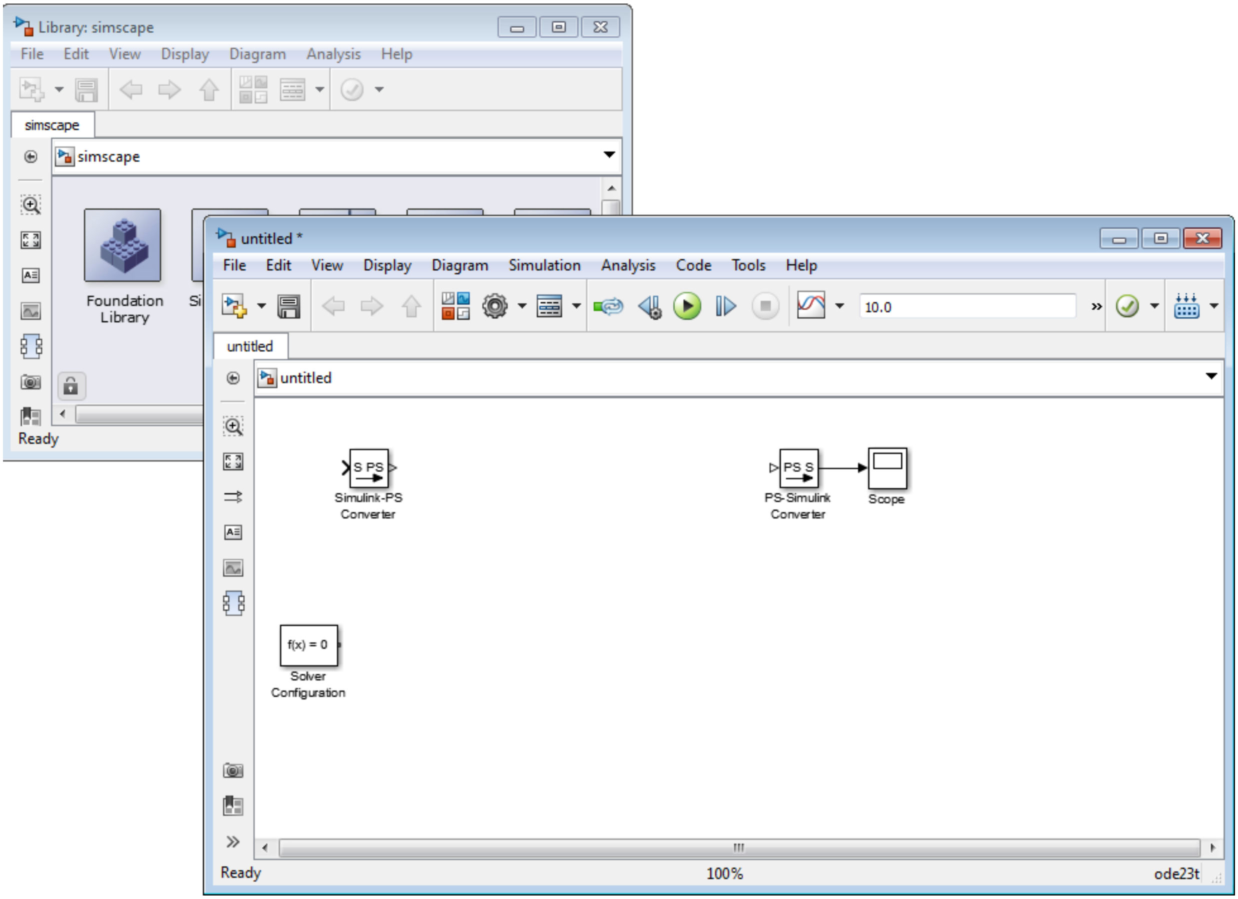
Creating a New Simscape Model

In this section...

Recommended Blocks and Solvers

Simscape models require certain blocks to be present in the model configuration, such as a Solver block, or domain-specific reference blocks. Other blocks, although not required, are highly likely to be needed, such as Simulink-PS Converter and PS-Simulink Converter blocks. An easy way to start a new Simscape model is by using the `ssc_new` function.

