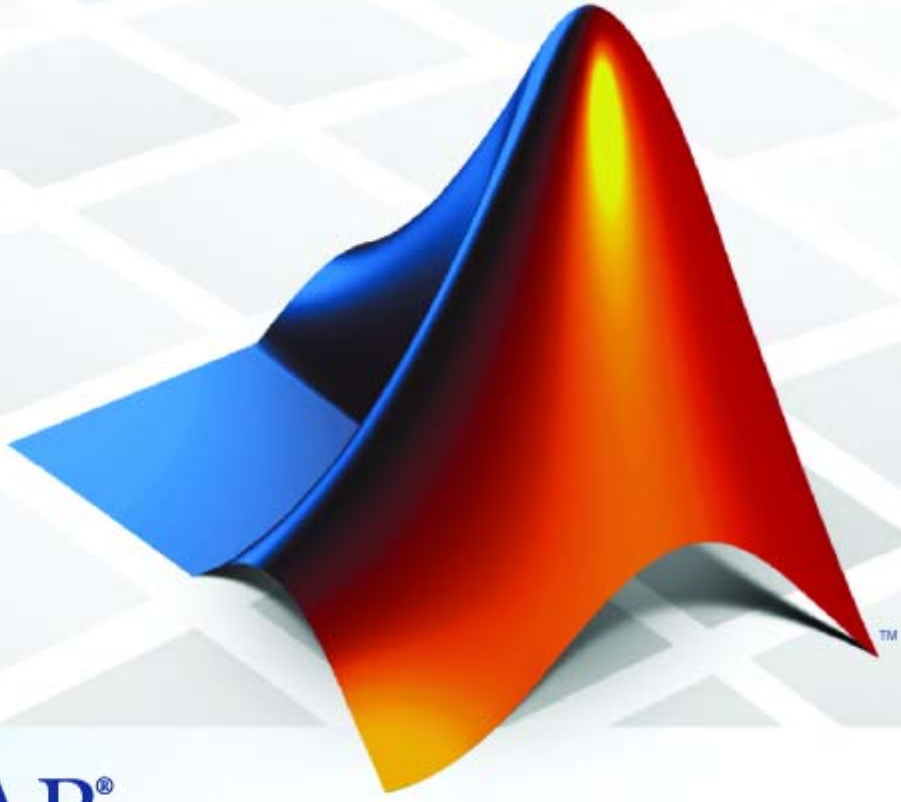
Simulink® Control Design™ 2

Getting Started Guide



MATLAB®
&SIMULINK®

The MathWorks™
Accelerating the pace of engineering and science

How to Contact The MathWorks

`www.mathworks.com`	Web
`comp.soft-sys.matlab`	Newsgroup
`www.mathworks.com/contact_TS.html`	Technical Support

`suggest@mathworks.com`	Product enhancement suggestions
`bugs@mathworks.com`	Bug reports
`doc@mathworks.com`	Documentation error reports
`service@mathworks.com`	Order status, license renewals, passcodes
`info@mathworks.com`	Sales, pricing, and general information

508-647-7000 (Phone)

508-647-7001 (Fax)

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

For contact information about worldwide offices, see the MathWorks Web site.

Simulink® Control Design™ Getting Started Guide

© COPYRIGHT 2004–2008 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See `www.mathworks.com/trademarks` for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

The MathWorks products are protected by one or more U.S. patents. Please see `www.mathworks.com/patents` for more information.

Revision History

June 2004	Online only	New for Version 1.0 (Release 14)
October 2004	Online only	Revised for Version 1.1 (Release 14SP1)
March 2005	Online only	Revised for Version 1.2 (Release 14SP2)
September 2005	Online only	Revised for Version 1.3 (Release 14SP3)
March 2006	First printing	Revised for Version 2.0 (Release 2006a)
September 2006	Online only	Revised for Version 2.0.1 (Release 2006b)
March 2007	Online only	Revised for Version 2.1 (Release 2007a)
September 2007	Online only	Revised for Version 2.2 (Release 2007b)
March 2008	Online only	Revised for Version 2.3 (Release 2008a)
October 2008	Online only	Revised for Version 2.4 (Release 2008b)

Contents

1 Introduction

Product Overview 1-2
Introduction 1-2
Purpose of Linearization 1-2
Role of Linearization in Compensator Design 1-3

Using the GUI Versus Command-Line Functions 1-4

Using the Documentation 1-5
Expected Background 1-5
How to Use This Guide 1-5
Online Documentation 1-6
Using Examples and Demos 1-6
Related Products 1-6

Linearization Quick Start 1-7

Compensator Design Quick Start 1-9

2 Tutorial — Computing a Steady-State Operating Point for a Simulink Model Using the GUI

About This Tutorial 2-2
Objectives 2-2
About the Model 2-2

Computing a Steady-State Operating Point 2-7
Why Compute a Steady-State Operating Point? 2-7
How to Compute a Steady-State Operating Point 2-7

Simulating the Model at the Steady-State Operating Point **2-14**
Steps for Simulating the Model **2-14**
Initializing the Simulink Model with the Steady-State Operating Point **2-14**
Simulating the Initialized Model **2-16**

3 Tutorial — Computing a Steady-State Operating Point for a Simulink Model Using the Command Line

About This Tutorial **3-2**
Objectives **3-2**
About the Model **3-2**

Computing a Steady-State Operating Point **3-7**
Why Compute a Steady-State Operating Point? **3-7**
How to Compute a Steady-State Operating Point **3-7**

Simulating the Model at the Steady-State Operating Point **3-11**
Steps for Simulating the Model **3-11**
Initializing the Simulink Model with the Steady-State Operating Point **3-11**
Simulating the Initialized Model **3-13**

4 Tutorial — Linearizing a Plant in a Single-Loop Control System Using the GUI

About This Tutorial **4-2**
Objectives **4-2**
About the Model **4-2**

Linearizing the Magnetic Ball Plant **4-6**

Why Linearize a Nonlinear Plant? 4-6
Overview of the Linearization Process 4-6
How to Linearize the Magnetic Ball Plant 4-7

Tutorial — Linearizing a Plant in a Single-Loop Control System Using the Command Line

5

About This Tutorial 5-2
Objectives 5-2
About the Model 5-2

Linearizing the Magnetic Ball Plant 5-6
Why Linearize a Nonlinear Plant? 5-6
Overview of the Linearization Process 5-6
How to Linearize the Magnetic Ball Plant 5-7

Tutorial — Designing a PID Compensator for a Single-Loop Feedback System

6

About This Tutorial 6-2
Objectives 6-2
About the Model 6-2
Requirements for the Compensator Design 6-6
Overview of the Compensator Design Process 6-6

Designing a PID Compensator Using the Ziegler-Nichols Tuning Algorithm 6-8

Tuning the PID Compensator Using Bode Graphical Tuning 6-17

Simulating the Closed-Loop Simulink Model 6-22

7 Beginning a Project

Creating or Opening a Simulink Model 7-2
Details of Possible Models 7-2
Example Model: The Magnetic Ball System 7-2

Beginning a New Linearization Task 7-7

8 Specifying Operating Points

What Are Operating Points? 8-2
Definition of an Operating Point 8-2
Equilibrium Operating Points 8-2
Simulink Model Operating Points 8-3

Why Are Operating Points Important? 8-6
Impact of Operating Points 8-6
Choosing an Operating Point for Accurate Linearization . . 8-7

Creating Operating Points 8-9
Simulink® Control Design Default Operating Point 8-9
Creating Operating Points from Specifications 8-9
Creating Operating Points from Known Values 8-15
Creating Operating Points from Simulation 8-18

Exporting and Saving Operating Points 8-21
Uses for Exported and Saved Operating Points 8-21
Exporting Operating Points 8-21
Saving Operating Points 8-22

9 Linearizing Models

What Is Linearization? 9-2
Linearization Background 9-2
Linearization of Simulink Models 9-3
Linearization Process Overview 9-3

Configuring Inputs and Outputs for the Linearized Model 9-5
What are Linearization Points? 9-5
Inserting Linearization Points 9-6
Removing Linearization Points 9-8
Performing Open-Loop Analysis 9-9
Inspecting Analysis I/Os 9-12

Linearizing the Model 9-15
Performing Linearization 9-15
Using the LTI Viewer 9-17

Viewing Linearization Results 9-18
Displaying Linearization Results 9-18
Highlighting Blocks in the Linearization 9-20
Inspecting the Linearization Results Block by Block 9-21
Comparing the Results of Multiple Linearizations 9-23

Validating and Troubleshooting Linearization Results 9-28

10 Designing Compensators

What Is Compensator Design? 10-2

Beginning a Compensator Design Task 10-3

Picking Blocks to Tune 10-5

Selecting Closed-Loop Responses to Design **10-7**

Selecting an Operating Point **10-10**

Creating a SISO Design Task **10-13**
What is a SISO Design Task? **10-13**
Configuring Design Plots **10-14**
Configuring Analysis Plots **10-16**

Completing the Design **10-22**
Tools for Compensator Design **10-22**
Storing and Retrieving Designs **10-26**
Writing the Design to the Simulink Model **10-28**

11 Saving and Exporting Your Work

Saving Projects **11-2**

Opening Previously Saved Projects **11-3**

Exporting Results **11-4**
Exporting Linearization Results **11-4**
Exporting Compensator Designs **11-5**
Exporting Operating Points **11-6**
Exporting and Restoring Linearization I/O Settings **11-6**

A Examples

Getting Started **A-2**

Linearization Example Using the Graphical Interface **A-2**

Index

1

Introduction

- "Product Overview" on page 1-2
- "Using the GUI Versus Command-Line Functions" on page 1-4
- "Using the Documentation" on page 1-5
- "Linearization Quick Start" on page 1-7
- "Compensator Design Quick Start" on page 1-9

Product Overview

In this section...
"Introduction" on page 1-2
"Purpose of Linearization" on page 1-2
"Role of Linearization in Compensator Design" on page 1-3

Introduction

The Simulink® Control Design™ software provides tools for linearization and compensator design for control systems and models. Linearized models often simplify compensator design and system analysis. This is useful in many industries and applications, including

- Aerospace: flight control, guidance, navigation
- Automotive: cruise control, emissions control, transmission
- Equipment manufacturing: motors, disk drives, servos

The Simulink Control Design software works with the Simulink® linearization engine and the Control System Toolbox™ SISO Design Tool. Use it to

- Compute operating points of models using specifications or simulation.
- Extract linear models from models.
- Tune compensator blocks in models with either single or multi-loop configurations.

The Simulink Control Design software provides a graphical user interface (GUI) for performing linearization and compensator design for Simulink models. This chapter introduces a Quick Start guide for using this GUI. The remaining chapters give more details on the linearization and compensator design tasks.

Purpose of Linearization

Many common control system analysis and design methodologies require linear, time-invariant models. However, control systems and physical models

created with Simulink are often nonlinear and time-varying. Linearization is the approximation of a nonlinear system as a linear system, based on the assumption that the system is almost linear within a certain range of operation. With a linearized model you can

- Use the Control System Toolbox LTI Viewer to display and analyze the dynamic behaviors of a model.
- Use the compensator design tools in the Control System Toolbox software, the Robust Control Toolbox™ software, and the Model Predictive Control Toolbox™ software to tune control systems.
- Express a model as a transfer function, state space model, or zero-pole-gain model.
- Determine the response of a model to arbitrary input signals.

A linearized model can provide a good approximation to a nonlinear system when created and used carefully. Factors affecting the accuracy of the approximation addressed in Chapter 7, “Beginning a Project” are

- Choice of operating points. See Chapter 8, “Specifying Operating Points”.
- Understanding the equations for the linearized model. See “What Is Linearization?” on page 9-2.
- Controlling the effect of feedback loops. See “What Is Open-Loop Analysis?” on page 9-9.

Role of Linearization in Compensator Design

Linearized models are especially important for designing compensators. Most compensator design methodologies, such as Bode plots, require a linear plant model. Since most real-world plant models are nonlinear, you must typically linearize the system before you design the compensators for it. As a result, the design of good compensators relies on a good linearization.

The model is automatically linearized for you during compensator design, but it is still important to understand the fundamentals of creating an accurate linear model. Additionally, you should always check that the compensator you designed for the linearized system also works for the nonlinear system. Typically, the compensator works well for the nonlinear system as long as the system does not vary widely from the operating point.

Using the GUI Versus Command-Line Functions

The Simulink Control Design GUI provides a graphical environment for control system linearization and design. With the graphical environment, you can easily inspect and analyze operating points and results of linearization. In addition, you can save and restore settings as well as export results to the MATLAB® workspace.

You can also linearize models using the command-line functions. With the functions, you can automate many of your linearization tasks and perform *batch linearization*, such as linearization of a system at several different values of a parameter. There are no Simulink Control Design functions specifically intended for designing compensators.

Using the Documentation

In this section...
"Expected Background" on page 1-5
"How to Use This Guide" on page 1-5
"Online Documentation" on page 1-6
"Using Examples and Demos" on page 1-6
"Related Products" on page 1-6

Expected Background

Users should be familiar with control systems design and analysis, and have experience creating Simulink models. Familiarity with the Control System Toolbox software is also desirable.

How to Use This Guide

To quickly get started linearizing models, read "Linearization Quick Start" on page 1-7 for a brief overview of the linearization process.

To quickly get started designing compensators, read "Compensator Design Quick Start" on page 1-9 for a brief overview of the compensator design process.

If you are new to linearization, read "What Are Operating Points?" on page 8-2 and "What Is Linearization?" on page 9-2. These sections introduce linearization concepts that are important for accurate creation and use of linearized models.

All users should read Chapter 9, "Linearizing Models" and Chapter 10, "Designing Compensators", which describe features and use of the Simulink Control Design GUI.

To automate the linearization process, or perform batch linearization, continue with "Linearizing Models Using Functions" in the *Simulink Control Design User's Guide*.

Online Documentation

Further documentation is available online or in the Help browser, including the function and block references: “Functions — Alphabetical List” and “Block Reference”.

Using Examples and Demos

The Simulink Control Design documentation uses several examples. You can access these examples by typing

```
demo
```

at the MATLAB prompt and selecting **Simulink Control Design** under the **Simulink** node. You can also access the examples by clicking the **Demos** tab in the Help browser.

Related Products

The MathWorks™ provides several products that are especially relevant to the kinds of tasks you can perform with Simulink Control Design software. For more information about these products, visit the MathWorks Web site at `http://www.mathworks.com/products/simcontrol/`.

Linearization Quick Start

To quickly get started linearizing models, follow these steps. The documentation gives additional information for each step in the process and describes the advanced functionality of the product.

1. Create or open a Simulink model and decide which portion of the model you need to linearize. See "Creating or Opening a Simulink Model" on page 7-2 for more information.

2. Open a linearization task for this model in the Control and Estimation Tools Manager by selecting **Tools > Control Design > Linear Analysis** from the model window. See "Beginning a New Linearization Task" on page 7-7 for more information.

3. Select inputs and outputs for the linearized model by inserting linearization points on the model diagram. To do this, right-click a signal line and select **Linearization Points**, and then select either **Input Point** or **Output Point** depending on whether you want the signal to be an input or output for the linearized model. To remove the effect of feedback loops from the linearized model, you can artificially open a loop by right-clicking the signal line and selecting **Linearization Points > Open Loop**. See "Configuring Inputs and Outputs for the Linearized Model" on page 9-5 for more information.

4. Specify the operating point for the linearized model in the **Operating Points** node of the Control and Estimation Tools Manager. To compute a steady-state operating point from state specifications, click the **Compute Operating Points** tab and enter the specifications into the table, then click **Compute Operating Points**. See Chapter 8, "Specifying Operating Points" for more information and for methods of specifying alternative operating points.

5. Linearize the model by first clicking the **Operating Points** tab within the **Linearization Task** node and then selecting the appropriate operating point from the list. To compute the linearized model, click the **Linearize Model** button. See "Linearizing the Model" on page 9-15 for more information.

6 Analyze the results with the **Linearization Result** pane and the **Linearization Inspector** pane within the **Model** node. Create plots of the results with the **Custom Views** node. See "Viewing Linearization Results" on page 9-18 for more information.

7 Export the linearized model to the MATLAB workspace by right-clicking the **Model** node and selecting **Export**. See Chapter 11, "Saving and Exporting Your Work", for more information.

Compensator Design Quick Start

To quickly get started designing compensators, follow these steps. The documentation gives additional information for each step in the process and describes the advanced functionality of the product.

1. Create or open a Simulink model. See “Creating or Opening a Simulink Model” on page 7-2 for more information.

2. Open a compensator design task for this model by selecting **Tools > Control Design > Compensator Design** from the model window. The compensator design task opens in the Control and Estimation Tools Manager. See “Beginning a Compensator Design Task” on page 10-3 for more information.

3. Within the **Tunable Blocks** pane of the **Simulink Compensator Design Task** node, click the **Add Blocks** button to select the blocks in your model whose parameters you want to tune. These are typically the compensators in your model. See “Picking Blocks to Tune” on page 10-5 for more information.

4. Within the **Closed-Loop Signals** pane, select the closed-loop system(s) whose response(s) you want to design. A closed-loop system is defined by an input point, such as a reference or disturbance signal, and an output point, such as a measured output or actuator signal.

   To define the closed-loop signals, insert linearization input and output points on the model diagram. Right-click a signal line and select **Linearization Points**, and then select either **Input Point** or **Output Point** depending on whether you want the signal to be an input or output for the closed-loop system. See “Selecting Closed-Loop Responses to Design” on page 10-7 for more information.

5. Within the **Operating Point** pane, select the operating point to linearize the model about when designing the compensators. To create a new operating point, click the **New Operating Point** button. See Chapter 8, “Specifying Operating Points” for more information and for methods of specifying operating points.

6. Click the **Tune Blocks** button to open the Design Configuration Wizard which guides you through the selection and configuration of the open- and

closed-loop systems you want to tune, and the design and analysis plots you want to use for this task. See “Creating a SISO Design Task” on page 10-13 for more information.

7 After completing the Design Configuration Wizard, a **SISO Design Task** node is created in the Control and Estimation Tools Manager. The **SISO Design Task** includes several tools for tuning the response of SISO systems:

- A graphical editing environment in the SISO Design Tool window that contains the design plots you configured in the Design Configuration Wizard
- An LTI Viewer window where you can view the time and frequency analysis plots that you configured in the Design Configuration Wizard
- A compensator editor where you can directly edit the block mask parameters or the poles and zeros of compensators in your system
- A tool that automatically generates compensators using PID, internal model control (IMC), or linear-quadratic-Gaussian (LQG) methods (uses the Control System Toolbox software)
- A response optimization tool that automatically tunes the system to satisfy design requirements (available when you have the Simulink® Response Optimization™ software)

See “Completing the Design” on page 10-22 for more information on designing compensators with the **SISO Design Task** node.

8 From the **SISO Design Task** node, click the **Design** node under the **Design History** node. Then click the **Update Simulink Block Parameters** button to write your tuned parameter values back to the Simulink model. See “Writing the Design to the Simulink Model” on page 10-28 for more information.

2

Tutorial — Computing a Steady-State Operating Point for a Simulink Model Using the GUI

- "About This Tutorial" on page 2-2
- "Computing a Steady-State Operating Point" on page 2-7
- "Simulating the Model at the Steady-State Operating Point" on page 2-14

About This Tutorial

In this section...

"Objectives" on page 2-2

"About the Model" on page 2-2

Objectives

In this tutorial, you learn how to use the Simulink Control Design GUI to accomplish the following tasks:

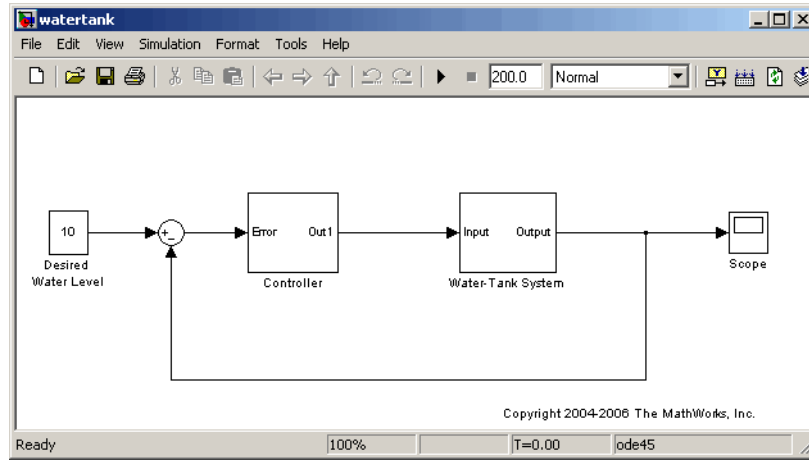
- Compute a steady-state operating point of a Simulink Model.
- Simulate the model at the steady-state operating point.

About the Model

- "The watertank Simulink Model" on page 2-2
- "Water-Tank Subsystem" on page 2-3
- "Controller Subsystem" on page 2-5

The watertank Simulink Model

The `watertank` model shown in the following figure contains the Water-Tank System plant and a simple proportional-integral-derivative controller, called Controller, in a single-loop feedback system.



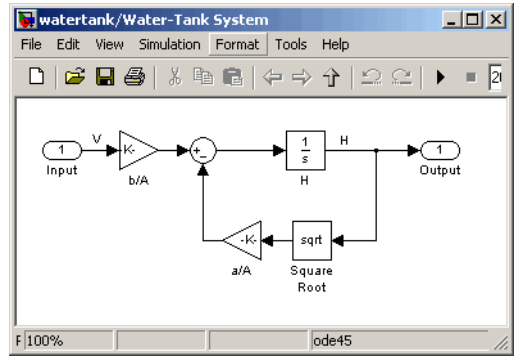
To view the Water-Tank System and the Controller, double-click the corresponding subsystem in the `watertank` model. For descriptions of these subsystems, see the following sections:

- "Water-Tank Subsystem" on page 2-3
- "Controller Subsystem" on page 2-5

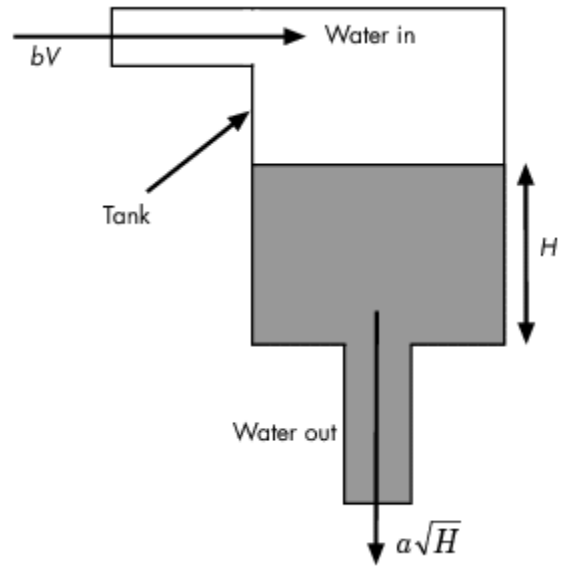
For information about creating Simulink models, see "Creating a Simulink Model".

Water-Tank Subsystem

The Water-Tank subsystem of the `watertank` model appears in the following figure.



This model represents the water-tank system depicted in the following figure.



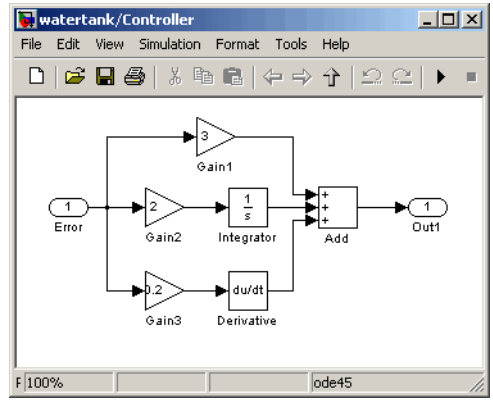
Water enters the tank from the top at a rate proportional to the voltage, V, applied to the pump. The water leaves through an opening in the tank base at a rate that is proportional to the square root of the water height, H, in the tank. The presence of the square root in the water flow rate results in a nonlinear plant.

The following table describes the variables, parameters, differential equations, states, inputs, and outputs of the water-tank system.

Variables	H is the height of water in the tank. Vol is the volume of water in the tank. V is the voltage applied to the pump.
Parameters	A is the cross-sectional area of the tank. b is a constant related to the flow rate into the tank. a is a constant related to the flow rate out of the tank.
Differential equation	$$\frac{d}{dt}Vol = A\frac{dH}{dt} = bV - a\sqrt{H}$$
States	H
Inputs	V
Outputs	H

Controller Subsystem

The Controller subsystem appears in the following figure.



This model represents a simple proportional-integral-derivative controller that controls the height of the water in the Water-Tank System. This model contains one state, *Integrator*.

Computing a Steady-State Operating Point

In this section...

"Why Compute a Steady-State Operating Point?" on page 2-7

"How to Compute a Steady-State Operating Point" on page 2-7

Why Compute a Steady-State Operating Point?

An *operating point* is a set of inputs, outputs, and states that describe the operating conditions of a system. *A steady-state operating point* is an operating point in which all states remain constant over time. Many real-world systems are designed to operate at steady-state operating points. Computing a steady-state operating point is required for:

- Analyzing system dynamics at steady state using simulation
- Linearizing a model at a steady-state operating point
- Designing a compensator for use at a steady-state operating point

For more information about steady-state operating points, see "Equilibrium Operating Points" on page 8-2.

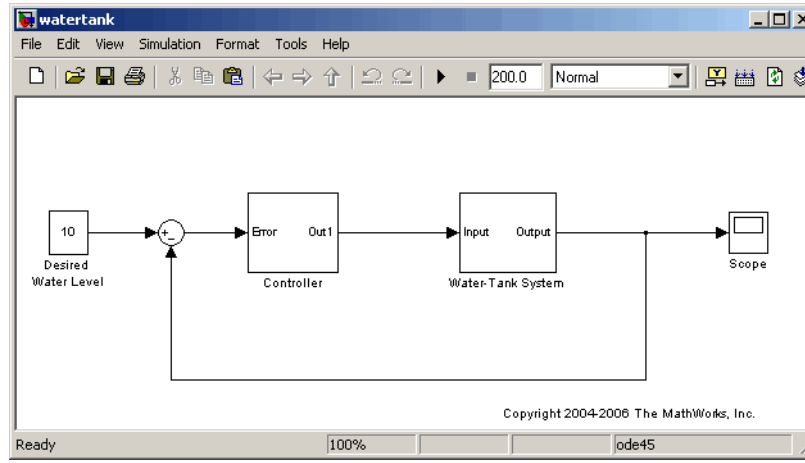
How to Compute a Steady-State Operating Point

To compute a steady-state operating point:

1. Open the `watertank` model by typing the following in the MATLAB command window:

   ```
   watertank
   ```

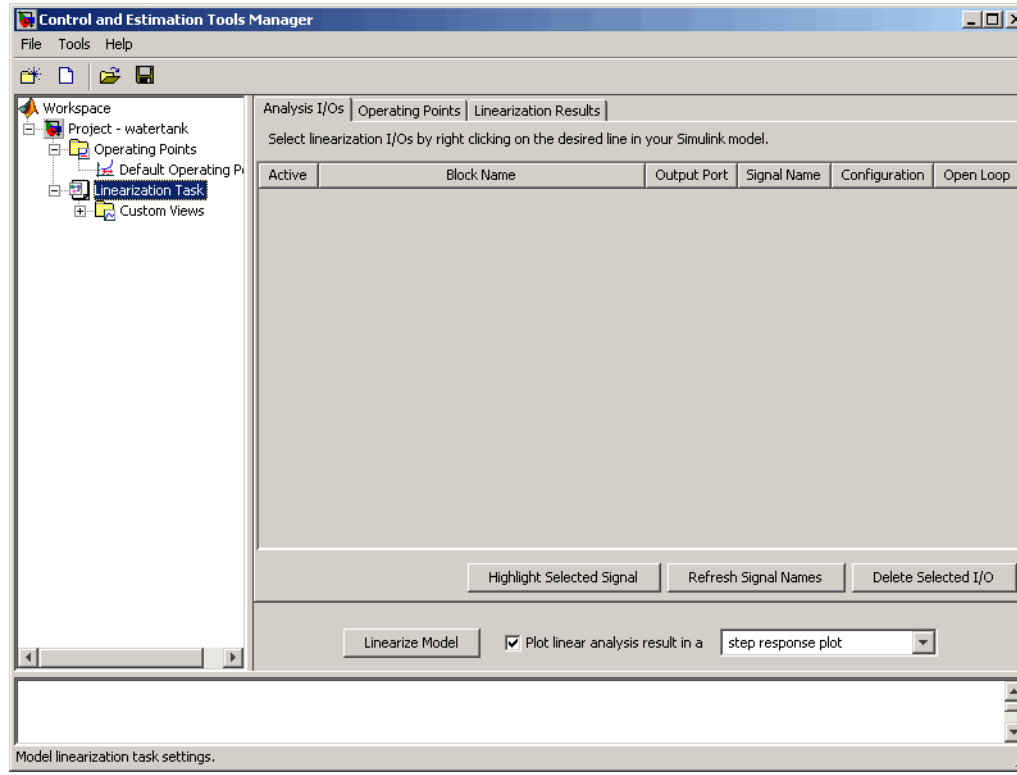
   The model opens in Simulink, as shown in the following figure.



2 In the `watertank` model window, select **Tools > Control Design > Linear Analysis**.

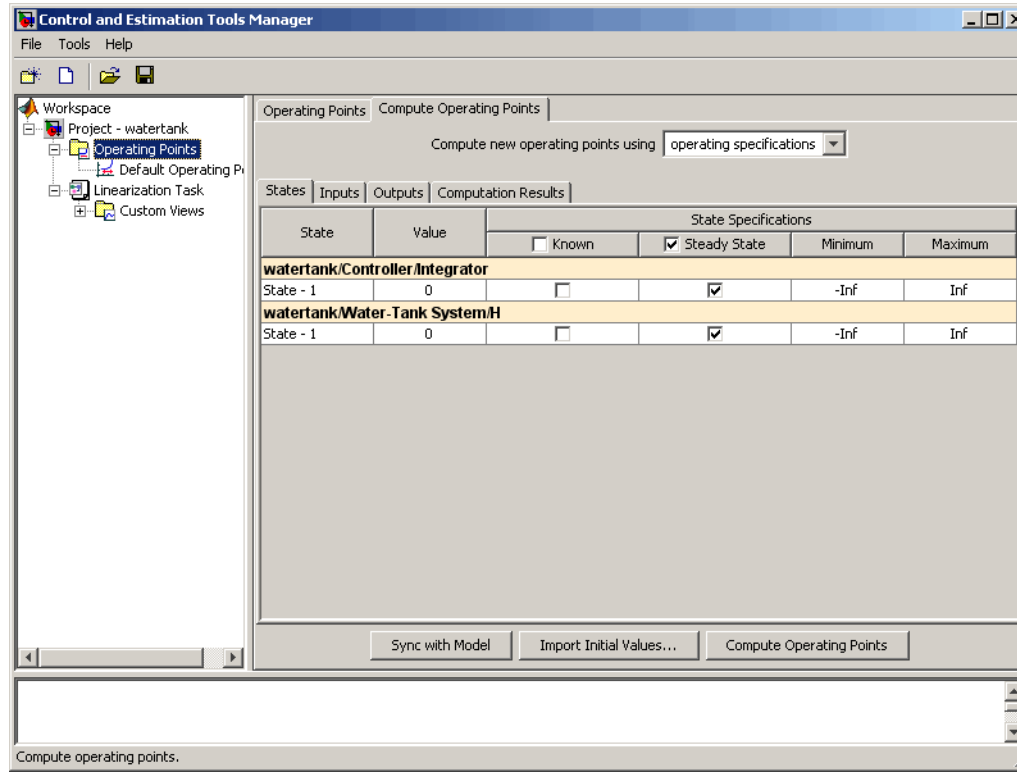
This action opens the Control and Estimation Tools Manager, and creates the following project nodes:

- **Operating Points** (used in this tutorial)
- **Linearization Task** (not used in this tutorial)



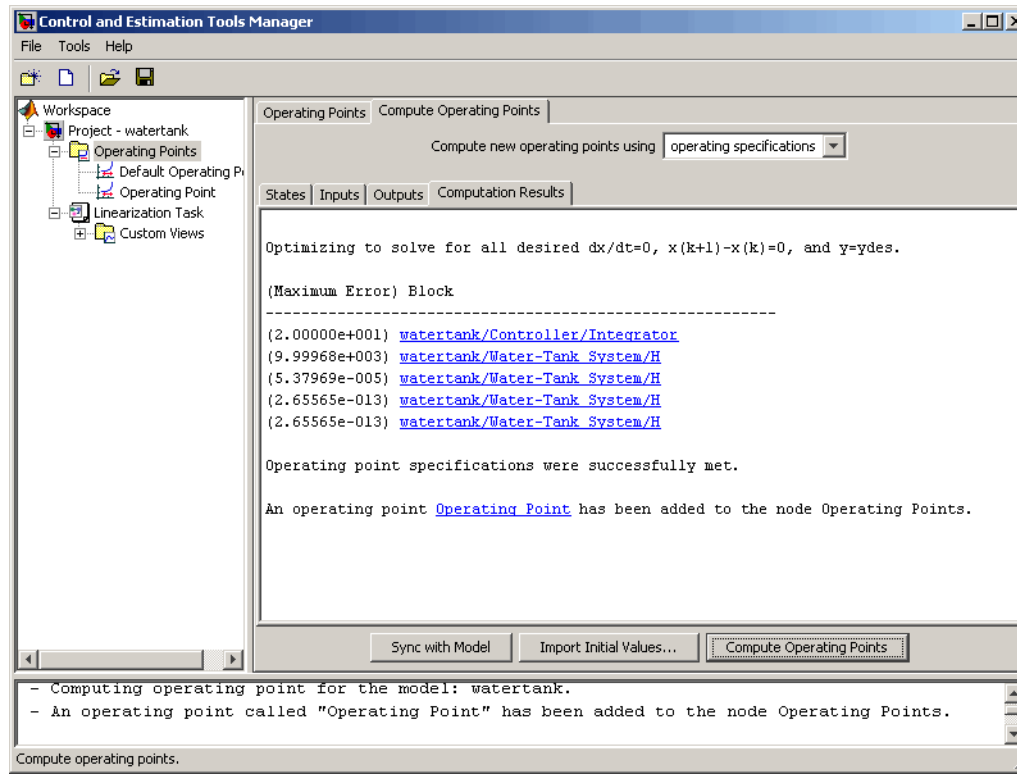
3 Select the **Operating Points** node. Then, select the **Compute Operating Points** tab.

By default, the **Steady State** check boxes are selected for both states in the model, *Integrator* and *H*. This selection indicate that a steady-state operating point will be computed.

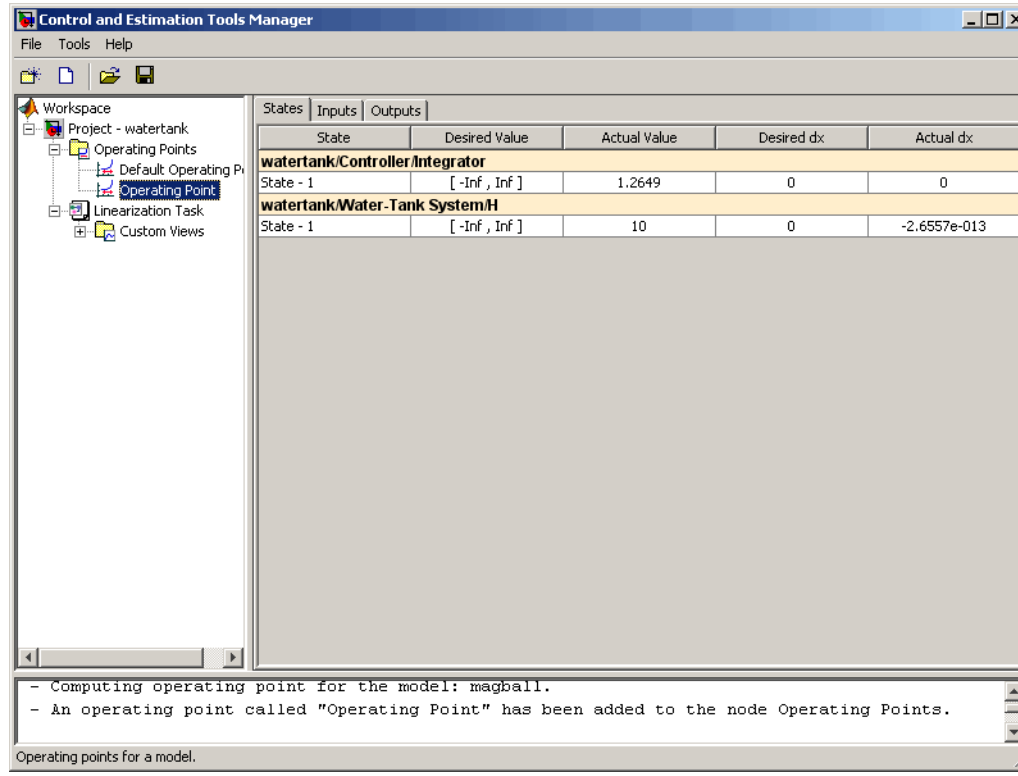


4 Click **Compute Operating Points**.

This action computes the operating point, and adds an **Operating Point** node under the **Operating Points** node. The results of the computation appear in the **Computation Results** tab.



To view the new operating point, select the new **Operating Point** node.



The new **Operating Point** node displays the following information for each state:

- **Actual Value**: Values of the states in the operating point.

  The values are 1.2649 for the state *Integrator* and 10 for the state *H*.

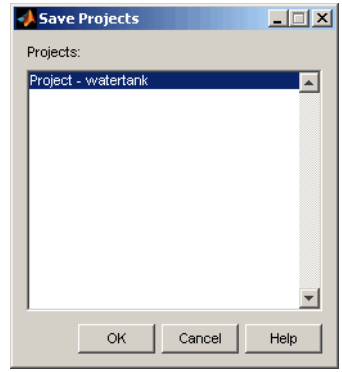
- **Actual dx**: The time derivative of the state. For a steady-state operating point, the time derivatives of all states are very close to or equal to zero.

  The time derivatives are 0 for the state *Integrator* and -2.6557e-013 for the state *H*. These values show that the operating point is at steady state.

5 Save the project, which now includes a steady-state operating point.

a In the Control and Estimation Tools Manager, select **File > Save**.

This action opens the Save Projects dialog box.



b In the Save Projects dialog box, click **OK**.

This action opens the Save Projects window.

c In the Save Projects window, enter a project name, and click **Save**.

