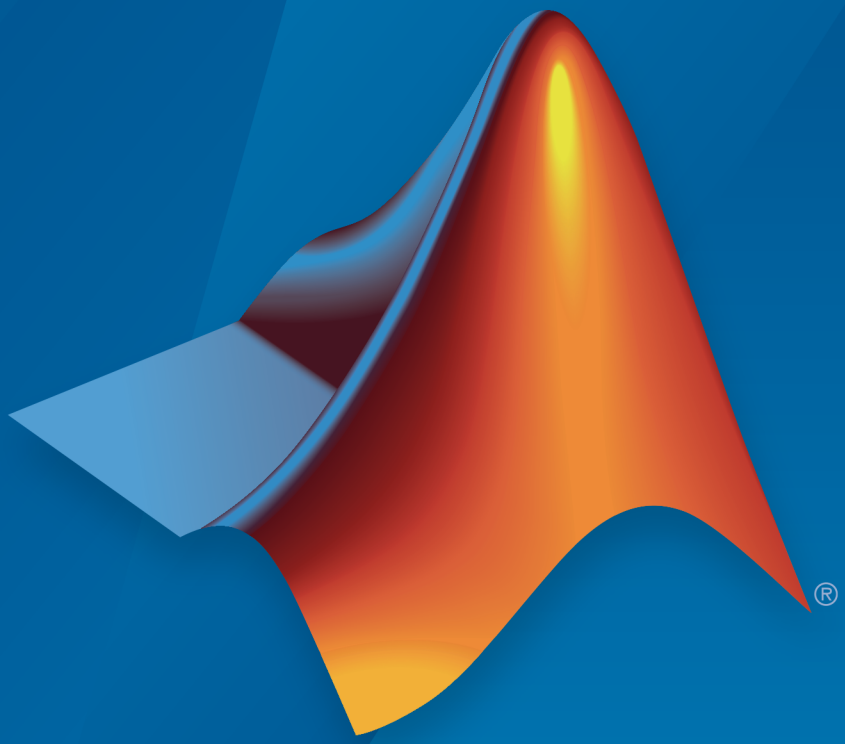
Simulink® Control Design™

Getting Started Guide



MATLAB® & SIMULINK®

R2015a

MathWorks®

How to Contact MathWorks

Latest news: www.mathworks.com

Sales and services: www.mathworks.com/sales_and_services

User community: www.mathworks.com/matlabcentral

Technical support: www.mathworks.com/support/contact_us

Phone: 508-647-7000

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

Simulink® Control Design™ Getting Started Guide

© COPYRIGHT 2004–2015 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

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	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)
March 2010	Online only	Revised for Version 3.1 (Release 2010a)
September 2010	Online only	Revised for Version 3.2 (Release 2010b)
April 2011	Online only	Revised for Version 3.3 (Release 2011a)
September 2011	Online only	Revised for Version 3.4 (Release 2011b)
March 2012	Online only	Revised for Version 3.5 (Release 2012a)
September 2012	Online only	Revised for Version 3.6 (Release 2012b)
March 2013	Online only	Revised for Version 3.7 (Release 2013a)
September 2013	Online only	Revised for Version 3.8 (Release 2013b)
March 2014	Online only	Revised for Version 4.0 (Release 2014a)
October 2014	Online only	Revised for Version 4.1 (Release 2014b)
March 2015	Online only	Revised for Version 4.2 (Release 2015a)

Contents

1 Product Overview

Simulink Control Design Product Description 1-2
Key Features 1-2

2 Steady-State Operating Points

What Is a Steady-State Operating Point? 2-2

Steady-State Operating Points (Trimming) From Specifications 2-3

magball Simulink Model 2-11

3 Linearization

Applications of Linearization 3-2

Open-Loop Response of Control System for Stability Margin Analysis 3-3

Bode Response of Simulink Model 3-7

watertank Simulink Model 3-11

4 PID Control Design

PID Controller Tuning in Simulink **4-2**

Design a Controller Using Automated Tuning and Bode Graphical Design **4-11**
About This Tutorial **4-11**
PID Control Design Using Robust-Response-Time Tuning Algorithm **4-15**
PID Control Design Using Bode Graphical Tuning **4-23**
Closed-Loop Simulation of Simulink Model **4-27**

1

Product Overview

Simulink Control Design Product Description

Linearize models and design control systems

Simulink® Control Design™ lets you design and analyze plants and control systems modeled in Simulink and automatically tune PID controller gains to meet performance requirements. You can also find operating points and compute exact linearizations of Simulink models at various operating conditions. Simulink Control Design provides tools that let you compute simulation-based frequency responses without modifying your model. A graphical interface lets you design and analyze arbitrary control structures modeled in Simulink, including cascaded, prefilter, regulation, and multiloop architectures.

Key Features

- Automatic tuning of PID Controller blocks from the Simulink library
- Nonintrusive operating point calculation (trimming) and linearization of Simulink models
- Simulation-based computation of a Simulink model's frequency response
- Graphical and automated tuning of arbitrary control systems within Simulink
- Numerical optimization of compensators to meet time-domain and frequency-domain requirements (with Simulink Design Optimization™)
- MATLAB® functions for developing automated linearization scripts and performing batch linearization

Steady-State Operating Points

- "What Is a Steady-State Operating Point?" on page 2-2
- "Steady-State Operating Points (Trimming) From Specifications" on page 2-3
- "magball Simulink Model" on page 2-11

What Is a Steady-State Operating Point?

A *steady-state operating point* of the model, also called equilibrium or *trim* condition, includes state variables that do not change with time.

A model might have several steady-state operating points. For example, a hanging pendulum has two steady-state operating points. A *stable steady-state operating point* occurs when a pendulum hangs straight down. That is, the pendulum position does not change with time. When the pendulum position deviates slightly, the pendulum always returns to equilibrium; small changes in the operating point do not cause the system to leave the region of good approximation around the equilibrium value.

An *unstable steady-state operating point* occurs when a pendulum points upward. As long as the pendulum points *exactly* upward, it remains in equilibrium. However, when the pendulum deviates slightly from this position, it swings downward and the operating point leaves the region around the equilibrium value.

When using optimization search to compute operating points for a nonlinear system, your initial guesses for the states and input levels must be in the neighborhood of the desired operating point to ensure convergence.

When linearizing a model with multiple steady-state operating points, it is important to have the right operating point. For example, linearizing a pendulum model around the stable steady-state operating point produces a stable linear model, whereas linearizing around the unstable steady-state operating point produces an unstable linear model.

Examples and How To

- “Steady-State Operating Points (Trimming) From Specifications” on page 2-3
- “Compute Operating Points at Simulation Snapshots”

More About

- “Steady-State Operating Point (Trimming)”
- “Simulink Model States Included in Operating Point Object”

Steady-State Operating Points (Trimming) From Specifications

This example shows how to compute a steady-state operating point, or equilibrium operating point, by specifying known (fixed) equilibrium states and minimum state values.

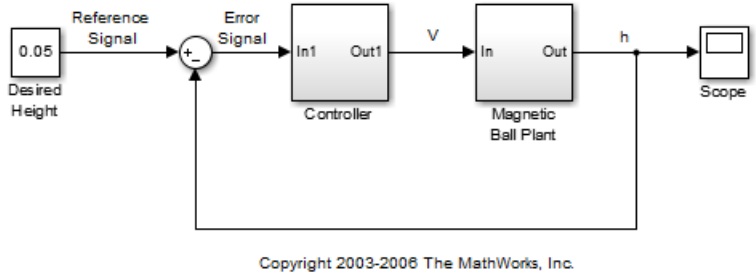
This example finds an operating point of a magnetic ball model at which the height of a levitating magnetic ball remains stable at a desired height of 0.05 m.

Code Alternative

Use `findop` to find operating point from specifications. For examples and additional information, see the `findop` reference page. Finding a steady-state operating point is also known as *trimming*.

1 Open the Simulink model.

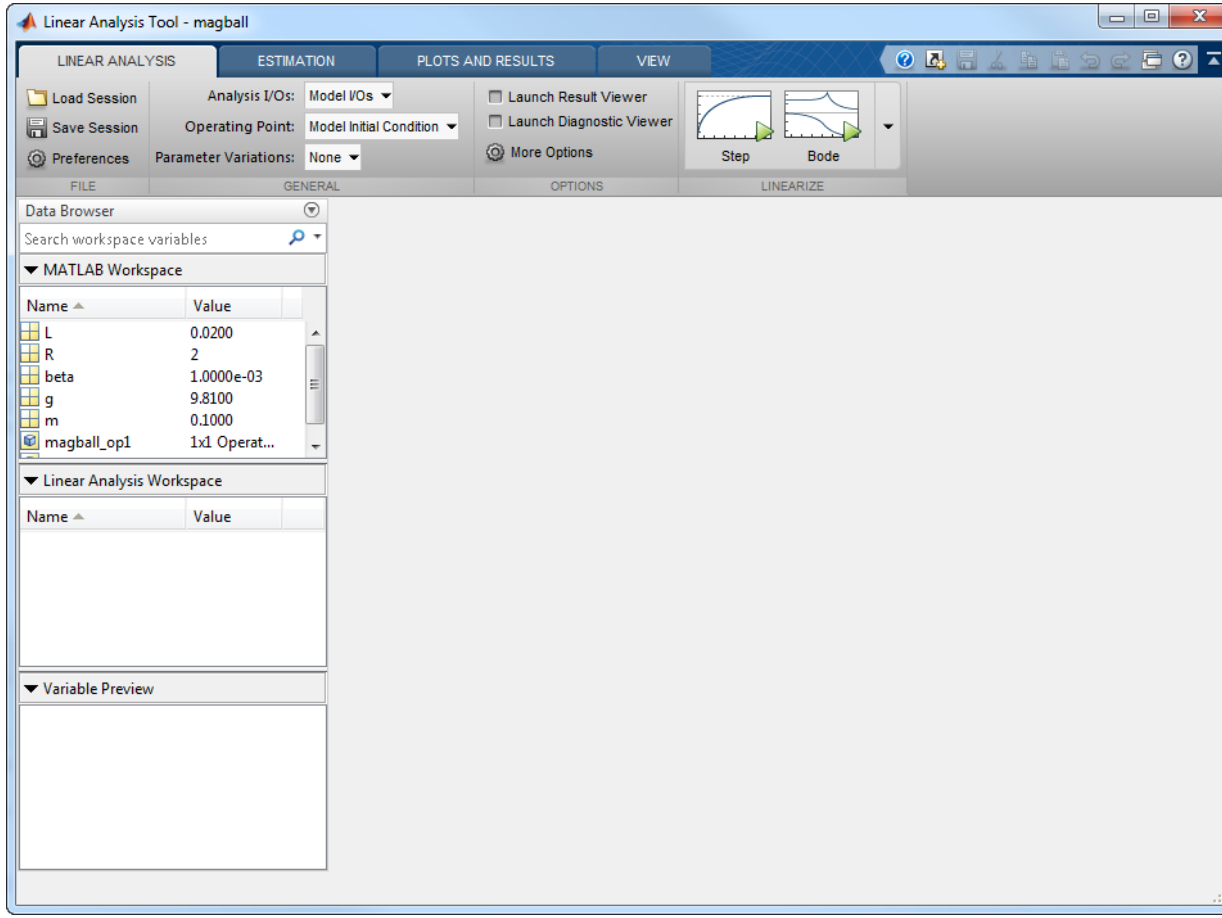
```
sys = 'magball';
open_system(sys)
```



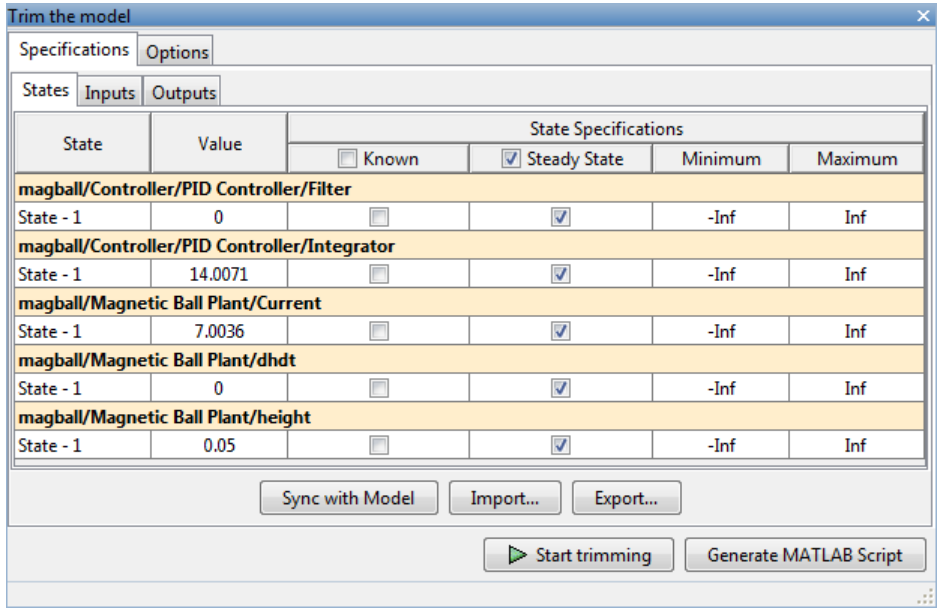
In this model, the height of the magnetic ball is represent by the plant output, h. Trim the model to find a steady state operating point at which $h = 0.05$.

