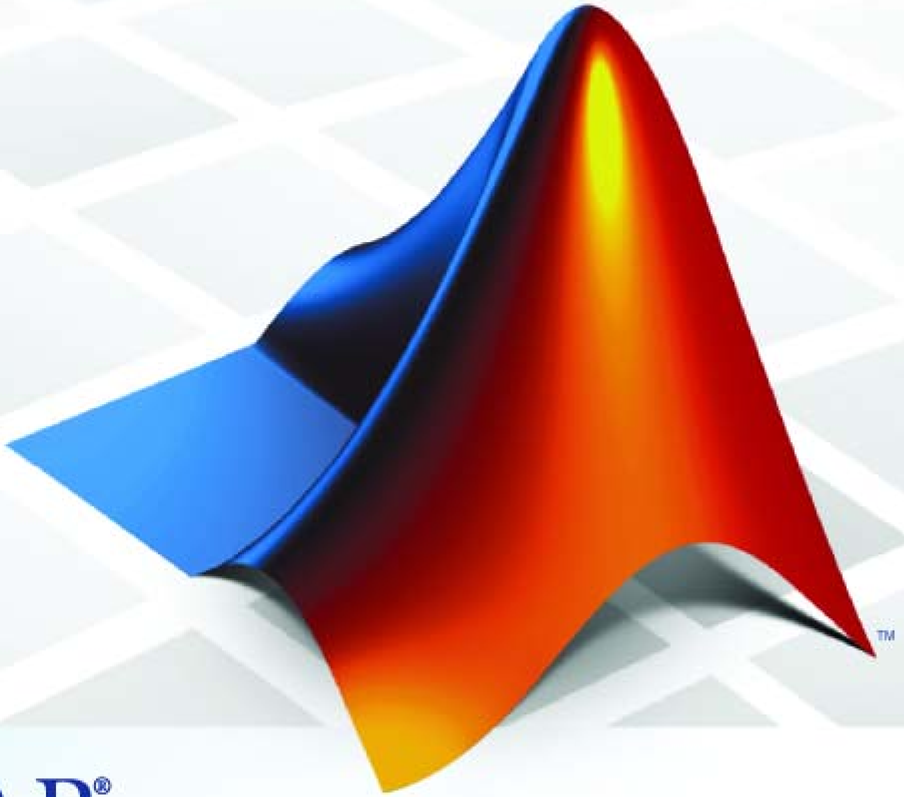
Simulink® Control Design™ 3

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

Trademarks

Patents

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	Second printing	Revised for Version 2.3 (Release 2008a)
October 2008	Online only	Revised for Version 2.4 (Release 2008b)
March 2009	Online only	Revised for Version 2.5 (Release 2009a)
September 2009	Third printing	Revised for Version 3.0 (Release 2009b)

Contents

1 Product Overview

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

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

3

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

4

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

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

A Examples

Index

Product Overview

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.

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 are

- Choice of operating points. See “Why Are Operating Points Important?”.
- Understanding the equations for the linearized model. See “What Is Linearization?”.
- Controlling the effect of feedback loops. See “What Is Open-Loop Analysis?”.

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

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, see Chapter 4, "Tutorial — Linearizing a Plant in a Single-Loop Control System Using the GUI".

To quickly get started designing compensators, see Chapter 6, "Tutorial — Designing a PID Compensator for a Single-Loop Feedback System".

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

All users should read "Exact Linearization Using the GUI" and "Designing Compensators", which describe features and use of the Simulink Control Design GUI.

To automate the linearization process, or perform batch linearization, continue with "Exact Linearization Using the Command Line" 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.

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/`.

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

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:

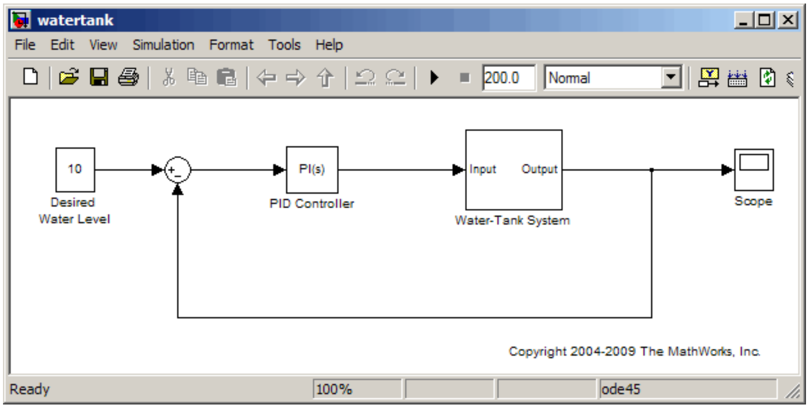
- Compute a steady-state operating point of a Simulink Model.
- Simulate the model at the steady-state operating point.

About the Model

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

The watertank Simulink Model

The `watertank` model shown in the following figure contains the Water-Tank System plant and a PI controller configured in the PID Controller block, in a single-loop feedback system.

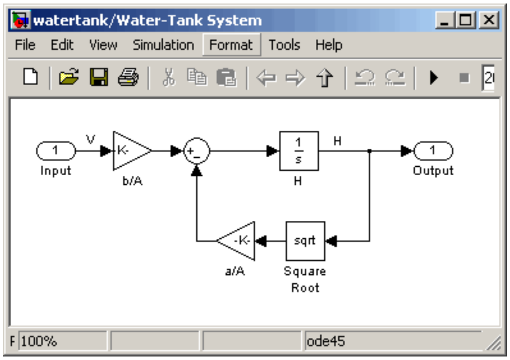


To view the Water-Tank System, double-click the corresponding subsystem in the watertank model. For descriptions of these subsystems, see "Water-Tank Subsystem" on page 2-3.

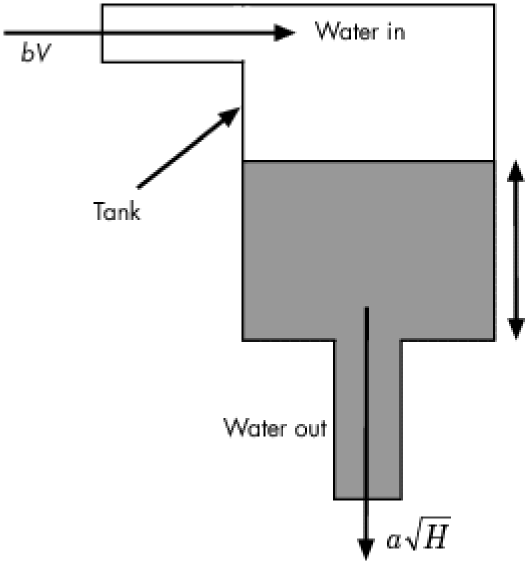
For information about creating Simulink models, see "Creating a Simulink Model".

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

Computing a Steady-State Operating Point

In this section...

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".

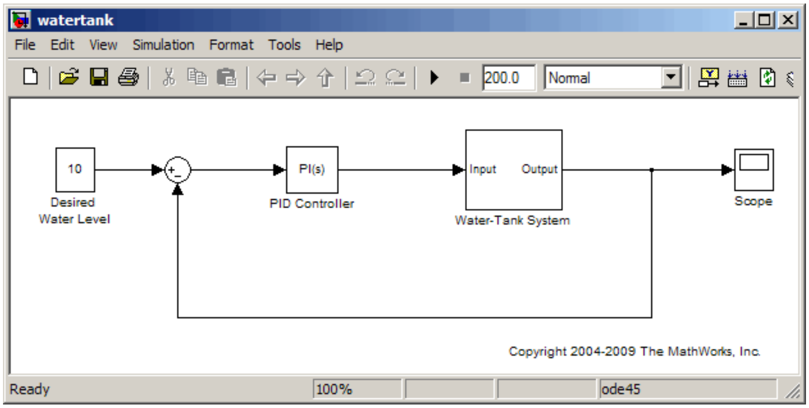
How to Compute a Steady-State Operating Point

To compute a steady-state operating point:

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

   ```
   watertank
   ```

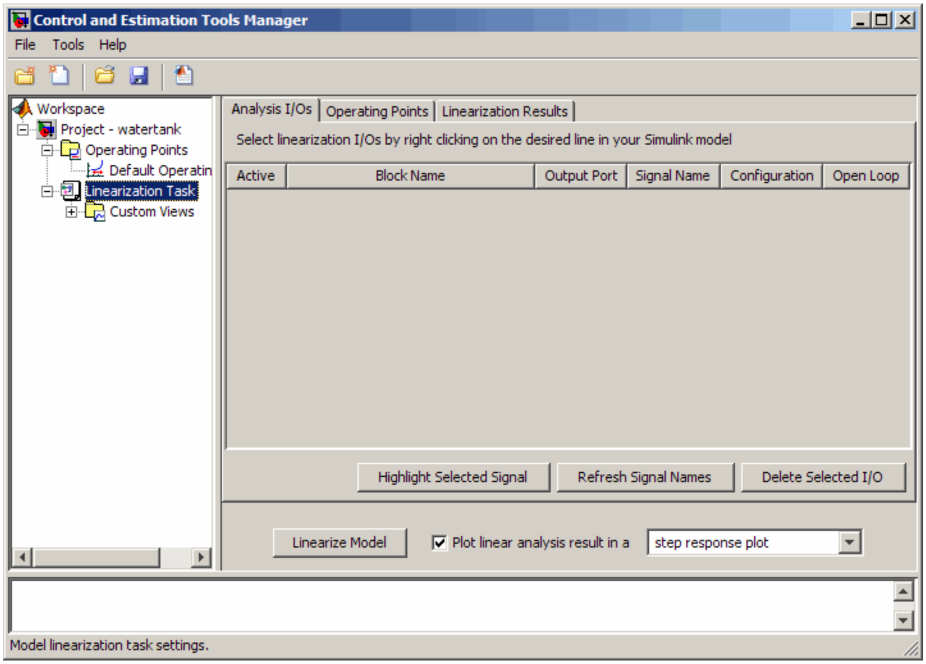
   The model opens in Simulink, as shown in the following figure.



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)



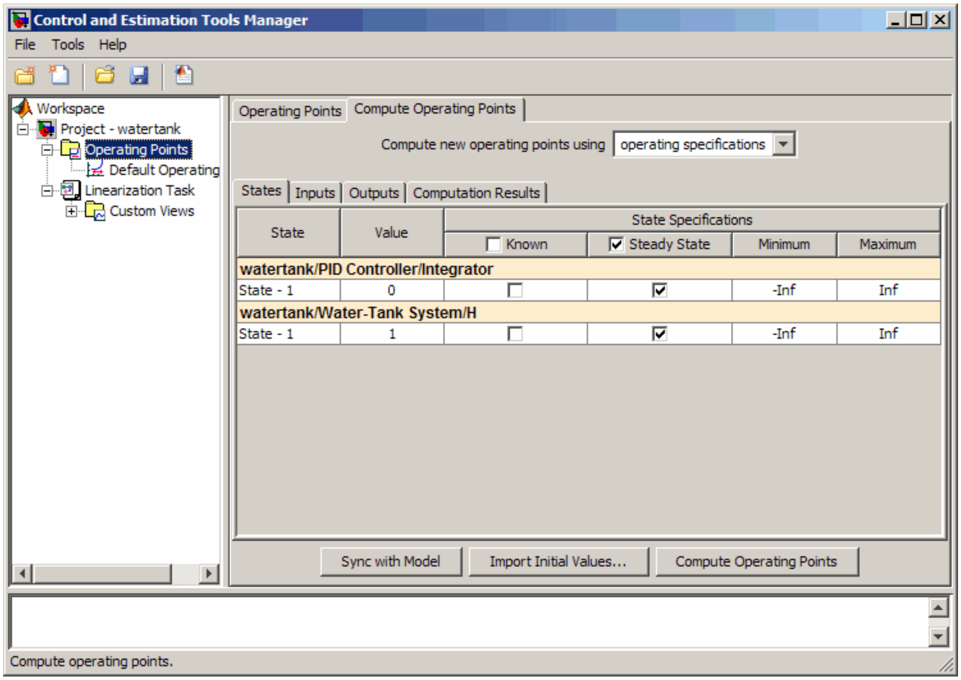
Select linearization I/Os by right clicking on the desired line in your Simulink model

☑ Plot linear analysis result in a: step response plot

Model linearization task settings.

