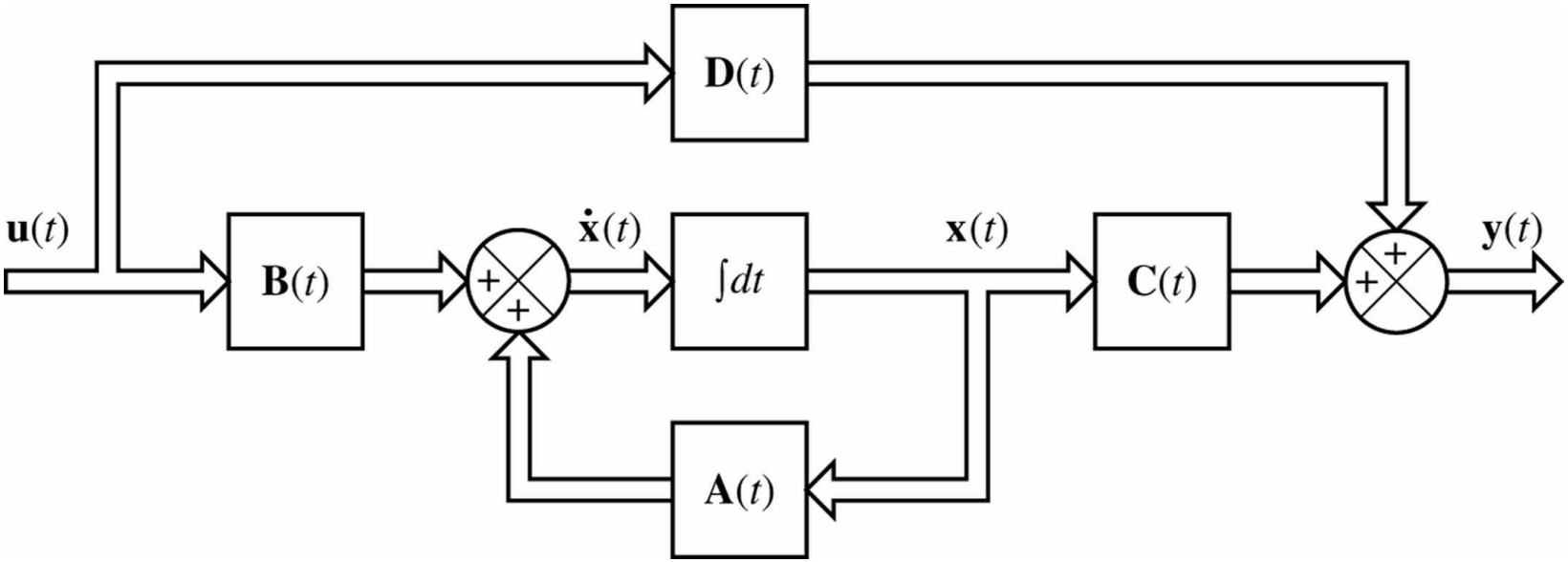
EEEN 460
Optimal Control

Spring 2020

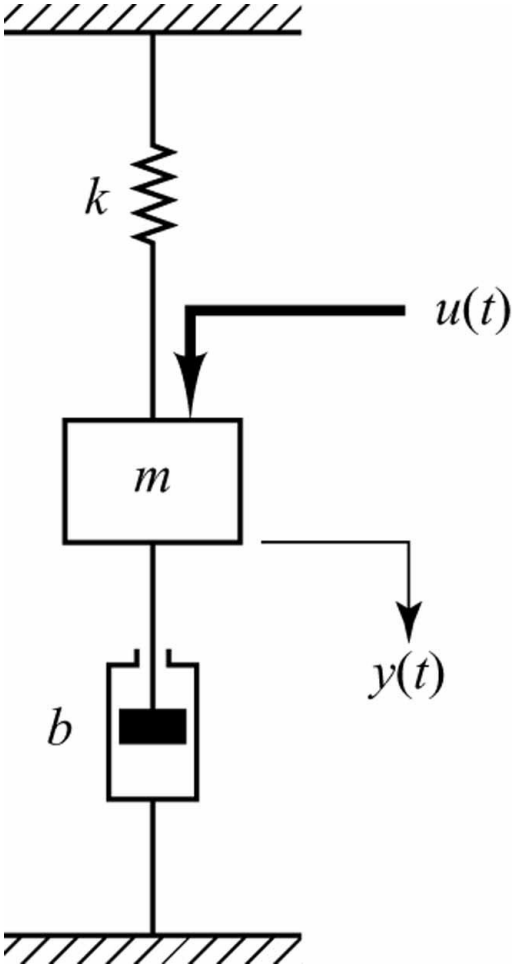
Lecture 6
Controller Design by Pole Placement
Controller Design by LQR