2 In the Simulink Editor, select **Analysis > Control Design > Linear Analysis**.

The Linear Analysis Tool for the model opens.



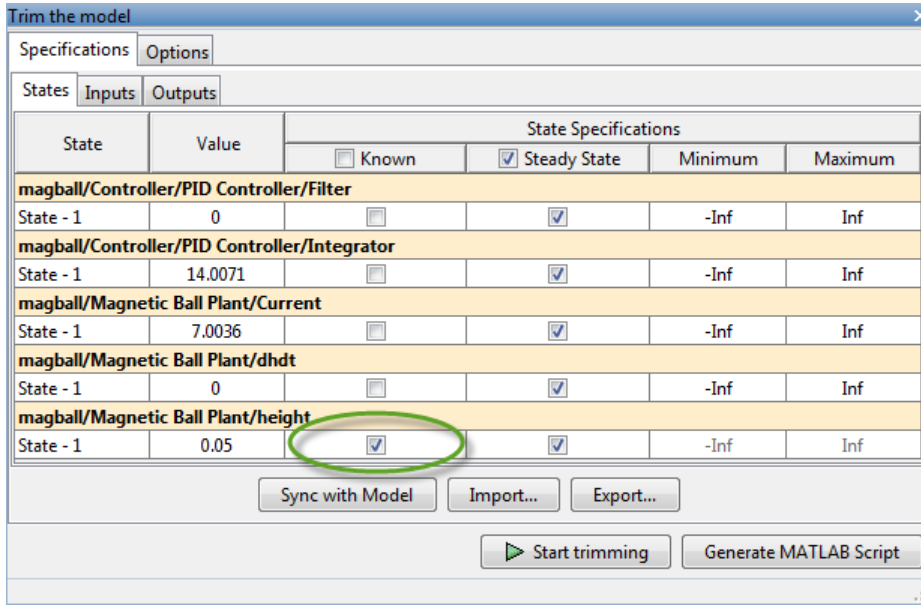
3 In the Linear Analysis Tool, in the **Operating Point** drop-down list, select `Trim Model`.



By default, in the **States** tab, the software specifies all model states to be at equilibrium, as shown by the check marks in the **Steady State** column. The **Inputs** and **Outputs** tabs are empty because this model does not have root-level input and output ports.

4 Specify a fixed height for the magnetic ball.

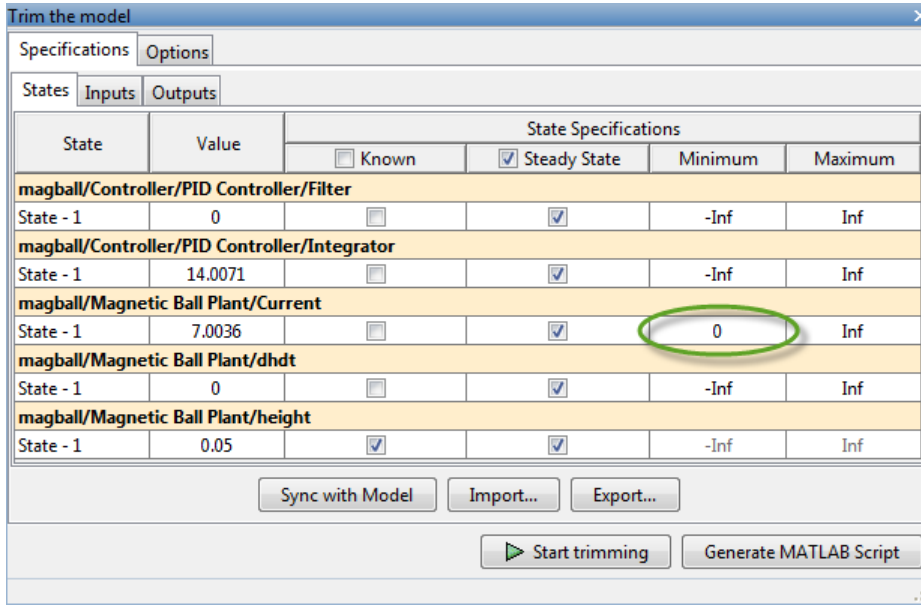
In the **States** tab, select **Known** for the **height** state.



The height of the ball matches the reference signal height (specified in the `Desired Height` block as `0.05`). Since it is known value, the height remains fixed during optimization.

5 Limit the plant current to positive values.

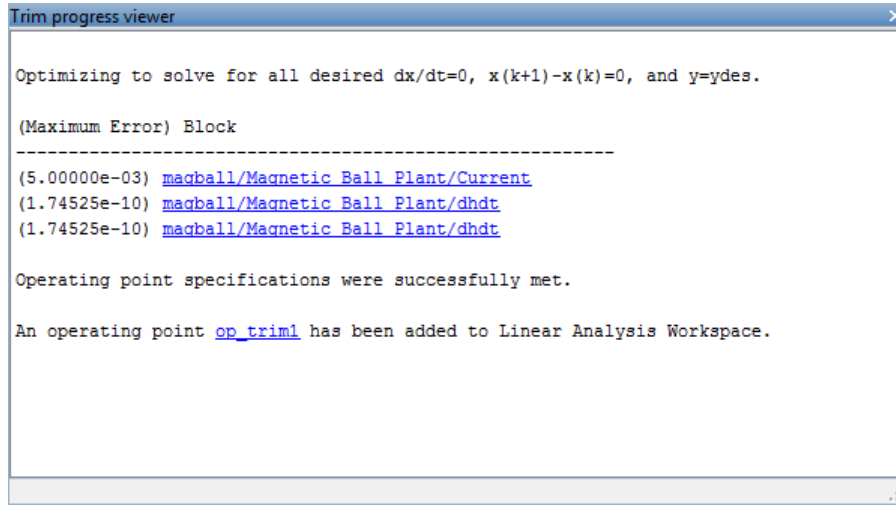
Enter `0` for the **Minimum** bound of the **Current** state.



Since a positive current is required to raise the height of the ball, setting he lower bound to 0 limits the optimization solution to the plant operating range.

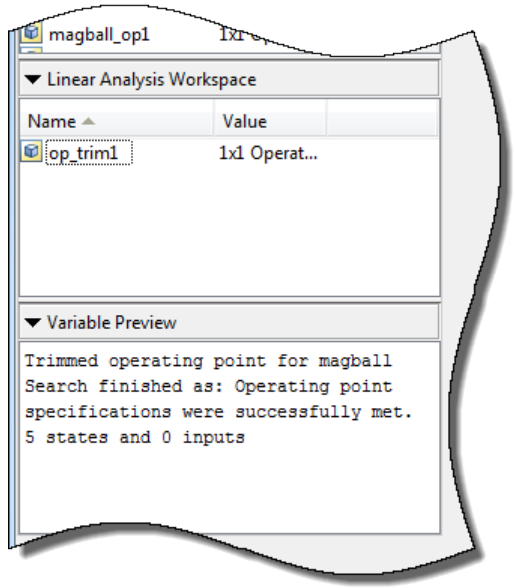
6 Click **Start trimming** to compute the operating point.

The software uses numerical optimization to find the operating point that meets your specifications.

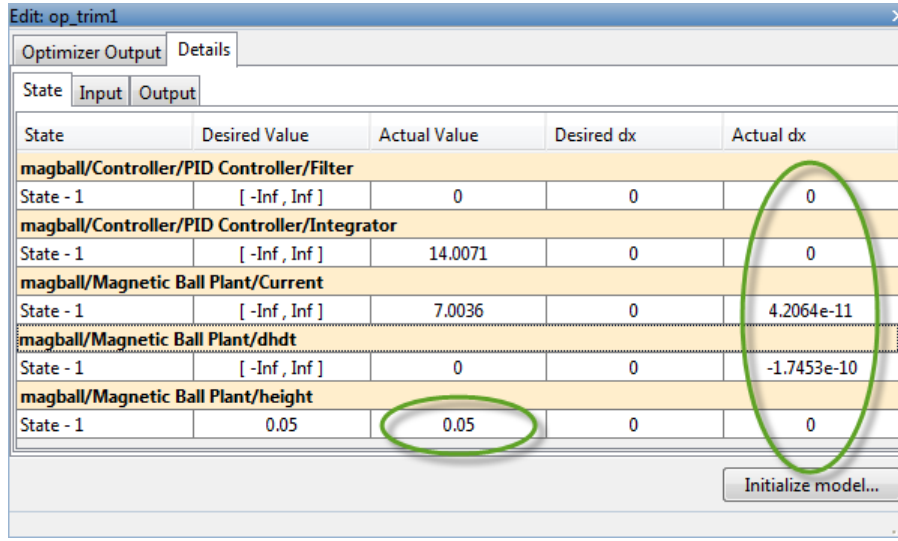


The Trim progress viewer shows that the optimization algorithm terminated successfully. The `(Maximum Error) Block` area shows the progress of reducing the error of a specific state or output during the optimization.

A new variable, `op_trim1`, appears in the **Linear Analysis Workspace**.



7 Double-click `op_trim1` in the **Linear Analysis Workspace** to evaluate whether the resulting operating point values meet the specifications.



In the **State** tab, the **Actual Value** for each state falls within the **Desired Value** bounds. The actual height of the ball is 0.05 m, as specified.

The **Actual dx** column shows the rates of change of the state values at the operating point. Since these values are at or near zero the states are not changing, showing that the operating point is in a steady state.

Related Examples

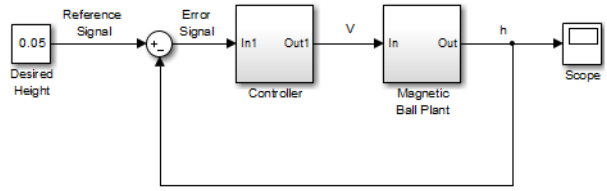
- "Steady-State Operating Points (Trimming) From Specifications" on page 2-3
- "Compute Operating Points at Simulation Snapshots"

More About

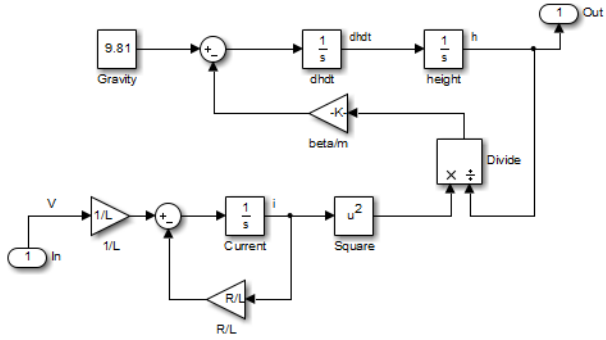
- "magball Simulink Model" on page 2-11
- "Steady-State Operating Point (Trimming)"

magball Simulink Model

The Simulink model `magball` includes the nonlinear Magnetic Ball Plant in a single-loop feedback system.

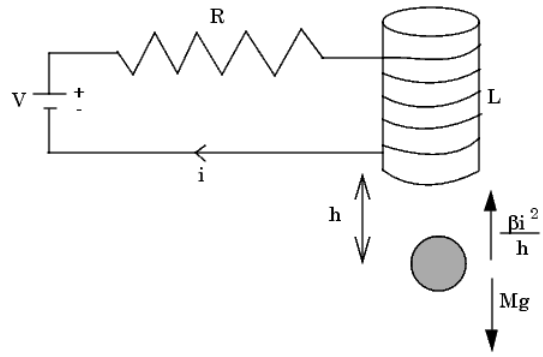


The Magnetic Ball Plant subsystem is shown in the following figure.