The project is saved as a MAT-file.

Tip You can open this project, and use the operating points for future linearization and compensator design. For more information about opening a saved project, see "Opening Previously Saved Projects" on page 11-3.

Simulating the Model at the Steady-State Operating Point

In this section...

"Steps for Simulating the Model" on page 2-14

"Initializing the Simulink Model with the Steady-State Operating Point" on page 2-14

"Simulating the Initialized Model" on page 2-16

Steps for Simulating the Model

In this portion of the tutorial, you simulate the model a steady-state operating point. You must have already computed this operating point, as described in "Computing a Steady-State Operating Point" on page 2-7.

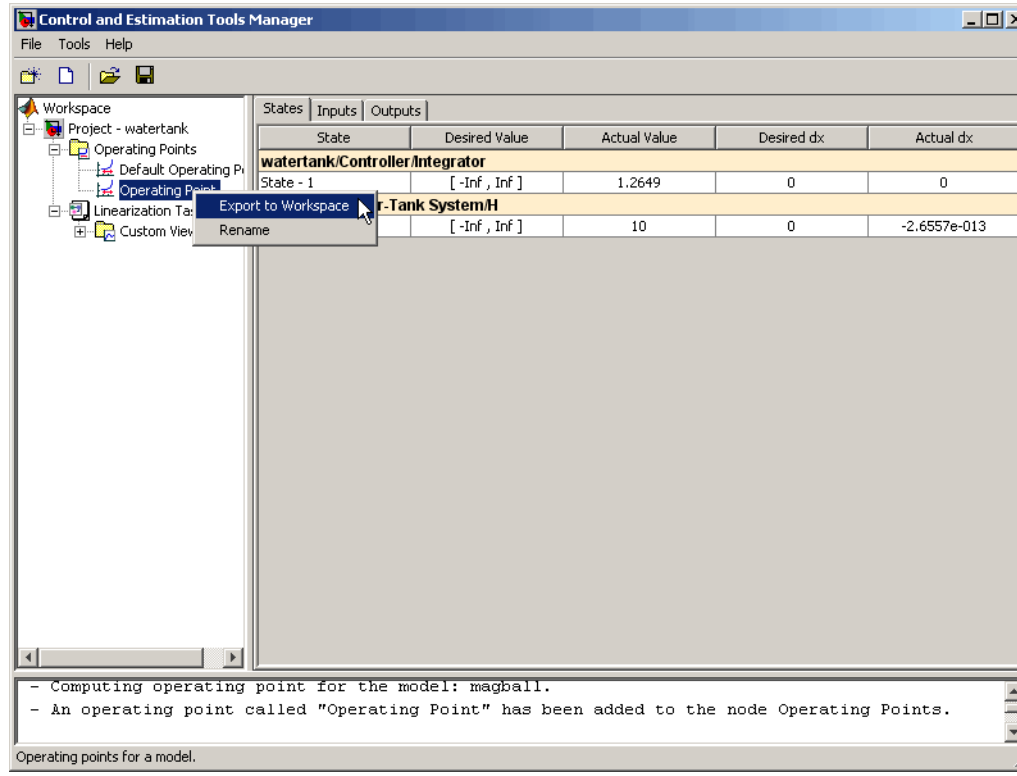
To simulate the model at the steady-state operating point you computed:

1. Initialize the Simulink model with the steady-state operating point. See "Initializing the Simulink Model with the Steady-State Operating Point" on page 2-14.
2. Simulate the initialized model. See "Simulating the Initialized Model" on page 2-16.

Initializing the Simulink Model with the Steady-State Operating Point

To initialize the model with the steady-state operating point:

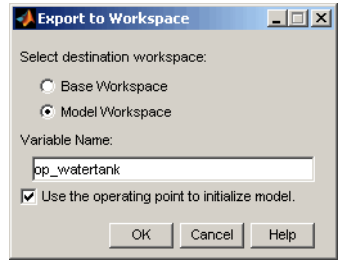
1. Right-click the **Operating Point** node, which contains the steady-state operating point you created, and select **Export to Workspace**.



This action opens the Export to Workspace dialog box.

2 In the Export to Workspace dialog box, make the following selections:

- For **Select destination workspace**, select the **Model Workspace** option.
- Select the **Use the operating point to initialize the model** check box.



3 Click **OK**.

This action loads the operating point into the Simulink model workspace.

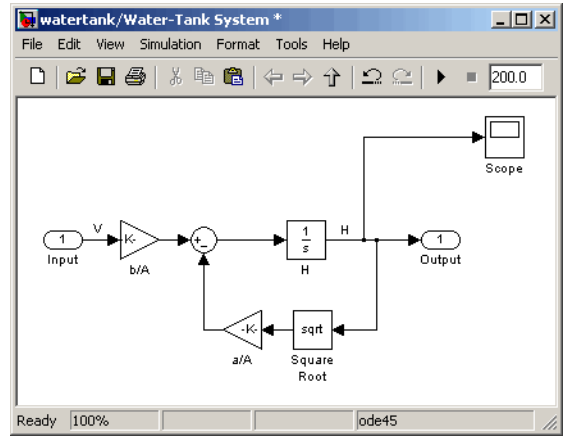
Tip You can view the operating point in the model workspace by selecting **View > Model Explorer** in the Simulink model window, and then selecting the **Model Workspace** node under the **watertank** node.

Simulating the Initialized Model

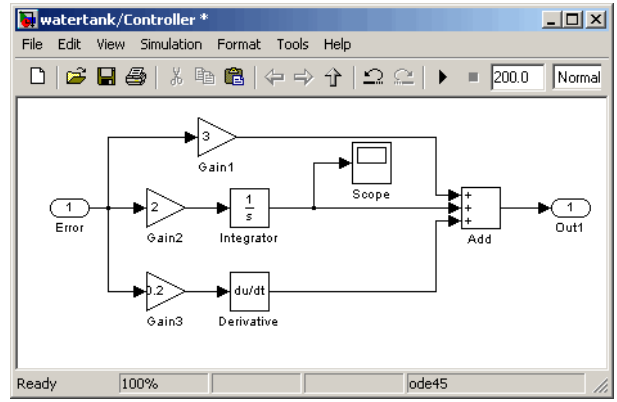
To simulate the initialized model:

1 Add a Scope block to each subsystem.

a In the Water-Tank System model, add a Scope block to the output signal of the integrator. This signal is the state *H*.



b In the Controller model, add a Scope block to the output signal of the integrator. This signal is the state *Integrator*.



2 Open the Scope block window for each model by double-clicking the Scope blocks in the corresponding Simulink model windows.

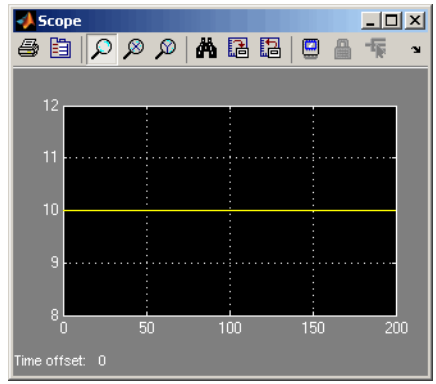
3 Simulate each model by clicking the play arrow in the Simulink model windows.

This action displays the states *H* and *Integrator* in their respective Scope windows. The Scope outputs shows these results:

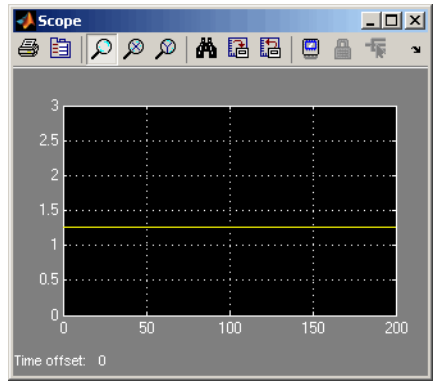
- The state *H* remains constant over time at the expected value of 10.

- The state *Integrator* remains constant over time at the expected value of 1.2649.

Both state values match the values found during the steady-state operating point computation.



Scope Output for State *H*



Scope Output for State *Integrator*

The simulations show that both of the states in the model remain constant over time.

3

Tutorial — Computing a Steady-State Operating Point for a Simulink Model Using the Command Line

- "About This Tutorial" on page 3-2
- "Computing a Steady-State Operating Point" on page 3-7
- "Simulating the Model at the Steady-State Operating Point" on page 3-11

About This Tutorial

In this section...
"Objectives" on page 3-2 "About the Model" on page 3-2

Objectives

In this tutorial, you learn how to use Simulink Control Design functions at the command line to accomplish the following tasks:

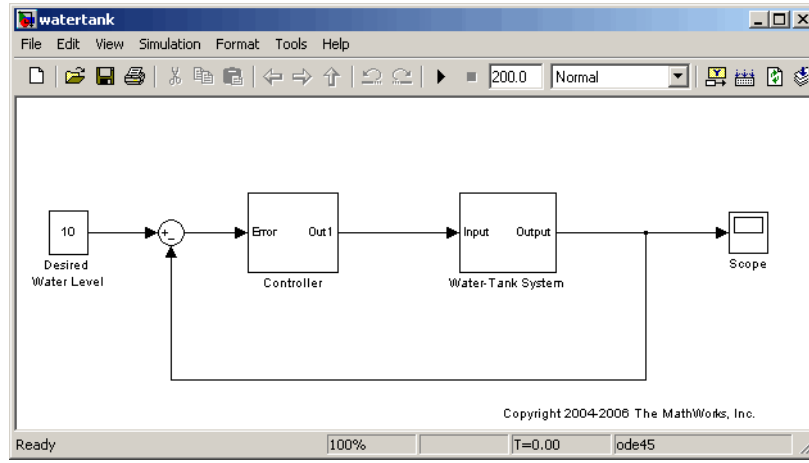
- Compute a steady-state operating point of a Simulink Model.
- Simulating the model at the steady-state operating point.

About the Model

- "The watertank Simulink Model" on page 3-2
- "Water-Tank Subsystem" on page 3-3
- "Controller Subsystem" on page 3-5

The watertank Simulink Model

The `watertank` model shown in the following figure contains the Water-Tank System plant. It also contains a simple proportional-integral-derivative controller, called Controller, in a single-loop feedback system.



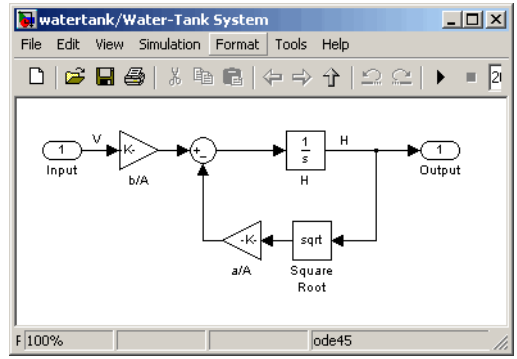
To view the Water-Tank System and the Controller, double-click the corresponding subsystem in the `watertank` model. For descriptions of these subsystems, see the following sections:

- "Water-Tank Subsystem" on page 3-3
- "Controller Subsystem" on page 3-5

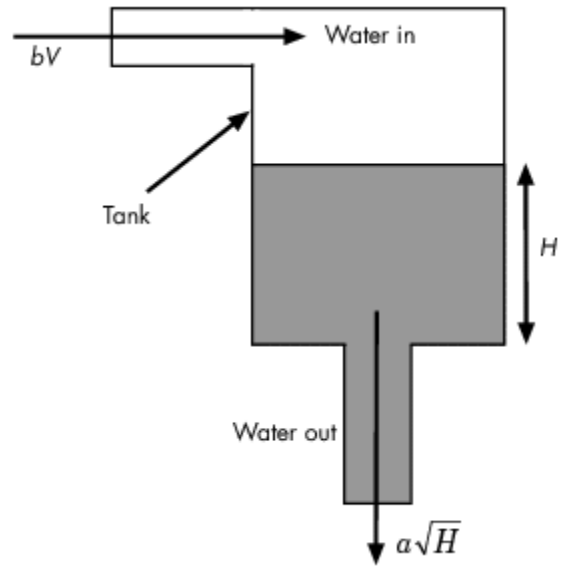
For information about creating Simulink models, see "Creating a Simulink Model".

Water-Tank Subsystem

The Water-Tank subsystem of the `watertank` model appears in the following figure.



This model represents the water-tank system depicted in the following figure.



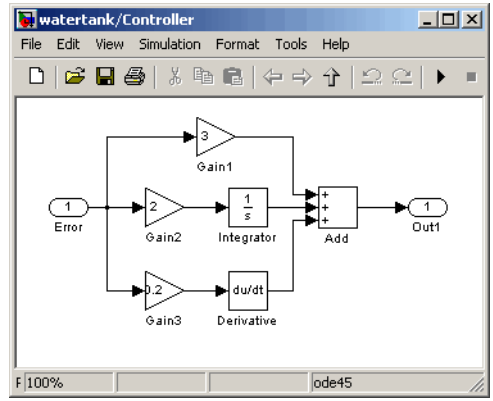
Water enters the tank from the top at a rate proportional to the voltage, V, applied to the pump. The water leaves through an opening in the tank base at a rate that is proportional to the square root of the water height in the tank. The presence of the square root in the water flow rate results in a nonlinear plant.

The following table describes the variables, parameters, differential equations, states, inputs, and outputs of the water-tank system.

Variables	H is the height of water in the tank. Vol is the volume of water in the tank. V is the voltage applied to the pump.
Parameters	A is the cross-sectional area of the tank. b is a constant related to the flow rate into the tank. a is a constant related to the flow rate out of the tank.
Differential equation	$$\frac{d}{dt}Vol = A\frac{dH}{dt} = bV - a\sqrt{H}$$
States	H
Inputs	V
Outputs	H

Controller Subsystem

The Controller subsystem of the `watertank` model appears in the following figure.



This model represents a simple proportional-integral-derivative controller that controls the height of the water in the Water-Tank System. This model contains one state, *Integrator*.

Computing a Steady-State Operating Point

In this section...

"Why Compute a Steady-State Operating Point?" on page 3-7

"How to Compute a Steady-State Operating Point" on page 3-7

Why Compute a Steady-State Operating Point?

An *operating point* is a set of inputs, outputs, and states that describe the operating conditions of a system. *A steady-state operating point* is an operating point in which all states remain constant over time. Many real-world systems are designed to operate at steady-state operating points. Computing a steady-state operating point is required for:

- Analyzing system dynamics at steady state
- Linearizing a model at a steady-state operating point
- Designing a compensator for use at a steady-state operating point

For more information about steady-state operating points, see "Equilibrium Operating Points" on page 8-2.

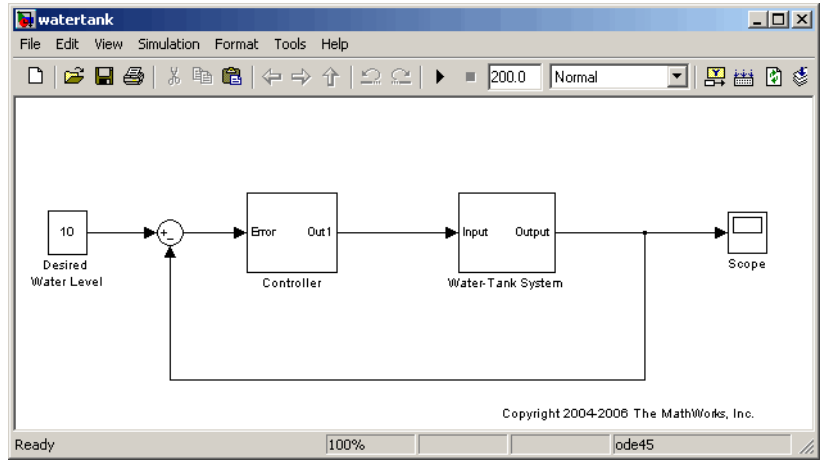
How to Compute a Steady-State Operating Point

To compute a steady-state operating point:

1. Open the `watertank` model by typing the following in the MATLAB command window:

   ```
   watertank
   ```

   The model opens in Simulink, as shown in the following figure.



2 Create an operating point specification object using the `operspec` command by typing the following:

```
watertank_spec = operspec('watertank')
```

This command returns the following result:

```
Operating Specification for the Model watertank.
 (Time-Varying Components Evaluated at time t=0)

States:
----------
(1.) watertank/Controller/Integrator
      spec:  dx = 0,  initial guess:            0
(2.) watertank/Water-Tank System/H
      spec:  dx = 0,  initial guess:            0

Inputs: None
----------

Outputs: None
----------
```

The operating point specification object `watertank_spec` contains objects for all the states, inputs, and outputs in the model. You can view the

operating point specification for a state using the `get` command. For example, to view the state *H*, type the following:

```
get(watertank_spec.States(2))
```

This command returns the following result:

```
      Block: 'watertank/Water-Tank System/H'
              StateName: ''
                      x: 0
                     Nx: 1
                     Ts: [0 0]
             SampleType: 'CSTATE'
      inReferencedModel: 0
                  Known: 0
            SteadyState: 1
                    Min: -Inf
                    Max: Inf
            Description: ''
```

`SteadyState` defaults to a value of 1 for both states, *H* and *Integrator*. This indicates that a steady-state operating point will be computed.

3 Compute the operating point from the operating point specification object `watertank_spec` using the `findop` command by typing the following:

```
[watertank_op,op_report]=findop('watertank',watertank_spec)
```

This command returns the following result:

```
Operating Point for the Model watertank.
 (Time-Varying Components Evaluated at time t=0)

States:
----------
(1.) watertank/Controller/Integrator
      x: 1.26
(2.) watertank/Water-Tank System/H
      x: 10

Inputs: None
----------
```

```
 Operating Report for the Model watertank.
 (Time-Varying Components Evaluated at time t=0)

Operating point specifications were successfully met.

States:
----------
(1.) watertank/Controller/Integrator
      x:          1.26      dx:             0 (0)
(2.) watertank/Water-Tank System/H
      x:            10      dx:    -2.66e-013 (0)

Inputs: None
----------

Outputs: None
----------
```

This operating point and operating point report shows the following information for each state:

- Values of the states, `x`, in the operating point.

  The values are 1.26 for the state *Integrator* and 10 for the state *H*.

- Time derivatives of the states, `dx`, with the desired value in parentheses. For a steady-state operating point, the time derivatives of all states are very close to or equal to zero.

  The time derivatives are 0 for the state *Integrator* and -2.66e-013 for the state *H*. These values show that the operating point is at steady state.

4 Save the operating point for future reuse using the `save` command by typing the following:

```
save watertank_op
```

Tip You can use the `load` command to reload this operating point.

Simulating the Model at the Steady-State Operating Point

In this section...

"Steps for Simulating the Model" on page 3-11

"Initializing the Simulink Model with the Steady-State Operating Point" on page 3-11

"Simulating the Initialized Model" on page 3-13

Steps for Simulating the Model

In this portion of the tutorial, you simulate the model a steady-state operating point.

You must have already computed this operating point, as described in "Computing a Steady-State Operating Point" on page 3-7. If you have not computed the operating point, you can instead load the operating point into the MATLAB workspace by typing `load watertank_ss_operating_point_cl` in the MATLAB command window.

To simulate the model at the steady-state operating point you computed, perform the following steps:

1. Initialize the Simulink model with the steady-state operating point. See "Initializing the Simulink Model with the Steady-State Operating Point" on page 3-11.
2. Simulate the initialized model. See "Simulating the Initialized Model" on page 3-13.

Initializing the Simulink Model with the Steady-State Operating Point

To initialize the model with the steady-state operating point:

1. In the `watertank` Simulink model window, select **Simulation > Configuration Parameters**.

   The Configuration Parameters dialog box opens.

2 Select the **Data Import/Export** node.

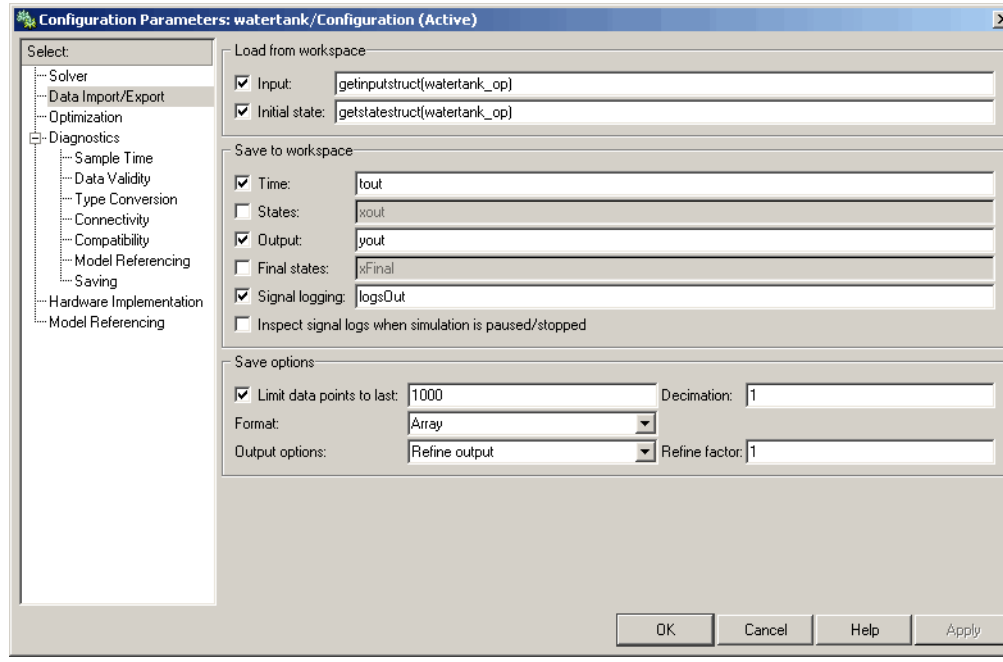
3 In the **Load from workspace** portion of the **Data Import/Export** node, do the following:

- For **Input**, select the check box, and type `getinputstruct(watertank_op)`.

  This action sets the inputs of the model operating point to the input values in `watertank_op`.

- For **Initial State**, select the check box, and type `getstatestruct(watertank_op)`.

  This action sets the initial states of the model operating point to the initial state values in `watertank_op`.



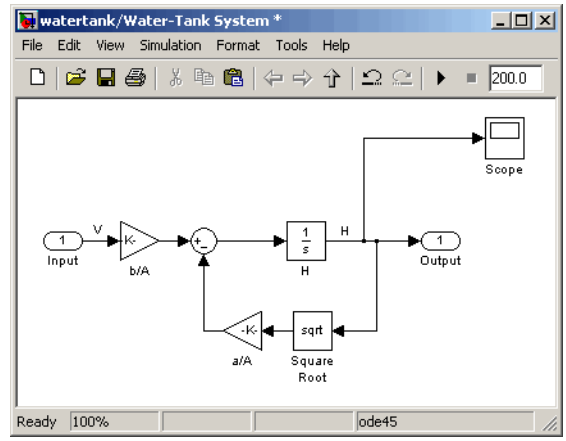
4 Click **OK**.

Simulating the Initialized Model

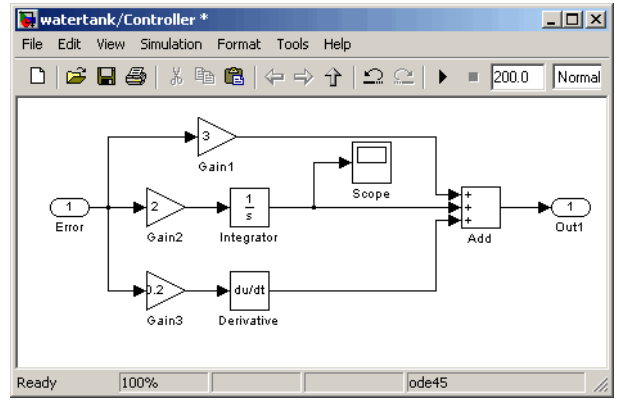
To simulate the initialized model:

1. Add a Scope block to each subsystem.

   a. In the Water-Tank System model, add a Scope block to the output signal of the integrator. This signal is the state *H*.



   b. In the Controller model, add a Scope block to the output signal of the integrator. This signal is the state *Integrator*.



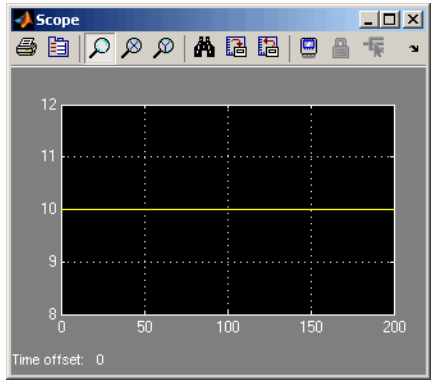
2. Open the Scope block window for each model by double-clicking the Scope blocks in the corresponding Simulink model windows.

3 Simulate each model by clicking the play arrow in the Simulink model windows.

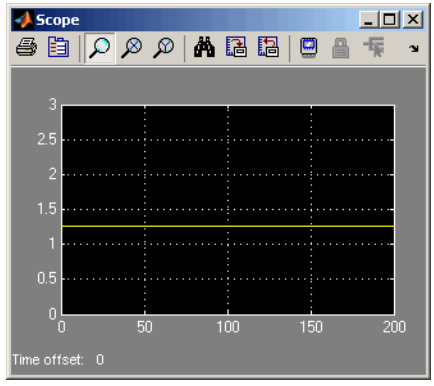
This action displays the states *H* and *Integrator* in their respective Scope windows. The Scope outputs shows these results:

- The state *H* remains constant over time at the expected value of 10.
- The state *Integrator* remains constant over time at the expected value of 1.2649.

Both state values match the values found during the steady-state operating point computation.



Scope Output for State *H*



Scope Output for State *Integrator*

The simulations show that both of the states in the model remain constant over time.

4

Tutorial — Linearizing a Plant in a Single-Loop Control System Using the GUI

- "About This Tutorial" on page 4-2
- "Linearizing the Magnetic Ball Plant" on page 4-6

About This Tutorial

In this section...

"Objectives" on page 4-2

"About the Model" on page 4-2

Objectives

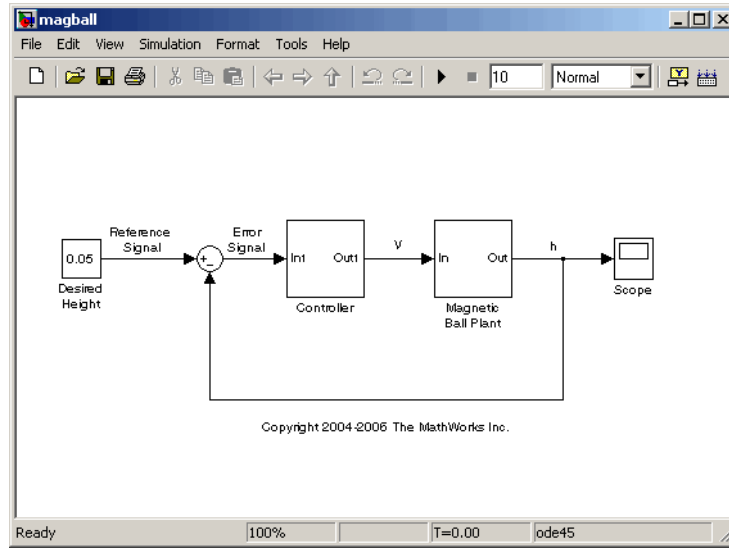
In this tutorial, you learn how to use the Simulink Control Design GUI to linearize a nonlinear plant in a single-loop control system about the operating point in the Simulink model.

About the Model

- "The `magball` Simulink Model" on page 4-2
- "Magnetic Ball Plant Subsystem" on page 4-3

The `magball` Simulink Model

The `magball` Simulink model shown in the following figure contains the nonlinear Magnetic Ball Plant and a controller in a single-loop feedback system.

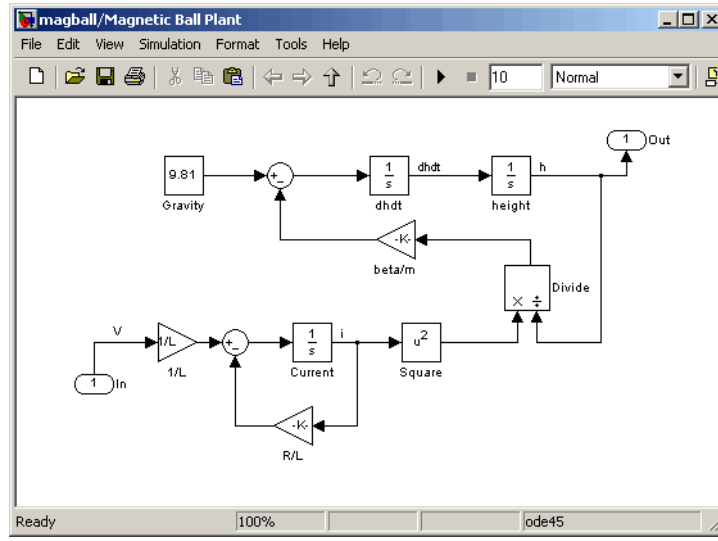


To view the model of the Magnetic Ball Plant subsystem, double-click the corresponding block in the `magball` model. The blocks in this model represent the mathematical system described in “Magnetic Ball Plant Subsystem” on page 4-3.

For information about creating Simulink models, see “Creating a Simulink Model”.

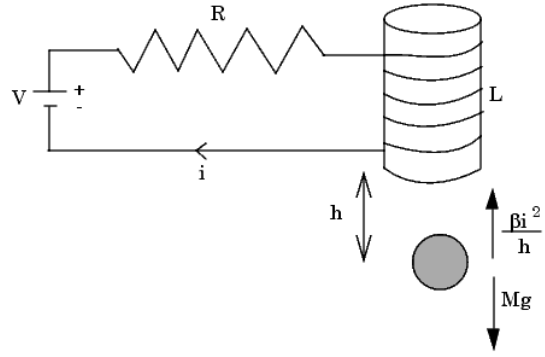
Magnetic Ball Plant Subsystem

The Magnetic Ball Plant subsystem of the `magball` model is shown in the following figure.



The Magnetic Ball Plant model represents an iron ball of mass *M*. This ball moves under the influence of the gravitational force, *Mg*, and an induced magnetic force, $\frac{\beta i^2}{h}$. The presence of the squared term in the induced magnetic force results in a nonlinear plant.

The inductor in the electric circuit, shown in the following figure, causes the induced magnetic force. This circuit also includes a voltage source and a resistor.



The following table describes the variables, parameters, differential equations, states, inputs, and outputs of the Magnetic Ball Plant subsystem.

Variables	h is the height of the ball. i is the current. V is the voltage in the circuit.
Parameters	M is the mass of the ball. g is the gravitational acceleration. β is a constant related to the magnetic force. L is the inductance of the coil. R is the resistance of the circuit.
Differential equations	The height of the ball, h, is described in the following equation: $$M\frac{d^2h}{dt^2} = Mg - \frac{\beta i^2}{h}$$ The current in the circuit, i, is described in the following equation: $$L\frac{di}{dt} = V - iR$$
States	h dh/dt i
Inputs	V
Outputs	h

Linearizing the Magnetic Ball Plant

In this section...
"Why Linearize a Nonlinear Plant?" on page 4-6
"Overview of the Linearization Process" on page 4-6
"How to Linearize the Magnetic Ball Plant" on page 4-7

Why Linearize a Nonlinear Plant?

Linearization is a linear approximation of a nonlinear system, based on the assumption that the system is approximately linear within a specific range of operation. This approximation is valid in a small region around the operating point of the system. An *operating point* is a set of inputs, outputs, and states that describe the operating conditions of a system. For more information about operating points and how they impact linearization, see "Why Are Operating Points Important?" on page 8-6.

In real-world problems, models are nonlinear. Because you need a linear, time-invariant model for most control design and analysis applications, you must linearize a nonlinear model before you can accomplish these goals.

For more information about linearization, see "What Is Linearization?" on page 9-2.

Overview of the Linearization Process

The process for linearizing the Magnetic Ball Plant in this tutorial includes the following tasks:

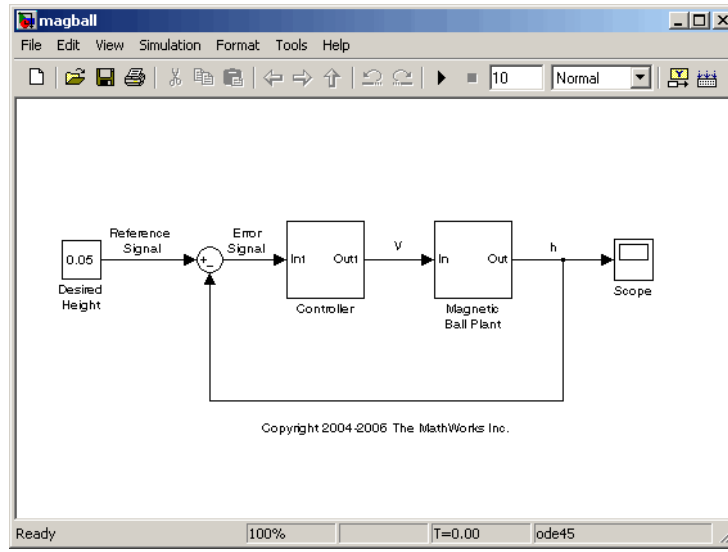
- Defining the portion of the model to linearize, also known as the *linearization path*.
- Removing the effects of a feedback loop in a single-loop control system.
- Linearizing about the existing operating point in the Simulink model.
- Viewing the linearization results in a step response plot and as state-space equations.

How to Linearize the Magnetic Ball Plant

1 Open the `magball` Simulink model by typing the following in the MATLAB Command Window:

```
magball
```

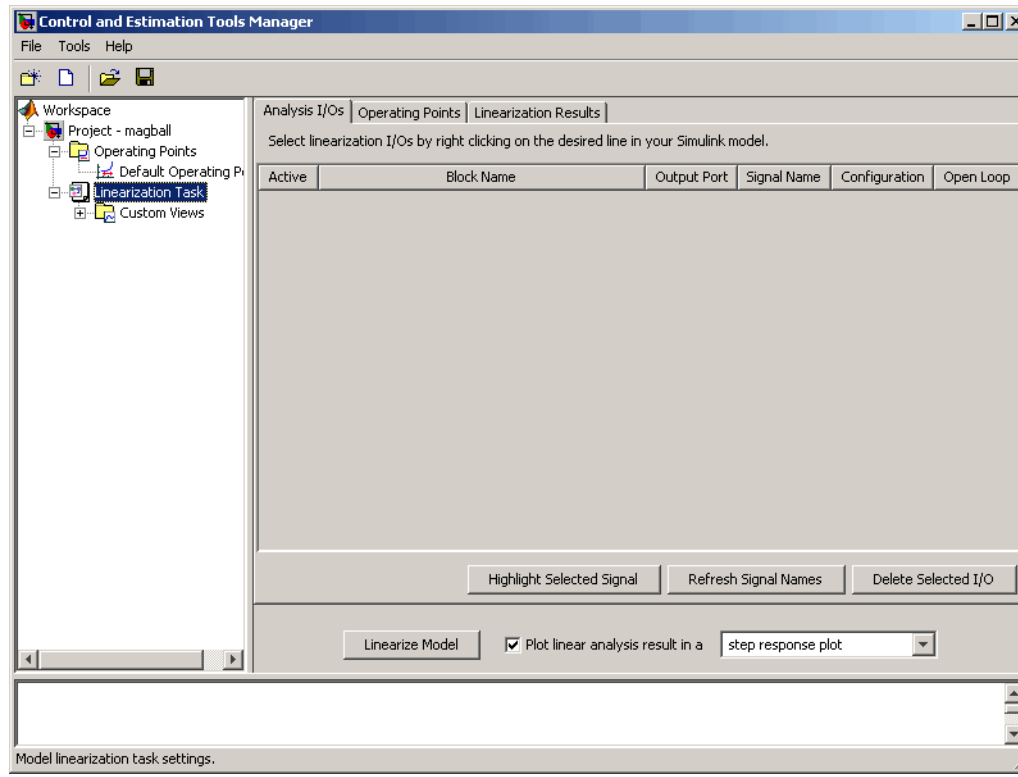
The model opens in Simulink as shown in the following figure.



2 In the `magball` model window, select **Tools > Control Design > Linear Analysis**.

This action opens the Control and Estimation Tools Manager and creates the following project nodes:

- **Operating Points** (not used in this tutorial)
- **Linearization Task** (used in this tutorial)



3 Define the portion of the model to linearize.

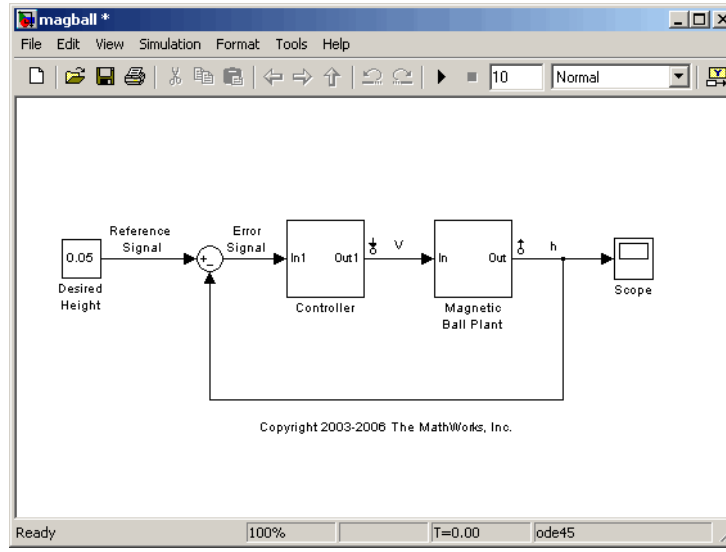
a In the `magball` model, right-click the input signal to the Magnetic Ball Plant subsystem, named `V`, and select **Linearization Points > Input Point**.

This action displays the symbol on the signal line. This symbol indicates the start of the linearization path.

