MATLAB/Simulink for Control Systems

Creating models:

1. Transfer function 1^{st} order

$$G(s) = \frac{1}{10s + 1}$$

MATLAB code:(m file)

```
num=[1];
den=[10 1];
sys=tf(num,den)
```

Output shown in the command window:

```
Transfer function:
   1
--------
10 s + 1
```

System Transfer function second order

$$G(s) = \frac{1}{s^2 + 3s + 2}$$

MATLAB code:

```
num=[1];
den=[1 3 2];
sys=tf(num,den)
```

Output :

```
Transfer function:
      1
-------------
s^2 + 3 s + 2
```

2. zero pole gain model:

MATLAB code:

```
sys=zpk([-5 -3],-2,10)
```

Output:

Zero/pole/gain:

```
10 (s+5) (s+3)
--------------
    (s+2)
```

MATLAB code:

```
sys=zpk([-1 -2],[-5 -1 -3],10)
```

Output:

Zero/pole/gain:

```
 10 (s+1) (s+2)
-----------------
(s+5) (s+3) (s+1)
```

3. State space model

$$\dot{X} = AX + Bu$$

$$Y = CX + Du$$

MATLAB code:

```
A=[0 1;-2 -3]; %(2x2)
B=[0; 1];   %(2x1)
C=[1 0];          %(1x2)
D=[0]; %(1x1)
sys=ss(A,B,C,D)
```

output:

```
a =

       x1  x2

   x1   0   1

   x2  -2  -3
```

```
b =

       u1

   x1   0

   x2   1

 c =

       x1  x2

   y1   1   0

d =

       u1

   y1   0
```

Continuous-time model.

continuous to discrete time models:

```
A=[0 1;-2 -3];
B=[0; 1];
C=[1 0];
D=[0];
sys=ss(A,B,C,D)
sysD=c2d(sys,1/6400,'zoh')
```

output:

```
a =
               x1          x2
   x1           1   0.0001562
   x2  -0.0003124      0.9995

b =
               u1
   x1  1.221e-008
   x2   0.0001562

c =
       x1  x2
   y1   1   0

d =
       u1
```

```
  y1   0
 Sampling time: 0.00015625
```

Discrete-time model.

Discrete to continuous time models:

```
sysC=d2c(sysD,'zoh')

a =

               x1          x2

  x1  6.739e-013           1

  x2          -2          -3

 b =

      u1

  x1   0

  x2   1

 c =

      x1  x2

  y1   1   0

 d =

      u1

  y1   0
```

Continuous-time model.

Discrete to discrete - resampling

```
sysD=d2d(sysD,1,'zoh')

a =

             x1        x2

  x1    0.6004    0.2325
```

x2   -0.4651 -0.09721

b =

u1

x1 0.1998

x2 0.2325

c =

x1 x2

y1   1   0

d =

u1

y1   0

Sampling time: 1

Discrete-time model.

Time domain analysis: step response, step response information, impulse response

```
num=[1];
den=[1 3 2];
sys=tf(num,den)
step(sys)
impulse(sys)
stepinfo(sys)
```

step or impulse response will appear in a figure window

step response information

output:

RiseTime: 2.5914

SettlingTime: 4.6015

SettlingMin: 0.4548

SettlingMax: 0.5000

Overshoot: 0

Undershoot: 0

Peak: 0.5000

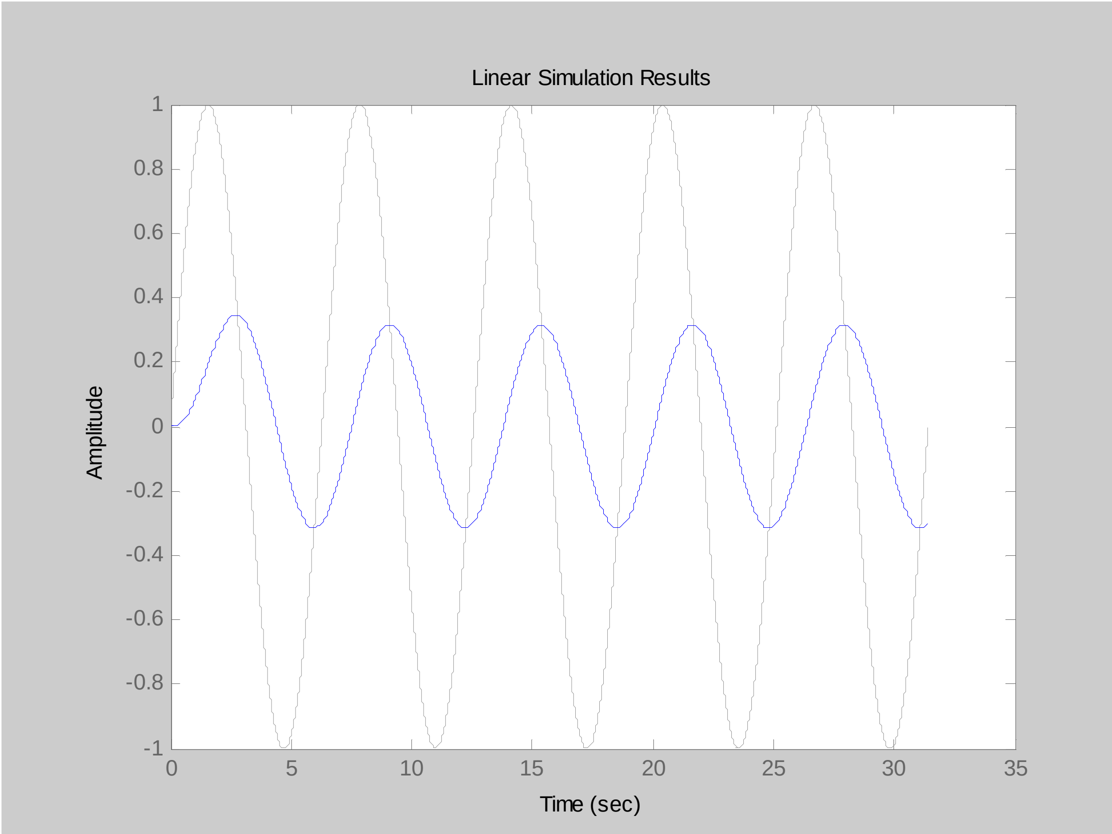
PeakTime: 13.6697

Free response with initial conditions (valid for state space models only)

```
A=[0 1;-2 -3];
B=[0; 1];
C=[1 0];
D=[0];
sys=ss(A,B,C,D)
initial(sys,[0.2 -0.5],10)
```

Linear Response to user defined input signal

```
num=[1];
den=[1 3 2];
sys=tf(num,den)
t=0:1/6400:(10*pi)
u=sin(t);
lsim(sys,u,t)
```

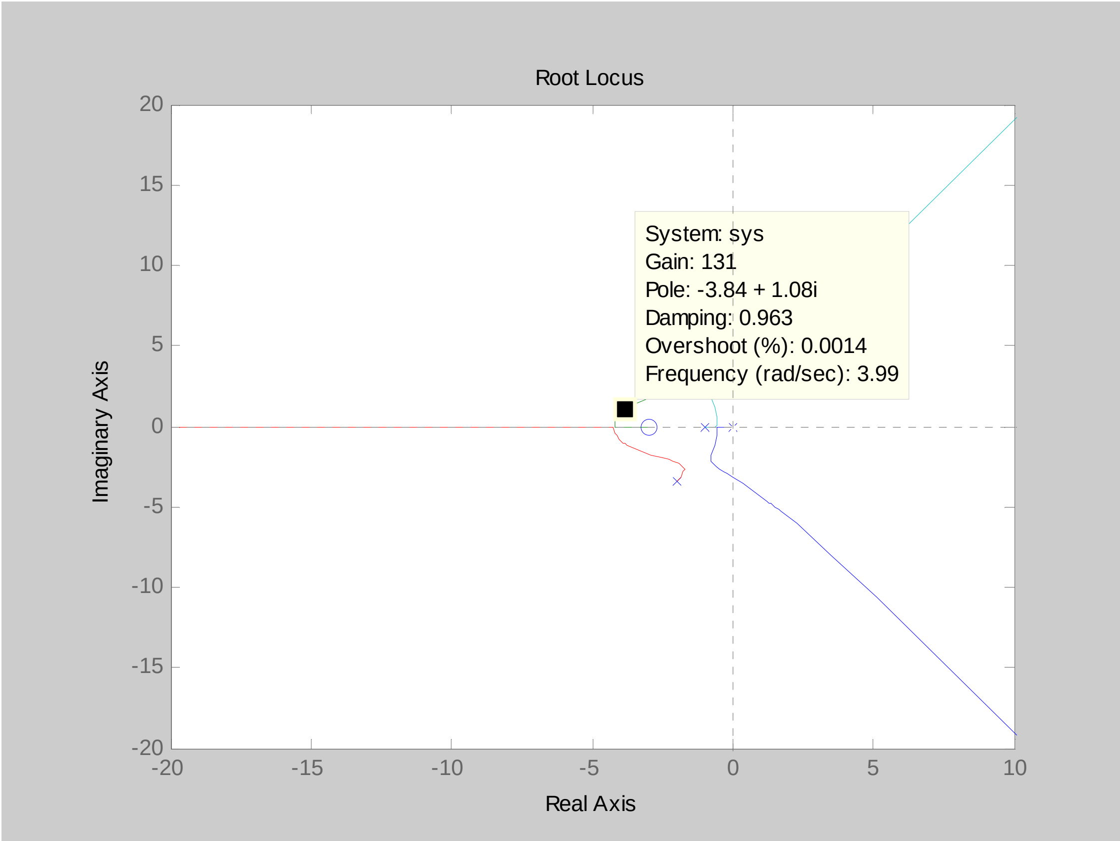


Root locus plot

```
num=[1 3];
den=[1 5 20 16 0];
sys=tf(num,den)
rlocus(sys)
```

output:
Transfer function:

```
        s + 3
---------------------------
s^4 + 5 s^3 + 20 s^2 + 16 s
```