When you type `ssc_new` at the MATLAB Command prompt, the software opens the main Simscape library and creates a new model prepopulated with certain blocks, as shown in the following illustration.

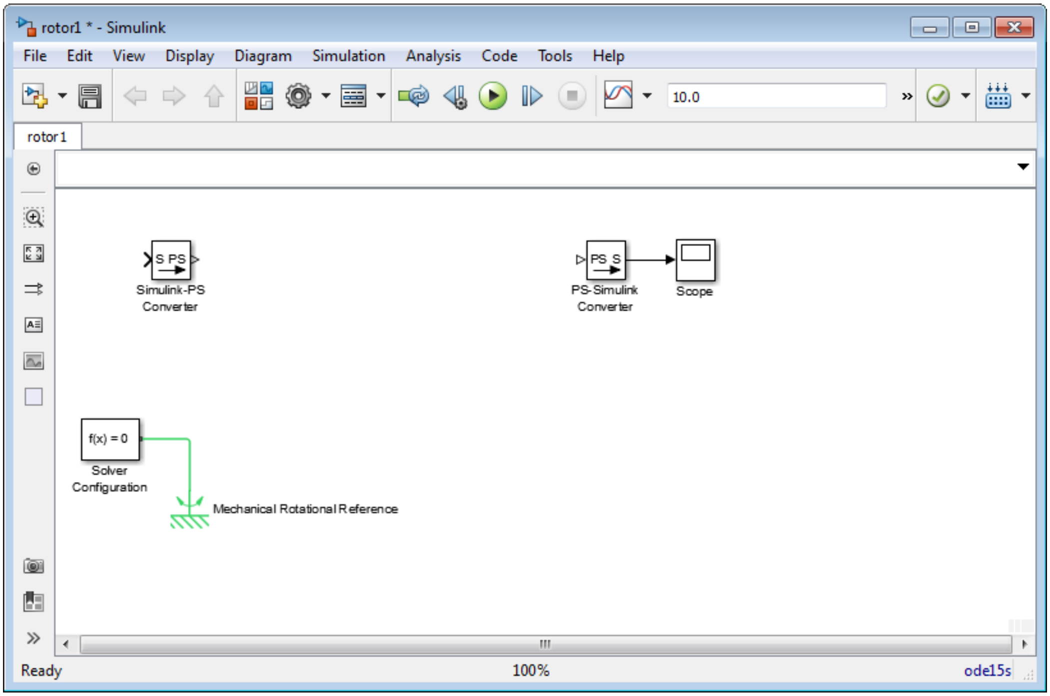


By default, the model name is not specified, the model contains a Solver Configuration block with the default solver set to `ode23t`, a Simulink-PS Converter block, and a PS-Simulink Converter block connected to a Scope block.

You can use the `ssc_new` function arguments to specify the model name, add a domain-specific reference block, and change the default solver. See the `ssc_new` reference page for details. For example, typing

```
ssc_new('rotor1','rotational','ode15s')
```

creates the following model.



After using `ssc_new`, continue developing your model by copying the blocks, as needed, and adding other blocks from the Simscape libraries.

For electrical models, you can also use the Creating A New Circuit example as a template for a new model. This example creates a new electrical model and opens an Electrical Starter Palette, which contains links to the most often used electrical components. Open the example by typing `ssc_new_elec` in the MATLAB Command Window and use **File > Save As** to save it under the desired model name. Then delete the unwanted components and add new ones from the Electrical Starter Palette and from Simscape libraries.

Data Logging Settings for New Model

Using data logging is a best practice for Simscape models because it provides access to important simulation and analysis tools. Therefore, the `ssc_new` function automatically turns on data logging for the whole model. It uses the default workspace variable name simlog to store simulation data, and limits the data to the last 10000 points to avoid slowing down simulation.

