

# **PSpice MATLAB Co-Simulation Tutorial**

**Product Version 23.1**  
**September 2023**

© 2023 Cadence Design Systems, Inc. All rights reserved.

Portions © Apache Software Foundation, Sun Microsystems, Free Software Foundation, Inc., Regents of the University of California, Massachusetts Institute of Technology, University of Florida. Used by permission. Printed in the United States of America.

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

Product PSpice contains technology licensed from, and copyrighted by: Apache Software Foundation, 1901 Munsey Drive Forest Hill, MD 21050, USA © 2000-2005, Apache Software Foundation. Sun Microsystems, 4150 Network Circle, Santa Clara, CA 95054 USA © 1994-2007, Sun Microsystems, Inc. Free Software Foundation, 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA © 1989, 1991, Free Software Foundation, Inc. Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, © 2001, Regents of the University of California. Daniel Stenberg, © 1996 - 2006, Daniel Stenberg. UMFPACK © 2005, Timothy A. Davis, University of Florida, (davis@cise.ulf.edu). Ken Martin, Will Schroeder, Bill Lorensen © 1993-2002, Ken Martin, Will Schroeder, Bill Lorensen. Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts, USA © 2003, the Board of Trustees of Massachusetts Institute of Technology. All rights reserved.

**Trademarks:** Trademarks and service marks of Cadence Design Systems, Inc. contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 800.862.4522. Open SystemC, Open SystemC Initiative, OSCI, SystemC, and SystemC Initiative are trademarks or registered trademarks of Open SystemC Initiative, Inc. in the United States and other countries and are used with permission.

MATLAB and SimuLink are registered trademarks of MathWorks. All other trademarks are the property of their respective holders.

**Restricted Permission:** This publication is protected by copyright law and international treaties and contains trade secrets and proprietary information owned by Cadence. Unauthorized reproduction or distribution of this publication, or any portion of it, may result in civil and criminal penalties. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. Unless otherwise agreed to by Cadence in writing, this statement grants Cadence customers permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used only in accordance with a written agreement between Cadence and its customer.
2. The publication may not be modified in any way.
3. Any authorized copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement.
4. The information contained in this document cannot be used in the development of like products or software, whether for internal or external use, and shall not be used for the benefit of any other party, whether or not for consideration.

**Disclaimer:** Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information. Cadence is committed to using respectful language in our code and communications. We are also active in the removal and/or replacement of inappropriate language from existing content. This product documentation may however contain material that is no longer considered appropriate but still reflects long-standing industry terminology. Such content will be addressed at a time when the related software can be updated without end-user impact.

**Restricted Rights:** Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor.

---

# Contents

---

<u>Introduction</u>	5
<u>Design Database</u>	5
<u>Software Requirements</u>	5
<u>Settings</u>	6
<u>References</u>	7
<u>Using PSpice MATLAB Interface</u>	9
<u>Overview of Sample Design Files</u>	10
<u>Completing the Schematic</u>	10
<u>Deciding Interfaces</u>	15
<u>Running Simulation and Viewing Results</u>	19
<u>Fine Tuning Parameters</u>	21

## PSpice MATLAB Co-Simulation Tutorial

---

---

# Introduction

---

This tutorial uses the example of an electronic cruise system to walk you through the various steps to perform co-simulation using the PSpice MATLAB Interface.

## Design Database

The `./examples` folder contains the [PSpice\\_COSim.zip](#) file that contains the design and model files used in this tutorial.

Extract the archive on your system and use the appropriate design or model file to perform the steps.

## Software Requirements

You need to have the following combination of The Mathworks and Cadence® OrCAD® and Allegro® products installed on your system:

1. The Mathworks products

*MATLAB R2019b* or onwards

2. Cadence OrCAD products or Cadence Allegro products

Install one of the following design entry tools:

- ☐ Capture or Capture CIS
- ☐ Design Entry HDL

Install the following simulator:

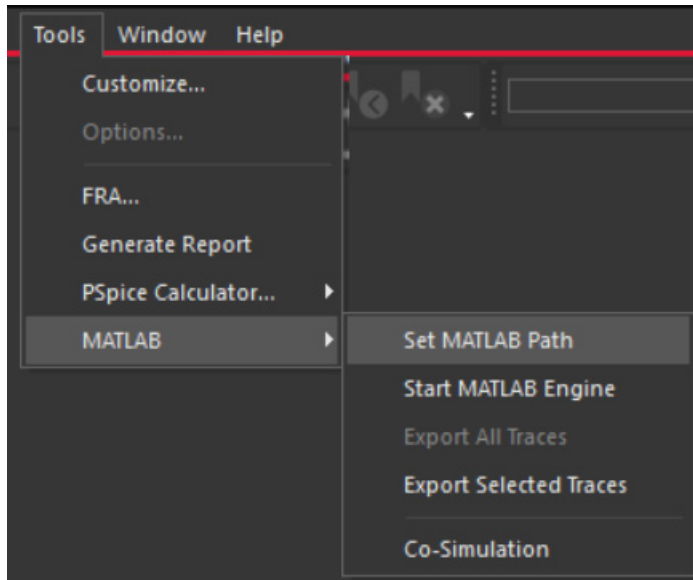
- ☐ PSpice A/D

3. Ensure that you have the license for one of these products in your license file.

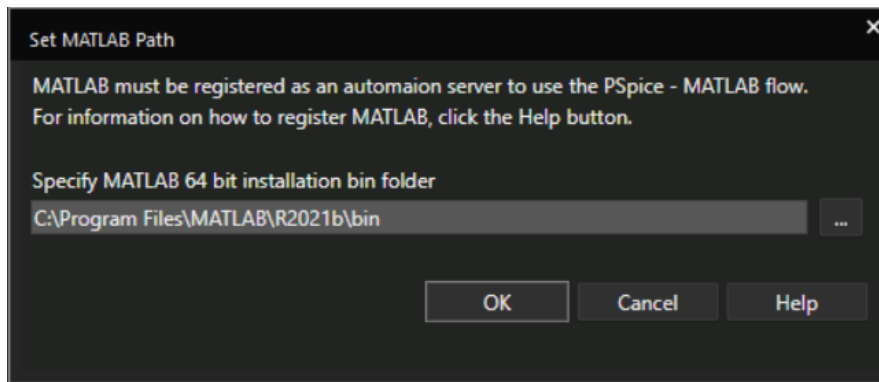
- ☐ OrCAD PSpice Designer Plus
- ☐ Allegro PSpice Systems Simulator

## Settings

Set the MATLAB path in PSpice using *Tools – MATLAB – Set MATLAB Path*.



