Table of Contents

Title:Control System-Second Order System

```
%Author:ShivaKumar Naga Vankadhara
%PS No:99003727
%Date:12/04/2021
%Version:1.0
```

This Document has equation for DC Motor

```
%Equation:Tdm/dt+m=tem
%T_F=1/Ts+1
```

Math analysis

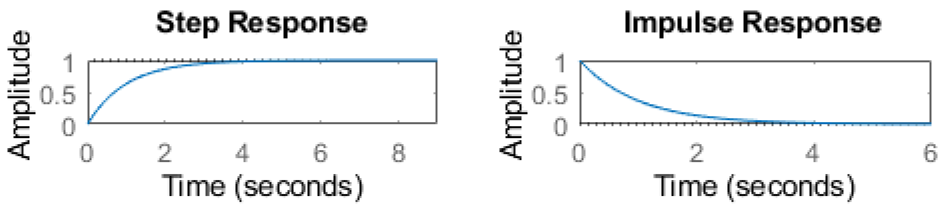
```
%dependent variables:m,temp
%independent variables:t
%constant:T
%Roots:-1/T
```

Basic

```
T=1
sys1 = tf([1],[T,1])
subplot(5,2,1)
step(sys1)
subplot(5,2,2)
impulse(sys1)
S = stepinfo(sys1)
p1=pole(sys1)
z1=zero(sys1)
```

```
T =
```

```
     1


sys1 =

    1
  -----
  s + 1

Continuous-time transfer function.


S =

  struct with fields:

        RiseTime: 2.1970
    SettlingTime: 3.9121
     SettlingMin: 0.9045
     SettlingMax: 1.0000
       Overshoot: 0
      Undershoot: 0
            Peak: 1.0000
        PeakTime: 10.5458


p1 =

    -1


z1 =

  0×1 empty double column vector
```

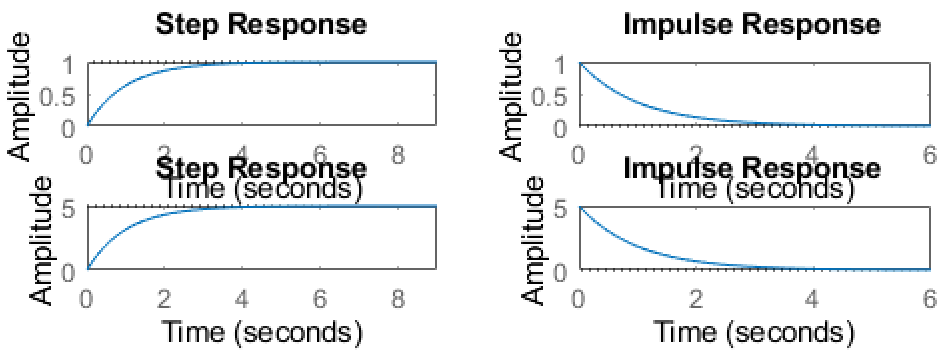


With Gain

```
T=1;
k=5;
sys_G = k*tf([1],[T,1])
subplot(5,2,3)
step(sys_G)
subplot(5,2,4)
impulse(sys_G)
S = stepinfo(sys_G)
p_g=pole(sys_G)
z_g=zero(sys_G)
```

```
sys_G =

    5
  -----
  s + 1

Continuous-time transfer function.


S =

  struct with fields:
```

```
        RiseTime: 2.1970
    SettlingTime: 3.9121
     SettlingMin: 4.5225
     SettlingMax: 4.9999
       Overshoot: 0
      Undershoot: 0
            Peak: 4.9999
        PeakTime: 10.5458


p_g =

    -1


z_g =

  0×1 empty double column vector
```

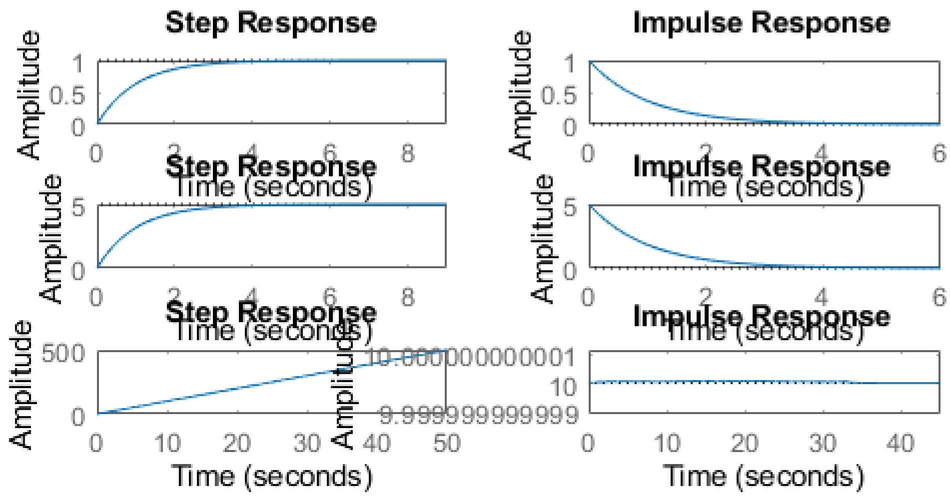


With PI

```
T=1;
k=5;
Kp=10;
```

```
I=tf([10],[1,0]);  %Ki
PI=Kp+I;
sys_PI = PI*tf([1],[T,1])
subplot(5,2,5)
step(sys_PI)
subplot(5,2,6)
impulse(sys_PI)
S = stepinfo(sys_PI)
p_pi=pole(sys_PI)
z_pi=zero(sys_PI)
```

```
sys_PI =

  10 s + 10
  ---------
   s^2 + s

Continuous-time transfer function.


S =

  struct with fields:

        RiseTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
      Undershoot: NaN
            Peak: Inf
        PeakTime: Inf


p_pi =

     0
    -1


z_pi =

    -1
```

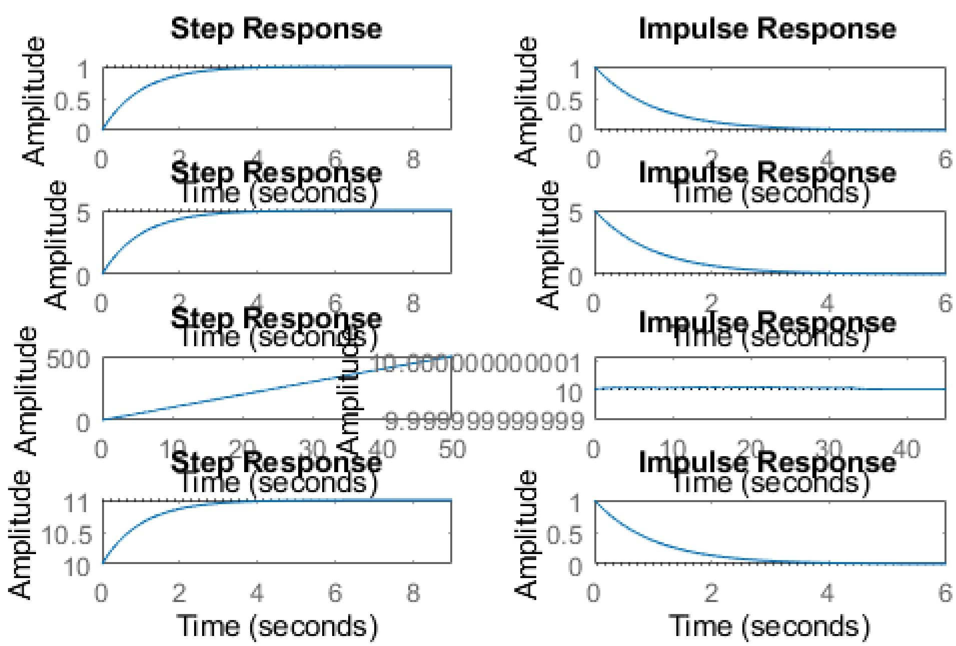


With PD

```
T=1;
k=5;
Kp=10;
D=tf([10,1],[0,1]);  %Kd
PD=Kp+D;
sys_PD = PD*tf([1],[T,1])
subplot(5,2,7)
step(sys_PD)
subplot(5,2,8)
impulse(sys_PD)
S = stepinfo(sys_PD)
p_pd=pole(sys_PD)
z_pd=zero(sys_PD)
```

```
sys_PD =

  10 s + 11
  ---------
    s + 1

Continuous-time transfer function.
```

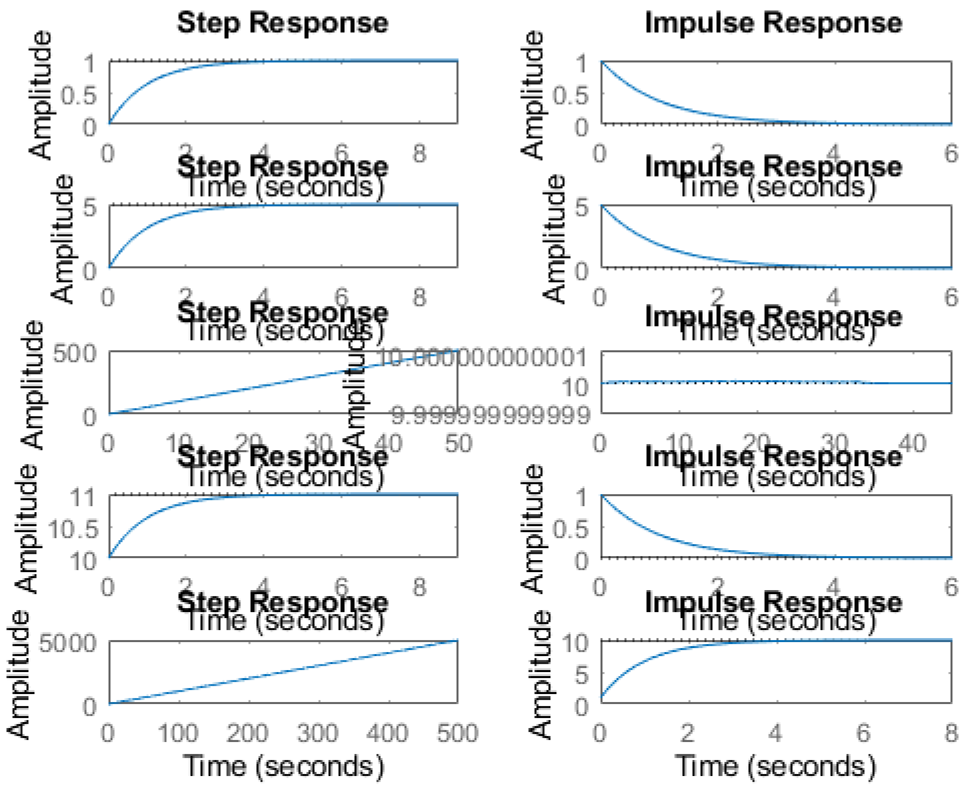
```
S =

  struct with fields:

        RiseTime: 2.1970
    SettlingTime: 3.9121
     SettlingMin: 10.9045
     SettlingMax: 11.0000
       Overshoot: 0
      Undershoot: 0
            Peak: 11.0000
        PeakTime: 10.5458


p_pd =

    -1


z_pd =

   -1.1000
```