When you create a new model using ssc_new, the model has the following data logging configuration:

- **Log simulation data** — All.
- **Log simulation statistics** — Off.
- **Open viewer after simulation** — Off.
- **Workspace variable name** — simlog.
- **Decimation** — 1.
- **Limit data points** — On.
- **Data history (last N steps)** — 10000.

For information on what these settings mean and how to change them, see “Data Logging Options”.

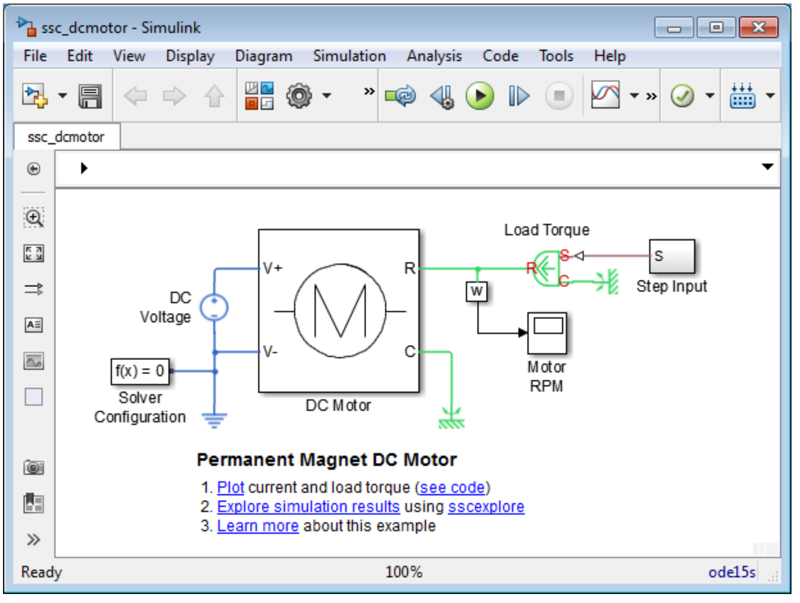
Evaluating Performance of a DC Motor

This example shows how to simulate systems that span electrical and mechanical domains. You learn how to model physical components with Simscape blocks, connect them into a realistic model, use Simulink blocks as well, and then simulate and modify a motor model.

The model is based on a Faulhaber Series 0615 DC-Micromotor. The model uses equivalent circuit parameters for the 1.5V motor to verify manufacturer-quoted no-load speed, no-load current, and stall torque. You can use the model to assess motor performance in a given application by adding the requisite mechanical load model.

Explore the Model

1 To open the Permanent Magnet DC Motor example model, type `ssc_dcmotor` in the MATLAB Command Window.



Main Model Window

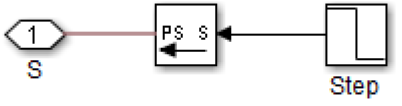
The main model window contains a DC Motor subsystem with two electrical and two mechanical rotational ports.

For improved readability of block diagrams, each Simscape domain uses a distinct default color and line style for the connection lines. In this block diagram, for example, the electrical circuit is indicated by the dark-blue color of the connection lines, while the connection lines between the mechanical rotational ports are light-green. Physical signal lines are brown.

The electrical ports of the motor connect to the electrical circuit, which consists of an Electrical Reference block, representing an electrical ground, and a 1.5 V DC voltage source.

On the mechanical side, a Mechanical Rotational Reference block represents a reference point for the other elements.