The MATLAB path should end at the `bin` folder, such as  
`<MATLAB_installation>\bin`.



Remember the following points related to setting up the MATLAB path:

- ☐ The MATLAB path need to be set only once in the Set MATLAB Path window. Once the path is set, you do not have to set it again on Capture relaunch.
- ☐ You have to set MATLAB path again, only if you have installed MATLAB or moved MATLAB to new location in your system.

- Once the MATLAB path is set, it gets saved in the `PSpice.ini` file.

For more information on setting MATLAB path and running MATLAB engine, see the *Setting Up the MATLAB Path* section in the *PSpice MATLAB Interface User Guide*.

## References

This tutorial does not explain the concepts and user interface elements. For information on these, refer to:

- *PSpice MATLAB Interface User Guide*
- *PSpice A/D User Guide*
- *OrCAD Capture User Guide*
- *Allegro Design Entry HDL User Guide*

# **PSpice MATLAB Co-Simulation Tutorial**

## Introduction

---



---

# Using PSpice MATLAB Interface

---

This chapter uses an example to run you through the steps in co-simulation using PSpice MATLAB Interface:

- Overview of Sample Design Files
- Completing the Schematic
- Deciding Interfaces
- Running Simulation and Viewing Results
- Fine Tuning Parameters

## Overview of Sample Design Files

This chapter uses an electric cruise system of a vehicle to run through the steps in using MATLAB interface with PSpice. The cruise system has a controller that feeds into a DC motor. A feedback system controls the speed by restricting it to a desired pre-set value. The speed control is achieved through a PI controller, which drives the system using the integral of weighted sum of error. The weighted sum of error is the difference between the output and the desired set point (in our case the speed).

Voltage control is achieved by duty cycle control method in two sections:

- Generating a pulse whose duty cycle is function of error output
- Implementing power circuit to drive output proportional to this pulse

The design database is provided in the examples directory as a zip (PSpice\_COSim.zip). Unzip this file on your system. You will find the OrCAD Capture project and the model files in the expanded directories.

**Note:** You will need OrCAD Capture, PSpice, MATLAB, and SIMULINK installed to be able to run the steps in this chapter. For more information on system requirements, refer to [Software Requirements](#).


## Completing the Schematic

The model file provided in the example folder contains an incomplete block diagram. You have to add blocks and specify the settings for the PSpice MATLAB interface. The blocks to be added are:

- PSpice: Add one PSpice block.
- Scope: Add two Scope blocks to view the simulation results.
- Mux: Add a Mux block to multiplex signals into the scope block.

**Note:** To learn more about creating block diagrams, refer to the *Creating and Setting Up a Block Diagram Using MATLAB* section in the *Creating a Schematic* chapter of the *PSpice MATLAB Interface User Guide*.

To prepare for simulation, perform the following steps:

1. Launch PSpice A/D.
2. Click *Tools – MATLAB – Co-Simulation*.
3. In the MATLAB tool, click the open file icon () or press *CTRL+O*.

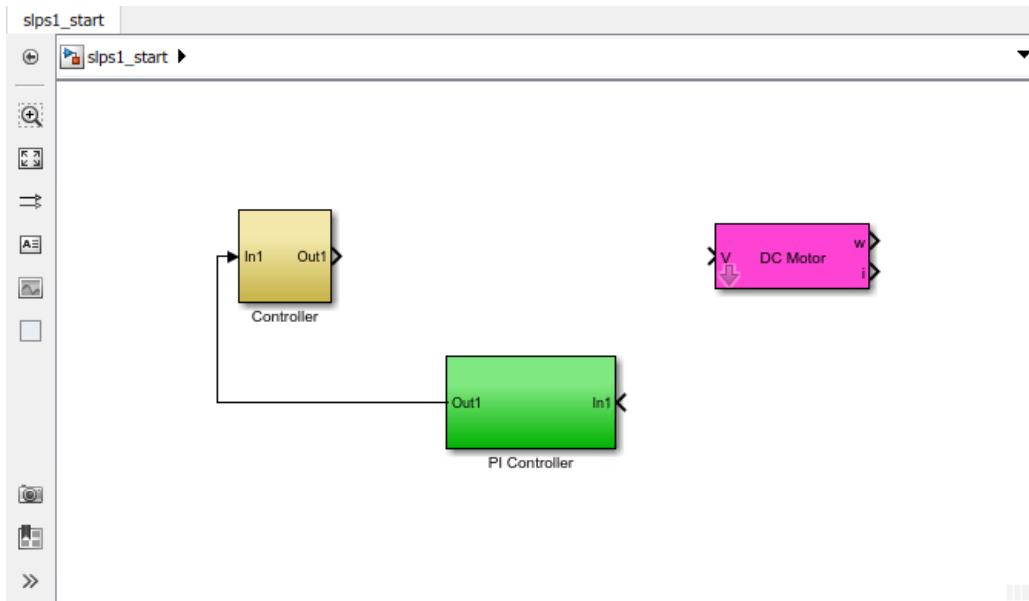
## PSpice MATLAB Co-Simulation Tutorial

### Using PSpice MATLAB Interface


---

4. Browse to the working directory and select `slps1_start.mdl`.

This is the incomplete model file created for you.



