SIMULINK

Aircraft Roll Motion Autopilot Development

Table of Contents:

1 CONTROL SYSTEM DESIGN OBJECTIVES

Consider the aircraft roll transfer function $G(s) = \frac{K}{Js^2 + cs} = \frac{114}{10s^2 + 4s}$

Design a feedback control system to control the aircraft roll motion with the following control design objectives:

<u>Design Objective 1</u>: Control the unconstraint aircraft motion resulting from an aileron input. Have a autopilot system that can maintain the aircraft at a constant bank angle

We wish to achieve this objective through feedback (FB)

<u>Design Objective 2</u>: Achieve a reasonable aircraft roll response.

We define 'reasonable response' using two control design specifications:

- DS1: Fast response time as measured by rise time
  $t_r \leq 1.5$ sec
- DS2: maximum percentage overshoot for step input less than 20%
  $M_p \leq 20\%$

2 UNCONSTRAINED AIRCRAFT ROLL MOTION

2.1 MODEL

Open a new model canvas and save it as ‘SIMULINK_airplane_roll_unitFB’

From Simulink Library Browser drag onto the new canvas the following blocks:

- select ‘Step’ from ‘Sources’
- select ’Transfer Fcn’ from ‘Continuous’
- select ‘Scope’ from ‘Sinks’, enter ‘Number of inputs’ = 2

Create annotation text boxes above each box as follows:

- ‘reference signal’ above the ‘Step’ box
- ‘aircraft roll dynamics’ above ‘Transfer Fcn’ box
- ‘display’ above ‘Scope’

Change properties of ‘Transfer Fcn’ block to represent the aircraft roll transfer function $G(s) = \dfrac{K}{Js^2 + cs} = \dfrac{114}{10s^2 + 4s}$

Connect the blocks:

- ‘reference signal’ → ‘aircraft dynamics’ → ‘display’ (1st port)
- ‘reference signal’ → ‘display’ (2nd port)

Use the pull down menu: ‘Edit → Copy Current View to Clipboard → Metafile’ to capture only the model as shown in Figure 1.

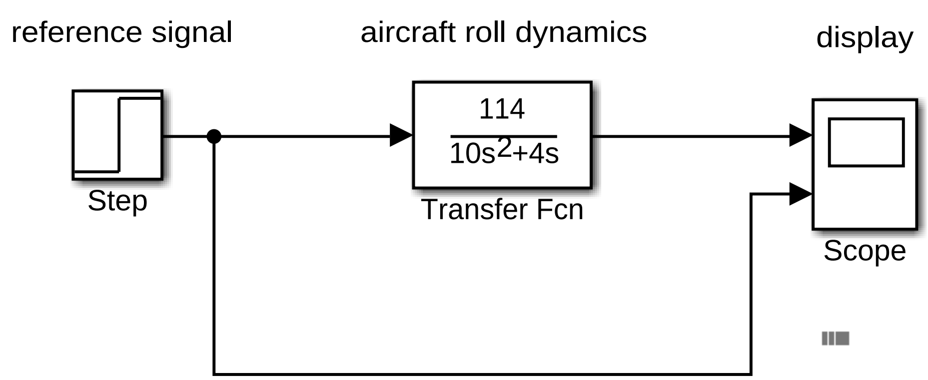


Figure 1

2.2 SIMULATION PARAMETERS

Use the pull down menu to open Configuration Parameters window, i.e., Simulation → Model Configuration Parameters

Make:

- Max step size: 1e-4
- Stop time: 2

The rest should remain unchanged (verify that they are the same as inFigure 2)

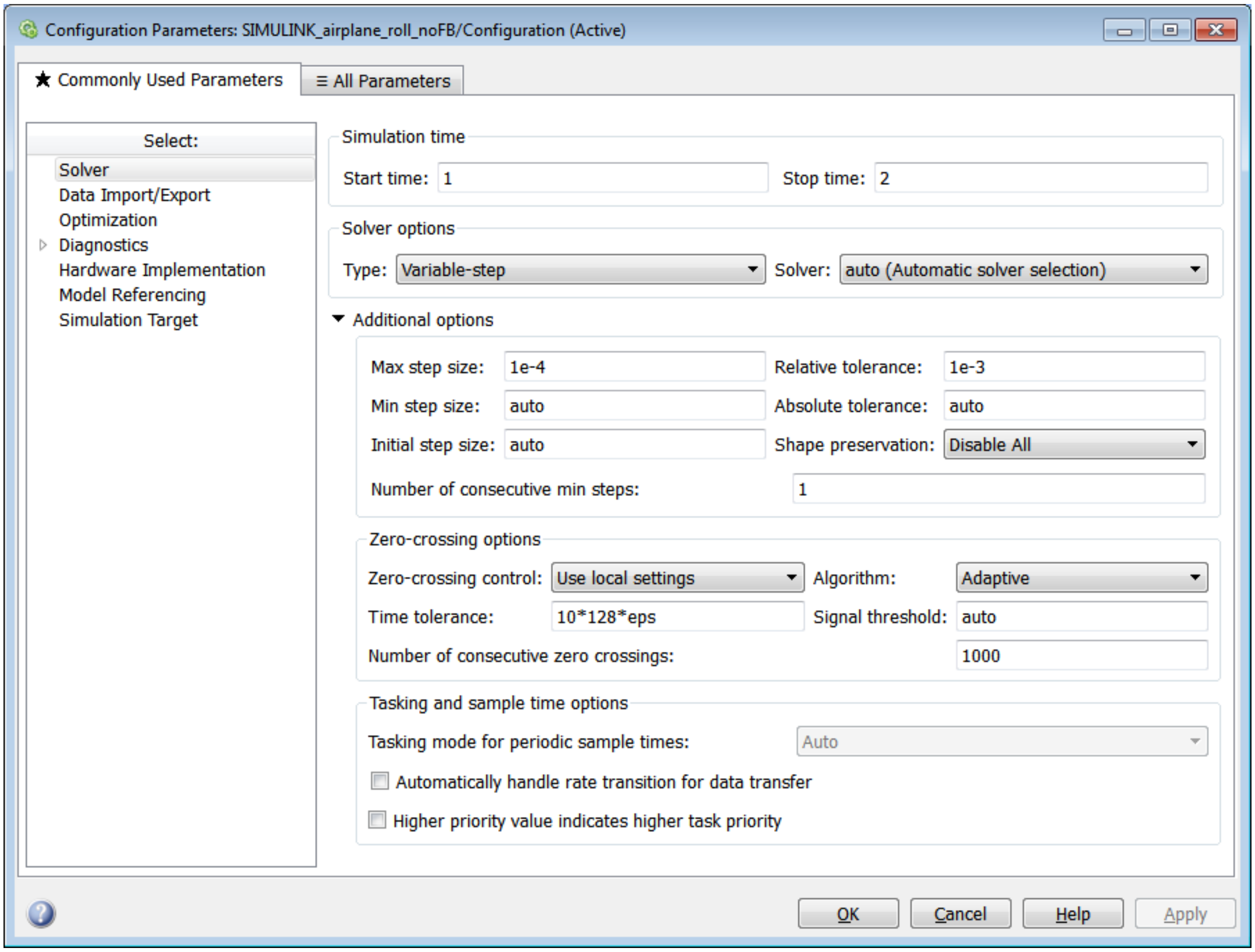


Figure 2

2.3 DISPLAY SETUP

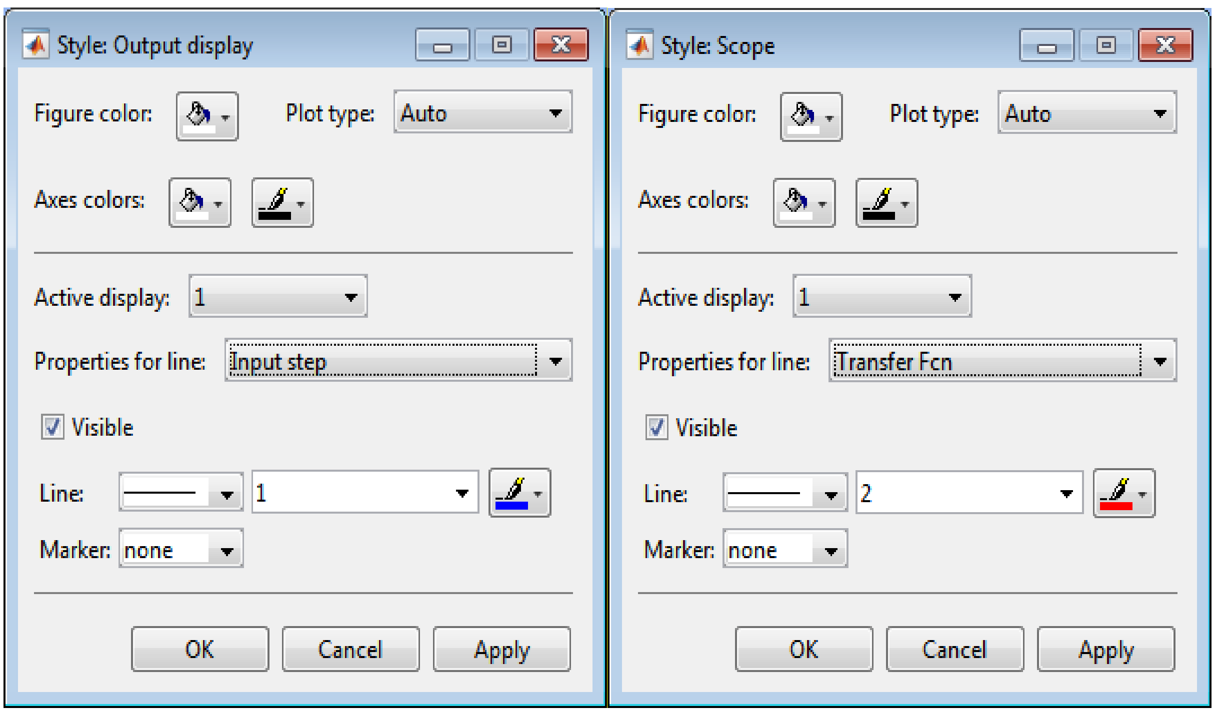
Double click on the ‘Scope’ block in the SIMULINK model. The ‘Scope’ display should open in a new window.

