

# EE3114 Systems and Control

## Lab - 4

Semester A, 2018-19

# MatLab

**Practice** (30 minutes)

**Exercise** (60 minutes)

**Break** (10 minutes)

# Simulink

**Exercise** (60 minutes)

# MatLab

**Practice** (30 minutes)

**Exercise** (60 minutes)

## Some Useful MatLab Codes

ss - Create/convert to a state-space model  
tf - Create/convert to a transfer function model

pole - System poles

zero - System zeros

zpk – System zero-pole-gain

pzmap - Pole-zero map

rank – Rank of a matrix

det – Determinant of a matrix

eig - Eigenvalues of a matrix

[V,D]=eig(A) - Eigenvectors and eigenvalues of A

ss2ss - State coordinate transformation

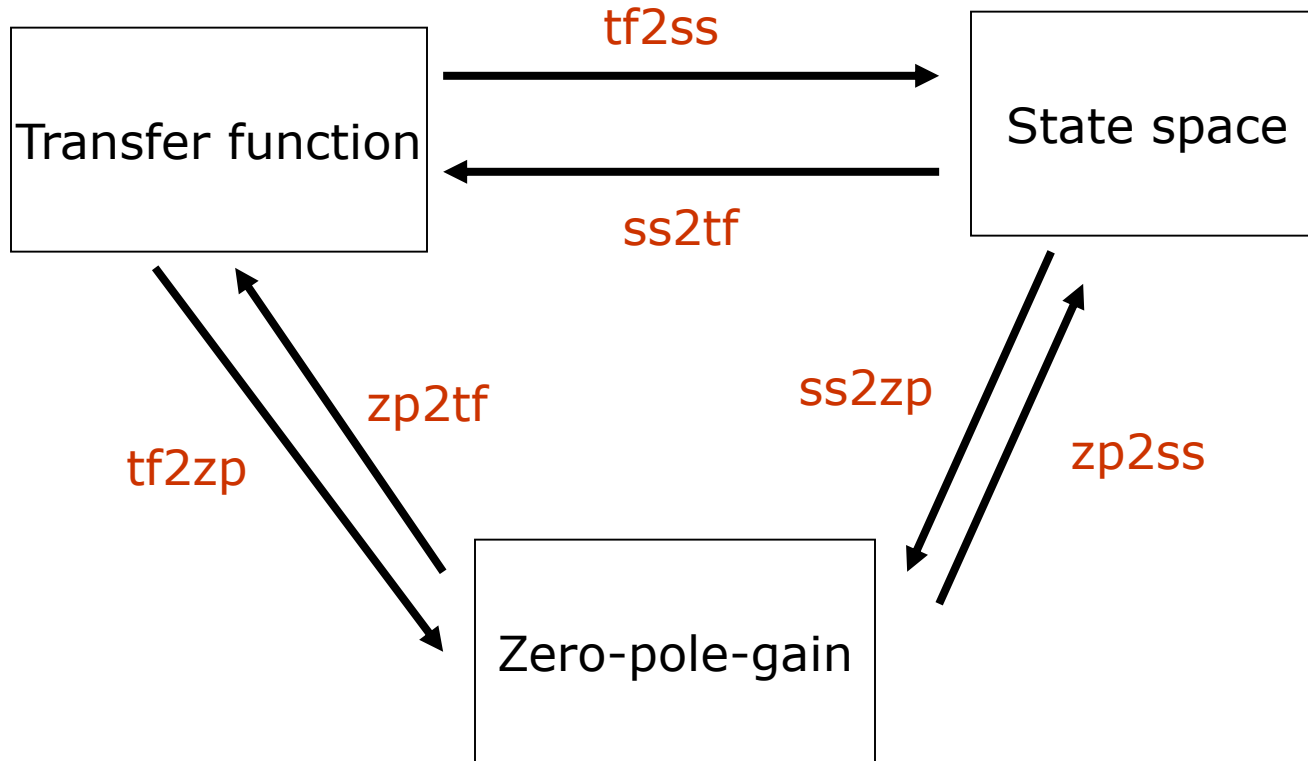
canon - State-space canonical forms

order – Order of a system

ctrb - Controllability matrix

obsv - Observability matrix

# Conversion between different models



# State-Space Representation of LTI Systems

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

**MatLab code:**

```
G=ss(A, B,C,D);
```

**Example:**

```
A=[1 2;3 4];
```

```
B=[5 6;7 8];
```

```
C=[9 10;11 12];
```

```
D=[13 14;15 16];
```

```
G=ss(A,B,C,D)
```

