PID Controller Tuning in Simulink

Control System Toolbox™ provides tools for manipulating and tuning PID controllers through the PID Tuner app as well as command-line functions.

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

- Introduction to the PID Tuner
- About the Model
- Design Overview
- Opening the PID Tuner
- Initial PID Design
- Displaying PID Parameters
- Adjusting PID Design in the PID Tuner
- Completing PID Design with Performance Trade-Off
- Writing the Tuned Parameters to PID Controller Block
- Completed Design

Introduction to the PID Tuner

The PID Tuner app in Control System Toolbox 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.

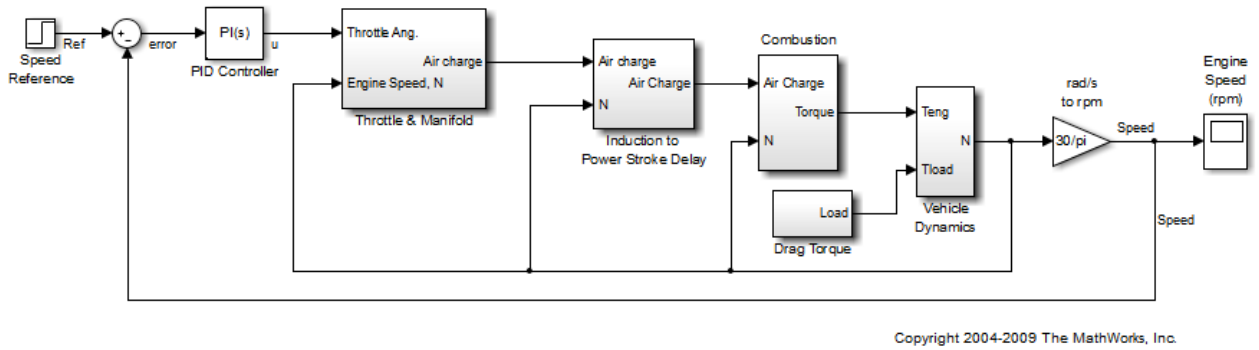
A typical design workflow with the PID Tuner involves the following tasks:

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.

The Model

You can open the engine speed control model with PID Controller block in Simulink.

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



Design Overview

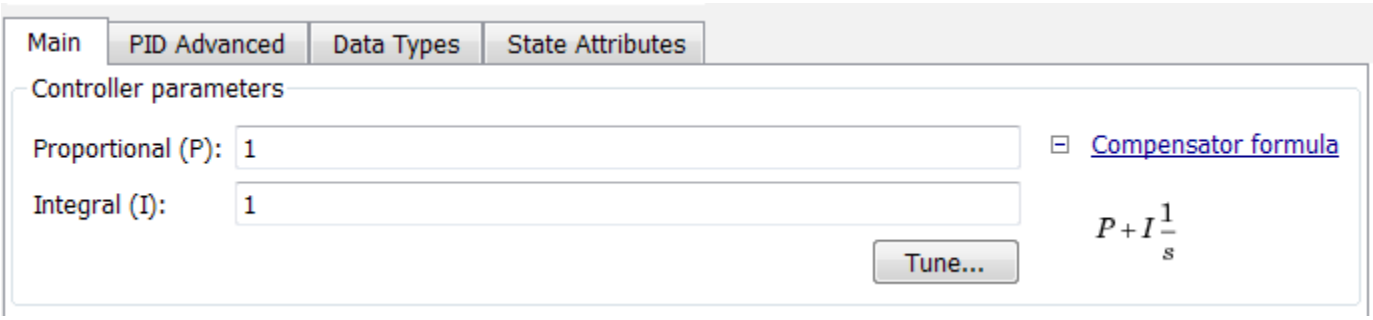
In this example, you design a PI controller in an engine speed control loop. The goal of the design is to track the reference signal from a Simulink step block `scdspeedctrlpidblock/Speed Reference`. The design requirement are:

- 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.**

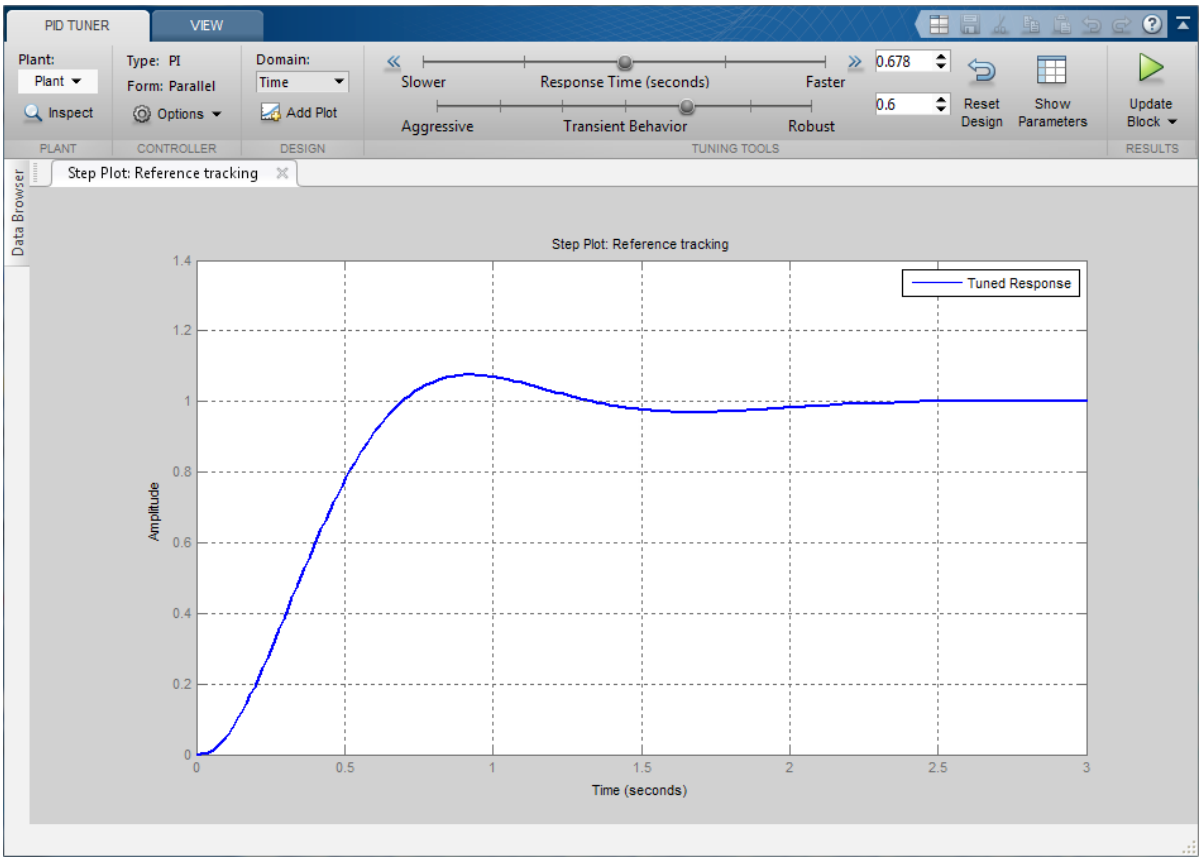


Initial PID Design

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.

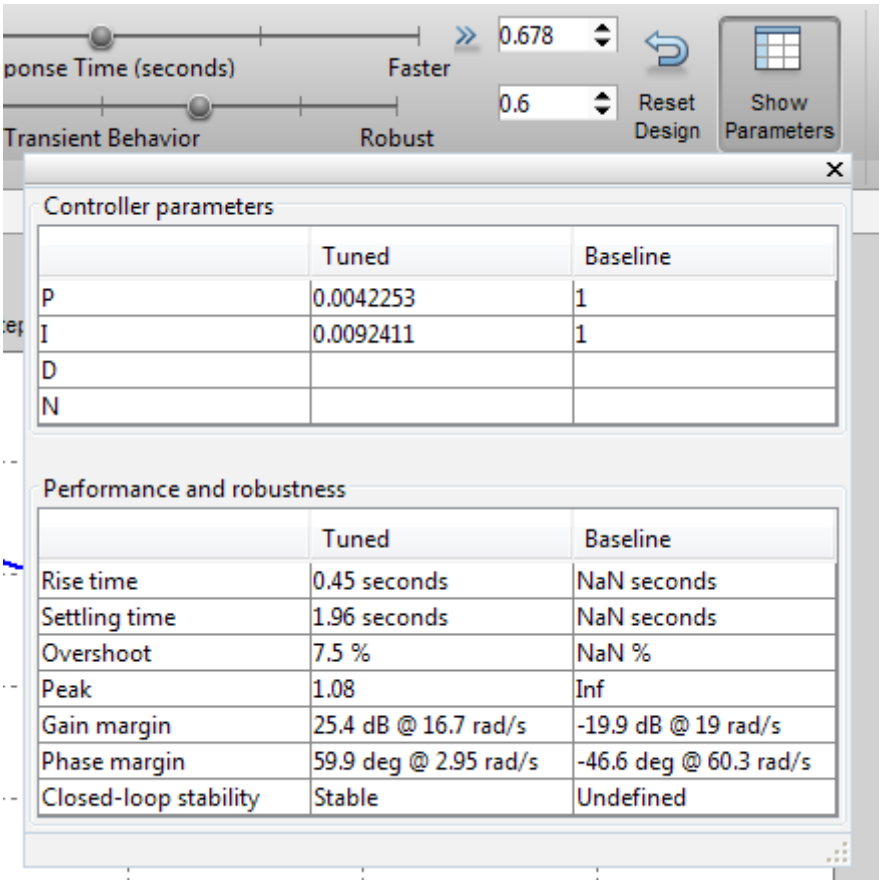
The PID Tuner computes an initial PI controller to achieve a reasonable tradeoff between performance and robustness. By default, step reference tracking performance displays in the plot.