Select ‘View →Style’ to open ‘Style Scope’ dialog box (Figure 3a).
Choose color white for:

- Figure color: ‘bucket’
- Axes colors: ‘bucket’

Choose color black for Axes colors: ‘brush’

Choose color blue for ‘Properties for line: Input step’ (Figure 3a)
Choose color red for ‘Properties for line: Transfer Fcn’ (Figure 3b)



(a) (b)

Figure 3

The ‘Scope’ display should look as shown in Figure 4:

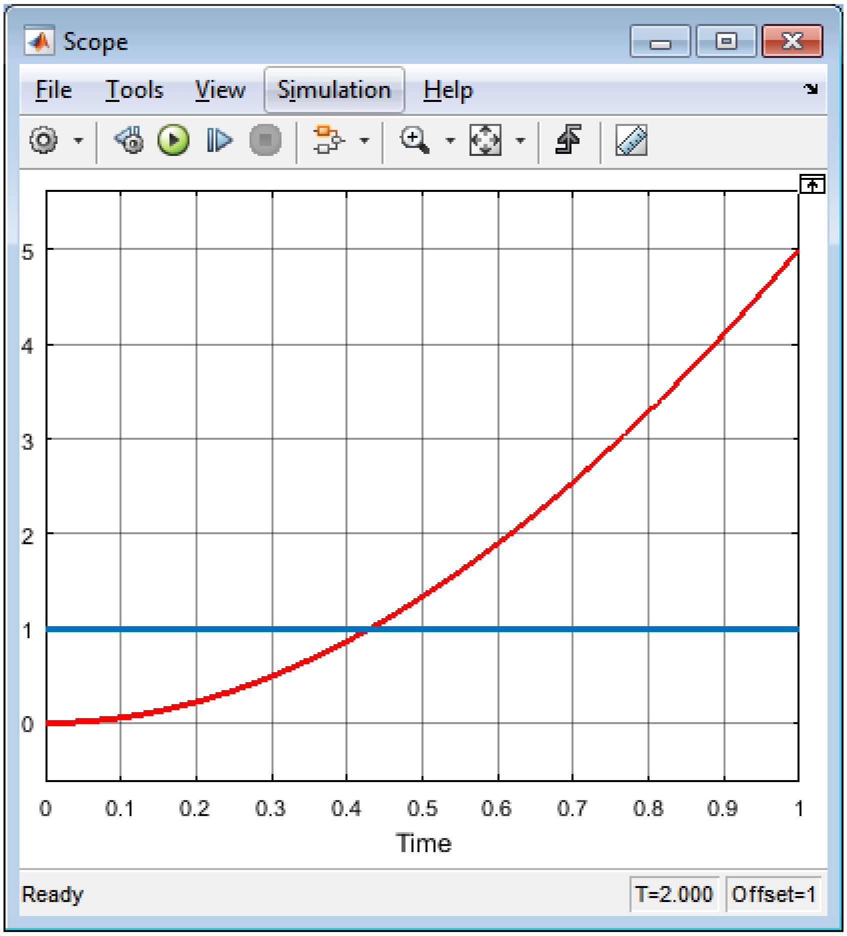


Figure 4

Note that the blue line represents the step input whereas the red line represent the aircraft roll response, which growth continuously. This situation is unacceptable. To counteract it, use feedback control as shown in next section.

3 UNIT FEEDBACK CONTROL OF AIRCRAFT ROLL MOTION

Open a new model canvas and save it as ‘SIMULINK_airplane_roll_unitFB’
From Simulink Library Browser drag onto the new canvas the following blocks:

- ‘Step’ from ‘Sources’
- ‘Sum’ from ‘Commonly Used Blocks’
- ‘Scope’ from ‘Sinks’
- ‘Transfer Fcn’ from ‘Continuous’

Create annotation text boxes above each box as follows:

- ‘reference signal’ above the ‘Step’ box
- ‘aircraft roll dynamics’ above ‘Transfer Fcn’ box
- ‘output signal’ above ‘Scope’

Change properties of ‘Transfer Fcn’ block to represent the aircraft roll transfer function $G(s) = \dfrac{114}{10s^2 + 4s}$

Change the second port of the ‘Sum’ block to negative (-). Do this by using ‘right-click’ → ‘Block Parameters: Sum’ and then putting ‘+-‘ in ‘List of signs’

Connect the blocks:

- ‘reference signal’ → ‘Sum’ → ‘aircraft dynamics’ → ‘display’
- output from ‘aircraft dynamics’ to negative (-) port of the ‘Sum block’ (this is the unit feedback closed loop)

Use the pull down menu: ‘Edit → Copy Current View to Clipboard → Metafile’ to capture only the model (Figure 5).

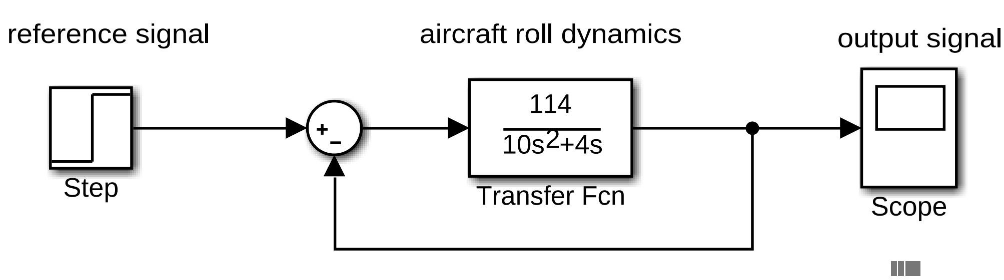


Figure 5

3.1 SIMULATION PARAMETERS

Use the pull down menu to open Configuration Parameters window, i.e.,
Simulation → Model Configuration Parameters
Make:

- Max step size: 1e-4
- Stop time: 40

The rest should remain unchanged (verify that they are the same as in the figure)

3.2 DISPLAY SETUP

Double click on the ‘Scope’ block in the SIMULINK model. The ‘Scope’ display should open in a new window.

Select ‘View →Style’ to open ‘Style Scope’ dialog box.
Choose color white for:

- Figure color
- Axes colors: bucket

Choose color black for Axes colors: brush
‘Properties for line: Transfer Fcn’: 1.5 weight, red (Figure 6).

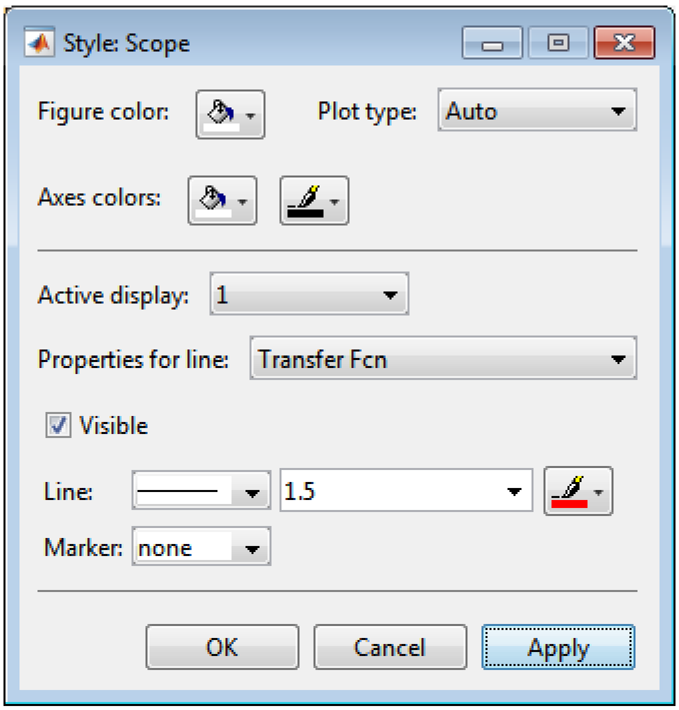


Figure 6

3.3 Aircraft Roll Response with Unit FB

Press the 'Run' button. The aircraft roll response with unit FB is displayed (Figure 7).

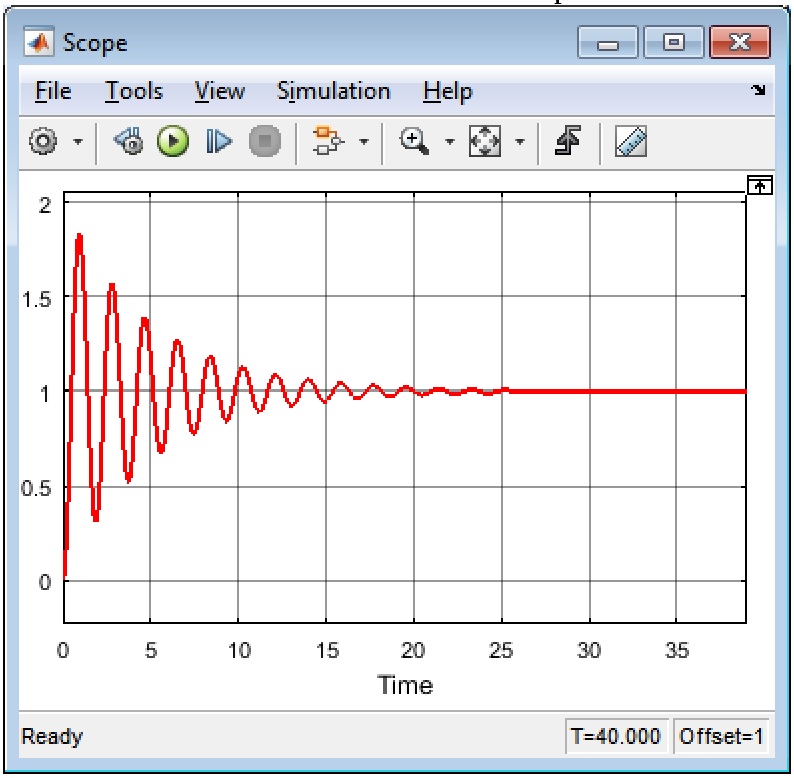


Figure 7

It is apparent that the response is unsatisfactory because:

- many oscillations until it settles down to $x_{ss} = 1$
- large overshoot ($x_p \approx 1.8$, $M_p \approx 80\%$)

4 VARIABLE GAIN FB CONTROL OF AIRCRAFT ROLL MOTION (P CONTROL)

To improve the aircraft FB response, we can try to use variable gain control. This is known as proportional control or 'P control'.

4.1 P-CONTROLLED AIRCRAFT ROLL MODEL

Add a gain box to the model to obtain the P-controller (Figure 8).

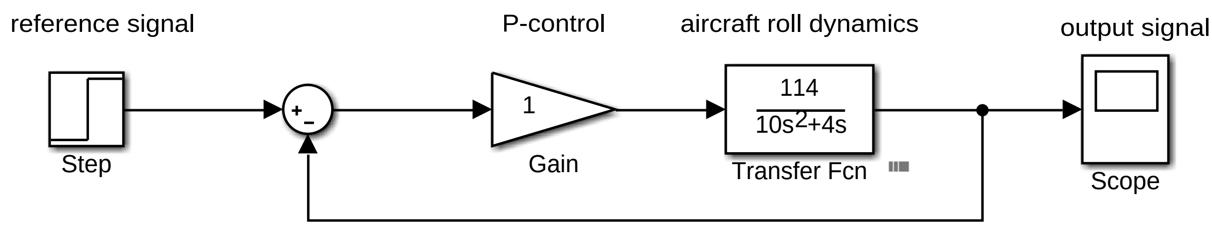


Figure 8

Save this model as 'SIMULINK_aircraft_roll_P_control'

4.2 DISPLAY SETUP

Have the display setup as before.
In addition, stop automatic axes scaling by doing the following:

- Tools → Axes Scaling → Axis Scaling Properties

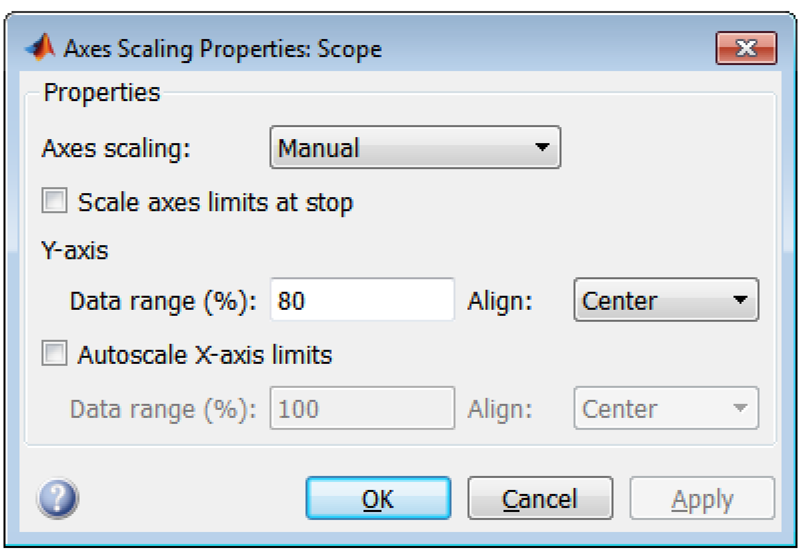


Figure 9

4.3 P-Control Gain Changes

The P-control gain Kp can be modified as follows:

Right click 'Gain' box, select 'Block Parameters'. The 'Block Parameters: Gain' dialog box open (Figure 10).