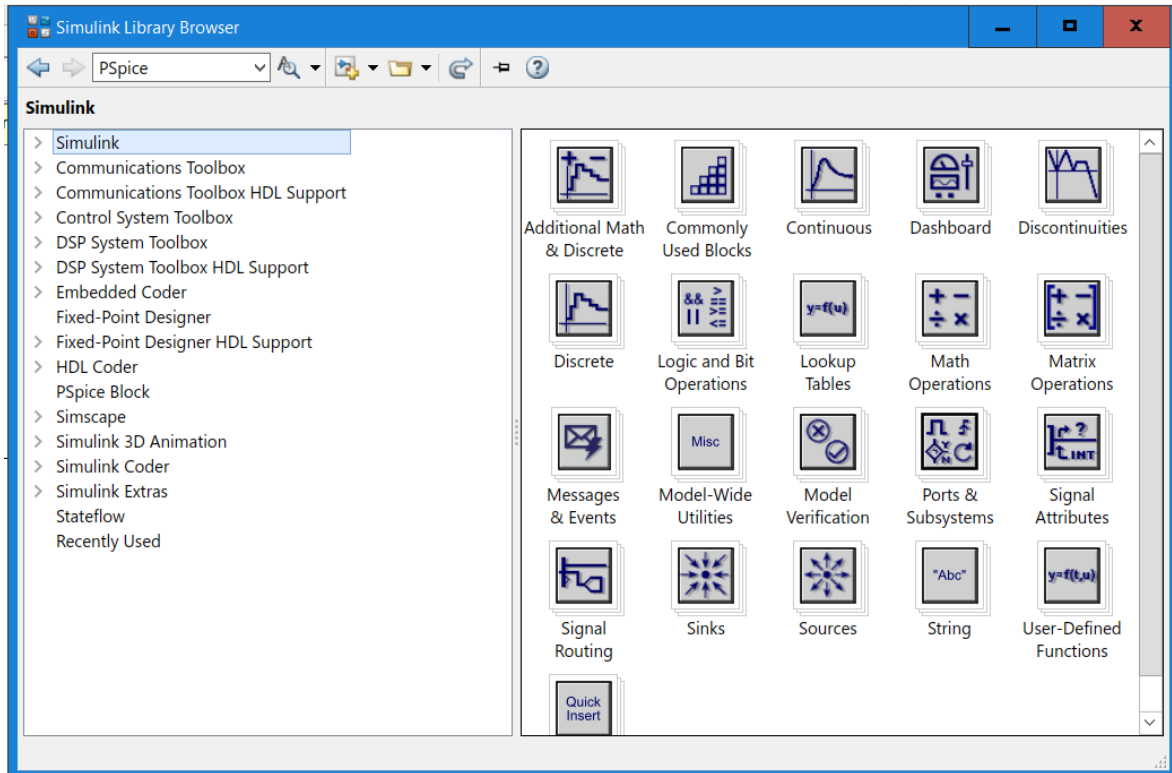
The block design represents the electric cruise system of a vehicle. It models the system using a DC Motor, PI Controller, and a Controller. You will add an PSpice block, two Scope blocks, and a Mux block to the model.

5. In this Simulink project window, click the Library Browser icon (  ) or select *View – Library Browser*.

# PSpice MATLAB Co-Simulation Tutorial

## Using PSpice MATLAB Interface

The *Simulink Library Browser* window opens.