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



Displaying PID Parameters

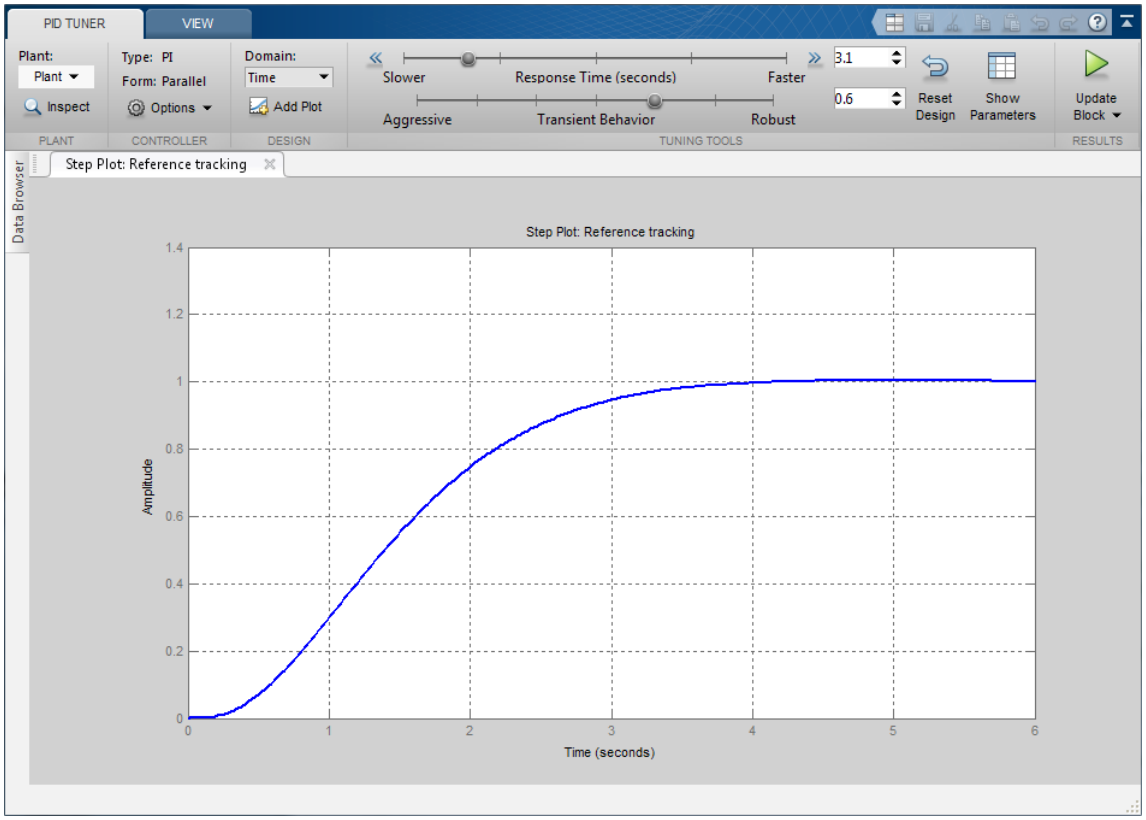
Click **Show parameters** to view controller parameters P and I, and a set of performance and robustness measurements. In this example, the initial PI controller design gives a settling time of 2 seconds, which meets the requirement.



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.