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.



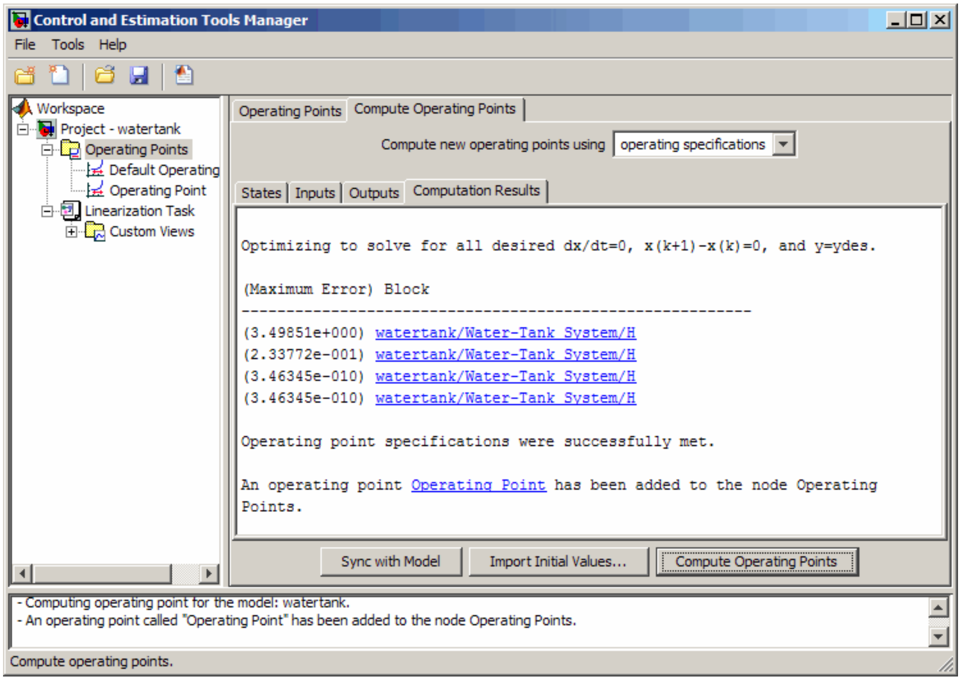
Compute new operating points using: operating specifications

State	Value	State Specifications			
		□ Known	☑ Steady State	Minimum	Maximum
watertank/PID Controller/Integrator					
State - 1	0	□	☑	-Inf	Inf
watertank/Water-Tank System/H					
State - 1	1	□	☑	-Inf	Inf

Compute operating points.

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.



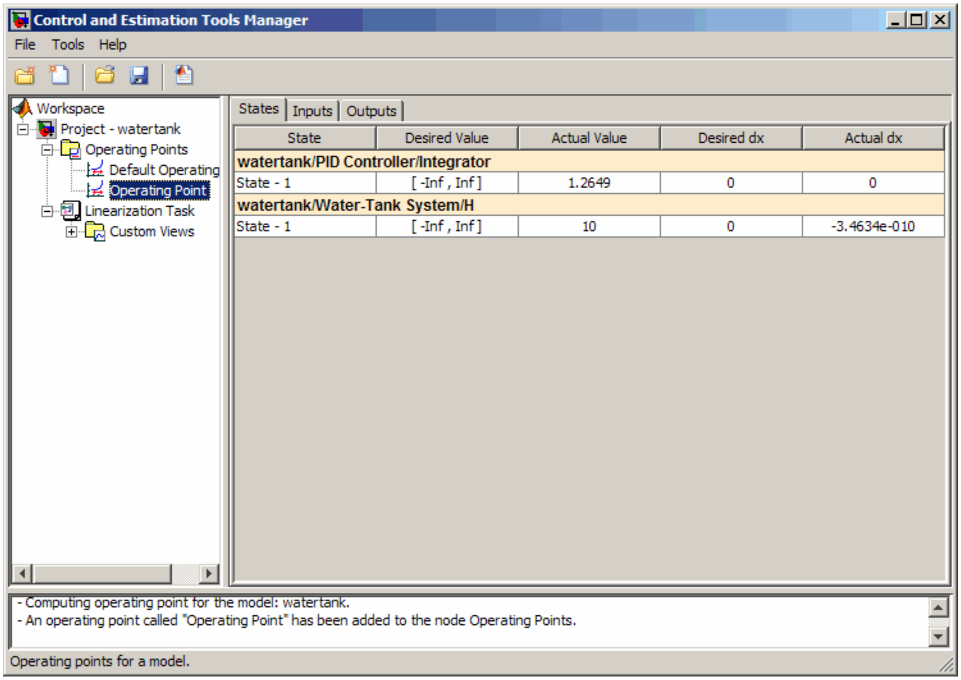
Compute new operating points using: operating specifications

```
Optimizing to solve for all desired dx/dt=0, x(k+1)-x(k)=0, and y=ydes.

(Maximum Error) Block
---------------------------------------------------------
(3.49851e+000) watertank/Water-Tank System/H
(2.33772e-001) watertank/Water-Tank System/H
(3.46345e-010) watertank/Water-Tank System/H
(3.46345e-010) watertank/Water-Tank System/H

Operating point specifications were successfully met.

An operating point Operating Point has been added to the node Operating
Points.
```

```
- Computing operating point for the model: watertank.
- An operating point called "Operating Point" has been added to the node Operating Points.
```

Compute operating points.

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



State	Desired Value	Actual Value	Desired dx	Actual dx
watertank/PID Controller/Integrator				
State - 1	[-Inf , Inf]	1.2649	0	0
watertank/Water-Tank System/H				
State - 1	[-Inf , Inf]	10	0	-3.4634e-010

- Computing operating point for the model: watertank.
- An operating point called "Operating Point" has been added to the node Operating Points.

Operating points for a model.

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.

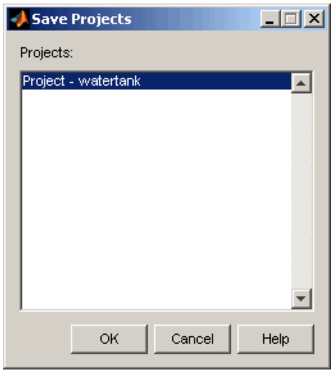
  The time derivatives are 0 for the state *Integrator* and -3.4634e-10 for the state *H*. These values show that the operating point is at steady state.

Tip To automatically generate MATLAB code that computes operating points as specified in the Control and Estimation Tools Manager, click 📄 or select **File > Generate MATLAB Code**.

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".

Simulating the Model at the Steady-State Operating Point

In this section...

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 2-6.

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

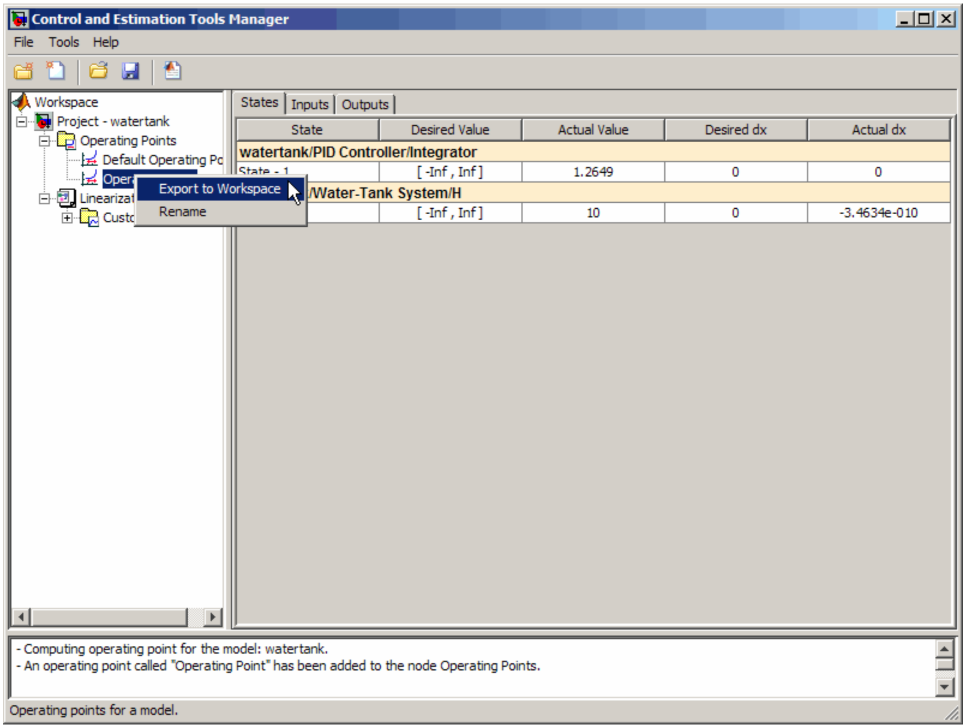
1. Initialize the Simulink model with the steady-state operating point. See "Initializing the Simulink Model with the Steady-State Operating Point" on page 2-13.

2. Simulate the initialized model. See "Simulating the Initialized Model" on page 2-15.

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**.



State	Desired Value	Actual Value	Desired dx	Actual dx
watertank/PID Controller/Integrator				
State - 1	[-Inf , Inf]	1.2649	0	0
/Water-Tank System/H				
	[-Inf , Inf]	10	0	-3.4634e-010

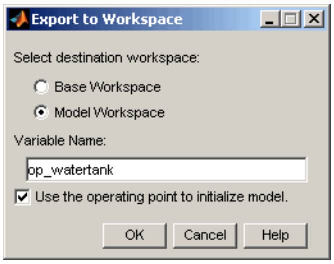
- Computing operating point for the model: watertank.
- An operating point called "Operating Point" has been added to the node Operating Points.

Operating points for a model.

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.



Select destination workspace:

○ Base Workspace

● Model Workspace

Variable Name:

op_watertank

☑ Use the operating point to initialize model.

3 Click **OK**.

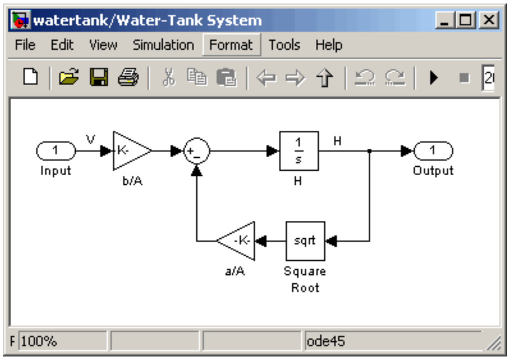
   This action loads the operating point into the Simulink model workspace.

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:

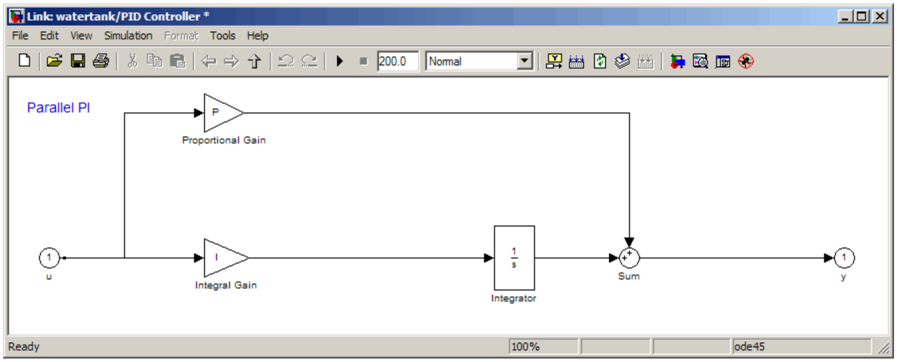
1 In the Water-Tank System subsystem, right-click the output signal of the integrator, which is the state *H*, and select **Create and Connect Viewer > Simulink > Scope**. This action opens a Scope for viewing the state *H*.


watertank/Water-Tank System
File Edit View Simulation Format Tools Help
V
1
Input
K-
b/A
+
-
1
s
H
H
1
Output
-K-
a/A
sqrt
Square Root
F 100%
ode45

2 Add a Scope for viewing the state *Integrator*:

   a In the `watertank` model, right-click the PID Controller block and select **Look Under Mask**.