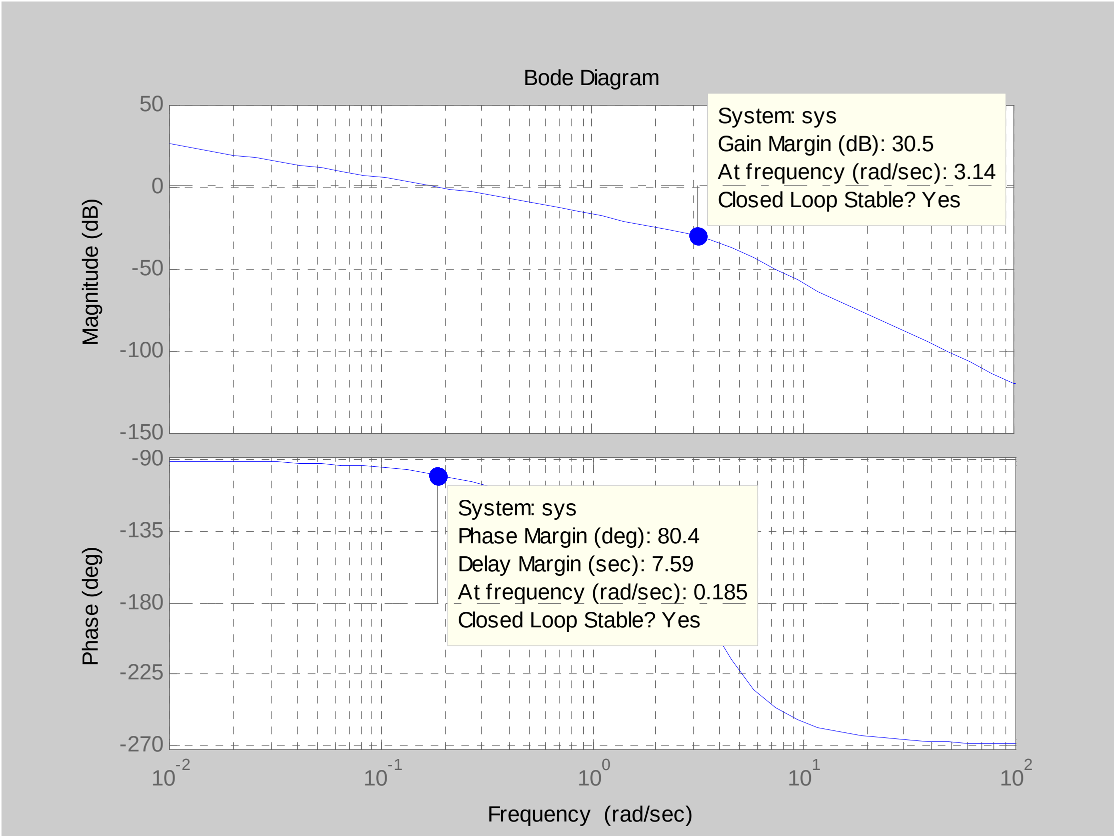


Frequency domain analysis

```
num=[1 3];
den=[1 5 20 16 0];
sys=tf(num,den)
bode(sys)
```

Transfer function:

```
        s + 3
--------------------------
s^4 + 5 s^3 + 20 s^2 + 16 s
```



To get bode magnitude diagram

```
num=[1 3];
den=[1 5 20 16 0];
sys=tf(num,den)
bodemag(sys)
```

gain and phase margin, gain cross over frequency, phase cross over frequency:

```
num=[1 3];
den=[1 5 20 16 0];
sys=tf(num,den)
bodemag(sys)
margin(sys)
[Gm,Pm,wgc,wpc]=margin(sys)
```

Output:

Transfer function:

```
       s + 3
```

----

s^4 + 5 s^3 + 20 s^2 + 16 s

Gm =

33.3225

Pm =

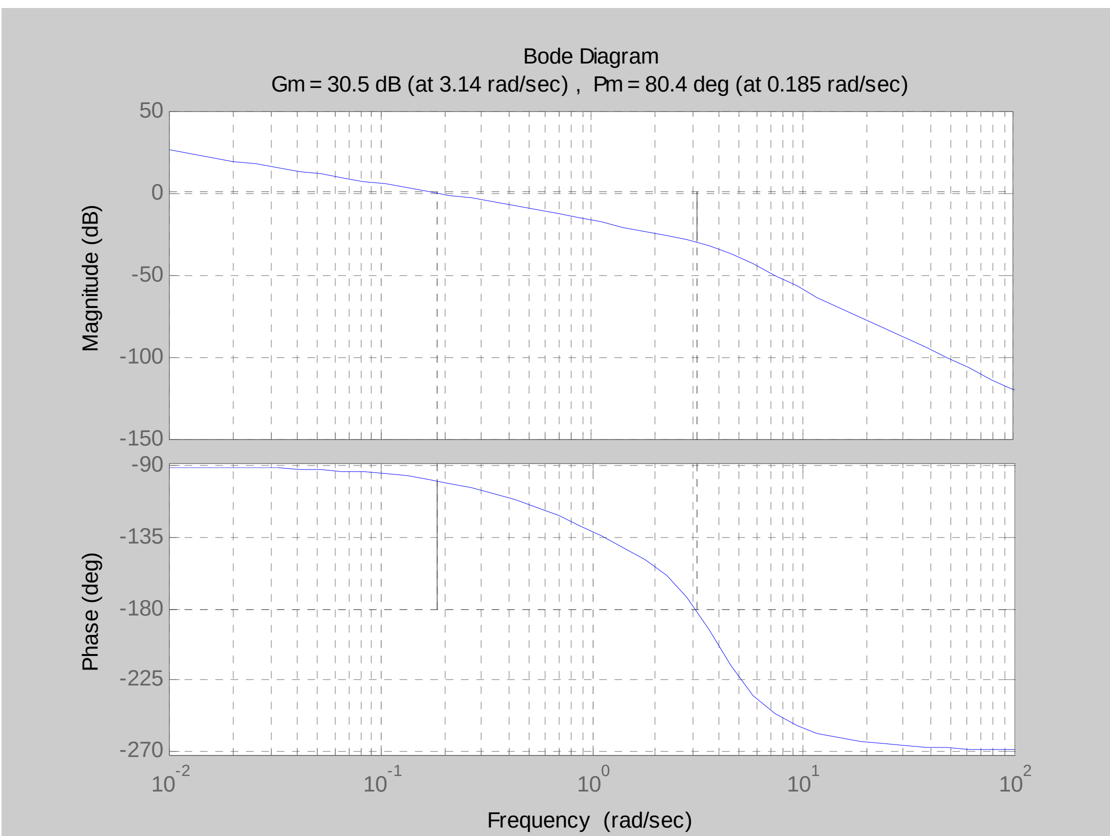
80.3979

wgc =

3.1406

wpc =

0.1849



```
num=[1 3];
den=[1 5 20 16 0];
sys=tf(num,den)
allmargin(sys)
```

output:

Transfer function:

```
          s + 3
---------------------------
s^4 + 5 s^3 + 20 s^2 + 16 s
```

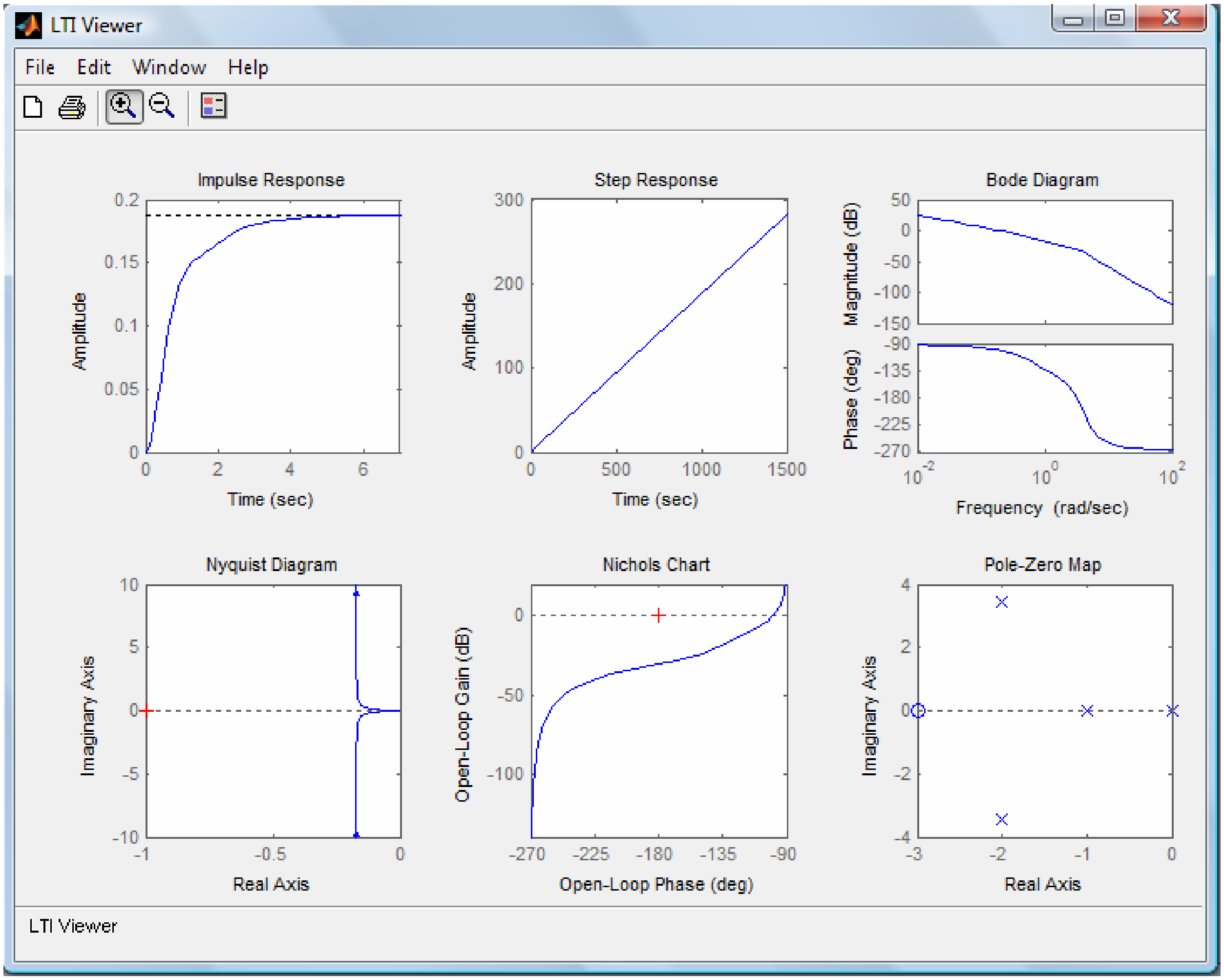
```
    GainMargin: 33.3225

    GMFrequency: 3.1406

    PhaseMargin: 80.3979

    PMFrequency: 0.1849

    DelayMargin: 7.5881

    DMFrequency: 0.1849

         Stable: 1
```

```
num=[1 3];
den=[1 5 20 16 0];
sys=tf(num,den)
ltiview
```