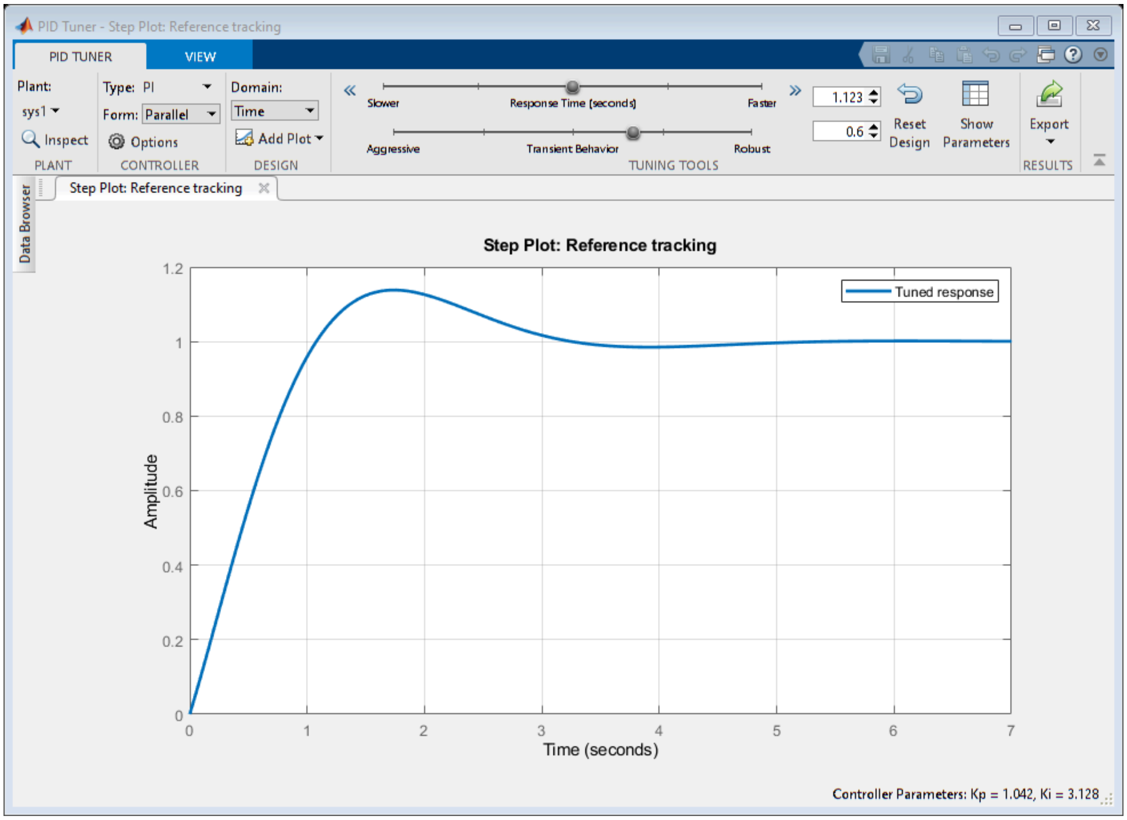
With PID

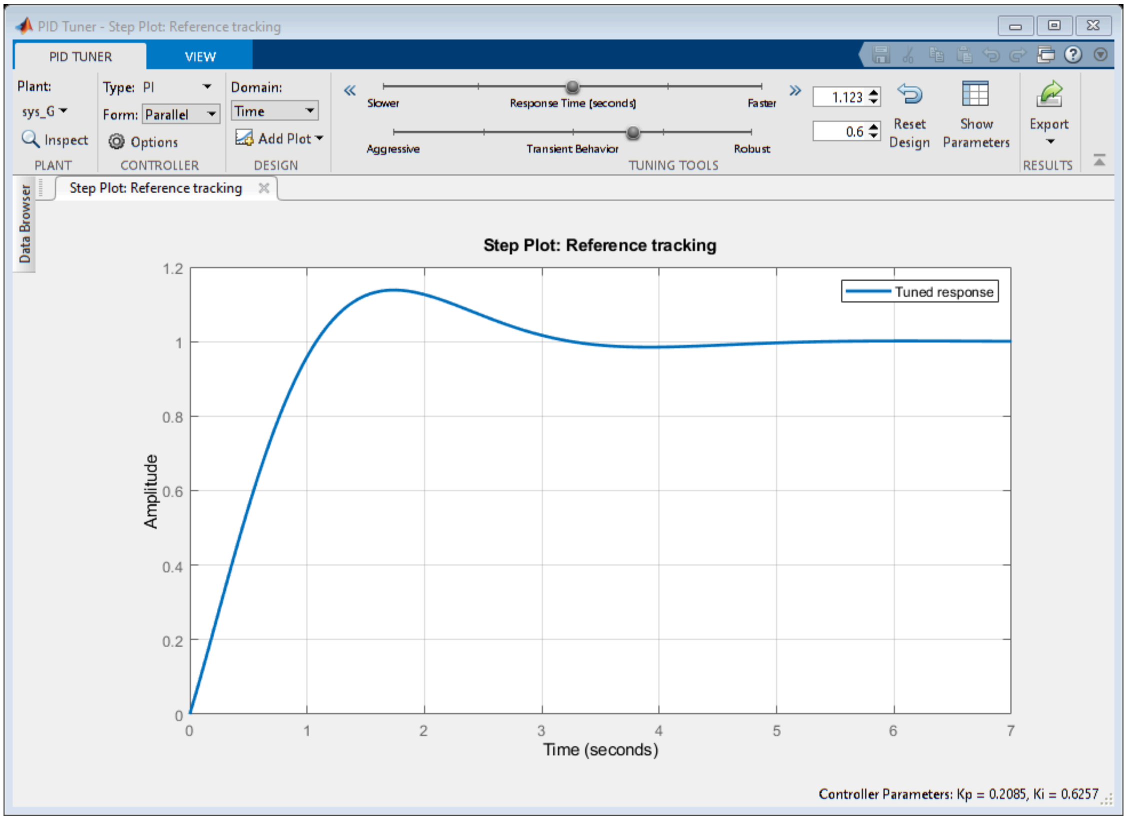
```
T=1;
k=5;
Kp=10;
D=tf([10,1],[0,1]);  %Kd
I=tf([10],[1,0]);  %Ki
PID=Kp+D+I;
sys_PID = PID*tf([1],[T,1])
subplot(5,2,9)
step(sys_PID)
subplot(5,2,10)
impulse(sys_PID)
S = stepinfo(sys_PID)
p_pid=pole(sys_PID)
z_pid=zero(sys_PID)
```

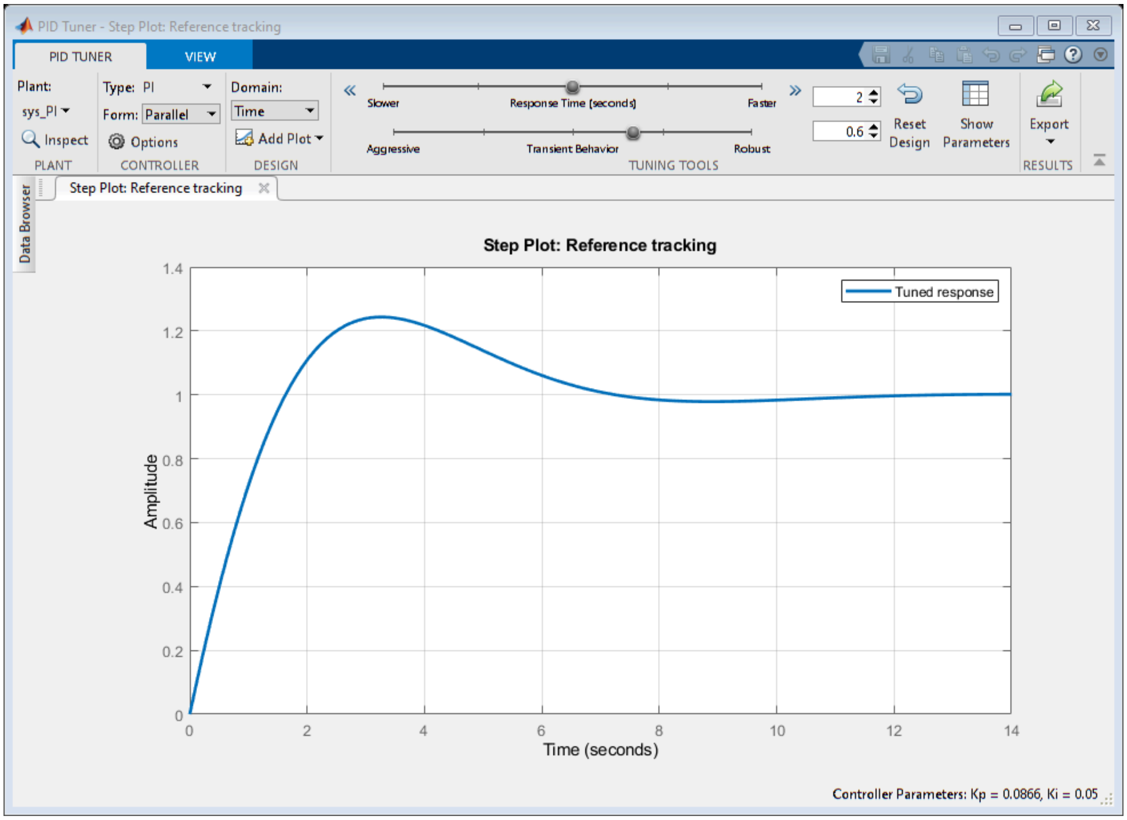
```
sys_PID =

  10 s^2 + 11 s + 10
  ------------------
       s^2 + s

Continuous-time transfer function.


S =

  struct with fields:

        RiseTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
      Undershoot: NaN
            Peak: Inf
        PeakTime: Inf


p_pid =

     0
    -1


z_pid =

  -0.5500 + 0.8352i
  -0.5500 - 0.8352i
```