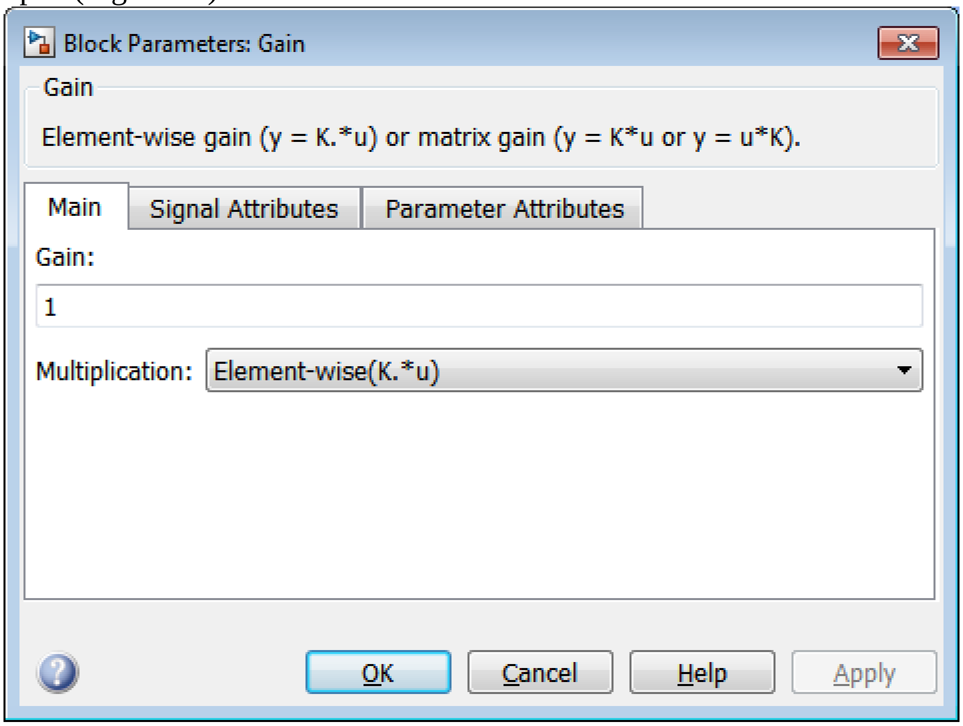


Figure 10

The gain default value is 'Gain: 1'. , i.e., Kp=1. After entering a new gain value, press 'Apply'. Keep the dialog box open for the next gain value. (simulation runs OK with the dialog box open.)

Try a number of different gains to see their effect on the response. Gains to be tried are:

- Kp=1
- Kp=10
- Kp=0.1
- Kp=0.01

4.3.1 Kp=1 Response

Put 'Gain = 1' in 'Block Parameters: Gain' dialog box and press 'Apply'. The response is shown in Figure 11.

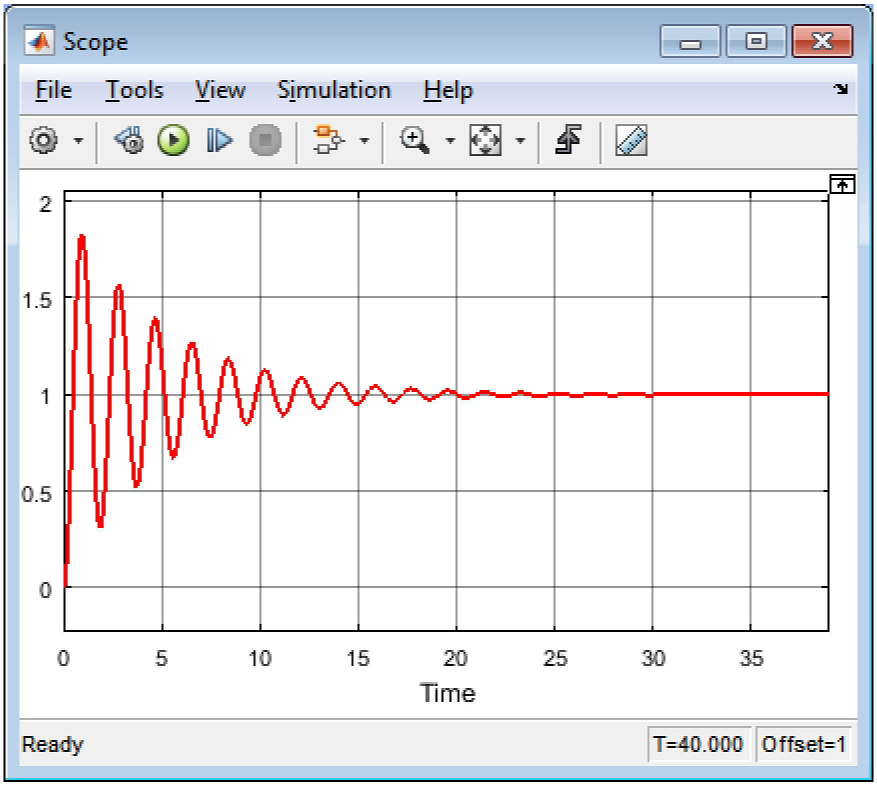


Figure 11

This Kp=1 case corresponds to Unit FB which we have already studied. It is unsatisfactory because

- many oscillations until it settles down to $x_{ss} = 1$
- large overshoot ($x_p \approx 1.8$, $M_p \approx 80\%$)

4.3.2 Kp=10 Response

Put 'Gain = 10' in 'Block Parameters: Gain' dialog box and press 'Apply'. The response is shown in Figure 12.

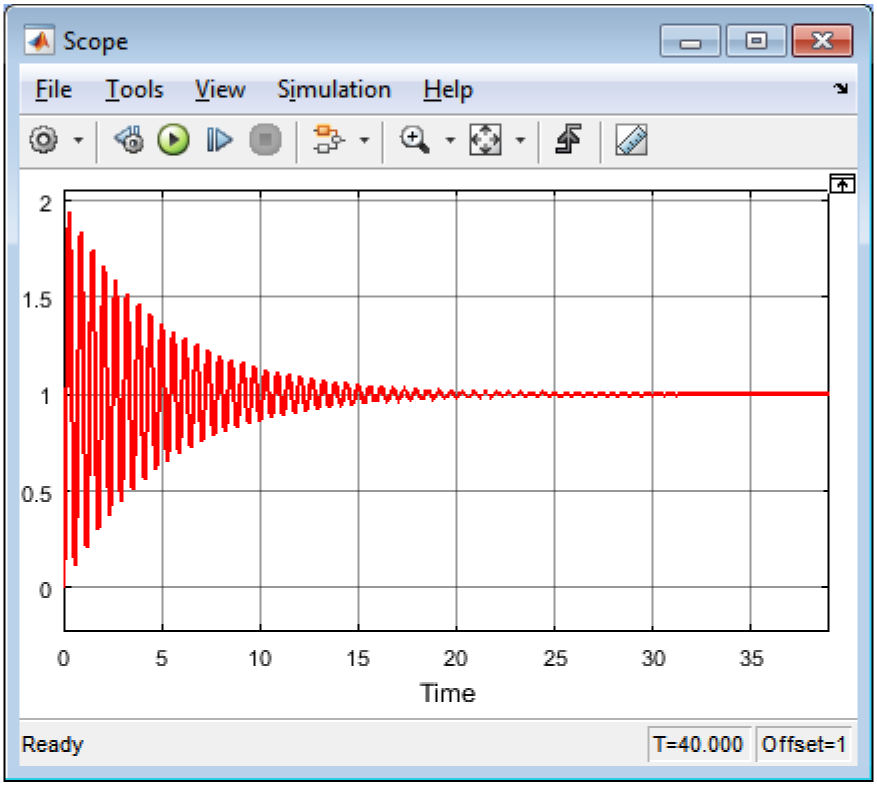


Figure 12

This Kp=10 response is unsatisfactory because

- very many oscillations until it settles down to $x_{ss} = 1$
- large overshoot ($x_p \approx 2$, $M_p \approx 100\%$)

4.3.3 Kp=0.1 Response

Put ‘Gain = 0.1’ in ‘Block Parameters: Gain’ dialog box and press ‘Apply’. The response is shown in Figure 13.

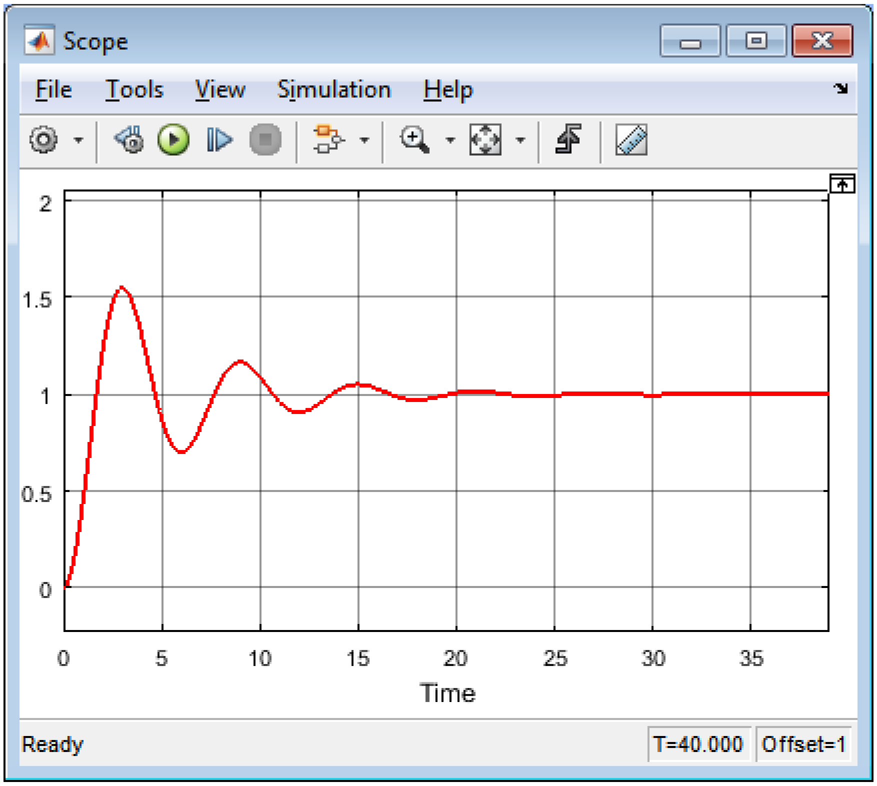


Figure 13

The response for Kp=0.1 seems better because:

- fewer oscillations until it settles down to $x_{ss} = 1$
- smaller overshoot ($x_p \approx 1.55$, $M_p \approx 55\%$)

4.3.4 Kp=0.01 Response

Put ‘Gain = 0.01’ in ‘Block Parameters: Gain’ dialog box and press ‘Apply’. The response is shown in Figure 14.