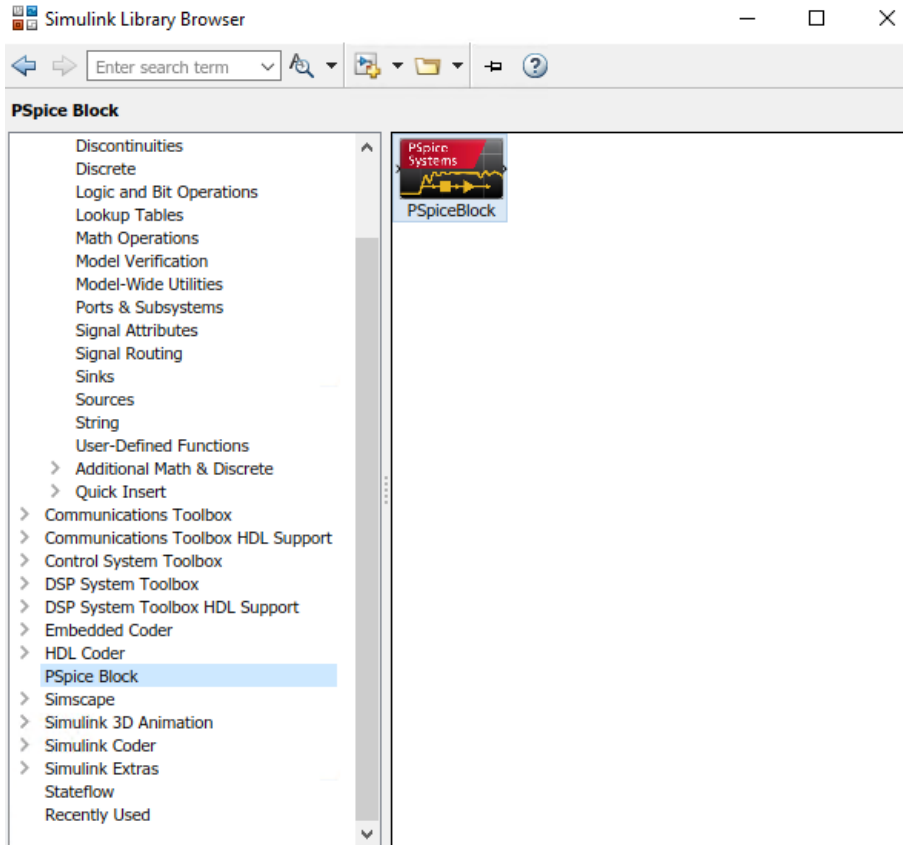


## PSpice MATLAB Co-Simulation Tutorial

### Using PSpice MATLAB Interface

---

6. Select *PSpice Block* from the left pane.

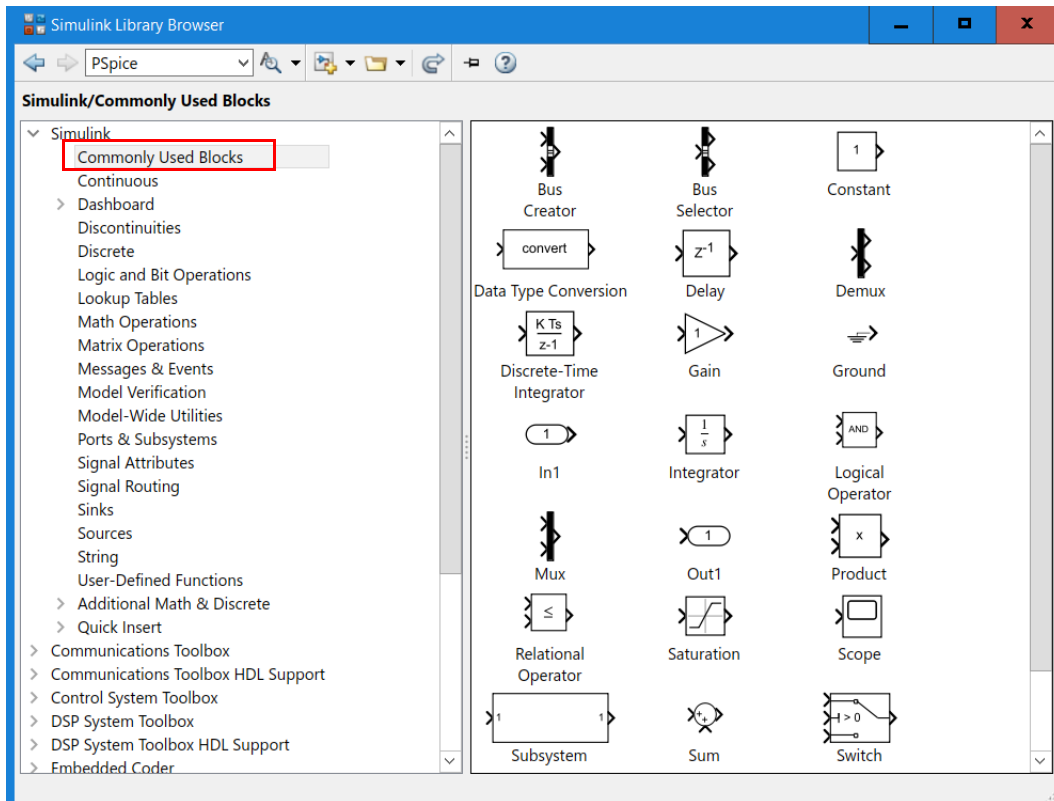


7. Right-click the *PSpice Block* icon in the right pane and select *Add block to model slps1\_start*, or drag this icon into the Simulink project window.

## PSpice MATLAB Co-Simulation Tutorial

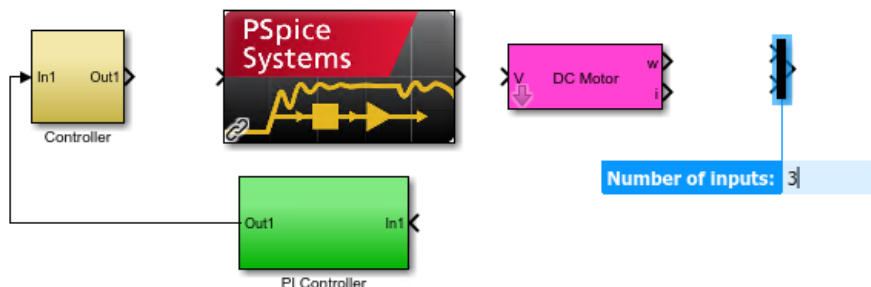
### Using PSpice MATLAB Interface

8. From the libraries list in the left pane, select *Commonly Used Blocks* under *Simulink*.



This library contains the Scope and Mux blocks that you will add.

9. Drag one instance of *Mux* to the `slps1_start` model canvas.
10. Specify the *Number of inputs* value to 3.



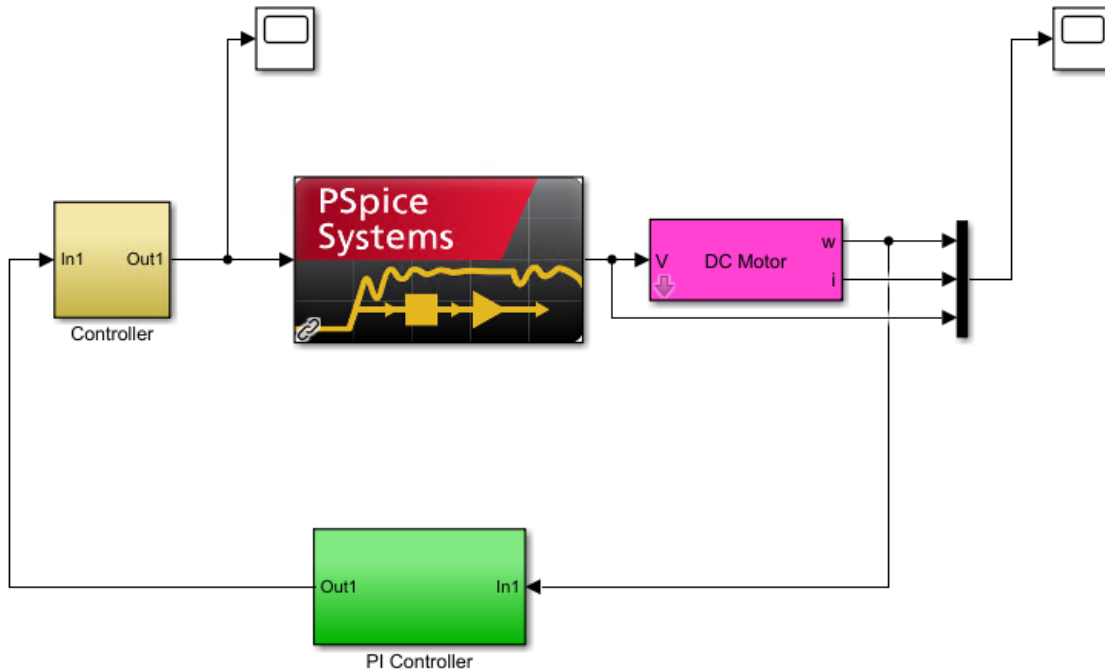
11. Drag two instances of *Scope* to the model canvas.
12. Connect blocks to create a model as shown in Figure 2-1.