      This action opens the PID Controller block mask.



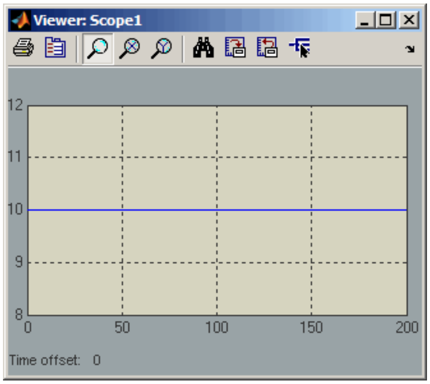
b In the block mask, right-click the output signal of the Integrator, which is the state *Integrator*, and select **Create and Connect Viewer > Simulink > Scope**. This action opens a Scope for viewing the state *Integrator*.

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

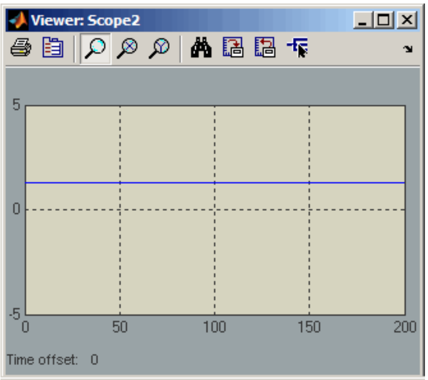
This action displays the states *H* and *Integrator* in their respective Scope windows. The Scope outputs shows these results:

- 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.

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

About This Tutorial

In this section...

Objectives

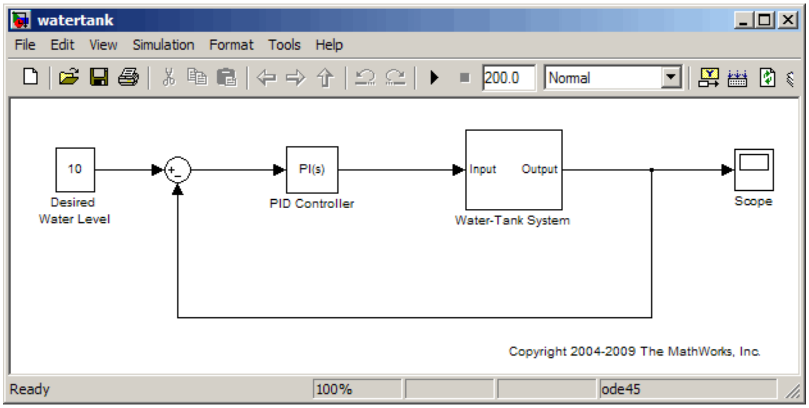
In this tutorial, you learn how to use Simulink Control Design functions at the command line to accomplish the following tasks:

- 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

The `watertank` model shown in the following figure contains the Water-Tank System plant and a PI controller configured in the PID Controller block, in a single-loop feedback system.

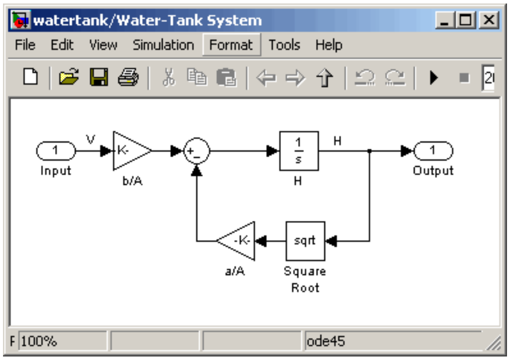


To view the Water-Tank System, double-click the corresponding subsystem in the `watertank` model. For descriptions of these subsystems, see “Water-Tank Subsystem” on page 3-3.

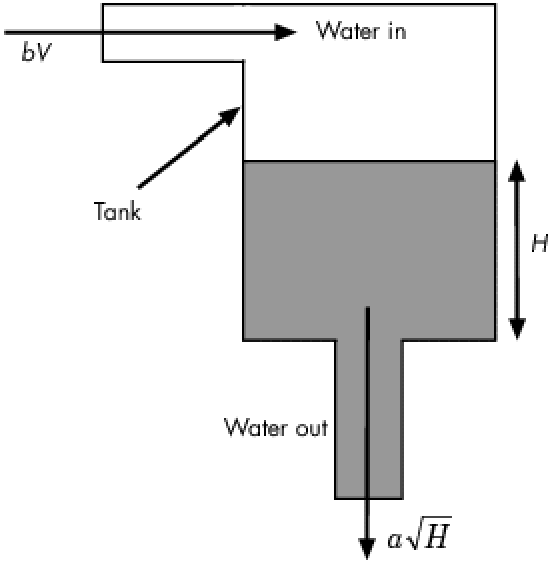
For information about creating Simulink models, see “Creating a Simulink Model”.

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

Computing a Steady-State Operating Point

In this section...

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".

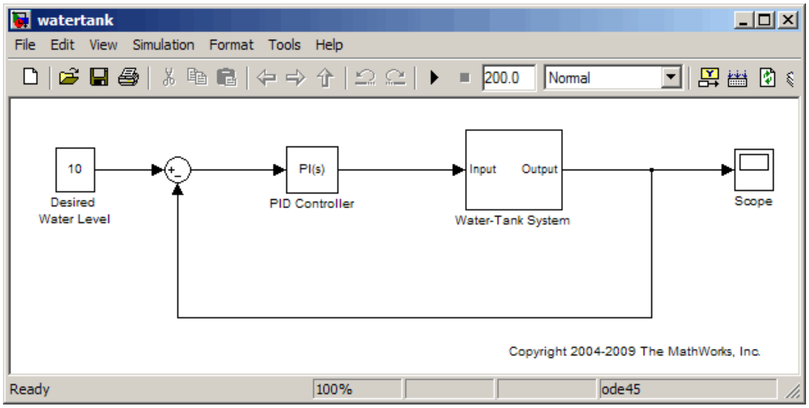
How to Compute a Steady-State Operating Point

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

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/PID 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/PID Controller/Integrator
      x:          1.26      dx:            0 (0)
(2.) watertank/Water-Tank System/H
      x:            10      dx:    -3.46e-010 (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 -3.46e-010 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

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 3-6.

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 3-10.

2. Simulate the initialized model. See "Simulating the Initialized Model" on page 3-11.

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.

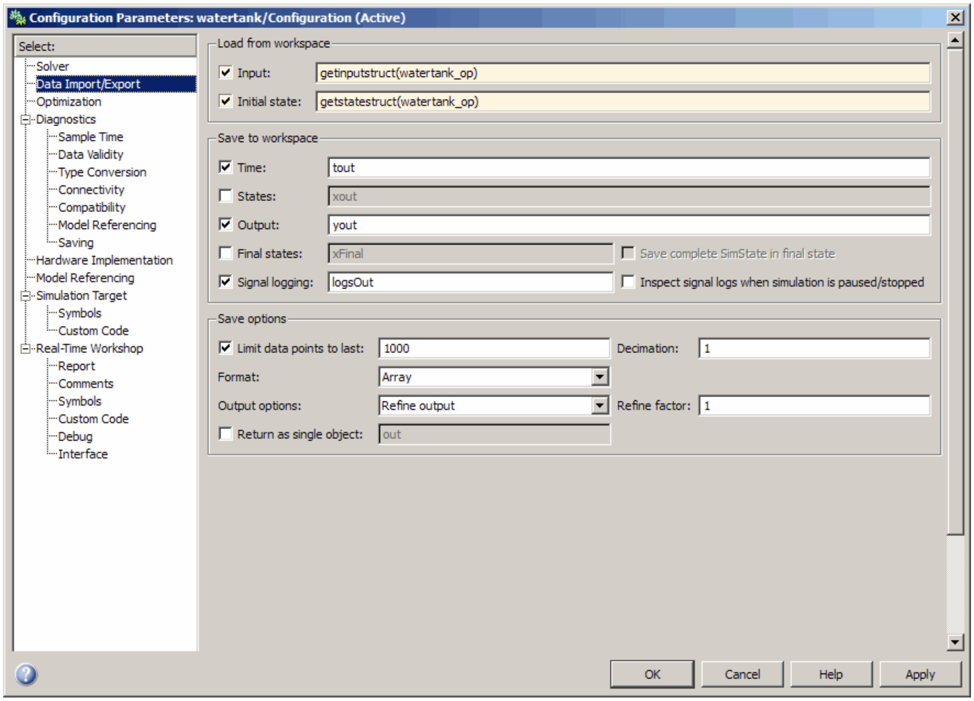
3 In the **Load from workspace** portion of the **Data Import/Export** node, do the following:

- 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`.



Configuration Parameters: watertank/Configuration (Active)

Load from workspace

☑ Input: getinputstruct(watertank_op)

☑ Initial state: getstatestruct(watertank_op)

Save to workspace

☑ Time: tout

□ States: xout

☑ Output: yout

□ Final states: xFinal □ Save complete SimState in final state

☑ Signal logging: logsOut □ Inspect signal logs when simulation is paused/stopped

Save options

☑ Limit data points to last: 1000 Decimation: 1

Format: Array

Output options: Refine output Refine factor: 1

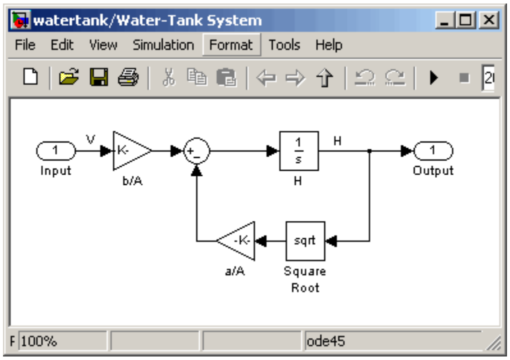
□ Return as single object: out

4 Click **OK**.

Simulating the Initialized Model

To simulate the initialized model:

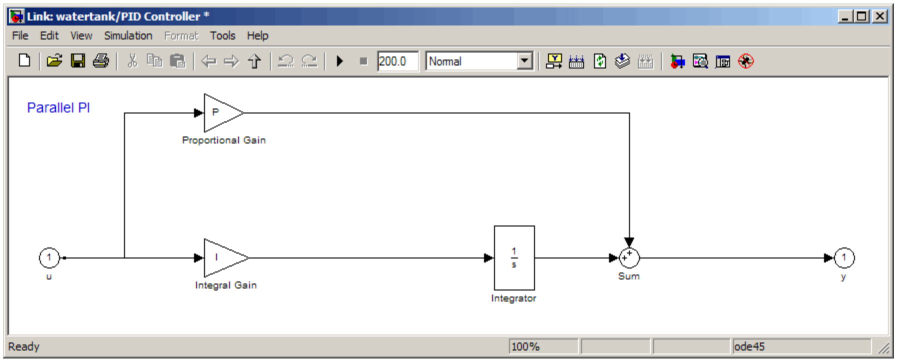
1 In the Water-Tank System subsystem, right-click the output signal of the integrator, which is the state *H*, and select **Create and Connect Viewer > Simulink > Scope**. This action opens a Scope for viewing the state *H*.


watertank/Water-Tank System
File Edit View Simulation Format Tools Help
1
Input
V
K-
b/A
1/s
H
H
1
Output
-K-
a/A
sqrt
Square Root
100%
ode45

2 Add a Scope for viewing the state *Integrator*:

a In the `watertank` model, right-click the PID Controller block and select **Look Under Mask**.

This action opens the PID Controller block mask.



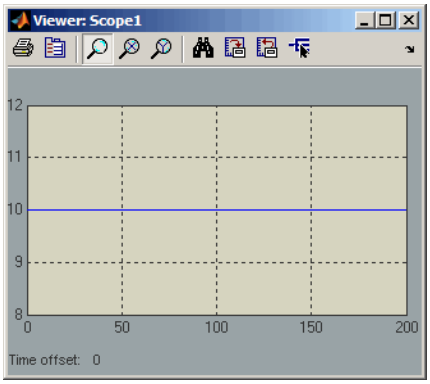
b In the block mask, right-click the output signal of the Integrator, which is the state *Integrator*, and select **Create and Connect Viewer > Simulink > Scope**. This action opens a Scope for viewing the state *Integrator*.