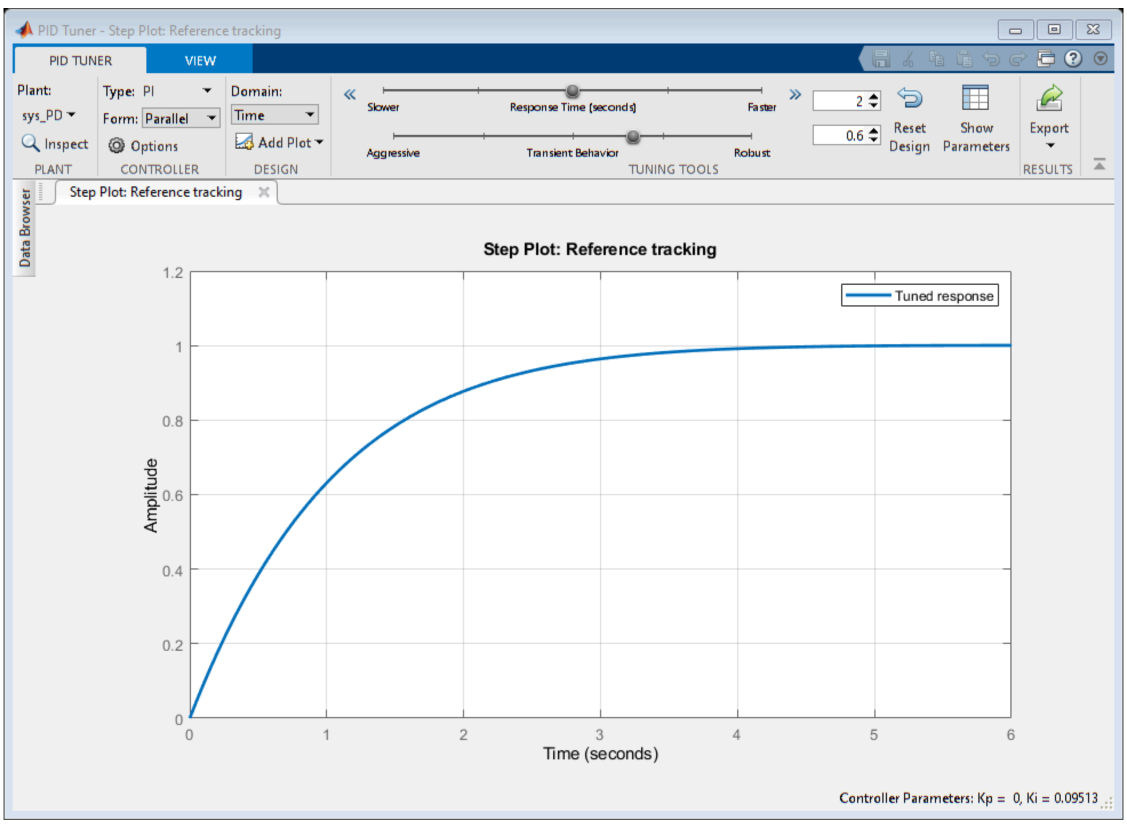

```
figure
pzmap(sys1)
pzmap(sys_G)
pzmap(sys_PI)
pzmap(sys_PD)
pzmap(sys_PID)


pidTuner(sys1)
pidTuner(sys_G)
pidTuner(sys_PI)
pidTuner(sys_PD)
pidTuner(sys_PID)
```


PID Tuner - Step Plot: Reference tracking
PID TUNER
VIEW
Plant:
sys1
Inspect
Type: PI
Form: Parallel
Options
Domain:
Time
Add Plot
Slower
Response Time (seconds)
Faster
Aggressive
Transient Behavior
Robust
1.123
0.6
Reset Design
Show Parameters
Export
PLANT
CONTROLLER
DESIGN
TUNING TOOLS
RESULTS
Data Browser
Step Plot: Reference tracking
Step Plot: Reference tracking
Tuned response
Amplitude
Time (seconds)
Controller Parameters: Kp = 1.042, Ki = 3.128

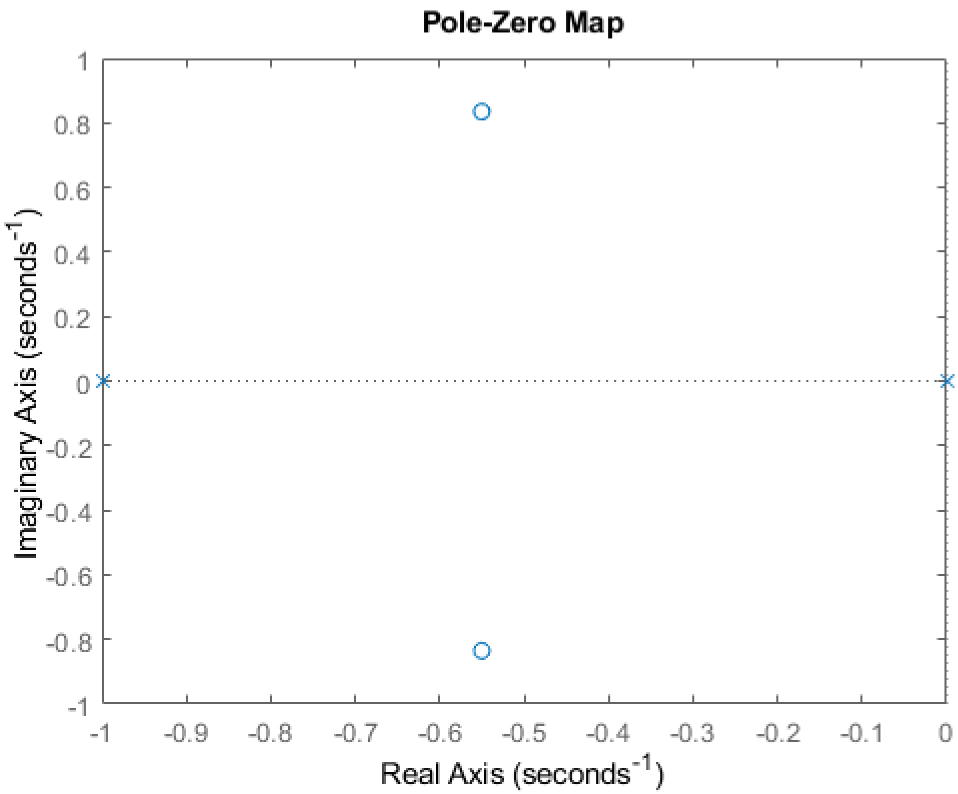

PID Tuner - Step Plot: Reference tracking
PID TUNER
VIEW
Plant:
sys_G
Inspect
Type: PI
Form: Parallel
Options
Domain:
Time
Add Plot
Slower
Response Time (seconds)
Faster
Aggressive
Transient Behavior
Robust
1.123
0.6
Reset Design
Show Parameters
Export
PLANT
CONTROLLER
DESIGN
TUNING TOOLS
RESULTS
Data Browser
Step Plot: Reference tracking
Step Plot: Reference tracking
Tuned response
Amplitude
Time (seconds)
Controller Parameters: Kp = 0.2085, Ki = 0.6257



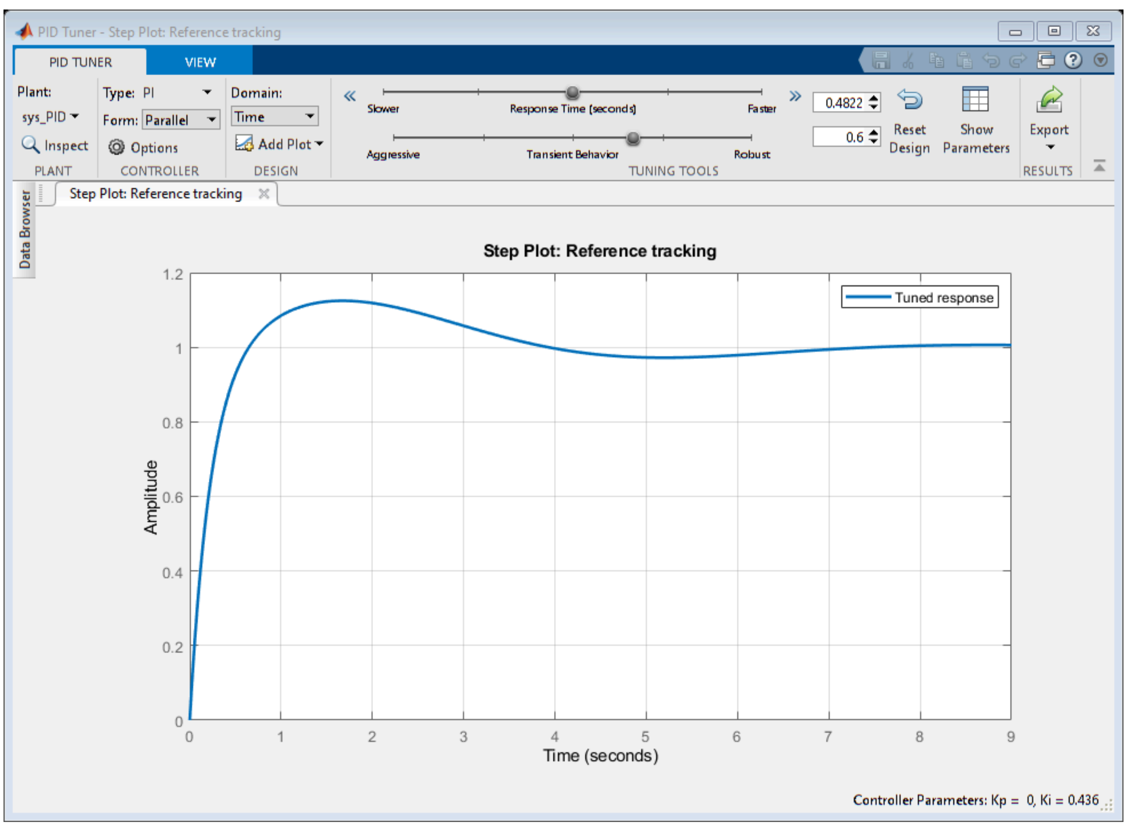
PID Tuner - Step Plot: Reference tracking
PID TUNER
VIEW
Plant:
sys_PI
Inspect
PLANT
Type: PI
Form: Parallel
Options
CONTROLLER
Domain:
Time
Add Plot
DESIGN
Slower
Response Time (seconds)
Faster
Aggressive
Transient Behavior
Robust
TUNING TOOLS
Reset Design
Show Parameters
Export
RESULTS
Data Browser
Step Plot: Reference tracking
Step Plot: Reference tracking
Tuned response
Amplitude
Time (seconds)
Controller Parameters: Kp = 0.0866, Ki = 0.05



PID Tuner - Step Plot: Reference tracking
PID TUNER
VIEW
Plant:
sys_PD
Inspect
PLANT
Type: PI
Form: Parallel
Options
CONTROLLER
Domain:
Time
Add Plot
DESIGN
Slower
Response Time (seconds)
Faster
Aggressive
Transient Behavior
Robust
TUNING TOOLS
Reset Design
Show Parameters
Export
RESULTS
Data Browser
Step Plot: Reference tracking
Step Plot: Reference tracking
Tuned response
Amplitude
Time (seconds)
Controller Parameters: Kp = 0, Ki = 0.09513



Pole-Zero Map
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})



PID Tuner - Step Plot: Reference tracking
PID TUNER
VIEW
Plant:
sys_PID
Inspect
Type: PI
Form: Parallel
Options
Domain:
Time
Add Plot
PLANT
CONTROLLER
DESIGN
Slower
Response Time (seconds)
Faster
Aggressive
Transient Behavior
Robust
TUNING TOOLS
0.4822
0.6
Reset Design
Show Parameters
Export
RESULTS
Data Browser
Step Plot: Reference tracking
Step Plot: Reference tracking
Tuned response
Amplitude
Time (seconds)
Controller Parameters: Kp = 0, Ki = 0.436

Analysis

```
%1.For the Basic the root lies on the left side of the imaginary axis
 that
% means the system is stable.
%Rise time is : 2.1970
%settling time is:3.9121 & Overshoot=0 for the basic
%2. For the system with gain also the root lies on the left side of
 the
%imaginary axis that means the system is stable.
%Rise time is:2.1970, settling time:3.9121, overshoot=0 for the gain.
 poles
%is also same only there is a change of amplitude.
%3. For the system with PI we got 2 poles one pole is at p1=0, p2=-1
 and
%one zero is at z=-1 so we can say that 1 pole will nullify the effect
 of
%zero and we will be remained with 1 pole left on the left side so we
 can
%say that system is stable.
%4. For the system with PD we got 1 pole at -1 and 1 zero at -1.10000
 on
%the left side of imaginary axis the settling time is 2.1970, R_t is
 3.9121
%5. For the system with PID controller we got 2 poles and 2 zeroes
 p1=0,
%p1=-1 and z1=-0.5500+0.8352i,z2=-0.5500-0.8352i the poles and zeores
 le on
%the left side of the imaginary axis again the system is stable again
 here
%also.
```