## PSpice MATLAB Co-Simulation Tutorial

### Using PSpice MATLAB Interface

To connect the blocks, place the pointer over any input or output port and the pointer changes to a cross. While pressing the left button drag the wire to the port you want to connect to. You can also easily connect two blocks by clicking a block and then, while pressing CTRL, clicking the next block.

**Figure 2-1 Completed and connected blocks**



The first *Scope* block (Scope) provides the simulation result for the Controller and the second *Scope* block (Scope1) provides the simulation results for the entire model.

The *Mux* block combines the three signals and provides a single signal to the *Scope* block.

#### 13. Choose *File – Save* to save the model.

The model file is complete with all required blocks. You will specify the interfaces in the next section.

## Deciding Interfaces

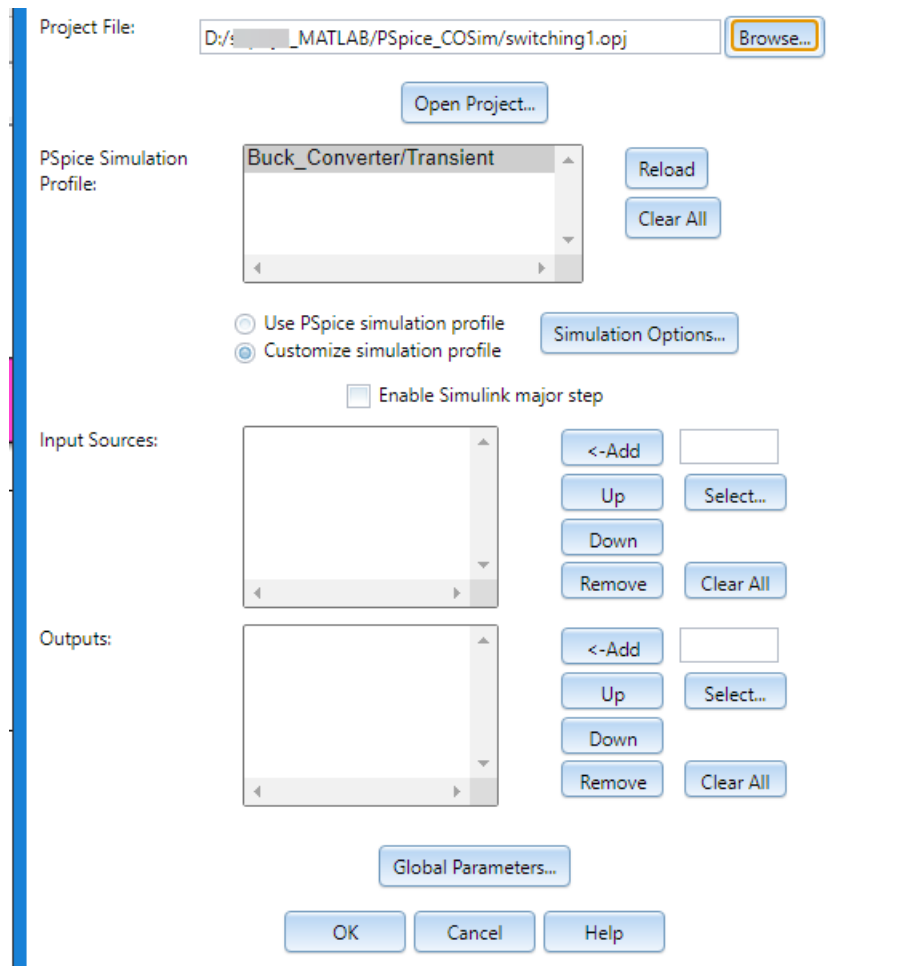
Use the Co-Simulation Settings dialog box to specify the interfaces which include the input sources and the outputs. To do so:

- ➔ In the Simulink project window, double-click the *PSpiceSystems* block.

## PSpice MATLAB Co-Simulation Tutorial

### Using PSpice MATLAB Interface

The *Co-Simulation Settings* dialog box opens.



To learn more about the *Co-Simulation Settings* dialog box, refer to the *Using the Co-Simulation Settings Window* section of the *Creating Simulation Models* chapter in *PSpice MATLAB Interface User Guide*.

To specify the interfaces, perform the following steps:

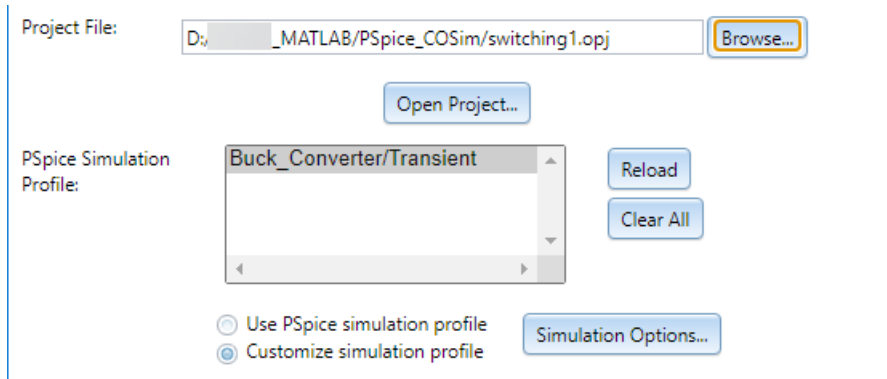
1. Click *Browse* and select *switching1.opj* to specify the OrCAD Capture project that has the design.