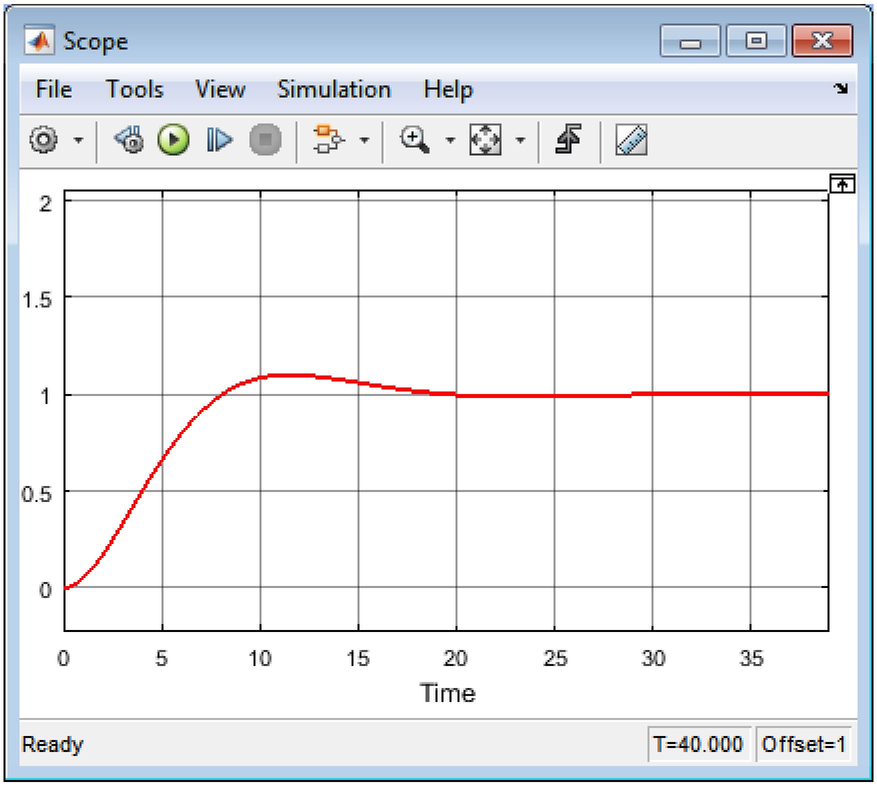


Figure 14

The response for Kp=0.01 seems even better because:

- almost no oscillation until it settles down to $x_{ss} = 1$
- much smaller overshoot ($x_p \approx 1.1$, $M_p \approx 10\%$)

However, the rise time is much longer than before ($t_r \approx 8$ sec). This is unacceptable because it makes the aircraft very sluggish.

We need to examine the mathematics of P-control. Recall:

$$\omega_n = \sqrt{\frac{K}{J}}\ ,\ \zeta = \frac{c}{2\sqrt{JK}}$$

where K is the overall forward gain (K=Kp*K1, with K1=114 for our aircraft roll model). Hence we can calculate:

- rise time: $t_r = \dfrac{\pi - \varphi}{\omega_n \sqrt{1-\zeta^2}}$ where $\varphi = \sin^{-1}\sqrt{1-\zeta^2}$
- overshoot: $M_p = e^{-\pi \frac{\zeta}{\sqrt{1-\zeta^2}}}$

A plot of these values is given in Figure 15.

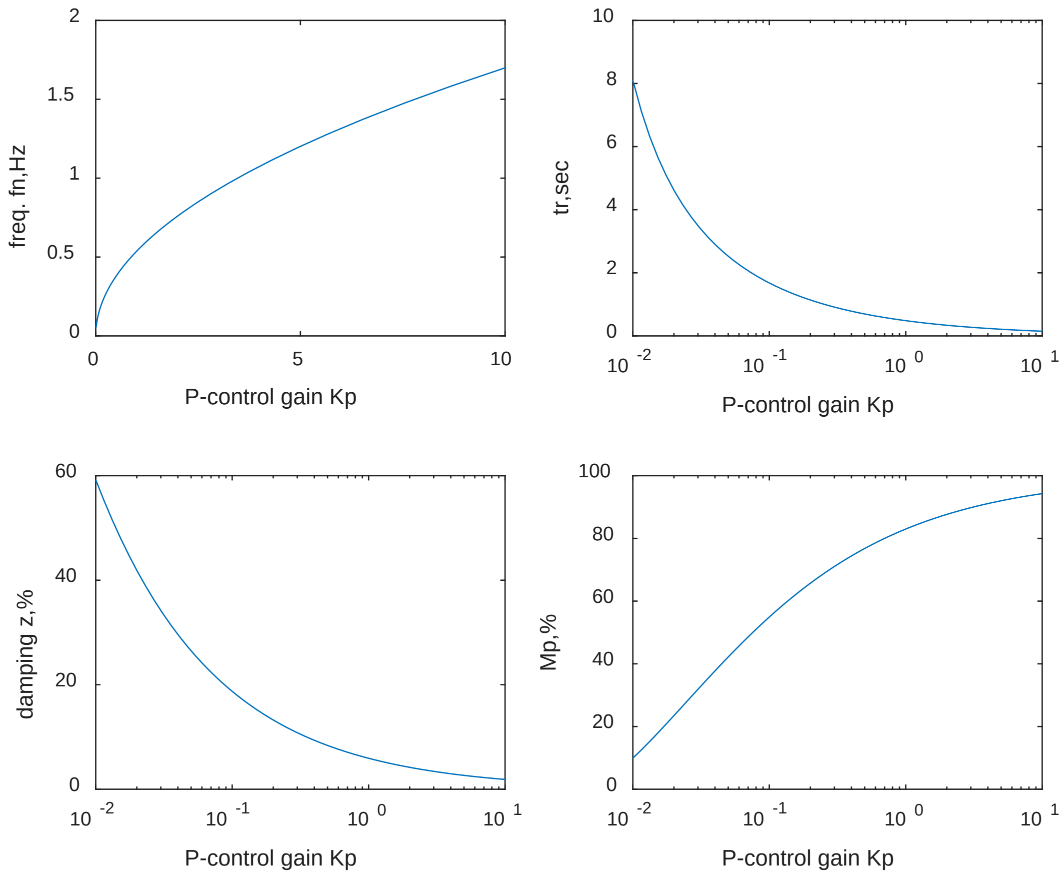


Figure 15

The above plot shows that small overshoot M_p happens at low gain values. However, at these low gain values, the rise time t_r becomes large. Hence, we conclude that **P-control gain has an opposite effect on the two performance indicators, M_p and t_r considered in this controller design.** We need to explore other control options.

5 PI CONTROL

PI control stands for ‘proportional + integrative control’.

5.1 PI CONTROL SETUP

The aircraft model with PI control has two branches, one for the P block (which is just a simple gain) and the other for the I block which consists of a gain followed by an integrator. The PI controlled aircraft model looks as shown in Figure 16.

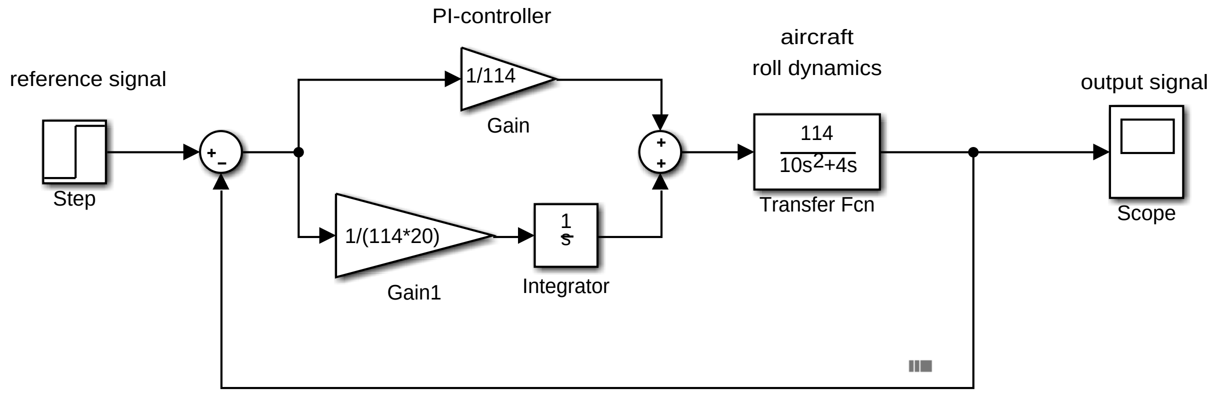


Figure 16

Construct this model and save it as ‘SIMULINK_aircraft_roll_PI_control_Kp_Ki’

Observe that the model has two gains:

- Proportional gain, Kp=1/114
- Integrative gain Ki=1/(114*20)

The resulting PI controller has the expression:

$$G_{PI}(s) = K_p + K_i \frac{1}{s}$$

5.2 AIRCRAFT ROLL RESPONSE WITH PI CONTROL

Run the PI control model. The resulting response will show up in the 'Scope' window (Figure 17):

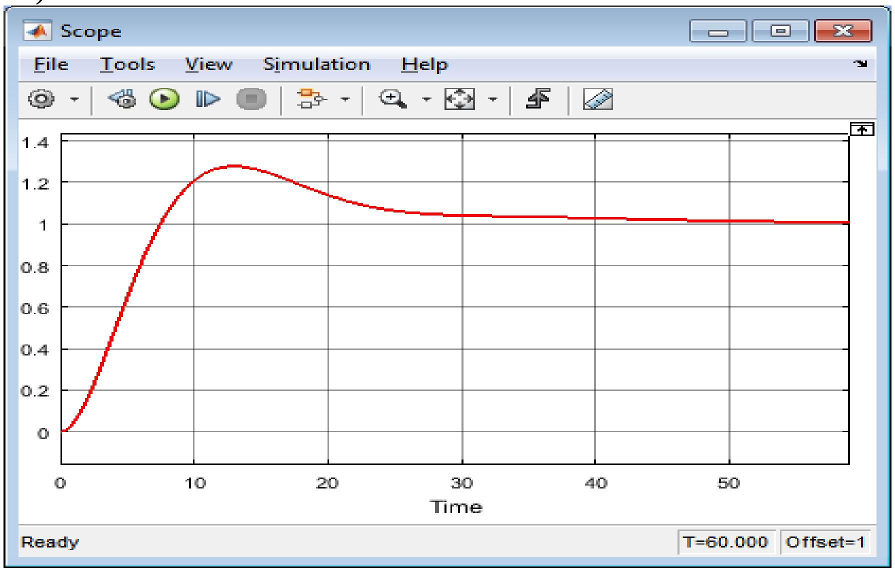


Figure 17

Press the 'Cursor Measurements' button and set the cursors to measure rise time and overshoot (Figure 18).