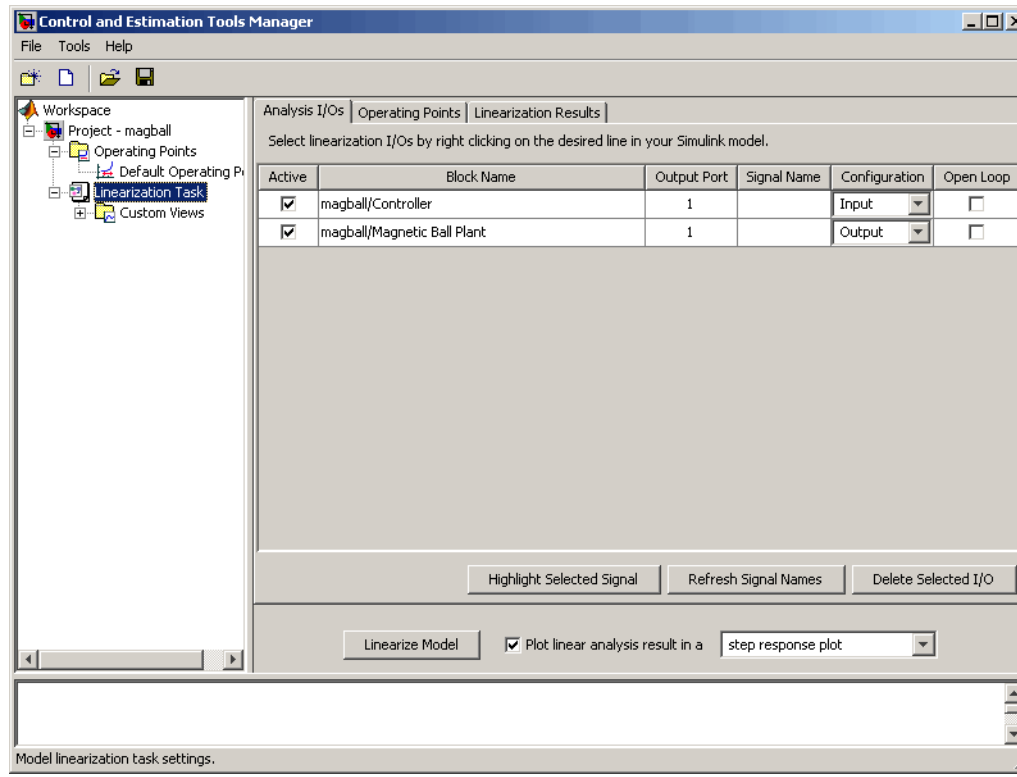
b Right-click the output signal from the Magnetic Ball Plant subsystem, named `h`, and select **Linearization Points > Output Point**.

This action displays the symbol on the signal line. This symbol indicates the end of the linearization path.

The Simulink model now resembles the following figure.

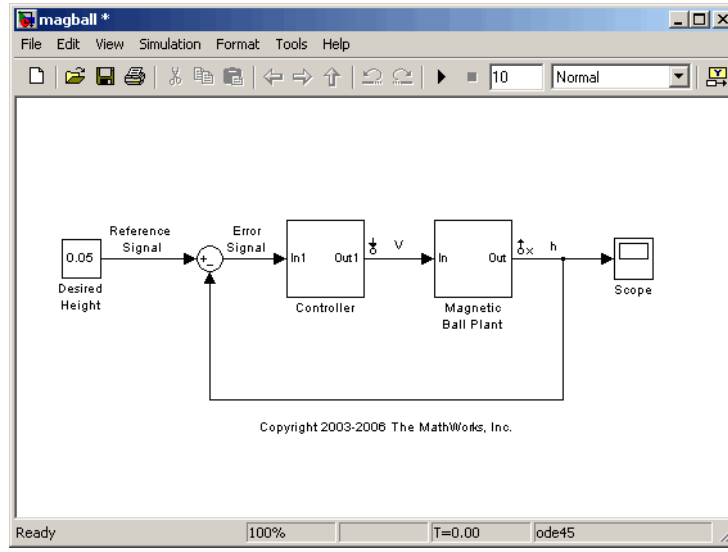


The linearization input and output points in the Simulink model also display in the **Analysis I/Os** tab of the **Linearization Task** node of the Controls and Estimation Tools Manager.

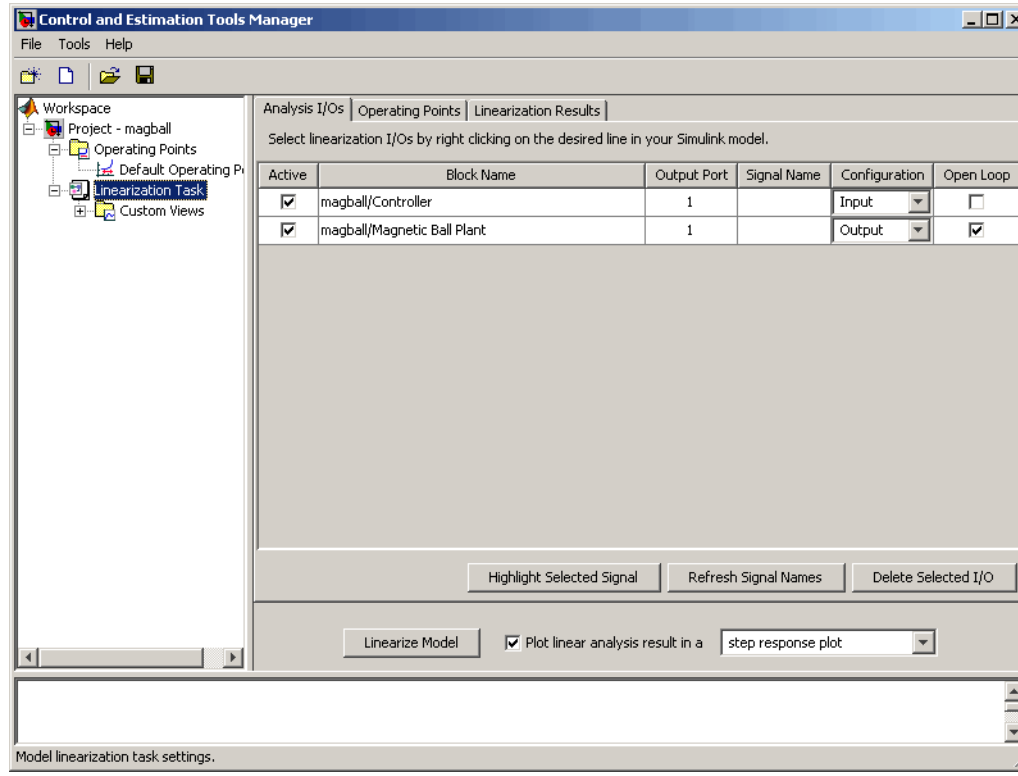


4 Remove the effects of the feedback loop for open-loop analysis by right-clicking the output signal from the Magnetic Ball Plant subsystem, named `h`. Then, select **Linearization Points > Open Loop**.

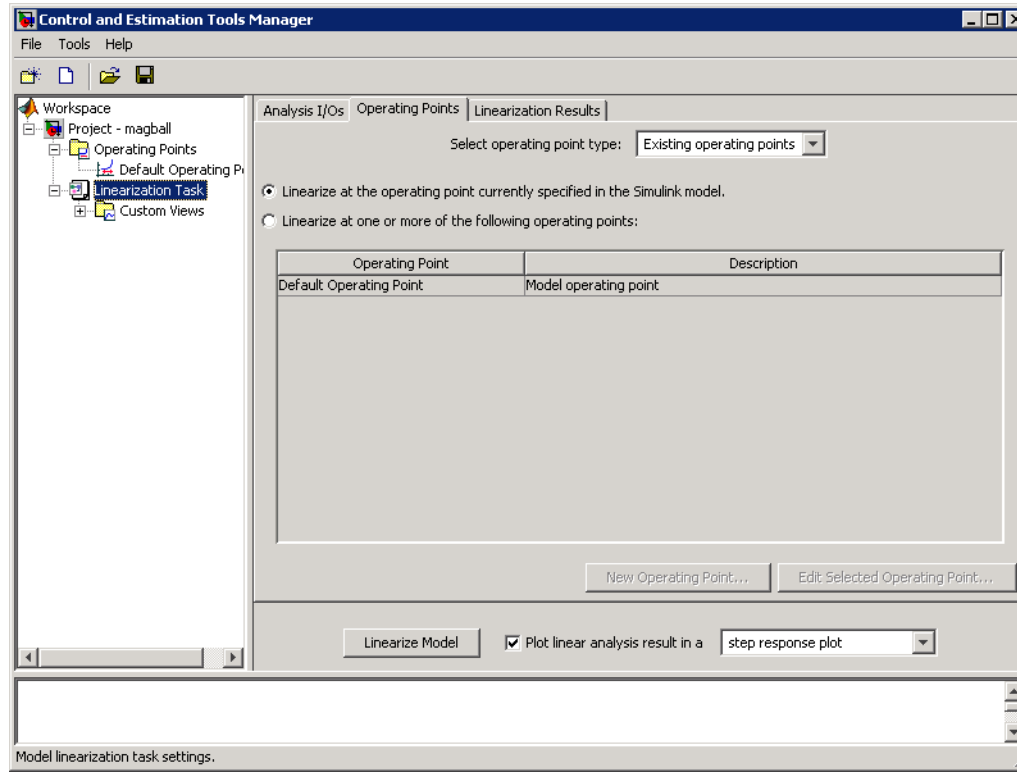
This action displays an `x` on the signal line, which indicates the location of the loop opening. Opening the loop ensures that the linearization result includes only the plant while preserving the model operating point. For more information on the affects of a feedback loop on linearization results, see “Performing Open-Loop Analysis” on page 9-9.



When you open the loop in the Simulink model, the **Open Loop** check box is selected for `magball/Magnetic Ball Plant` in the **Analysis I/Os** tab of the Control and Estimation Tools Manager.



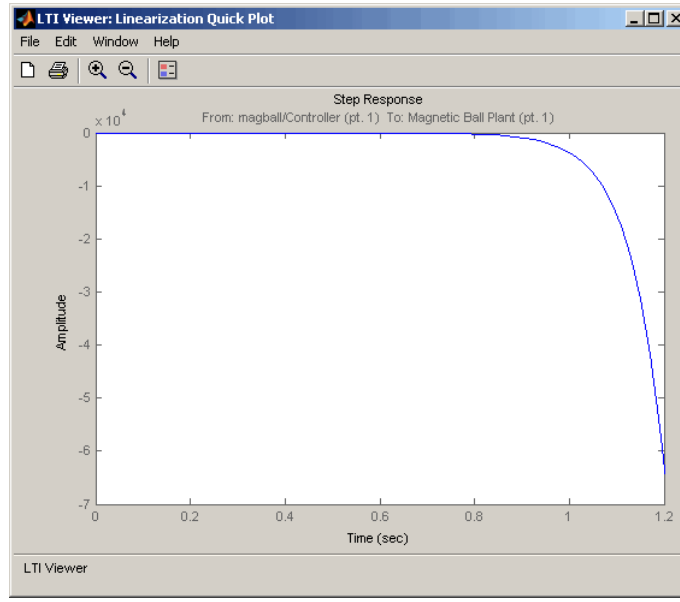
5 In the **Operating Points** tab of the **Linearization Task** node, verify that the **Linearize at the operating point currently specified in the Simulink model** option is selected. By default, this option is selected when you open the **Operating Points** tab.



Tip You can view the operating point currently specified in the Simulink model by selecting the **Default Operating Point** node under the **Operating Points** node.

6 Click **Linearize Model**.

This action computes the linearized model, adds a **Model** node under the **Linearization Task** node, and opens the LTI Viewer GUI, which displays a step response plot of the linear model.



The step response decreases exponentially after about 0.8 seconds, which indicates that the plant model is unstable. The linear model provides an accurate approximation of the nonlinear Magnetic Ball Plant, which is also unstable.

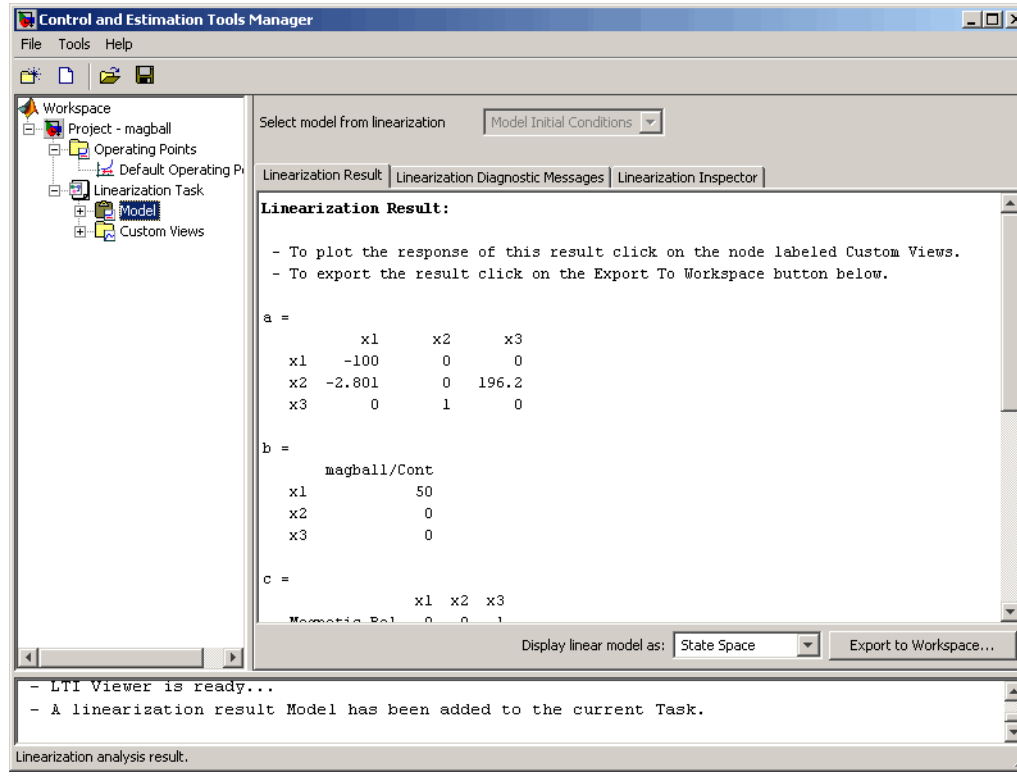
Tip You can use the right-click menu of the LTI Viewer GUI to display a different plot or add characteristics to the response plot, such as peak response and settling time. For more information about working with response plots, see "LTI Viewer" in the *Control System Toolbox Getting Started Guide*.

Tip For information about designing a stabilizing controller for the Magnetic Ball Plant, see Chapter 10, "Designing Compensators".

---

7 View the state-space matrices of the linearization result by selecting the **Model** node under the **Linearization Task** node.

This action opens the **Linearization Results** tab shown in the following figure.

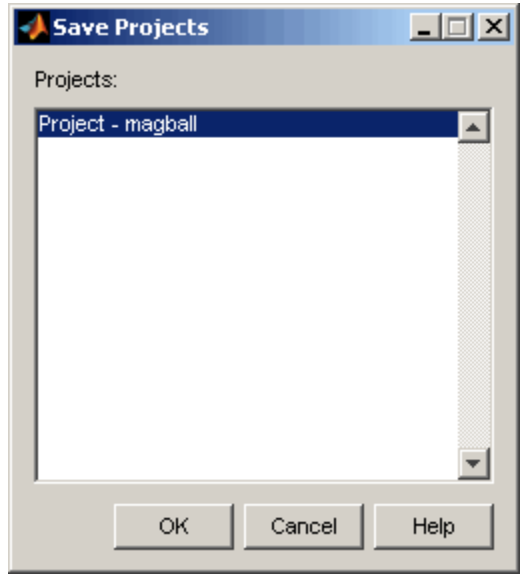


Tip To display linearization results using a different mathematical form, such as zero-pole-gain or transfer function, select the corresponding option from the **Display linear model as** list.

8 Save the Control and Estimation Tools Manager project, which contains the linearization results.

a In the Control and Estimation Tools Manger, select **File > Save**.

This action opens the Save Projects dialog box.



b In the Save Projects dialog box, click **OK**.

This action opens the Save Projects window.

c In the Save Projects window, enter a project name, and click **Save**.

This action saves the project as a MAT-file.

Tip You can load this project by selecting **File > Load** in the Control and Estimation Tools Manager.

5

Tutorial — Linearizing a Plant in a Single-Loop Control System Using the Command Line

- "About This Tutorial" on page 5-2
- "Linearizing the Magnetic Ball Plant" on page 5-6

About This Tutorial

In this section...
"Objectives" on page 5-2 "About the Model" on page 5-2

Objectives

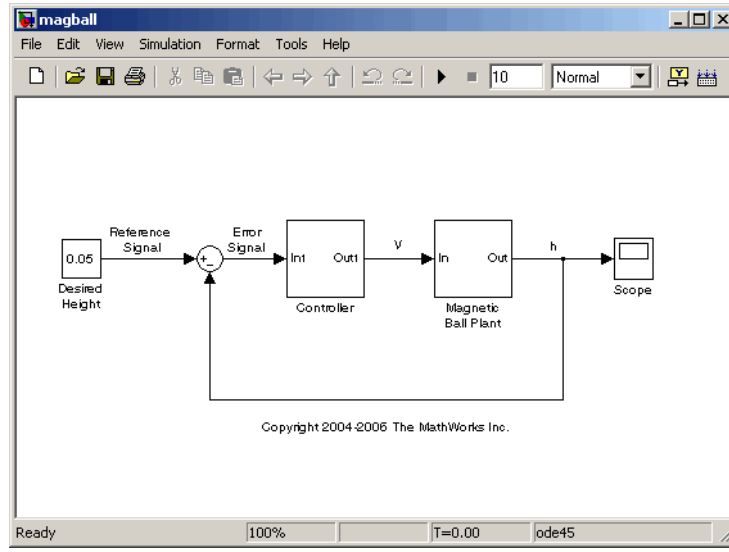
In this tutorial, you learn how to use Simulink Control Design functions at the command line to linearize a nonlinear plant in a single-loop control system about the existing operating point in the Simulink model.

About the Model

- "The magball Simulink Model" on page 5-2
- "Magnetic Ball Plant Subsystem" on page 5-3

The magball Simulink Model

The `magball` Simulink model shown in the following figure contains the nonlinear Magnetic Ball Plant and a controller in a single-loop feedback system.

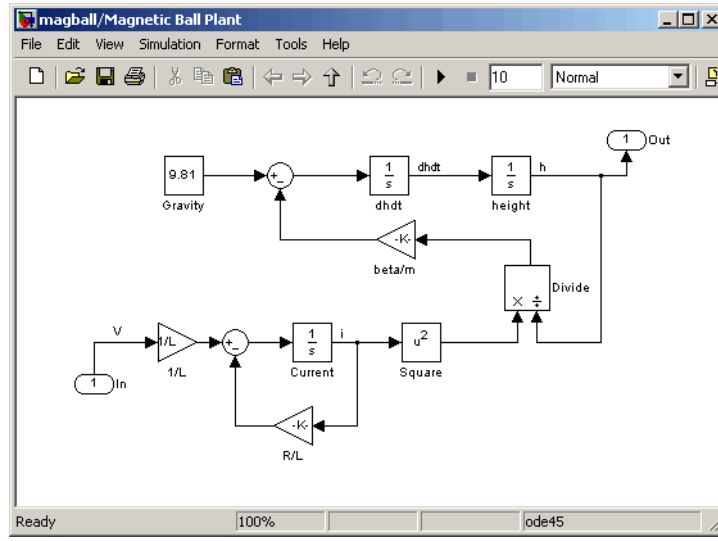


To view the model of the Magnetic Ball Plant subsystem, double-click the corresponding block in the `magball` model. The blocks in this model represent the mathematical system described in “Magnetic Ball Plant Subsystem” on page 5-3.

For information about creating Simulink models, see “Creating a Simulink Model”.

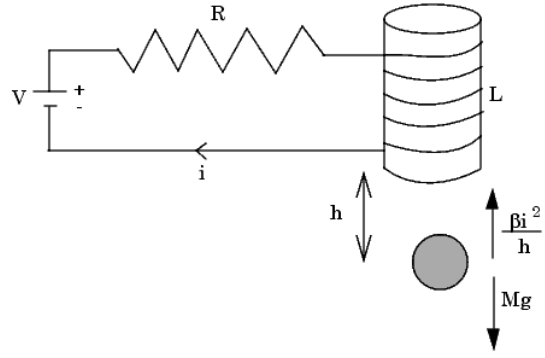
Magnetic Ball Plant Subsystem

The Magnetic Ball Plant subsystem of the `magball` model is shown in the following figure.



The Magnetic Ball Plant model represents an iron ball of mass *M*. This ball moves under the influence of the gravitational force, *Mg*, and an induced magnetic force, $\frac{\beta i^2}{h}$. The presence of the squared term in the induced magnetic force results in a nonlinear plant.

The inductor in the electric circuit, shown in the following figure, causes the induced magnetic force. This circuit also includes a voltage source and a resistor.



The following table describes the variables, parameters, differential equations, states, inputs, and outputs of the Magnetic Ball Plant subsystem.

Variables	h is the height of the ball. i is the current. V is the voltage in the circuit.
Parameters	M is the mass of the ball. g is the gravitational acceleration. β is a constant related to the magnetic force. L is the inductance of the coil. R is the resistance of the circuit.
Differential equations	The height of the ball, h, is described in the following equation: $$M\frac{d^2h}{dt^2} = Mg - \frac{\beta i^2}{h}$$ The current in the circuit, i, is described in the following equation: $$L\frac{di}{dt} = V - iR$$
States	h dh/dt i
Inputs	V
Outputs	h

Linearizing the Magnetic Ball Plant

In this section...
"Why Linearize a Nonlinear Plant?" on page 5-6
"Overview of the Linearization Process" on page 5-6
"How to Linearize the Magnetic Ball Plant" on page 5-7

Why Linearize a Nonlinear Plant?

Linearization is a linear approximation of a nonlinear system, based on the assumption that the system is approximately linear within a specific range of operation. This approximation is valid in a small region around the operating point of the system. An *operating point* is a set of inputs, outputs, and states that describe the operating conditions of a system. For more information about operating points and how they impact linearization, see "Why Are Operating Points Important?" on page 8-6.

In real-world problems, models are nonlinear. Because you need a linear, time-invariant model for most control design and analysis applications, you must linearize a nonlinear model before you can accomplish these goals.

For more information about linearization, see "What Is Linearization?" on page 9-2.

Overview of the Linearization Process

The process for linearizing the Magnetic Ball Plant in this tutorial includes the following tasks:

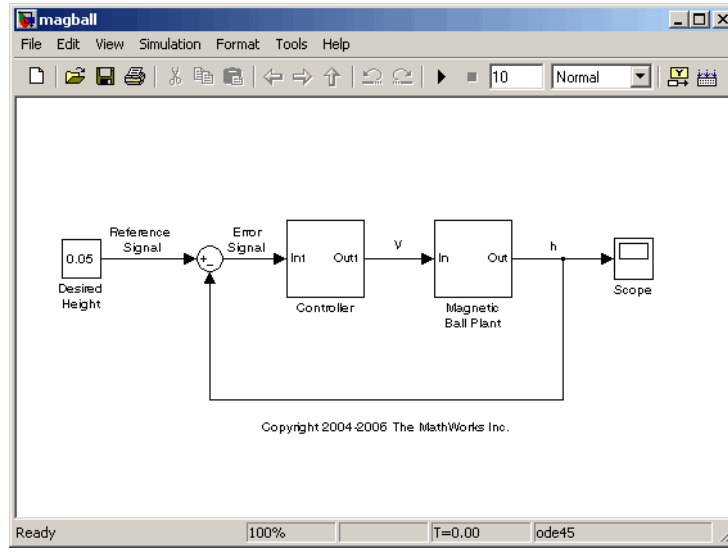
- Defining the portion of the model to linearize, also known as the *linearization path*.
- Removing the effects of a feedback loop in a single-loop control system.
- Linearizing about the existing operating point in the Simulink model.
- Viewing the linearization results in a step response plot and as state-space equations.

How to Linearize the Magnetic Ball Plant

1 Open the magnetic ball model by typing the following in the MATLAB Command Window:

```
magball
```

The model opens in Simulink as shown in the following figure.



2 Define the portion of the model to linearize using the `linio` command.

a Insert a linearization input point before the Magnetic Ball Plant by typing the following command:

```
magball_io(1)=linio('magball/Controller',1,'in')
```

This command creates the `magball_io` linearization I/O object in the MATLAB workspace. This object contains a linearization I/O setting for the input point.

```
Linearization IOs:
--------------------------
Block magball/Controller, Port 1 is marked with the following properties:
```

```
 - No Loop Opening
 - An Input Perturbation
 - No signal name. Linearization will use the block name
```

b Insert a linearization output point after the Magnetic Ball Plant by typing the following command:

```
magball_io(2)=linio('magball/Magnetic Ball Plant',1,'out')
```

This command updates the `magball_io` object to include a second linearization I/O setting for the output point.

```
Linearization IOs:
--------------------------
Block magball/Controller, Port 1 is marked with the following properties:
 - No Loop Opening
 - An Input Perturbation
 - No signal name. Linearization will use the block name


Block magball/Magnetic Ball Plant, Port 1 is marked with the following properties:
 - An Output Measurement
 - No Loop Opening
 - No signal name. Linearization will use the block name
```

3 Remove the effects of the feedback loop for open-loop analysis by typing the following command:

```
magball_io(2).OpenLoop='on'
```

This command updates the `magball_io` object to include a loop opening at the output signal of the Magnetic Ball Plant. Opening the loop ensures that the linearization result includes only the plant while preserving the model operating point. For more information on the effects of a feedback loop on linearization results, see “Performing Open-Loop Analysis” on page 9-9.

```
Linearization IOs:
--------------------------
Block magball/Controller, Port 1 is marked with the following properties:
 - No Loop Opening
 - An Input Perturbation
 - No signal name. Linearization will use the block name
```

```
Block magball/Magnetic Ball Plant, Port 1 is marked with the following properties:
 - An Output Measurement
 - A Loop Opening
 - No signal name. Linearization will use the block name
```

4 Perform the linearization using the `linearize` command by typing the following:

```
magball_lin=linearize('magball',magball_io)
```

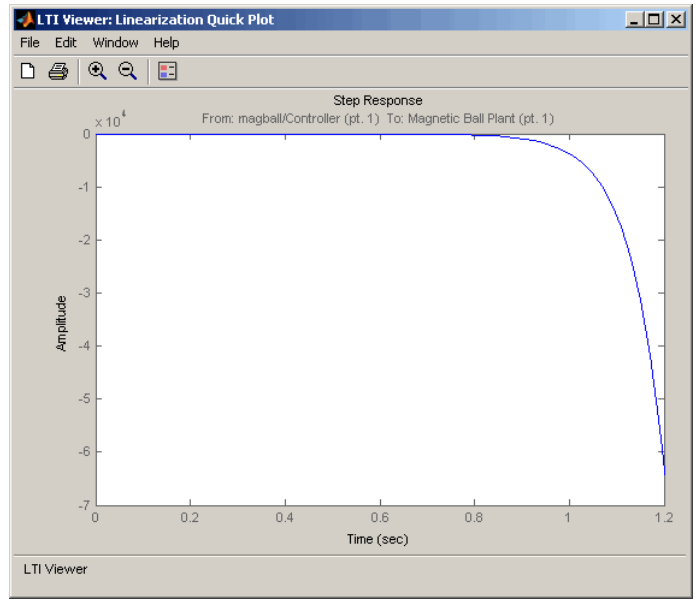
This command linearizes the portion of the model defined in the `magball_io` object about the operating point currently specified in the model. This command returns the following linearization result as a state-space object.

```
a =
                 magball/Magn  magball/Magn  magball/Magn
   magball/Magn             0             0             1
   magball/Magn             0          -100             0
   magball/Magn         196.2        -2.801             0

b =
                 magball/Cont
   magball/Magn             0
   magball/Magn            50
   magball/Magn             0

c =
                 magball/Magn  magball/Magn  magball/Magn
   Magnetic Bal             1             0             0

d =
                 magball/Cont
   Magnetic Bal             0

Continuous-time model.
```

5 View the step response of the linearized model by typing the following command:

```
ltiview(magball_lin)
```

This command opens the LTI Viewer GUI, which displays the step response plot of the linear model.



The step response decreases exponentially after about 0.8 seconds, which indicates that the plant model is unstable. The linear model provides an accurate approximation of the nonlinear Magnetic Ball Plant, which is also unstable.

Tip You can use the right-click menu of the LTI Viewer GUI to display a different plot or add characteristics to the response plot, such as peak response and settling time. For more information about working with response plots, see "LTI Viewer" in the *Control System Toolbox Getting Started Guide*.

Tip For information about designing a stabilizing controller for the Magnetic Ball Plant, see Chapter 10, "Designing Compensators".

---

6 Save the linearized model and I/O object of the `magball` model using the `save` command by typing the following:

```
save magball_project magball_lin magball_io
```

This command creates a file named `magball_project.mat` in the current directory.

Tip You can use the `load` command to reload this project.

6

Tutorial — Designing a PID Compensator for a Single-Loop Feedback System

- "About This Tutorial" on page 6-2
- "Designing a PID Compensator Using the Ziegler-Nichols Tuning Algorithm" on page 6-8
- "Tuning the PID Compensator Using Bode Graphical Tuning" on page 6-17
- "Simulating the Closed-Loop Simulink Model" on page 6-22

About This Tutorial

In this section...

"Objectives" on page 6-2

"About the Model" on page 6-2

"Requirements for the Compensator Design" on page 6-6

"Overview of the Compensator Design Process" on page 6-6

Objectives

In this tutorial, you learn how to use the Simulink Control Design GUI to design a PID compensator for a single-loop feedback system that is operating at the operating conditions specified in the Simulink model. You accomplish the following tasks:

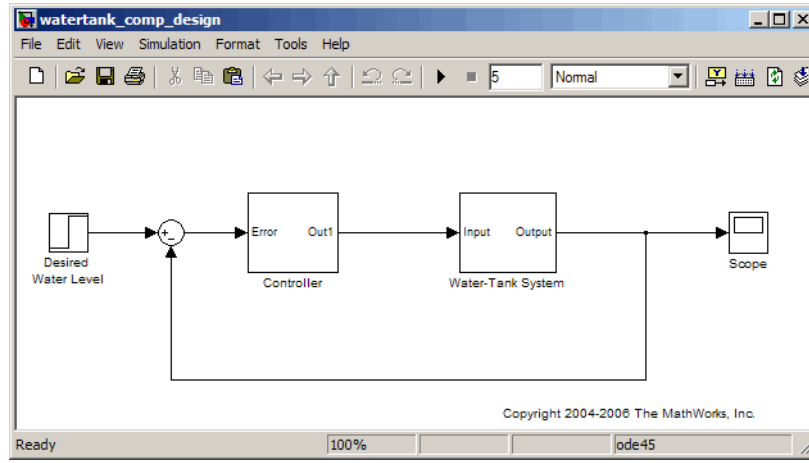
- Configure the model and GUI for compensator design.
- Design a PID compensator using Ziegler-Nichols tuning and Bode graphical design.
- Simulate the closed-loop nonlinear model.

About the Model

- "The watertank_comp_design Simulink Model" on page 6-3
- "Water-Tank Subsystem" on page 6-3
- "Controller Subsystem" on page 6-6

The watertank_comp_design Simulink Model

The `watertank_comp_design` model, shown in the following figure, contains the Water-Tank System plant and a simple proportional-integral-derivative (PID) controller, called Controller, in a single-loop feedback system.



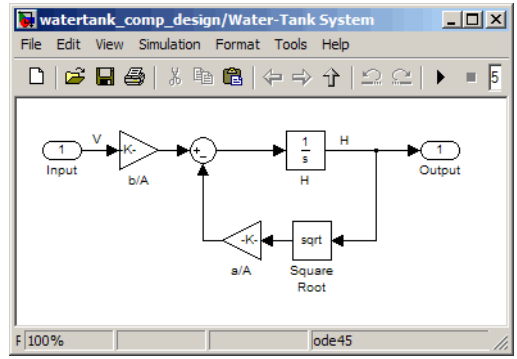
To view the Water-Tank System and the Controller, double-click the corresponding subsystem in the `watertank_comp_design` model. For descriptions of these subsystems, see the following topics:

- "Water-Tank Subsystem" on page 6-3
- "Controller Subsystem" on page 6-6

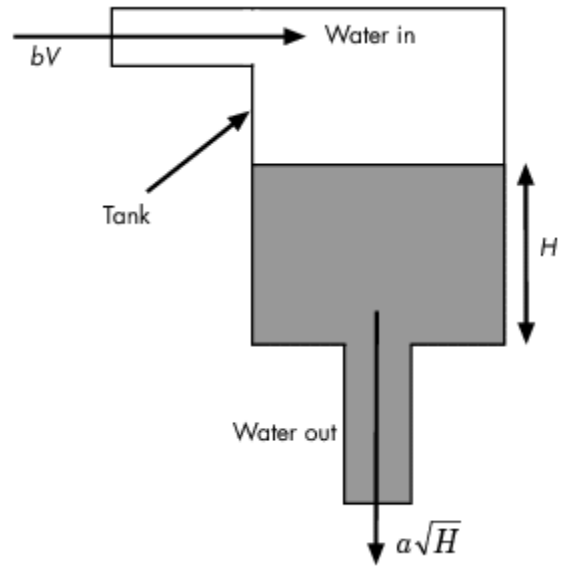
For information about creating Simulink models, see "Creating a Simulink Model".

Water-Tank Subsystem

The Water-Tank subsystem of the `watertank_comp_design` model appears in the following figure.



This model represents the water-tank system depicted in the following figure.



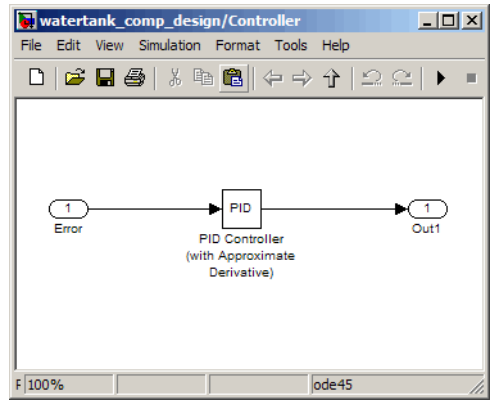
Water enters the tank from the top at a rate proportional to the voltage, V, applied to the pump. The water leaves through an opening in the tank base at a rate that is proportional to the square root of the water height, H, in the tank. The presence of the square root in the water flow rate results in a nonlinear plant.

The following table describes the variables, parameters, differential equations, states, inputs, and outputs of the water-tank system.

Variables	H is the height of water in the tank. Vol is the volume of water in the tank. V is the voltage applied to the pump.
Parameters	A is the cross-sectional area of the tank. b is a constant related to the flow rate into the tank. a is a constant related to the flow rate out of the tank.
Differential equation	$$\frac{d}{dt}Vol = A\frac{dH}{dt} = bV - a\sqrt{H}$$
States	H
Inputs	V
Outputs	H

Controller Subsystem

The Controller subsystem appears in the following figure.



This model contains a PID Controller block that controls the height of the water in the Water-Tank System.

Requirements for the Compensator Design

The compensator you design in this tutorial must control the Water-Tank System response as follows:

- The overshoot is less than 5%.
- The rise time is less than 5 seconds.

Overview of the Compensator Design Process

The process for designing a compensator for the Water-Tank System in this tutorial includes the following tasks:

- Configuring the model and GUI for the design.
- Designing a PID compensator using Ziegler-Nichols tuning.
- Tuning the compensator using the Bode design technique.
- Simulating the closed-loop Simulink model with the compensator design to analyze the system dynamics.

Simulink Control Design tools support linear control design. Although the Water-Tank System is nonlinear, you do not need to linearize this nonlinear plant model as a separate step–Simulink Control Design software automatically linearizes the model about the model operating point when you do not specify a different operating point. The linearization provides a valid approximation of the nonlinear model in a region around the operating point. For more information about linearization and how the operating point impacts linearization results, see “What Is Linearization?” on page 9-2 and “Why Are Operating Points Important?” on page 8-6

Designing a PID Compensator Using the Ziegler-Nichols Tuning Algorithm

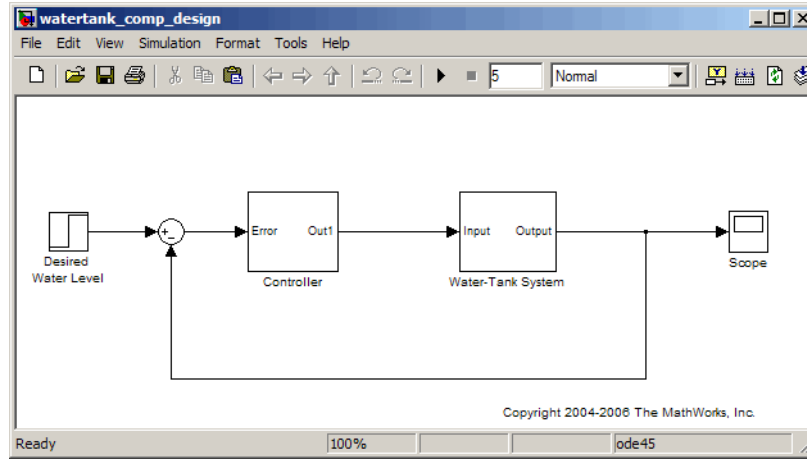
In this portion of the tutorial, you design a compensator using the automated PID Ziegler-Nichols open-loop tuning algorithm. This tuning method computes the proportional, integral, and derivative gains using the Chien-Hrones-Resnick (CHR) setting with a 20% overshoot.

To design a PID compensator:

1. Open the `watertank_comp_design` model by typing the model name in the MATLAB Command Window:

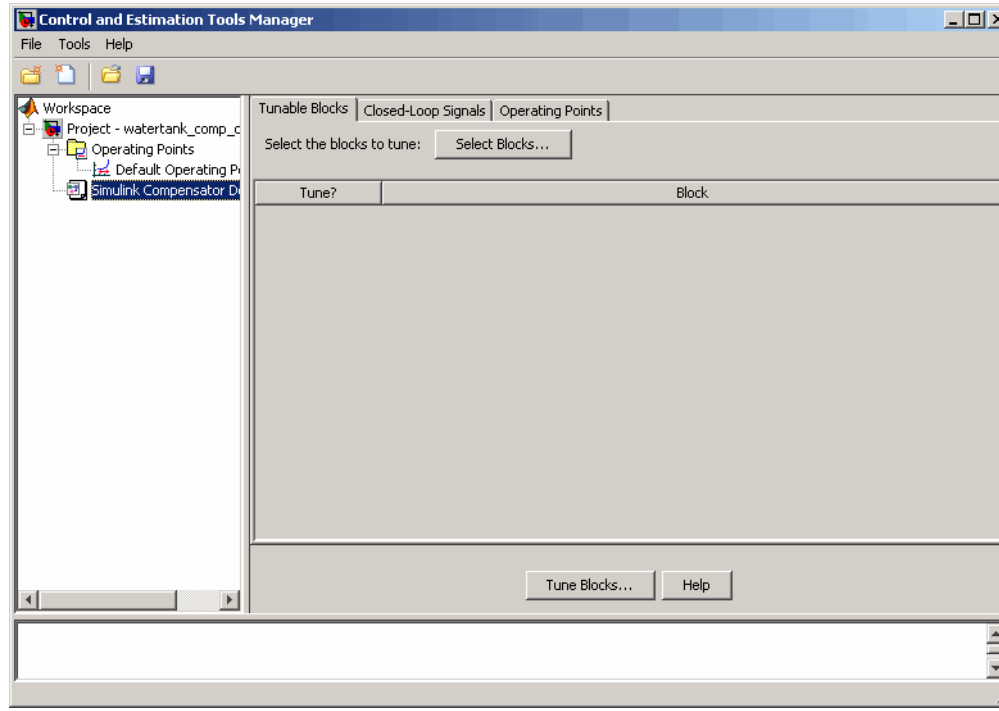
   ```
   watertank_comp_design
   ```

   The command opens the `watertank_comp_design` model in Simulink, as shown in the following figure.