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

Examples and How To

"Steady-State Operating Points (Trimming) From Specifications" on page 2-3

3

Linearization

- "Applications of Linearization" on page 3-2
- "Open-Loop Response of Control System for Stability Margin Analysis" on page 3-3
- "Bode Response of Simulink Model" on page 3-7
- "watertank Simulink Model" on page 3-11

Applications of Linearization

Linearization is useful in model analysis and control design applications. After you linearize a Simulink model at a specific operating point, you can use your linear model to:

- Compute the Bode response of the Simulink model.
- Evaluate loop stability margins by computing open-loop response.
- Obtain linear state-space, transfer-function, or zero-pole-gain representation of the combined Simulink model that contains only linear blocks.
- Analyze and compare plant response near different operating points.
- Design linear controller

  Classical control system analysis and design methodologies require linear, time-invariant models. Simulink Control Design automatically linearization the plant when you tune your compensator. See “PID Control Design Using Robust-Response-Time Tuning Algorithm” on page 4-15.
- Analyze closed-loop stability.
- Measure the size of resonances in frequency response by computing closed-loop linear model for control system.
- Generate controllers with reduced sensitivity to parameter variations and modeling errors (requires Robust Control Toolbox™).

Examples and How To

- “Open-Loop Response of Control System for Stability Margin Analysis” on page 3-3
- “Bode Response of Simulink Model” on page 3-7
- “Steady-State Operating Points (Trimming) From Specifications” on page 2-3

More About

“Linearizing Nonlinear Models”

Open-Loop Response of Control System for Stability Margin Analysis

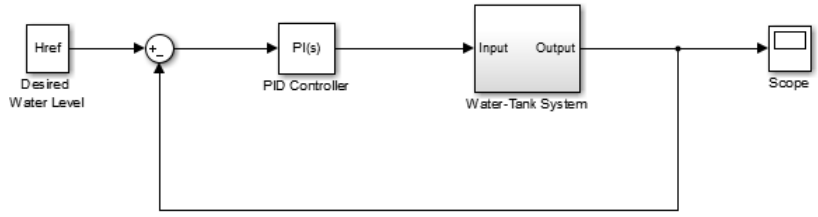
This example shows how to use the Linear Analysis Tool to analyze the open-loop response of a control system.

Compute a linear model of the combined controller-plant system without the effects of the feedback signal. Use a Bode plot of the resulting linear model to see the open-loop response.

1 Open Simulink model.

```
sys = 'watertank';
open_system(sys)
```

The Water-Tank System block represents the plant in this control system and contains all of the system nonlinearities.



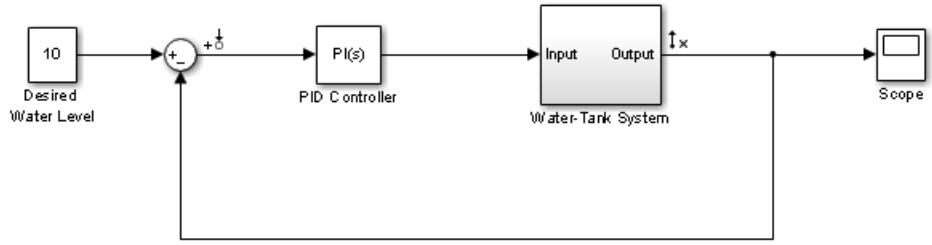
Copyright 2004-2012 The MathWorks, Inc.

2 In the Simulink Editor, define the portion of the model to linearize:

a Right-click the PID Controller block input signal (the output of the Sum block). Select **Linear Analysis Points > Input Perturbation**.

b Right-click the Water-Tank System output signal, and select **Linear Analysis Points > Open-loop Output**.

Annotations appear in the model indicating which signals are designated as linearization I/O points.

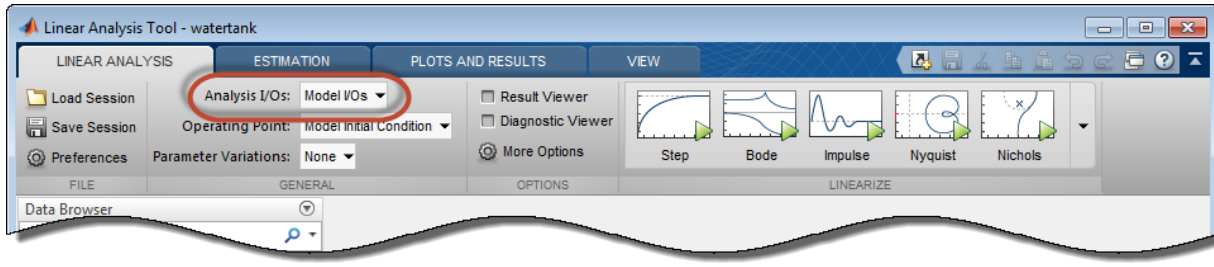


Tip Alternatively, if you do not want to introduce changes to the Simulink model, you can specify the linearization I/O points in the Linear Analysis Tool. See "Specify Portion of Model to Linearize in Linear Analysis Tool".

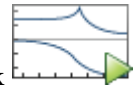
3 Open the Linear Analysis Tool for the model.

In the Simulink Editor, select **Analysis > Control Design > Linear Analysis**.

By default, the I/O points you specified in the model are the selected Analysis I/Os for linearization, as displayed in the **Analysis I/Os** menu.



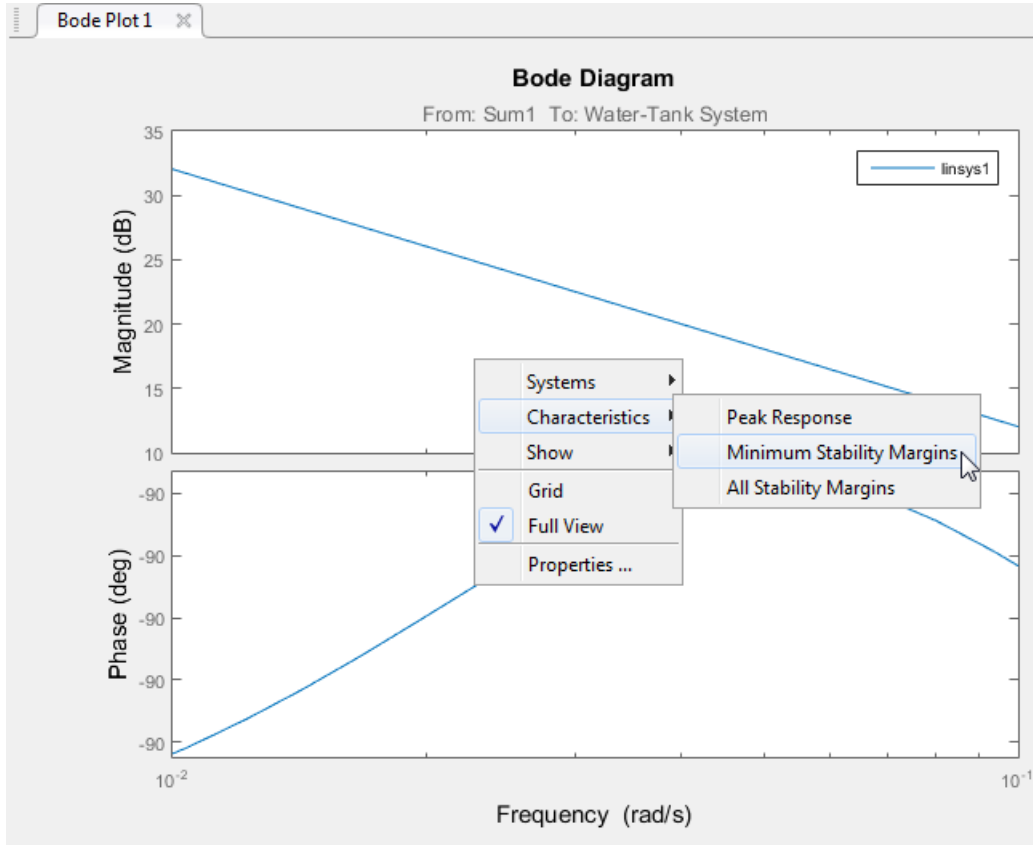
4 Linearize the model with the specified I/Os, and generate a Bode plot of the linearized model.

Click  **Bode**. The Bode plot of the linearized plant appears.

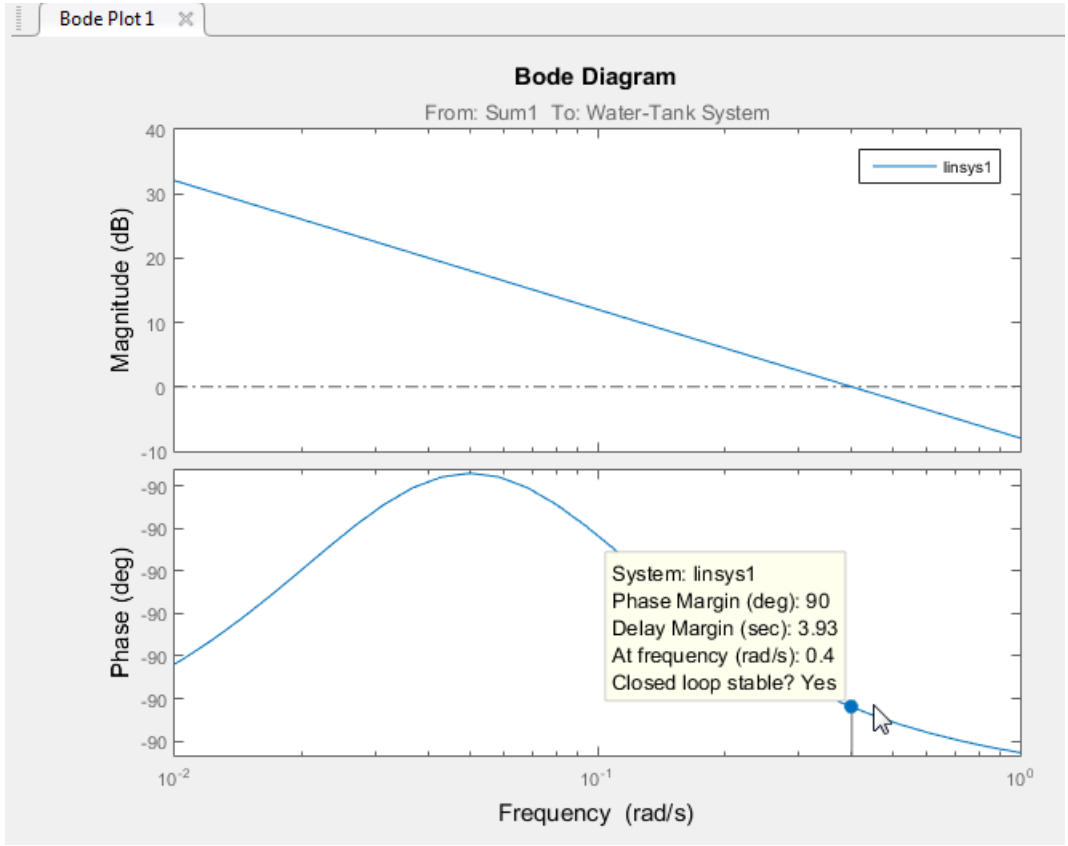
Tip Instead of a Bode plot, generate other response types by clicking the corresponding button in the plot gallery.

5 View the minimum stability margins for the model.

Right-click the plot and select **Characteristics > Minimum Stability Margins**.



The Bode plot displays the phase margin marker. Click the marker to show a data tip that contains the phase margin value.



6 Close Simulink model.

```
bdclose(sys);
```

Related Examples

- "Bode Response of Simulink Model" on page 3-7
- "Steady-State Operating Points (Trimming) From Specifications" on page 2-3

More About

- "Linearizing Nonlinear Models"
- "watertank Simulink Model" on page 3-11

Bode Response of Simulink Model

This example shows how to use the Linear Analysis Tool to linearize a model at the operating point specified in the model. The model operating point consists of the model initial state values and input signals.

The Linear Analysis Tool linearizes at the model operating point by default. If you want to specify a different operating point for linearization, see "Linearize at Trimmed Operating Point".

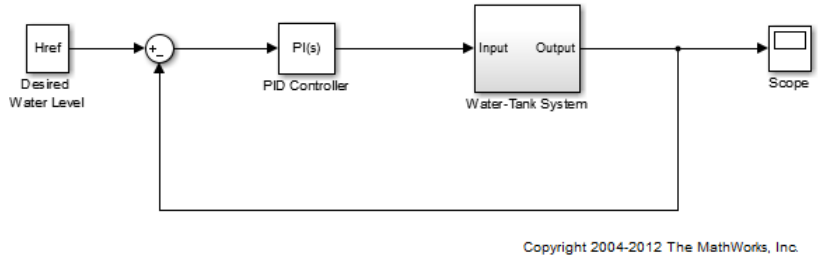
Code Alternative

Use `linearize`. For examples and additional information, see the `linearize` reference page.

1 Open Simulink model.

```
sys = 'watertank';
open_system(sys)
```

The Water-Tank System block represents the plant in this control system and includes all of the system nonlinearities.



2 Open the Linear Analysis Tool for the model.

In the Simulink Editor, select **Analysis > Control Design > Linear Analysis**.