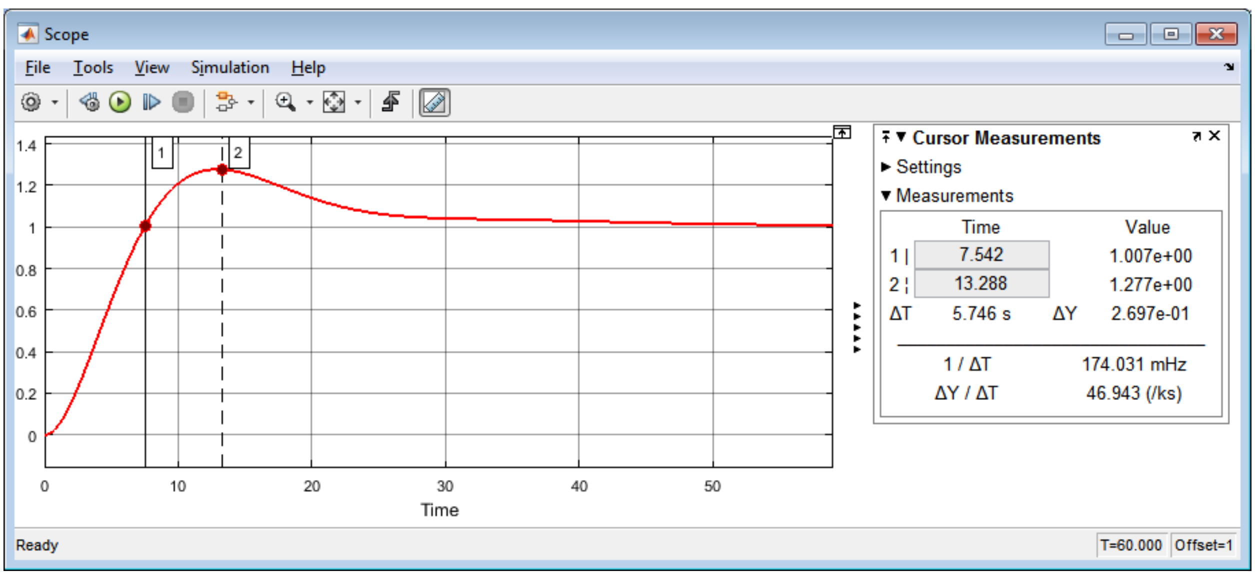


Figure 18

The results are:

- rise time $t_r = 7.542$ sec
- overshoot $M_p = 27.7\%$

These values are better than the values obtained with P-control. However, they are still below the target values. Need to try something else.

6 PD CONTROL

PD control stands for ‘proportional + derivative control’.

6.1 PD CONTROL SETUP

The aircraft model with PI control has two branches, one for the proportional P-block (which is just a simple gain) and the other for the derivative D-block. The PD controlled aircraft model looks as shown in Figure 19.

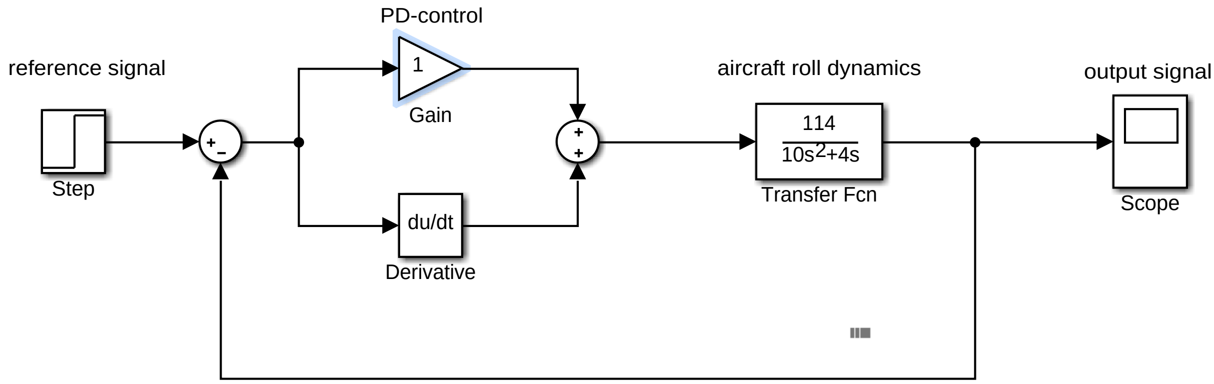


Figure 19

Construct this model and save it as ‘SIMULINK_aircraft_roll_PD_control’

6.2 AIRCRAFT ROLL RESPONSE WITH PD CONTROL

Set the model run time to 20 sec and run it. The response is as shown in Figure 20.

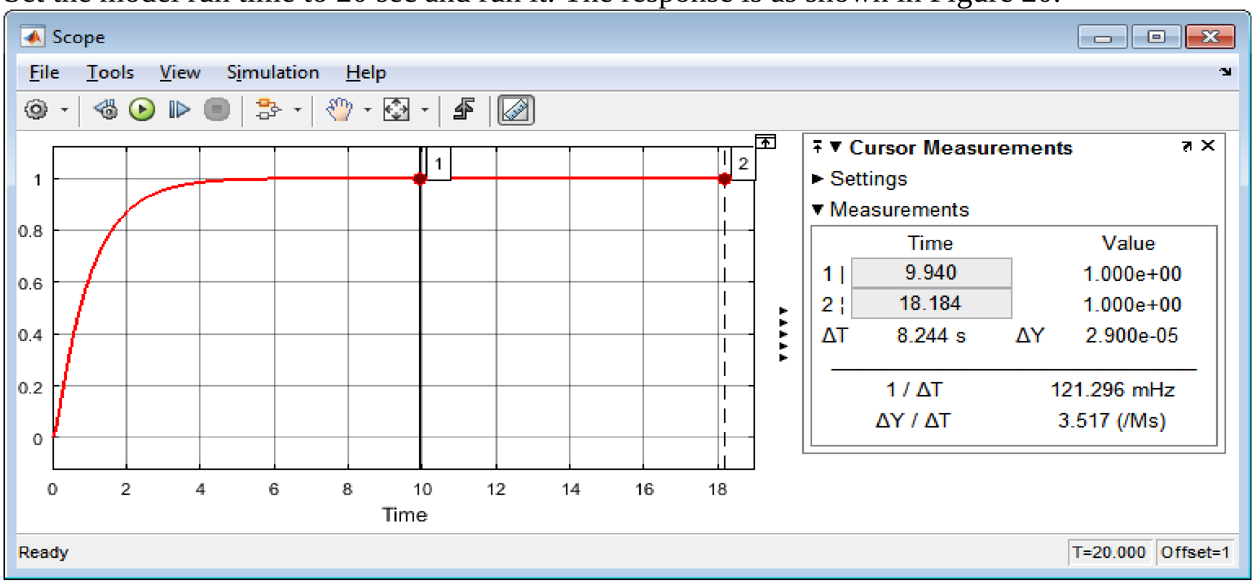


Figure 20

The response convergence continuously towards the target value 1, but the rise time is around 10 sec.

This is unacceptable. But we can now try different value of the P gain Kp.

Increase P control gain to Kp=10. The model looks as shown in Figure 21.

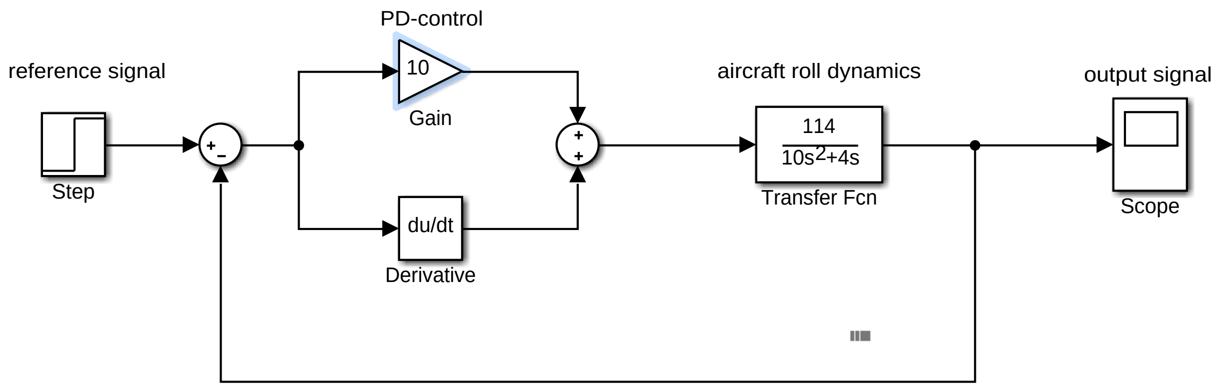


Figure 21

The response of this model with Kp=10 is as shown in Figure 22.

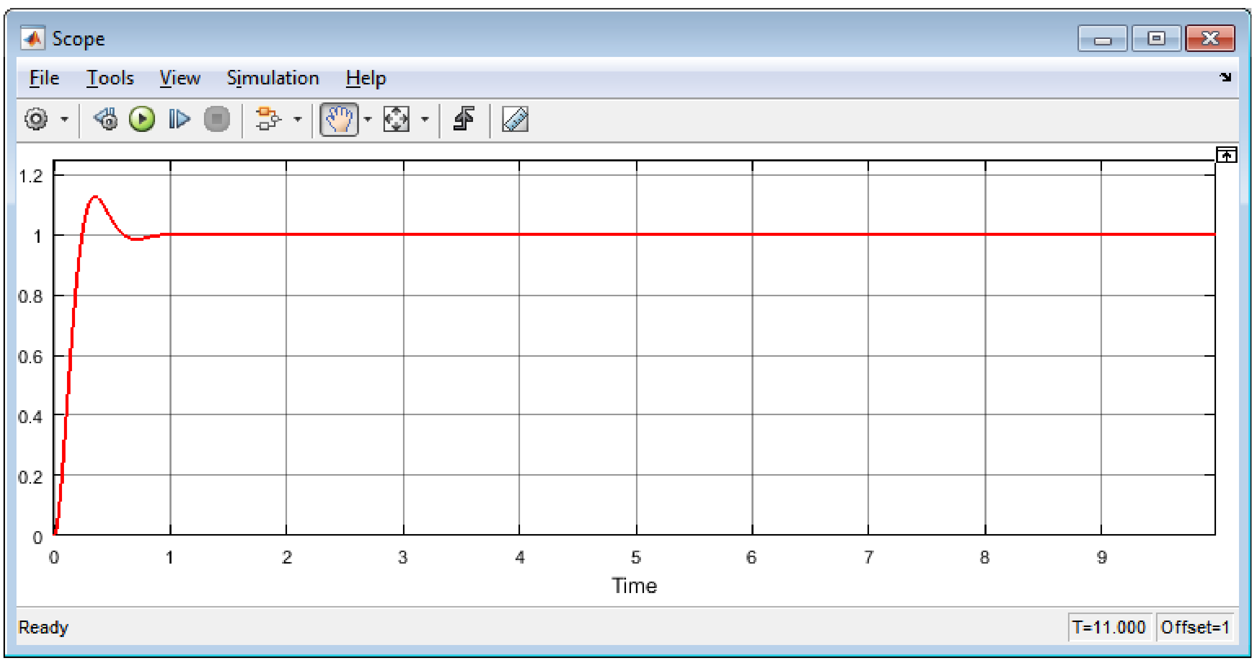


Figure 22

Note that the response has become much crispier but overshoot is now present.

Press the ‘Cursor Measurements’ button to get measurements (Figure 23).