2. In the `watertank_comp_design` model window, select **Tools > Control Design > Compensator Design**.

   This action opens the Control and Estimation Tools Manager with the **Simulink Compensator Design Task** node selected.



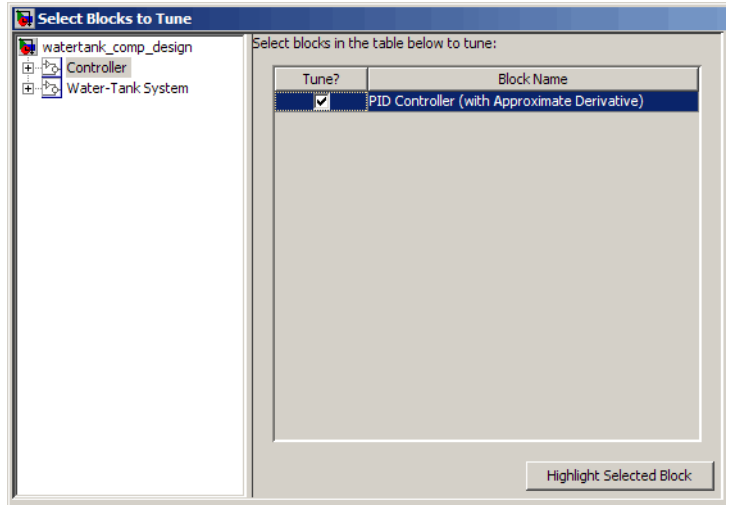
3 Select the PID Controller block as the block to tune.

a In the **Tunable Blocks** tab, click **Select Blocks**.

This action opens the Select Blocks to Tune window.

b In the **watertank_comp_design** tree, select the **Controller** subsystem.

c Select the **Tune?** checkbox for **PID Controller (with Approximate Derivatives)**.



d Click **OK**.

4 Define the closed-loop systems for which you want to analyze the response.

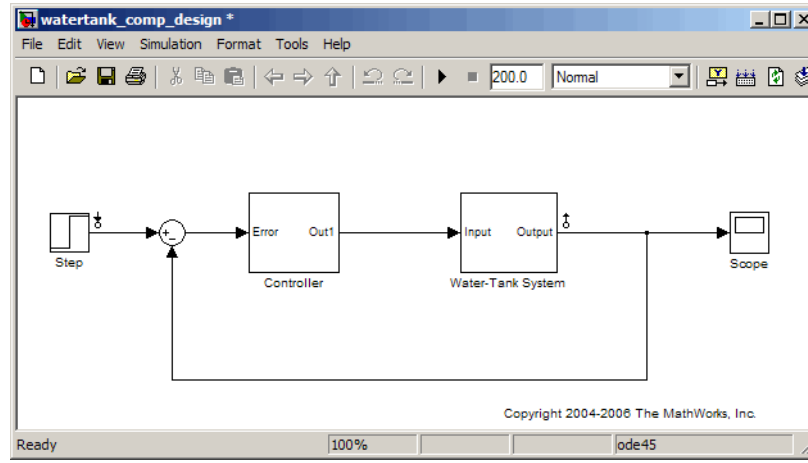
a In the `watertank_comp_design` model, right-click the output of the Desired Water Level block, and select **Linearization Points > Input Point**.

This action displays the symbol on the signal line. This symbol indicates the input of the closed-loop path.

b Right-click the output signal from the Water-Tank System, and select **Linearization Points > Output Point**.

This action displays the symbol on the signal line. This symbol indicates the output of the closed-loop path.

The Simulink model now resembles the following figure.



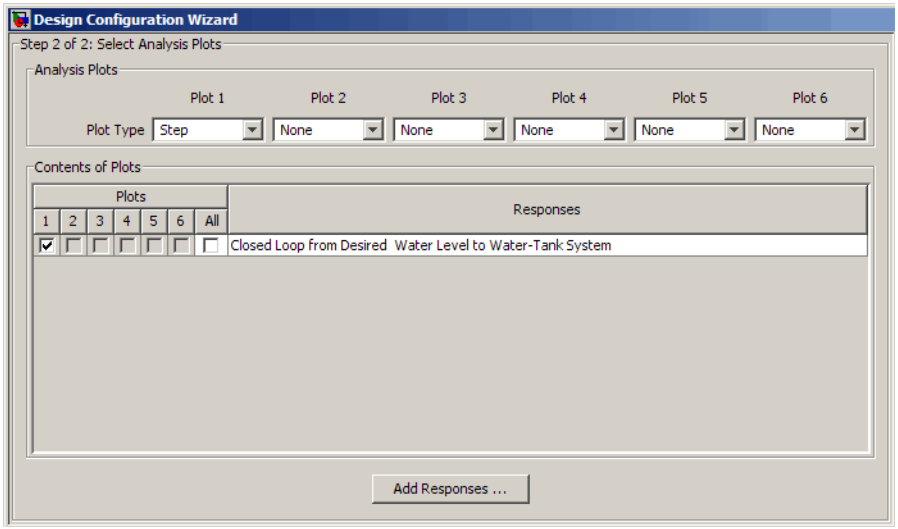
5 In the Control and Estimation Tools Manager, click **Tune Blocks** to open the Design Configuration Wizard. Click **Next**.

6 In Step 1 of the Design Configuration Wizard accept the default settings and click **Next**.

7 In Step 2 of the Design Configuration Wizard, specify the type of plot for analyzing the response.

   a In the **Analysis Plots** area, select `Step` for the **Plot Type** corresponding to **Plot 1**.

   b In the **Plots** section of the **Contents in Plots** pane, select **1** for **Closed Loop from Step to Water-Tank System**.



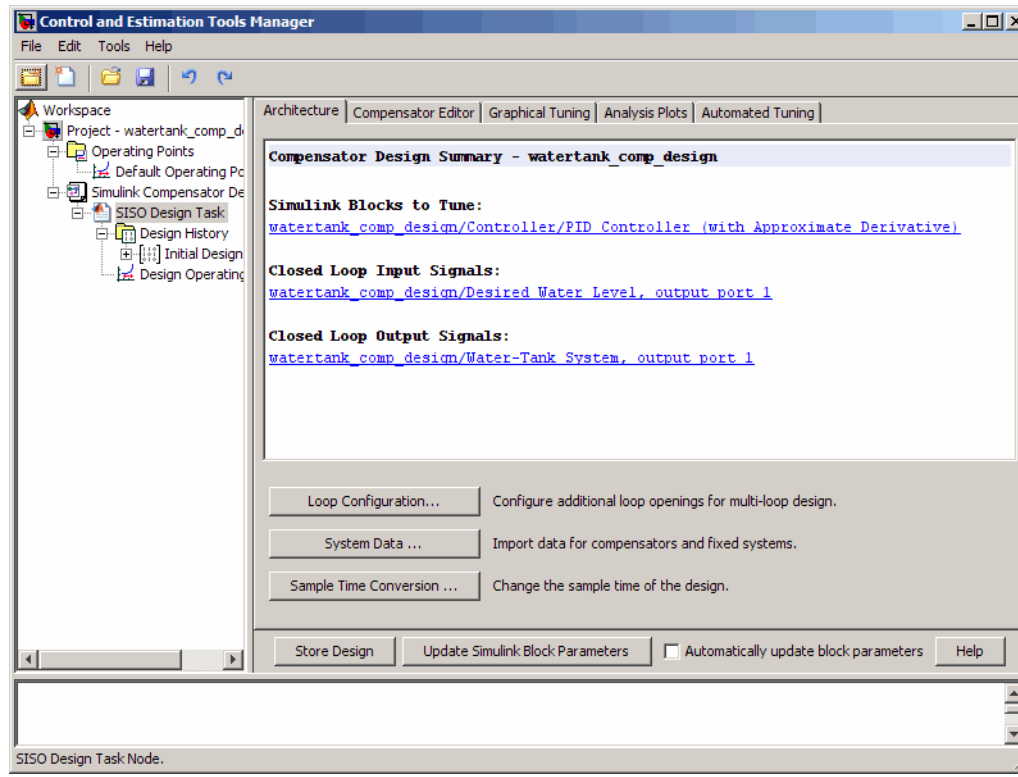
8 Click **Finish**.

The software performs the following actions:

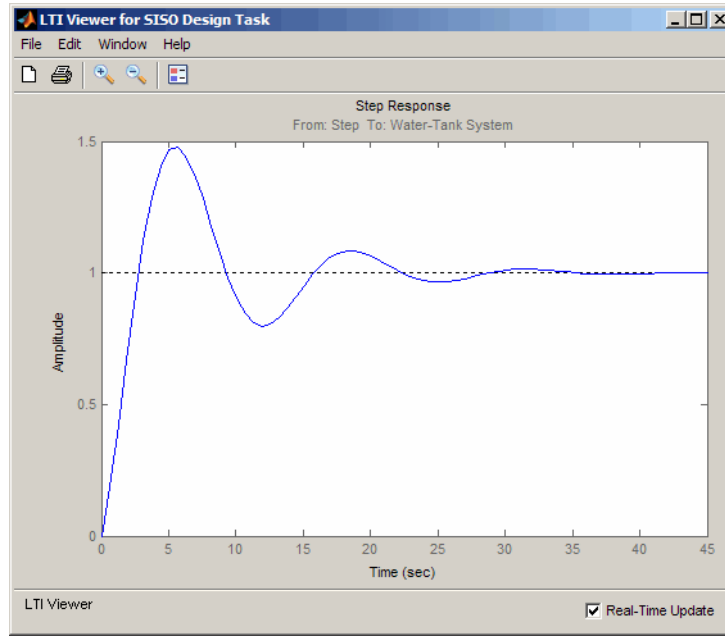
- Linearizes the Simulink model about the operating point specified in the model.
- Creates a **SISO Design Task** node under the **Simulink Compensator Design Task** node.
- Opens the following plot windows:
  - LTI Viewer for SISO Design Task window, which shows the closed-loop Step Response plot of the linearized model
  - SISO Design for SISO Design Task window, which is empty

    You do not use in this window in this section of the tutorial. Keep this window open for the next section of the tutorial.

The Control and Estimation Tools manager resembles the following figure.



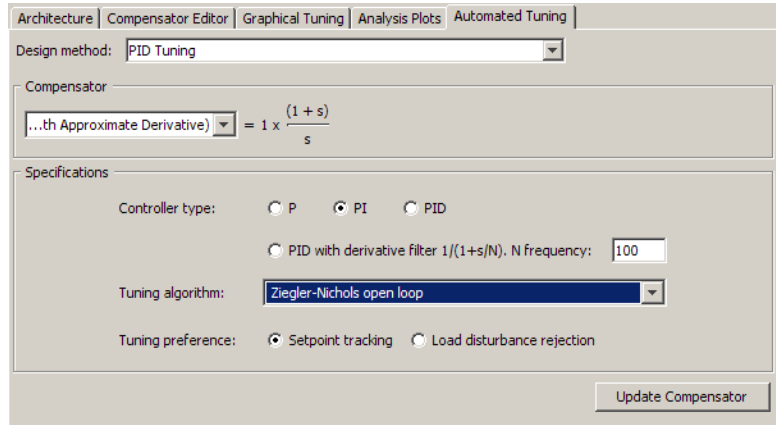
The Step Response plot shows an overshoot that does not meet the overshoot design requirement of less than 5%.



9 In the **Automated Tuning** tab of the **SISO Design Task** node in the Control and Estimation Tools Manager, select `PID Tuning` as the **Design method**.

10 In the **Specifications** area, select the following options:

- **Controller type**: `PI`
- **Tuning algorithm**: `Ziegler-Nichols open loop`



11 Click **Update Compensator**.

This action computes the PI values for the compensator using the Ziegler-Nichols open-loop tuning algorithm and updates the Step Response plot.

Tip You can view the PI values in the **Parameter** tab of the **Compensator Editor** tab in the **SISO Design Task** node.

---

12 Evaluate whether the compensator design meets the design requirements by analyzing the overshoot and the rise time, as follows:

a Right-click the Step Response plot and select the following options:

- **Characteristics > Peak Response**
- **Characteristics > Rise Time**

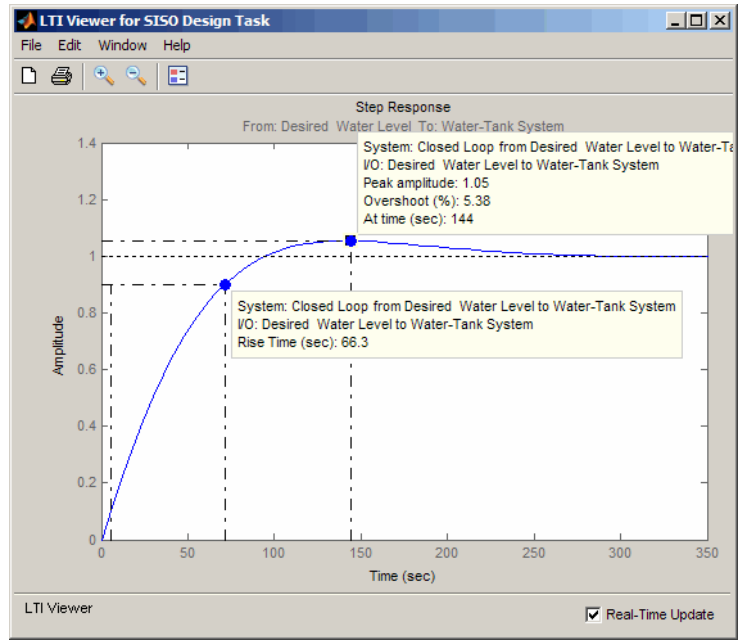
  These actions add a plot marker to the plot for each characteristic, shown as blue dots.

b Left-click each blue dot to open the corresponding data marker.

The data markers show the following response characteristics:

- The overshoot is 5.38%.

- The rise time is 66.3 seconds.



This system response with the PID compensator slightly exceeds the maximum allowed overshoot of 5%. The rise time is much slower than the required rise time of 5 seconds.

You decrease the rise time by increasing the gain of the compensator, as described in “Tuning the PID Compensator Using Bode Graphical Tuning” on page 6-17.

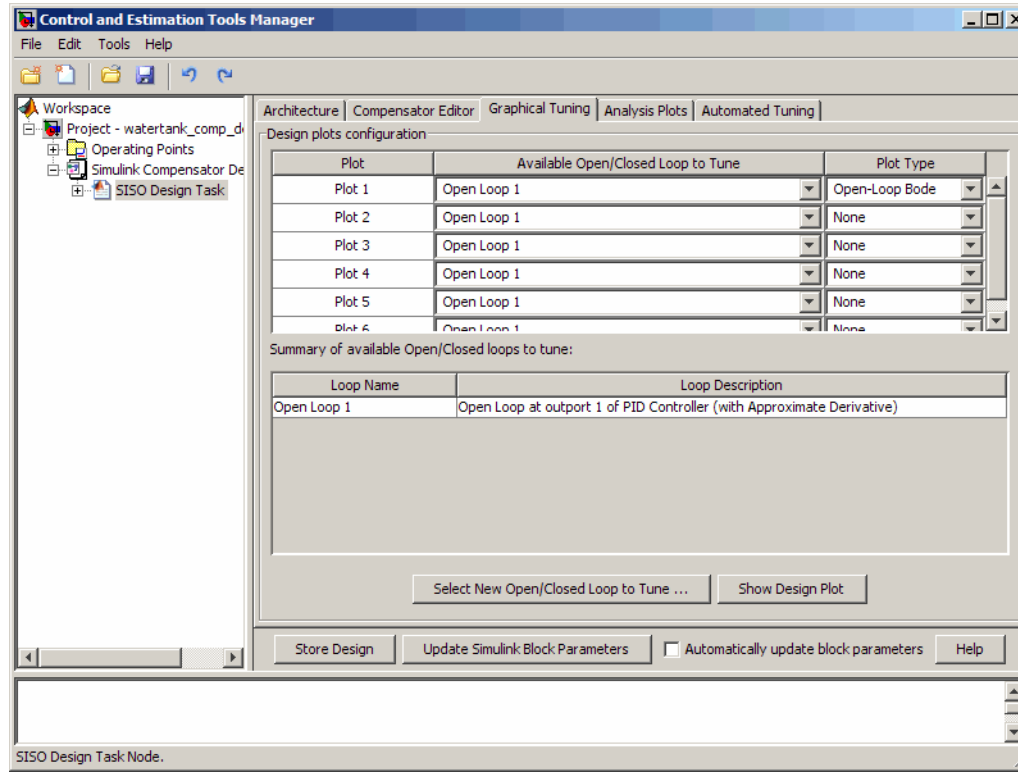
Tuning the PID Compensator Using Bode Graphical Tuning

In this portion of the tutorial, you decrease the rise time of the Water-Tank System response by increasing the gain in the compensator using Bode graphical tuning. Bode graphical tuning lets you design a compensator by manipulating Bode diagrams of the open-loop response. This process is also called loop shaping.

You must have already designed an initial compensator using PID tuning, as described in "Designing a PID Compensator Using the Ziegler-Nichols Tuning Algorithm" on page 6-8.

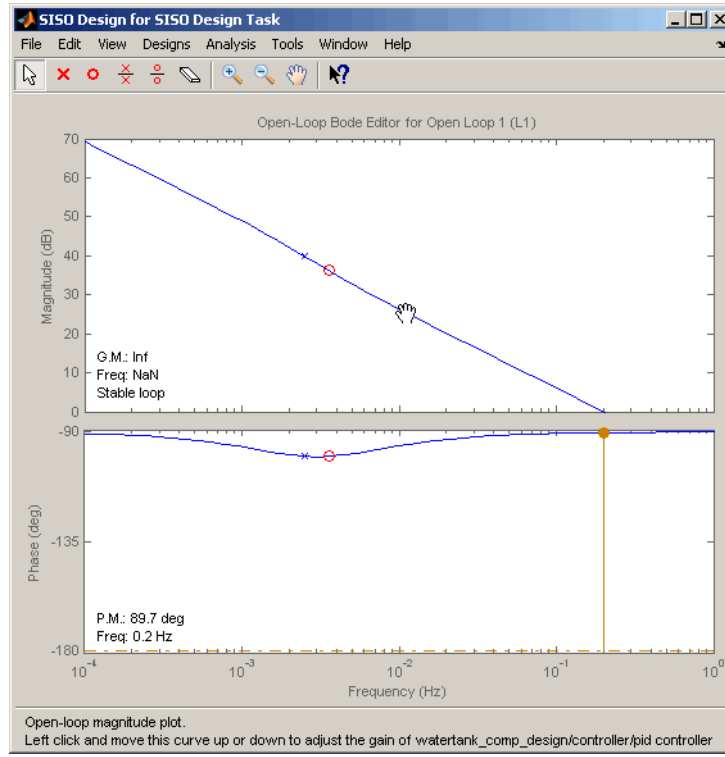
To design a compensator using Bode graphical tuning:

1. In the Control and Estimation Tools Manager, select the **Graphical Tuning** tab of the **SISO Design Task** node.
2. In the **Plot Type** cell that corresponds to **Plot 1**, select **Open-Loop Bode**.



This action creates an Open-Loop Bode plot in the SISO Design for SISO Design Task window. This plot shows a Bode plot of the linearized model with the compensator designed using automated PID tuning.

3 In the SISO Design window, drag the Bode Magnitude line upward to increase the gain. As you adjust the gain, view the affects on the closed-loop response in the Step Response plot.



By increasing the gain, you increase the bandwidth and speed up the response. One possible compensator design that meets the tutorial requirements has the following parameters:

- P = `5.0368`
- I = `0.11434`
- D = `0`

Tip You can view the parameter values corresponding to the gain adjustment you made in the Bode Magnitude plot in the **Compensator Editor** tab of the SISO Design Task. You can also adjust the parameter values in this tab.

4 Evaluate whether the compensator design meets the design requirements by analyzing the overshoot and the rise time, as follows:

a Right-click the Step Response plot and select the following options, if you have not done so already:

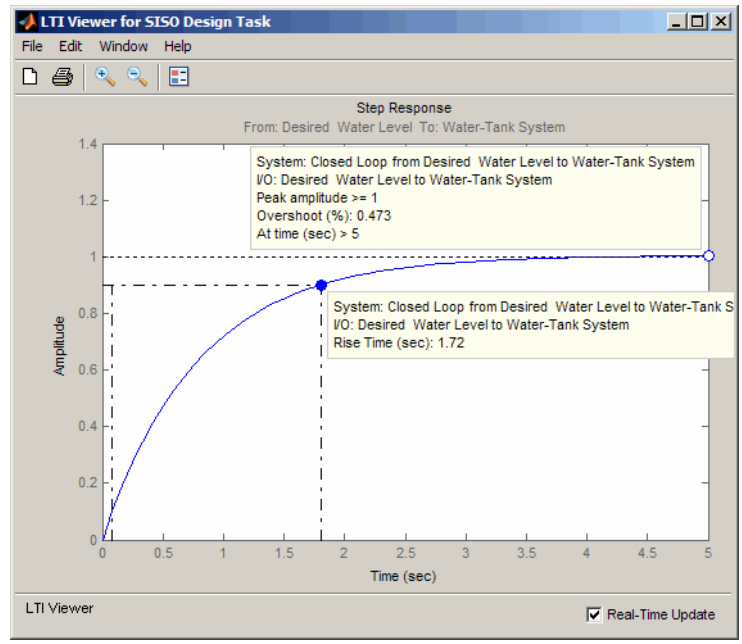
- **Characteristics > Peak Response**
- **Characteristics > Rise Time**

  These actions add a plot marker to the plot for each characteristic, shown as blue dots.

b Left-click each blue dot to open the corresponding data marker.

The data markers show the following response characteristics:

- The overshoot is 0.437%.
- The rise time is 1.72 seconds.



This compensator design satisfies the design requirements of less than 5% overshoot and less than 5 second rise time.

Simulating the Closed-Loop Simulink Model

In this portion of the tutorial, you simulate the nonlinear closed-loop Simulink model that includes a PID compensator to determine if the design meets the requirements.

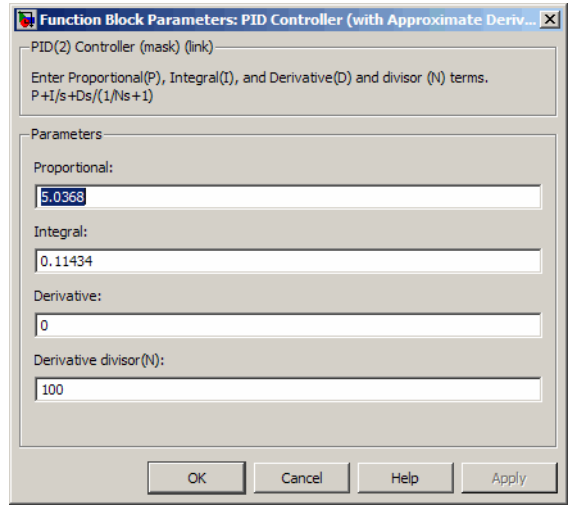
You must have already designed the compensator, as described in "Tuning the PID Compensator Using Bode Graphical Tuning" on page 6-17.

To simulate the model:

1. In the Control and Estimation Tools Manager **SISO Design Task** node, click **Update Simulink Block Parameters**.

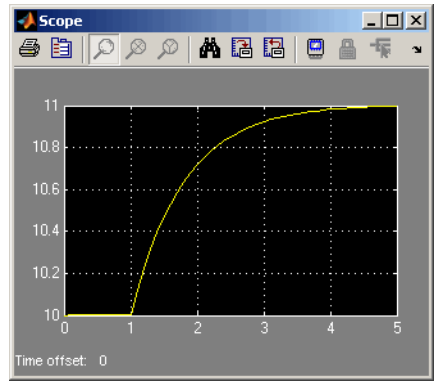
   This action writes the compensator parameters into the PID Controller block of the Controller subsystem in the Simulink model.

Tip You can view the PID Controller block parameters in the Function Block Parameters Dialog box. To open this dialog box, double-click the PID Controller block.



2 In the Simulink model, double-click the Scope block to open the Scope block window.

3 In the Simulink model, click to simulate the model. Then, click to autoscale the axis.



This action updates the Scope window with the response of the nonlinear model with the compensator design. This simulation shows that the rise time is less than 5 seconds and there is minimal overshoot. Thus, this compensator design meets the requirements of less than 5% overshoot and less than 5 second rise time.

7

Beginning a Project

This chapter introduces a Simulink model of a magnetic ball system that the remaining sections use to illustrate the Simulink Control Design features. A linearization task is created for the model in the Control and Estimation Tools Manager, a graphical environment that manages the Simulink Control Design linearization and compensator design projects. The key features of a project in the Control and Estimation Tools Manager are also presented.

- "Creating or Opening a Simulink Model" on page 7-2
- "Beginning a New Linearization Task" on page 7-7

Creating or Opening a Simulink Model

In this section...

"Details of Possible Models" on page 7-2

"Example Model: The Magnetic Ball System" on page 7-2

Details of Possible Models

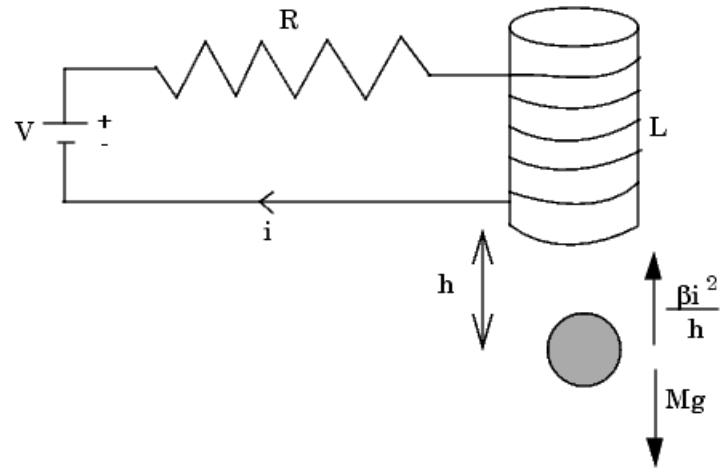
The first step in the linearization or compensator design process is to create or open a Simulink model of your system. The model can have any number of inputs and outputs (including none), and any number of states. The model can include user-defined blocks or S-functions. Your model can involve multiple compensators in addition to the plant, multiple feedback loops, and any number of subsystems.

Example Model: The Magnetic Ball System

This section introduces an example model, the magnetic ball system, that the remaining sections and chapters use to illustrate the process of linearizing a model or designing a compensator.

Magnetic Ball System

The electronic circuit in the following figure consists of a voltage source, a resistor, and an inductor in the form of a tightly wound coil. An iron ball beneath the inductor experiences a gravitational force as well as an induced magnetic force (from the inductor) that opposes the gravitational force.



Model Equations

A differential equation for the force balance on the ball is given by

$$M\frac{d^2h}{dt^2} = Mg - \frac{\beta i^2}{h}$$

where M is the mass of the ball, h is the height (position) of the ball, g is the acceleration due to gravity, i is the current, and ß is a constant related to the magnetic force experienced by the ball. This equation describes the height, h, of the ball due to the unbalanced forces acting upon it.

The current in the circuit also varies with time and is given by the following differential equation

$$L\frac{di}{dt} = V - iR$$

where L is the inductance of the coil, V is the voltage in the circuit, and R is the resistance of the circuit.

The system of equations has three states:

$$h, \frac{dh}{dt}, i$$

The system also has one input (V), and one output (h). It is a nonlinear system due to the term in the equation involving the square of i and the inverse of h.

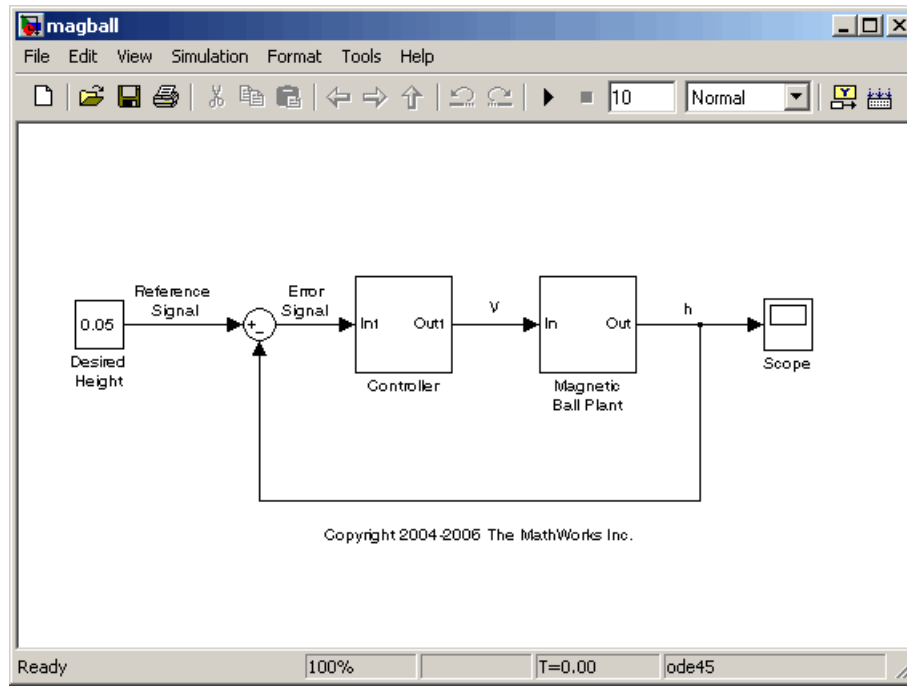
Due to its nonlinearity, you cannot analyze this system using methods for linear-time-invariant (LTI) systems such as step response plots, bode diagrams, and root-locus plots. However, you can linearize the model using theSimulink Control Design software to approximate the nonlinear system as an LTI system. Linearization also occurs automatically when designing a compensator. This linearized system can then use the LTI Viewer for display and analysis and the SISO Design Tool for compensator design. Refer to "Purpose of Linearization" on page 1-2 for a discussion of the uses of linearized models and "Linearization of Simulink Models" on page 9-3 for a discussion of the linearization process.

Opening the Model

To open the model for the magnetic ball example, type

```
magball
```

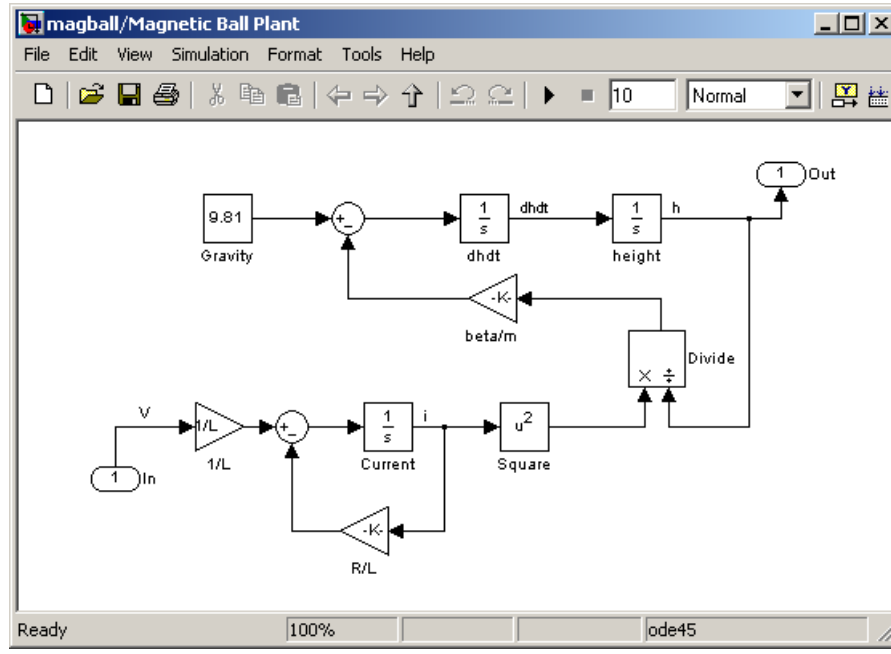
at the MATLAB prompt. The magnetic ball system opens in the Simulink model viewer as shown in this figure.



The `magball` model consists of

- The magnetic ball system itself, within the subsystem labeled Magnetic Ball Plant.
- A Controller subsystem that controls the height of the ball by balancing the forces acting on it.
- A reference signal that sets the desired height of the ball.
- A `Scope` block that displays the height of the ball as a function of time.

Double-click a block to view its contents. The Controller block contains a zero-pole-gain model. The Magnetic Ball Plant block is shown in this figure.



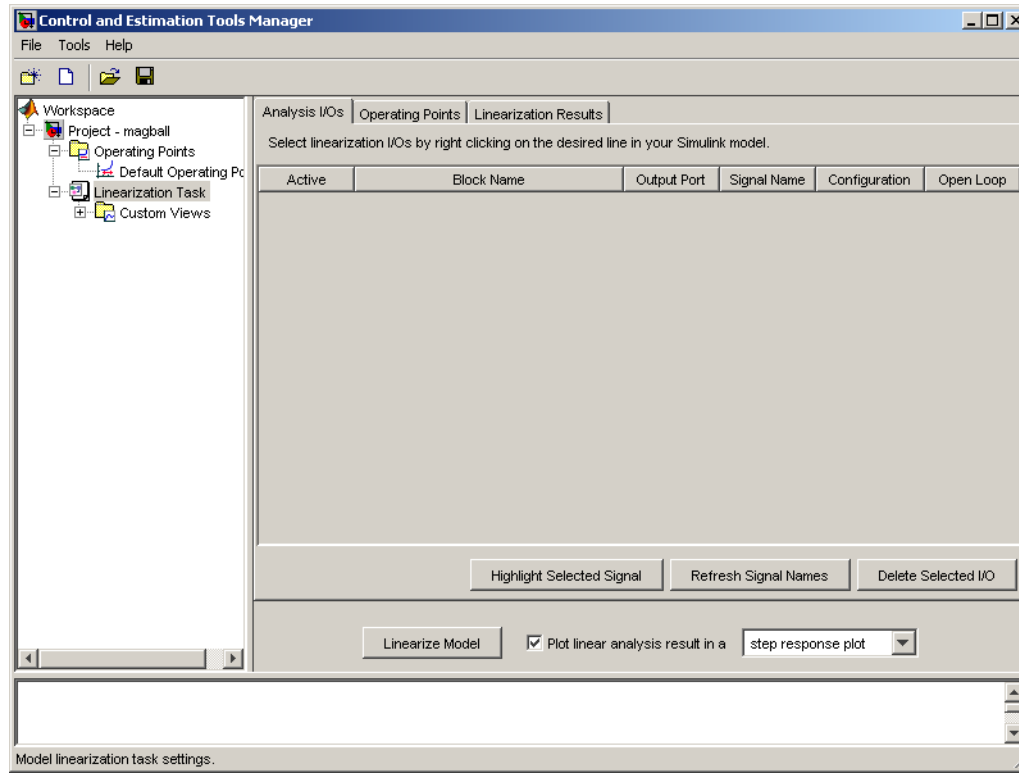
The input to the Magnetic Ball Plant system, which is also the output of the Controller subsystem, is the voltage, *V*. The output is the height of the ball, *h*. The system contains three states within the three integrators: `height`, `dhdt`, and `Current`.

Values of the parameters are given as M=0.1 kg, g=9.81 m/s^2, R=2 Ohm, L=0.02 H, and β=0.001.

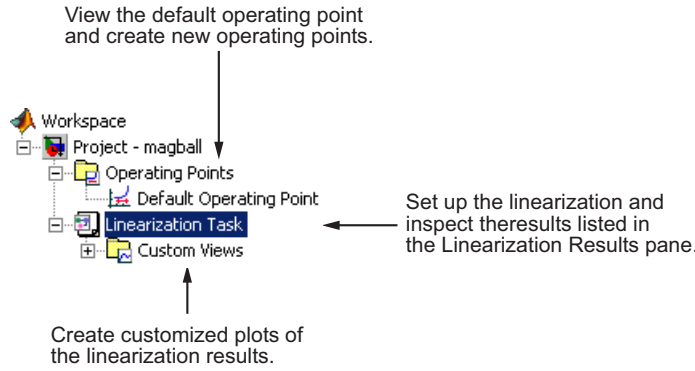
Beginning a New Linearization Task

The Control and Estimation Tools Manager provides a graphical environment for performing, and managing, control and estimation tasks such as linearization and compensator design. Within this environment you can create projects which can include several tasks, including Simulink® Parameter Estimation™ tasks and Model Predictive Control Toolbox tasks, when they are installed.

To open a new project within the Control and Estimation Tools Manager for linearization of the magball model, select **Tools > Control Design > Linear Analysis** from the `magball` window. The Control and Estimation Tools Manager opens and creates a new linearization task, as shown in the following figure.



The left pane of the Control and Estimation Tools Manager shows the project tree, which contains all your current projects. At this stage you should have just one project, **Project - magball**. Select a node within the tree to display its contents in the pane on the right.



- For information on the **Operating Points** node or the **Operating Points** pane within the **Linearization Task** node, refer to Chapter 8, "Specifying Operating Points".
- For information on the **Analysis I/Os** pane within the **Linearization Task** node, refer to "Configuring Inputs and Outputs for the Linearized Model" on page 9-5.
- For information on the **Linearization Results** pane within the **Linearization Task** node and inspecting linearization results, refer to "Viewing Linearization Results" on page 9-18.
- For information on **Custom Views**, refer to "Viewing Linearization Results" on page 9-18.

8

Specifying Operating Points

- "What Are Operating Points?" on page 8-2
- "Why Are Operating Points Important?" on page 8-6
- "Creating Operating Points" on page 8-9
- "Exporting and Saving Operating Points" on page 8-21

What Are Operating Points?

In this section...