output:

import system

choose plot configurations



System interconnections:

num=[1];
den=[1 4];
sys1=tf(num,den)
sys2=tf([1],[1 2 5])

output:

Transfer function:

  1

-----

s + 4

Transfer function:

1

----

s^2 + 2 s + 5

```
num=[1];
den=[1 4];
sys1=tf(num,den)
sys2=tf([1],[1 2 5])
sys=parallel(sys1,sys2)
```

output:

Transfer function:

s^2 + 3 s + 9

----

s^3 + 6 s^2 + 13 s + 20

```
num=[1];
den=[1 4];
sys1=tf(num,den)
sys2=tf([1],[1 2 5])
sys=series(sys1,sys2)
```

Transfer function:

1

----

s^3 + 6 s^2 + 13 s + 20

```
num=[1];
den=[1 4];
sys1=tf(num,den)
sys2=tf([1],[1 2 5])
sys=feedback(sys1,sys2,-1)
```

Transfer function:

s^2 + 2 s + 5

```
-----------------------
s^3 + 6 s^2 + 13 s + 21


num=[1];
den=[1 4];
sys1=tf(num,den)
sys2=tf([1],[1 2 5])
sys=sys1+sys2
sys=sys1-sys2
sys=sys1*sys2
```

controllability and observability

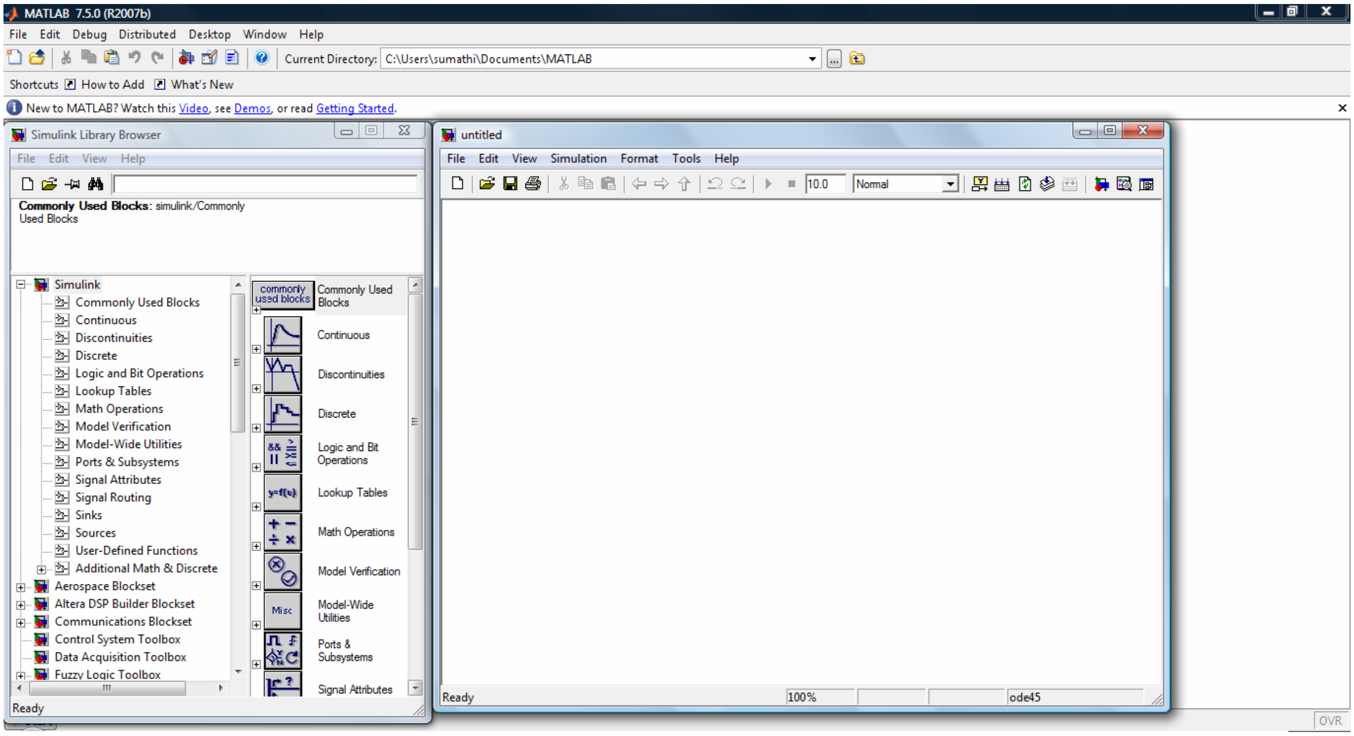
```
A=[0 1;-2 -3];
B=[0; 1];
C=[1 0];
D=[0];
sys=ss(A,B,C,D)
ctrb(sys)
obsv(sys)
canon(sys,'companion')


a =

       x1  x2

   x1   0   1

   x2  -2  -3

 b =

       u1

   x1   0

   x2   1



c =

       x1  x2

   y1   1   0
```

```
d =

      u1

  y1   0

Continuous-time model.

ans =

     0     1

     1    -3

ans =

     1     0

     0     1
```