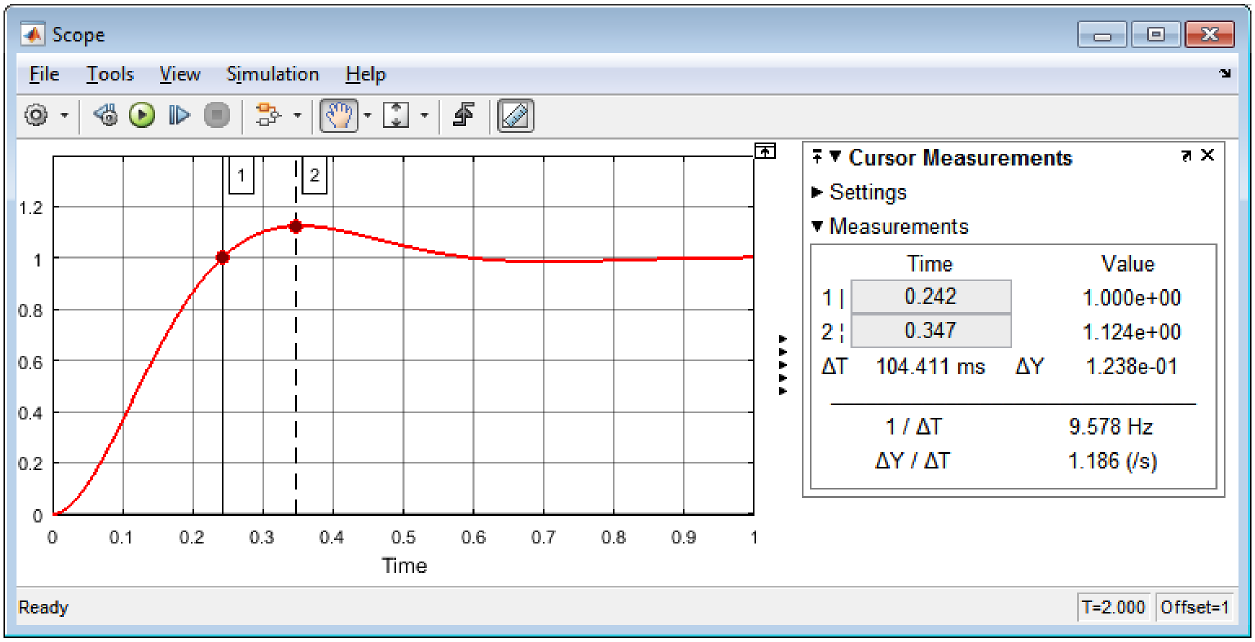


Figure 23

The cursor measurements allow us to determine $t_r = 0.242$ sec and $M_p = 12.4\%$.

Conclusion:

PD control with Kp=10 seems to give a reasonable response:

- short rise time $t_r = 0.242$ sec
- small overshoot $M_p = 12.4\%$

Note of caution: I cannot reproduce these results exactly in MATLAB control systems toolbox. It is perhaps because PD control is not a stable control. In practice, the D part of a PD controller is difficult to implement because differentiation is not easy to obtain with electrical system; therefore, the D part of the controller is implemented as a compensator $s/(T_1 s+1)$.

7 PID CONTROL

PID control stands for ‘proportional + integrative + derivative control’.

7.1 PID CONTROL SETUP

The aircraft model with PID control has two branches, uses the PID block which can be found in ‘Continuous’ part of the library. The PI controlled aircraft model looks as shown in Figure 24.

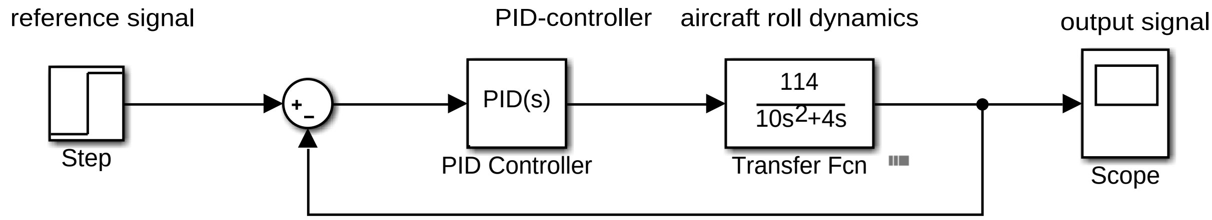


Figure 24

Construct this model and save it as ‘SIMULINK_aircraft_roll_PID_control’.
The PID block represents the following controller transfer function:

$$G_c(s) = K_P + \frac{K_I}{s} + K_D s$$

The default values at startup are $K_p = 1,\ K_I = 1,\ K_D = 0$. The response with these default values is as shown in Figure 25.

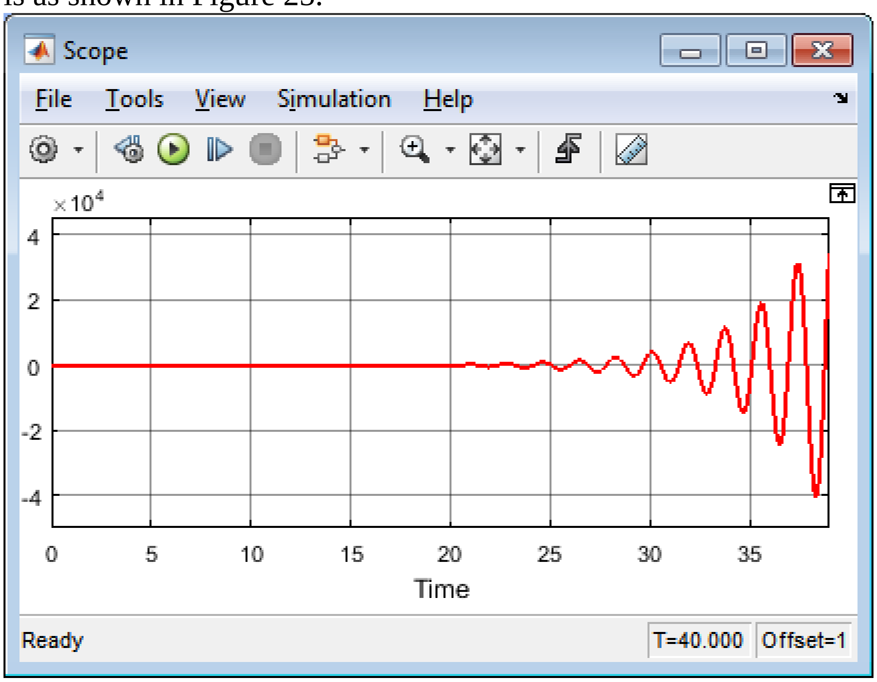


Figure 25

This is an unstable response and needs to be corrected! We will adjust the PID controller parameters to obtain a stable response which meets the design specifications. This process of adjustment is called ‘tuning’.

7.2 Automatic Tuning of the PID Controller

Double click the PID box and open it (Figure 26).

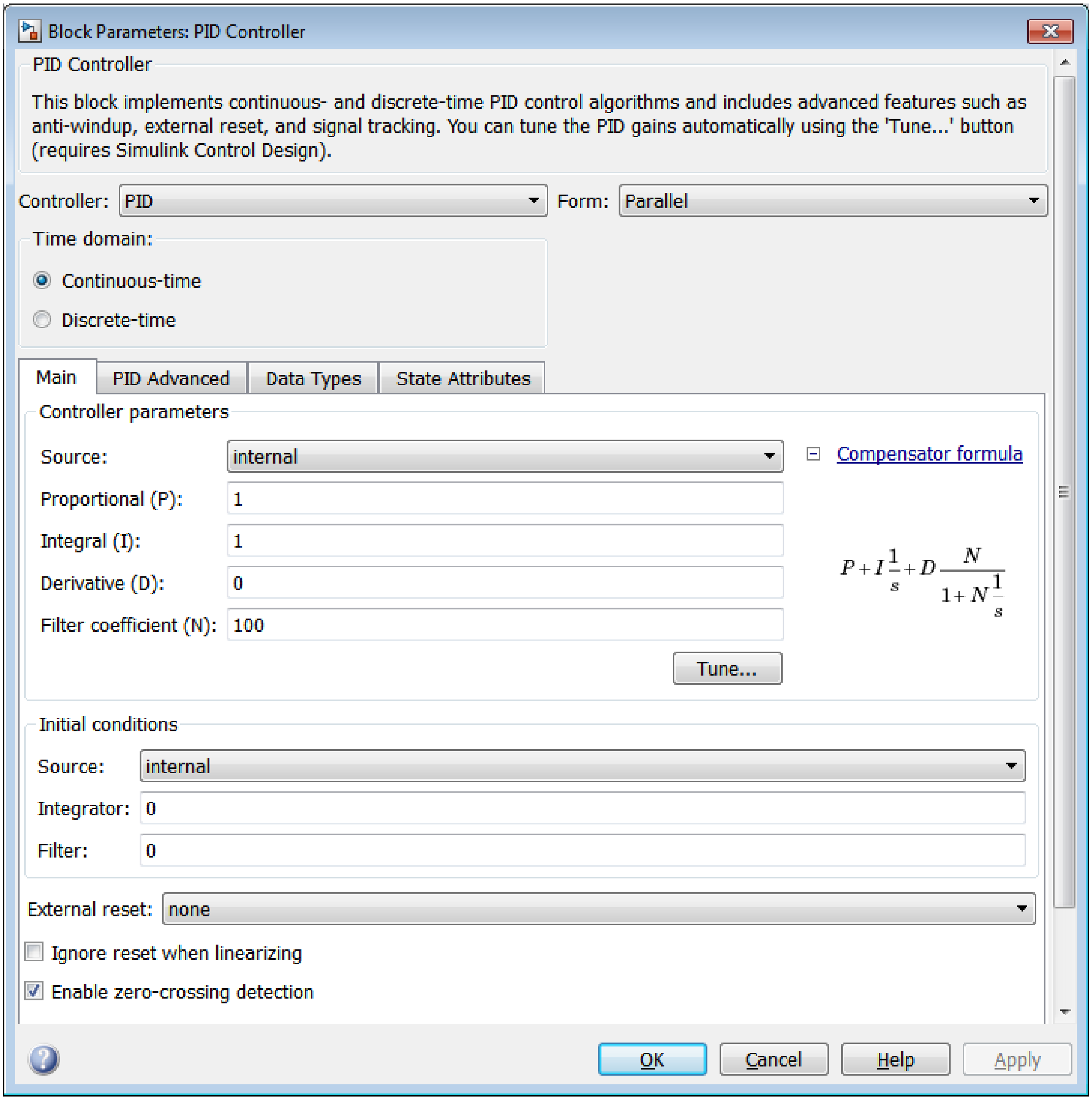


Figure 26

Next, press the 'Tune…' button. This will open a separate window for tuning, i.e., the 'PID Tuner' window shown in Figure 27.

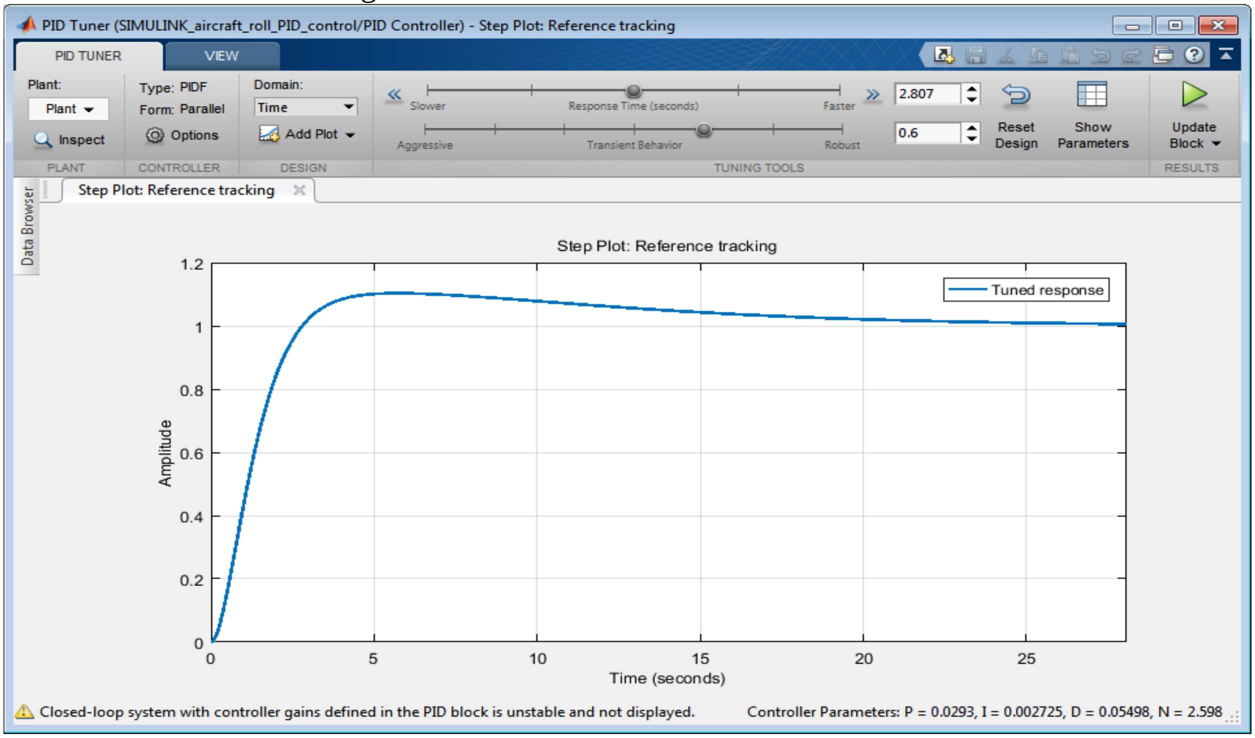


Figure 27

Note the message at the bottom of the window in the left corner: 'Closed-loop system with controller gains defined in the PID block is unstable and not displayed'.