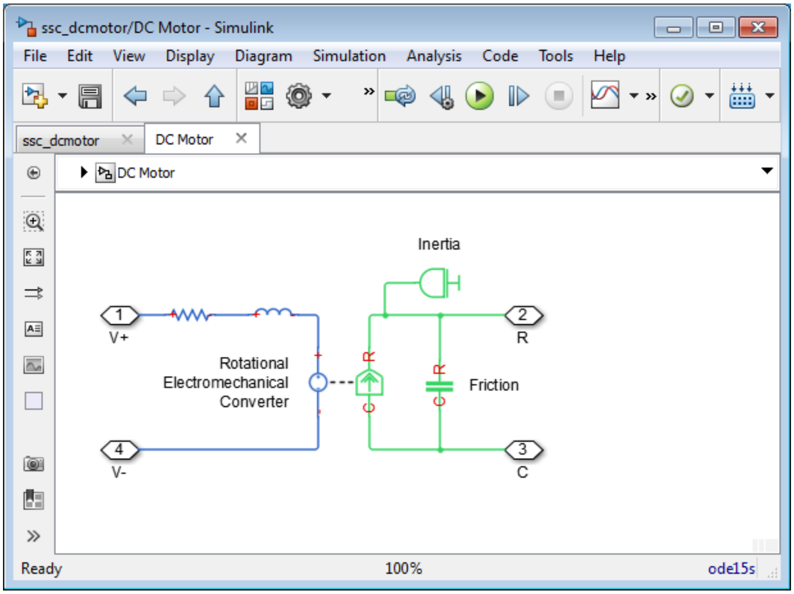
The motor load is represented by the block named Load Torque, which is an Ideal Torque Source block. On one side it is connected to a Mechanical Rotational Reference block and on the other side to the motor shaft. The load torque is specified by the Step Input subsystem, connected through a physical signal to the control port S of the Load Torque block.



The Step Input subsystem contains a regular Simulink Step source, which provides the control signal. A Simulink-PS Converter block converts the control signal into a physical signal and applies it to the control port of the Load Torque block through the Connection Port block S.

The diagram also contains a Solver Configuration block, which is required in any Simscape model. It contains parameters relevant to numerical algorithms for Simscape simulations.

2 Double-click the DC Motor subsystem to open it.



DC Motor Subsystem

The motor consists of an electrical circuit and a mechanical rotational circuit, connected by the Rotational Electromechanical Converter block. The electrical circuit consists of a Resistor block and an Inductor block. It contains two electrical ports, corresponding to the V+ and V- electric terminals of the motor. The mechanical circuit contains a Rotational Friction block, an Inertia block, and two mechanical rotational ports, C and R, corresponding to the motor case and rotor, respectively. Notice how the C and R ports of the Friction block and the Rotational Electromechanical Converter block are connected to the C and R ports of the motor, to preserve the correct direction of variables in the Physical Network.

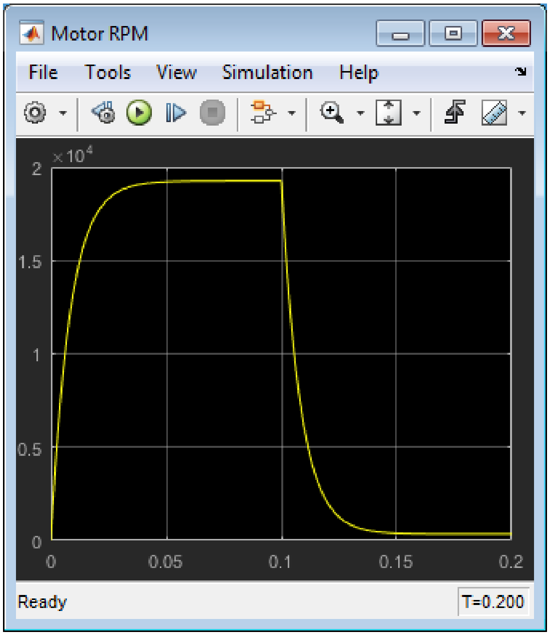
Run the Model

1. Double-click the Motor RPM scope to open it. During simulation, this window displays the shaft speed as a function of time.

2. In the toolbar of the model window, click  to start the simulation. The Simscape solver evaluates the model, calculates the initial conditions, and runs the simulation.