Companion form

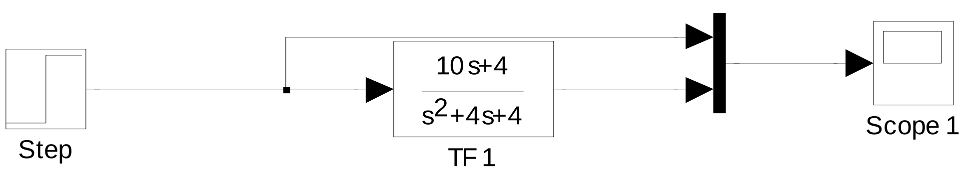
```
a =

      x1  x2

  x1   0  -2

  x2   1  -3

b =

      u1

  x1   1

  x2   0

c =

      x1  x2

  y1   0   1

d =

      u1

  y1   0
```

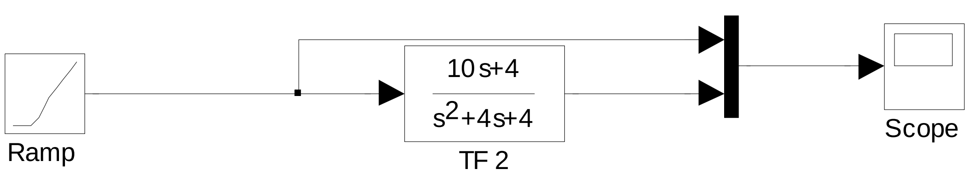
Simulink



Time Response

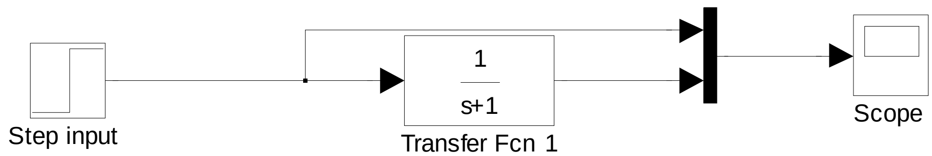
Step, Ramp responses