In the right bottom corner, we read: 'Controller Parameters: P=0.0293, I=0.002725, D=0.05498, N=2.598'. These controller parameters have been obtained by the tuner through an internal algorithm.

Press the 'Show parameters' button in the upper right corner to open Figure 28.

Controller Parameters

	Tuned
P	0.029305
I	0.0027248
D	0.054976
N	2.598

Performance and Robustness

	Tuned
Rise time	1.85 seconds
Settling time	20.3 seconds
Overshoot	10.3 %
Peak	1.1
Gain margin	-Inf dB @ 0 rad/s
Phase margin	69 deg @ 0.712 rad/s
Closed-loop stability	Stable

Figure 28

In this window, we read Rise time = 1.85 sec and Overshoot = 10.3%. Note that MATLAB defines the rise time in the 10%--90% range To evaluate the 0--100% rise time t_r,. read the upper-right side window next the ‘Response Time (seconds) Faster’. Here, we read $t_r = 2.807$ sec.

It is apparent that the overshoot is within specifications but the rise time is to long. Need to do some tuning adjustments manually.

7.3 Manual Adjustment of the PID Tuner

To tune the system manually, use the ‘Response time (seconds)’ slider. When the slider is moved to the right, the response time becomes faster.

Under ‘Options’, check the box ‘Show Block Response’ to display the original response (‘Block response’) besides the ‘Tuned response’.

Move the slider until the time t_r shown in the small window on the right is less than 1.5 sec (Figure 29).

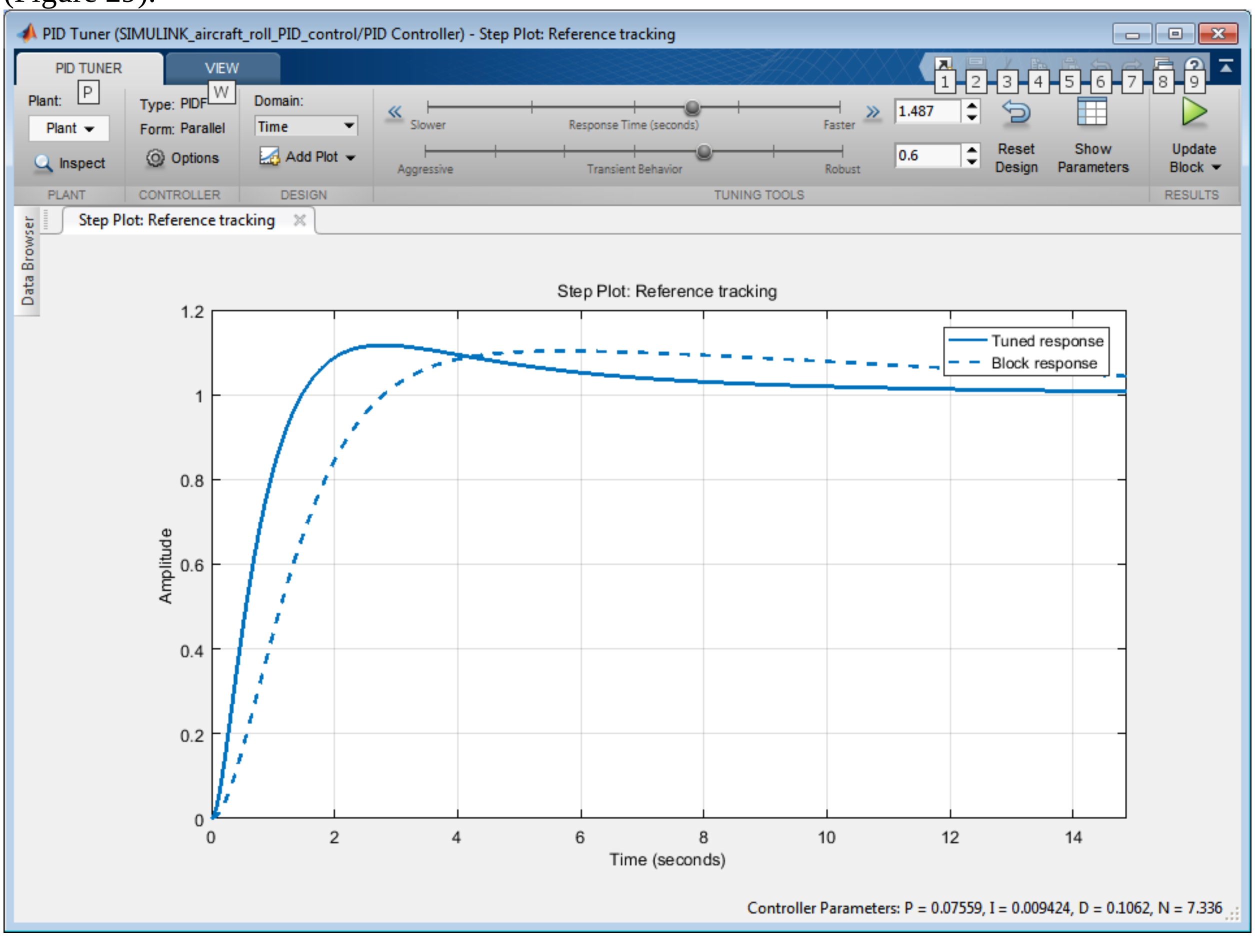
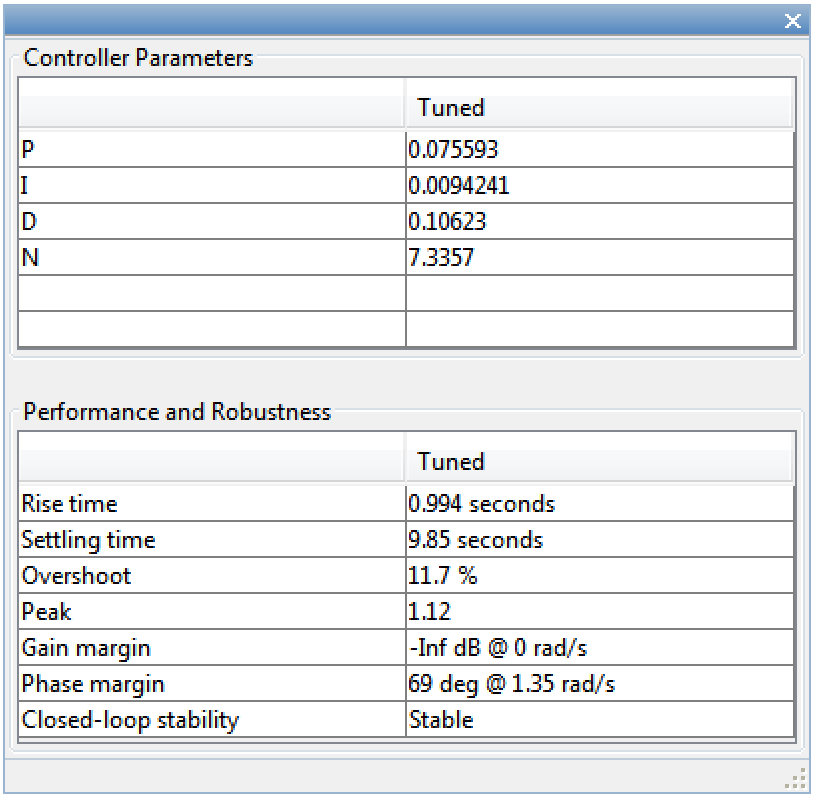


Figure 29

We read $t_r = 1.487$ sec

At this moment, the ‘Block Parameters PID’ window looks as shown in Figure 30.



Controller Parameters

	Tuned
P	0.075593
I	0.0094241
D	0.10623
N	7.3357

Performance and Robustness

	Tuned
Rise time	0.994 seconds
Settling time	9.85 seconds
Overshoot	11.7 %
Peak	1.12
Gain margin	-Inf dB @ 0 rad/s
Phase margin	69 deg @ 1.35 rad/s
Closed-loop stability	Stable

Figure 30

In this window, we read ‘Overshoot = 11.7%’ which means $M_p = 11.7\%$ which is less than the required 20%.

It is now apparent that the design specification have been met. The final PID settings are read from the ‘Block Parameters PID’ window as:

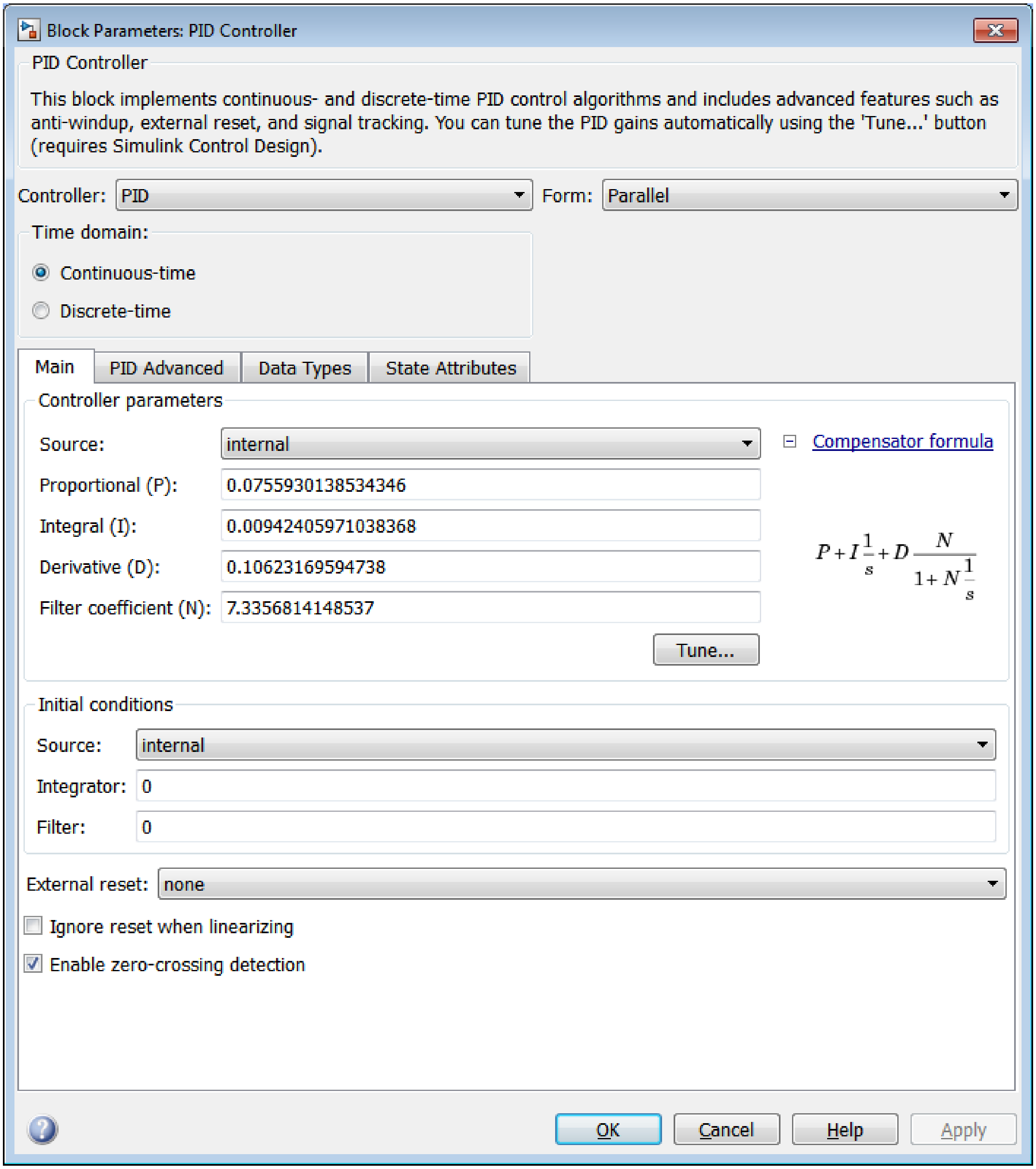
- ‘P= 0.075593’, i.e., Kp = 0.075593
- ‘I = 0.0094241’, i.e., Ki = 0.0094241
- ‘D = 0.10623’, Kd = 0.10623
- N = 7.3357