With POsitive feedback

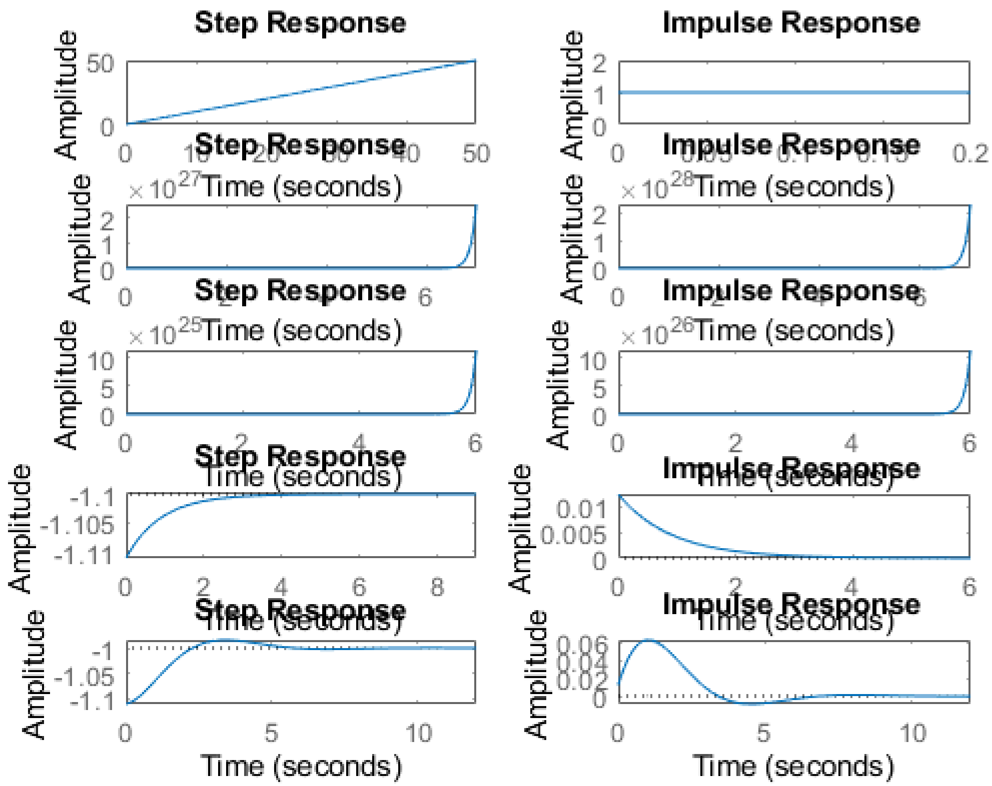
```
figure
T=1
sys = tf([1],[T,1])
sys_P=feedback(sys,-1)
subplot(5,2,1)
step(sys_P)
subplot(5,2,2)
impulse(sys_P)
S = stepinfo(sys_P)
p1=pole(sys_P)
z1=zero(sys_P)

T=1;
CF=10;
sys = CF*tf([1],[T,1]);
sys_G_P=feedback(sys,-1);
subplot(5,2,3)
step(sys_G_P)
```

```
subplot(5,2,4)
impulse(sys_G_P)
S = stepinfo(sys_G_P)
p_g=pole(sys_G_P)
z_g=zero(sys_G_P)

T=1;
Kp=10;
I=tf([10],[1,0]);  %Ki
PI=Kp+I;
sys = PI*tf([1],[T,1]);
sys_PI_P=feedback(sys,-1);
subplot(5,2,5)
step(sys_PI_P)
subplot(5,2,6)
impulse(sys_PI_P)
S = stepinfo(sys_PI_P)
p_pi=pole(sys_PI_P)
z_pi=zero(sys_PI_P)

T=1;
Kp=10;
D=tf([10,1],[0,1]);  %Kd
PD=Kp+D;
sys = PD*tf([1],[T,1]);
sys_PD_P=feedback(sys,-1);
subplot(5,2,7)
step(sys_PD_P)
subplot(5,2,8)
impulse(sys_PD_P)
S = stepinfo(sys_PD_P)
p_pd=pole(sys_PD_P)
z_pd=zero(sys_PD_P)

T=1
Kp=10;
D=tf([10,1],[0,1]);  %Kd
I=tf([10],[1,0]);  %Ki
PID=Kp+D+I;
sys = PID*tf([1],[T,1]);
sys_PID_P=feedback(sys,-1);
subplot(5,2,9)
step(sys_PID_P)
subplot(5,2,10)
impulse(sys_PID_P)
S = stepinfo(sys_PID_P)
p_pid=pole(sys_PID_P)
z_pid=zero(sys_PID_P)


T =

     1
```

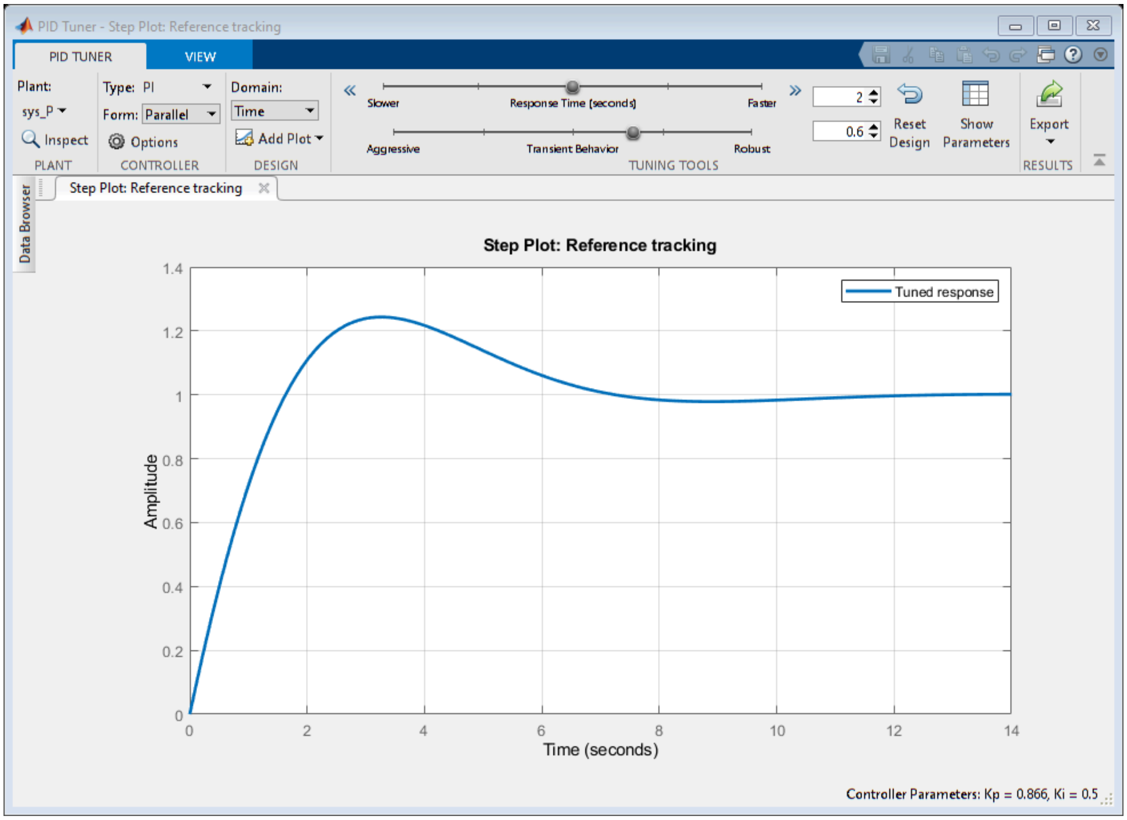
```
sys =

    1
  -----
  s + 1

Continuous-time transfer function.


sys_P =

  1
  -
  s

Continuous-time transfer function.


S =

  struct with fields:

        RiseTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
      Undershoot: NaN
            Peak: Inf
        PeakTime: Inf


p1 =

     0


z1 =

  0×1 empty double column vector


S =

  struct with fields:

        RiseTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
      Undershoot: NaN
            Peak: Inf
```

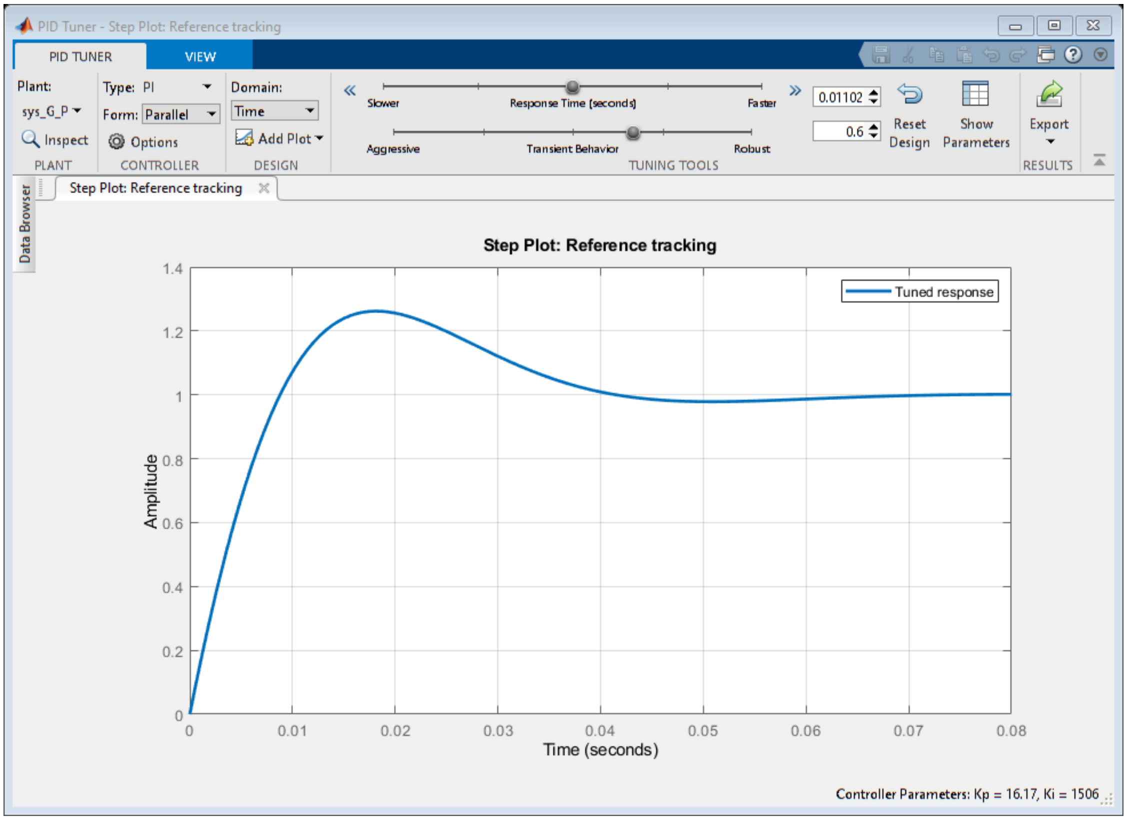
```
        PeakTime: Inf


p_g =

     9


z_g =

  0×1 empty double column vector


S =

  struct with fields:

        RiseTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
      Undershoot: NaN
            Peak: Inf
        PeakTime: Inf


p_pi =

    10
    -1


z_pi =

    -1


S =

  struct with fields:

        RiseTime: 1.9773
    SettlingTime: 3.5209
     SettlingMin: -1.1011
     SettlingMax: -1.1000
       Overshoot: 1.0101
      Undershoot: 0
            Peak: 1.1111
        PeakTime: 0


p_pd =
```