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

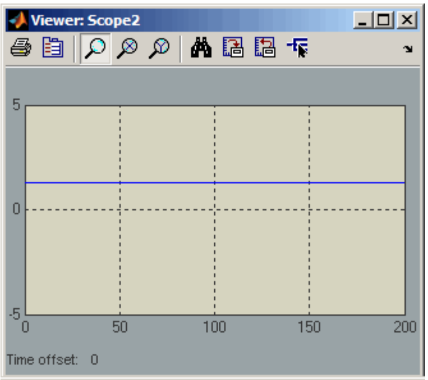
This action displays the states *H* and *Integrator* in their respective Scope windows. The Scope outputs shows these results:

- 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

In this section...

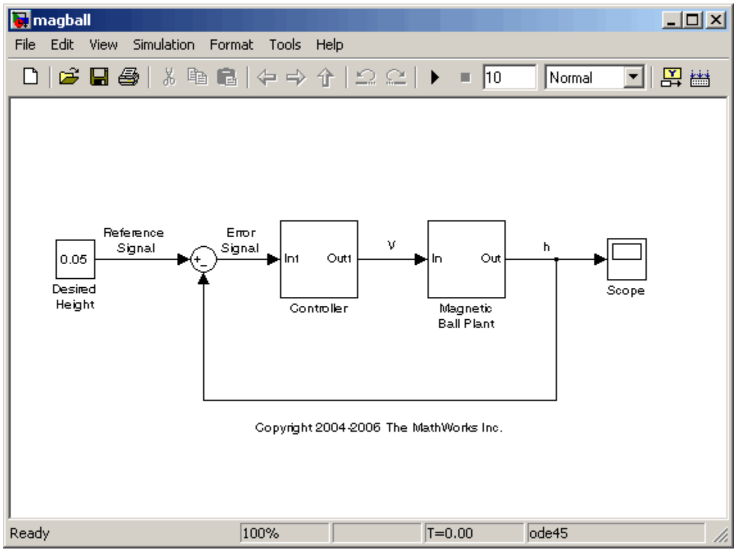
Objectives

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

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

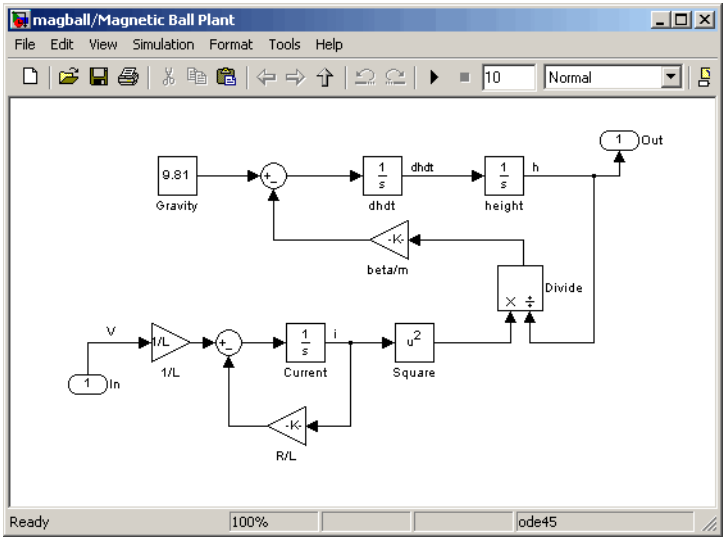


To view the model of the Magnetic Ball Plant subsystem, double-click the corresponding block in the `magball` model. The blocks in this model represent the mathematical system described in "Magnetic Ball Plant Subsystem" on page 4-3.

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

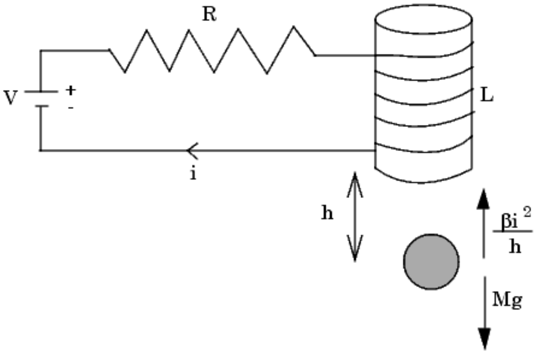
Magnetic Ball Plant Subsystem

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?

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?".

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?".

Overview of the Linearization Process

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

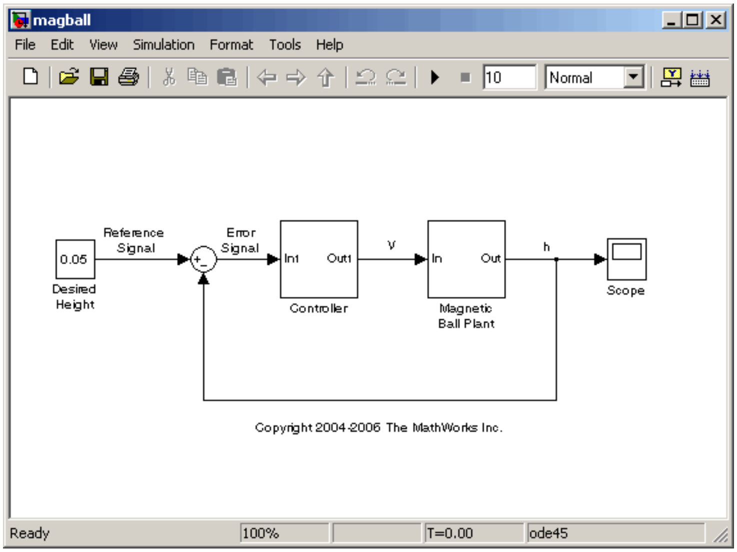
- Defining the portion of the model to linearize, also known as the *linearization path*.
- Removing the effects of a feedback loop in a single-loop control system.
- Linearizing about the existing operating point in the Simulink model.
- Viewing the linearization results in a step response plot and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

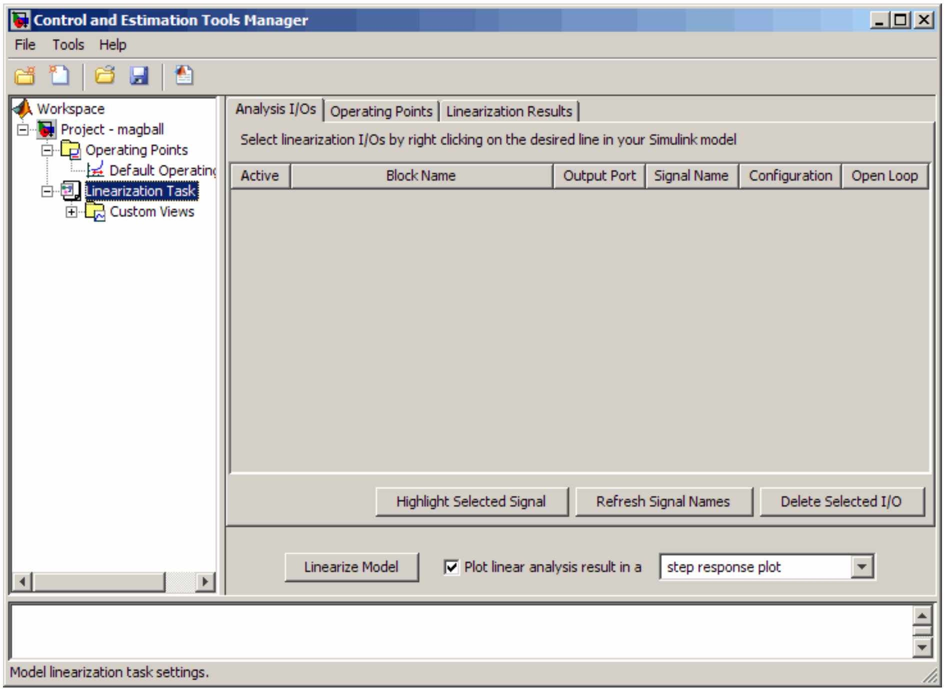
The model opens in Simulink as shown in the following figure.



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)



Select linearization I/Os by right clicking on the desired line in your Simulink model

☑ Plot linear analysis result in a step response plot

Model linearization task settings.

3 Define the portion of the model to linearize.

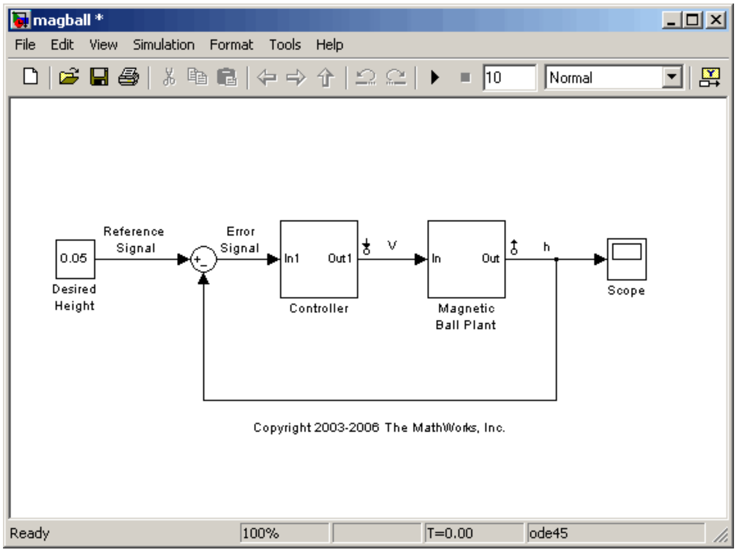
a In the `magball` model, right-click the input signal to the Magnetic Ball Plant subsystem, named `V`, and select **Linearization Points > Input Point**.

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

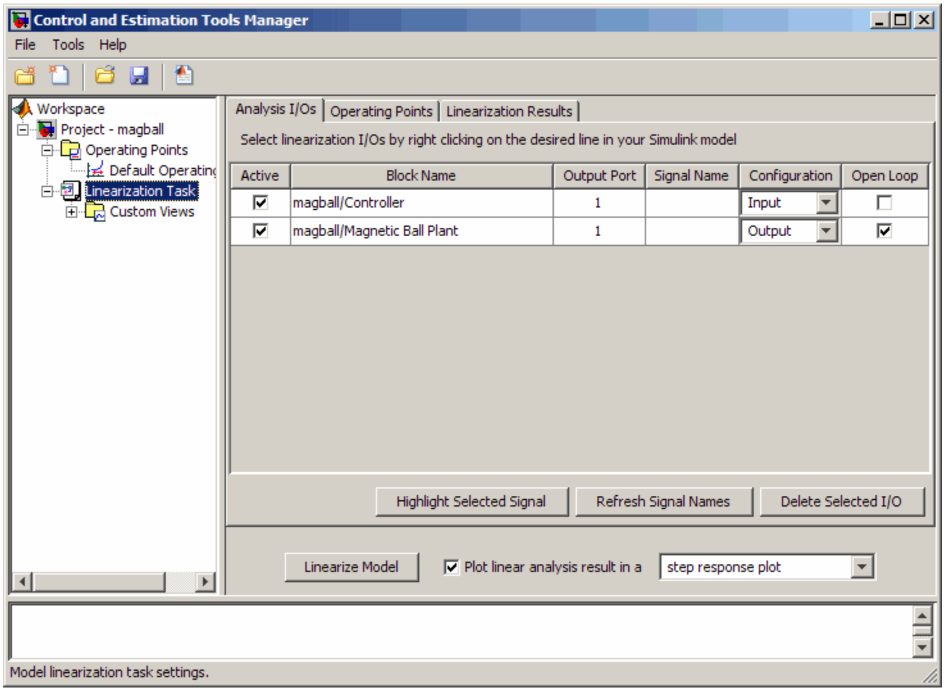
b Right-click the output signal from the Magnetic Ball Plant subsystem, named `h`, and select **Linearization Points > Output Point**.

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 Control and Estimation Tools Manager.