- Consider the following LTI system:

$$\ddot{y} + 6 \dot{y} + 5y = 4\ddot{u} + 3u$$

- Find its state-space model

## MatLab code

```
>> sys = ss([0 1; -5 -6],[0;1],[3,4],0)
```

a =

    x1  x2  
x1  0   1  
x2 -5  -6

c =

    x1  x2  
y1  3   4

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -5 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$



$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

b =

    u1  
x1  0  
x2  1

d =

    u1  
y1  0

$$y = \begin{bmatrix} 3 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

**Example:**

Converse to the following transfer function to state-space representation :

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

Hint: Use function tf2ss()

**MatLab code:**

```
num=[1 2 3];  
den=[1 3 3 1];  
[A B C D]=tf2ss(num,den)
```

**Result:**

A =

$$\begin{bmatrix} -3 & -3 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

B =

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

C =

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

D =

$$0$$

# Unforced system response to initial conditions

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x}, & \mathbf{x}(0) &= \mathbf{x}_0 \\ y &= \mathbf{C}\mathbf{x}\end{aligned}$$

## Example:

Calculate the unforced response of the following system:

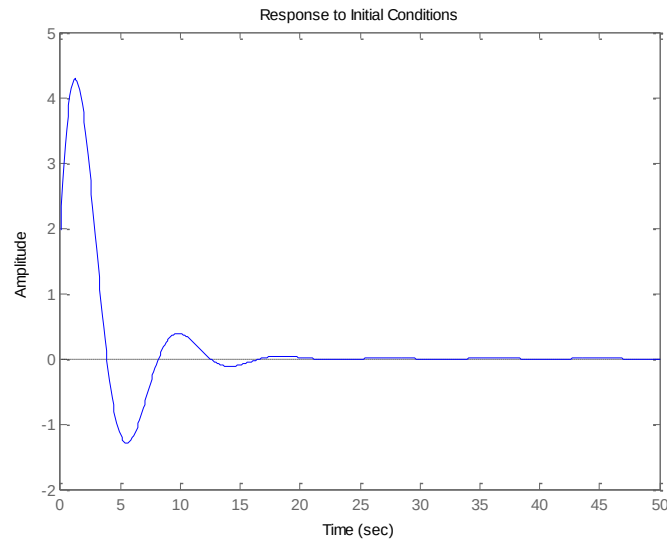
$$\dot{\mathbf{x}} = \begin{bmatrix} -0.5572 & -0.7814 \\ 0.7814 & 0 \end{bmatrix} \mathbf{x}, \quad \mathbf{x}_0 = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$$

$$\mathbf{y} = \begin{bmatrix} 1.9691 & 6.4493 \end{bmatrix} \mathbf{x}$$

## MatLab code:

```
A=[-0.5572 -0.7814;0.7814 0];  
B=[0; 0];  
C=[1.9691 6.4493];  
D=[0];  
x0=[1;0];  
t=0:0.1:50; %simulation time t  
initial(A,B,C,D,x0,t); %calculate the unforced response  
plot(t,x)
```

## Result:



# Controllability and Observability

## Example:

Determine the controllability and observability of the following system:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -3 & 1 \\ 1 & -3 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \mathbf{u}(t)$$

$$\mathbf{y}(t) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \mathbf{x}(t)$$

## MatLab code:

```
A=[-3 1;1 -3];  
B=[1 1;1 1];  
C=[1 1;1 -1];  
D=0;
```

```
Matrix_controllability=ctrb(A,B); %Compute the controllability matrix  
rank_Matrix_controllability=rank(Matrix_controllability) % Matrix rank
```

```
Matrix_observability=obsv(A,C); % Compute the observability matrix  
rank_Matrix_observability=rank(Matrix_observability)
```

## Result:

rank\_Matrix\_controllability =

1

rank\_Matrix\_observability =

2

→ This is an uncontrollable but observable system

# Designing state-feedback controllers

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t)$$

$$u(t) = -\mathbf{K}\mathbf{x}(t)$$

## Example:

For the following system, determine the state-feedback control gain so as to guarantee that the expected poles of the closed-loop system are: [-2 -3 -2+i -2-i]

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1.836 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 36.79 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 5.56 \\ 0 \\ -17.36 \end{bmatrix} u(t)$$