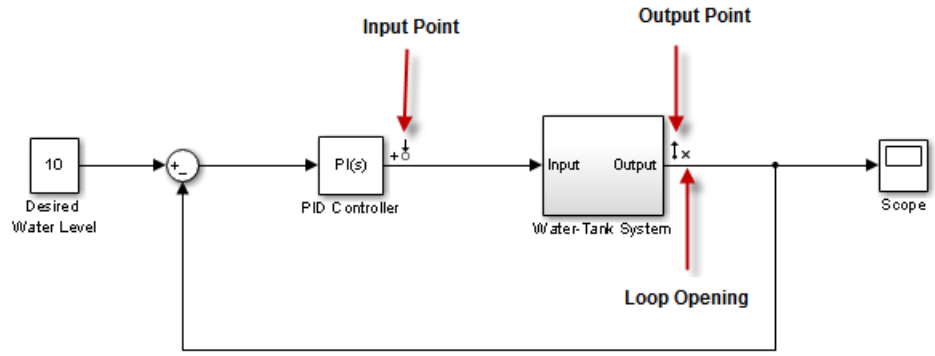
3 In the Simulink Editor, define the portion of the model to linearize:

a Right-click the PID Controller block output signal, which is the input to the plant. Select **Linear Analysis Points > Input Perturbation**.

b Right-click the Water-Tank System output signal, and select **Linear Analysis Points > Open-loop Output**.

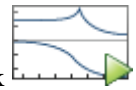
Inserting this open loop point removes the effects of the feedback signal on the linearization without changing the model operating point.

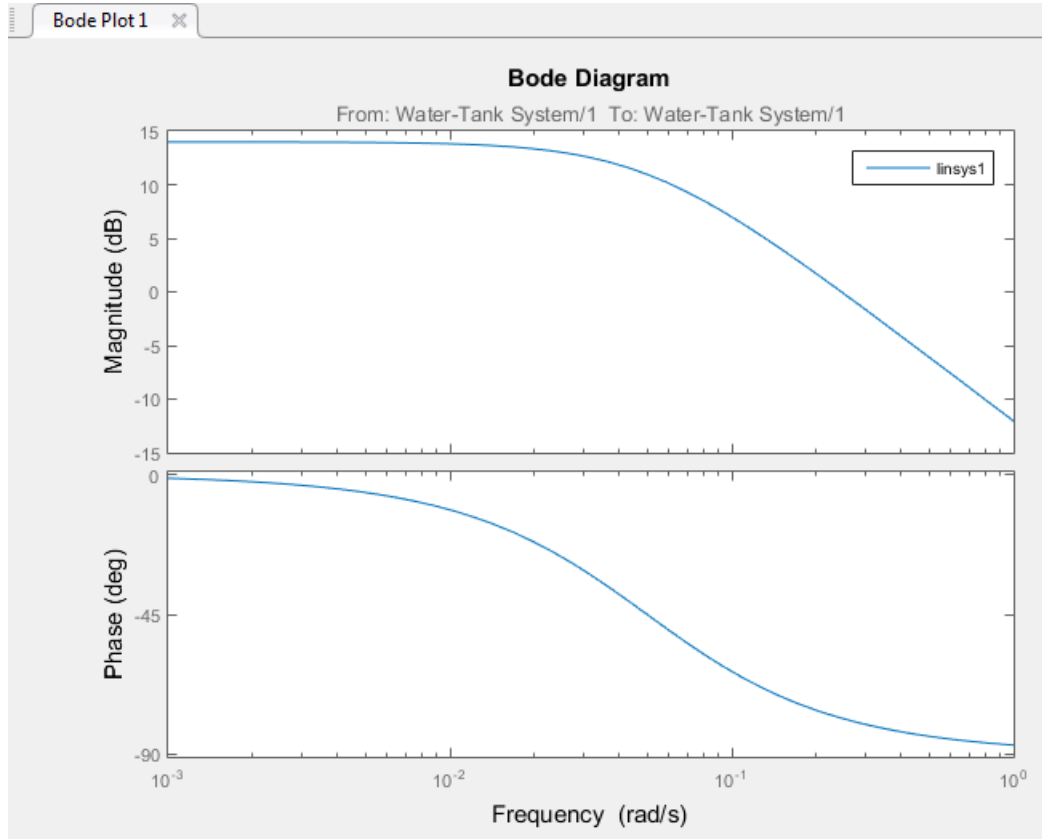
When you add linear analysis points, marker appear at their locations in the model.



Tip Alternatively, if you do not want to introduce changes to the Simulink model, you can specify the linearization I/O points in the Linear Analysis Tool. See "Specify Portion of Model to Linearize in Linear Analysis Tool".

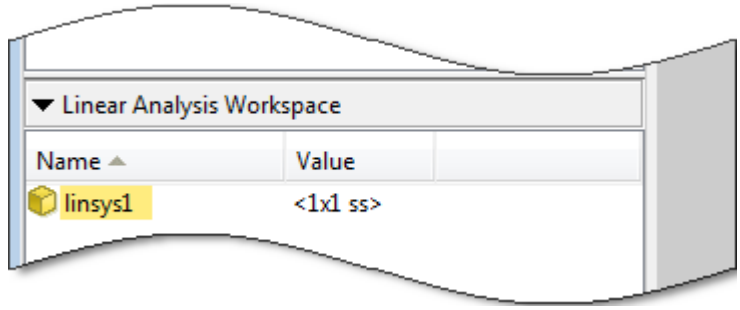
4 Linearize the model with the specified I/Os, and generate a Bode plot of the linearized model.

Click  **Bode**. The Bode plot of the linearized plant appears.



Tip Instead of a Bode plot, generate other response types by clicking the corresponding button in the plot gallery.

The linearized system, `linsys1`, appears in the Linear Analysis Workspace.



`linsys1` represents the system linearized at the model operating point. If you do not specify an operating point for linearization, the Linear Analysis Tool uses the model operating point by default.

5 Close Simulink model.

```
bdclose(sys);
```

Related Examples

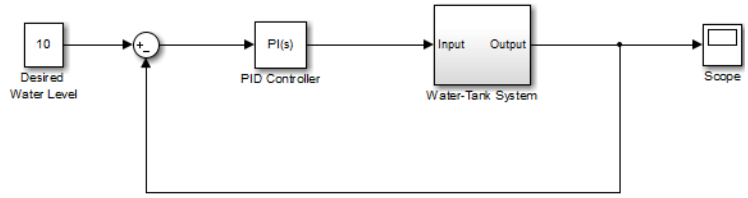
- "Open-Loop Response of Control System for Stability Margin Analysis" on page 3-3
- "Steady-State Operating Points (Trimming) From Specifications" on page 2-3

More About

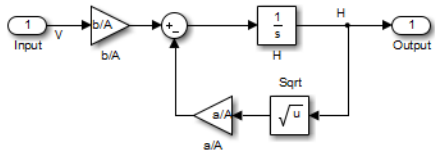
- "Linearizing Nonlinear Models"
- "watertank Simulink Model" on page 3-11

watertank Simulink Model

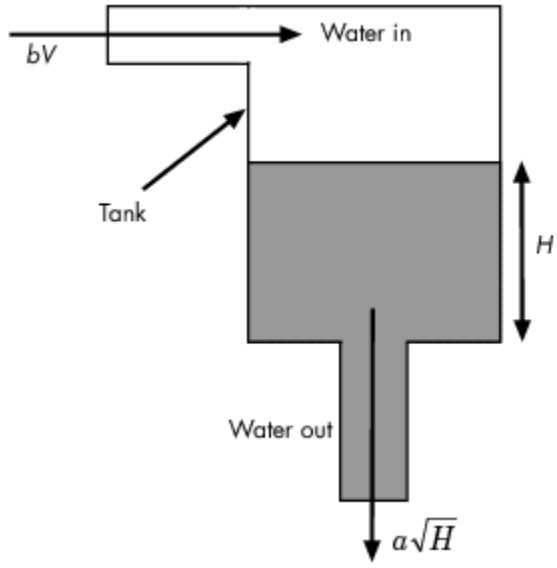
The Simulink model `watertank` includes the nonlinear Water-Tank System plant and a PI controller in a single-loop feedback system.



The Water-Tank System is shown 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

PID Control Design

- "PID Controller Tuning in Simulink" on page 4-2
- "Design a Controller Using Automated Tuning and Bode Graphical Design" on page 4-11

PID Controller Tuning in Simulink

This example shows how to automatically tune a PID Controller block using PID Tuner.

Introduction of the PID Tuner

PID Tuner provides a fast and widely applicable single-loop PID tuning method for the Simulink® PID Controller blocks. With this method, you can tune PID controller parameters to achieve a robust design with the desired response time.

A typical design workflow with the PID Tuner involves the following tasks:

(1) Launch the PID Tuner. When launching, the software automatically computes a linear plant model from the Simulink model and designs an initial controller.

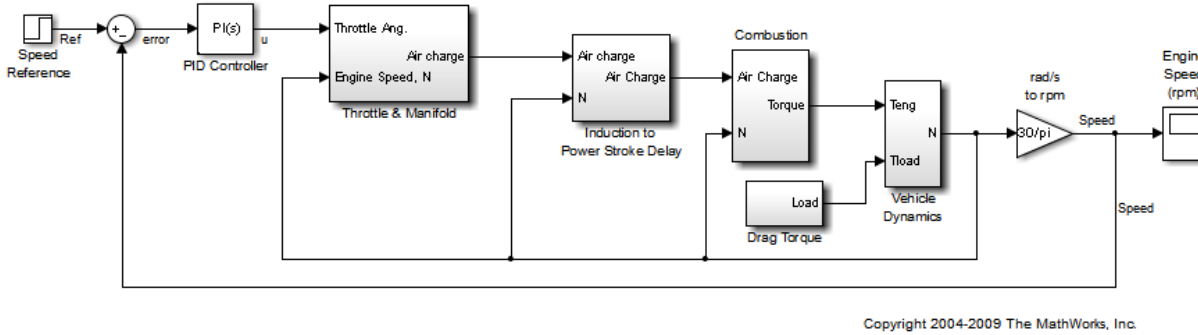
(2) Tune the controller in the PID Tuner by manually adjusting design criteria in two design modes. The tuner computes PID parameters that robustly stabilize the system.

(3) Export the parameters of the designed controller back to the PID Controller block and verify controller performance in Simulink.

Opening the Model

Open the engine speed control model with PID Controller block and take a few moments to explore it.

```
open_system('scdspeedctrlpidblock');
```



Design Overview

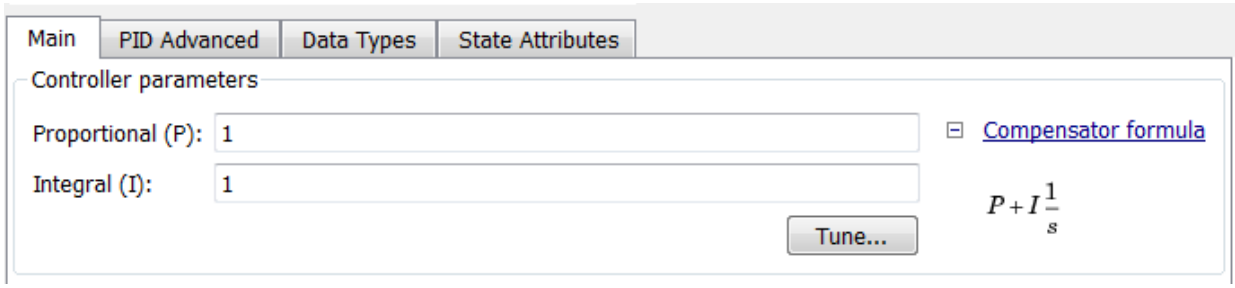
In this example, you design a PI controller in an engine speed control loop. The goal of the design is to track the reference signal from a Simulink step block `scdspeedctrlpidblock/Speed Reference`. The design requirement are:

- Settling time under 5 seconds
- Zero steady-state error to the step reference input.

In this example, you stabilize the feedback loop and achieve good reference tracking performance by designing the PI controller `scdspeedctrl/PID Controller` in the PID Tuner.

Opening the PID Tuner

To launch the PID Tuner, double-click the PID Controller block to open its block dialog. In the **Main** tab, click **Tune**.