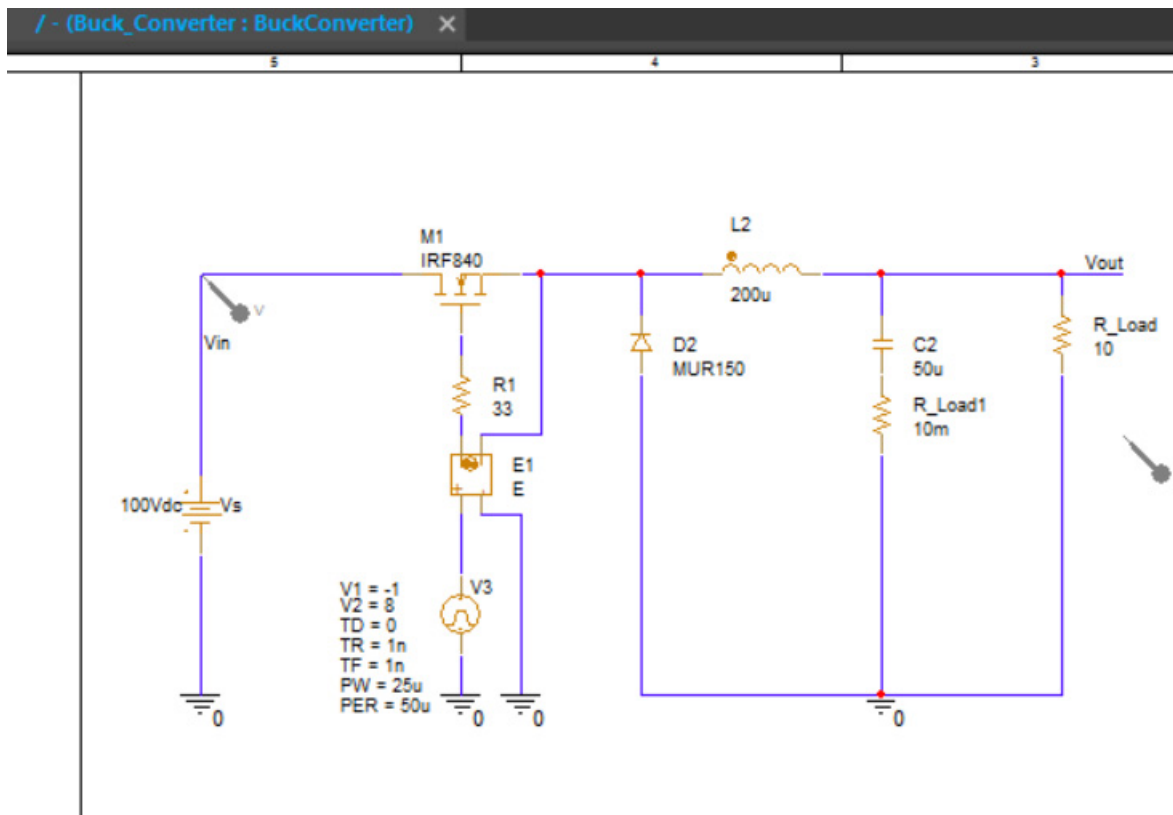
## PSpice MATLAB Co-Simulation Tutorial

### Using PSpice MATLAB Interface

Note that PSpice Simulation Profile is set automatically.



2. Click *Open Project* to open the project in OrCAD Capture to view the interfaces.



Note the  $V_{out}$  signal, and the sources  $V_s$  and  $V3$ .

**Note:** To learn more about using a schematic design editor, refer to the *Creating a Schematic* chapter of *PSpice MATLAB Interface User Guide*.

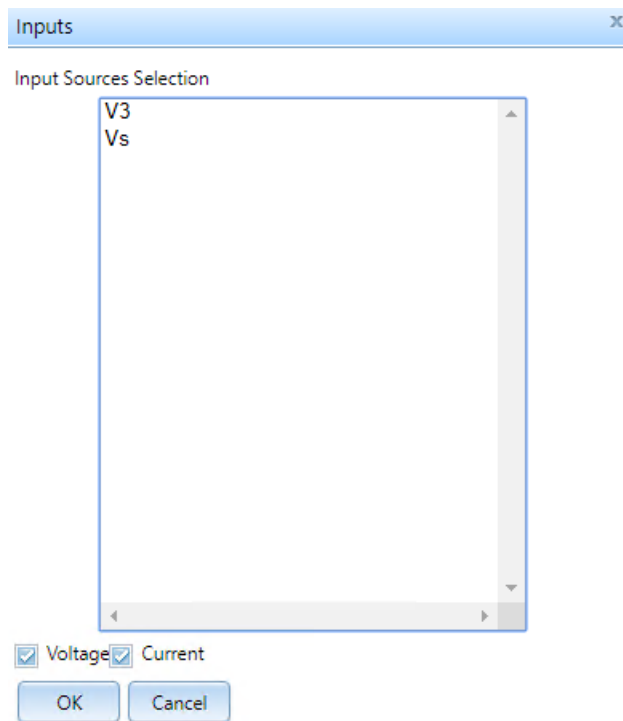
## PSpice MATLAB Co-Simulation Tutorial

### Using PSpice MATLAB Interface

---

3. Choose *PSpice – Create Netlist*.
4. Minimize Capture.
5. In the *Co-Simulation Settings* dialog box, click *Select* in the Input Sources group to specify the input source.

Note that the input sources defined in the Capture design are listed in the *Inputs* dialog box that opens.



6. Select *V3* to specify this as the input source.
7. Click *OK* to close the dialog box.
8. Click *Select* in the Outputs group to specify the output.

The *Outputs* dialog box opens.