Controller parameters

	Tuned	Baseline
P	0	1
I	0.0021263	1
D		
N		

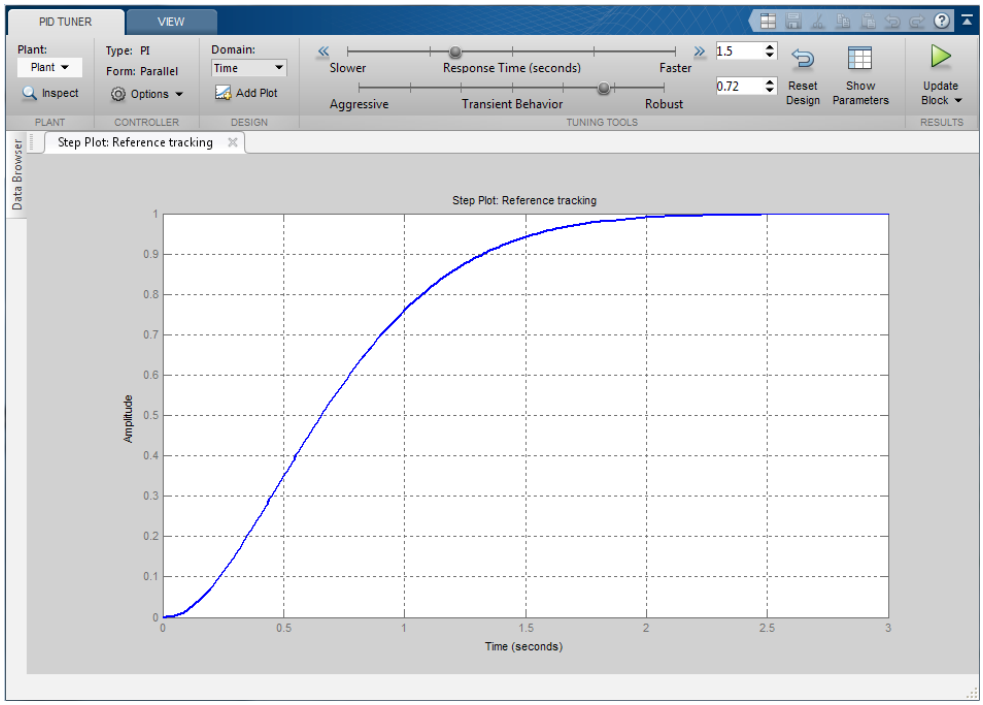
Performance and robustness

	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:



Controller parameters

	Tuned	Baseline
P	0.0014551	1
I	0.0043791	1
D		
N		

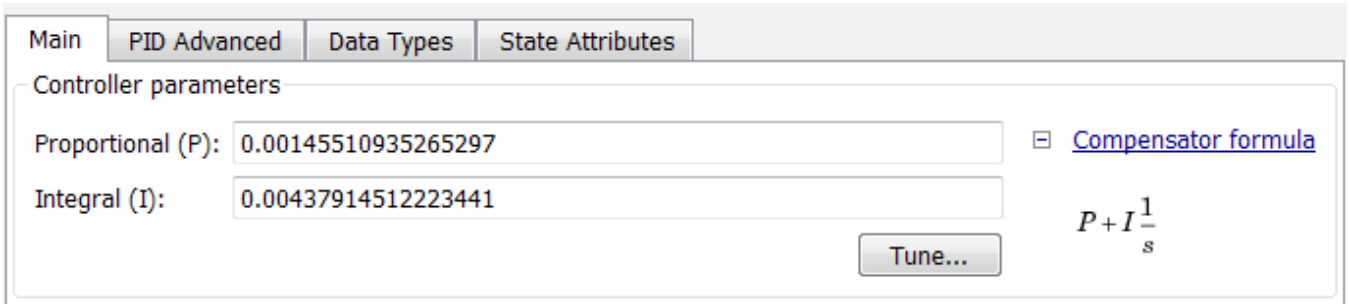
Performance and robustness

	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

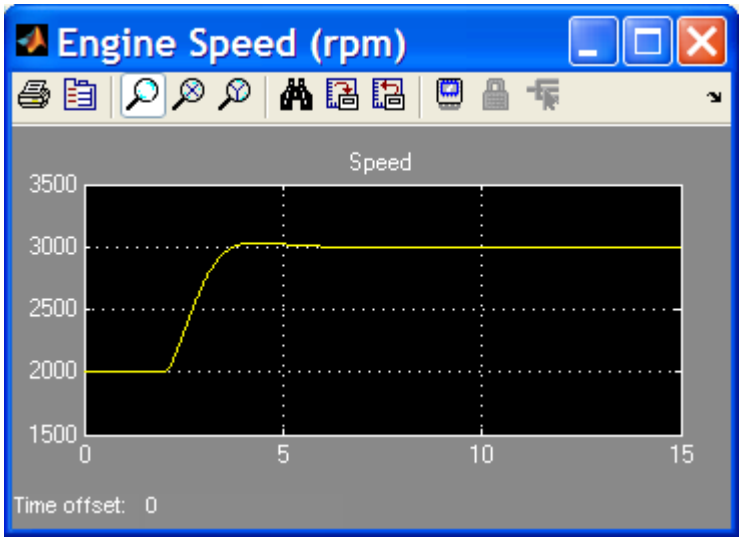
After you are happy with the controller performance on the linear plant model, you can test the design on the nonlinear model. To do this, click **Update Block** in the PID Tuner. This action writes the parameters back to the PID Controller block in the Simulink model.

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')
```