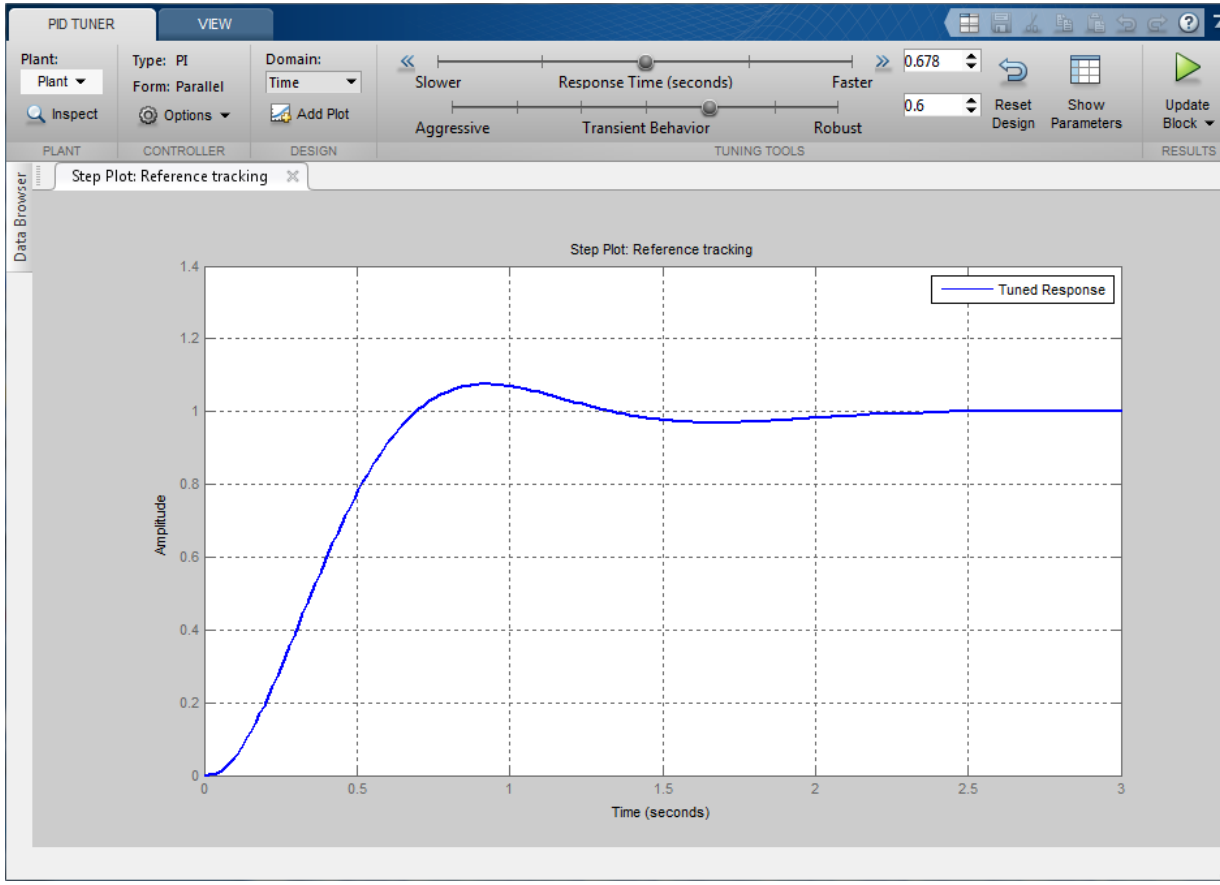


Initial PID Design

When the PID Tuner launches, the software computes a linearized plant model seen by the controller. The software automatically identifies the plant input and output, and uses the current operating point for the linearization. The plant can have any order and can have time delays.

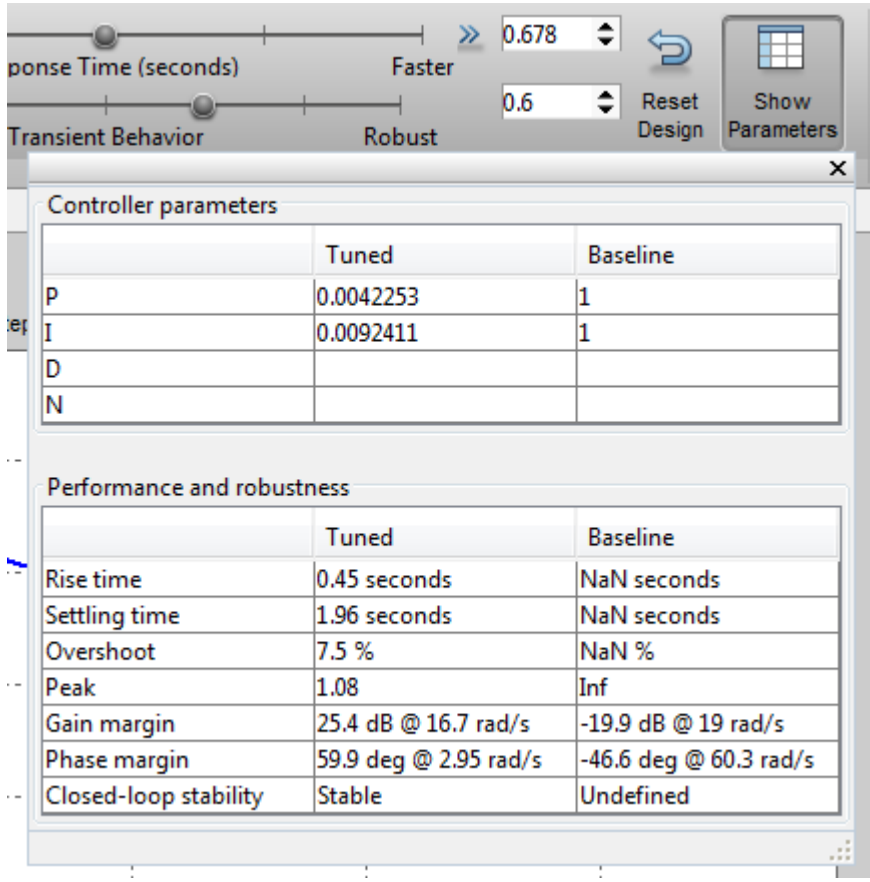
The PID Tuner computes an initial PI controller to achieve a reasonable tradeoff between performance and robustness. By default, step reference tracking performance displays in the plot.

The following figure shows the PID Tuner dialog with the initial design:



Displaying PID Parameters

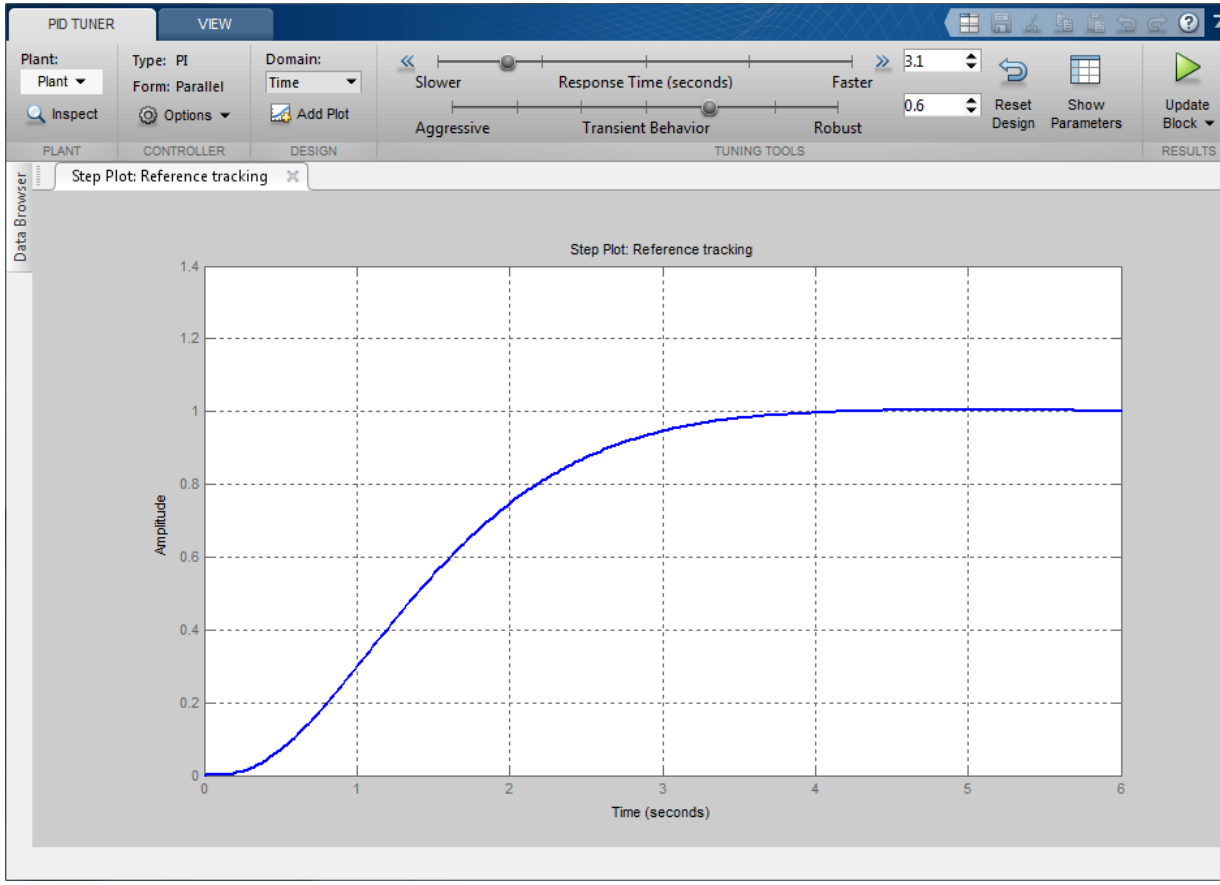
Click **Show parameters** to view controller parameters P and I, and a set of performance and robustness measurements. In this example, the initial PI controller design gives a settling time of 2 seconds, which meets the requirement.

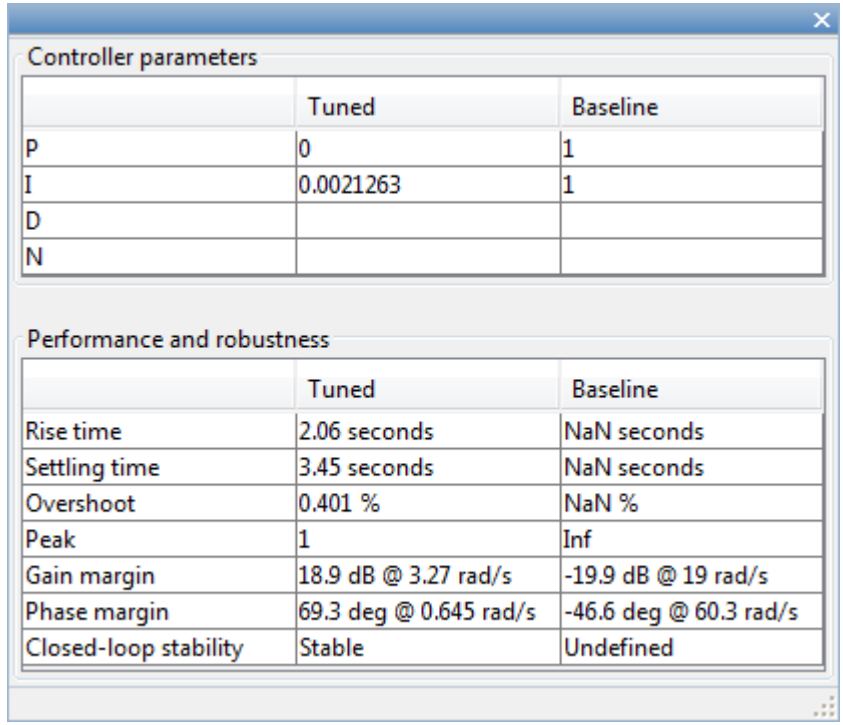


Adjusting PID Design in the PID Tuner

The overshoot of the reference tracking response is about 7.5 percent. Since we still have some room before reaching the settling time limit, you could reduce the overshoot by increasing the response time. Move the response time slider to the left to increase the closed loop response time. Notice that when you adjust response time, the response plot and the controller parameters and performance measurements update.

The following figure shows an adjusted PID design with an overshoot of zero and a settling time of 4 seconds. The designed controller effectively becomes an integral-only controller.