State space Representation

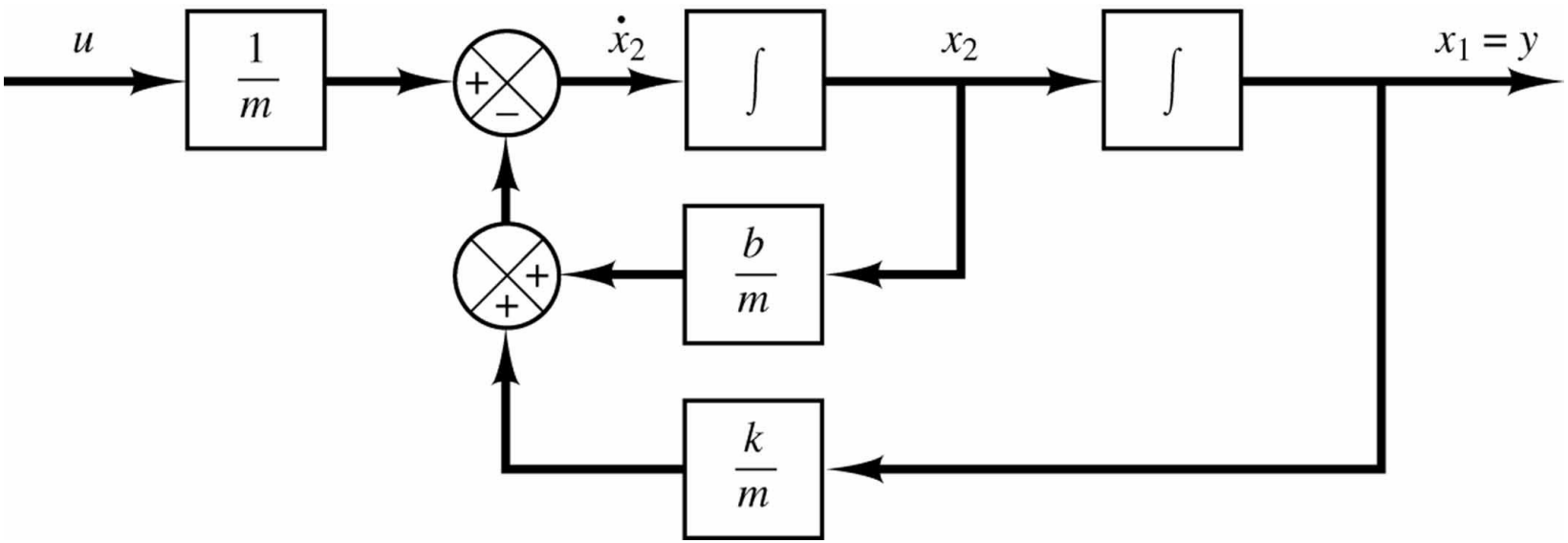
Block diagram of the linear, continuous-time control system represented in state space.



Spring-Mass-Damper system

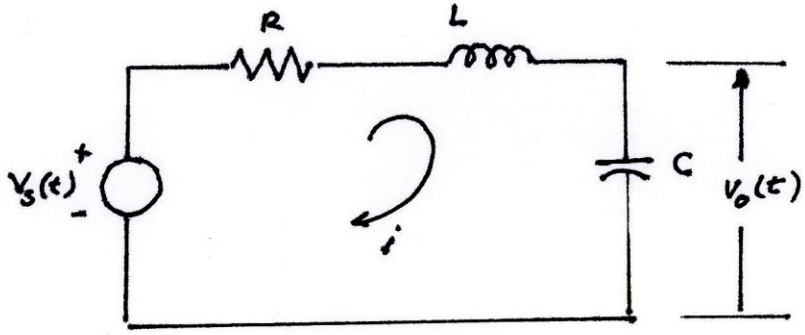


State-Space Representation of the Spring-Mass-Damper system



STATE - SPACE METHOD

Example 1.