"Definition of an Operating Point" on page 8-2

"Equilibrium Operating Points" on page 8-2

"Simulink Model Operating Points" on page 8-3

Definition of an Operating Point

The *operating point* of a dynamic system defines its overall *state* at a given time. For example, in a model of a car engine, variables such as engine speed, throttle angle, engine temperature, and surrounding atmospheric conditions typically describe the operating point. It is important to specify the operating point accurately because it affects the system's behavior. For example, the behavior of a car engine can vary greatly when it operates at high or low elevations.

Equilibrium Operating Points

An *equilibrium operating point* remains steady and constant with time; all states in the model are at equilibrium. It is also known as a *steady state* or *trimmed operating point*. For example, a car operating on cruise control on a flat road maintains a constant speed. Its operating point is steady, or at equilibrium.

A hanging pendulum provides an example of a stable equilibrium operating point. When the pendulum hangs straight down, its position does not change with time because it is at an equilibrium position. When its position deviates slightly from this position, it always returns to the equilibrium; small changes in the operating point do not cause the system to leave the region of good approximation around the equilibrium value.

A pendulum that points upward provides an example of an unstable equilibrium operating point. As long as the pendulum points *exactly* upward, it is steady at this equilibrium state. However, when the pendulum deviates slightly from this state, it swings downward and the operating point leaves the region around the equilibrium value.

Simulink Model Operating Points

This section includes the following topics:

- "Operating Point Versus Simulink Full-Model Operating Point" on page 8-3
- "Example of a Simulink Model Operating Point" on page 8-4

Operating Point Versus Simulink Full-Model Operating Point

The Simulink full model operating point includes information from all of the blocks in a Simulink model. When you use the Simulink Control Design GUI or the MATLAB command line to create operating points for a model, you are actually creating an *operating point object* (what is referred to as the operating point). The operating point is a subset of the Simulink full-model operating point.

The following table shows the different types of blocks that make up a full-model operating point and which types are included in the operating point.

Makeup of Simulink Full-Model Operating Point

Block Types	Example of Blocks	Included in Operating Point?
Blocks with double value states and root level inport blocks with double data type	Integrator, State Space, Transfer Function, Inport	Yes
Root level inport blocks with nondouble or complex data type	Inport	No
Blocks with internal state representation that impacts block outputs	Backlash, Memory, Stateflow	No
Source blocks with outputs specified by block dialog parameters	Constant, Step	No

The operating point includes only the block information most commonly redefined by users. This information provides you with three places to make changes to the parameters of these common blocks in your model:

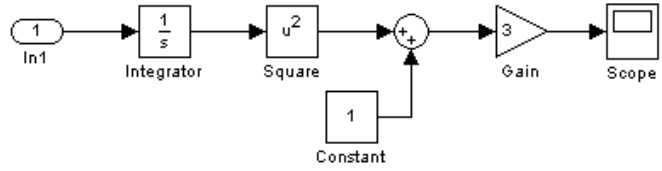
- Directly in the Simulink Control Design Control and Estimation Tools Manager GUI
- At the MATLAB command line
- In the Simulink model

Note If you want to make changes to any block not included in the operating point, you must make the changes directly in the Simulink model.

Example of a Simulink Model Operating Point

The following figure shows a simple Simulink model that has one block with state (the integrator block) and therefore one state, `x1`. This state depends on the initial conditions set in the integrator block. The value of this state and the input from the inport block define the operating point in the Simulink Control Design software.

The square block output is derived from just the initial condition of the integrator block while the gain block has its output derived from both the initial condition of the integrator block and output of the constant block. The derivative dx/dt of the integrator block state is defined by the output level of the root inport block.



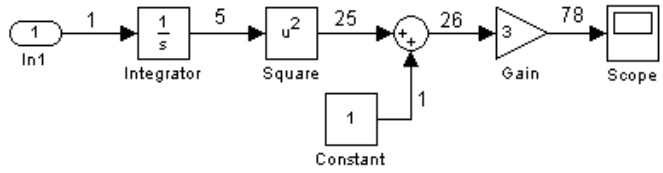
The state, `x1`, defines the signal levels at the input and output of every block in the model by propagating through the blocks in the model as follows.

1 The integrator block initial condition of `x0 = 5` sets the state `x1 = 5`.

2 The state, `x1`, propagates in the direction of the arrows through the blocks in the model, defining input and output signals on each block, as described in the following table.

Block	Input Signal Level	Block Operation	Output Signal Level
Square	5	squares	25
Sum	25 from square, 1 from constant	sums	26
Gain	26	multiplies by 3	78

The following figure shows these input and output signal levels for each block.



Why Are Operating Points Important?

In this section...

"Impact of Operating Points" on page 8-6

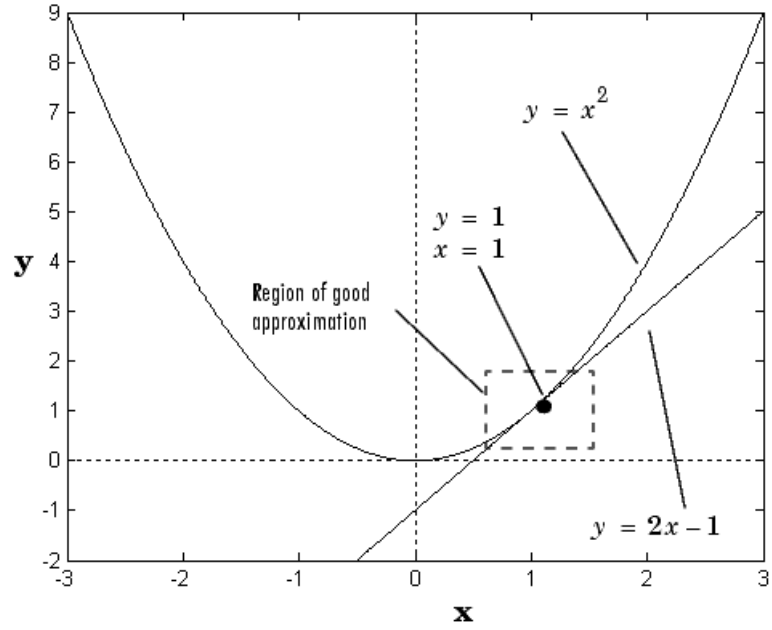
"Choosing an Operating Point for Accurate Linearization" on page 8-7

Impact of Operating Points

A linearized model is an approximation that is valid in a region around the operating point of the system where the linearization took place. Near the operating point the approximation is good, while far away it may be poor. A linearized model of a car being operated at 3000 ft. is very accurate at elevations close to 3000 ft. but less accurate as the car travels higher or lower.

Example of a Linear Approximation of a Nonlinear Function

The following figure shows a nonlinear function, $y = x^2$, and a linear function, $y = 2x - 1$. The linear function is an approximation to the nonlinear function about the operating point x=1, y=1. Near this operating point, the approximation is good. Away from this operating point, the approximation is poor. The precise boundaries of this region are often somewhat arbitrary. The following figure shows a possible region of good approximation for the linearization of $y = x^2$.



Choosing an Operating Point for Accurate Linearization

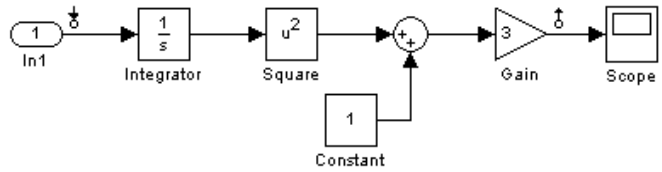
Your choice of operating point is important when you:

- Linearize a Simulink model. The choice of operating point determines the accuracy of the linear approximation.
- Designing compensators with Simulink Control Design software. A Compensator Design Task uses linearization when analyzing a Simulink model.

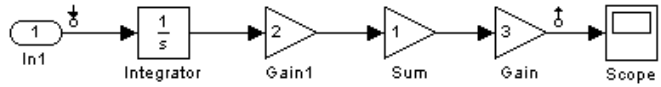
Tip Choose an operating point that is very close to the expected operating values of the system. One option is to use an equilibrium operating point, described in "Equilibrium Operating Points" on page 8-2.

Example of Linearization Results About Two Different Operating Points

A model can have two entirely different linearizations when the linearization is performed about different operating points. The following model can be linearized using the Simulink Control Design software.



The linearization result for this model is shown in the following figure.



When this linearization is performed about two different operating points, two different linearization results occur as shown in the following table.

Operating Point	Linearization Result
Initial Condition = 5, State `x1` = 5	30/s
Initial Condition = 0, State `x1` = 0	0

Note The operating point consists of values for *all* the states in the model, although only those states between the linearization points are linearized. The whole model contributes to the operating point values of the states, inputs, and outputs of the portion of the model you are linearizing.

Creating Operating Points

In this section...

"Simulink® Control Design Default Operating Point" on page 8-9

"Creating Operating Points from Specifications" on page 8-9

"Creating Operating Points from Known Values" on page 8-15

"Creating Operating Points from Simulation" on page 8-18

Simulink Control Design Default Operating Point

A default operating point is automatically created and stored in a node in the project tree every time you open the Control and Estimation Tools Manager unless you indicate otherwise. The default operating point consists of a snapshot of initial condition variables in the Simulink model. Initial condition variables are a special set of variables that can be set in the Simulink model. These variable include initial conditions for blocks with state, such as the integrator or state space blocks, and output values of root level inport blocks. For more information on setting these variables, see "Importing and Exporting Data" in the Simulink User's Guide.

By clicking on the **Default Operating Point** node, you can view these values and modify them for use in a linearization. The output levels for blocks like a constant block remain defined in the model.

Creating Operating Points from Specifications

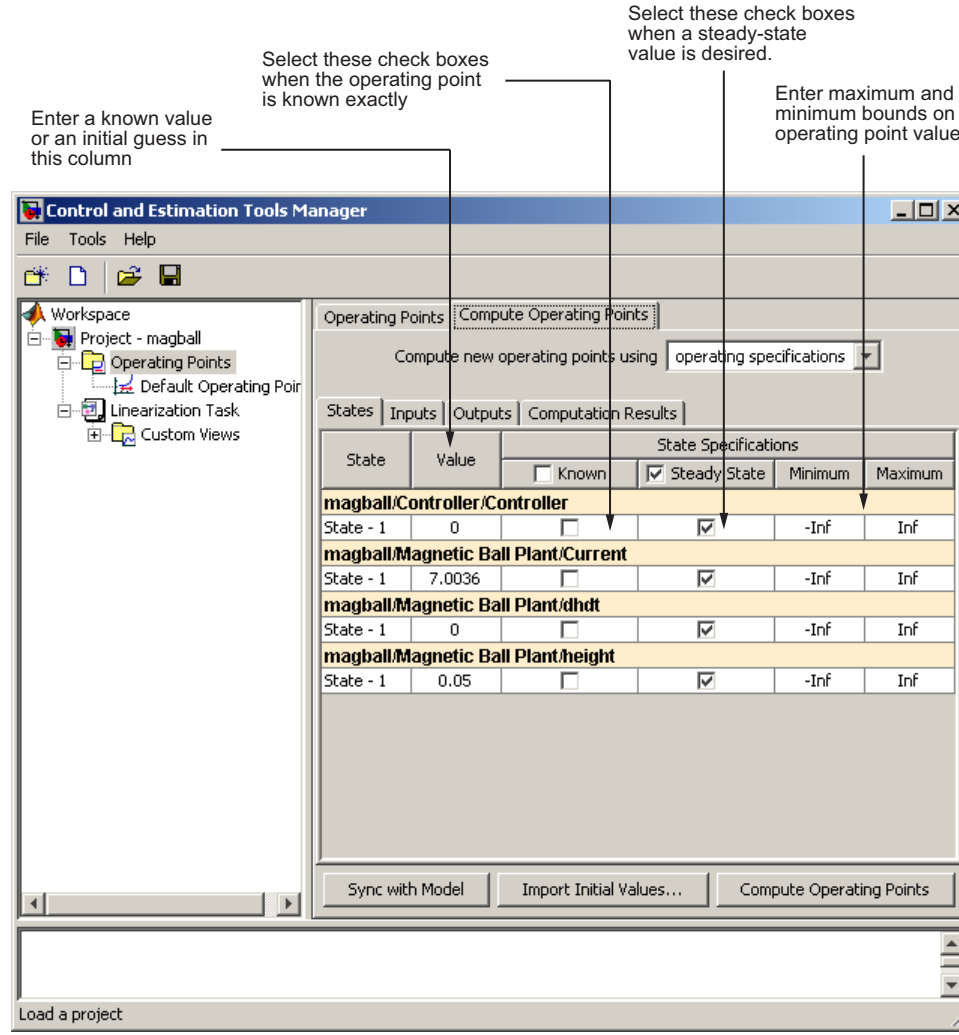
This section continues the magball example from "Beginning a New Linearization Task" on page 7-7. At this stage in the example, a linearization task has already been created for the model.

You compute the operating point from specifications when you only know partial or implicit information. Typical operating point specifications search for steady state or equilibrium operating points.

1 Create a new operating point by either:

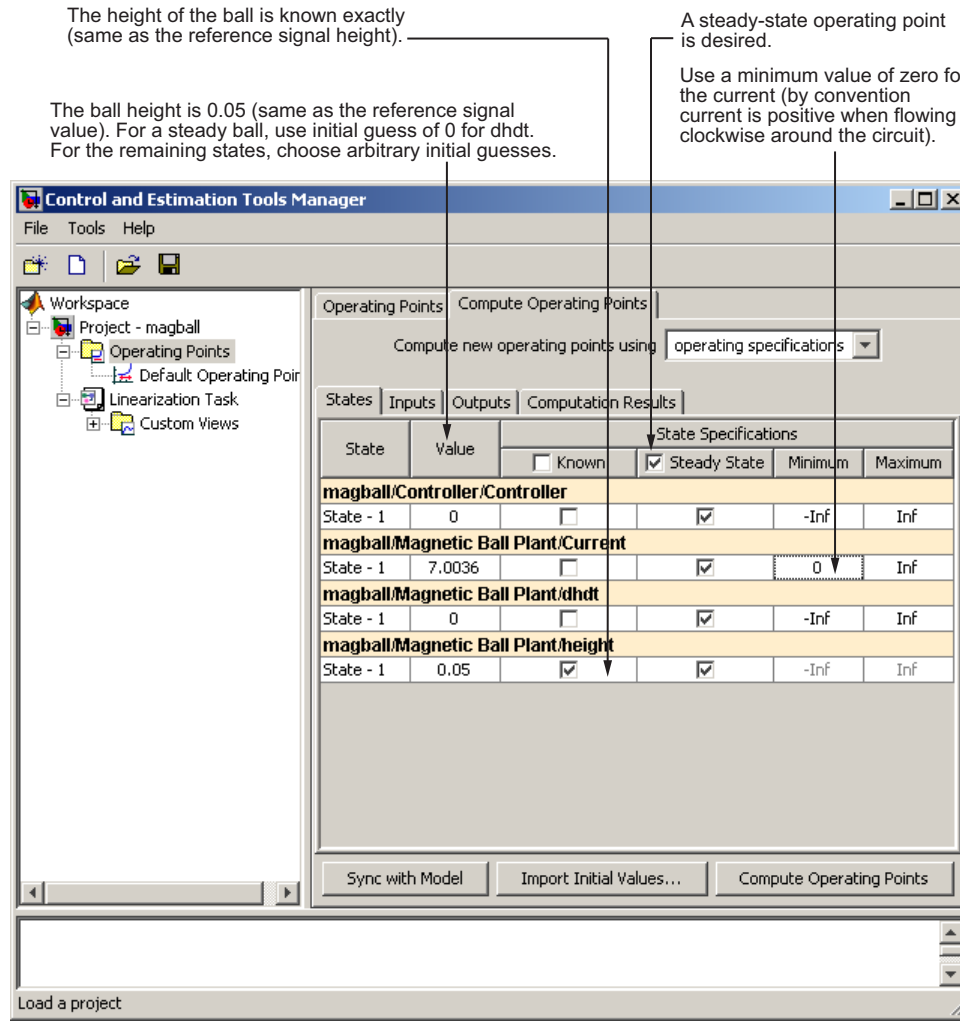
- Selecting the **Operating Points** node and then clicking the **Compute Operating Points** tab
- Clicking the **New Operating Point** button on the **Operating Points** pane of the **Linearization Task** node

2 From the **Compute new operating points using** list, select `operating specifications`. The Control and Estimation Tools Manager window should now resemble the following figure.



3 Enter operating point specifications in the table, such as any known values or constraints on signal values. Switch between states, inputs, and outputs using the tabs on the left.

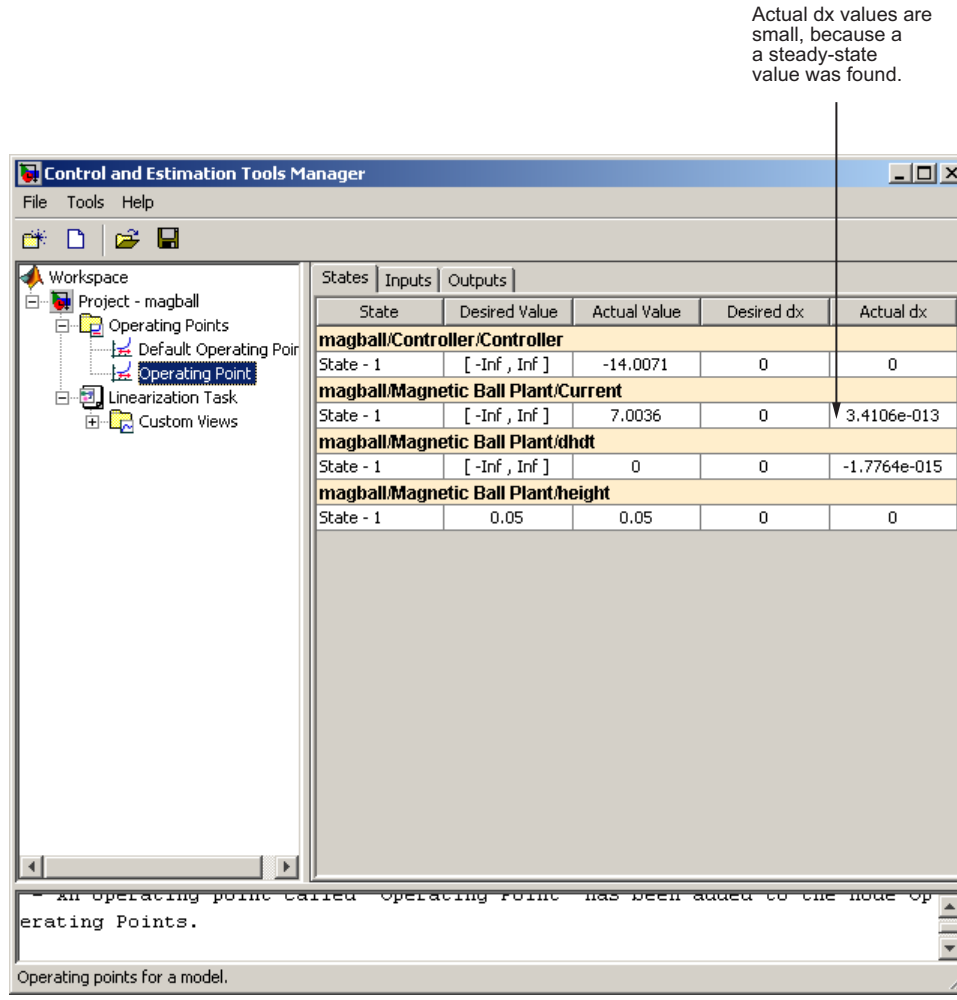
A suitable set of specifications for the `magball` model is shown in the following figure. This model does not contain any root-level input or output ports and, as a result, the **Inputs** and **Outputs** panes are empty. You can still constrain the output signal of any block by adding an **Output Constraint** linearization point to the model. See "Constraining Outputs" in the *Simulink Control Design User's Guide* for information about adding an output constraint to the specifications for an operating point.



When you add states, inputs, or outputs to the model, or remove them from the model, click the **Sync with Model** button to update the operating point table to reflect these changes.

4 Click **Compute Operating Points**. The Simulink Control Design software finds an operating point that closely matches the specifications and adds the new operating point, labeled **Operating Point**, to the **Operating Points** node. Some specifications, even values specified as **Known**, may

not be met exactly. Select the operating point in the project tree to view its contents and assess the results.



For information on options that you can set when finding operating points from partial specifications, see “Changing Optimization Settings” in the *Simulink Control Design User’s Guide*.

Note Inputs and outputs for the operating point are not the same as the linearization input and output points, or analysis I/Os, used to define the inputs and outputs of the linearized model. Instead, operating-point inputs and outputs define the full operating point of a Simulink model, along with any additional, user-defined output constraints.

Creating Operating Points from Known Values

This section continues the magball example from "Creating Operating Points from Specifications" on page 8-9. At this stage in the example, a linearization task has already been created for the model, and a steady state operating point has been computed from specifications.

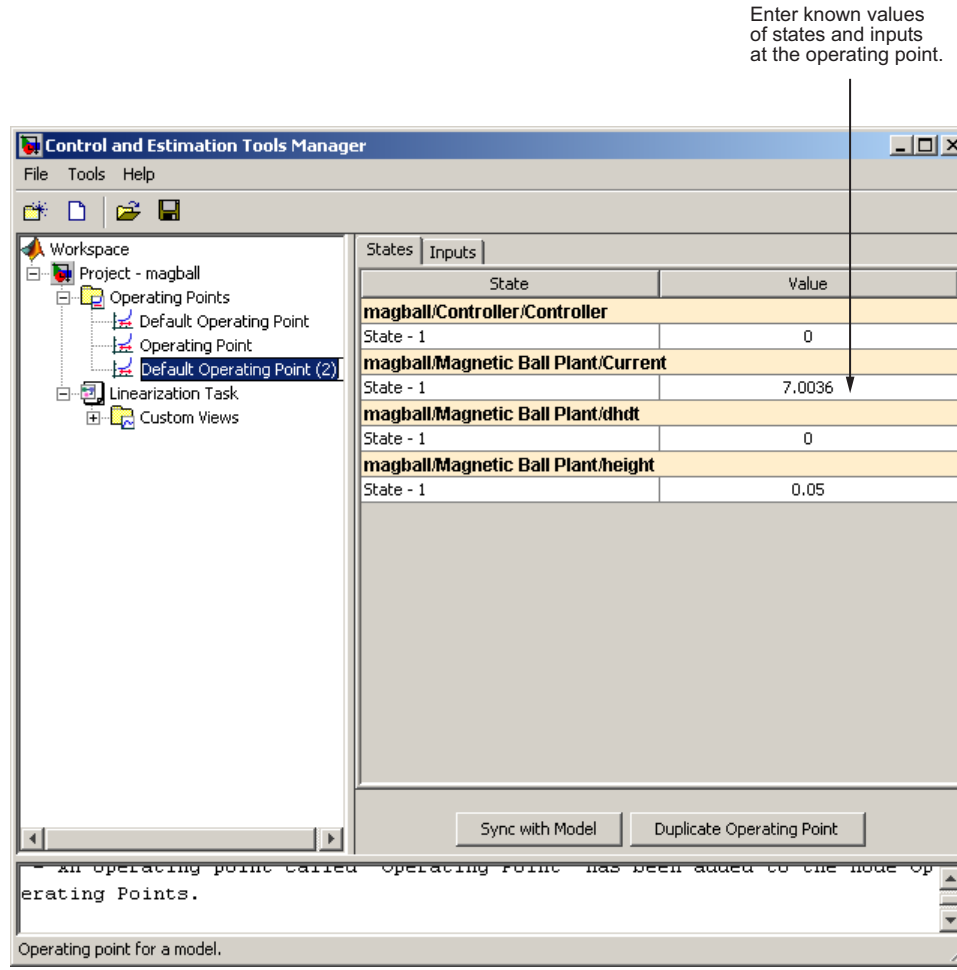
When you know the values of *all* states and inputs at the operating point, you can create a new operating point in the Control and Estimation Tools Manager and manually edit the operating point values:

1. Select the **Operating Points** node in the left pane and then click the **Operating Points** tab in the right pane.

2. Click the **New** button in the bottom-right corner to create a new operating point under the **Operating Points** node. This new operating point is labeled **Default Operating Point (2)**.

3 View the details of **Default Operating Point (2)** (shown in the following figure) by either:

- Selecting it under the **Operating Points** node in the project tree
- Selecting it in the **Operating Points** pane of the **Linearization Task** node, and then clicking **View**



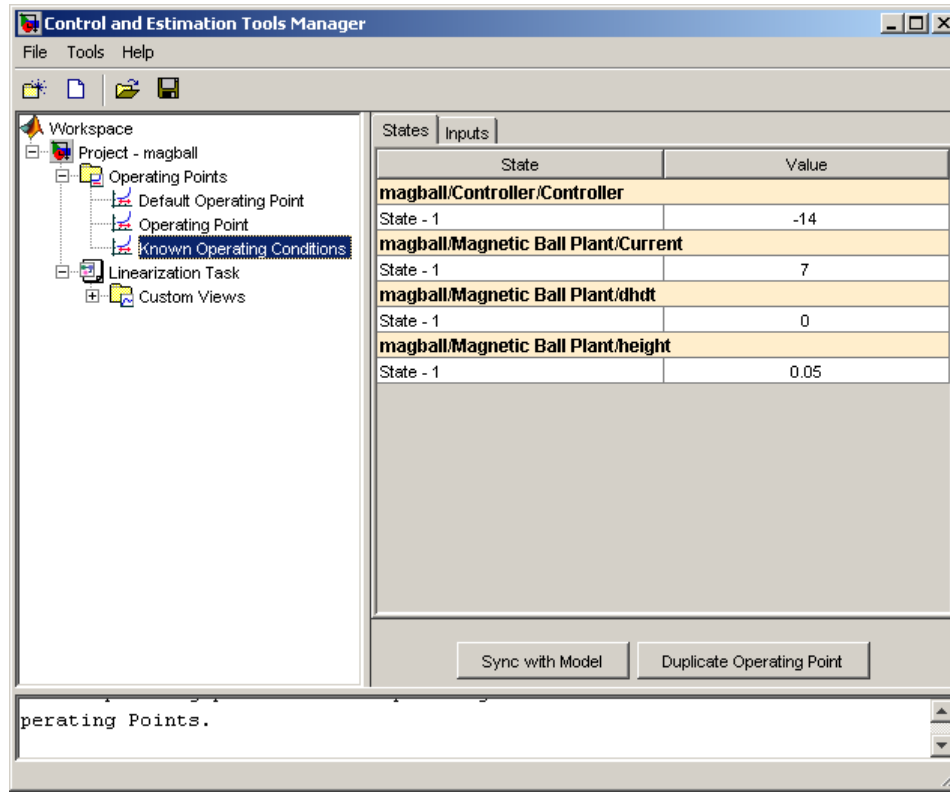
4 Edit the operating point by entering new values in the table. Switch between state and input values using the tabs on the left.

Change the value of **State-1** of **Controller** to `-14` and the value of **Current** to `7`. The model does not contain any root-level input ports, and, as a result, the **Inputs** pane is empty.

When you add states, inputs, or outputs to the model, or remove them from the model, click the **Sync with Model** button to update the operating point table to reflect these changes.

5 Rename the operating point by right-clicking **Default Operating Point (2)** under the **Operating Points** node, selecting **Rename**, and entering a new name in the dialog box.

For example, label this operating condition `Known Operating Conditions`. The pane should now resemble the following figure.



Creating Operating Points from Simulation

You can create an operating point from a simulation of your model at the following simulation points:

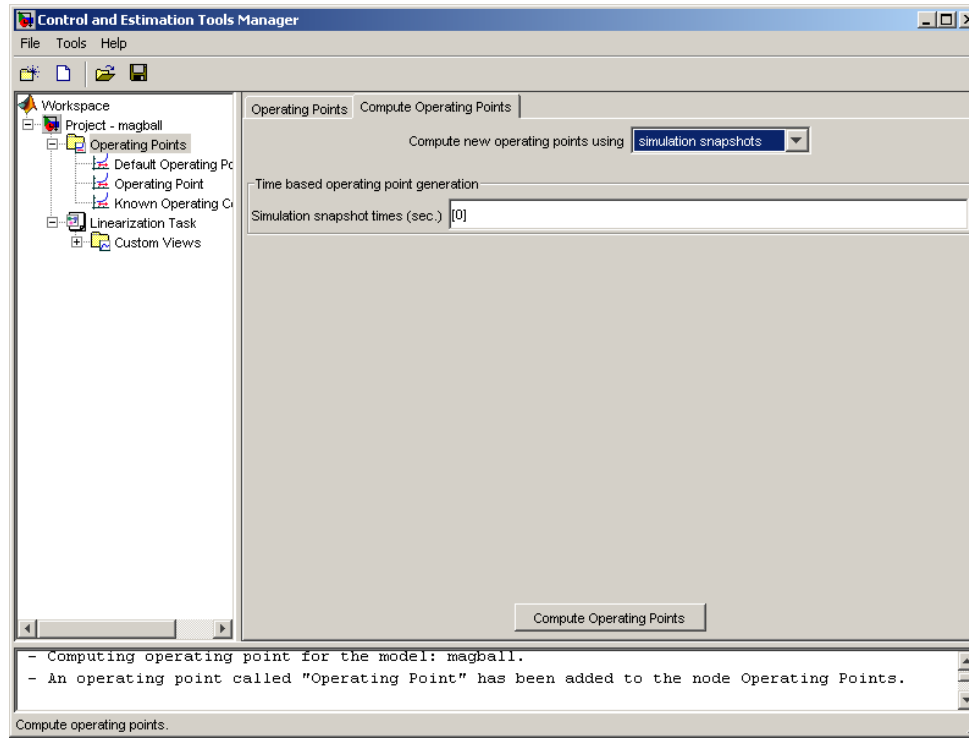
- Specified simulation times, such as when the simulation reaches a steady state solution (see example in this section).
- Events during a specified simulation interval (see "Creating Operating Points at Simulation Events" in the *Simulink Control Design User's Guide*).

This section continues the magball example from "Creating Operating Points from Known Values" on page 8-15. At this stage in the example, a linearization task has already been created for the model, a steady state operating point has been computed from specifications, and a completely known operating point has been specified.

To create operating points at specified simulation times:

1. Open the **Compute Operating Points** tab by either:
   - Selecting the **Operating Points** node in the project tree, and then clicking the **Compute Operating Points** tab
   - Clicking the **New Operating Point** button on the **Operating Points** pane of the **Linearization Task** node

2 From the **Compute new operating points using** list, select `simulation snapshots`. The window should now resemble the following figure.

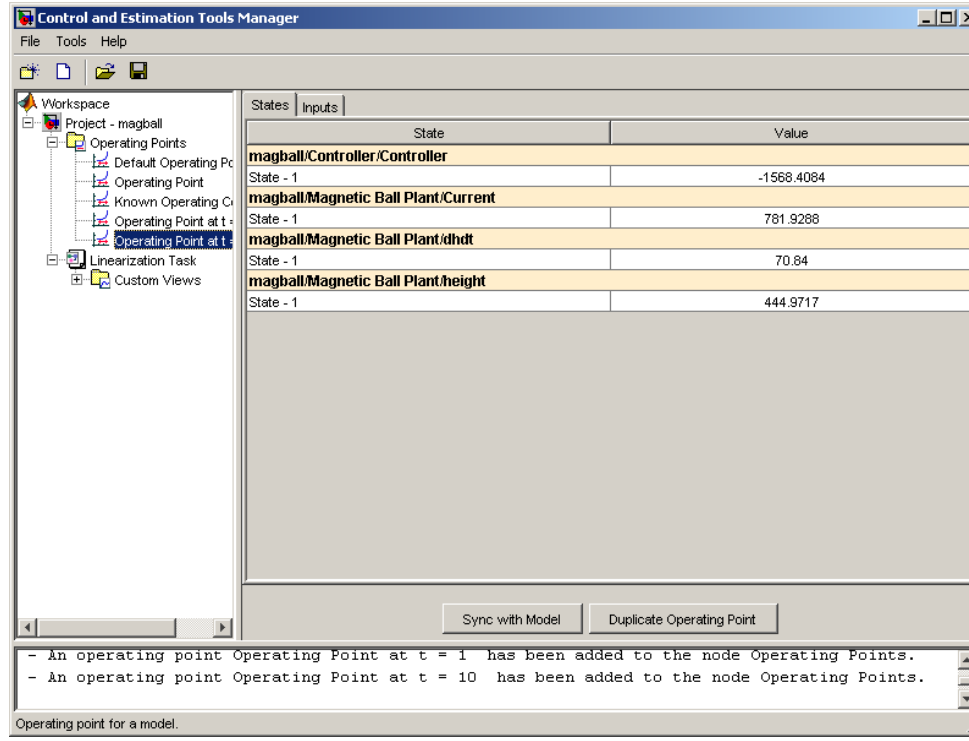


3 Enter a vector of times in the **Simulation snapshot times (sec)** field. Enter `[1,10]` to compute operating points at `t=1` and `t=10`.

4 Click **Compute Operating Points**. The Simulink Control Design software simulates the model, extracts operating points, labeled **Operating Point**

at t=1 and **Operating Point at t=10**, and adds them to the **Operating Points** node in the project tree.

To view the contents of the operating point you created, select the operating point in the project tree as shown in this figure.



Note When you add states, inputs, or outputs to the model, or remove them from the model, click the **Sync with Model** button to update the operating point table to reflect these changes.

Exporting and Saving Operating Points

In this section...

"Uses for Exported and Saved Operating Points" on page 8-21

"Exporting Operating Points" on page 8-21

"Saving Operating Points" on page 8-22

Uses for Exported and Saved Operating Points

You can export and save operating points that you created and then use them in the following ways:

- In a future Simulink Control Design task.
- With command-line functions.
- To initialize a Simulink model for simulation.

Exporting Operating Points

This section continues the magball example. At this stage in the example, several operating points have been created.

In "Creating Operating Points from Specifications" on page 8-9 you computed an operating point from specifications.

To export it to the MATLAB workspace:

1. Under the **Operating Points** node in the Control and Estimation Tools Manager, right-click the operating point labeled **Operating Point** that you computed from specifications. Select **Export to Workspace**.
2. In the Export to Workspace dialog box
   a. Select **Base Workspace** to export the operating point to the MATLAB workspace.
   b. Enter `Operating_Point` for the **Variable Name** parameter.
   c. Click **OK** to export the operating point and close the dialog box.

Saving Operating Points

After you have exported the operating point to the MATLAB workspace, you can save it in a MAT-file for later use. To save the operating point `Operating_Point` in a file named `magball_operating_points.mat`, enter the following command:

```
save magball_operating_point Operating_Point
```

9

Linearizing Models

- "What Is Linearization?" on page 9-2
- "Configuring Inputs and Outputs for the Linearized Model" on page 9-5
- "Linearizing the Model" on page 9-15
- "Viewing Linearization Results" on page 9-18
- "Validating and Troubleshooting Linearization Results" on page 9-28

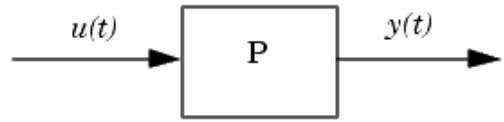
What Is Linearization?

In this section...
"Linearization Background" on page 9-2
"Linearization of Simulink Models" on page 9-3
"Linearization Process Overview" on page 9-3

Linearization Background

A linearized model is an approximation to a nonlinear system, which is valid in a small region around the operating point of the system. Engineers often use linearization in the design and analysis of control systems and physical models.

The following figure shows a visual representation of a nonlinear system as a block diagram. The diagram consists of an external input signal, $u(t)$, a measured output signal, $y(t)$, and the nonlinear system that describes the system's states and its dynamic behavior, P.



You can also express a nonlinear system in terms of the state space equations

$$\dot{x}(t) = f\left(x(t), u(t), t\right)$$
$$y(t) = g\left(x(t), u(t), t\right)$$

where $x(t)$ represents the system's states, $u(t)$ represents the inputs, and $y(t)$ represents the outputs. In these equations, the variables vary

continuously with time. Discrete-time and multi-rate models are discussed in "Linearization of Discrete-Time Models" in the online documentation. A linear time-invariant approximation to this nonlinear system is valid in a region around the operating point at $t=t_0$, $x(t_0)=x_0$, and $u(t_0)=u_0$. If the values of the system's states, $x(t)$ and inputs, $u(t)$ are close enough to the operating point, the system will behave approximately linearly.

Linearization of Simulink Models

Simulink uses a series of connected *blocks* to model physical systems and control systems. Input and output signals connect the blocks, which represent mathematical operations. The nonlinear system, P, in the previous figure, represents a series of connected Simulink blocks.

The Simulink Control Design software linearizes both continuous and discrete-time nonlinear systems by computing the state-space matrices of the linearized model, A, B, C, and D, using one of the linearization algorithms described in "Understanding Analysis in the Simulink Control Design Software" in the *Simulink Control Design User's Guide*.

Linearization Process Overview

The main steps to linearize a model using the Simulink Control Design GUI are as follows:

1. Create or open a model. See "Creating or Opening a Simulink Model" on page 7-2.
2. Create a new linearization task on the Control and Estimation Tools Manager. See "Beginning a New Linearization Task" on page 7-7.
3. Specify an operating point for the model. See Chapter 8, "Specifying Operating Points".
4. Insert linearization input and output points in the model. See "Configuring Inputs and Outputs for the Linearized Model" on page 9-5.
5. Linearize the model. See "Linearizing the Model" on page 9-15.
6. Inspect and validate the linearization. See "Viewing Linearization Results" on page 9-18.

7 Save your project and export the results to the MATLAB Workspace. See Chapter 11, "Saving and Exporting Your Work".

The first three steps in this process were completed in the previous chapters. This chapter continues the magball model example to give a detailed discussion of the remaining steps.

You can also use the Simulink Control Design command-line functions to linearize a model. "Linearizing Models Using Functions" in the *Simulink Control Design User's Guide* gives detailed information on these functions. For a discussion of the advantages and disadvantages of the GUI versus the command-line interface, refer to "Using the GUI Versus Command-Line Functions" on page 1-4.

Configuring Inputs and Outputs for the Linearized Model

In this section...
"What are Linearization Points?" on page 9-5
"Inserting Linearization Points" on page 9-6
"Removing Linearization Points" on page 9-8
"Performing Open-Loop Analysis" on page 9-9
"Inspecting Analysis I/Os" on page 9-12

What are Linearization Points?

Before linearizing the model, you must first configure the model diagram. This involves selecting input and output linearization points (also called analysis I/Os), and then, if necessary, inserting open-loop points for performing open-loop analysis. You can inspect the selected linearization points using the **Analysis I/Os** pane within the **Linearization Task** node of the Control and Estimation Tools Manager.