Select linearization I/Os by right clicking on the desired line in your Simulink model

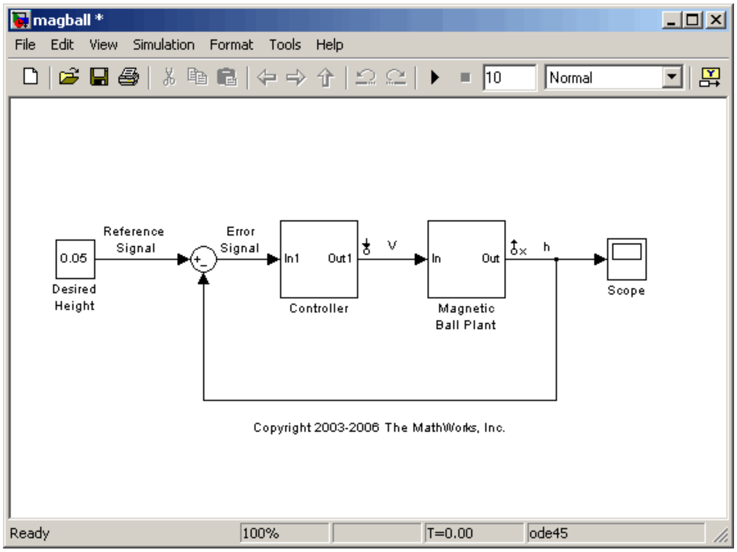
Active	Block Name	Output Port	Signal Name	Configuration	Open Loop
☑	magball/Controller	1		Input	□
☑	magball/Magnetic Ball Plant	1		Output	☑

☑ Plot linear analysis result in a step response plot

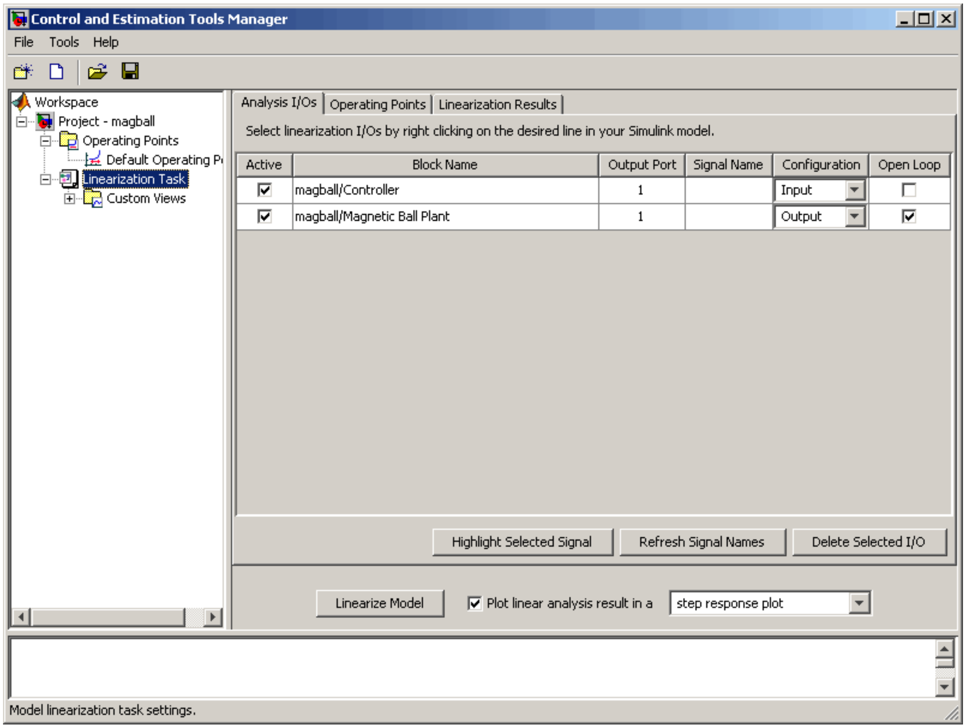
Model linearization task settings.

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”.



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.



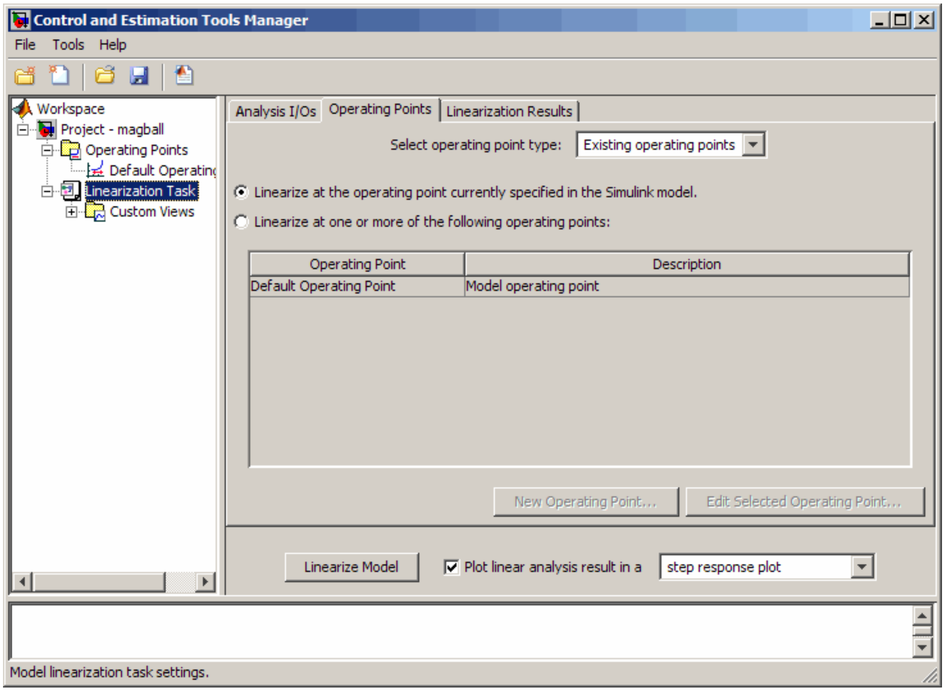
Select linearization I/Os by right clicking on the desired line in your Simulink model.

Active	Block Name	Output Port	Signal Name	Configuration	Open Loop
☑	magball/Controller	1		Input	☐
☑	magball/Magnetic Ball Plant	1		Output	☑

☑ Plot linear analysis result in a step response plot

Model linearization task settings.

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.



Select operating point type: Existing operating points

● Linearize at the operating point currently specified in the Simulink model.

○ Linearize at one or more of the following operating points:

Operating Point	Description
Default Operating Point	Model operating point

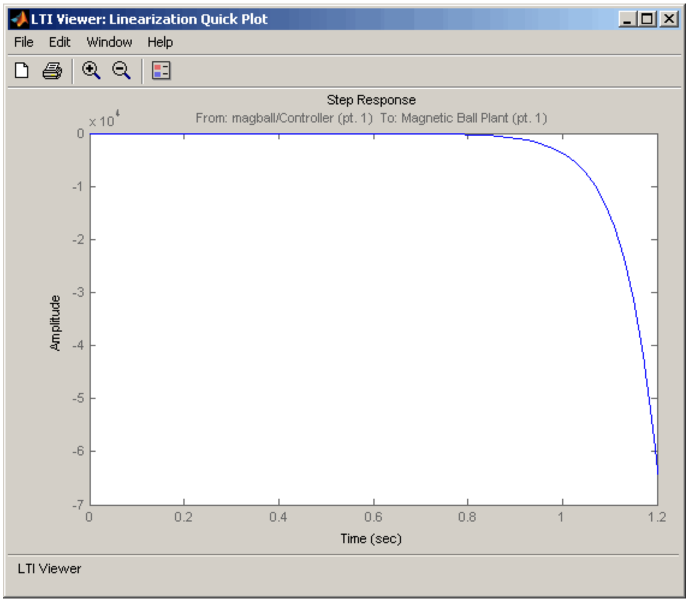
☑ Plot linear analysis result in a step response plot

Model linearization task settings.

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.

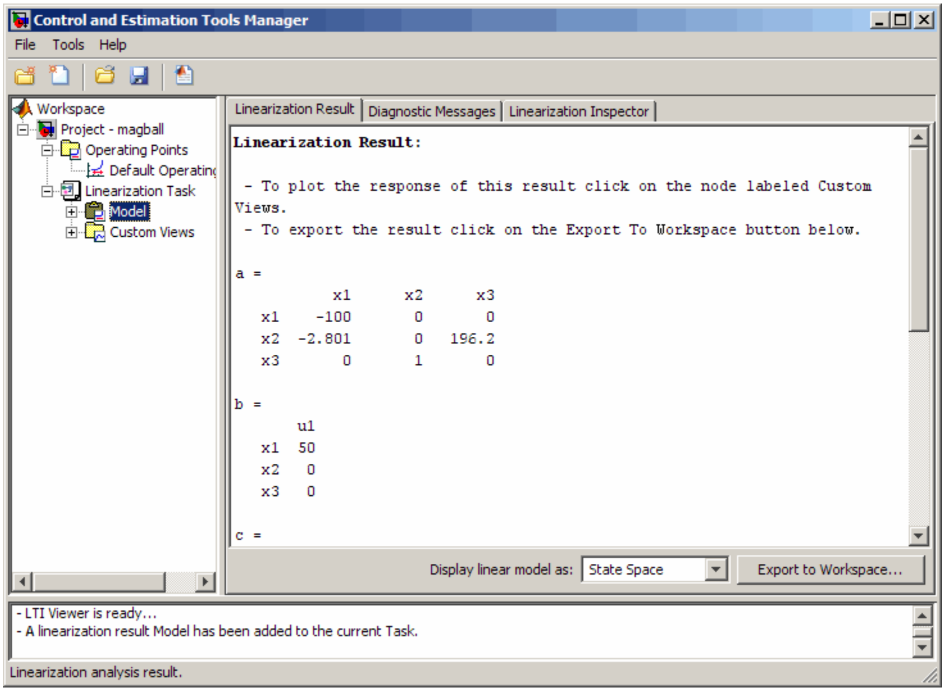
Tip You can use the right-click menu of the LTI Viewer GUI to display a different plot or add characteristics to the response plot, such as peak response and settling time. For more information about working with response plots, see "LTI Viewer" in the *Control System Toolbox Getting Started Guide*.

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

Tip To automatically generate MATLAB code that linearizes your model as specified in the Control and Estimation Tools Manager, click or select **File > Generate MATLAB Code**.

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.



```
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 =
       u1
   x1  50
   x2   0
   x3   0

c =
```

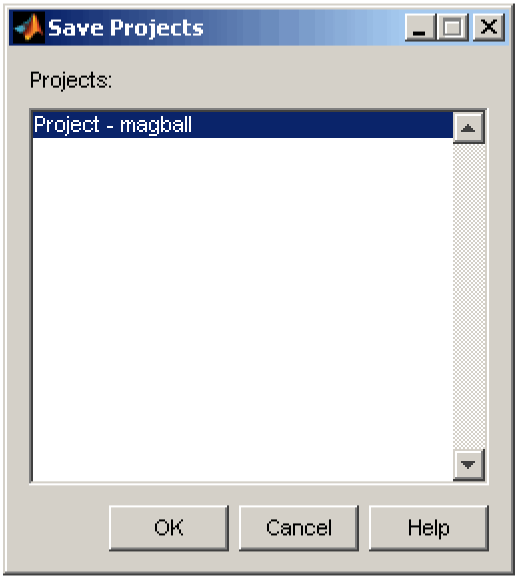
```
- LTI Viewer is ready...
- A linearization result Model has been added to the current Task.
Linearization analysis result.
```

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.

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

About This Tutorial

In this section...