$R = 1\,\Omega$

$L = 1/4\,H$

$C = 4/3\,F$

$$i = i_L = i_C = C\frac{dv}{dt}$$

$$R i_L + L\frac{di_L}{dt} + v_C = v_s$$

] state equations

$$v_o = v_C$$

] output equation

$$\frac{di_L}{dt} = -\frac{R}{L} i_L - \frac{1}{L} v_c + \frac{1}{L} v_s$$

$$\frac{dv_c}{dt} = \frac{1}{C} i_L$$

$$\begin{bmatrix} \frac{di_L}{dt} \\ \frac{dv_C}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} v_s$$

$$v_o = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix}$$

$i_L = x_1$

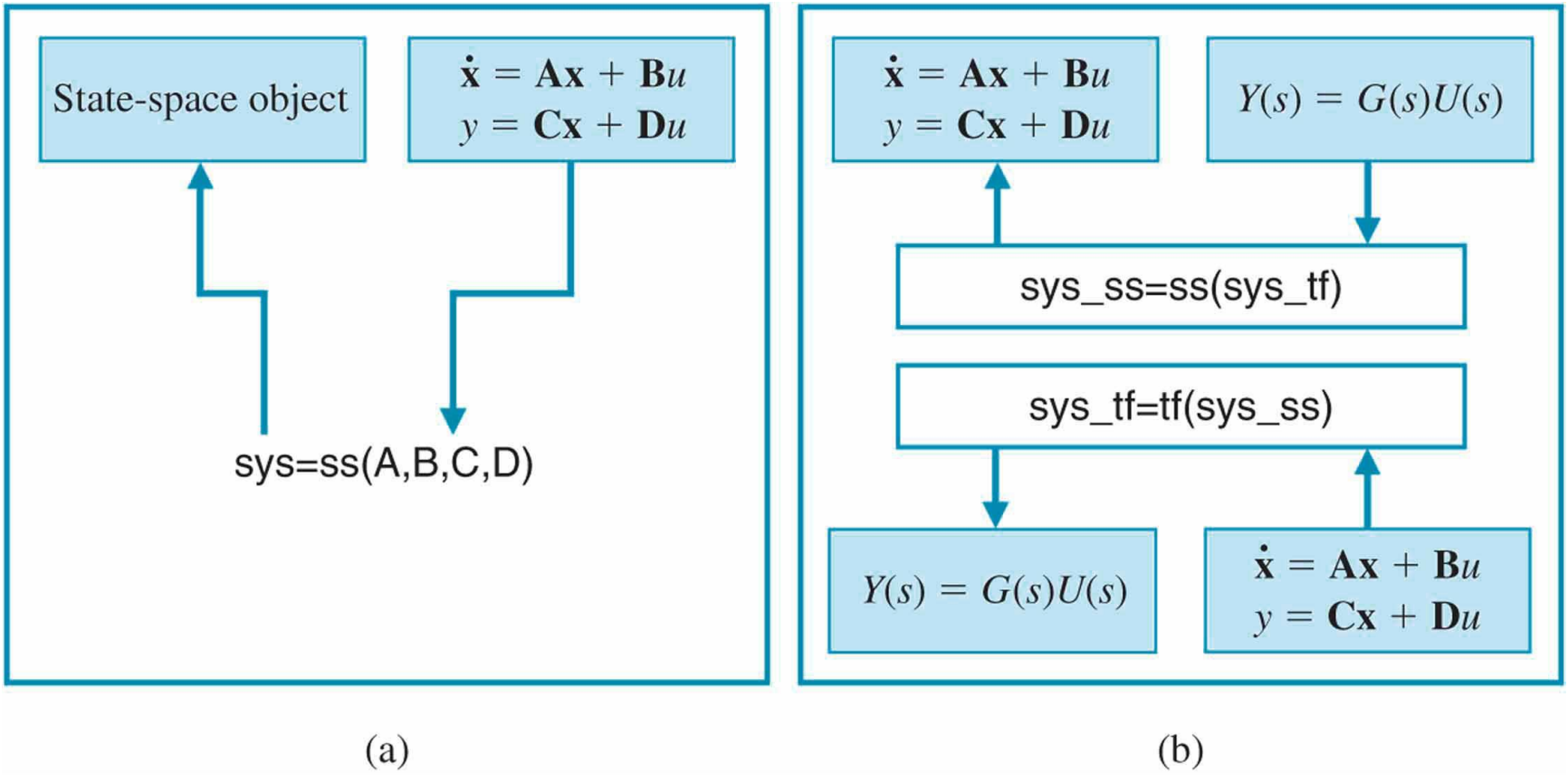
$v_C = x_2$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -4 & -3 \\ \frac{3}{4} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 4 \\ 0 \end{bmatrix} u_o$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} u_o$$