A linearization input point defines an input to the linearized model while a linearization output point defines an output of the linearized model. Additionally, when the linearized models are computed using numerical perturbation, an input point is the point on the diagram where the small perturbation to the input signal is introduced and an output point is the point on the diagram where the small perturbation to the output signal is measured. Linearization input and output points are not the same as *operating points* which define the state of the model at the point of linearization.

The region between the input and output points defines the portion of the model that you want to linearize, unless a feedback loop feeds the output signal back into another section of the model. In some cases you might want to remove the effect of a feedback signal. For example, you might want to linearize only the plant model within a feedback control loop. When such a feedback loop is present, you can remove the effect of the loop without manually breaking signal lines by inserting an open-loop point. Instructions for inserting open-loop points are in "Performing Open-Loop Analysis" on page 9-9.

You can use the Simulink Control Design software to linearize the whole model, or any blocks or subsystems within the model. To define the system you are linearizing, place linearization points before and after it in the model diagram.

Note Linearization input and output points should not be confused with Simulink Inport and Outport blocks. Input and output points define linear analysis inputs and outputs, while Inport and Outport blocks define the operating point of the system.

Inserting Linearization Points

This section continues the magball example from "Creating Operating Points from Simulation" on page 8-18. At this stage in the example, a linearization task has been created and operating points have been specified.

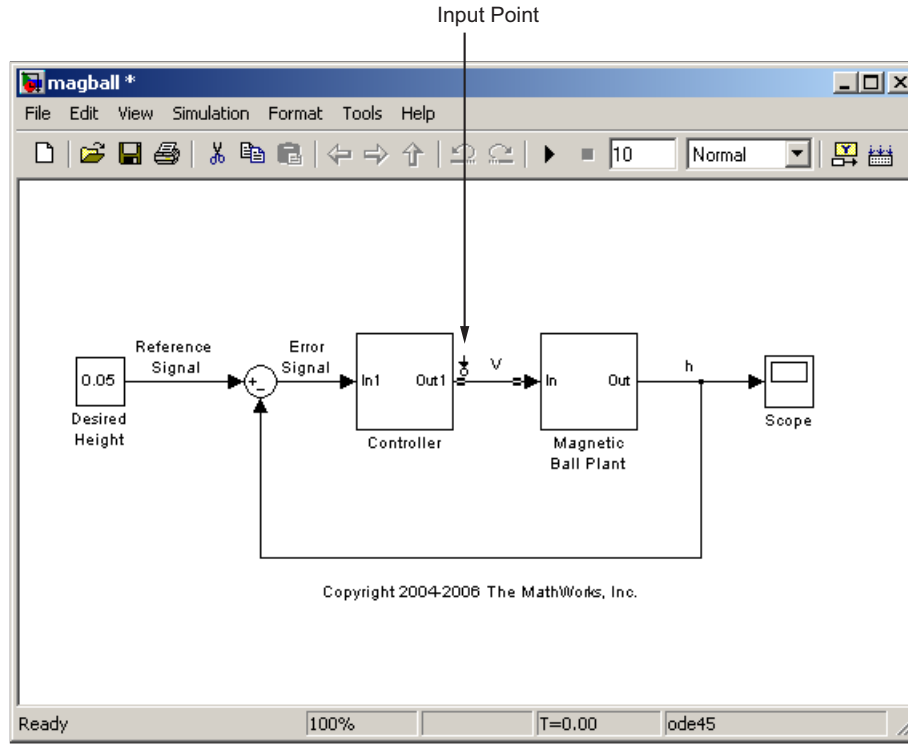
For a definition of linearization points, also known as analysis I/Os, see "What are Linearization Points?" on page 9-5.

In the magnetic ball model, the nonlinearities are all contained within the Magnetic Ball Plant. To linearize this subsystem:

1. Select the **Linearization Task** node within the Control and Estimation Tools Manager.

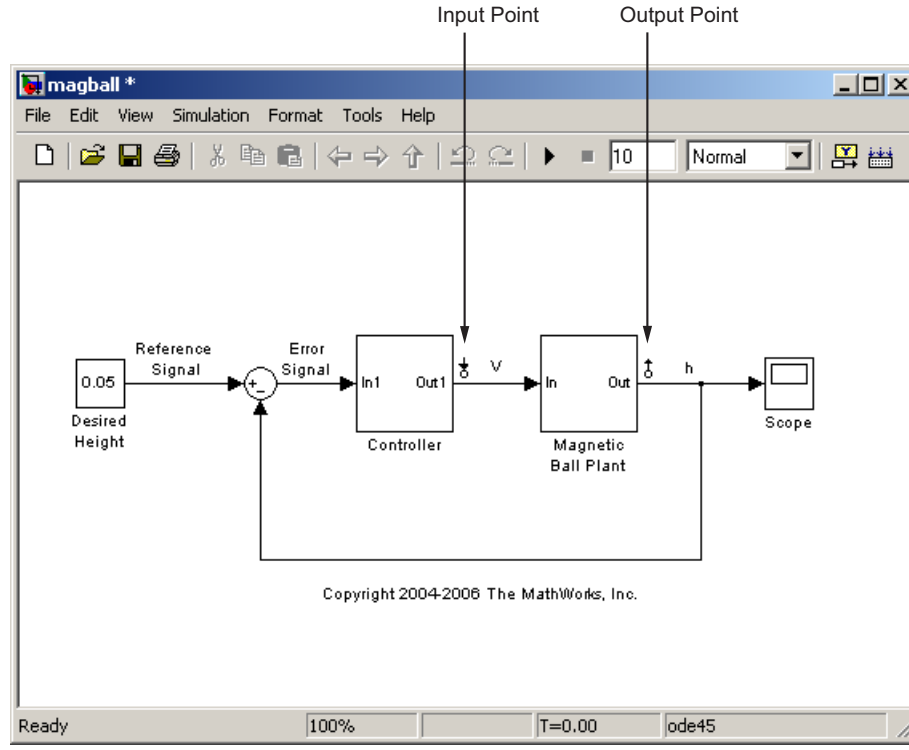
2. On the magball model diagram, position the mouse on the signal line between the Magnetic Ball Plant and the Controller. Right-click and select **Linearization Points > Input Point** from the menu.

   The model diagram now contains a small arrow pointing toward a circle just above the signal line between the Controller and the Magnetic Ball Plant, as in the following figure. This symbol indicates an input point for linearization has been placed there.



3 Right-click the signal line after the Magnetic Ball Plant and select **Linearization Points > Output Point** from the menu.

A small arrow pointing away from a circle on the signal line appears after the Magnetic Ball Plant indicating an output point for linearization has been placed there. The diagram should now look like that in the following figure.



Note All linearization points are referenced to the output port of the block the signal line originates from. For example, placing a linearization point anywhere on the feedback loop in the figure above will result in a linearization point at the output of the Magnetic Ball Plant block.

To inspect the linearization points in the Control and Estimation Tools Manager, see “Inspecting Analysis I/Os” on page 9-12.

Removing Linearization Points

To remove a linearization point from a signal line in your model, repeat the same process as for inserting a linearization point. For example, to remove an input point, right-click the signal line containing the input point and select

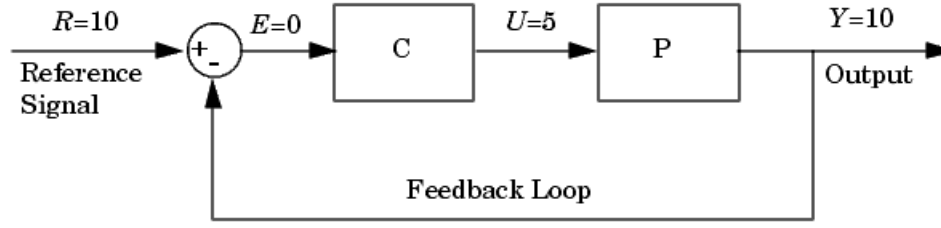
Linearization Points > Input Point from the menu. The input point disappears from the diagram.

Performing Open-Loop Analysis

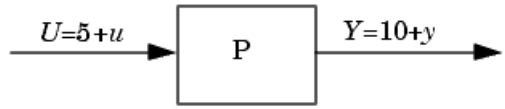
Due to the presence of feedback loops in a model, the input and output points might not completely define the portion of the model you want to linearize. In these cases, to remove the effect of signals feeding back into the portion of the model you are linearizing, you might choose to insert an open-loop point. "What Is Open-Loop Analysis?" on page 9-9 discusses the concepts behind open-loop analysis. "Inserting Loop Openings" on page 9-11 continues the magball example by inserting open-loop linearization points in the magball model. "Open-Loop Analysis Using Functions" in the online documentation gives methods for assigning open-loop points in Simulink models using functions.

What Is Open-Loop Analysis?

Many control systems contain feedback loops. An example of such a system is shown in this figure.

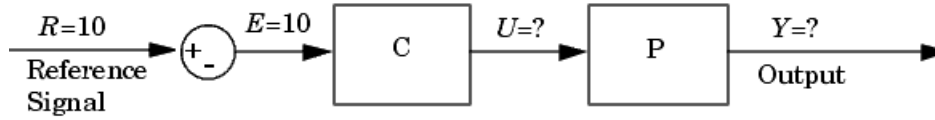


The model represented in this figure is at equilibrium. Consider linearizing the plant, P, about this equilibrium operating point by changing the input signal, U, by a small amount, u, and measuring the change in the output signal, y. The portion of the system that you want to linearize is shown in the following figure.



However, due to the presence of the feedback loop, the change in the output signal will feed back into the controller, C, and then into the plant. This affects the behavior of the system you are linearizing. In fact, if C and P were linear, the linearized model between U and Y would be $\frac{P(s)}{1+C(s)P(s)}$ rather than $P(s)$.

You could manually remove the feedback signal from the model in an attempt to resolve this issue. However, as shown in the following figure, this changes the operating point of the system since the error signal, E, is now equal to the reference signal, R. Linearizing about this new operating point would change the linearization results. Of course, this only makes a difference for nonlinear models. When the model is already linear, it has the same form regardless of the operating point.

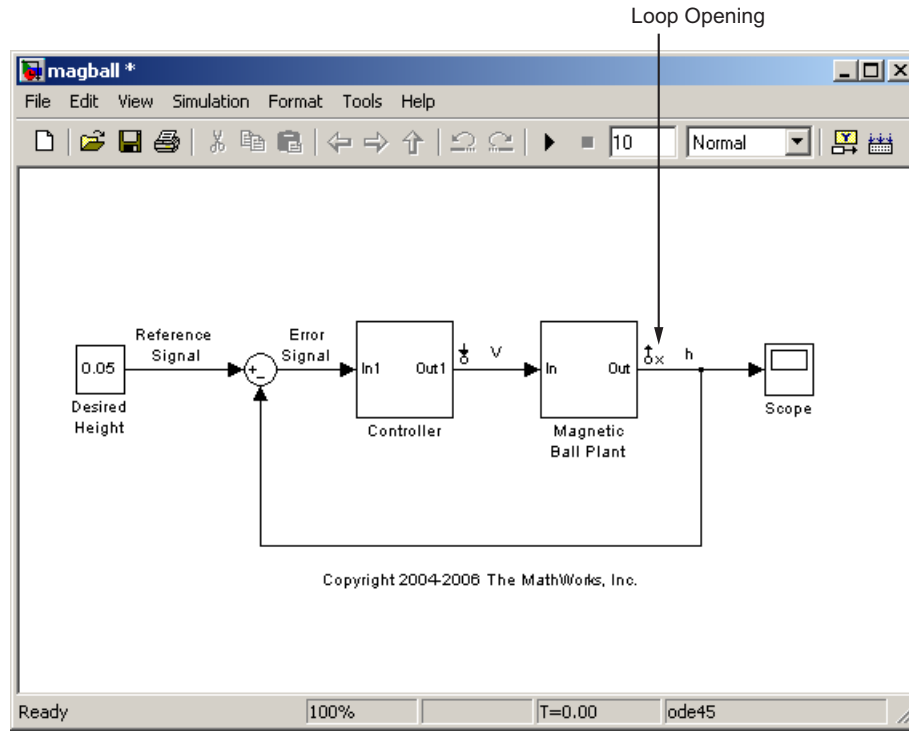


When linearizing Simulink models, label an input or output point as open loop. Doing so ensures that the output signal is not fed back into the model but keeps the operating point the same. In other words, in the linear case, you would compute $P(s)$ rather than $\frac{P(s)}{1+C(s)P(s)}$.

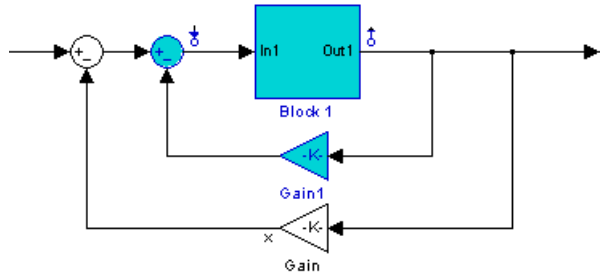
Inserting Loop Openings

This section continues the example from "Inserting Linearization Points" on page 9-6. At this stage in the example, a linearization task has already been created for the model, operating points have been specified, and linearization input and output points have been inserted.

To linearize only the Magnetic Ball Plant, right-click the signal line containing the output point and select **Linearization Points > Open Loop**. This inserts a small x next to the output point in the diagram, representing a loop opening.



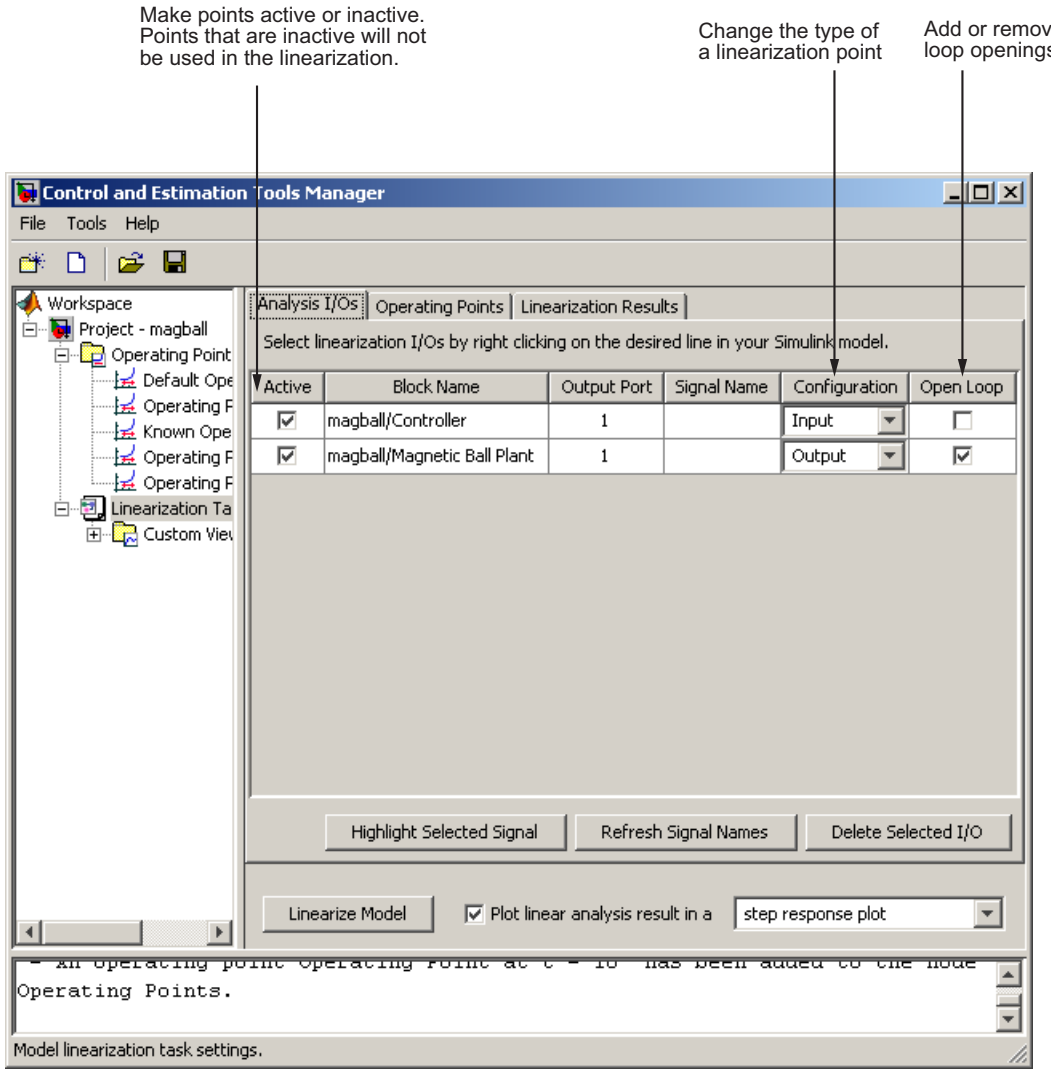
Note You do not need to place a loop opening in the same place as an input or output point. For example, in the following figure the highlighted blocks are included in the linearization. The loop opening is placed after the gain on the outer feedback loop, which removes the effect of this loop from the linearization. Placing a loop opening at the same place as the output point would have removed the effect of the inner loop from the linearization as well.



Inspecting Analysis I/Os

This section continues the example from "Inserting Loop Openings" on page 9-11. At this stage in the example, a linearization task has already been created for the model, operating points have been specified, linearization input and output points have been inserted, and loop openings have been created.

To view the linearization points, click the **Analysis I/Os** tab in the **Linearization Task** node in the Control and Estimation Tools Manager, as shown in the following figure. Use this pane to inspect and make changes to your linearization points.


Make points active or inactive. Points that are inactive will not be used in the linearization.
Change the type of a linearization point
Add or remove loop openings
Control and Estimation Tools Manager
File Tools Help
Workspace
Project - magball
Operating Point
Default Ope
Operating F
Known Ope
Operating F
Operating F
Linearization Ta
Custom Viev
Analysis I/Os
Operating Points
Linearization Results
Select linearization I/Os by right clicking on the desired line in your Simulink model.
Active
Block Name
Output Port
Signal Name
Configuration
Open Loop
magball/Controller
1
Input
magball/Magnetic Ball Plant
1
Output
Highlight Selected Signal
Refresh Signal Names
Delete Selected I/O
Linearize Model
Plot linear analysis result in a
step response plot
Operating Points.
Model linearization task settings.

Note You cannot make changes to the **Analysis I/Os** or perform open-loop analysis when you select `Numerical Perturbation` as the **Linearization Algorithm** in the Linearization Task Options window (accessed by selecting **Tools > Options** from the Control and Estimation Tools Manager window). See "Changing Linearization Settings and Algorithms" in the online *Simulink Control Design User's Guide* for information.

Linearizing the Model

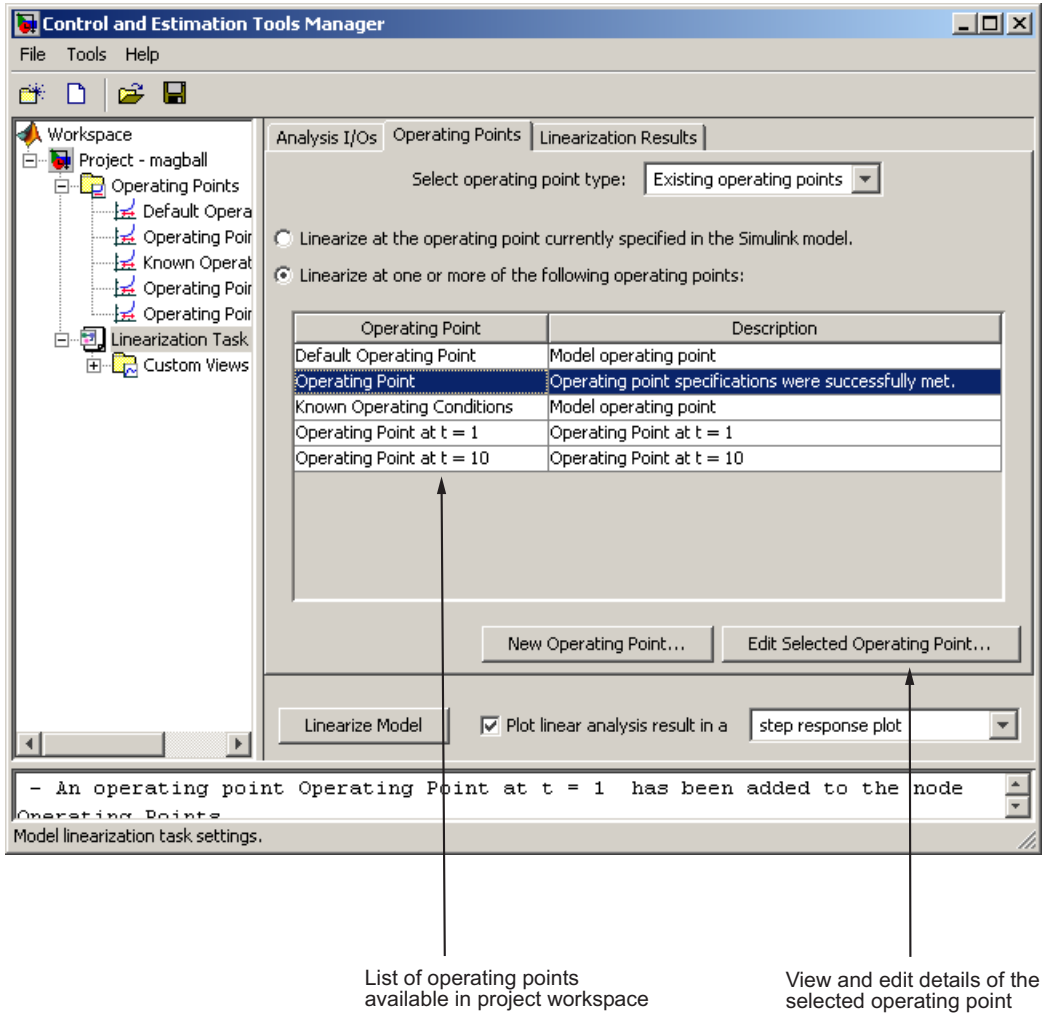
In this section...
"Performing Linearization" on page 9-15
"Using the LTI Viewer" on page 9-17

Performing Linearization

This section continues the example from "Inspecting Analysis I/Os" on page 9-12. At this stage in the example, a linearization task has been created, operating points have been specified, and linearization points have been inserted.

In this step of the linearization task, you linearize the model using one of the operating points the you created in the project workspace. For information about additional linearization option, see "Linearizing Models in the Control and Estimation Tools Manager" in the *Simulink Control Design User's Guide*.

Click the **Operating Points** tab in the **Linearization Task** node to select one or more operating points for the linearization. To linearize the magnetic ball model around the operating point computed from partial specifications, first select the **Linearize at one or more of the following operating points** option button. Then, select **Operating Point** in the list, as shown in the following figure.

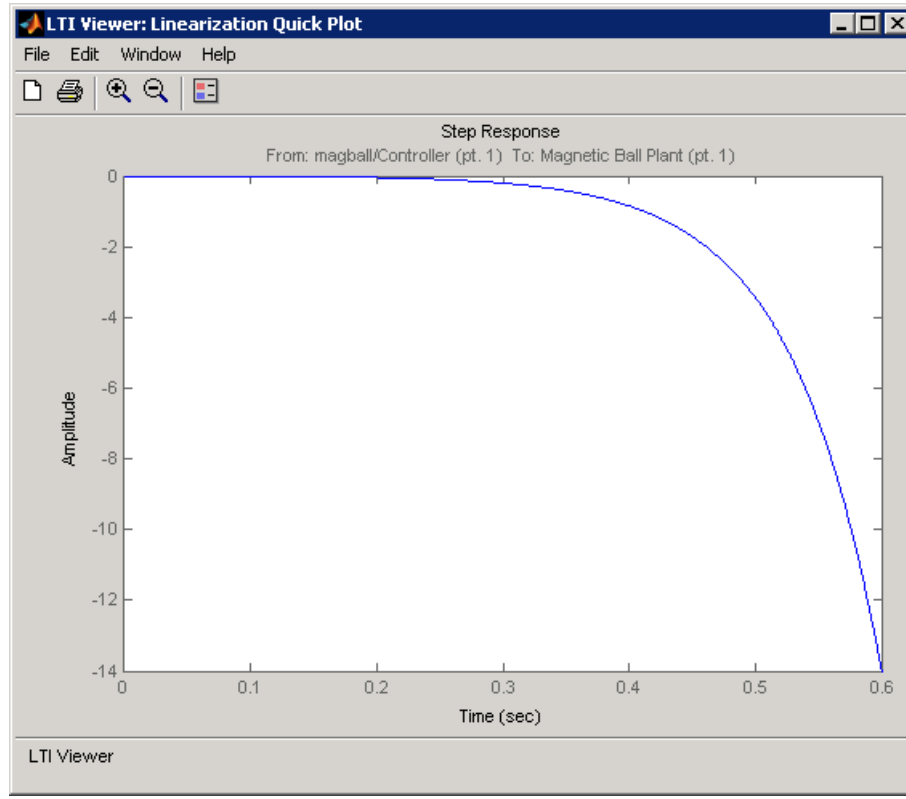


To compute the linearized model:

1. In the menu to the right of the **Linearize Model** button, select from one of the nine options for displaying the results of the linearization. Clear the check box next to this menu when you do not want to display the results.

2 Click the **Linearize Model** button at the bottom of the pane. The linearized model is computed and displayed in the LTI Viewer.

The following figure shows the step response for the linearized magball model.



Using the LTI Viewer

To add characteristics such as settling time or peak response to your plot, right-click anywhere in the plot area and select an option from the menu. Add data markers by clicking the point you want to mark. In addition, you can display up to six plots at one time. To change the number of plots, select **Edit > Plot Configurations**. Export the linearized model to the workspace by selecting **File > Export**. For more information, see "LTI Viewer" in the *Control System Toolbox Getting Started Guide*.

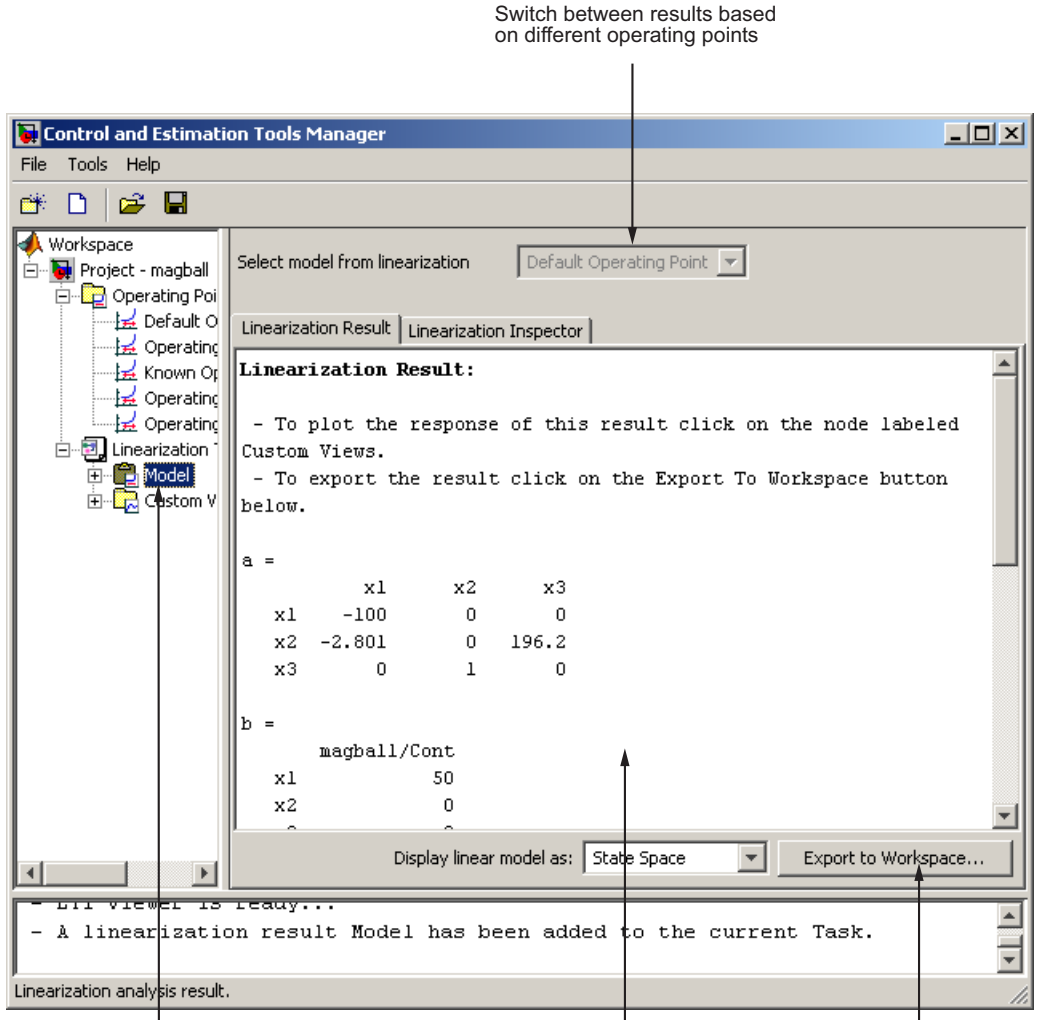
Viewing Linearization Results

In this section...
"Displaying Linearization Results" on page 9-18
"Highlighting Blocks in the Linearization" on page 9-20
"Inspecting the Linearization Results Block by Block" on page 9-21
"Comparing the Results of Multiple Linearizations" on page 9-23

Displaying Linearization Results

This section continues the magball example from "Linearizing the Model" on page 9-15. At this stage in the example, a linearization task has been created, operating points have been specified, linearization points have been inserted, and a linearized model has been computed.

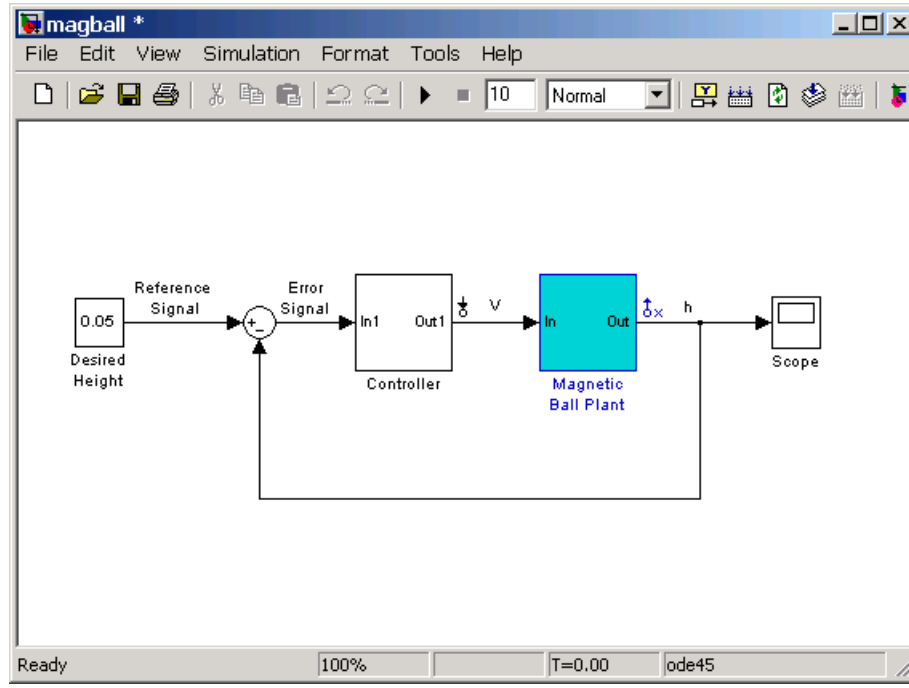
To display the **Linearization Result** pane, as shown in the following figure, select the **Model** node in the project tree. To delete the linearization result, right-click **Model** and select **Delete** from the menu. The **Linearization Result** pane displays a mathematical representation of the linearized model. By default it appears as a state space system and it displays the state space matrices. However, within the pane you can choose to view the model as a zero-pole-gain system or as a transfer function. You can also export the model to the MATLAB workspace.


Switch between results based on different operating points
Control and Estimation Tools Manager
File Tools Help
Workspace
Project - magball
Operating Poi
Default O
Operating
Known Op
Operating
Operating
Linearization
Model
Custom V
Select model from linearization
Default Operating Point
Linearization Result
Linearization Inspector
Linearization Result:
- To plot the response of this result click on the node labeled Custom Views.
- To export the result click on the Export To Workspace button below.
a =
x1 x2 x3
x1 -100 0 0
x2 -2.801 0 196.2
x3 0 1 0
b =
magball/Cont
x1 50
x2 0
Display linear model as:
State Space
Export to Workspace...
- A linearization result Model has been added to the current Task.
Linearization analysis result.
View the operating points used in the analysis by clicking the + to the left of the Model node
Linearization Result gives the state-space matrices of the linearized model
Export the linearized model to the workspace

Highlighting Blocks in the Linearization

This section continues the magball example from "Displaying Linearization Results" on page 9-18. At this stage in the example, a linearization task has been created, operating points have been created, linearization points have been inserted, and a linearized model has been computed.

To verify that the correct portion of the model was linearized, you can highlight the blocks in the model that were used in the linearization. To do this, right-click the **Model** node and select **Highlight Blocks in Linearization**. The following figure shows that the Magnetic Ball Plant block, including all blocks within it, was used in the linearization.



To remove the highlighting, right-click the **Model** node and select **Remove Highlighting**.

Note You cannot use the **Highlight Blocks in Linearization** or **Remove Highlighting** option for models you linearize with numerical-perturbation linearization (when `Numerical perturbation` is selected as the **Linearization Algorithm** parameter in the Linearization Task Options window).

Inspecting the Linearization Results Block by Block

This section continues the example from "Displaying Linearization Results" on page 9-18. At this stage in the example, a linearization task has been created, operating points have been created, linearization points have been inserted, and a linearized model has been computed.

To verify that blocks of interest linearized as expected, you can inspect linearization results block by block using the Simulink Control Design Linearization Inspector. An example of a block of interest is a block with a sharp discontinuity.

Note For information on troubleshooting blocks that did not linearize as expected, see "Troubleshooting Linearization Results".

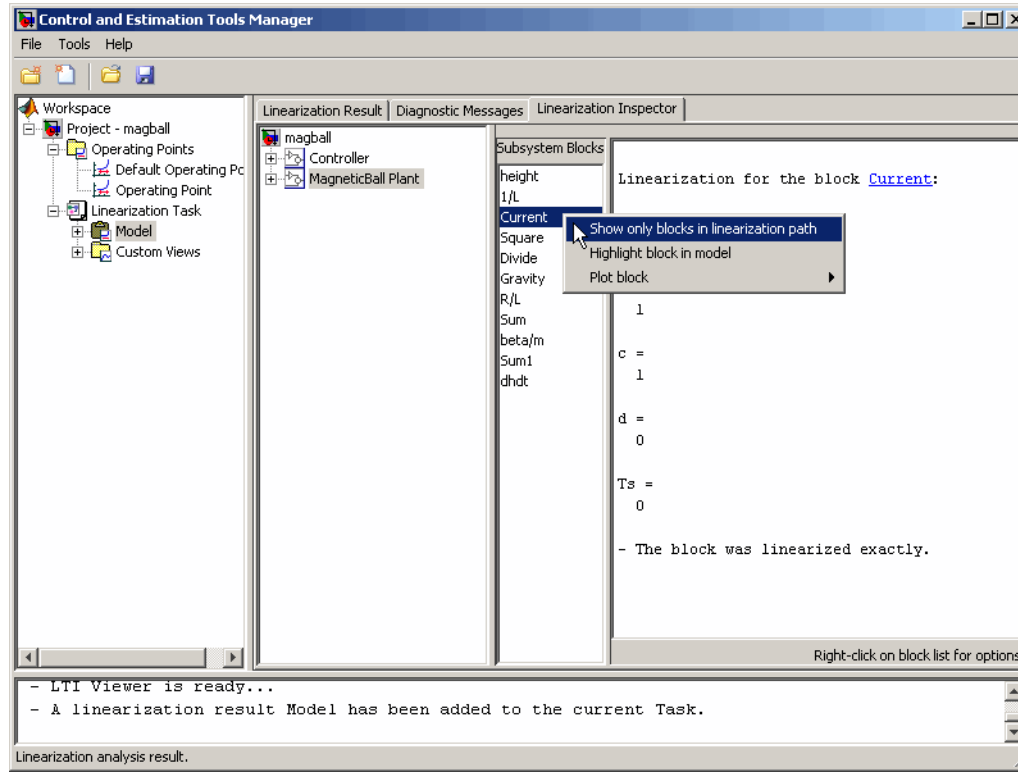
To inspect the linearization results for the *Current* block in the Magnetic Ball Plant:

1. Select the **Model** node. Then, select the **Linearization Inspector** tab.
2. In the list of blocks under **magball**, select the **MagneticBall Plant** subsystem.
3. In the **Subsystems Blocks** column, select the **Current** block.

Tip You can filter the Subsystem Blocks list to include only blocks in the linearization path (the blocks between the input and output points). To filter the list, right-click the list, and select **Show only blocks in linearization path**.

This action displays the following in the Linearization Inspector pane:

- State-space matrices *A*, *B*, *C*, and *D* for the linearized *Current* block.
- Sample time Ts, which equals zero and shows that the Magnetic Ball Plant is continuous.
- A note saying that the block was linearized exactly.



You can plot the linearization results for a block. To create a step plot of the *Current* block linearization results, right-click the **Current** block in the **Subsystems Blocks** column and select **Plot block > Step Plot**.

Note The **Linearization Inspector** pane is not available for models that are linearized with numerical-perturbation linearization (when `Numerical perturbation` is selected as the **Linearization Algorithm** parameter in the Linearization Task Options window). Such linearizations have no individual block linearizations to inspect.

Disabling the Linearization Inspector

When you do not want diagnostic information for your linearization, you disable the linearization inspector.

To disable the linearization inspector:

1. Select **Tools > Options** in the Control and Estimation Tools Manager window. Then, click the **Linearization** tab.
2. Deselect the **Store Linearization Diagnostics and Inspector data with linearization** option.