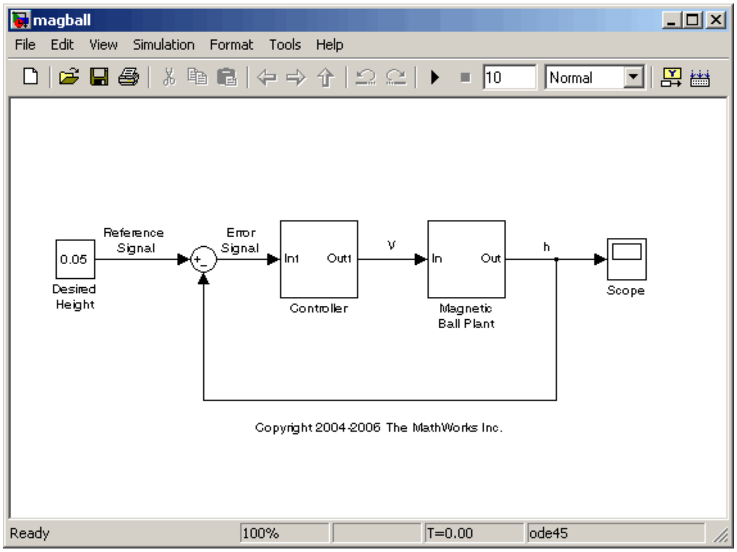
Objectives

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

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

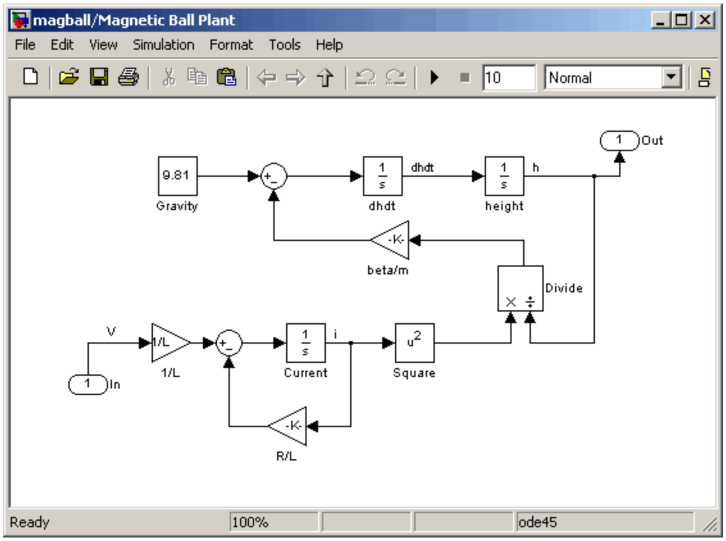


To view the model of the Magnetic Ball Plant subsystem, double-click the corresponding block in the `magball` model. The blocks in this model represent the mathematical system described in "Magnetic Ball Plant Subsystem" on page 5-3.

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

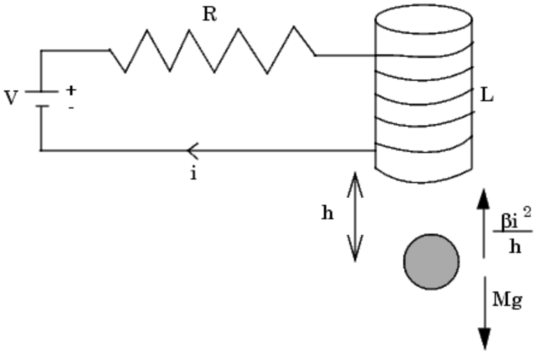
Magnetic Ball Plant Subsystem

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?

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?".

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?".

Overview of the Linearization Process

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

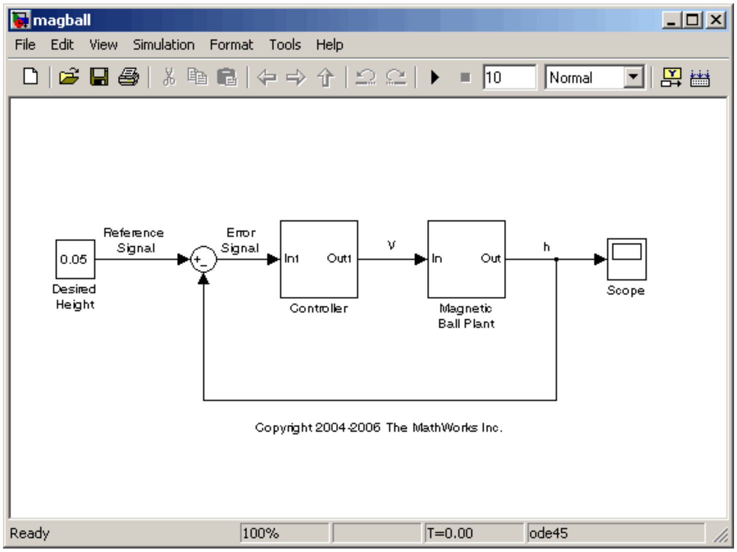
- Defining the portion of the model to linearize, also known as the *linearization path*.
- Removing the effects of a feedback loop in a single-loop control system.
- Linearizing about the existing operating point in the Simulink model.
- Viewing the linearization results in a step response plot and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

```
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)
   ```

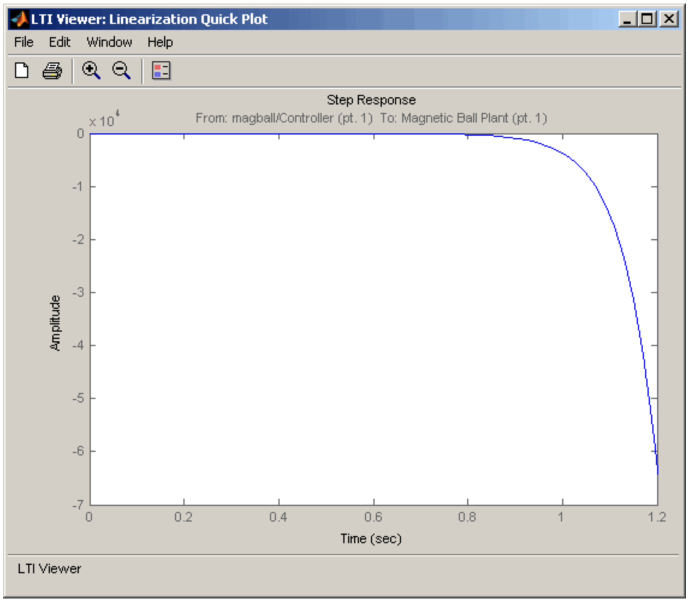
   This command linearizes the portion of the model defined in the `magball_io` object about the operating point currently specified in the model. This command returns the following linearization result as a state-space object.