Conversion between linear system models

(a) The **ss** function**.** (b) Linear system model conversion.



Conversion from transfer function to a state-space representation

convert.m

```
% Convert G(s) = (2s^2+8s+6)/(s^3+8s^2+16s+6)
% to a state-space representation
%
num=[2 8 6]; den=[1 8 16 6]; sys_tf=tf(num,den);
sys_ss=ss(sys_tf);
```

(a)

```
>>convert
a =
        x1      x2      x3
 x1    -8      -4      -1.5
 x2     4       0       0
 x3     0       1       0

b =
        u1
 x1     2
 x2     0
 x3     0

c =
        x1      x2      x3
 y1     1       1       0.75

d =
        u1
 y1     0
```

(b)

Conversion from state-space representation to a transfer function

Find the transfer function of the following system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -80 & -28 & -7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ -14 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

```
[num, den] = ss2tf(A, B, C, D)
```

Conversion from state-space representation to a transfer function

```
%Finding the transfer function
%System is represented in State-Space
%**** Enter matrices A,B,C and D
A=[0 1 0;0 0 1;-80 -28 -7];
B=[0; 1;-14];
C=[1 0 0];
D=[0];
[num,den]=ss2tf(A,B,C,D)
```

```
num =

         0         0    1.0000   -7.0000


den =

    1.0000    7.0000   28.0000   80.0000
```

Computing the step response

Obtain the unit step response of the system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

We put this system into following form

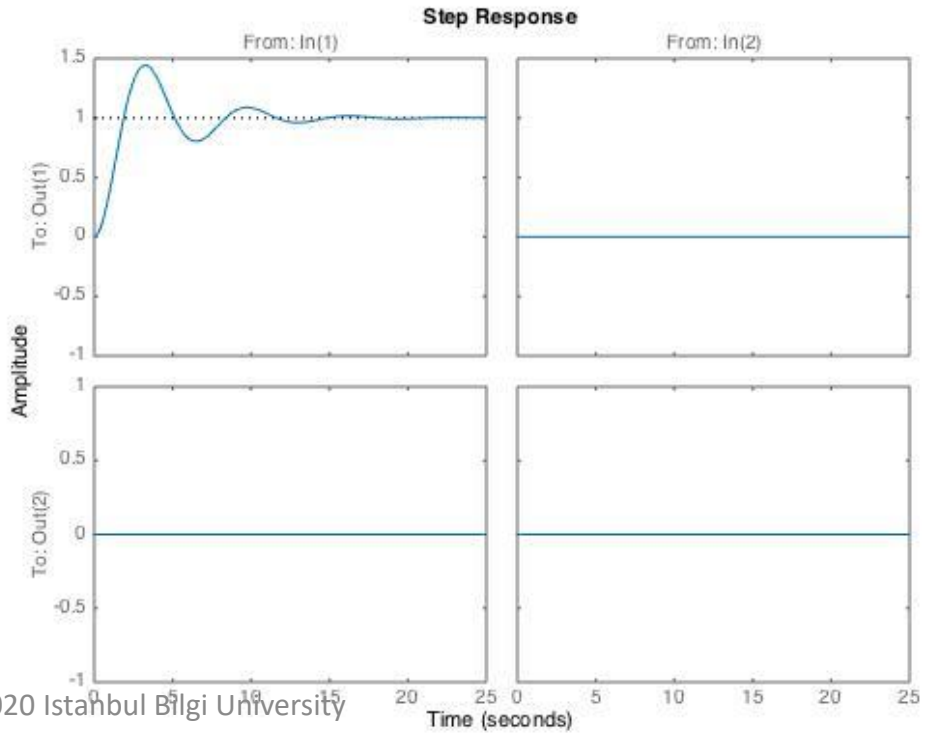
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ -1 & -0.5 \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}}_{B} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}}_{C} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}_{D} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