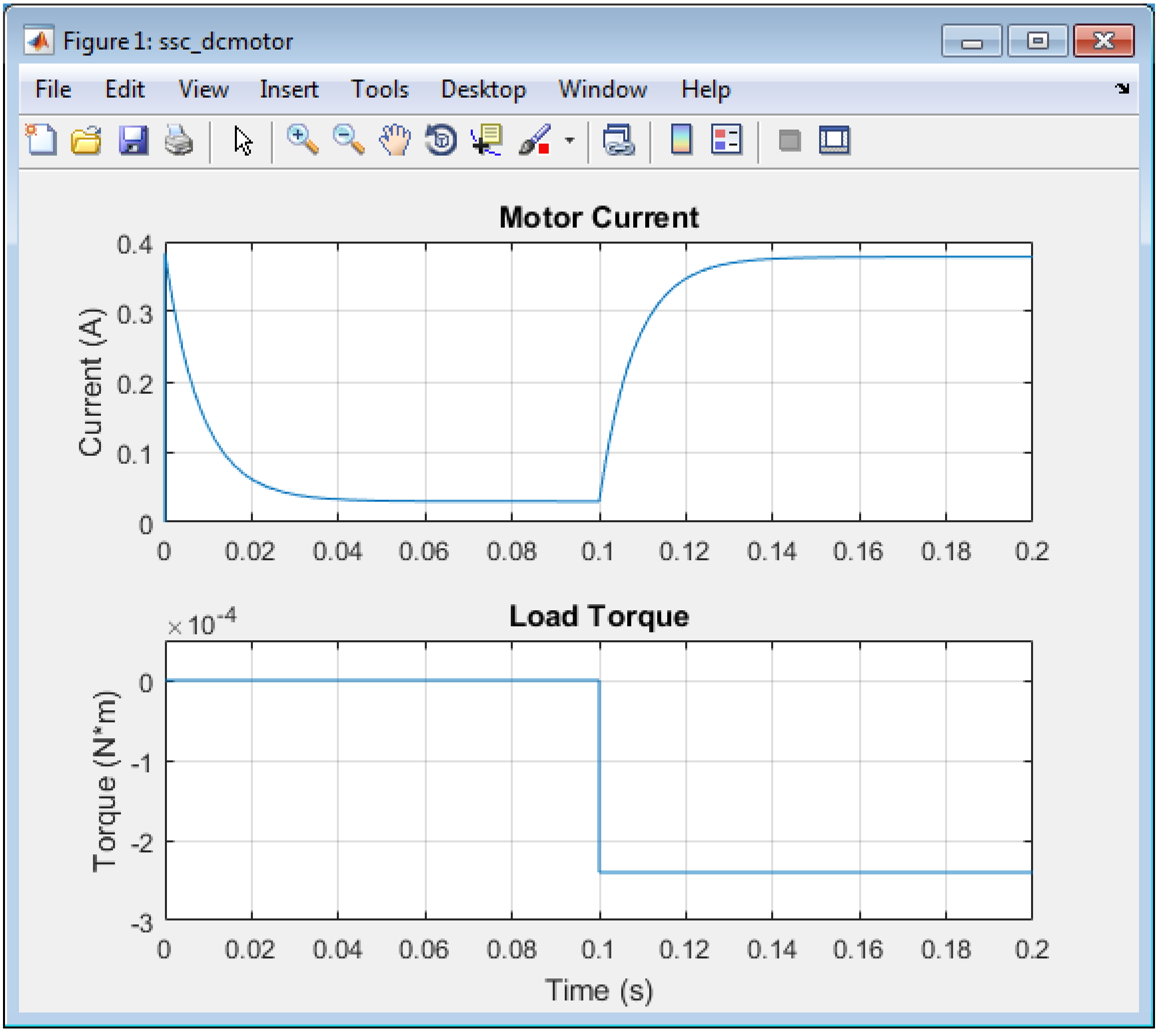
This process might take a few seconds. The message in the bottom-left corner of the model window provides the status.

3 Examine the simulation results in the Motor RPM scope window.



For the first 0.1 seconds, the motor has no external load, and the speed builds up to the no-load value. Then at 0.1 seconds, the stall torque is applied as a load to the motor shaft. Zooming in on the Motor RPM scope shows that the model matches the manufacturer parameters for no-load speed and stall torque.