   ```
   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 "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.

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

About This Tutorial

In this section...

Objectives

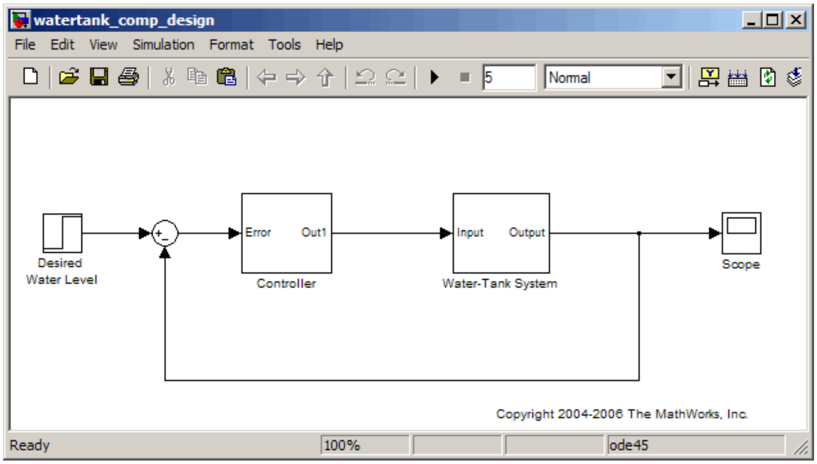
In this tutorial, you learn how to use the Simulink Control Design GUI to design a PID compensator for a single-loop feedback system that is operating at the operating conditions specified in the Simulink model. You accomplish the following tasks:

- 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

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.



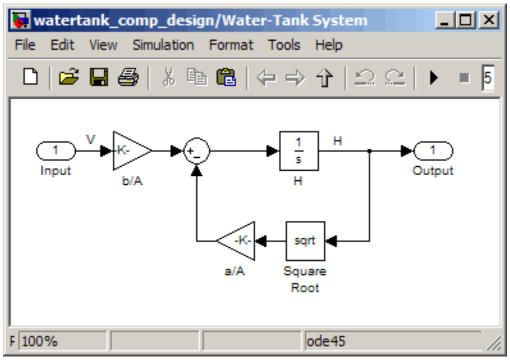
To view the Water-Tank System and the Controller, double-click the corresponding subsystem in the watertank_comp_design model. For descriptions of these subsystems, see the following topics:

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

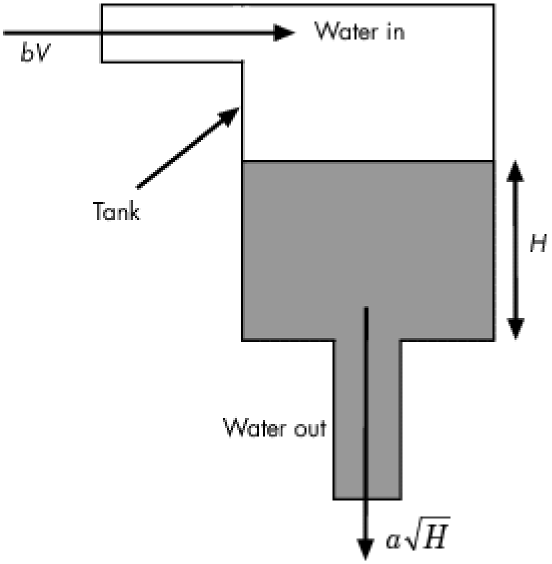
For information about creating Simulink models, see "Creating a Simulink Model".

Water-Tank Subsystem

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


watertank_comp_design/Water-Tank System
File Edit View Simulation Format Tools Help
1
Input
V
-K-
b/A
+
-
1
s
H
H
1
Output
-K-
a/A
sqrt
Square Root
100%
ode45

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



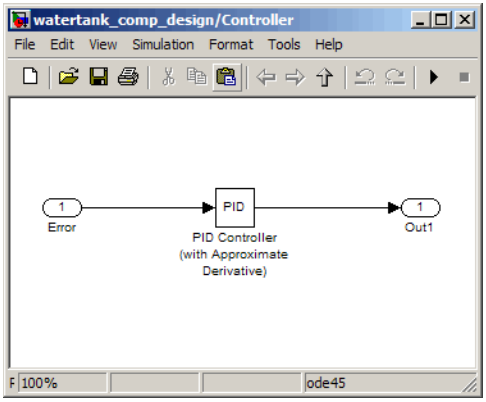
Water enters the tank from the top at a rate proportional to the voltage, V, applied to the pump. The water leaves through an opening in the tank base at a rate that is proportional to the square root of the water height, H, in the tank. The presence of the square root in the water flow rate results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


watertank_comp_design/Controller
File Edit View Simulation Format Tools Help
1
Error
PID
PID Controller
(with Approximate
Derivative)
1
Out1
100%
ode45

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?” and “Why Are Operating Points Important?”

Designing a PID Compensator Using the Ziegler-Nichols Tuning Algorithm

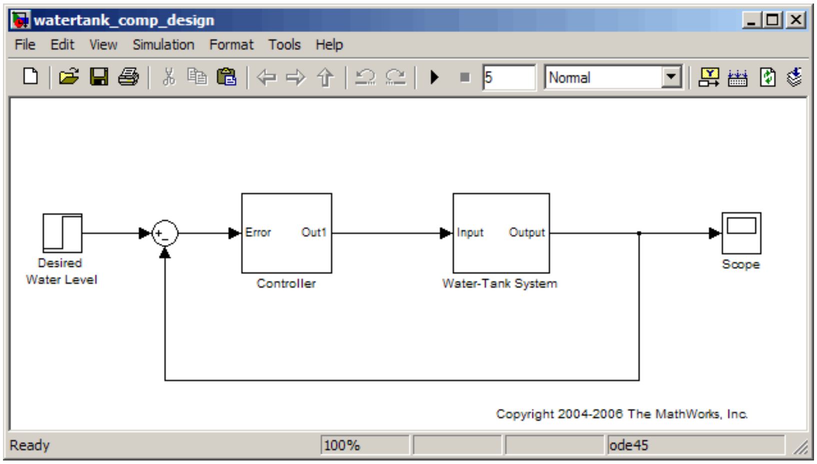
In this portion of the tutorial, you design a compensator using the automated PID Ziegler-Nichols open-loop tuning algorithm. This tuning method computes the proportional, integral, and derivative gains using the Chien-Hrones-Resnick (CHR) setting with a 20% overshoot.

To design a PID compensator:

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

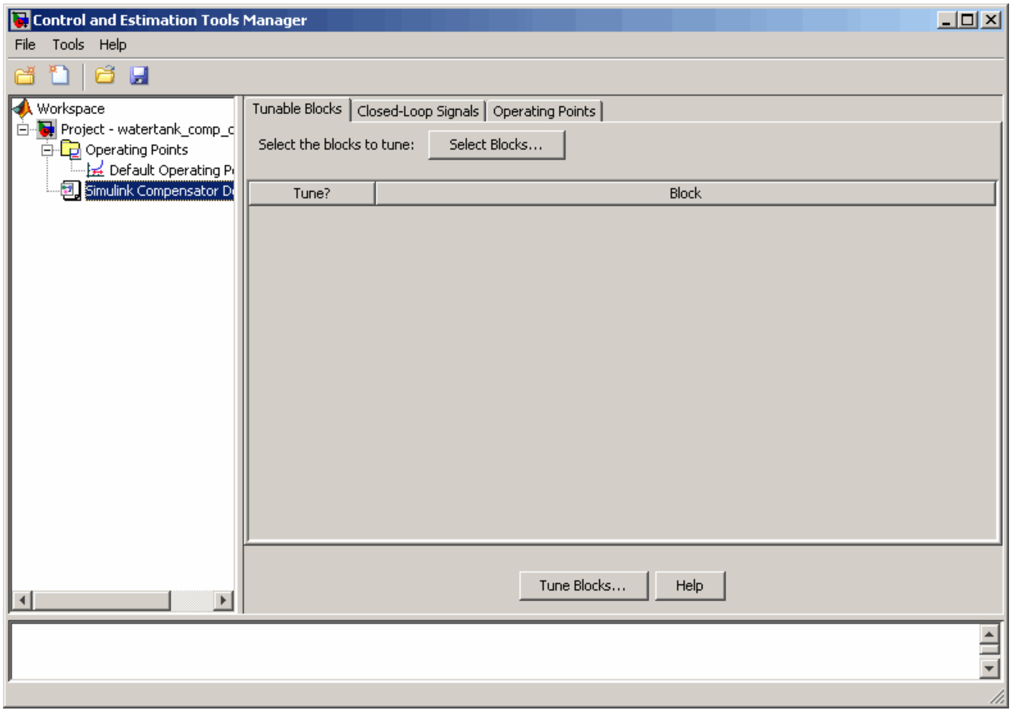
```
watertank_comp_design
```

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.



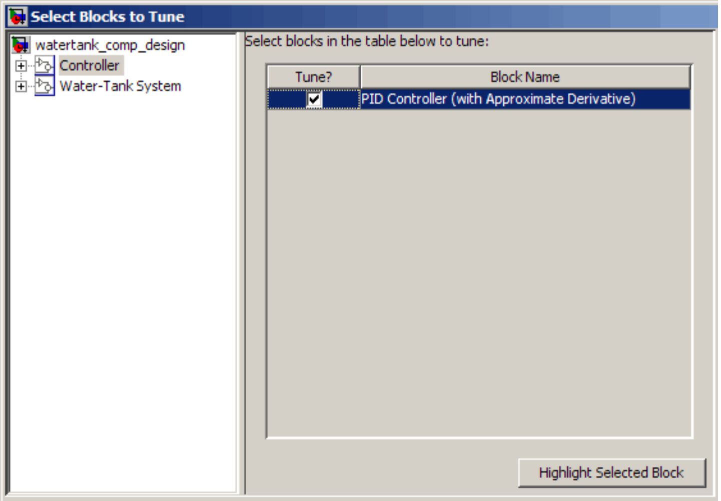
3 Select the PID Controller block as the block to tune.

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)**.



Select blocks in the table below to tune:

Tune?	Block Name
☑	PID Controller (with Approximate Derivative)

d Click **OK**.

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

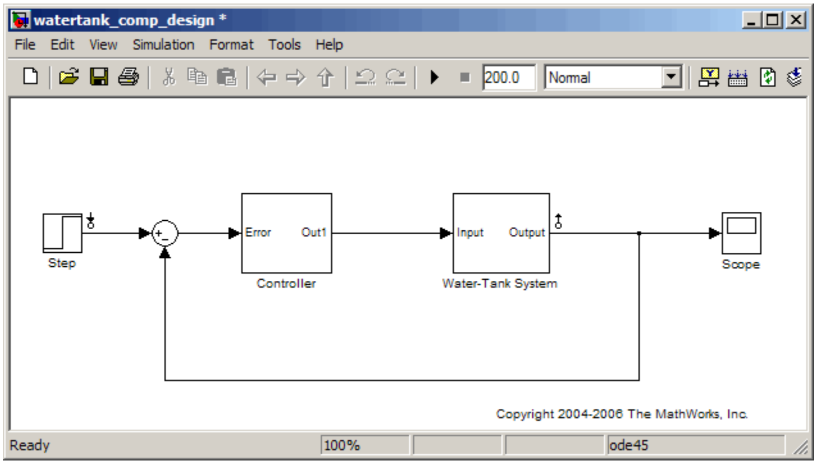
a In the `watertank_comp_design` model, right-click the output of the Desired Water Level block, and select **Linearization Points > Input Point**.

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.

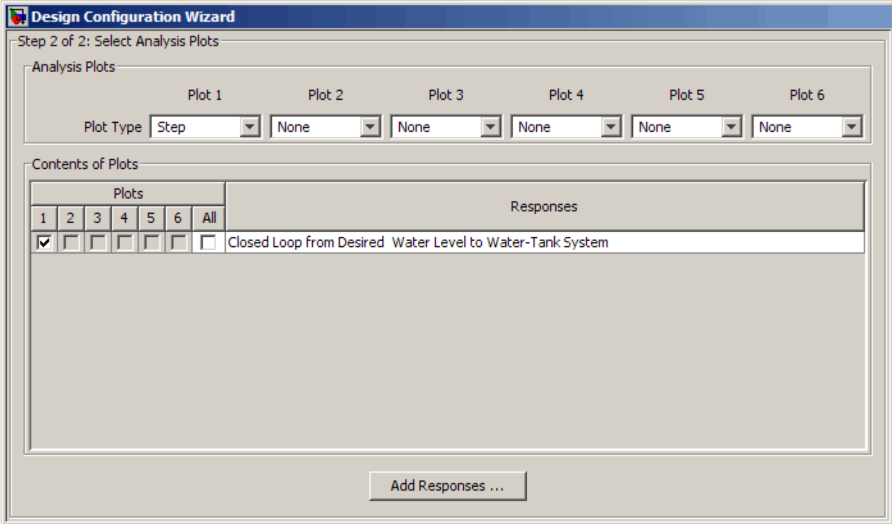


5 In the Control and Estimation Tools Manager, click **Tune Blocks** to open the Design Configuration Wizard. Click **Next**.

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**.



Design Configuration Wizard

Step 2 of 2: Select Analysis Plots

Analysis Plots

	Plot 1	Plot 2	Plot 3	Plot 4	Plot 5	Plot 6
Plot Type	Step	None	None	None	None	None

Contents of Plots

Plots							Responses
1	2	3	4	5	6	All	
☑	□	□	□	□	□	□	Closed Loop from Desired Water Level to Water-Tank System

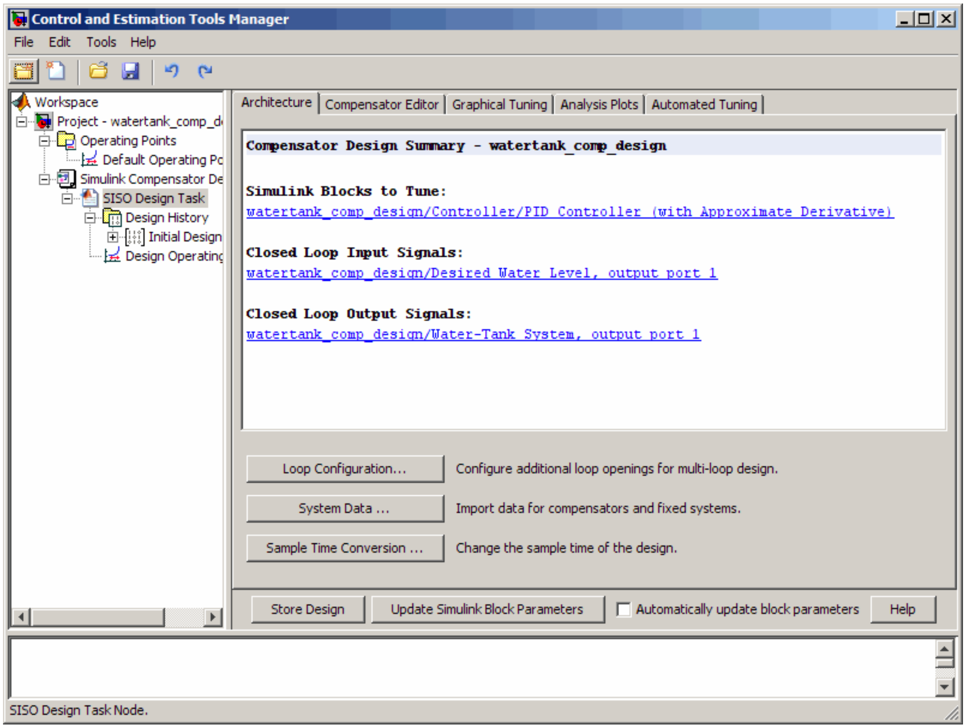
8 Click **Finish**.

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.



```
Compensator Design Summary - watertank_comp_design

Simulink Blocks to Tune:
watertank_comp_design/Controller/PID Controller (with Approximate Derivative)

Closed Loop Input Signals:
watertank_comp_design/Desired Water Level, output port 1