PID TUNER
VIEW
Plant:
Plant
Inspect
Type: PI
Form: Parallel
Options
Domain:
Time
Add Plot
Slower
Response Time (seconds)
Faster
3.1
0.6
Aggressive
Transient Behavior
Robust
Reset Design
Show Parameters
Update Block
PLANT
CONTROLLER
DESIGN
TUNING TOOLS
RESULTS
Data Browser
Step Plot: Reference tracking
Step Plot: Reference tracking
Amplitude
Time (seconds)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0
1
2
3
4
5
6



Controller parameters

	Tuned	Baseline
P	0	1
I	0.0021263	1
D		
N		

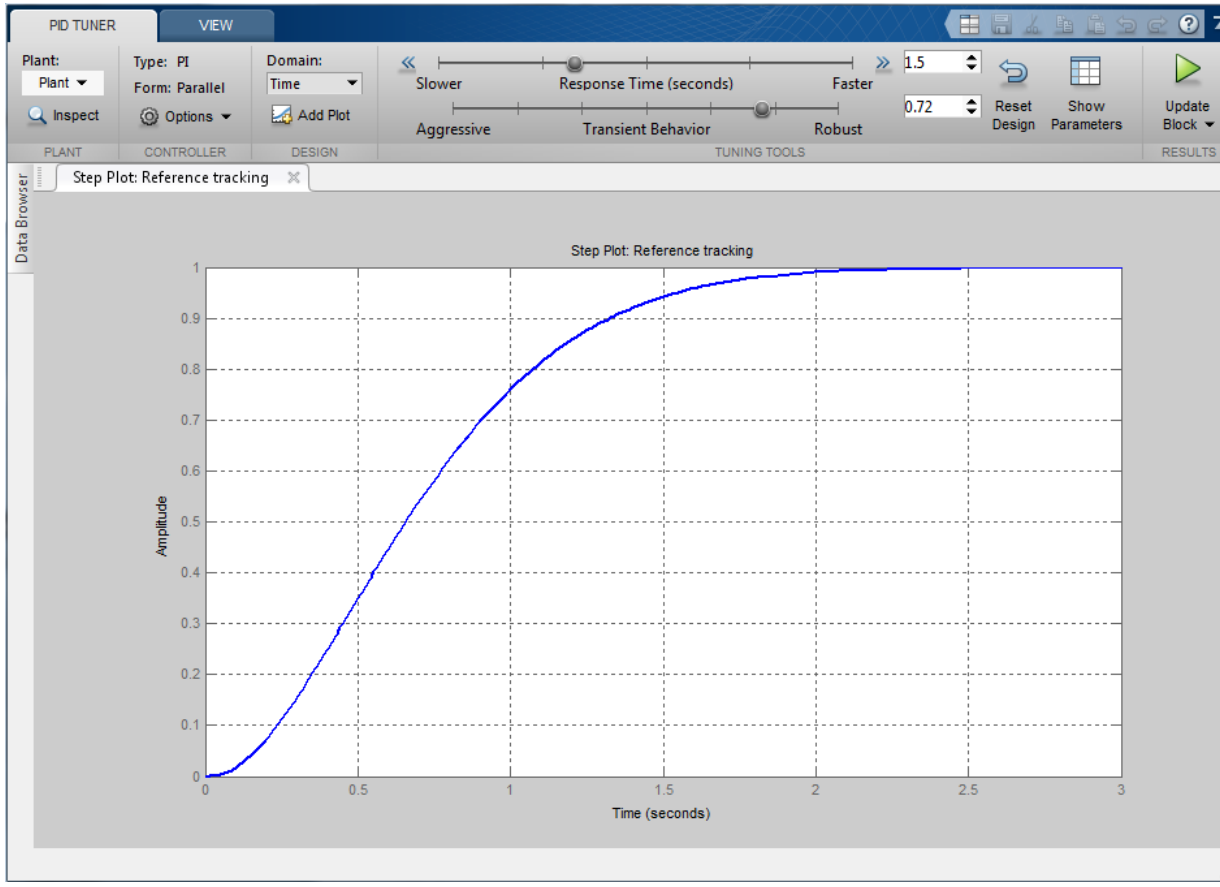
Performance and robustness

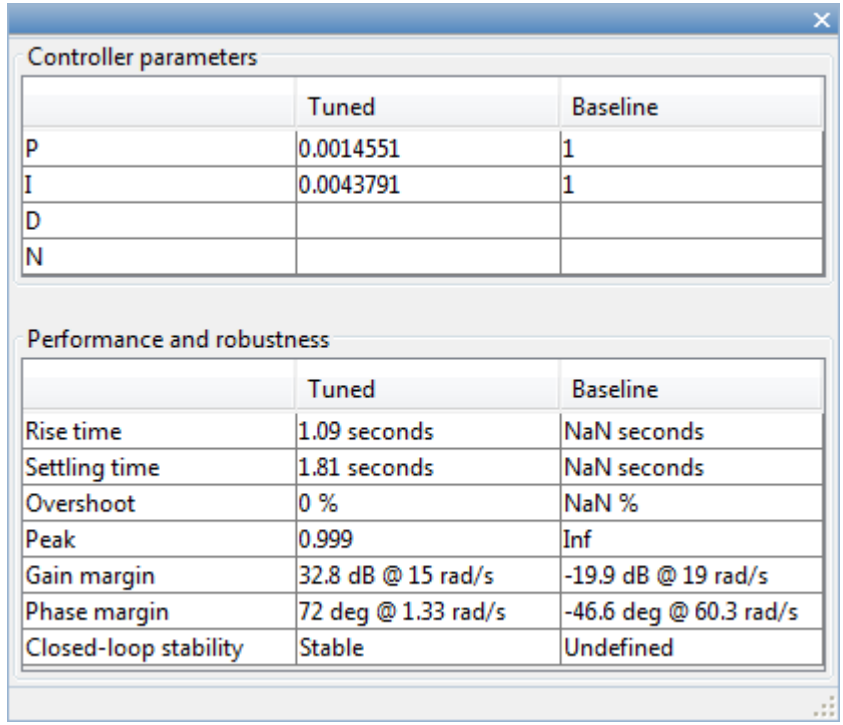
	Tuned	Baseline
Rise time	2.06 seconds	NaN seconds
Settling time	3.45 seconds	NaN seconds
Overshoot	0.401 %	NaN %
Peak	1	Inf
Gain margin	18.9 dB @ 3.27 rad/s	-19.9 dB @ 19 rad/s
Phase margin	69.3 deg @ 0.645 rad/s	-46.6 deg @ 60.3 rad/s
Closed-loop stability	Stable	Undefined

Completing PID Design with Performance Trade-Off

In order to achieve zero overshoot while reducing the settling time below 2 seconds, you need to take advantage of both sliders. You need to make control response faster to reduce the settling time and increase the robustness to reduce the overshoot. For example, you can reduce the response time from 3.4 to 1.5 seconds and increase robustness from 0.6 to 0.72.

The following figure shows the closed-loop response with these settings:


PID TUNER
VIEW
Plant:
Plant
Inspect
Type: PI
Form: Parallel
Options
Domain:
Time
Add Plot
Slower
Response Time (seconds)
Faster
1.5
Aggressive
Transient Behavior
Robust
0.72
Reset Design
Show Parameters
Update Block
PLANT
CONTROLLER
DESIGN
TUNING TOOLS
RESULTS
Data Browser
Step Plot: Reference tracking
Step Plot: Reference tracking
Amplitude
Time (seconds)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.5
1.5
2
2.5
3



Controller parameters

	Tuned	Baseline
P	0.0014551	1
I	0.0043791	1
D		
N		

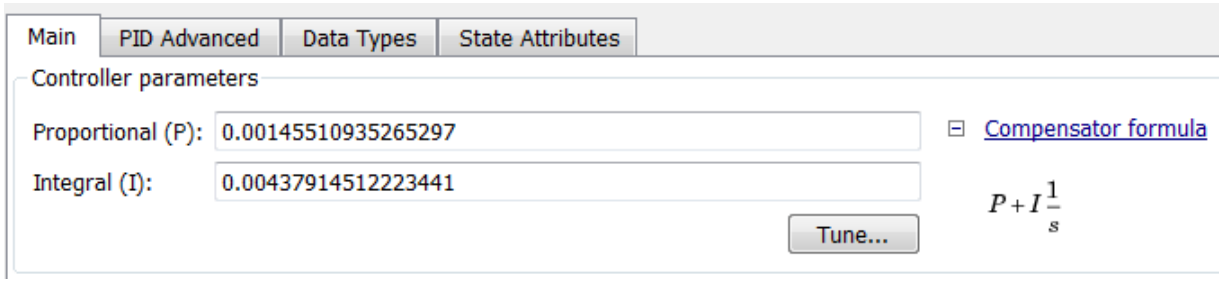
Performance and robustness

	Tuned	Baseline
Rise time	1.09 seconds	NaN seconds
Settling time	1.81 seconds	NaN seconds
Overshoot	0 %	NaN %
Peak	0.999	Inf
Gain margin	32.8 dB @ 15 rad/s	-19.9 dB @ 19 rad/s
Phase margin	72 deg @ 1.33 rad/s	-46.6 deg @ 60.3 rad/s
Closed-loop stability	Stable	Undefined

Writing the Tuned Parameters to PID Controller Block

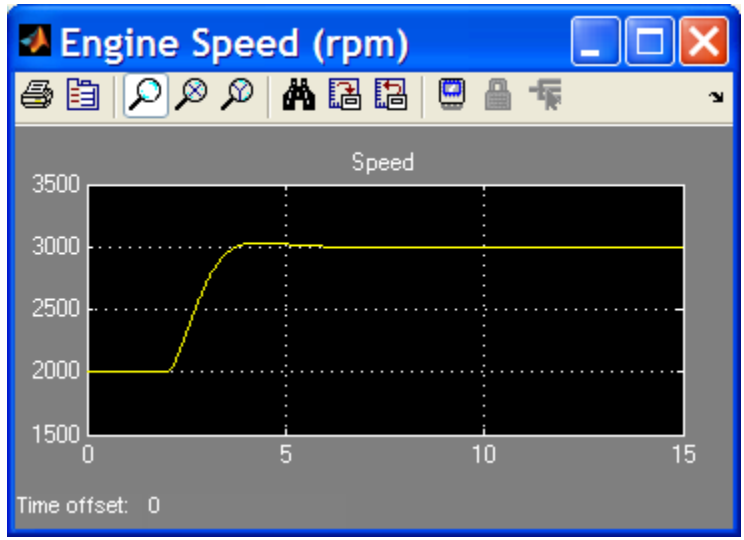
After you are happy with the controller performance on the linear plant model, you can test the design on the nonlinear model. To do this, click **Update Block** in the PID Tuner. This action writes the parameters back to the PID Controller block in the Simulink model.

The following figure shows the updated PID Controller block dialog:



Completed Design

The following figure shows the response of the closed-loop system:



The response shows that the new controller meets all the design requirements.

You can also use the SISO Compensator Design Tool to design the PID Controller block. When the PID Controller block belongs to a multi-loop design task. See the example "Single Loop Feedback/Prefilter Compensator Design".

```
bdclose('scdspeedctrlpidblock')
```

Design a Controller Using Automated Tuning and Bode Graphical Design

In this section...

"About This Tutorial" on page 4-11
"PID Control Design Using Robust-Response-Time Tuning Algorithm" on page 4-15
"PID Control Design Using Bode Graphical Tuning" on page 4-23
"Closed-Loop Simulation of Simulink Model" on page 4-27

About This Tutorial

- "Objectives" on page 4-11
- "About the Model" on page 4-11
- "Requirements for the Compensator Design" on page 4-14
- "Overview of the Compensator Design Process" on page 4-14

Objectives

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

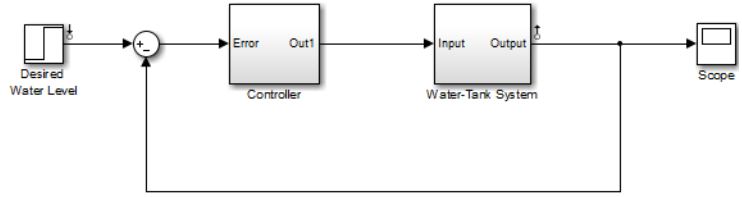
- Configure the model and GUI for compensator design.
- Design a PID compensator using the robust-response-time tuning algorithm and Bode graphical design.
- Simulate the closed-loop nonlinear model.

About the Model

- "watertank_comp_design Simulink Model" on page 4-11
- "Water-Tank Subsystem" on page 4-12
- "Controller Subsystem" on page 4-14

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.

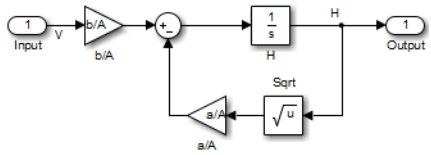


To view the Water-Tank System and the Controller, double-click the corresponding subsystem in the `watertank_comp_design` model. For descriptions of these subsystems, see the following topics:

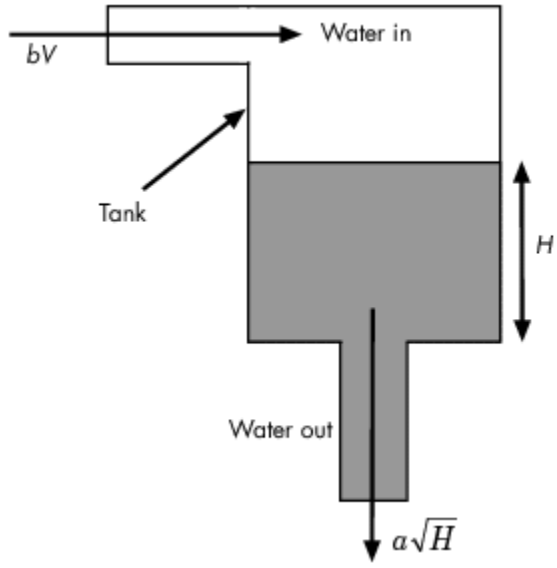
- "Water-Tank Subsystem" on page 4-12
- "Controller Subsystem" on page 4-14

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.