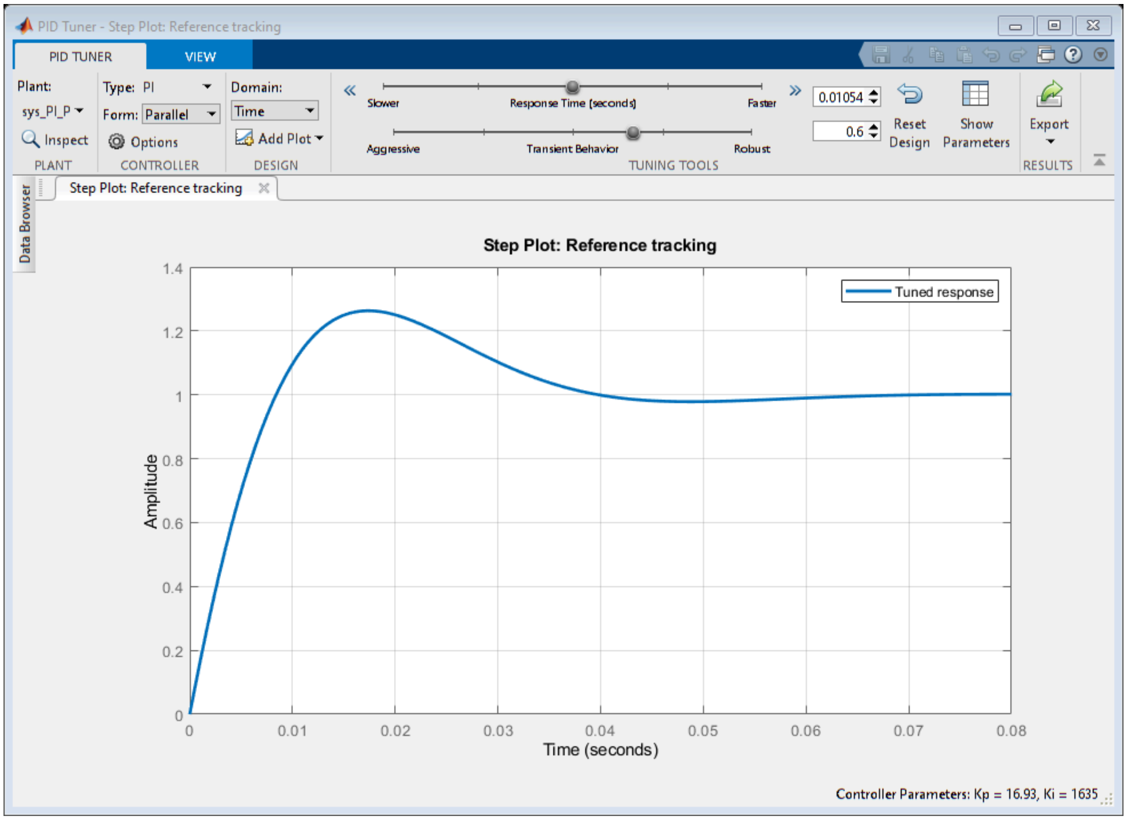
```
   -1.1111


z_pd =

   -1.1000


T =

     1


S =

  struct with fields:

        RiseTime: 1.5943
    SettlingTime: 7.1081
     SettlingMin: -1.0101
     SettlingMax: -0.9841
       Overshoot: 11.1111
      Undershoot: 0
            Peak: 1.1111
        PeakTime: 0


p_pid =

  -0.5556 + 0.8958i
  -0.5556 - 0.8958i


z_pid =

  -0.5500 + 0.8352i
  -0.5500 - 0.8352i
```

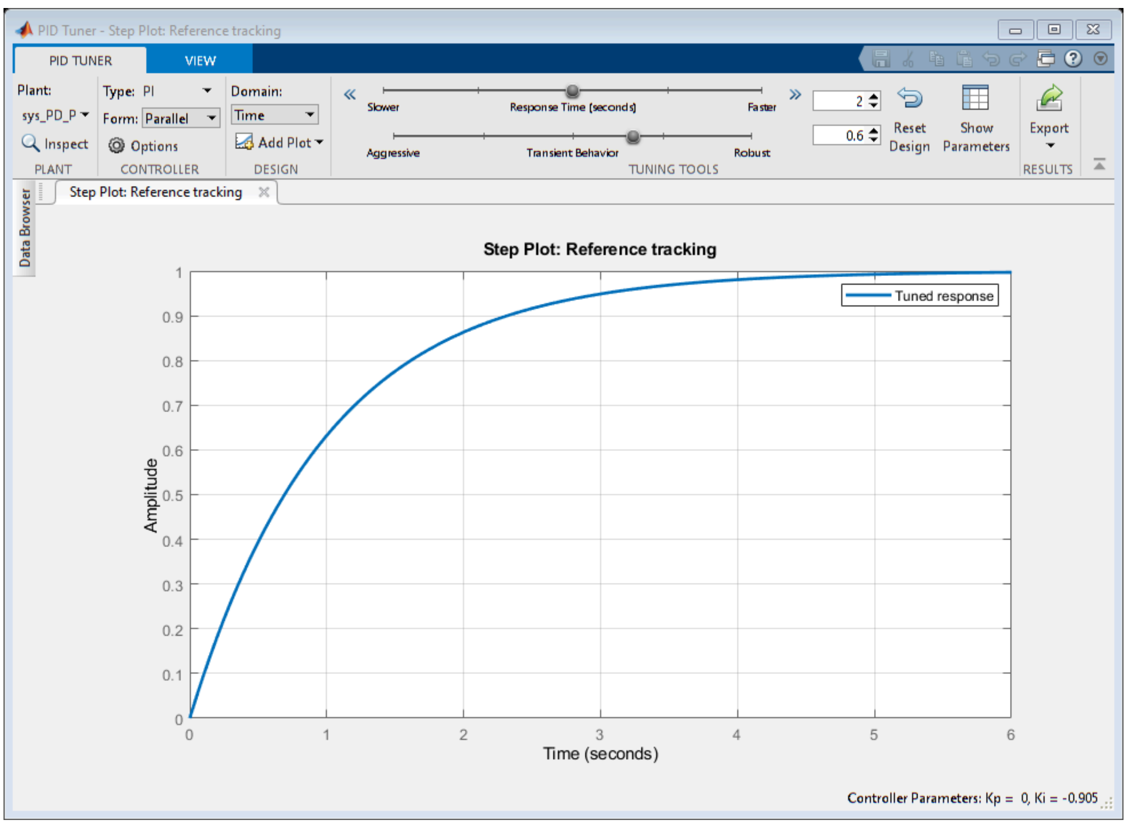


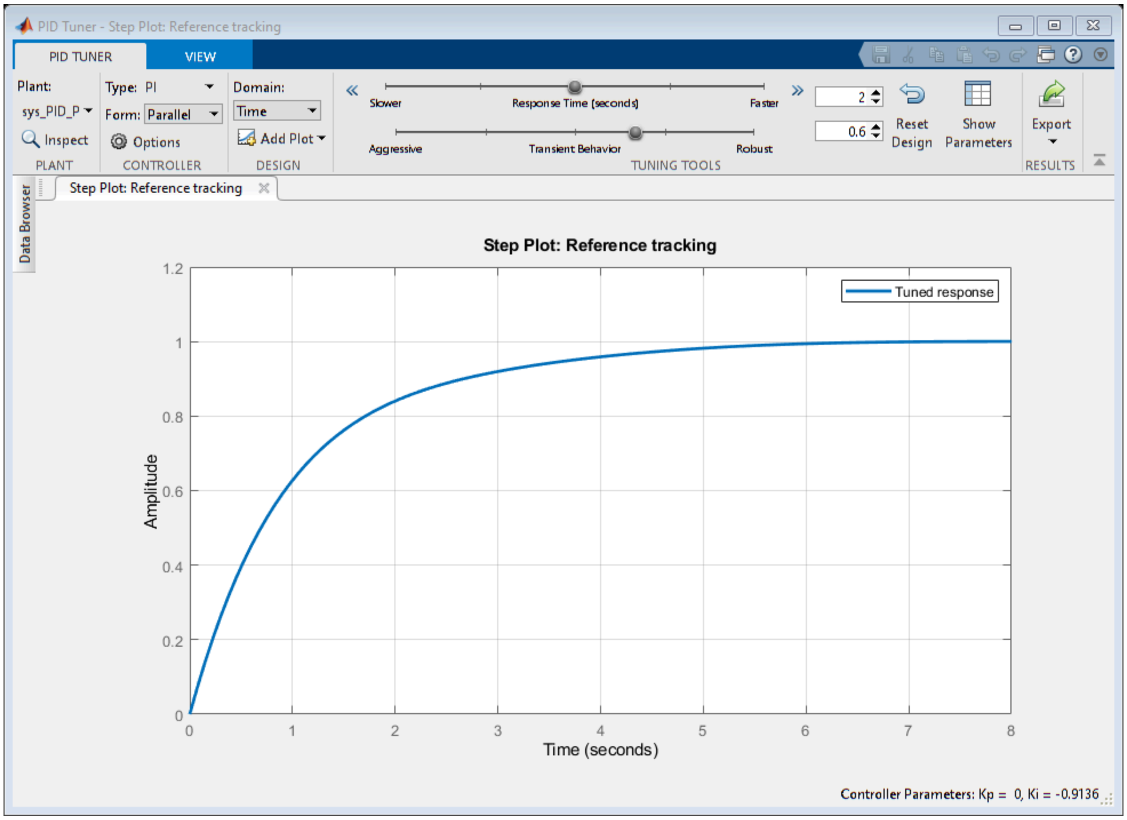

```
figure
hold on
pzmap(sys_P)
pzmap(sys_G_P)
pzmap(sys_PI_P)
pzmap(sys_PD_P)
pzmap(sys_PID_P)


pidTuner(sys_P)
pidTuner(sys_G_P)
pidTuner(sys_PI_P)
pidTuner(sys_PD_P)
pidTuner(sys_PID_P)
```


PID Tuner - Step Plot: Reference tracking
PID TUNER
VIEW
Plant:
sys_P
Inspect
Type: PI
Form: Parallel
Options
Domain:
Time
Add Plot
Slower
Response Time (seconds)
Faster
Aggressive
Transient Behavior
Robust
2
0.6
Reset Design
Show Parameters
Export
PLANT
CONTROLLER
DESIGN
TUNING TOOLS
RESULTS
Data Browser
Step Plot: Reference tracking
Step Plot: Reference tracking
Tuned response
Amplitude
Time (seconds)
Controller Parameters: Kp = 0.866, Ki = 0.5