**MatLab code:**

```
A=[0 1 0 0; 0 0 -1.836 0; 0 0 0 1; 0 0 36.79 0];  
B=[0;5.56;0;-17.36];  
Expected_poles=[-2 -3 -2+i -2-i];  
K=acker(A,B,Expected_poles)
```

**Result:**

K =

-0.1737 -0.2838 -3.9606 -0.6093

**So,**

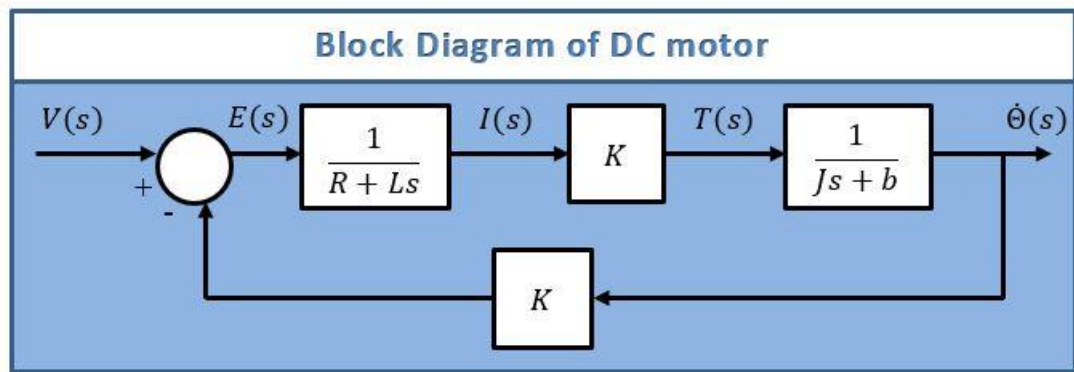
$$u = -Kx(t) = -[-0.1737 \quad -0.2838 \quad -3.9606 \quad -0.6093] x(t)$$

## Exercise (60 minutes)

### DC Motor Position Control using MatLab State-Space Analysis

**Table 1. Physical Parameters of a DC Motor**

|                                     |                                     |
|-------------------------------------|-------------------------------------|
| Motor constant $K$                  | $45 \times 10^{-3} \text{ N.m/A}$   |
| Rotor inertia $J$                   | $35 \times 10^{-6} \text{ N.m.s}^2$ |
| Motor viscous friction constant $b$ | $2 \times 10^{-3} \text{ N.m.s}$    |
| Resistance $R$                      | $1 \Omega$                          |
| Inductance $L$                      | $0.5 \text{ H}$                     |



Transfer function of the motor:

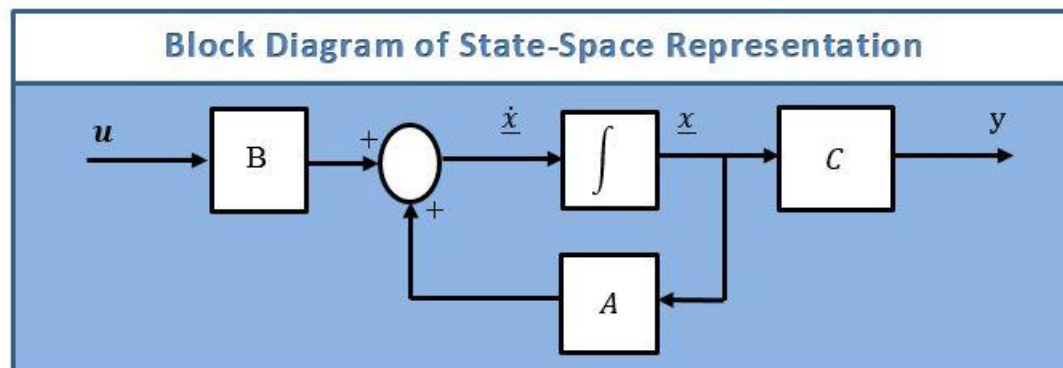
$$G(s) = \frac{\Theta(s)}{V(s)} = \frac{K}{s((R+Ls)(Js+b)+K^2)}$$

State-space representation:

$$\begin{aligned}\frac{dx}{dt} &= Ax + Bu \\ y &= Cx\end{aligned}$$

where

$$u = V, \quad y = \theta, \quad x = \begin{bmatrix} \theta \\ \dot{\theta} \\ i \end{bmatrix}, \quad \frac{dx}{dt} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ di/dt \end{bmatrix}$$



## Task 1.1 State-Space Representation

(**Hint** is given on the next page)

- Find the state-space representation of the DC motor
- Find the order of the system
- Construct the controllability and observability matrices of the system
- Determine if the system is controllable?
- Determine if the system is observable?