Closed Loop Output Signals:
watertank_comp_design/Water-Tank System, output port 1
```

Loop Configuration... Configure additional loop openings for multi-loop design.

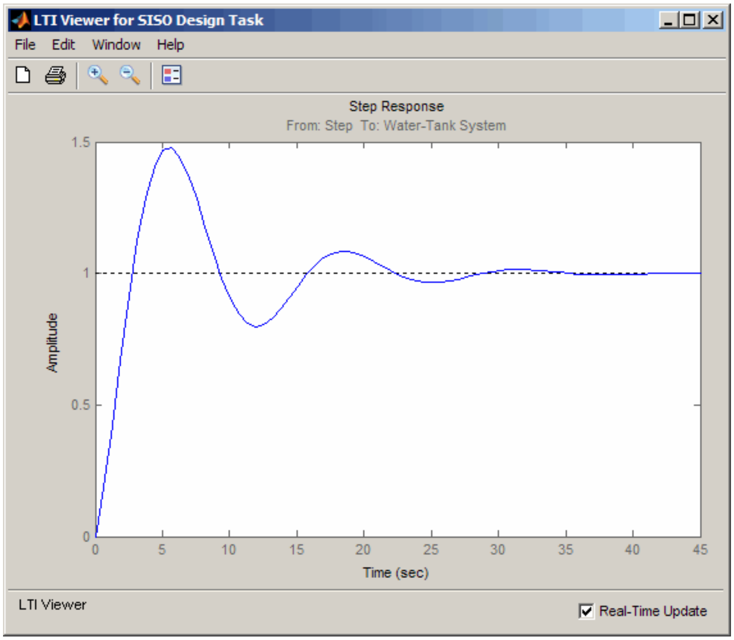
System Data ... Import data for compensators and fixed systems.

Sample Time Conversion ... Change the sample time of the design.

□ Automatically update block parameters

SISO Design Task Node.

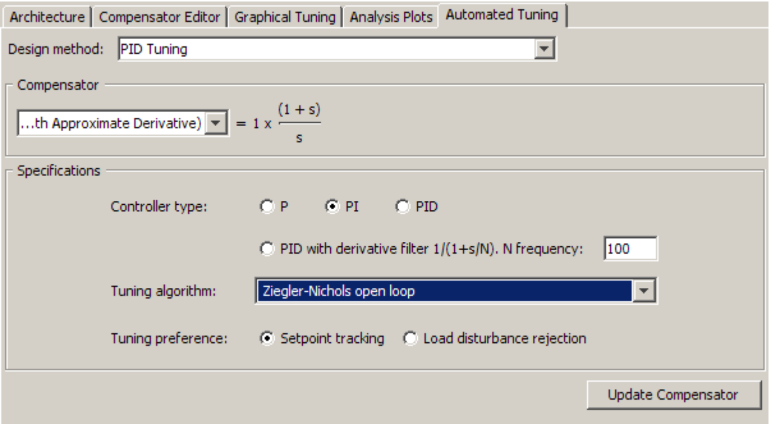
The Step Response plot shows an overshoot that does not meet the overshoot design requirement of less than 5%.



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`



Design method: PID Tuning

...th Approximate Derivative) $= 1 \times \frac{(1 + s)}{s}$

Controller type: ○ P ● PI ○ PID

○ PID with derivative filter 1/(1+s/N). N frequency: 100

Tuning algorithm: Ziegler-Nichols open loop

Tuning preference: ● Setpoint tracking ○ Load disturbance rejection

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**

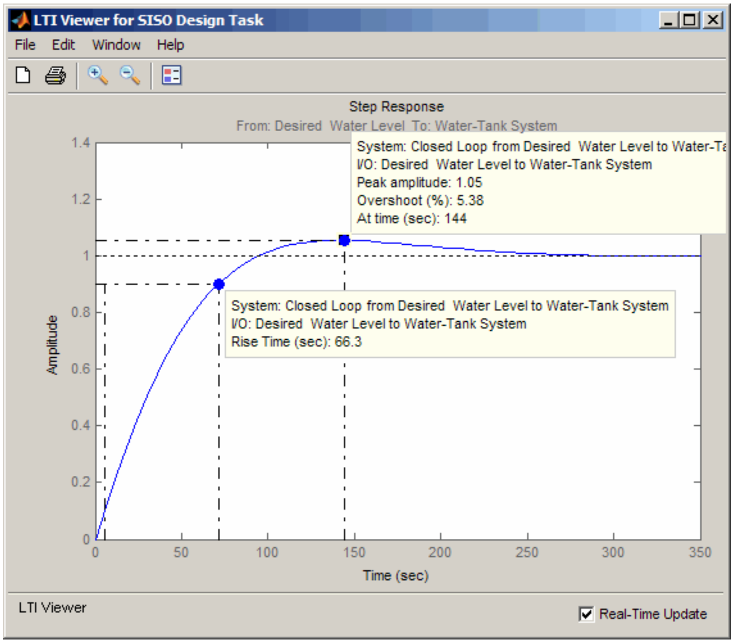
These actions add a plot marker to the plot for each characteristic, shown as blue dots.

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.


LTI Viewer for SISO Design Task
File
Edit
Window
Help
Step Response
From: Desired Water Level To: Water-Tank System
System: Closed Loop from Desired Water Level to Water-Ta
I/O: Desired Water Level to Water-Tank System
Peak amplitude: 1.05
Overshoot (%): 5.38
At time (sec): 144
System: Closed Loop from Desired Water Level to Water-Tank System
I/O: Desired Water Level to Water-Tank System
Rise Time (sec): 66.3
Amplitude
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
250
300
350
Time (sec)
LTI Viewer
☑ Real-Time Update

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.

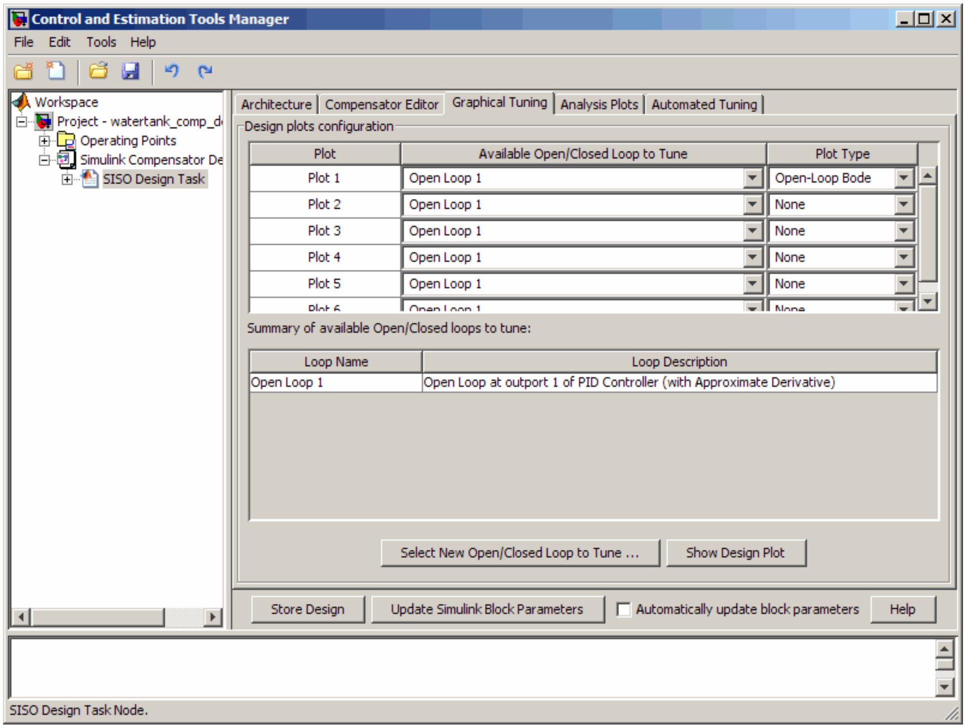
Tuning the PID Compensator Using Bode Graphical Tuning

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**.



Design plots configuration

Plot	Available Open/Closed Loop to Tune	Plot Type
Plot 1	Open Loop 1	Open-Loop Bode
Plot 2	Open Loop 1	None
Plot 3	Open Loop 1	None
Plot 4	Open Loop 1	None
Plot 5	Open Loop 1	None
Plot 6	Open Loop 1	None

Summary of available Open/Closed loops to tune:

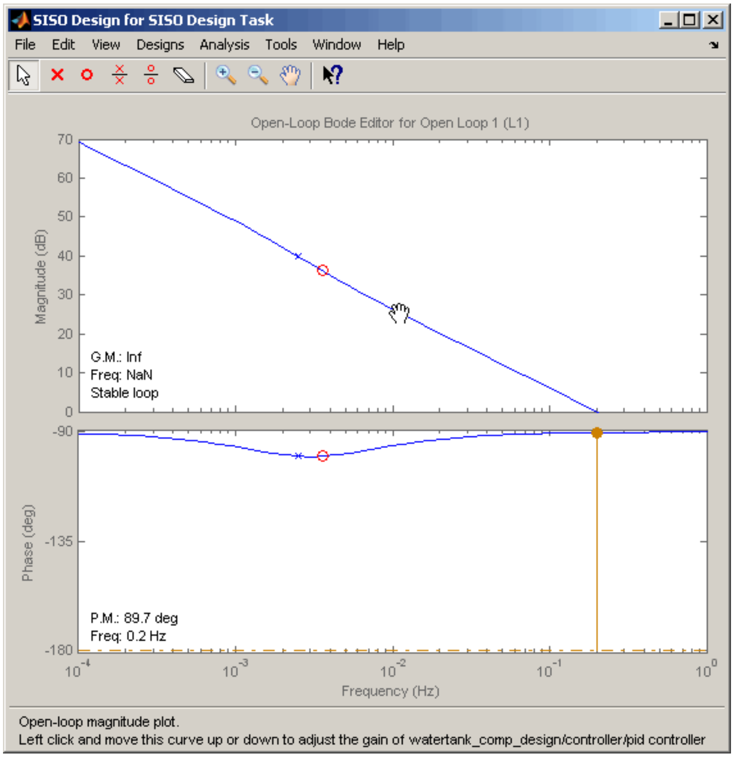
Loop Name	Loop Description
Open Loop 1	Open Loop at outport 1 of PID Controller (with Approximate Derivative)

☐ Automatically update block parameters

SISO Design Task Node.

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.



G.M.: Inf
Freq: NaN
Stable loop

P.M.: 89.7 deg
Freq: 0.2 Hz

Open-loop magnitude plot.
Left click and move this curve up or down to adjust the gain of watertank_comp_design/controller/pid controller

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:

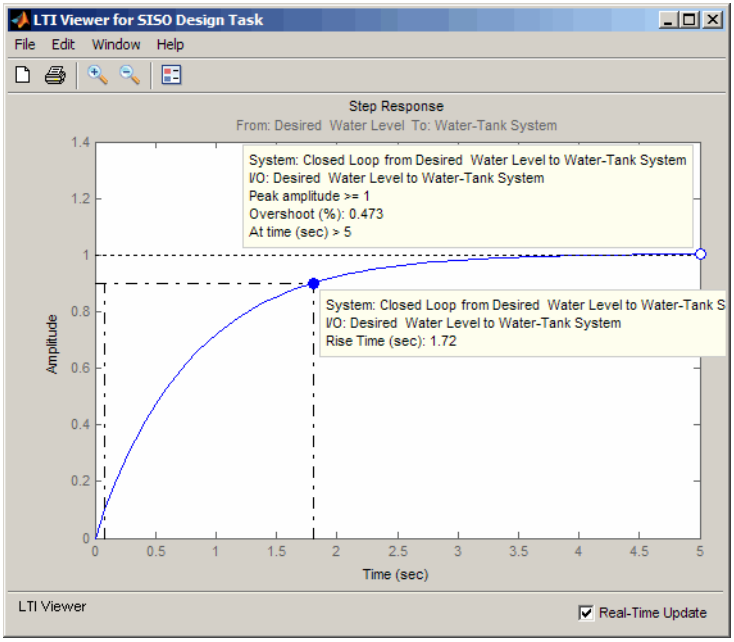
- **Characteristics > Peak Response**
- **Characteristics > Rise Time**

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.



System: Closed Loop from Desired Water Level to Water-Tank System
I/O: Desired Water Level to Water-Tank System
Peak amplitude >= 1
Overshoot (%): 0.473
At time (sec) > 5

System: Closed Loop from Desired Water Level to Water-Tank S
I/O: Desired Water Level to Water-Tank System
Rise Time (sec): 1.72

☑ Real-Time Update

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.

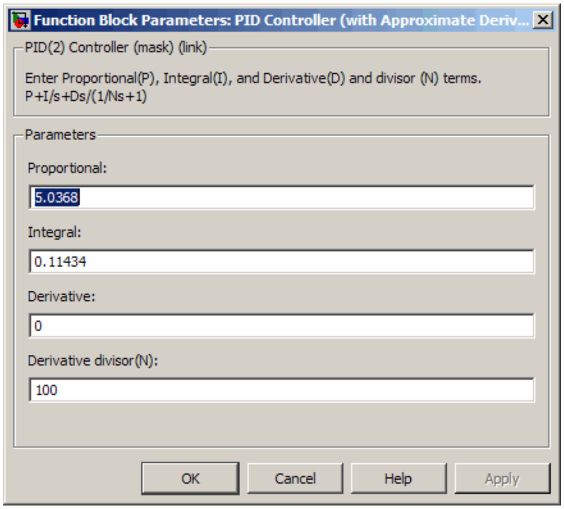
You must have already designed the compensator, as described in "Tuning the PID Compensator Using Bode Graphical Tuning" on page 6-17.

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.



Function Block Parameters: PID Controller (with Approximate Deriv...

PID(2) Controller (mask) (link)

Enter Proportional(P), Integral(I), and Derivative(D) and divisor (N) terms.
P+I/s+Ds/(1/Ns+1)

Parameters

Proportional: 5.0368

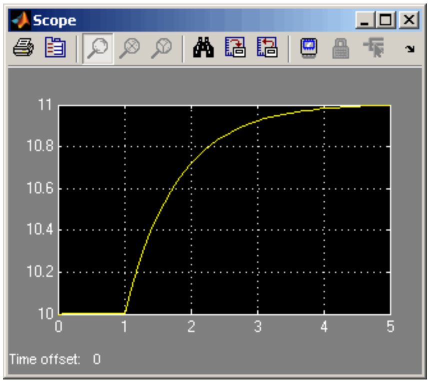
Integral: 0.11434

Derivative: 0

Derivative divisor(N): 100

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.

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”

Index