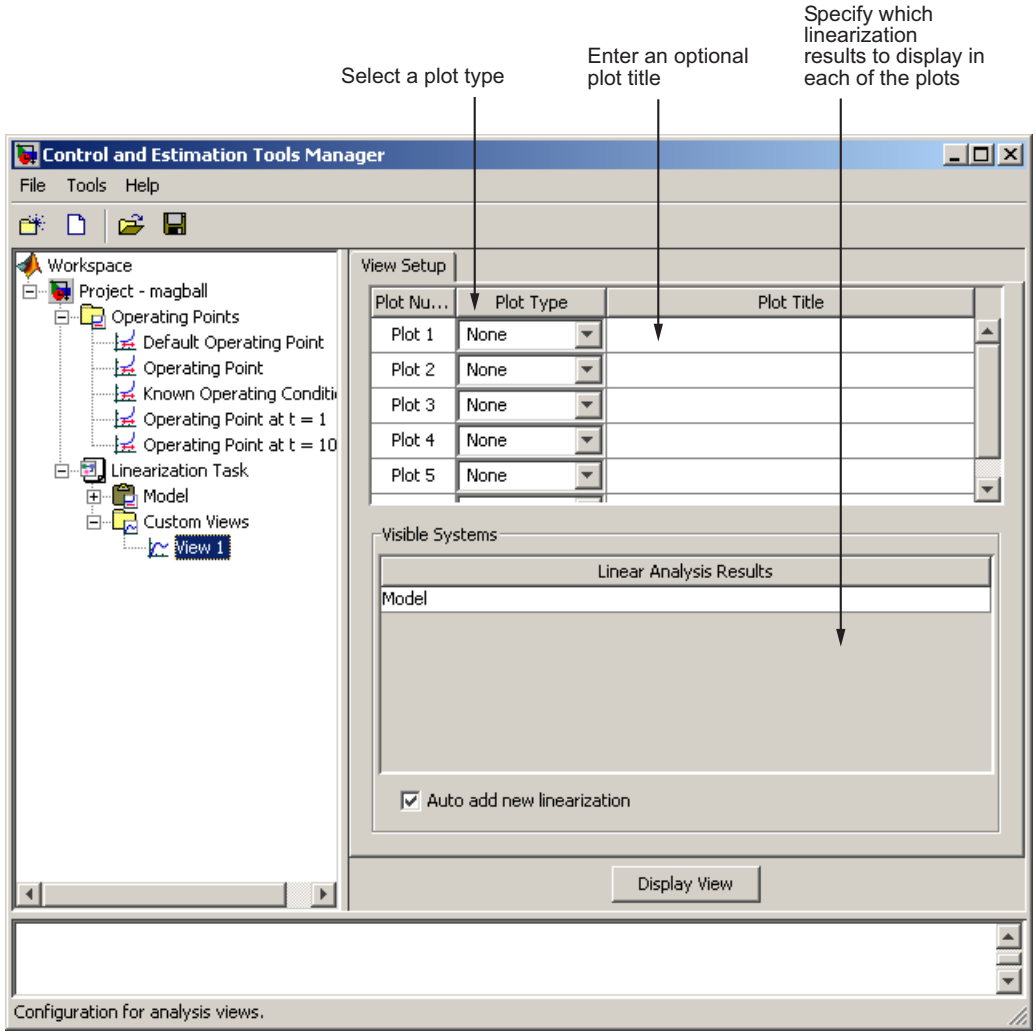
Note You can also disable the linearization inspector globally in the Simulink Control Design tab of the MATLAB preferences dialog box. When you set this global preference, it takes effect in each subsequent creation of linearization tasks and persists from session to session.

For more information on changing linearization options, see "Changing Linearization Options".

Comparing the Results of Multiple Linearizations

This section continues the magball example from "Inspecting the Linearization Results Block by Block" on page 9-21. At this stage in the example, a linearization task has been created, operating points have been created, linearization points have been inserted, and a linearized model has been computed and inspected.

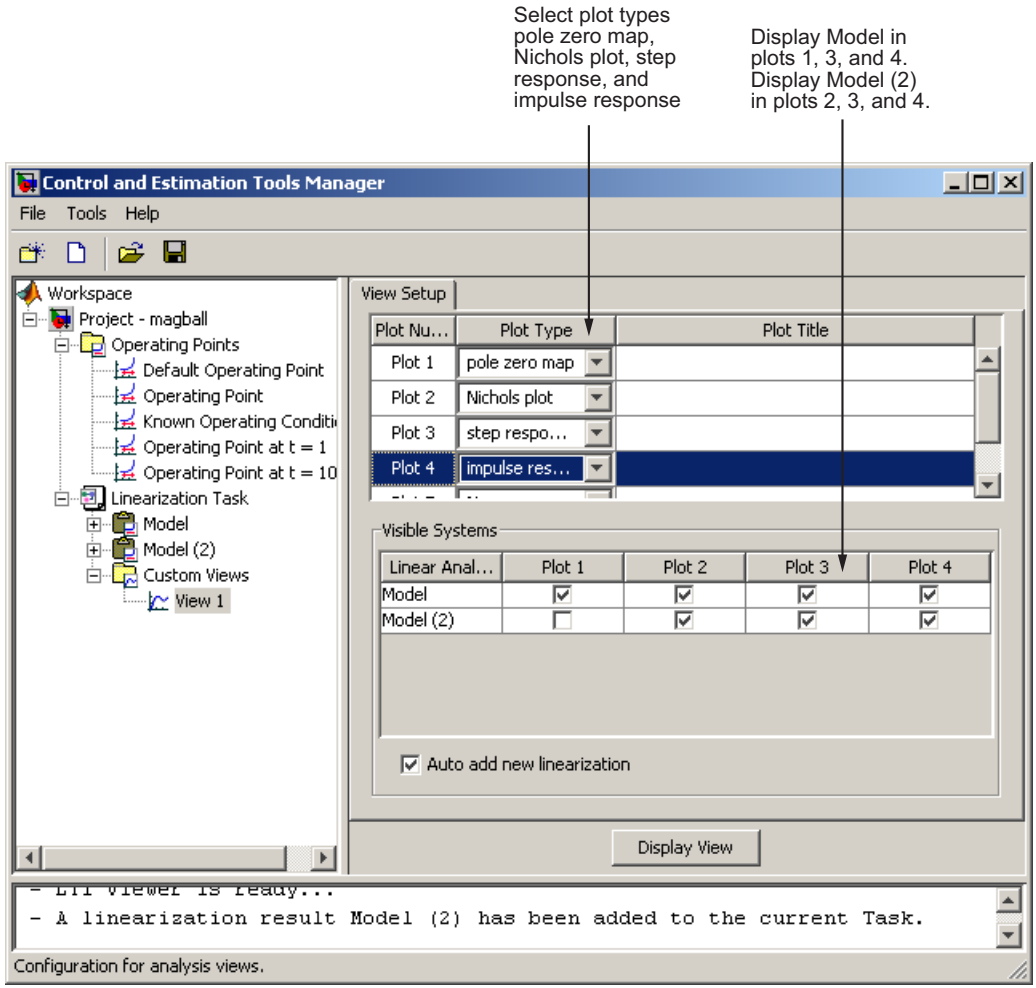
The LTI Viewer automatically displays a **Linearization Quick Plot** of the linearization results after the linearization. To create additional, customized plots, especially for comparing multiple linearizations, right-click the **Custom Views** node under the **Linearization Task** node and select **Add View**. A new view, **View1**, is added to the **Custom Views** node. Select the **View1** node to display the **View Setup** pane and set up this view.



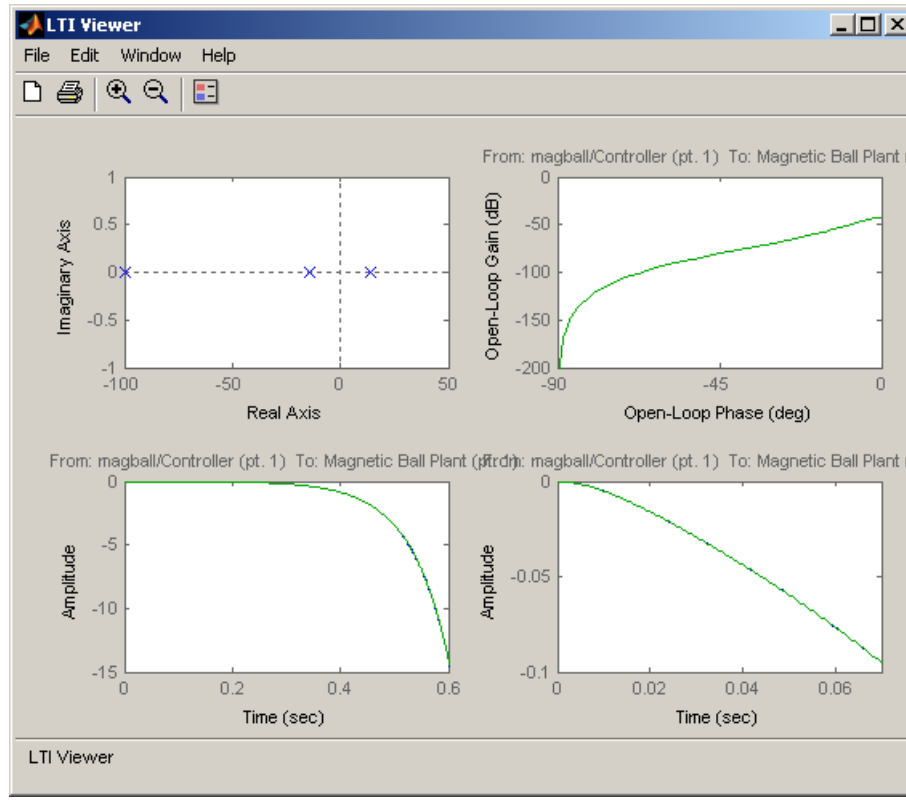
Example

Before configuring the **View Setup** pane for the magnetic ball system, run another linearization of this model using the operating point labeled **Known Operating Conditions**, that you computed in "Creating Operating Points from Known Values" on page 8-15. This adds **Model (2)** to the **Linearization Task** node.

Configure **View Setup** as shown in the following figure:



Click **Display View** to display the LTI Viewer, as shown in the following figure:



Before proceeding to Chapter 10, “Designing Compensators”, close the Control and Estimation Tools Manager and the magball model. You do not need to save any projects or any changes to the model.

Validating and Troubleshooting Linearization Results

For information about validating linearization results by comparing simulations of the original and linearized models, see “Validating Linearization Results”.

For information about troubleshooting unexpected linearization results, see “Troubleshooting Linearization Results”.

10

Designing Compensators

- "What Is Compensator Design?" on page 10-2
- "Beginning a Compensator Design Task" on page 10-3
- "Picking Blocks to Tune" on page 10-5
- "Selecting Closed-Loop Responses to Design" on page 10-7
- "Selecting an Operating Point" on page 10-10
- "Creating a SISO Design Task" on page 10-13
- "Completing the Design" on page 10-22

What Is Compensator Design?

Compensator design is the process of designing compensators for a control system so that the system behaves in a desired way. Compensators in Simulink models are represented by blocks such as Transfer function, Zero-Pole-Gain, and PID blocks. These blocks can act as feedback controllers, pre-filters, feedforward controllers, sensors, etc. Compensator design for Simulink models can be as simple as adjusting gains in a one of these blocks, or as complicated as adding, deleting, or moving poles and zeros in multiple compensators over multiple feedback loops.

Compensator design methodologies often make use of tools such as Bode diagrams, root-locus diagrams, or response plots. These tools require that the plant model is linear; however, most real-world systems are nonlinear. The Simulink Control Design software simplifies the task of designing compensators for Simulink models by automatically linearizing the model before creating a SISO Design Task in which you can edit and design the compensators using a variety of tools.

Compensator design in the Control and Estimation Tools Manager involves the following steps:

1 “Picking Blocks to Tune” on page 10-5

2 “Selecting Closed-Loop Responses to Design” on page 10-7

3 “Selecting an Operating Point” on page 10-10

4 “Creating a SISO Design Task” on page 10-13

5 “Completing the Design” on page 10-22

Beginning a Compensator Design Task

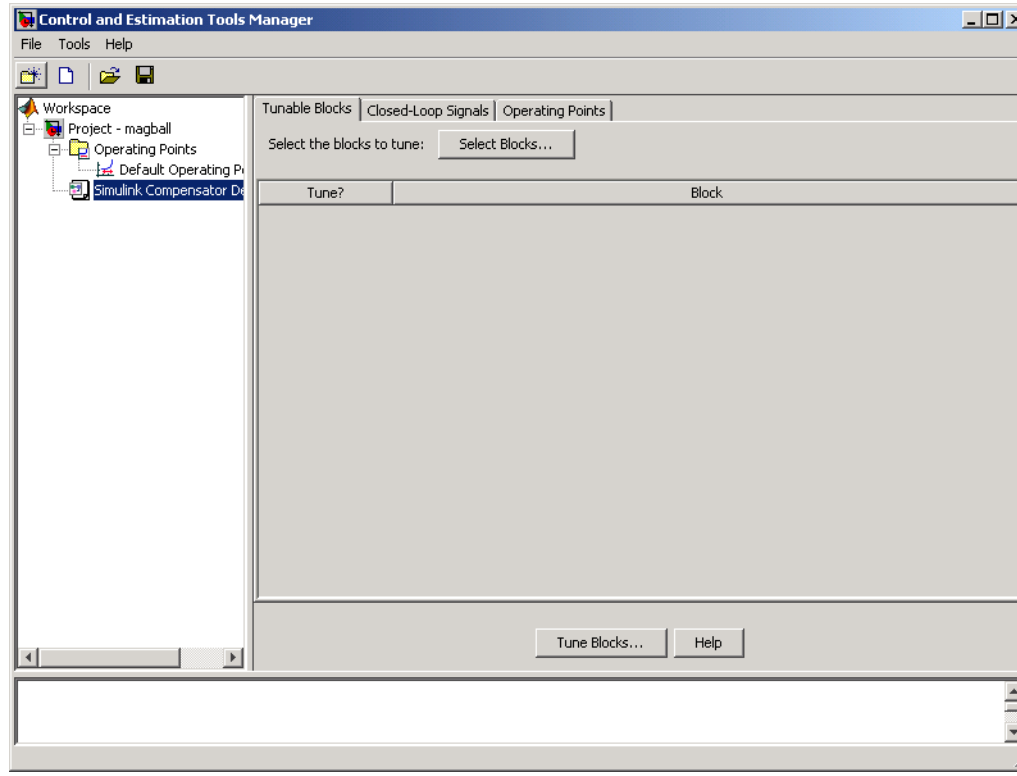
This chapter continues the magball example from “Example Model: The Magnetic Ball System” on page 7-2.

Before you begin this compensator design example, close the Control and Estimation Tools Manager and the magball model, if you have them open, to make sure you are working with a fresh version of the magball model. You do not need to save any projects or any changes to the model.

To begin a new compensator design task for the magball model:

1. Enter `magball` at the MATLAB command line to open the magball model.
2. Select **Tools > Control Design > Compensator Design** from the `magball` window.

   The Control and Estimation Tools Manager opens and creates a new compensator design task, as shown in the following figure.



The project tree in the left pane of the Control and Estimation Tools Manager now shows a **Simulink Compensator Design Task** node as part of **Project - magball** in addition to the **Operating Points** node. You can select a node within the tree to display its contents in the right pane.

- For information on the **Tunable Blocks** pane within the **Simulink Compensator Design Task** node, refer to “Picking Blocks to Tune” on page 10-5.
- For information on the **Closed-Loop Signals** pane within the **Simulink Compensator Design Task** node, refer to “Selecting Closed-Loop Responses to Design” on page 10-7.
- For information on the **Operating Points** node or the **Operating Points** pane within the **Simulink Compensator Design Task** node, refer to “Selecting an Operating Point” on page 10-10.

Picking Blocks to Tune

This section continues the magball example from "Beginning a Compensator Design Task" on page 10-3. At this stage in the example, a compensator design task has been created.

In this step of the compensator design task, you select the tunable blocks in your model. The tunable blocks are those that you tune in the SISO Design Tool to achieve the desired response of your system. They are typically the compensators in the system.

Blocks that are tunable in a Simulink Control Design Task include

- Gain
- PID
- LTI
- State-space (both continuous and discrete)
- Zero-Pole-Gain (both continuous and discrete)
- Transfer function (both continuous and discrete)

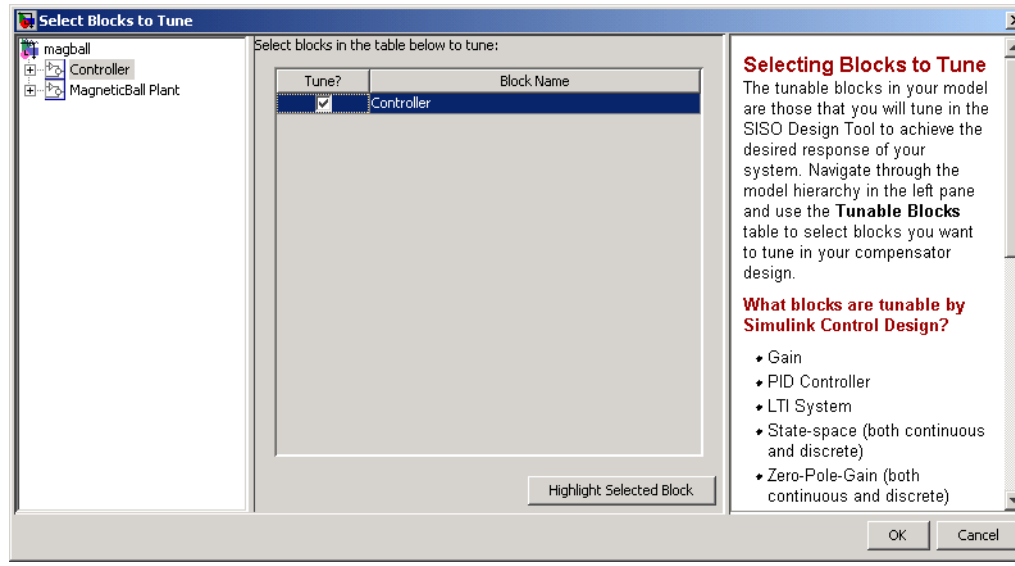
For the complete list of tunable blocks, see "What Blocks Are Tunable?" in the *Simulink Control Design User's Guide*.

Note If your model contains Model blocks with normal mode model references to other models, you can select tunable blocks in the referenced models for compensator design.

In this example, you tune the compensator block called Controller inside the Controller subsystem. To add this as a tunable block for the magball model:

1. Select the **Simulink Compensator Design Task** node.
2. In the **Tunable Blocks** pane, click the **Select Blocks** button. The Select Blocks to Tune dialog box opens.

3 Select the Controller subsystem in the left pane to display that subsystem's tunable blocks within the center pane. Within the center pane, select the check box next to the Controller block's name.



4 Click **OK** to apply your selections, and close the dialog box.

Selecting Closed-Loop Responses to Design

This section continues the magball example from "Picking Blocks to Tune" on page 10-5. At this stage in the example a compensator design task has been created, and tunable blocks have been selected.

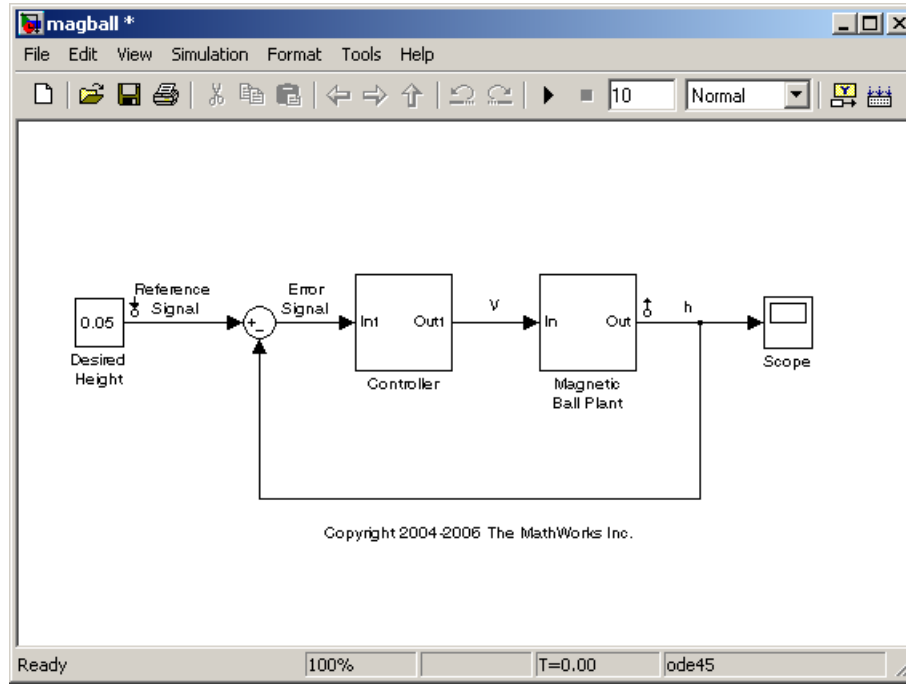
In this step of the compensator design task, you will select the closed loops whose responses you want to design in your model. A closed-loop system is defined by an input point, such as a reference or disturbance signal, and an output point, such as a measured output or actuator signal. The **Simulink Compensator Design Task** uses linearization points on the signal lines of the model to determine the closed-loop systems. For more information on linearization points, see "Configuring Inputs and Outputs for the Linearized Model" on page 9-5.

In this example you will design the response of the closed-loop system from the reference signal to the output of the plant model. To set up linearization points to define this closed-loop system, perform the following steps:

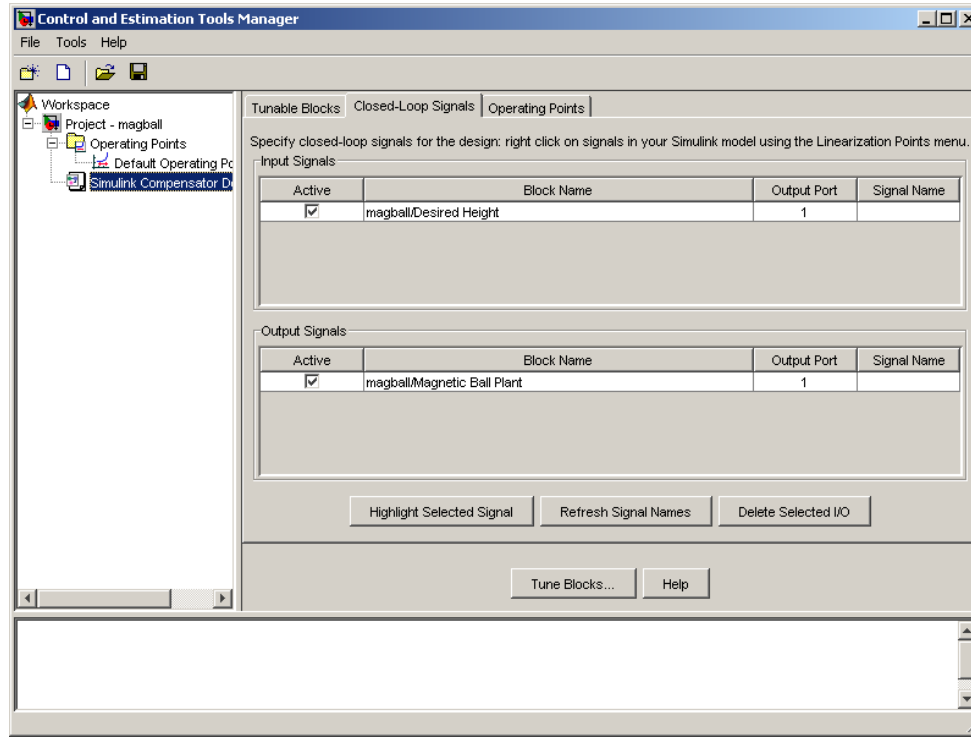
1. On the magball model diagram, position the mouse on the Reference signal between the Desired Height block and the Sum block. Right-click and select **Linearization Points > Input Point** from the menu to add an input point.

2. Position the mouse on the signal line at the output of the Magnetic Ball Plant block. Right-click and select **Linearization Points > Output Point** from the menu to add an output point.

The magball model should now appear as follows:



Within the Control and Estimation Tools Manager, click the **Closed-Loop Signals** tab of the **Simulink Compensator Design Task** node to view the input and output points in the model.



Within this pane you can view the input and output signals in the model and use the **Active** column to select the ones you want to use to define closed-loop systems for compensator design.

Selecting an Operating Point

This section continues the magball example from "Selecting Closed-Loop Responses to Design" on page 10-7. At this stage in the example, a compensator design task has been created, tunable blocks have been selected, and closed-loop signals have been selected.

In this step of the compensator design task, you will select the operating point that you want to use in the compensator design. The Simulink Control Design software uses the operating point when it linearizes the model before creating a SISO Design Task.

Note A compensator designed for the linearized model is likely to control the behavior of the nonlinear model only in a small region around the operating point that the model was linearized at. Therefore it is important that the linearization of the model is accurate and the selection of the operating point about which the system is linearized is an important step in the compensator design process.

In Chapter 8, "Specifying Operating Points", you created operating points for this model, exported one to the MATLAB workspace, and saved it in a MAT-file. This example imports the operating point that you saved. If you did not already compute and save an operating point for the magball model, you can import an operating point that was installed along with the Simulink Control Design demos.

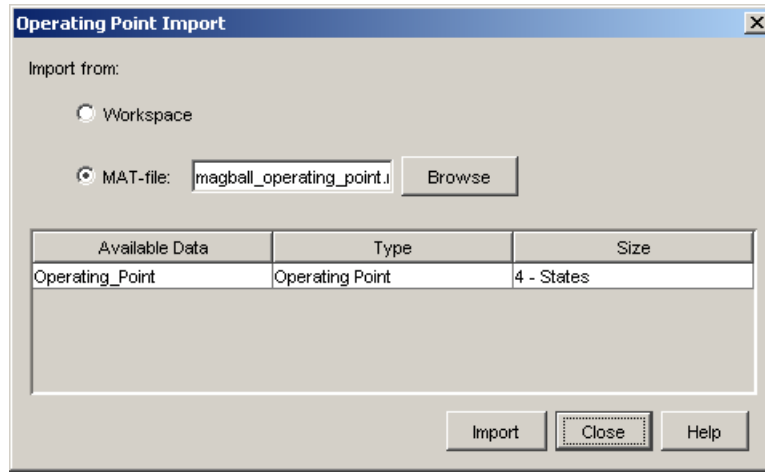
To import an operating point for compensator design, perform the following steps:

1. Select the **Operating Points** node in the Control and Estimation Tools Manager.
2. Click the **Import** button, in the bottom-right corner of the Control and Estimation Tools Manager.
3. In the Operating Point Import dialog box, select **MAT-file** as the location to import from.

4. Click **Browse** and locate the file `magball_operating_point.mat` that you previously saved. If you did not previously save an operating point, browse to *`matlabroot`*`/toolbox/slcontrol/slctrldemos/magball_operating_point.mat`.

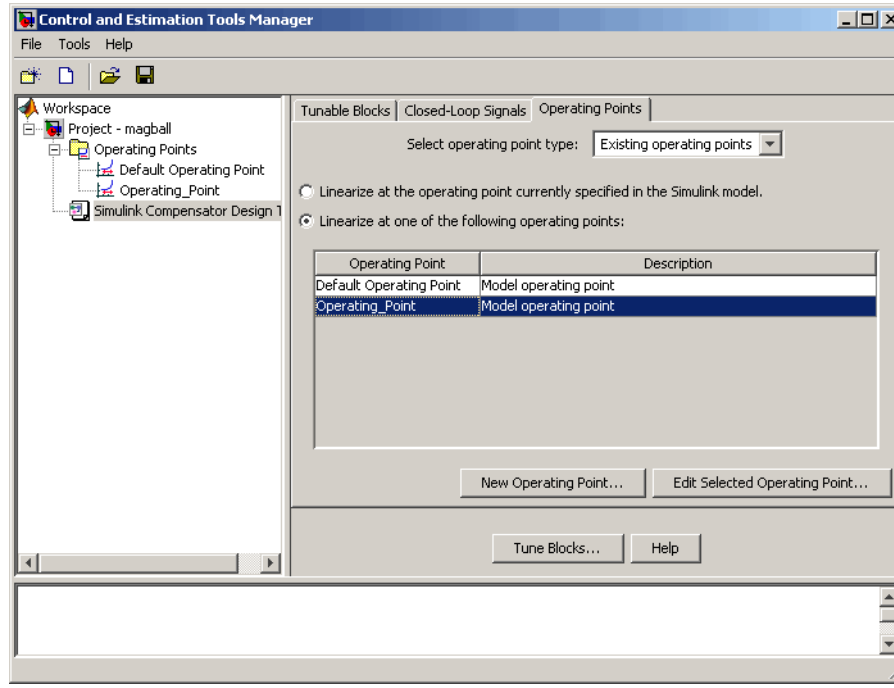
5. Click **Open** to return to the Operating Point Import dialog box.

   The Operating Point Import dialog box now shows all the operating points available within the selected MAT-file. In this case just a single operating point is contained in the MAT-file.



6. Select this operating point and click **Import** to import it into the Control and Estimation Tools Manager.

Click the **Operating Points** tab in the **Simulink Compensator Design Task** node to select an operating point for the compensator design. For this example, you should use the operating point that you just imported, called **Operating_Point**. To specify this operating point, first select the **Linearize at one of the following operating points** option button. Then select **Operating_Point** in the list, as shown in the following figure.



For information on performing compensator design at specified simulation times and events, see "Linearizing at an Operating Point" in the *Simulink Control Design User's Guide*.

Creating a SISO Design Task

In this section...

"What is a SISO Design Task?" on page 10-13

"Configuring Design Plots" on page 10-14

"Configuring Analysis Plots" on page 10-16

What is a SISO Design Task?

This section continues the magball example from "Selecting an Operating Point" on page 10-10. At this stage in the example, a compensator design task has been created, and tunable blocks, closed-loop signals, and an operating point have been selected.

In this step of the compensator design task, you will create and configure a **SISO Design Task** in the Control and Estimation Tools Manager. The **SISO Design Task** includes several tools for tuning the response of SISO systems:

- A graphical editing environment in the SISO Design Tool window that contains design plots such as root-locus, and Bode diagrams
- An LTI Viewer window where you can view time and frequency analysis plots of the system
- A compensator editor where you can directly edit the block mask parameters or the poles and zeros of compensators in your system
- A tool that automatically generates compensators using PID, internal model control (IMC), or linear-quadratic-Gaussian (LQG) methods (uses the Control System Toolbox software)
- A response optimization tool that automatically tunes the system to satisfy design requirements (available when you have the Simulink Response Optimization product)

The Design Configuration Wizard guides you through the selection of the open- and closed-loop systems you want to design and the configuration of the design and analysis plots you want to use in the **SISO Design Task**. To launch the wizard, click the **Tune Blocks** button in the Simulink Compensator Design Task node. The wizard opens in a separate window.

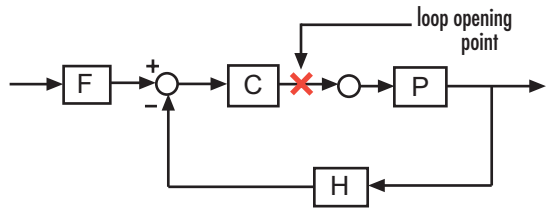
The first page of the wizard provides an overview of the design configuration process and lists some issues to consider when selecting design and analysis plots. Click **Next** to continue to step 1 of the design configuration process on the second page of the wizard.

Configuring Design Plots

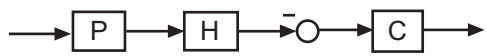
In step 1, select the open- and closed-loop systems that you want to design in your model, and up to six corresponding design plots you want to use.

Open-loop design allows you to design the response of a closed feedback loop in your model by artificially opening the loop and designing the response of this *open-loop* system. The open-loop design plots use rules of linear control theory to determine the dynamics of the closed-loop system from those of the open-loop system. Open-loop design is typically used to tune compensators that lie inside feedback loops.

A set of default open-loop systems is created for your model, shown in the lower half of the wizard. To create these open-loop systems, the software artificially opens the feedback loop at the output signal of each tunable block (at the X in the following figure) and unwraps the closed-loop system to create the corresponding open-loop system.



The unwrapped open-loop system, which is -*CPH*, is shown in the following figure. The open-loop design plots show the negative of the unwrapped open-loop, which is *CPH*. This configuration allows you to design controllers using a negative feedback architecture.



Note that elements that are outside the feedback loop, such as the prefilter **F**, are not seen in the open-loop system.

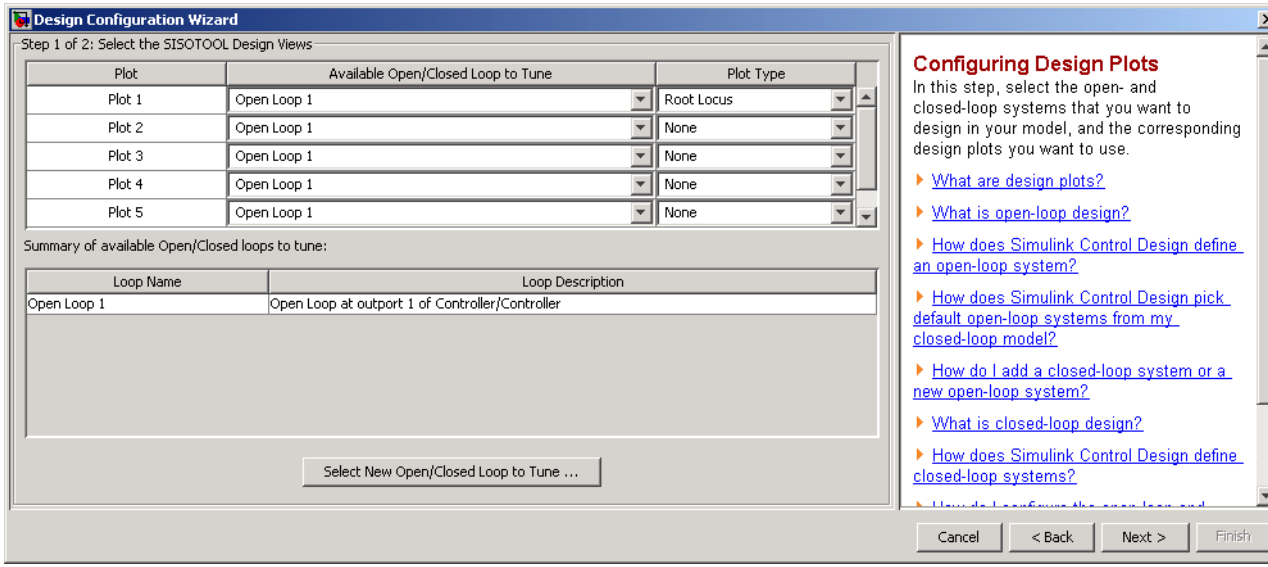
In this example, you will tune the response of **Open Loop 1** which is defined by a loop opening at the output of the Controller block. This open-loop system contains the plant model and the controller. To design this system, select `Open Loop 1` from the menu next to **Plot 1** in the wizard.

Next, select a design plot to use for this open-loop system. Design plots are interactive plots within the SISO Design Tool. You can use them to graphically tune parameters and manually move, add, or remove poles and zeros of the tunable blocks to tune and design the dynamics of open- and closed-loop systems in your model. The following table shows the design plots, along with their uses, available in the SISO Design Tool.

Type of Design Plot	Available Plots in the SISO Design Tool	Use to tune blocks that act as
Open-loop	Root Locus, Nichols, Open-loop bode	Feedback elements
Closed-loop	Closed-loop bode	Feedforward or prefilter elements

You can also use the design plots to specify requirements for stability, performance, or both to use in using optimization-based automated tuning.

For this example, select `Root Locus` from the menu next to **Plot 1** to use this plot type as the design plot for **Open Loop 1**. Step 1 of the wizard should now look similar to the following figure.



Click **Next** to proceed to step 2 of the wizard.

Configuring Analysis Plots

In this step, select the closed-loop responses that you want to view while designing your model, and the corresponding analysis plots you want to use to view them.

Analysis plots are plots that show the responses or dynamics of a closed or open loop systems or tunable blocks in your model. Although you cannot directly edit the analysis plots by graphically moving gains, poles, zeros, etc., changes that you make in the design plots, compensator editor, or automated design tools will affect the responses in the analysis plots. Possible analysis plots include

- Step response
- Impulse response
- Bode and Bode magnitude
- Nyquist

- Nichols
- Pole/Zero

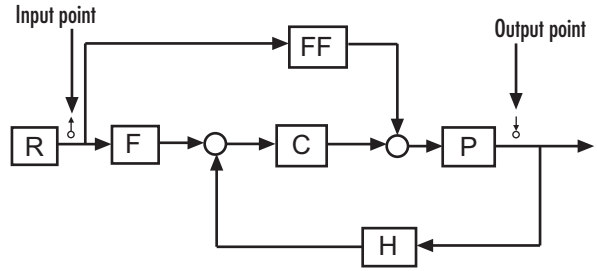
You can use analysis plots to

- Analyze closed-loop, open-loop, and tuned block responses in your control system.
- Define stability and performance requirements for optimization-based automated tuning.

For this example, select `Step` from the menu for **Plot 1** to create a step response analysis plot.

Next, select the closed-loop system that you want to display in this plot. A closed-loop system is a system that has not had any feedback loops opened for open-loop design. It typically defines the system whose response you want to control and it lies between the input and output signals of interest, for example between a reference signal and the plant output signal.

Linearization input and output points placed on signal lines in your model define closed-loop systems. The closed-loop system includes all blocks in the path between the input and the output.