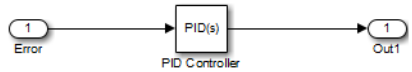
The following table describes the variables, parameters, differential equations, states, inputs, and outputs of the water-tank system.

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 PID controller you design in this tutorial must control the Water-Tank System response such that the:

- Overshoot is less than 5%.
- 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 the robust response time tuning algorithm.
- 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 work only with linear plant models. Because the Water-Tank System is nonlinear, Simulink Control Design automatically linearizes the

model about the model operating point, by default. 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 “Linearizing Nonlinear Models”.

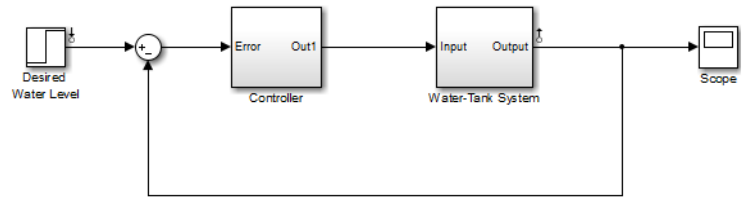
PID Control Design Using Robust-Response-Time Tuning Algorithm

In this portion of the tutorial, you design a compensator using the automated PID robust-response-time tuning algorithm. This tuning method tunes the PID gains to maximize bandwidth and optimize phase margin.

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

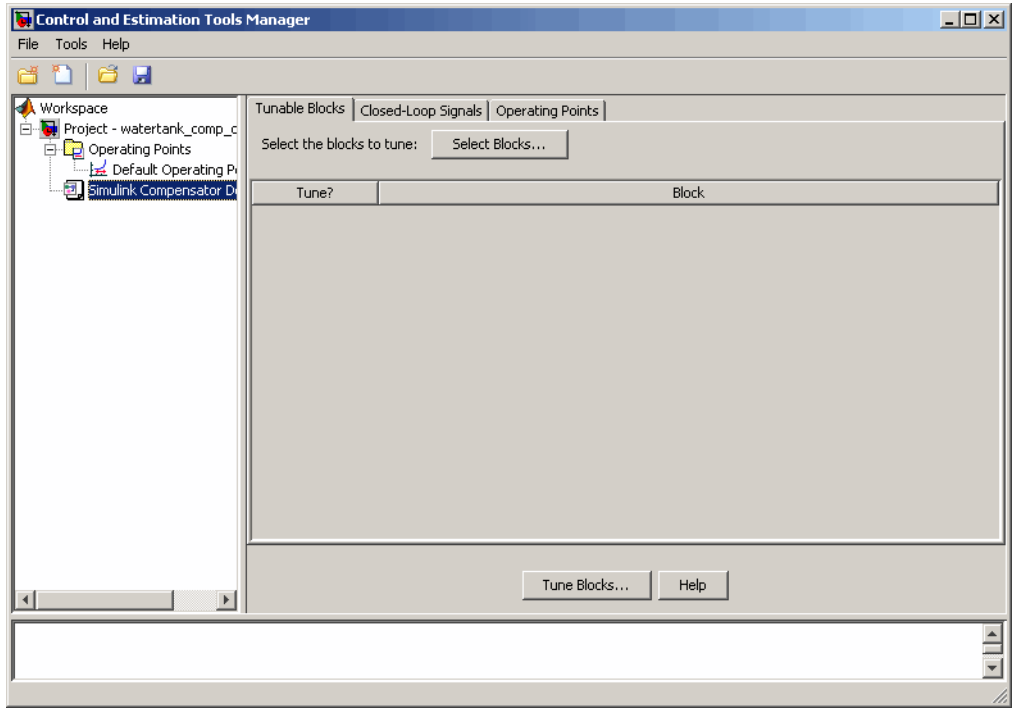
```
watertank_comp_design
```

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 **Analysis > Control Design > Control System Designer**.

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



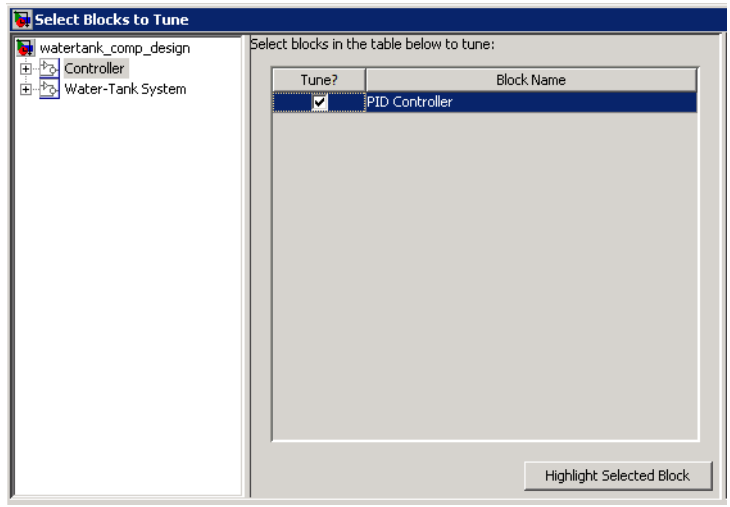
3 Select the PID Controller block as the block to tune.

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?** check box for **PID Controller**.



d Click **OK**.

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

The input and output points of the closed-loop path are already defined in the `watertank_comp_design` model. If you needed to add or define them, you would use the following steps:

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

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

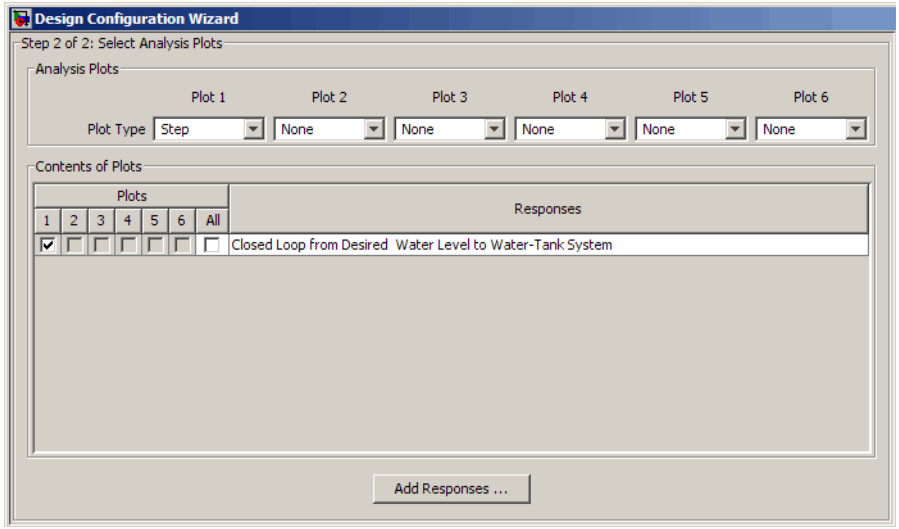
5 In the Control and Estimation Tools Manager, click **Tune Blocks** to open the Design Configuration Wizard. Click **Next**.

6 Step 1 of the Design Configuration Wizard prompts you to select the design plots you will use to tune the controller. 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 Desired Water Level to Water-Tank System**.



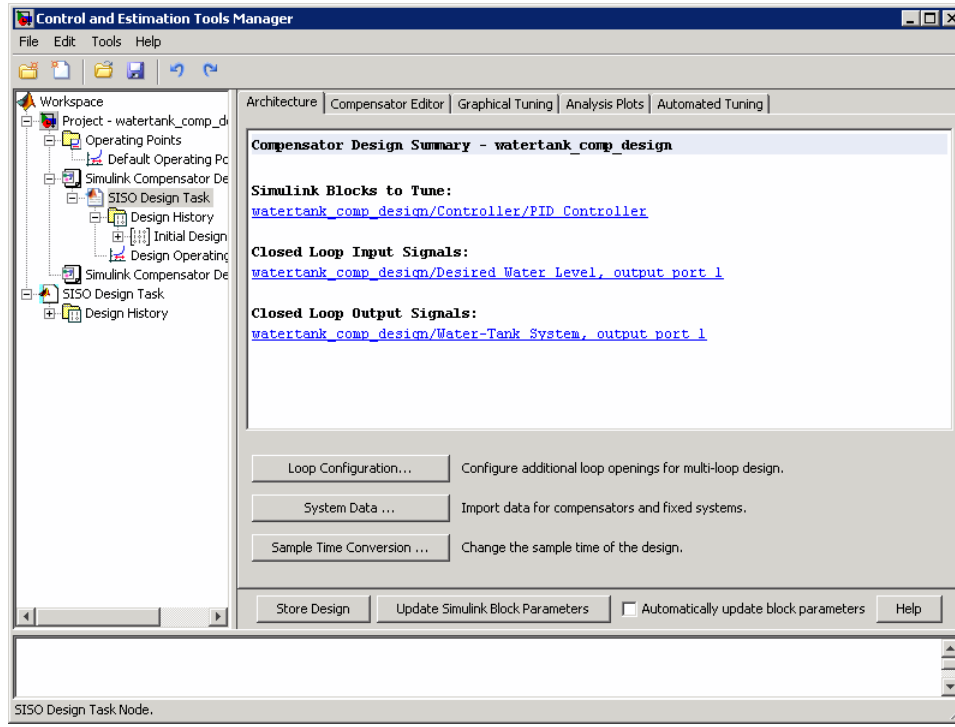
8 Click **Finish**.

The software performs the following actions:

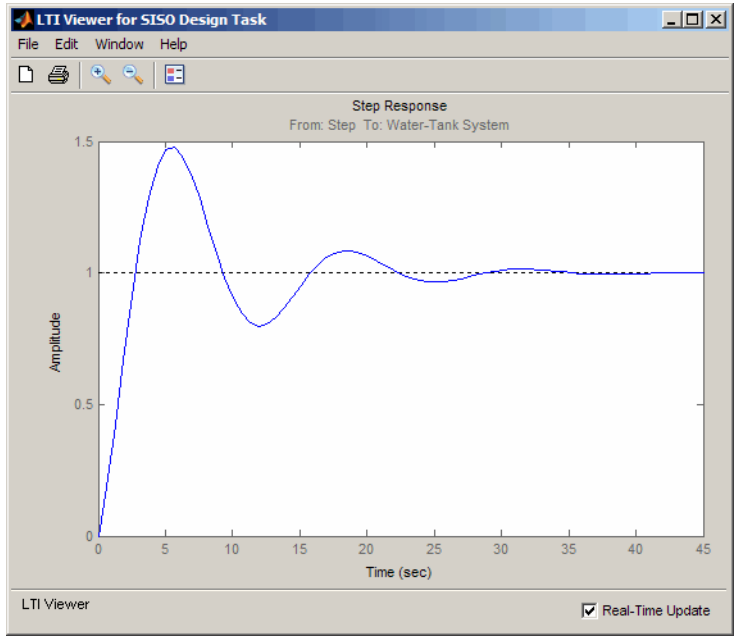
- Linearizes the Simulink model about the operating point specified in the model.
- Creates a **SISO Design Task** node under the **Simulink Compensator Design Task** node.
- Opens the following plot windows:
  - Linear System Analyzer for SISO Design Task window, which shows the closed-loop Step Response 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.



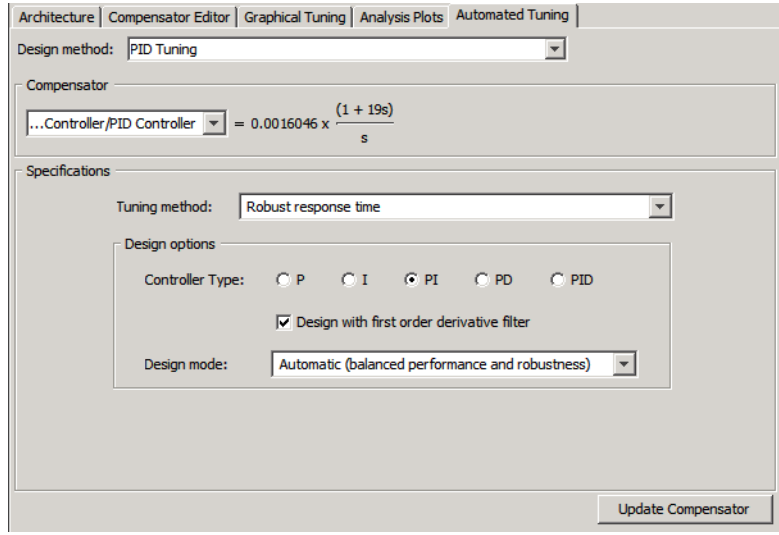
The Step Response plot shows an overshoot that does not meet the overshoot design requirement of less than 5%.



