ANALYSIS AND SIMULATION OF CONTROL SYSTEMS USING MATLAB

MATLAB can be used to analyze systems described by transfer functions or state space. Since transfer functions are ratio of polynomials, let us see how MATLAB handles polynomials.

POLYNOMIALS

Polynomials in MATLAB are represented by a vector containing the coefficients in descending order.

- For example, the roots of the polynomial

$p(s) = s^3 - 5s^2 - 1$

can be found as shown ⇒

```
>> p=[1 -5 0 1];
>> r=roots(p)
r =
    4.9593
    0.4698
   -0.4292
```

- The coefficients of the polynomial can be found from the roots as shown ⇒

```
>> p=poly(r)
p =
    1.0000   -5.0000    0.0000    1.0000
```

- Product of polynomials can be accomplished easily using MATLAB. Suppose we want to expand the following polynomial

  $n(s) = (3s^3 - s + 1)(s^2 + s)(s + 3)$

  The MATLAB commands for this operation is shown

```
p1=[3 0 -1 1];p2=[1 1 0];p3=[1 3];
>> p1p2=conv(p1,p2)
p1p2 =
     3     3    -1     0     1     0
>> n=conv(p1p2,p3)
n =
     3    12     8    -3     1     3     0
```

OR

```
>> conv(conv([3 0 -1 1],[1 1 0]),[1 3])
ans =
     3    12     8    -3     1     3     0
```


Copyright © Electrical Engineering Department, KFUPM.

- To evaluate any polynomial for any given value of the argument, the POLYVAL command can be used as shown

```
value=polyval(n,-4)
value =
       2244
value=polyval(n,-3)
value =
    0
```

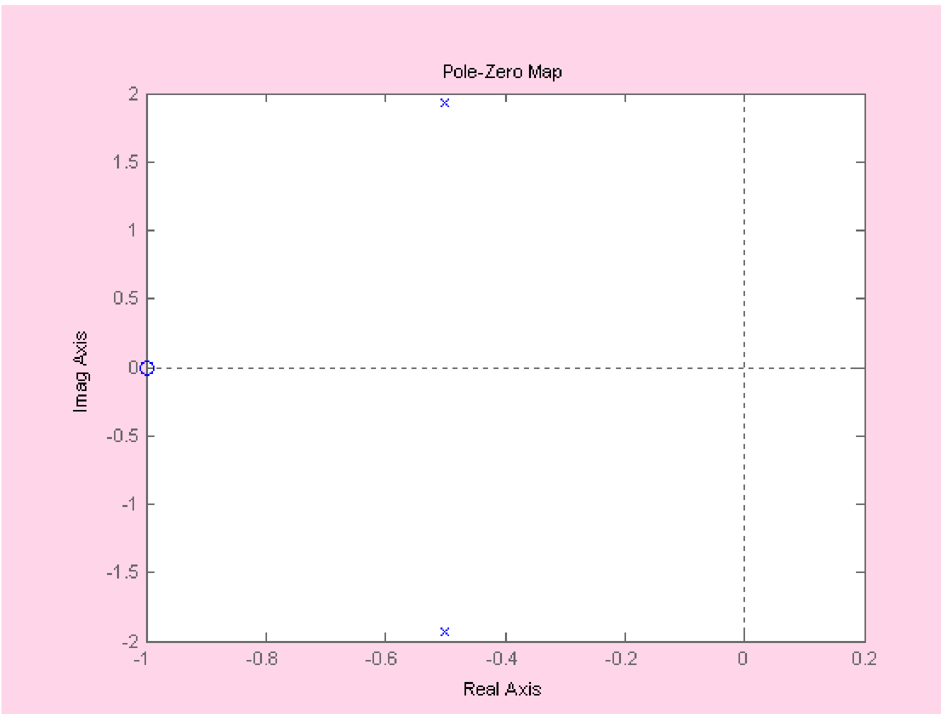
TRANSFER FUNCTIONS

- To find or plot the poles and zeros of any transfer function [TF], the PZMAP command can be used as shown. Suppose the TF is given by

$$G(s)=\frac{s+1}{s^2+s+4}$$

```
>> num=[ 1 1];den=[1 1 4];
>> [p,z]=pzmap(num,den)
p =
  -0.5000 + 1.9365i
  -0.5000 - 1.9365i
z =
   -1
```

```
pzmap([ 1 1],[1 1 4])
```



Copyright © Electrical Engineering Department, KFUPM.