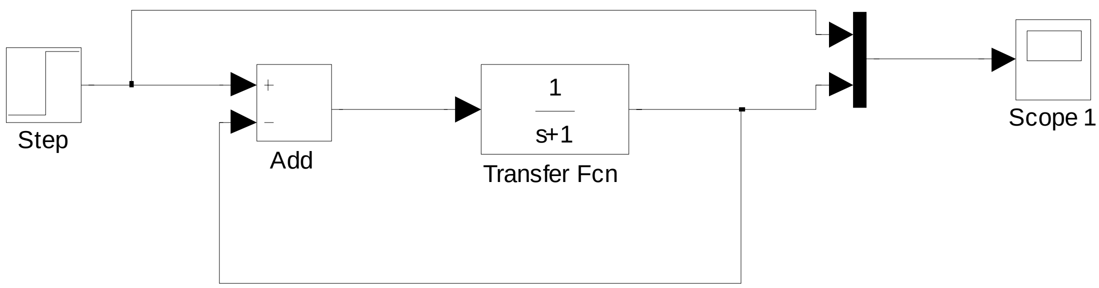


Open and closed systems: Transfer function model

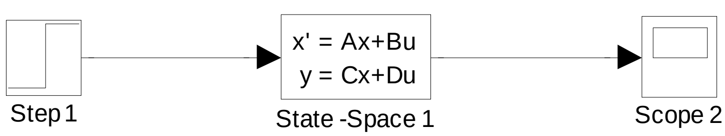
open loop system

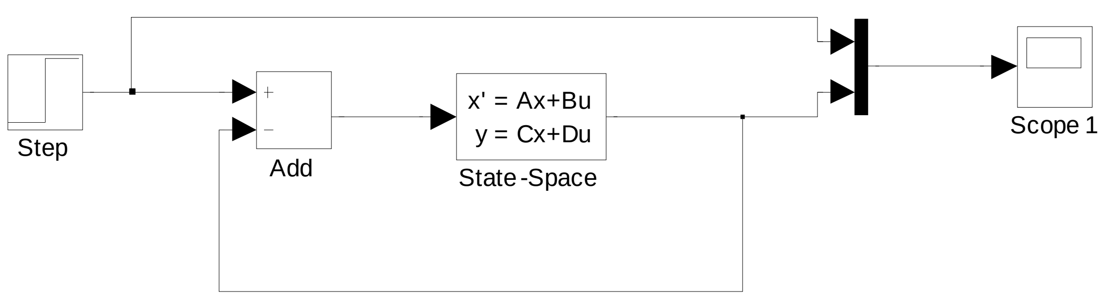


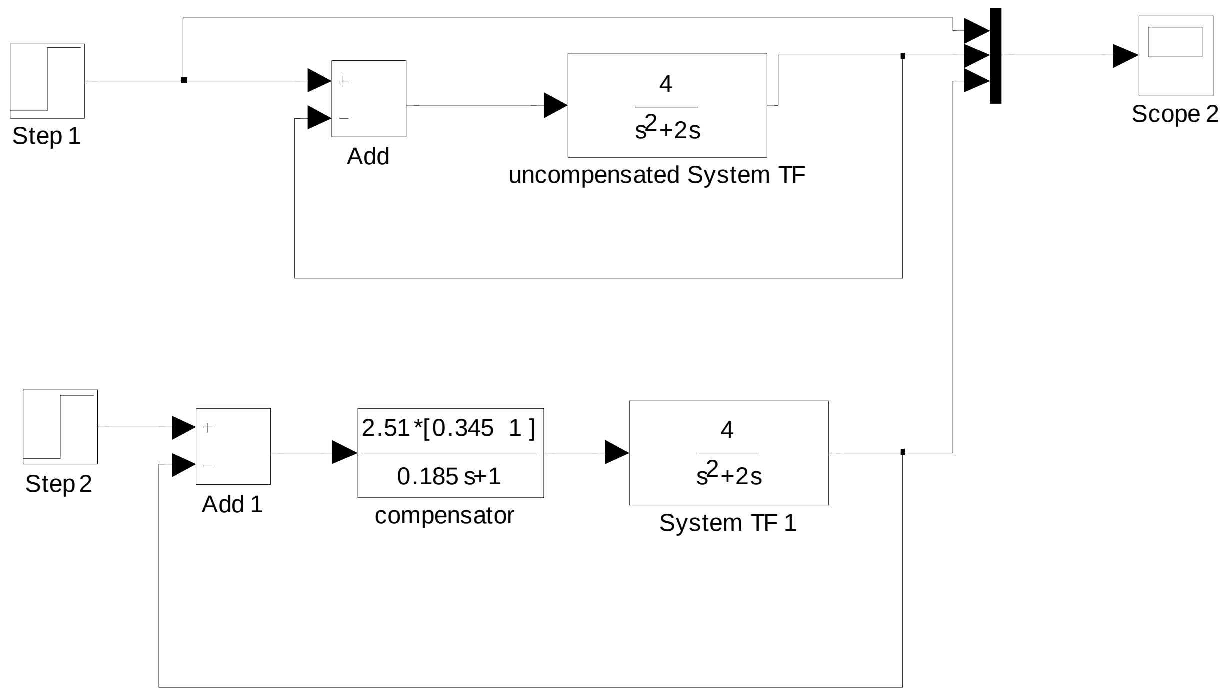
Closed loop system



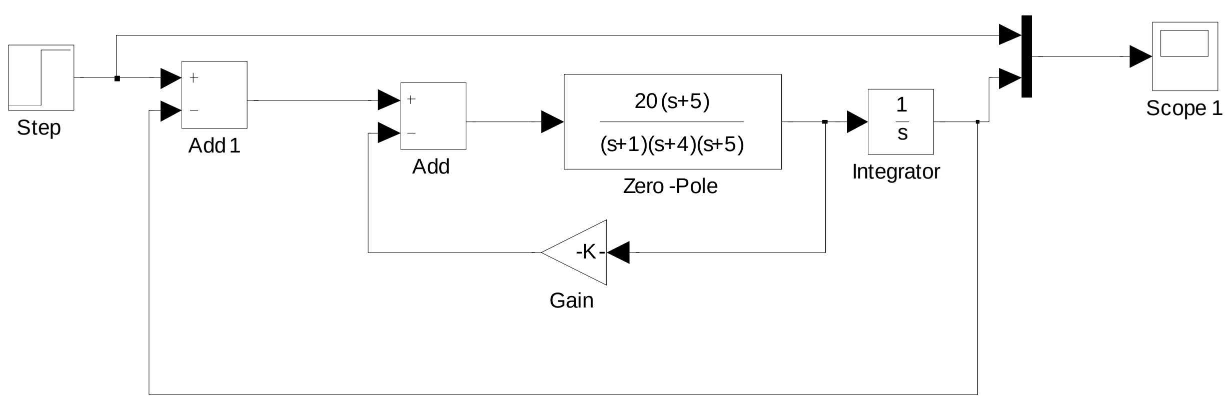
Open and closed loop responses: state space model