\_\_\_\_/0.5

Hint:

DC Motor model

$$\mathbf{u} = V, \quad y = \theta, \quad \mathbf{x} = \begin{bmatrix} \theta \\ \dot{\theta} \\ i \end{bmatrix}, \quad \frac{d\mathbf{x}}{dt} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ di/dt \end{bmatrix}$$

$$\dot{\theta} = \dot{\theta}$$

$$\ddot{\theta} = \frac{Ki - b\dot{\theta}}{J}$$

$$\frac{di}{dt} = \frac{V - K\dot{\theta} - Ri}{L}$$

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ di/dt \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -b/J & K/J \\ 0 & -K/L & -R/L \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/L \end{bmatrix} V$$

$$y = [1 \quad 0 \quad 0] \begin{bmatrix} \theta \\ \dot{\theta} \\ i \end{bmatrix}$$

... next page

%MatLab code

% Set the motor parameters

$K = 45\text{E-}3$ ;

$J = 3.5\text{E-}5$ ;

$b = 2\text{E-}3$ ;

$R = 1$ ;

$L = 0.5$ ;

% Define the state space variables

$A = [0 \ 1 \ 0$ ;

$0 \ -b/J \ K/J$ ;

$0 \ -K/L \ -R/L]$ ;

$B = [0; 0; 1/L]$ ;

$C = [1 \ 0 \ 0]$ ;

$D = 0$ ;

% Create the system

Motor\_SS = ?

... .. next page

% Create controllability and observability matrices

controllability = ?

observability = ?

% Check the system order, rank; determinants of the controllability and

% observability matrices

sys\_order = ?

det\_control = ?

rank\_control = ?

det\_observ = ?

rank\_observ = ?

... .. next page

Next, determine  $K$  to satisfy the following requirements:

### Controller Design Criteria

- 2% settling time to be less than 0.5s
- Overshoot of the step response to be limited to 10%
- Zero steady-state error

## Task 1.2: Pole Placement

(**Hint** is given on the next page)

The poles of a standard second-order system are located at

$$\left(-\zeta \pm \sqrt{\zeta^2 - 1}\right) \omega_n$$

where  $\zeta$  is the damping ratio and  $\omega_n$  is the natural frequency.

To determine the desired pole positions:

$$\text{Overshoot (os)} = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \Rightarrow \zeta = \frac{-\ln(os)}{\sqrt{\pi^2 + \ln(os)^2}}$$

$$2\% \text{ Settling Time (Ts)} = \frac{-\ln(2\%)}{\zeta\omega_n}$$

Calculate a damping ratio and natural frequency that would ensure the poles of the system are in appropriate positions to meet the design criteria.

Hint:

$$\text{Overshoot} < 0.1 = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}$$

$$\zeta > \frac{-\ln(0.1)}{\sqrt{\pi^2 + \ln(0.1)^2}} = 0.5912$$

$$2\% \text{ Settling Time} < 0.5 = \frac{-\ln(0.02)}{\zeta\omega_n}$$

$$\omega_n > \frac{-\ln(0.02)}{0.5\zeta} = 13.2352$$

Poles should be placed further to the left of:

$$(-\zeta \pm \sqrt{\zeta^2 - 1})\omega_n = -7.8240 \pm j10.6749$$

depending on the selection of  $\zeta$  and  $\omega_n$

... next page

## Hint:

Desired overshoot 10%:  $os = 0.1$

Desired 2% settling time:  $ts = 0.5$

$$\text{So, } \zeta = \frac{-\ln(os)}{\sqrt{\pi^2 + \ln(os)^2}}, \omega_n \cong \frac{-\ln(0.02)}{\zeta * ts}$$

Poles:

$$p1, p2: \left( -\zeta \pm \sqrt{\zeta^2 - 1} \right) \omega_n$$

$$p3: -20$$

% MatLab code

% Calculate the damping ratio and natural frequency

zeta = ?

wn = ?

% Round these up to ensure greater than the desired criteria

zeta = ceil(zeta\*100)/100

wn = ceil(wn)

% Set the desired poles of the system using the round-up numbers

p1 = ?

p2 = ?

p3 = ?