- To write a system in a transfer function form, the PRINTSYS command can be used as shown

```
num=[ 1 1];den=[1 1 4];
>> printsys(num,den)

num/den =

      s + 1
  --------------
  s^2 +  s + 4
```

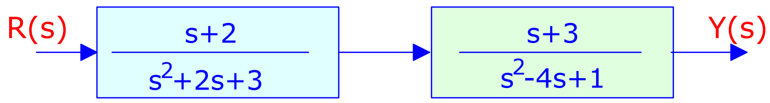
- To create a transfer function form, the TF command can be used as shown

```
num=[ 1 1];den=[1 1 4];
>> G=tf(num,den)

Transfer function:
   s + 1
---------------
s^2 + s + 4
```

BLOCK DIAGRAMS

Block diagram reduction can be carried out using MATLAB commands. The following operations are examples of block diagram reduction.

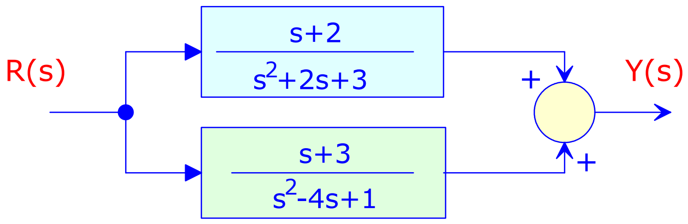
- Series connection




```
>> num1=[ 1 2];den1=[1 2 3 ];num2=[ 1 3];den2=[1 -4 1];
>> [num,den]=series(num1,den1,num2,den2)
num =
    0     0     1     5     6
den =
    1    -2    -4   -10     3
>> G=tf(num,den)

Transfer function:
       s^2 + 5 s + 6
-----------------------------
s^4 - 2 s^3 - 4 s^2 - 10 s + 3
```

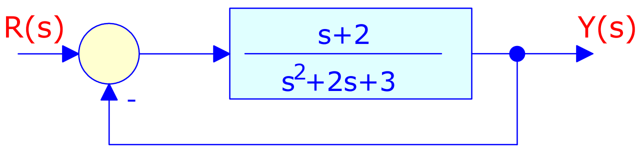

Copyright © Electrical Engineering Department, KFUPM.

- Parallel connection




```
>> num1=[ 1 2];den1=[1 2 3 ];num2=[ 1 3];den2=[1 -4 1];
>> [num,den]=parallel(num1,den1,num2,den2)
num =
     0     2     3     2    11
den =
     1    -2    -4   -10     3
>> G=tf(num,den)

Transfer function:
   2 s^3 + 3 s^2 + 2 s + 11
-----------------------------
s^4 - 2 s^3 - 4 s^2 - 10 s + 3
```

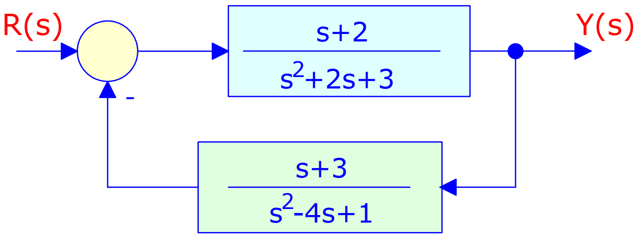
- Closed loop connection




```
>> num1=[ 1 2];den1=[1 2 3 ];
>> [num,den]=cloop(num1,den1,-1)
num =
     0     1     2
den =
     1     3     5

>> G=tf(num,den)

Transfer function:
    s + 2
-------------
s^2 + 3 s + 5
```


Copyright © Electrical Engineering Department, KFUPM.

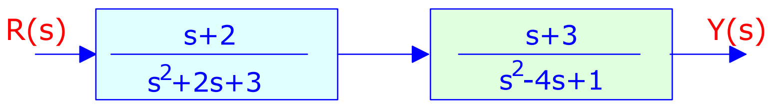
- Feedback connection




```
num1=[ 1 2];den1=[1 2 3 ];num2=[ 1 3];den2=[1 -4 1];
>> [num,den]=feedback(num1,den1,num2,den2,-1)
num =
     0     1    -2    -7     2
den =
     1    -2    -3    -5     9

>> printsys(num,den)

num/den =

     s^3 - 2 s^2 - 7 s + 2
   -----------------------------
   s^4 - 2 s^3 - 3 s^2 - 5 s + 9
```

- Connect

To derive state-space model for block diagram interconnection.