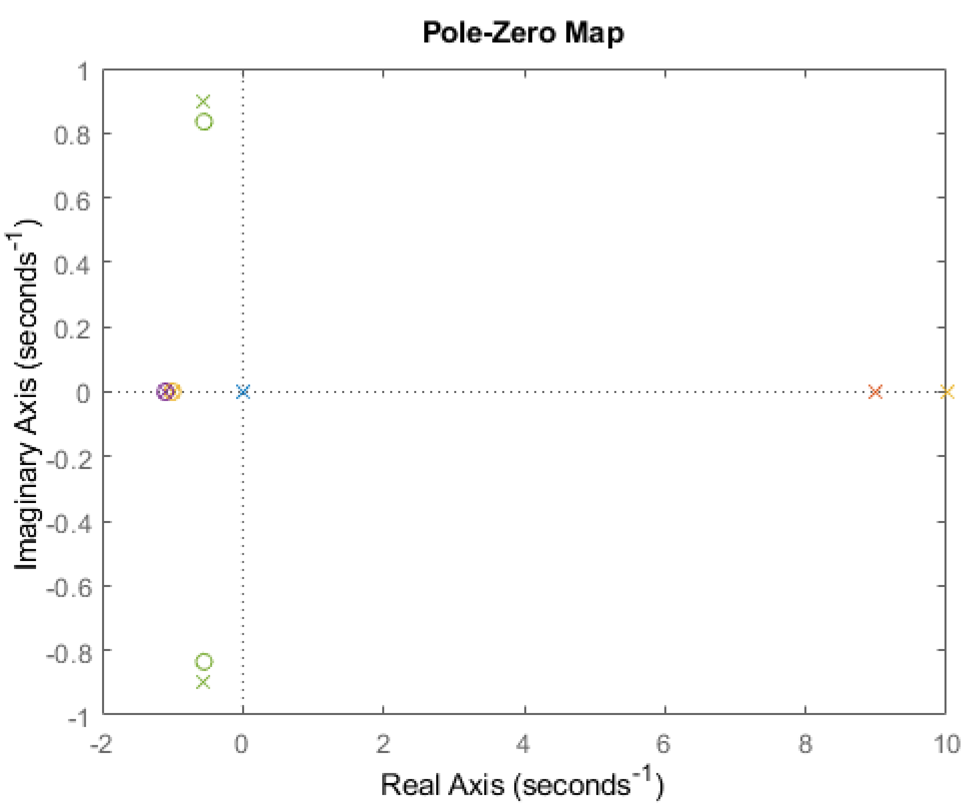

PID Tuner - Step Plot: Reference tracking
PID TUNER
VIEW
Plant:
sys_G_P
Inspect
Type: PI
Form: Parallel
Options
Domain:
Time
Add Plot
Slower
Response Time (seconds)
Faster
Aggressive
Transient Behavior
Robust
0.01102
0.6
Reset Design
Show Parameters
Export
PLANT
CONTROLLER
DESIGN
TUNING TOOLS
RESULTS
Data Browser
Step Plot: Reference tracking
Step Plot: Reference tracking
Tuned response
Amplitude
Time (seconds)
Controller Parameters: Kp = 16.17, Ki = 1506


PID Tuner - Step Plot: Reference tracking
PID TUNER
VIEW
Plant:
sys_PI_P
Inspect
PLANT
Type: PI
Form: Parallel
Options
CONTROLLER
Domain:
Time
Add Plot
DESIGN
Slower
Response Time (seconds)
Faster
Aggressive
Transient Behavior
Robust
TUNING TOOLS
0.01054
0.6
Reset Design
Show Parameters
Export
RESULTS
Data Browser
Step Plot: Reference tracking
Step Plot: Reference tracking
Tuned response
Amplitude
Time (seconds)
Controller Parameters: Kp = 16.93, Ki = 1635


PID Tuner - Step Plot: Reference tracking
PID TUNER
VIEW
Plant:
sys_PD_P
Inspect
PLANT
Type: PI
Form: Parallel
Options
CONTROLLER
Domain:
Time
Add Plot
DESIGN
Slower
Response Time (seconds)
Faster
Aggressive
Transient Behavior
Robust
TUNING TOOLS
2
0.6
Reset Design
Show Parameters
Export
RESULTS
Data Browser
Step Plot: Reference tracking
Step Plot: Reference tracking
Tuned response
Amplitude
Time (seconds)
Controller Parameters: Kp = 0, Ki = -0.905



PID Tuner - Step Plot: Reference tracking
PID TUNER
VIEW
Plant:
sys_PID_P
Inspect
PLANT
Type: PI
Form: Parallel
Options
CONTROLLER
Domain:
Time
Add Plot
DESIGN
Slower
Response Time (seconds)
Faster
Aggressive
Transient Behavior
Robust
TUNING TOOLS
2
0.6
Reset Design
Show Parameters
Export
RESULTS
Data Browser
Step Plot: Reference tracking
Step Plot: Reference tracking
Tuned response
Amplitude
Time (seconds)
Controller Parameters: Kp = 0, Ki = -0.9136



Pole-Zero Map
Imaginary Axis (seconds-1)
Real Axis (seconds-1)

Analysis

1.With the positive feed back system by giving the gain as 10 we got a

```
%pole at p=9 that says that system is unstable.
% 2.with the Positive feed back system by givng the PI controller we
 got 2
% poles 1 at p1=10,p2=-1 and 1 zero at z1=-1 so the pole and one zero
% nullify each other and left a pole on the left side of imaginary
 axis
% making the system stable.
% 3.With the Pd controller we can see that 1 zero is getting added,
 and 1
% pole is getting fixated at -1.1111 and a zero at -1.10000 as pole is
% located at the left side of the imaginary axis the system is stable
 with
% a rise time 1.9773, and settling time of 3.5209 with a overshoot of
 1.010
% 4.With the PID controller we can see that w eare getting complex
% conjugate poles and pair. p1=-0.5556+0.8958i,p2=-0.5556-0.8958i and
% zeroes arwe z1=-0.5500+0.8352i, z2=-0.5500-0.8352i anfd the
 s_t=7.1081,
% R_t=1.5943
% 5.So By observing the above mentioned settling time and rise time of
 the
% different controllers we are getting a stable system with PID
 controller.
```

With Negative feedback

```
figure
T=1;
sys = tf([1],[T,1])
sys_N=feedback(sys,1)
subplot(5,2,1)
step(sys_N)
subplot(5,2,2)
impulse(sys_N)
S = stepinfo(sys_N)
p_n=pole(sys_N)
z_n=zero(sys_N)

T=1;
CF=10;
sys = CF*tf([1],[T,1])
sys_G_N=feedback(sys,1)
subplot(5,2,3)
step(sys_G_N)
subplot(5,2,4)
impulse(sys_G_N)
S = stepinfo(sys_G_N)
p_gn=pole(sys_G_N)
z_gn=zero(sys_G_N)
```

```
T=1;
Kp=10;
I=tf([10,0],[1,0]);  %Ki
PI=Kp+I;
sys = PI*tf([1],[T,1])
sys_PI_N=feedback(sys,1)
subplot(5,2,5)
step(sys_PI_N)
subplot(5,2,6)
impulse(sys_PI_N)
S = stepinfo(sys_PI_N)
p_npi=pole(sys_PI_N)
z_npi=zero(sys_PI_N)

T=1;
Kp=10;
D=tf([10,1],[0,1]);  %Kd
PD=Kp+D;
sys = PD*tf([1],[T,1])
sys_PD_N=feedback(sys,1)
subplot(5,2,7)
step(sys_PD_N)
subplot(5,2,8)
impulse(sys_PD_N)
S = stepinfo(sys_PD_N)
p_npd=pole(sys_PD_N)
z_npd=zero(sys_PD_N)

T=1;
Kp=10;
D=tf([10,1],[0,1])  %Kd
I=tf([10],[1,0])  %Ki
PID=Kp+D+I
sys = PID*tf([1],[T,1])
sys_PID_N=feedback(sys,1)
subplot(5,2,9)
step(sys_PID_N)
subplot(5,2,10)
impulse(sys_PID_N)
S = stepinfo(sys_PID_N)
p_npid=pole(sys_PID_N)
z_npid=zero(sys_PID_N)
```

```
sys =

    1
  -----
  s + 1

Continuous-time transfer function.
```

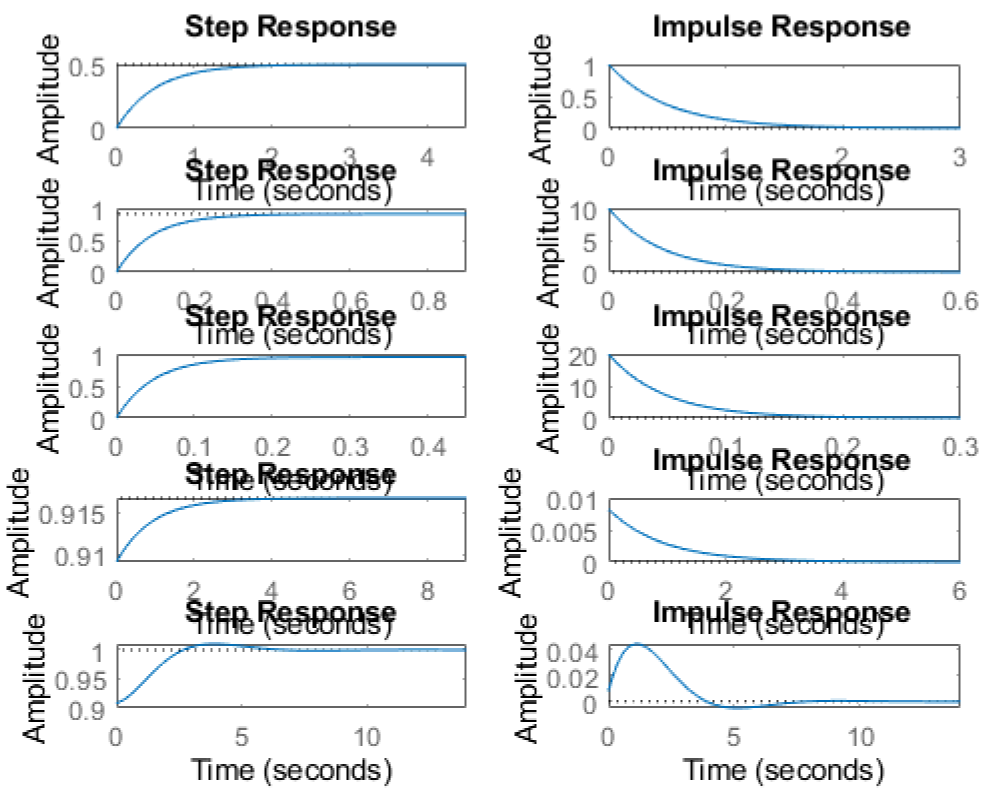
```
sys_N =

    1
  -----
  s + 2

Continuous-time transfer function.


S =

  struct with fields:

        RiseTime: 1.0985
    SettlingTime: 1.9560
     SettlingMin: 0.4523
     SettlingMax: 0.5000
       Overshoot: 0
      Undershoot: 0
            Peak: 0.5000
        PeakTime: 5.2729


p_n =

    -2


z_n =

  0×1 empty double column vector


sys =

   10
  -----
  s + 1

Continuous-time transfer function.


sys_G_N =

    10
  ------
  s + 11

Continuous-time transfer function.


S =

  struct with fields:
```

```
        RiseTime: 0.1997
    SettlingTime: 0.3556
     SettlingMin: 0.8223
     SettlingMax: 0.9091
       Overshoot: 0
      Undershoot: 0
            Peak: 0.9091
        PeakTime: 0.9587


p_gn =

   -11


z_gn =

  0×1 empty double column vector


sys =

   20 s
  -------
  s^2 + s

Continuous-time transfer function.


sys_PI_N =

     20 s
  ----------
  s^2 + 21 s

Continuous-time transfer function.


S =

  struct with fields:

        RiseTime: 0.1046
    SettlingTime: 0.1863
     SettlingMin: 0.8614
     SettlingMax: 0.9524
       Overshoot: 0
      Undershoot: 0
            Peak: 0.9524
        PeakTime: 0.5022


p_npi =
```

```
     0
   -21


z_npi =

     0


sys =

  10 s + 11
  ---------
    s + 1

Continuous-time transfer function.


sys_PD_N =

  10 s + 11
  ---------
  11 s + 12

Continuous-time transfer function.


S =

  struct with fields:

        RiseTime: 2.0139
    SettlingTime: 3.5861
     SettlingMin: 0.9159
     SettlingMax: 0.9167
       Overshoot: 0
      Undershoot: 0
            Peak: 0.9167
        PeakTime: 9.6670


p_npd =

   -1.0909


z_npd =

   -1.1000


D =
```