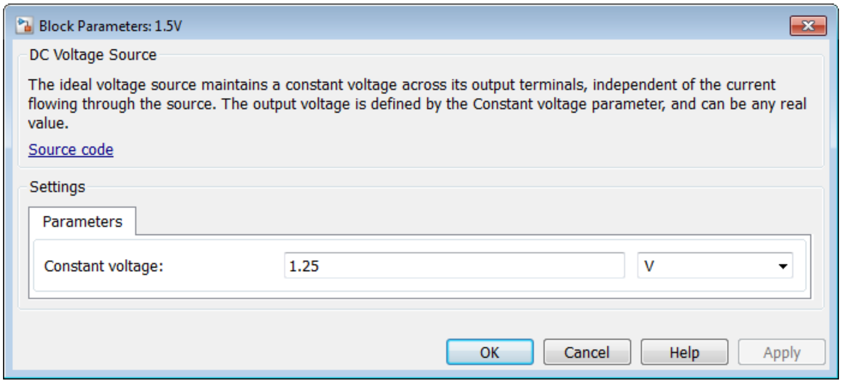
4 The example model also shows how you can use MATLAB code to analyze the simulation results. To plot the current and load torque, click the `Plot` hyperlink in the model annotation. The current shown in the figure matches the manufacturer parameters for no-load current.



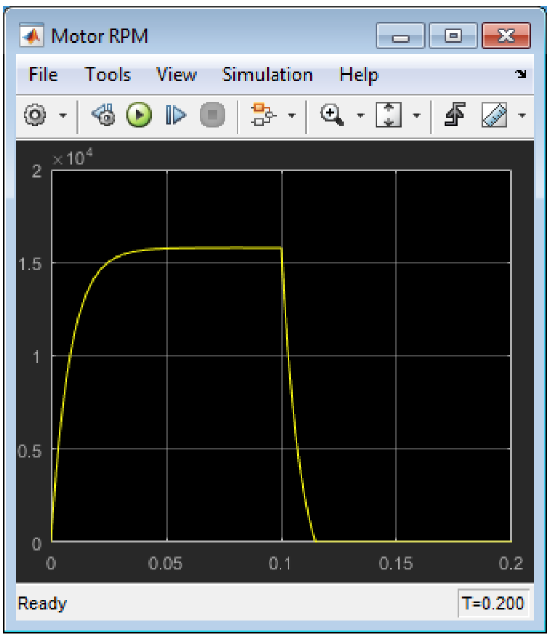
Change the Supply Voltage

Reduce the supply voltage to 1.25 volts (to simulate the battery running down) and vary the load torque to find the maximum torque at this reduced voltage.

1 Double-click the DC Voltage Source block. Set **Constant voltage** to `1.25 V`.

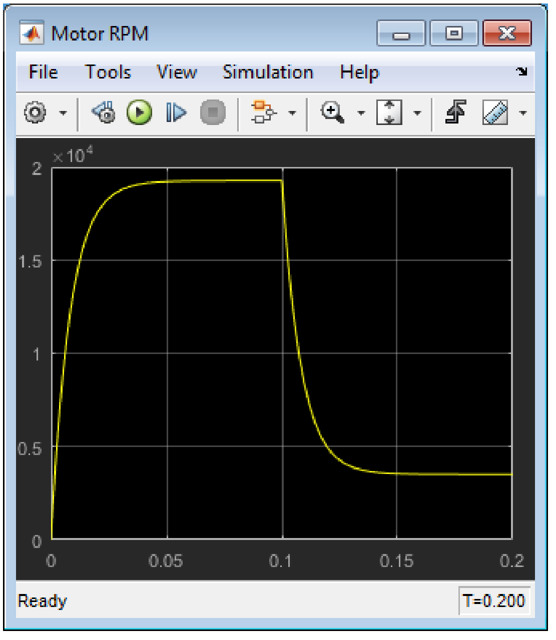


2 Run the simulation. Note the effect of reduced voltage on the no-load speed.



3 Try varying the load torque to find the maximum torque at this reduced voltage. Open the Step Input subsystem and double-click the Step source block. Enter different final values for the input signal and rerun the simulation.