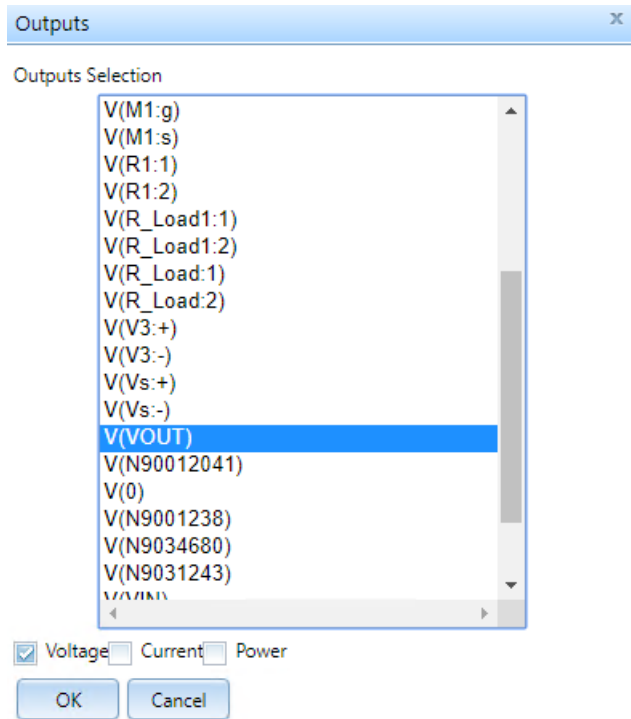
Note that this dialog box lists all the sources from the design. You can uncheck *Currents* and *Power* to display only voltages.

## PSpice MATLAB Co-Simulation Tutorial

### Using PSpice MATLAB Interface

---

9. Select  $V(VOUT)$  to specify this as the output.




10. Click *OK* to close the *Outputs* dialog box.
11. Click *OK* in the *Co-Simulation Settings* dialog box.

The interfaces are specified and the design is ready for simulation.

## Running Simulation and Viewing Results

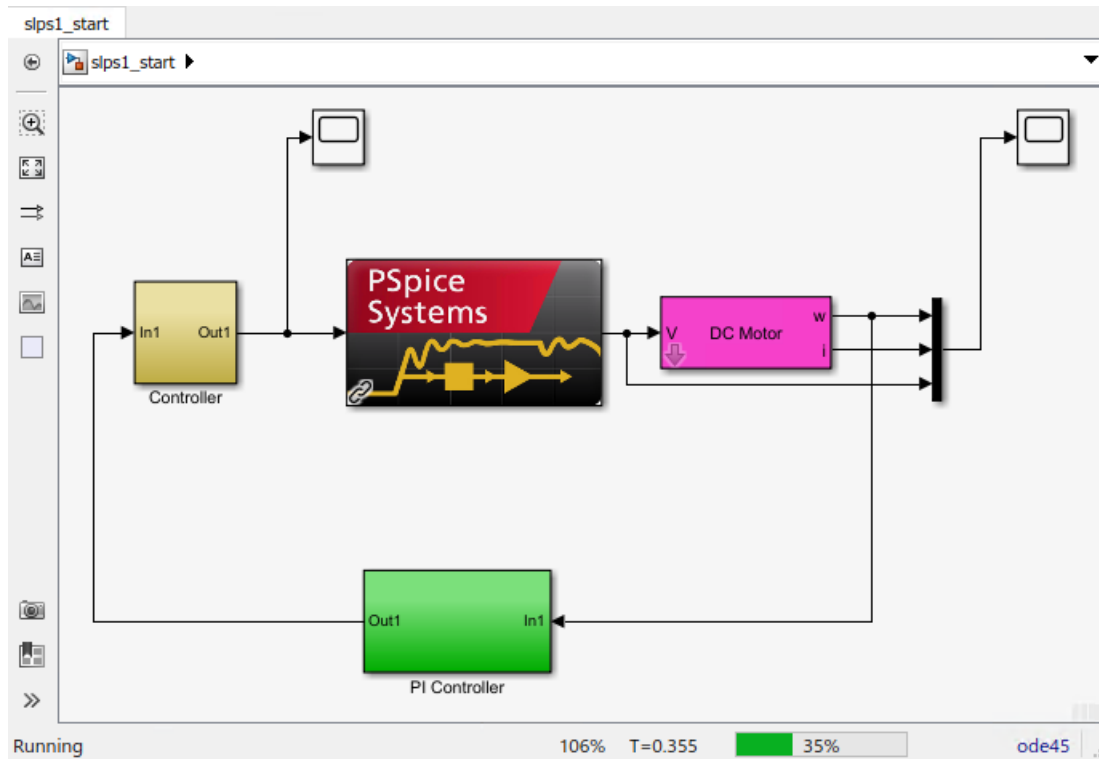
To run the simulation and view results:

1. In the Simulink project window, select *Simulation – Run*, or click the *Run* icon (  ).

## PSpice MATLAB Co-Simulation Tutorial

### Using PSpice MATLAB Interface

Note the status message on the Status Bar. It shows *Running*, the *T* value, and the percentage completion. The status message changes to *Ready* when the simulation is complete.