The aircraft roll response now meets the design specifications, i.e.,

- DS1: Fast response time as measured by rise time
  $t_r = 1.487 < 1.5$ sec
- DS2: maximum percentage overshoot for step input less than 20%
  $M_p = 11.7 < 20\%$

7.4 Verification of the PID Tuning Process

In the ‘PID Tuner’ window, press the ‘Update Block’ green arrow. Look in the ‘Block Parameters: PID Controller’ window to verify that the PID parameters have been updated. The window should look like Figure 31.


Block Parameters: PID Controller
PID Controller
This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).
Controller: PID
Form: Parallel
Time domain:
Continuous-time
Discrete-time
Main
PID Advanced
Data Types
State Attributes
Controller parameters
Source: internal
Compensator formula
Proportional (P): 0.0755930138534346
Integral (I): 0.00942405971038368
Derivative (D): 0.10623169594738
Filter coefficient (N): 7.3356814148537
$P+I\frac{1}{s}+D\frac{N}{1+N\frac{1}{s}}$
Tune...
Initial conditions
Source: internal
Integrator: 0
Filter: 0
External reset: none
Ignore reset when linearizing
Enable zero-crossing detection
OK
Cancel
Help
Apply
Figure 31

Now, reduce the SIMULINK run time to 20 sec and run the model (Figure 32).

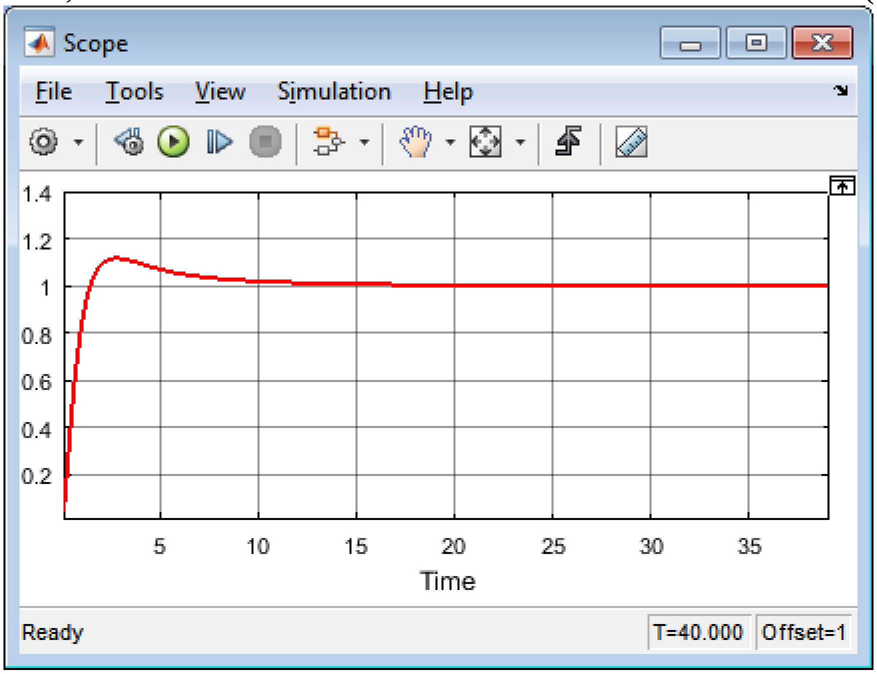


Figure 32

Press the ‘Cursor Measurements’ button. The window should look like Figure 33.

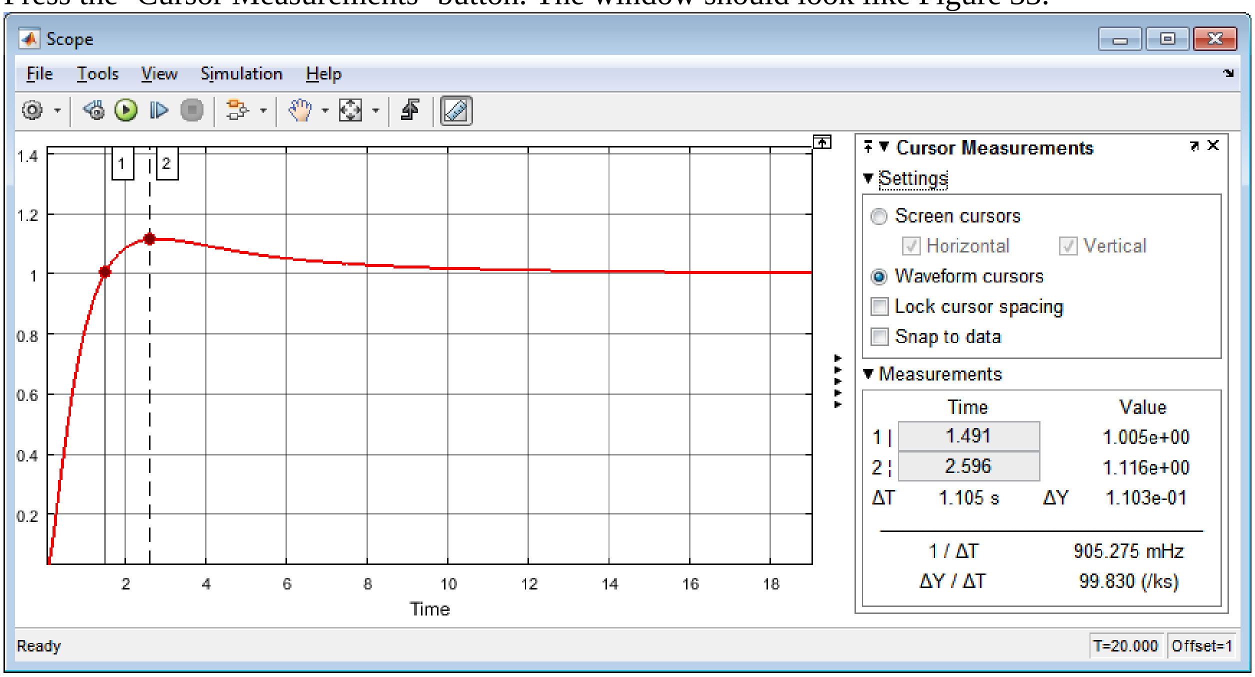


Figure 33

Measure the rise time and overshoot.

- $t_r = 1.483 \leq 1.5$ sec
- $M_p = 11.6 \leq 20\%$

It apparent that the two design specifications are satisfied.

To verify the MATLAB 10%--90% rise time of 0.994 seconds indicated in the previously shown 'Controller Parameters' window, try to measure this rise time in the 'Scope' window.

To do this measurement, we need higher resolution; hence, reduce the run time to 5 sec and run again. The Scope window should look like Figure 34.

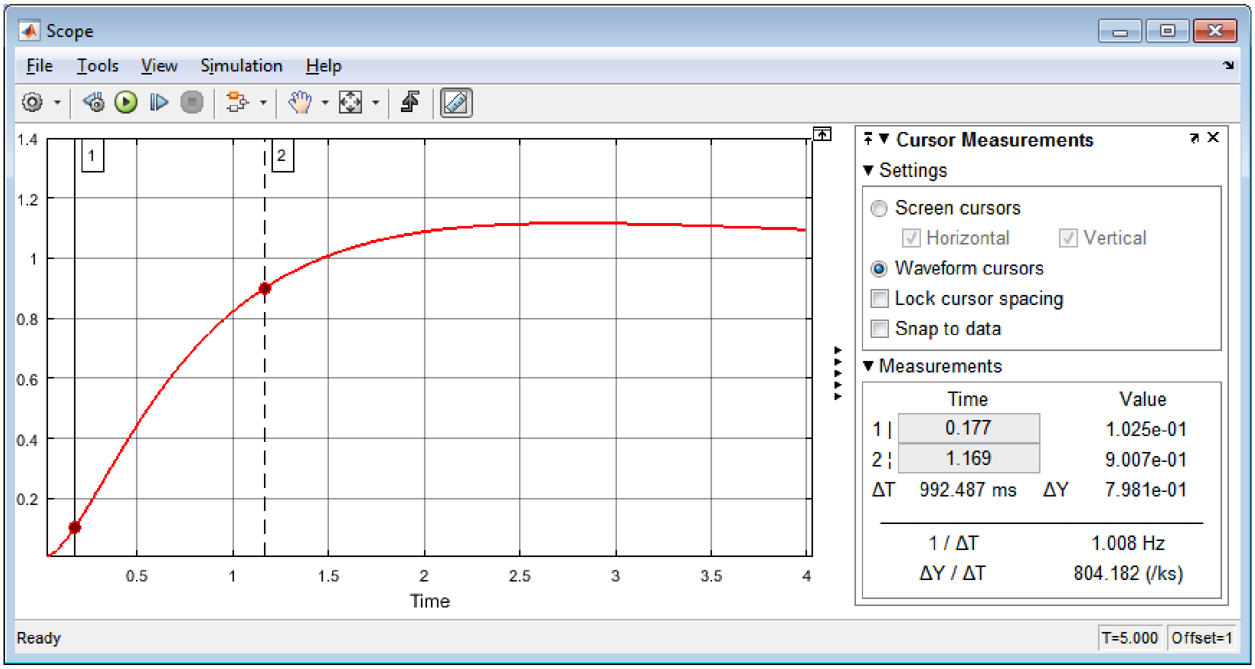


Figure 34

The cursors 1 and 2 are used to identify the 10% and 90% points on the curve. To determine the rise time from this image, read the time difference between the two cursors, i.e., $\Delta T = 992.487 \text{ ms}$. This is the rise time between 10% and 90%. It agrees with the 0.994 seconds indicated in the previously shown 'Controller Parameters' window.