### Task 1.3: State-Feedback Control

(**Hint** is given on the next page)

Assuming the whole state vector  $\mathbf{x}$  is available for feedback.

Design a controller to satisfy the design criteria using a state feedback law

$$u = r - K_c \mathbf{x}$$

where  $r$  is the reference input and the state-space system becomes

$$\frac{d\mathbf{x}}{dt} = (A - BK_c)\mathbf{x} + Br$$

$$y = C\mathbf{x}$$

- By using pole-placement technique, determine the feedback gain matrix  $K_c$

Hint: *Use two dominant poles to control the system so as to ensure that the third pole is real and faster than the other poles.*

- Check the response of the system to a step input, to see whether it meets all of the design criteria or not?

## Hint:

% Determine the feedback matrix

$K_c = ?$

% Create a closed-loop state-space system

Motor\_SS\_CL = ?

% Implement a step response

figure(1), step(Motor\_SS\_CL)

stepinfo(Motor\_SS\_CL)

% Calculate the input scale factor (this is only for rescaling)

len = size(A,1);

Nxu = [A B; C D] \ [zeros(len,1); 1];

N = Nxu(len+1) + Kc\*Nxu(1:len);

% Create an updated system

Motor\_SS\_CL1 = ss(A-B\*Kc, N\*B, C, D)

figure(2), step(Motor\_SS\_CL1)

stepinfo(Motor\_SS\_CL1)

# Simulink

**Exercise** (60 minutes)

**Some Useful  
Simulink Blocks:**

*gain*

*integrator*

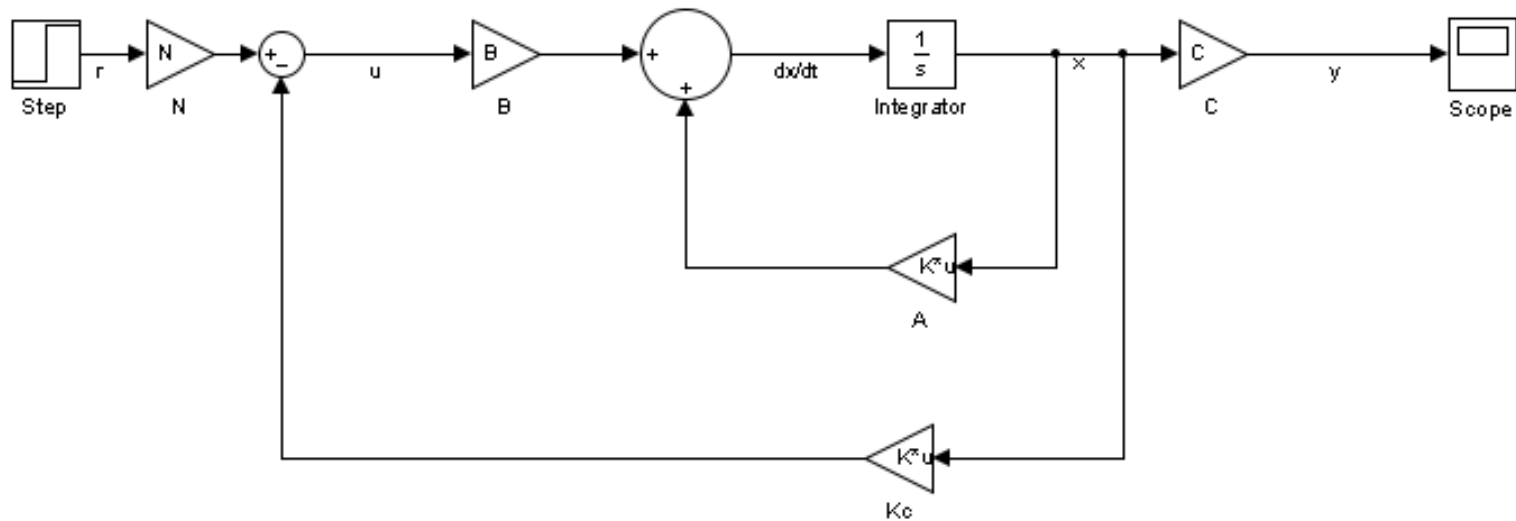
*step*

*scope*

*add*

## Task 2.1: State Controller

- Use Simulink to create a model for your state-space controller
- Using a step input to validate your model



## Task 2.2 State Estimator

In reality, one cannot always measure the whole state vector  $\mathbf{x}$   
So, it is needed to build an Observer to estimate them.