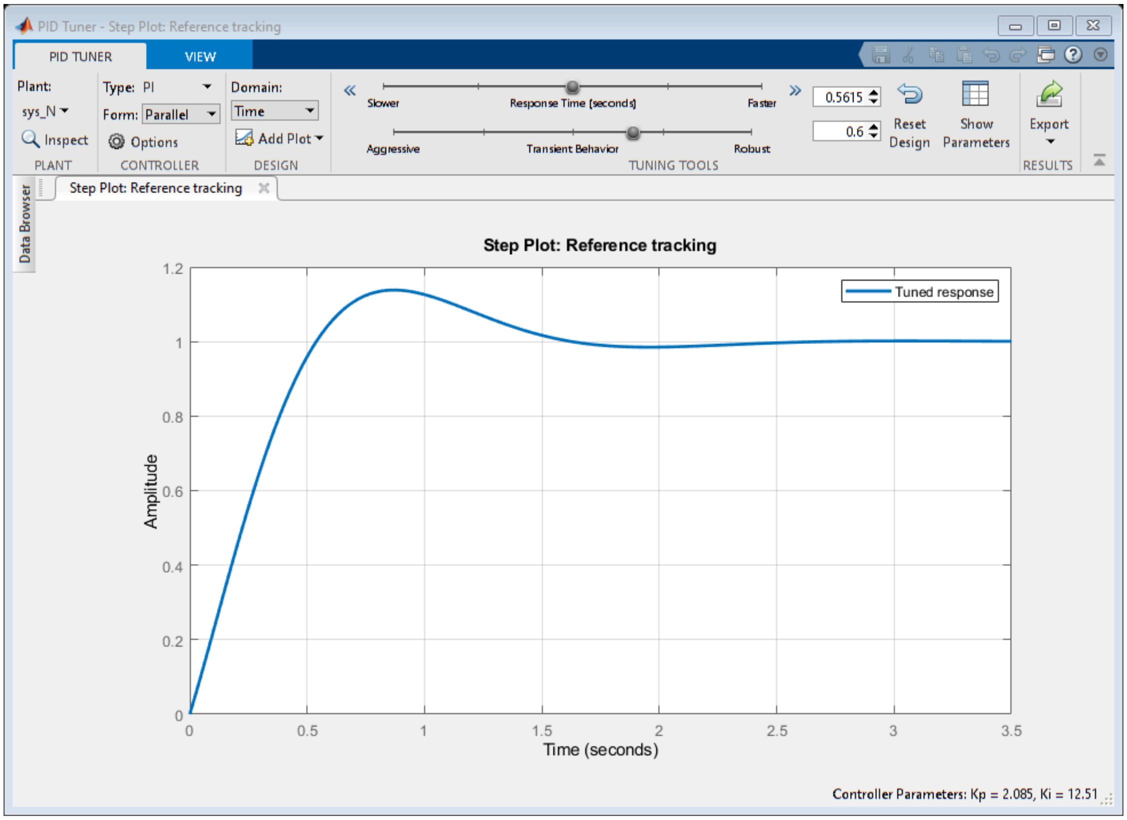
```
  10 s + 1

Continuous-time transfer function.


I =

  10
  --
  s

Continuous-time transfer function.


PID =

  10 s^2 + 11 s + 10
  ------------------
          s

Continuous-time transfer function.


sys =

  10 s^2 + 11 s + 10
  ------------------
       s^2 + s

Continuous-time transfer function.


sys_PID_N =

  10 s^2 + 11 s + 10
  ------------------
  11 s^2 + 12 s + 10

Continuous-time transfer function.


S =

  struct with fields:

        RiseTime: 1.8654
    SettlingTime: 6.0686
     SettlingMin: 0.9929
     SettlingMax: 1.0102
       Overshoot: 1.0208
      Undershoot: 0
            Peak: 1.0102
        PeakTime: 3.8837
```

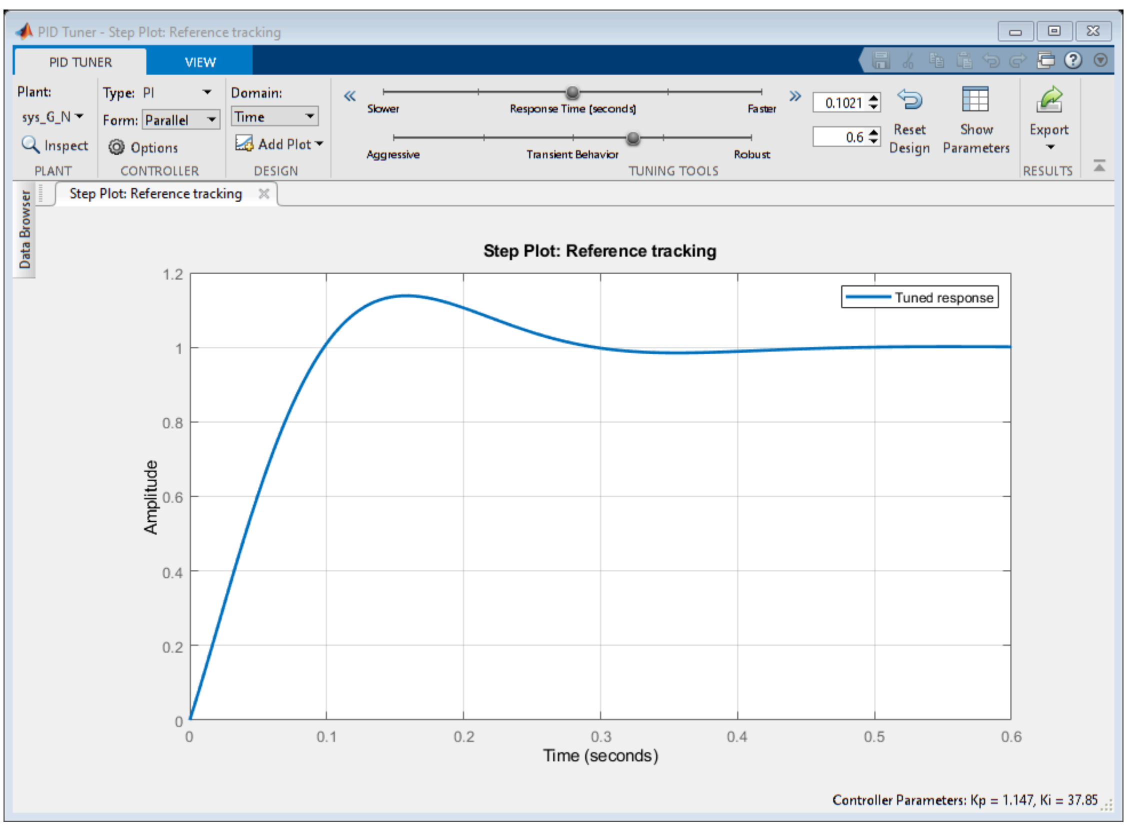
```
p_npid =

  -0.5455 + 0.7820i
  -0.5455 - 0.7820i


z_npid =

  -0.5500 + 0.8352i
  -0.5500 - 0.8352i
```