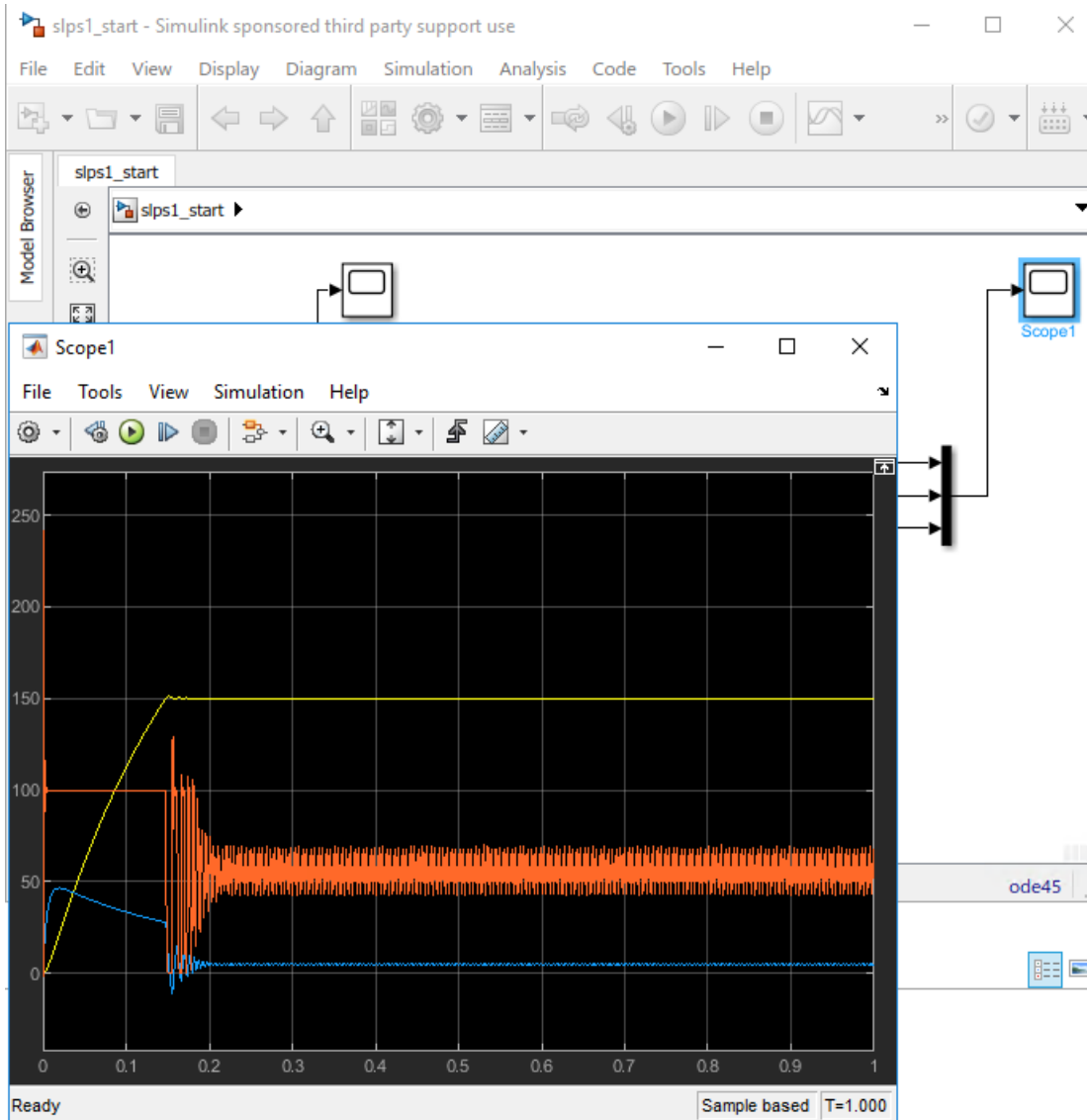


2. Double-click the *Scope1* block.

## PSpice MATLAB Co-Simulation Tutorial

### Using PSpice MATLAB Interface

A window with the simulation results open.



If you see a markedly different result, open the *Co-Simulation Settings* dialog box and click *Reload* to load the PSpice circuit file. Then click *OK* to close the dialog box and run the simulation again.

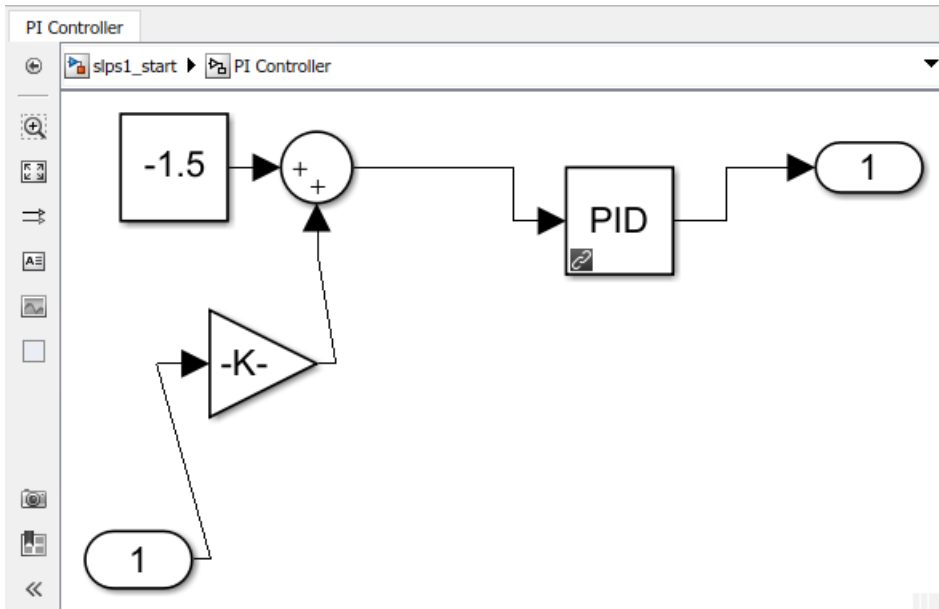
## Fine Tuning Parameters

You can fine tune simulation parameters to improve the simulation results. In the example model file you can, for example, change the controller parameters. To do so:

## PSpice MATLAB Co-Simulation Tutorial

### Using PSpice MATLAB Interface

1. Double-click the *PI Controller* block. This opens the slps1\_start/PI Controller model. The *PI Controller* block is implemented using many other basic blocks as shown in the following block diagram.



2. Double-click the *PID Controller1* block to observe and, if necessary, fine tune the parameters for better results.

The dialog box titled 'Block Parameters: PID Controller1' shows the configuration for the PID Controller. It includes a section for 'PID Controller (mask) (link)' with instructions to enter expressions for proportional, integral, and derivative terms. The 'Parameters' section contains three input fields: 'Proportional' set to '1E3', 'Integral' set to '1E-1', and 'Derivative' set to '0'. The dialog has 'OK', 'Cancel', 'Help', and 'Apply' buttons at the bottom.

You can run the simulation again after making changes.