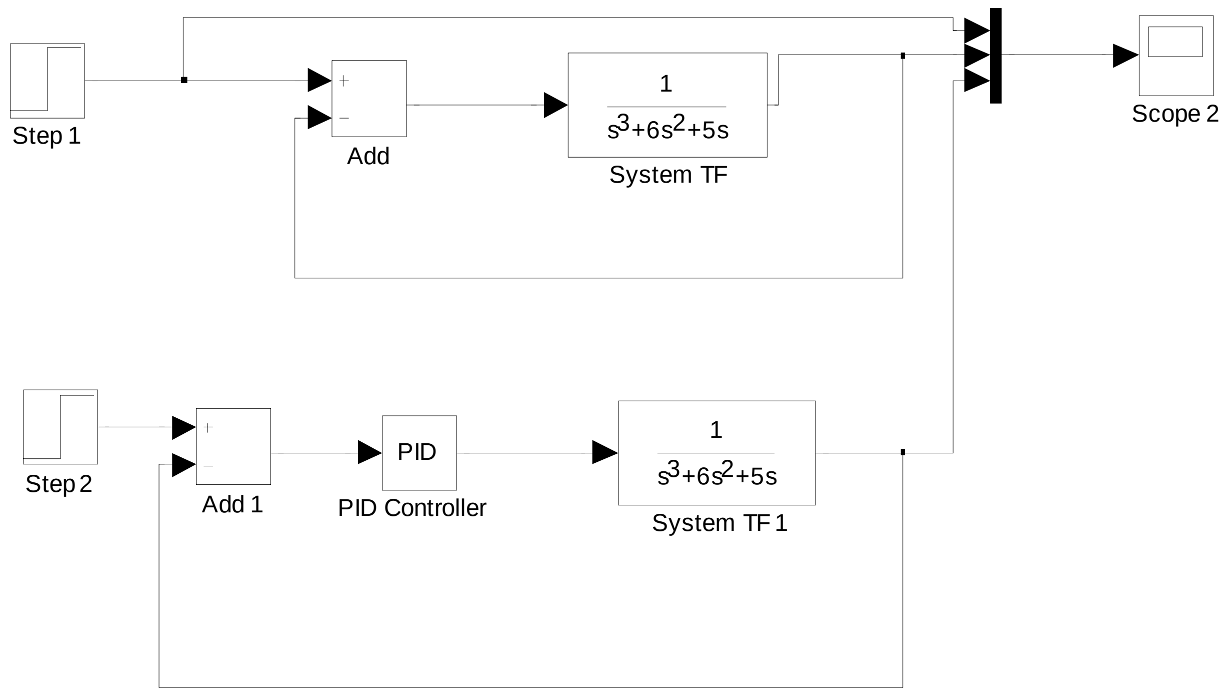




Step 1
Add
4
s2+2s
uncompensated System TF
Scope 2
Step 2
Add 1
2.51*[0.345 1]
0.185 s+1
compensator
4
s2+2s
System TF 1

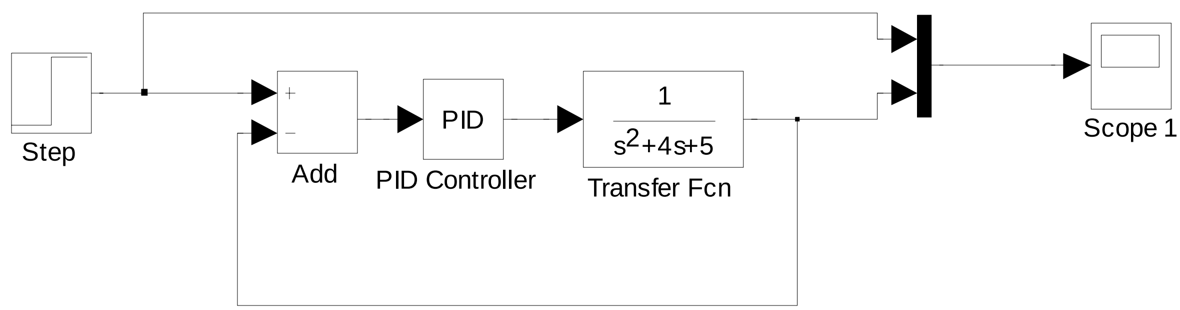


Step
Add 1
Add
20 (s+5)
(s+1)(s+4)(s+5)
Zero -Pole
1
s
Integrator
Scope 1
-K-
Gain



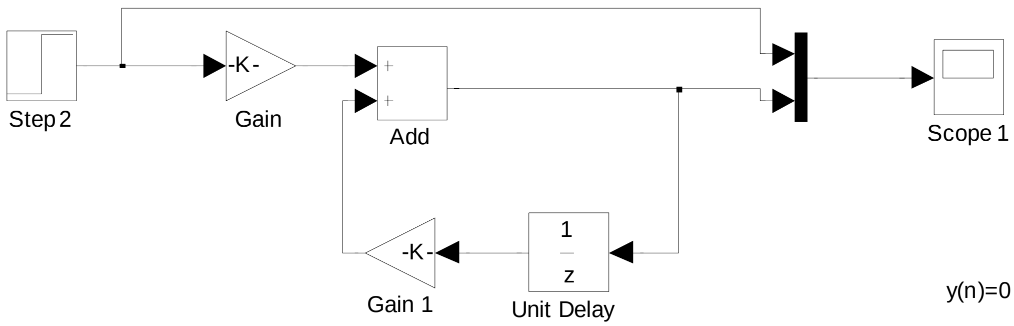
Step 1
Add
1
s3+6s2+5s
System TF
Scope 2
Step 2
Add 1
PID
PID Controller
1
s3+6s2+5s
System TF 1

PID



Step
Add
PID
PID Controller
1
s2+4s+5
Transfer Fcn
Scope 1

Discrete time systems



$y(n)=0.95y(n-1)+0.05x(n)$