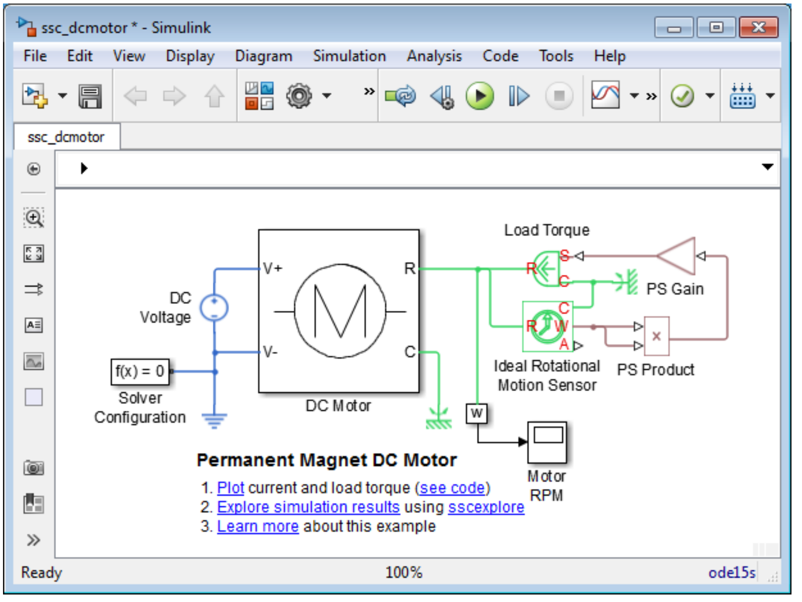
The next illustration shows the simulation results for **Final value** set to `-0.2e-3`, which corresponds to `(1.25/1.5)*0.24mNm`, as the magnitude of the torque-speed curve is proportional to voltage for a DC motor.



Change the Motor Load

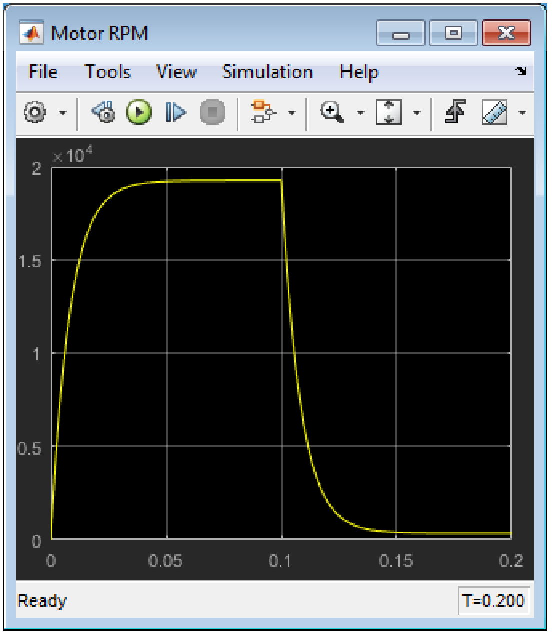
Replace the torque source with a simple mechanical load, for example, a fan, for which the torque is defined by `alpha*speed^2`, where `alpha` is `-1e-10` Nm/(rad/s)^2.

1. Delete the Step Input subsystem from the model.
2. In the **Simscape** block library, open **Foundation Library** > **Mechanical** > **Mechanical Sensors**.
3. Drag the Ideal Rotational Motion Sensor block to the model window.
4. Open **Foundation Library** > **Physical Signals** > **Functions**.
5. Drag the PS Product block and the PS Gain block to the model window.
6. Connect the blocks as shown in the following illustration. To rotate a block, select it and press **Ctrl+R**.



7 Double-click the Gain block to open its dialog box. Enter **Gain** value of `-1e-10` and click **OK**.

8 Run the simulation and assess motor performance with the new load.


Motor RPM
File Tools View Simulation Help
×10^4
2
1.5
1
0.5
0
0 0.05 0.1 0.15 0.2
Ready
T=0.200