Now, add an observer to your control model:

- Calculate the observer gain matrix (MatLab may be used)

Hint: *Poles of the observer should be about  $4 \times |\text{poles of controller}|$*

*(Note: MatLab will not place 2 poles to be in the same position)*

- On Simulink, add the state estimator to your state space controller and validate the output

(See: next page)

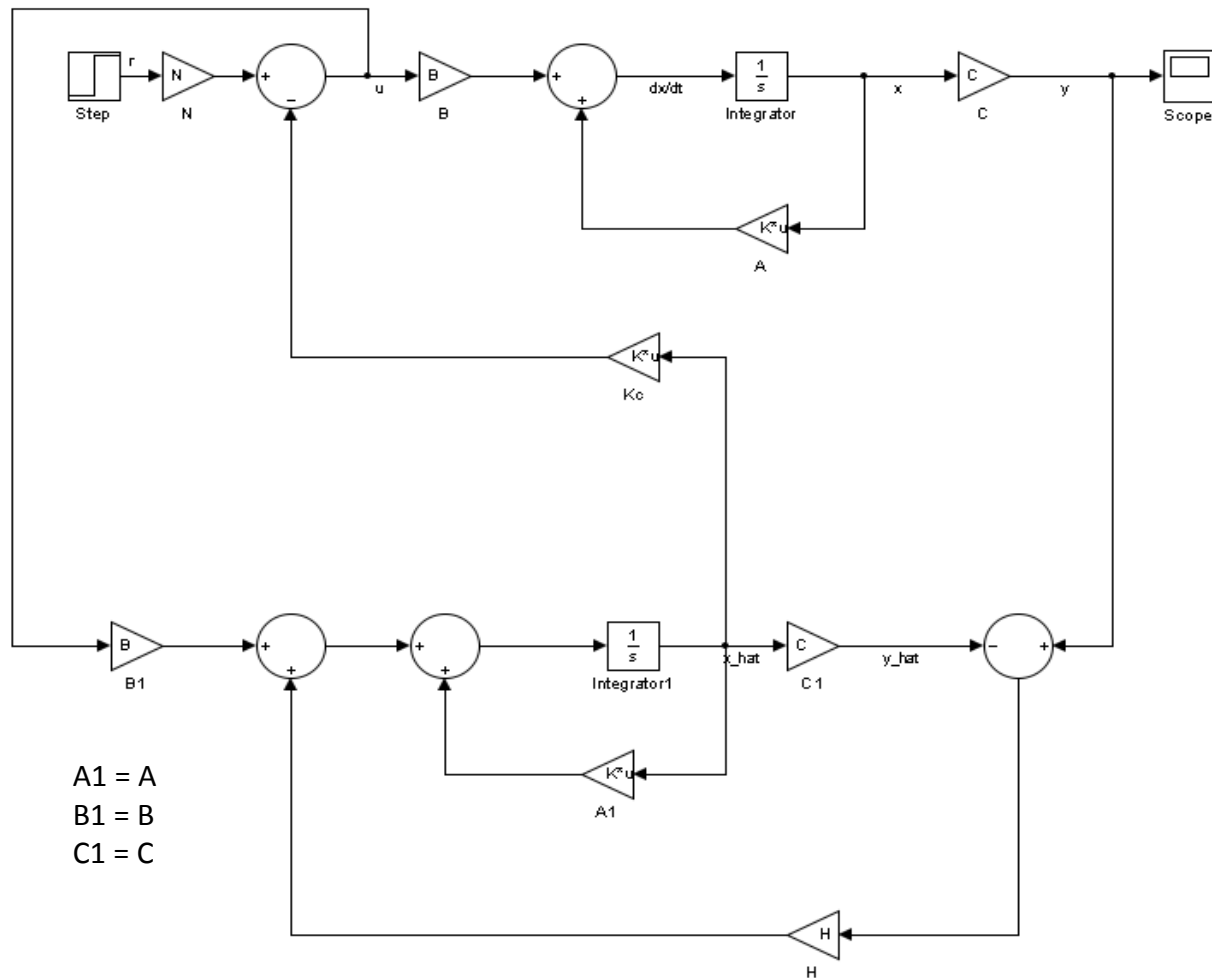
% Calculate the observer roots

$o1 = 4*abs(p1)$

$o2 = 4*abs(p2)+1$

$o3 = 4*abs(p3)$

$H = place(A',C',[o1\ o2\ o3])'$



**END**