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 method**: `Robust response time`



11 Click **Update Compensator**.

This action computes the PI values for the compensator using the robust response time 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:

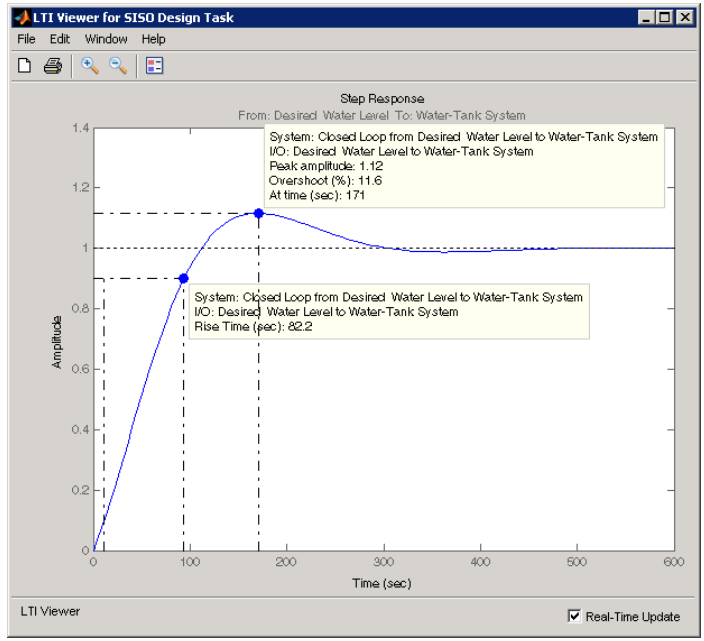
- **Characteristics > Peak Response**
- **Characteristics > Rise Time**

  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 11.6%.
- The rise time is 82.2 seconds.



This system response with the PID compensator 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 "PID Control Design Using Bode Graphical Tuning" on page 4-23.

Tip You can also decrease the rise time by adjusting the loop bandwidth. First, select `Interactive (adjustable performance and robustness)` from the **Design Mode** menu. Then, move the **Bandwidth** slider to the right. Finally, click **Update Compensator** to design a new compensator for the new target bandwidth.

PID Control Design Using Bode Graphical Tuning

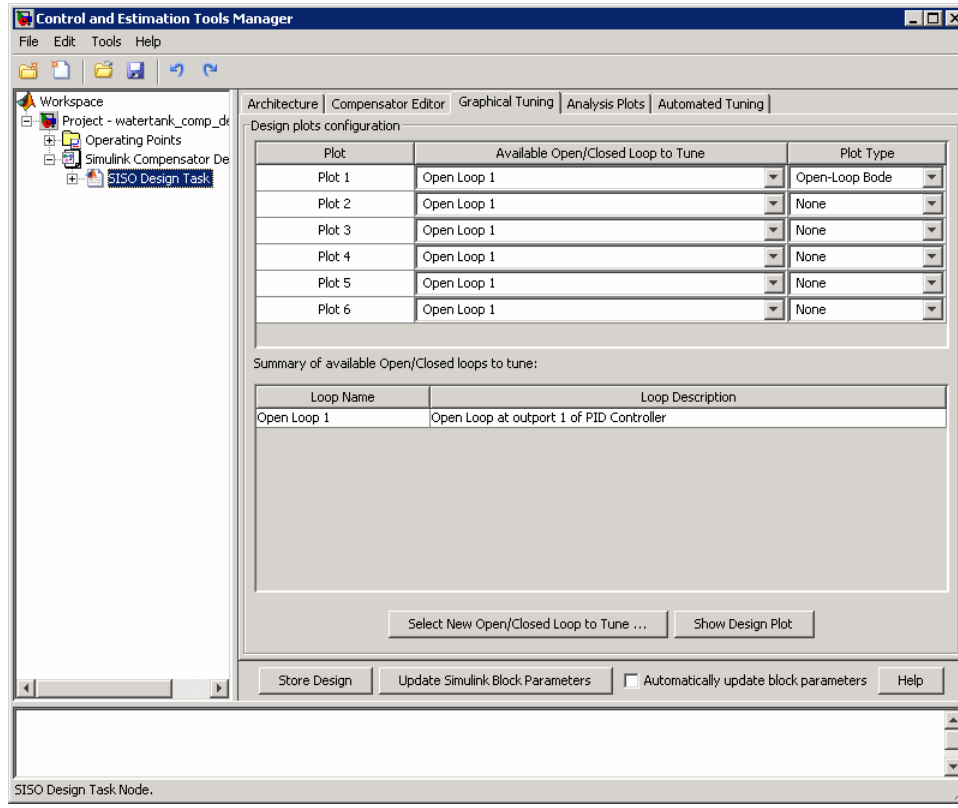
In this example, you decrease the rise time of the Water-Tank System response by increasing the compensator gain using Bode graphical tuning.

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 "PID Control Design Using Robust-Response-Time Tuning Algorithm" on page 4-15.

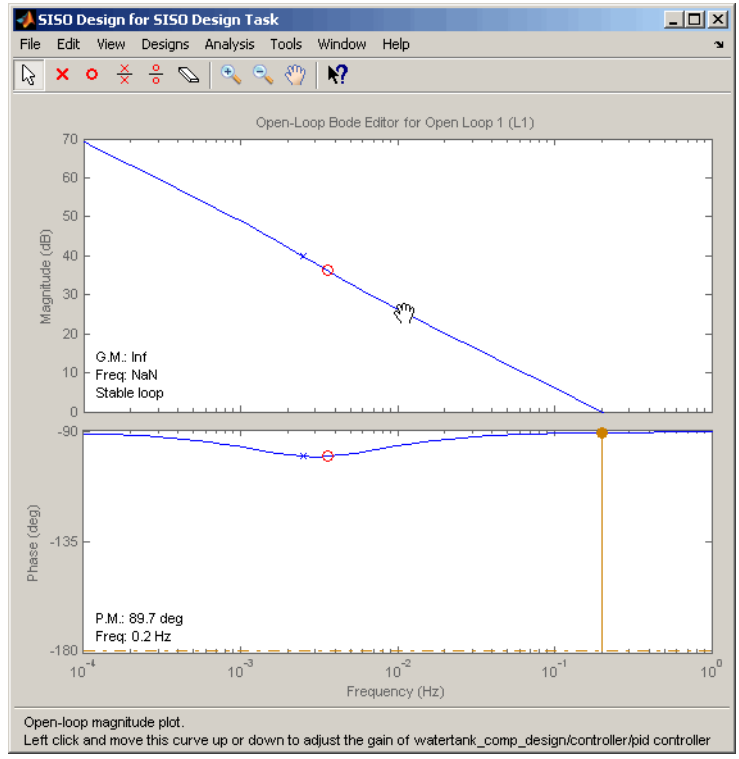
To design a compensator using Bode graphical tuning:

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:

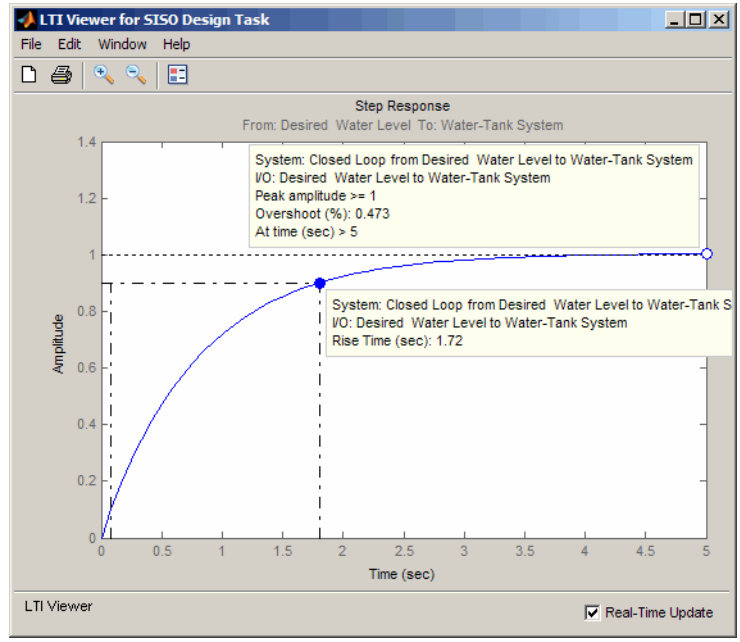
- **Characteristics > Peak Response**
- **Characteristics > Rise Time**

  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.

Closed-Loop Simulation of Simulink Model

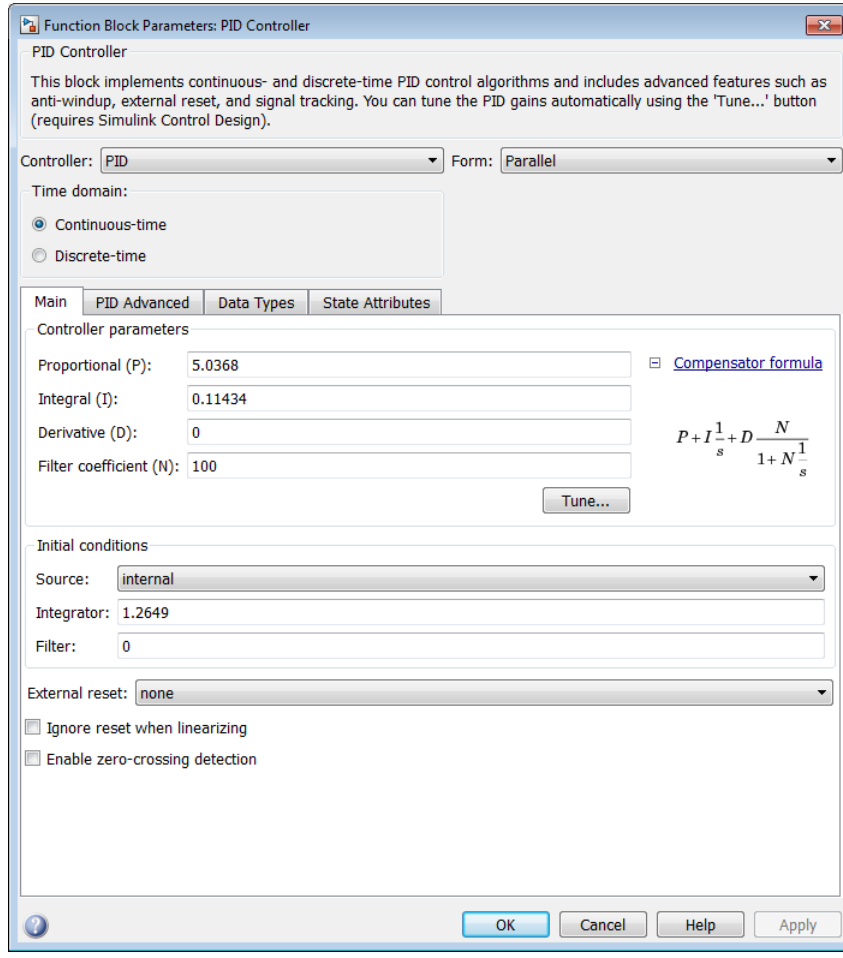
In this example, you simulate the nonlinear closed-loop Simulink model that includes a PID controller to determine how well the design meets the requirements.

You must have already designed the compensator, as described in "PID Control Design Using Bode Graphical Tuning" on page 4-23.

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

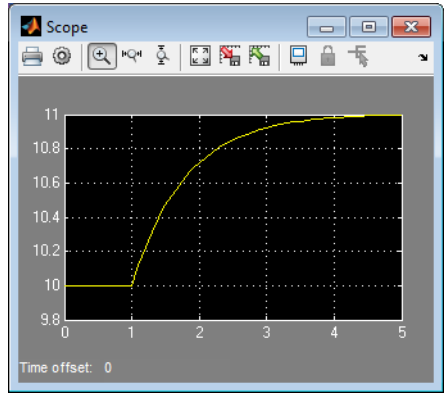
This action writes the compensator parameters into the PID Controller block of the Controller subsystem in the Simulink model.

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 Simulate the model.



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.