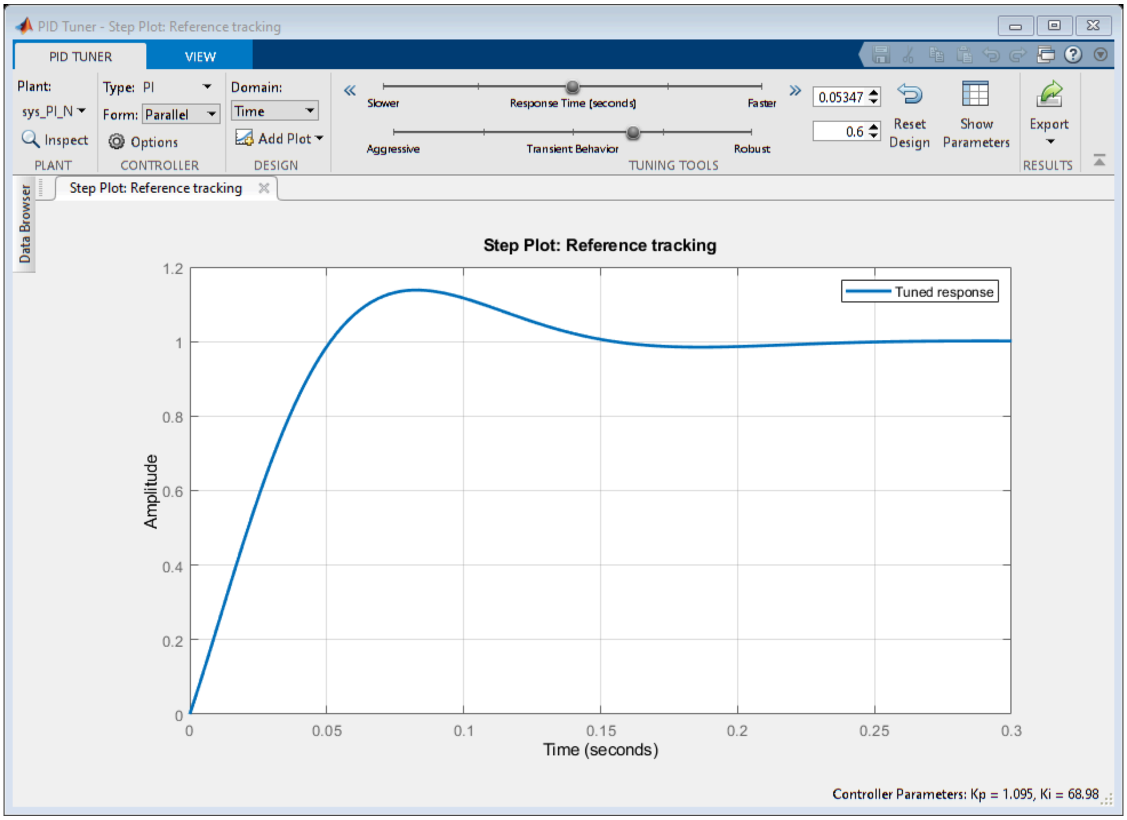


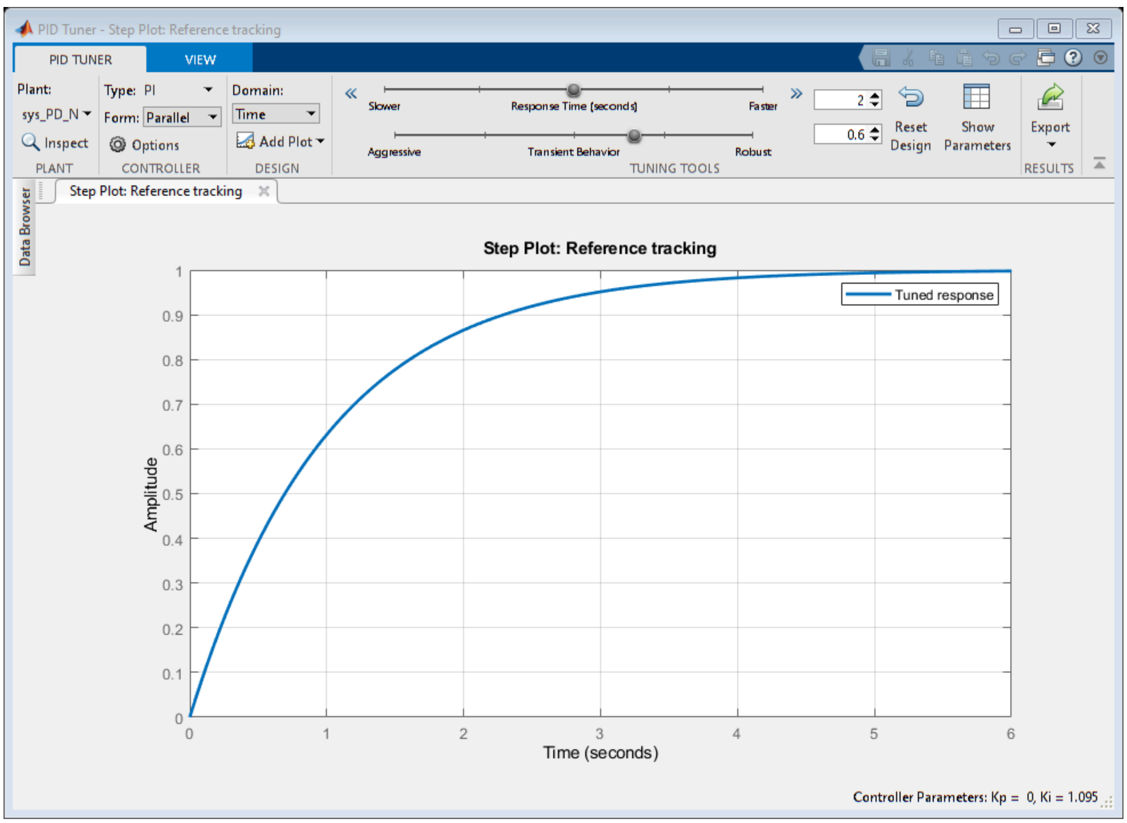

```
figure
hold on
pzmap(sys_N)
pzmap(sys_G_N)
pzmap(sys_PI_N)
pzmap(sys_PD_N)
pzmap(sys_PID_N)


pidTuner(sys_N)
pidTuner(sys_G_N)
pidTuner(sys_PI_N)
pidTuner(sys_PD_N)
```

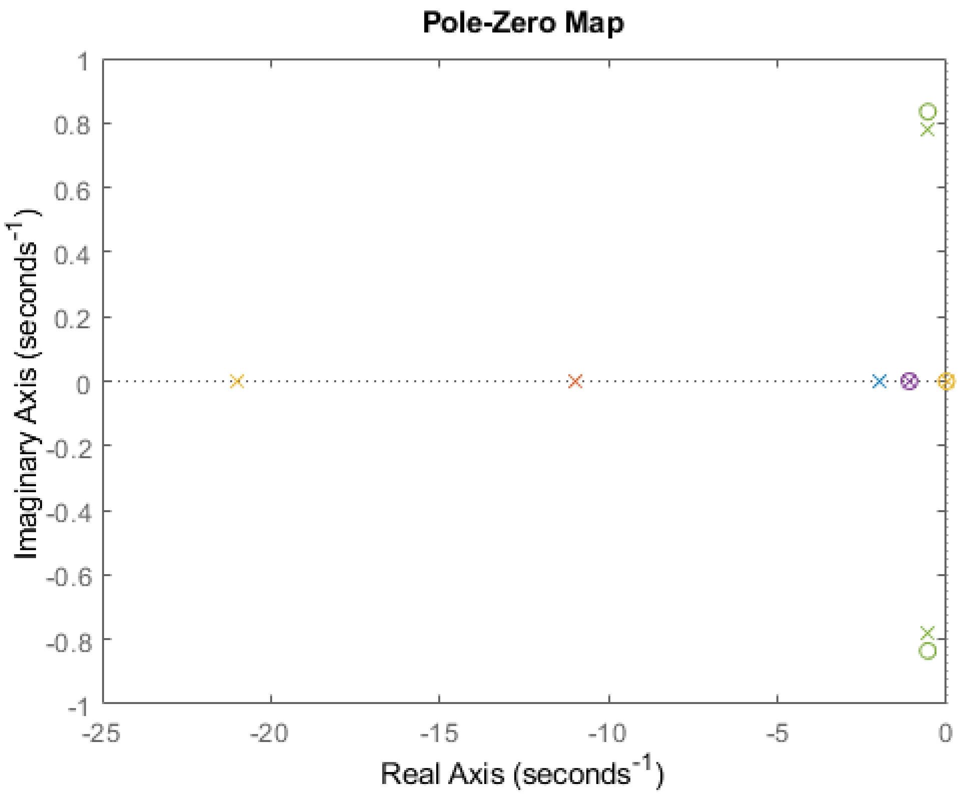
```
pidTuner(sys_PID_N)
```

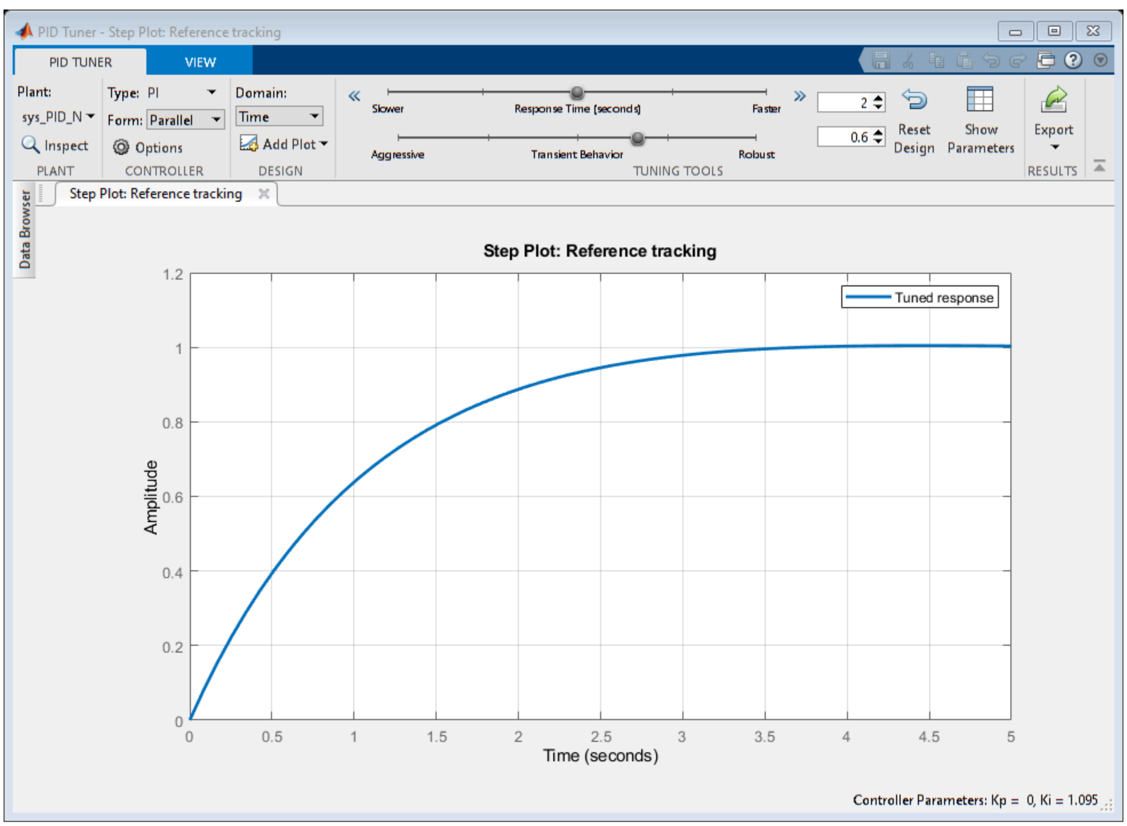





PID Tuner - Step Plot: Reference tracking
PID TUNER
VIEW
Plant:
sys_PI_N
Inspect
PLANT
Type: PI
Form: Parallel
Options
CONTROLLER
Domain:
Time
Add Plot
DESIGN
Slower
Response Time (seconds)
Faster
Aggressive
Transient Behavior
Robust
TUNING TOOLS
0.05347
0.6
Reset Design
Show Parameters
Export
RESULTS
Data Browser
Step Plot: Reference tracking
Step Plot: Reference tracking
Tuned response
Amplitude
Time (seconds)
Controller Parameters: Kp = 1.095, Ki = 68.98


PID Tuner - Step Plot: Reference tracking
PID TUNER
VIEW
Plant:
sys_PD_N
Inspect
PLANT
Type: PI
Form: Parallel
Options
CONTROLLER
Domain:
Time
Add Plot
DESIGN
Slower
Response Time (seconds)
Faster
Aggressive
Transient Behavior
Robust
TUNING TOOLS
2
0.6
Reset Design
Show Parameters
Export
RESULTS
Data Browser
Step Plot: Reference tracking
Step Plot: Reference tracking
Tuned response
Amplitude
Time (seconds)
Controller Parameters: Kp = 0, Ki = 1.095


Pole-Zero Map
Imaginary Axis (seconds^-1)
Real Axis (seconds^-1)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-25
-20
-15
-10
-5
0


PID Tuner - Step Plot: Reference tracking
PID TUNER
VIEW
Plant:
sys_PID_N
Inspect
PLANT
Type: PI
Form: Parallel
Options
CONTROLLER
Domain:
Time
Add Plot
DESIGN
Slower
Response Time (seconds)
Faster
Aggressive
Transient Behavior
Robust
TUNING TOOLS
2
0.6
Reset Design
Show Parameters
Export
RESULTS
Data Browser
Step Plot: Reference tracking
Step Plot: Reference tracking
Tuned response
Amplitude
Time (seconds)
Controller Parameters: Kp = 0, Ki = 1.095

Analysis

1.with negaitve feed back gain we get 1 pole at p1=-11 which has a rise time of 0.1997, settling time of 0.3556 the system is stable. 2.with negative feed back Pi controller we get 2 poles at p1=-10,p2=-1 and a zero at z=-1, because of integrator in PI controller we are getting an extra pole in it now Rise-time=0.2197,settling time=0.3912 as the poles are on the left side of imaginary axis we can say that system is stable. 3.with a negative feed back PID controller we are getting complex conjugate poles and zeroes which are z1=-0.5500+0.8352i,z2=-0.5500-0.8352i,p1=-0.5455+0.7820i, p2=-0.5455+0.7820i the settling time is 1.8654 and the rise time is6.0686 so we can say that PID controller can not make the system more stable than PI and PD controllers did.

Published with MATLAB® R2021a