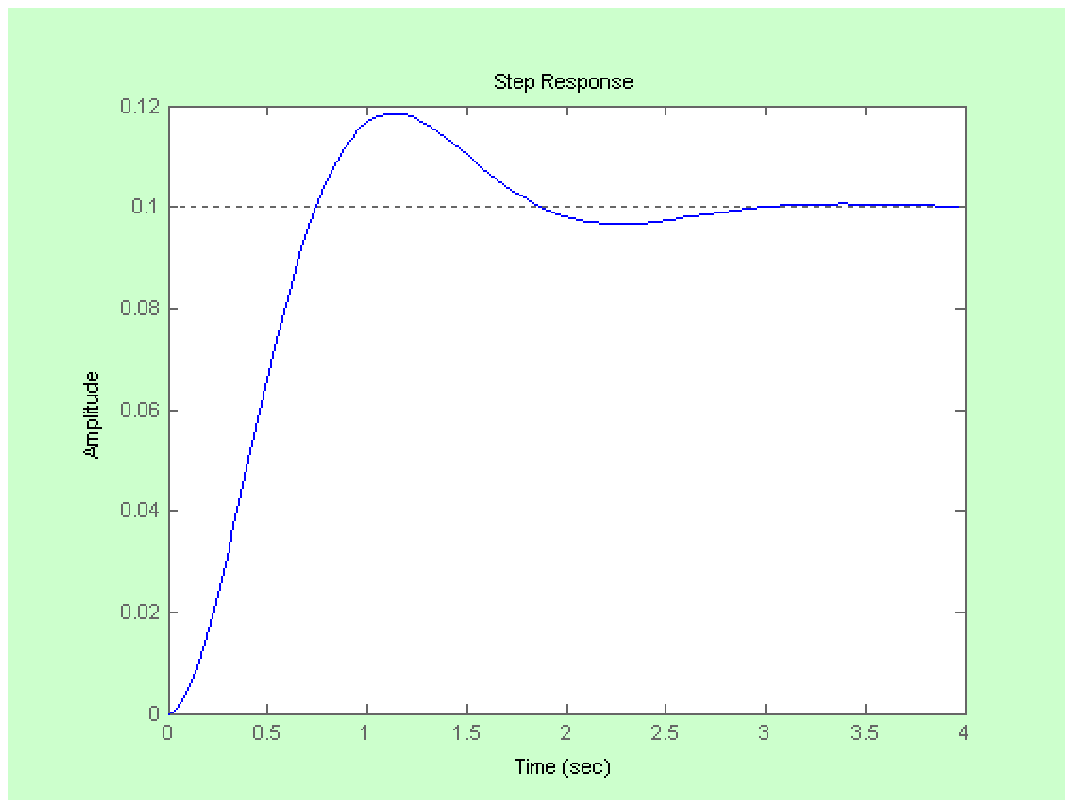

```
>> n1=[ 1 2];d1=[1 2 3 ];n2=[ 1 3];d2=[1 -4 1];nblocks=2;blkbuild;Q=[2 1];
State model [a,b,c,d] of the block diagram has 2 inputs and 2 outputs.
>> inputs=[1];outputs=[2];
>> [A,B,C,D]=connect(a,b,c,d,Q,inputs,outputs);
>> [NUM,DEN]=ss2tf(A,B,C,D,1)
NUM =
         0   -0.0000    1.0000    5.0000    6.0000
DEN =
    1.0000   -2.0000   -4.0000  -10.0000    3.0000
>> NUM=NUM(3:5);tf(NUM,DEN)

Transfer function:
        s^2 + 5 s + 6
-----------------------------
s^4 - 2 s^3 - 4 s^2 - 10 s + 3
```


Copyright © Electrical Engineering Department, KFUPM.

STEP RESPONSE

The step response of control systems is very important. It can be simulated using the STEP command as shown

```
num=[ 1 ];den=[1 3 10 ];
>> t=[0:0.01:5];
>> step(num,den)
```



Copyright © Electrical Engineering Department, KFUPM.