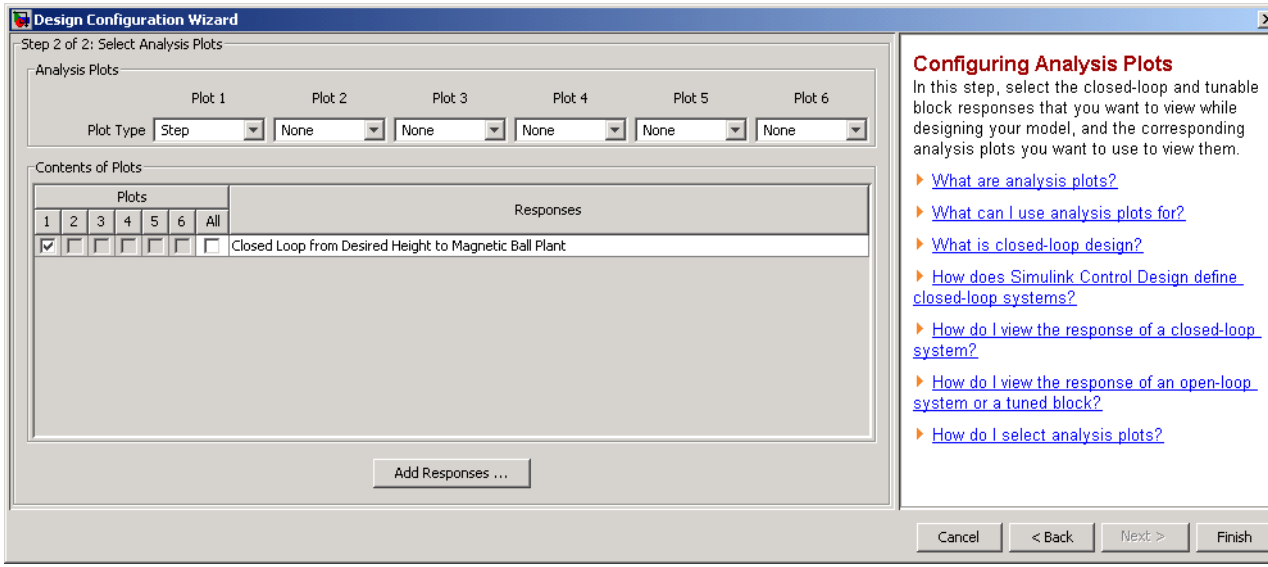


Closed-loop control system

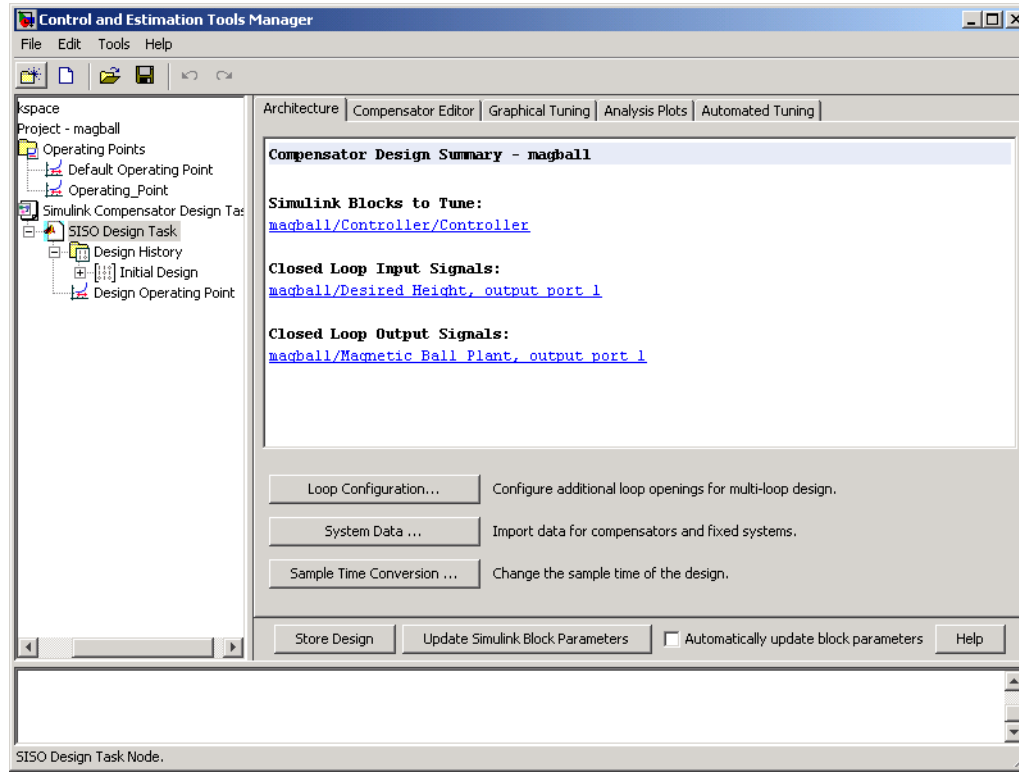
The software automatically displays a list of up to four closed-loop systems in your model, based on the input and output points on the signal lines. In this example, only one closed-loop system appears in the wizard, the closed-loop from the Desired Height signal to the output of the Magnetic Ball Plant Model, because the system only has one input and one output point. You can

add additional closed-loop responses, as well as open-loop and tunable block responses. To add a new response, click the **Add Responses** button and complete the Select a New Response to Analyze dialog box.

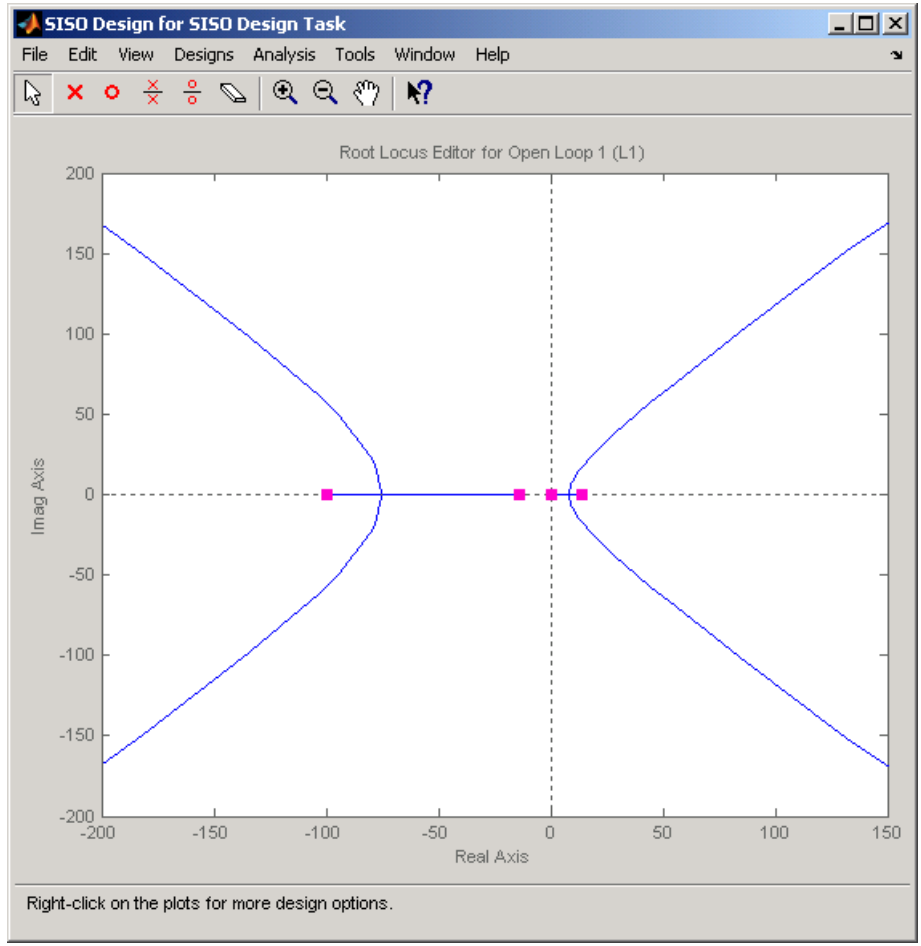
To display the current closed-loop system in the step response plot of **Plot 1**, select the check box under **Plot 1** to the left of the closed-loop system. Step 2 of the wizard should now look similar to the following figure.



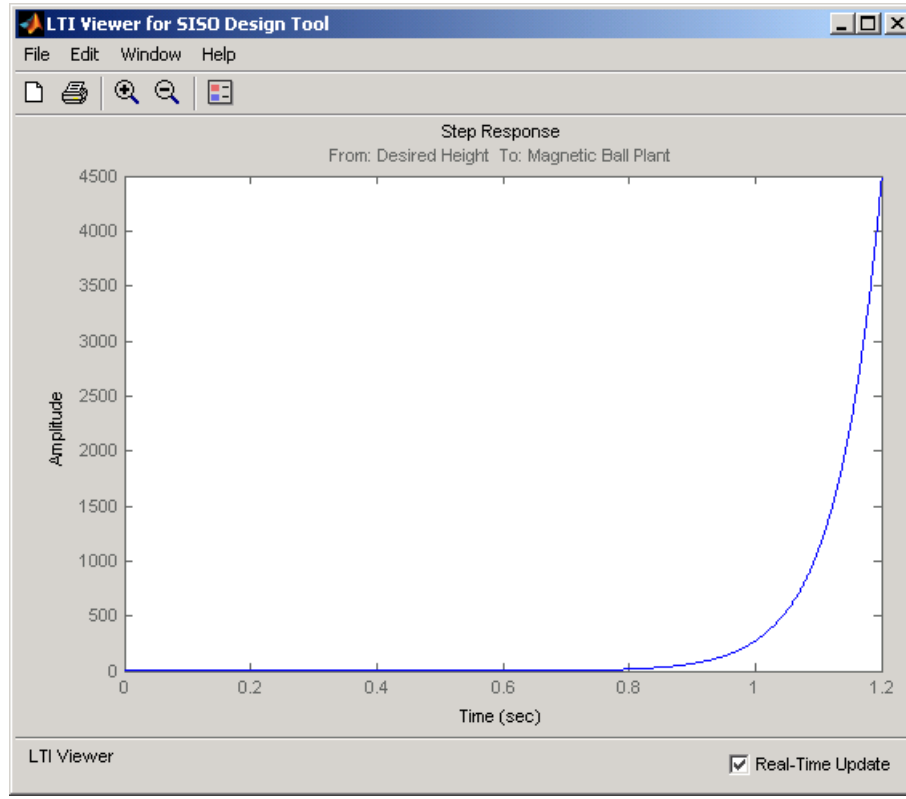
Click **Finish** to complete the wizard and create the **SISO Design Task** underneath the **Simulink Compensator Design Task** node within the Control and Estimation Tools Manager, as shown in the following figure.



The **SISO Design Task** also includes the design plots you configured in the Design Configuration Wizard. They appear within the SISO Design Tool window, as shown in the following figure.



In addition, the **SISO Design Task** also includes the analysis plots you configured in the Design Configuration Wizard. They appear within the LTI Viewer window, as shown in the following figure.


LTI Viewer for SISO Design Tool
File Edit Window Help
Step Response
From: Desired Height To: Magnetic Ball Plant
Amplitude
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0 0.2 0.4 0.6 0.8 1 1.2
Time (sec)
LTI Viewer
Real-Time Update

Completing the Design

In this section...
"Tools for Compensator Design" on page 10-22
"Storing and Retrieving Designs" on page 10-26
"Writing the Design to the Simulink Model" on page 10-28

Tools for Compensator Design

This section continues the magball example from "Creating a SISO Design Task" on page 10-13. At this stage in the example, a compensator design task has been created, tunable blocks, closed-loop signals, and an operating point have been selected, design and analysis plots have been created, and a **SISO Design Task** node has been created in the Control and Estimation Tools Manager.

In this step of the compensator design task, you will complete the design of the compensator in the magball model, using the **SISO Design Task** node. For a more detailed discussion of the **SISO Design Task** node, refer to the Control System Toolbox documentation.

The **SISO Design Task** node contains five panes with various tools for designing the compensators in your system.

- **Architecture**:
  - Configure loops for multi-loop design by opening signals to remove the effects of other feedback loops.
  - Import compensators into your system.
  - Convert the sample time of the system or switch between different sample times to design different compensators.
- **Compensator Editor**:
  - Directly edit the poles, zeros, and gains of the compensator.
  - Add or remove poles and zeros to the compensators.

- **Graphical Tuning**:
  - Configure design plots in the SISO Design Tool.
  - Use design plots to graphically manipulate the response of the system.
- **Analysis Plots**:
  - Configure analysis plots in the LTI Viewer.
  - Use analysis plots to view the response of open- or closed-loop systems.
- **Automated Tuning**: Design compensators using one of several automated methods.
  - Automatically generate compensators using PID, internal model control (IMC), or linear-quadratic-Gaussian (LQG) methods (uses Control System Toolbox software).
  - Use a response optimization tool that automatically tunes the system to satisfy design requirements (available when you have the Simulink Response Optimization product).

You can use any of these design methods, or a combination of methods, to design the compensators for your system. A suitable final design for the Controller of the magball model is:

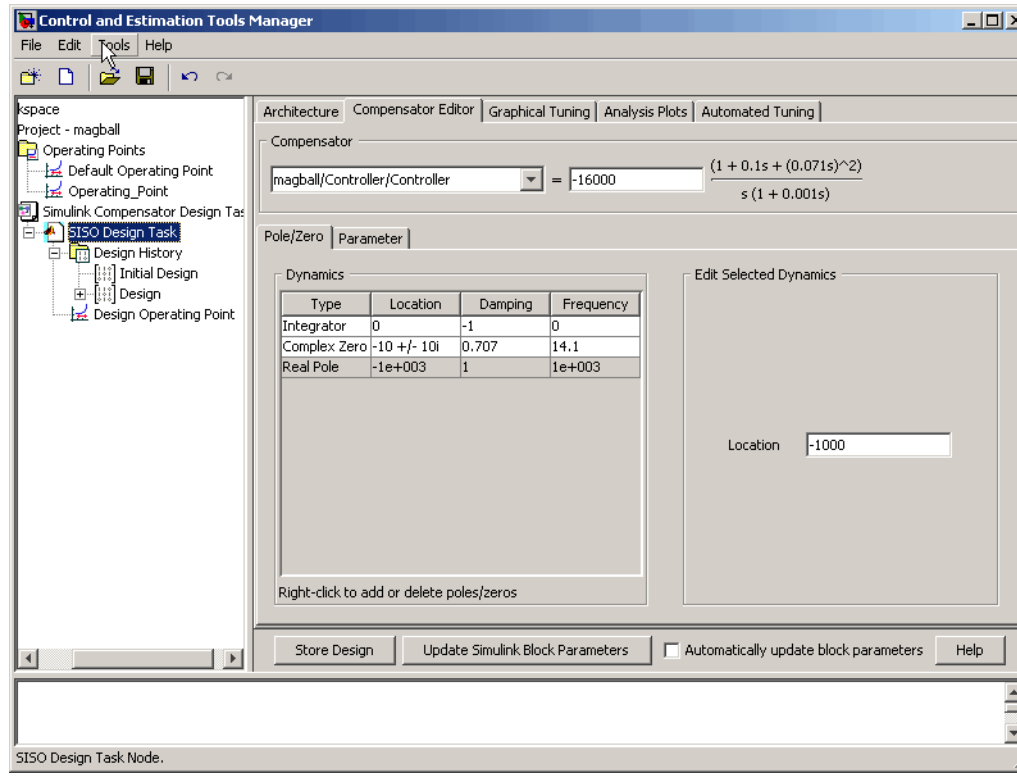
- Gain: -16000
- Integrator at the origin
- Complex zeros at -10±10i
- Real pole at -1000

You can use the **Compensator Editor** in the **SISO Design Task** node to specify these settings. The initial design contains an integrator at the origin. Specify the remaining settings as follows:

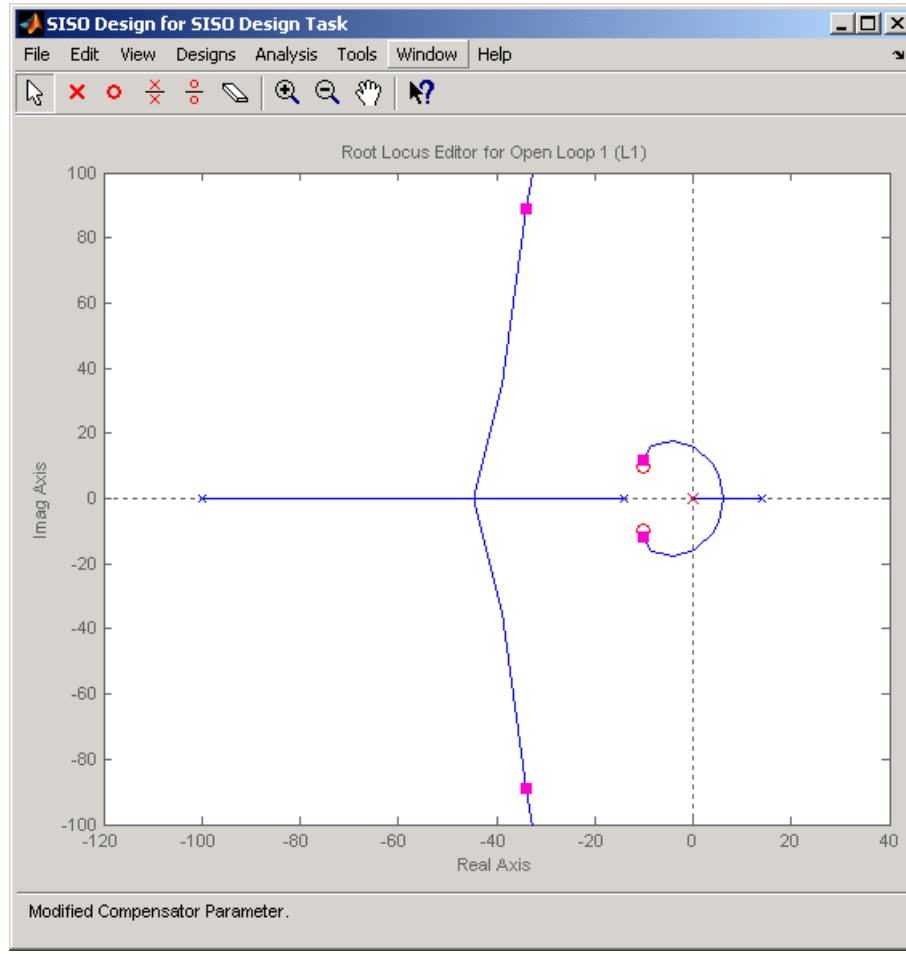
- Gain — Enter `-16000` in the text box to the right of the equal sign in the **Compensator** area.
- Complex zeros — In the **Dynamics** table, right-click and then select **Add Pole/Zero > Complex Zero**. Select the new complex zero in the **Dynamics** table. In the **Edit Selected Dynamics** table:

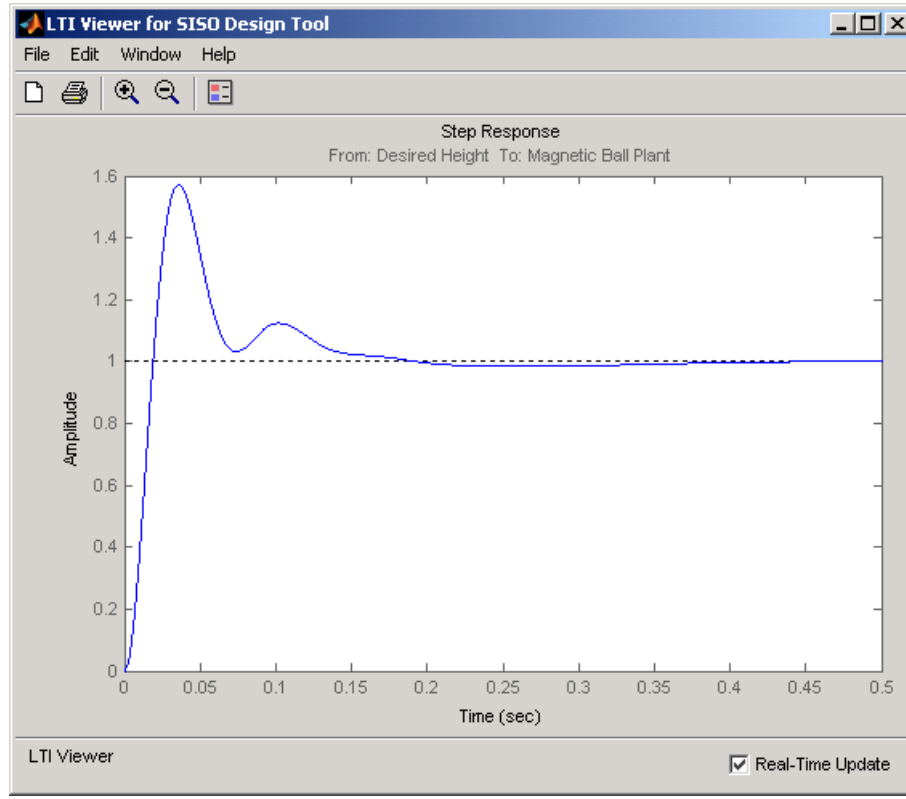
  - Enter `-10` in the **Real Part** field.
  - Enter `10` in the **Imaginary Part** field.
- Real pole — In the **Dynamics** table, right-click and then select **Add Pole/Zero > Real Pole**. Select the new real pole in the **Dynamics** table. In the **Edit Selected Dynamics** table:
  - Enter `-1000` in the **Location** field.

The Control and Estimation Tools Manager should now appear as follows:



With these settings, the root-locus diagram and step-response plot should look similar to the following figure.


SISO Design for SISO Design Task
File Edit View Designs Analysis Tools Window Help
Root Locus Editor for Open Loop 1 (L1)
Imag Axis
Real Axis
100
80
60
40
20
0
-20
-40
-60
-80
-100
-120
-100
-80
-60
-40
-20
0
20
40
Modified Compensator Parameter.



Storing and Retrieving Designs

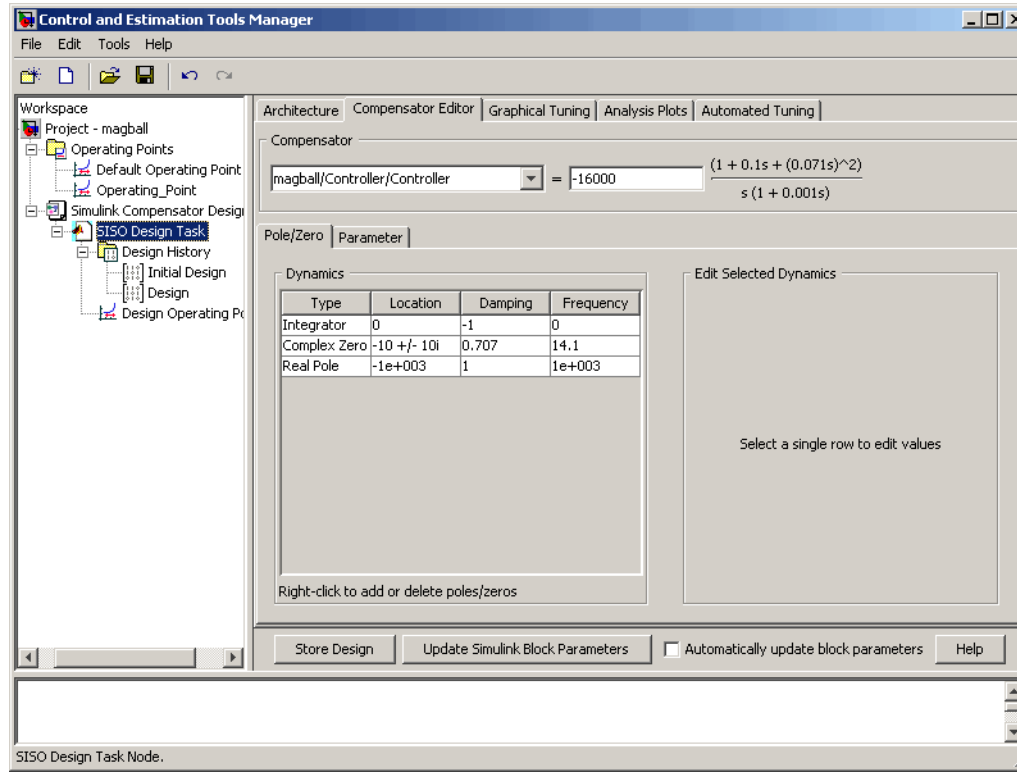
When you design a compensator within a **Simulink Compensator Design Task**, you can store the current design. You can retrieve the stored design at any time. This storage and retrieval capability lets you continue designing and still be able to return to a previously saved version of the design.

This section continues the example from "Completing the Design" on page 10-22. At this stage in the example, a compensator has been designed to control the system. To store the design within the **SISO Design Task** node, perform the following steps:

1 Select the **SISO Design Task** node in the Control and Estimation Tools Manager.

2 Underneath the **SISO Design Task** panes, click the **Store Design** button.

Clicking this button adds the current design to the **Design History** node as shown in the following figure. The default name for the design is **Design**.



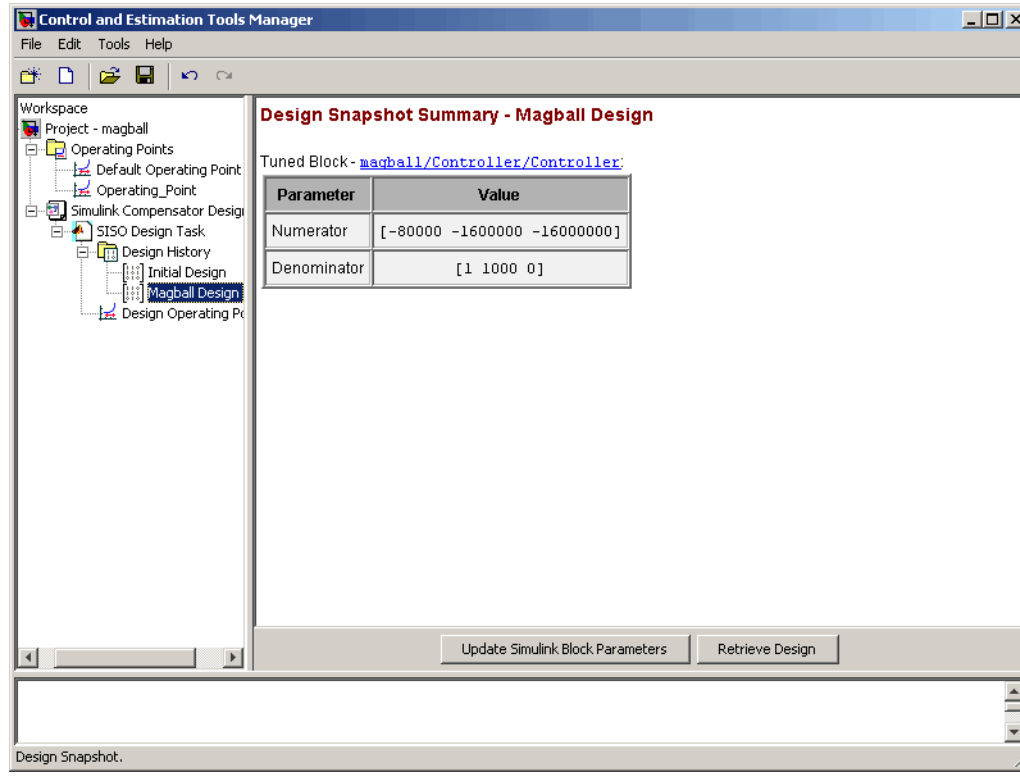
To rename the design to something more descriptive:

1 Right-click the **Design** node underneath the **Design History** node.

2 Select **Rename** from the right-click menu.

3 Enter a name for your design. For this example, call the design **Magball Design**.

The Control and Estimation Tools Manager should now appear as follows:



Note After you store a compensator design, you can continue to refine it. If you make undesired changes, you can retrieve the stored design by selecting it in the **Design History** node and then clicking the **Retrieve Design** button.

Writing the Design to the Simulink Model

When designing a compensator within a **Simulink Compensator Design Task** node, you can write the compensator design to the Simulink model. This is useful when

- You want to see how the current design performs in the full nonlinear model.

- You have completed the design and you want to update the model with the newly designed parameters.

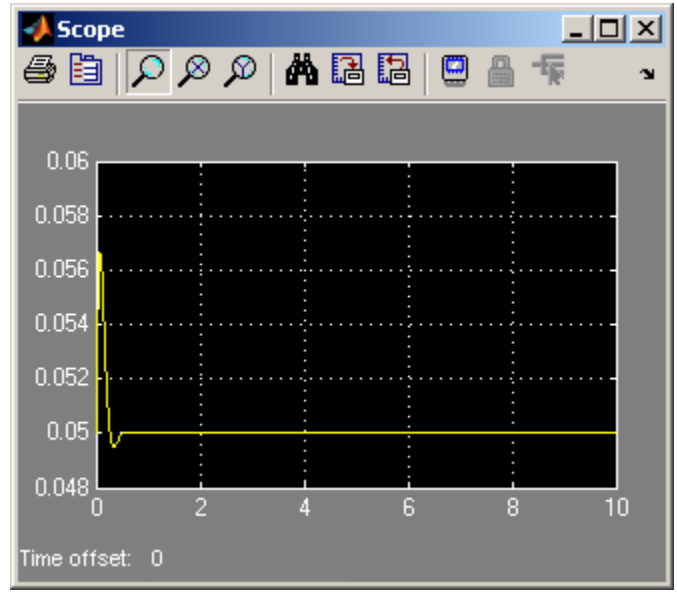
This section continues the example from "Storing and Retrieving Designs" on page 10-26. At this stage in the example a compensator has been designed to control the system and the design has been stored within the **SISO Design Task** node. To write the stored design to the magball model, perform the following steps:

1. Select the **Magball Design** node under the **Design History** node in the Control and Estimation Tools Manager.

2. Click the **Update Simulink Block Parameters** button.

You can now simulate the magball model containing the newly designed Controller block. For more information on simulating models, see "Running Simulations" in the Simulink documentation.

After simulation, the Scope block of the magball model should look similar to the following figure.



As you can see, the system is now stable and the height of the magnetic ball settles at the desired height of 0.05 m.

11

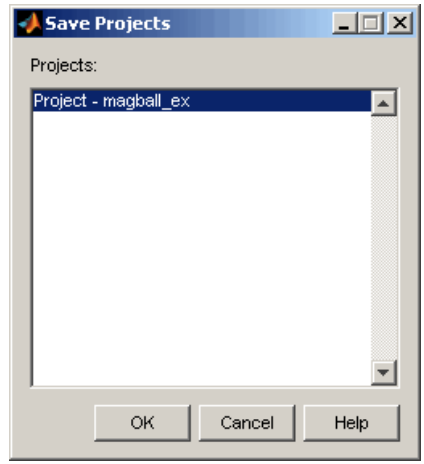
Saving and Exporting Your Work

- "Saving Projects" on page 11-2
- "Opening Previously Saved Projects" on page 11-3
- "Exporting Results" on page 11-4

Saving Projects

A Control and Estimation Tools Manager project can consist of multiple tasks such as linearization, compensator design, and operating points tasks as well as Simulink Parameter Estimation tasks and Model Predictive Control Toolbox tasks. Each task contains data, objects, and results for the analysis of a particular model.

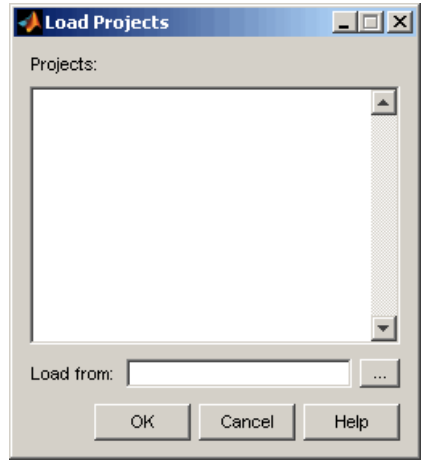
1. To save a project as a MAT-file, select **File > Save** from the Control and Estimation Tools Manager window.



2. In the Save Projects dialog box, select one or more projects you want to save. You can save multiple projects in one file. Click **OK**, and browse to the folder where you want to save the project. Enter the project name, and click **Save**.

Opening Previously Saved Projects

1 To open previously saved projects, select **File > Load** from the Control and Estimation Tools Manager window. This opens the Load Projects dialog box.



2 Choose a project-file by either browsing for the directory and file, or typing the full path and filename in the **Load from** field. Project files are always MAT-files. After you specify the file, the projects contained in this file appear in the list. Select one or more projects in the list, and then click **OK**. When a file contains multiple projects, you can choose to load them all or just a few.

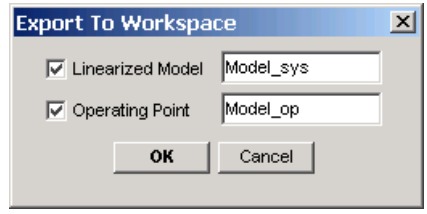
Exporting Results

In this section...

"Exporting Linearization Results" on page 11-4
"Exporting Compensator Designs" on page 11-5
"Exporting Operating Points" on page 11-6
"Exporting and Restoring Linearization I/O Settings" on page 11-6

Exporting Linearization Results

To export linearization results and the corresponding operating points, right-click the results node, **Model**, under the **Linearization Task** node and select **Export** from the menu. In the Export To Workspace dialog box, choose new names for the linearized model and operating point, or accept the defaults, and then click **OK**.



The MATLAB workspace now contains two new objects, `Model_op` and `Model_sys`. To see this, type

```
who
```

at the MATLAB prompt. This returns

```
Your variables are:

L           Model_sys   beta        m
Model_op    R           g
```

Alternatively, you can export the results to the MATLAB workspace by selecting **File > Export** from the LTI Viewer window or by clicking the

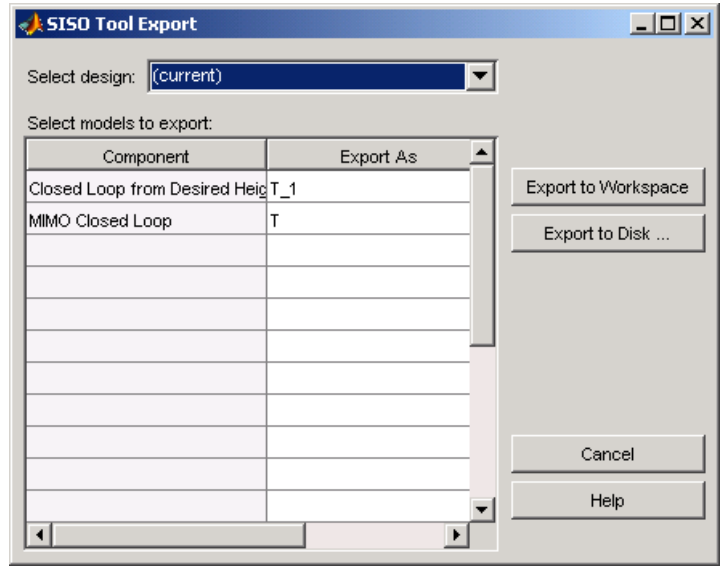
Export to Workspace button at the bottom of the **Linearization Summary** pane within the **Model** node.

By right-clicking the results node, **Model**, you can also delete results.

Exporting Compensator Designs

To export a compensator design to the MATLAB workspace:

1. Select **File > Export** from the SISO Design Tool window.
2. In the SISO Tool Export dialog box, use the **Select design** list to choose the design you want to export.
3. In the list, select the compensators to export, and then click either **Export to Workspace** or **Export to Disk**.



Exporting Operating Points

After creating operating points, you can use the Export To Workspace dialog box to export them for use outside of the Control and Estimation Tools Manager. You can use the exported operating point to perform analysis at the MATLAB command line or to initialize a model for simulation.

1. Under **Select destination workspace**, select either
   - **Base workspace** to export the operating point to the MATLAB workspace where you can use it with Simulink Control Design command-line functions
   - **Model workspace** to export the operating point to the Model workspace where you can save it with the model for future use.
2. Enter a name for the exported operating point.
3. Select **Use the operating point to initialize model** when you want to use the operating point values as initial conditions for the states and inputs in the model. The initial values are automatically set in the **Data Import/Export** pane of the Configuration Parameters dialog box. Simulink software uses these initial conditions when simulating the model.

Exporting and Restoring Linearization I/O Settings

To export linearization I/O settings to the MATLAB workspace, use the `getlinio` function. You can save these settings using the `save` function. Use them in a later session by reloading them with the `load` function. Upload them to the model diagram with the `setlinio` function.

For more information, see the function reference pages for `getlinio` and `setlinio`.

A

Examples

Use this list to find examples in the documentation.

Getting Started

Chapter 2, “Tutorial — Computing a Steady-State Operating Point for a Simulink Model Using the GUI”
Chapter 3, “Tutorial — Computing a Steady-State Operating Point for a Simulink Model Using the Command Line”
Chapter 4, “Tutorial — Linearizing a Plant in a Single-Loop Control System Using the GUI”
Chapter 5, “Tutorial — Linearizing a Plant in a Single-Loop Control System Using the Command Line”

Linearization Example Using the Graphical Interface

“Example Model: The Magnetic Ball System” on page 7-2

Index

A

analysis I/Os. *See* input and output points

C

compensator design
  analysis plots 10-16
  beginning a task 10-3
  closed-loop responses 10-7
  creating a SISO Design Task 10-13
  design methods 10-22
  design plots 10-14
  exporting 11-5
  open-loop systems 10-14
  operating points 10-10
  overview 10-2
  quick start guide 1-9
  retrieving designs 10-26
  storing designs 10-26
  tunable blocks 10-5
  writing to Simulink model 10-28
compensator design tasks
  opening 11-3
  saving 11-2
Control and Estimation Tools Manager
  compensator design task 10-3
  opening 7-7
  overview 7-8
  SISO Design Task 10-13

E

equilibrium operating points 8-2
  computing 8-9
equilibrium states
  computing 8-9

I

inport and outport blocks
  difference from input and output points 9-6
input and output points
  inspecting with GUI 9-12
  purpose 9-5
  removing 9-8
  specifying 9-6
input points. *See* input and output points

L

linearization
  accurate 8-7
  beginning a new project 7-7
  benefits of command-line functions 1-4
  benefits of GUI 1-4
  choosing operating points 8-7
  custom plots 9-23
  displaying results 9-18
  factors affecting 1-3
  highlighting blocks 9-20
  inspecting 9-21
  operating points 8-6
  overview 9-3
  purpose of 1-2
  quick start guide 1-7
  removing results 9-18
  results 9-18
  role in compensator design 1-3
  using GUI 9-15
linearization method
  for nonlinear models 9-2
  for Simulink models 9-3
linearization points
  adding 9-6
  deleting 9-8
  removing 9-8
  selecting 9-6

linearization results
custom plots 9-23
exporting 11-4
in LTI viewer 9-17
inspecting 9-21
linearization tasks
opening 11-3
saving 11-2
loop opening
purpose 9-9

M

magnetic ball
equations 7-3
example 7-2
Simulink model 7-4

O

open-loop analysis 9-9
operating points
choosing 8-7
computing equilibrium 8-9
computing from specifications 8-9
creating 8-9
default operating point 8-9
definition 8-2
equilibrium 8-2
extracted from simulation times 8-18
impact 8-6
importance 8-6
saving and exporting 8-21
Simulink model 8-3
specifying with completely known values 8-15
specifying with GUI 8-1
steady state 8-2
trimming 8-9
output points. *See* input and output points

P

project
linearization 7-7

R

retrieving compensator designs 10-26

S

Simulink Control Design Product 1-2
Simulink model operating points 8-3
SISO Design Task
creating 10-13
steady state operating points 8-2
computing 8-9
storing compensator designs 10-26

T

trimming
operating points 8-9