step (A, B, C, D)

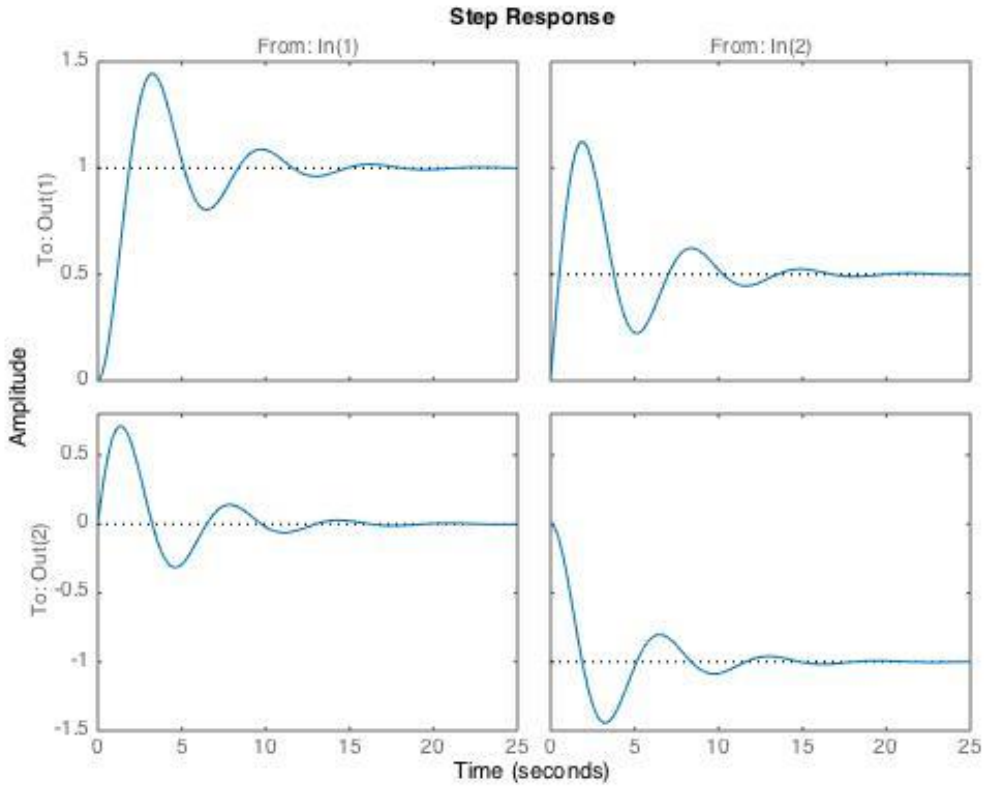
Computing the step reponse

```
%Unit Step Response
%System is represented in State-Space
%**** Enter matrices A,B,C and D
A=[0 1;-1 -0.5];
B=[0 0;1 0];
C=[1 0;0 0];
D=[0 0;0 0];
%Calculate the response and plot
step(A,B,C,D);
```



Computing the step response

```
%Unit Step Response
%System is represented in State-Space
%**** Enter matrices A,B,C and D
A=[0 1;-1 -0.5];
B=[0 1;1 0];
C=[1 0;0 1];
D=[0 0;0 0];
%Calculate the response and plot
step(A,B,C,D);
```



Plotting the root locus diagram

Plotting the Root Locus

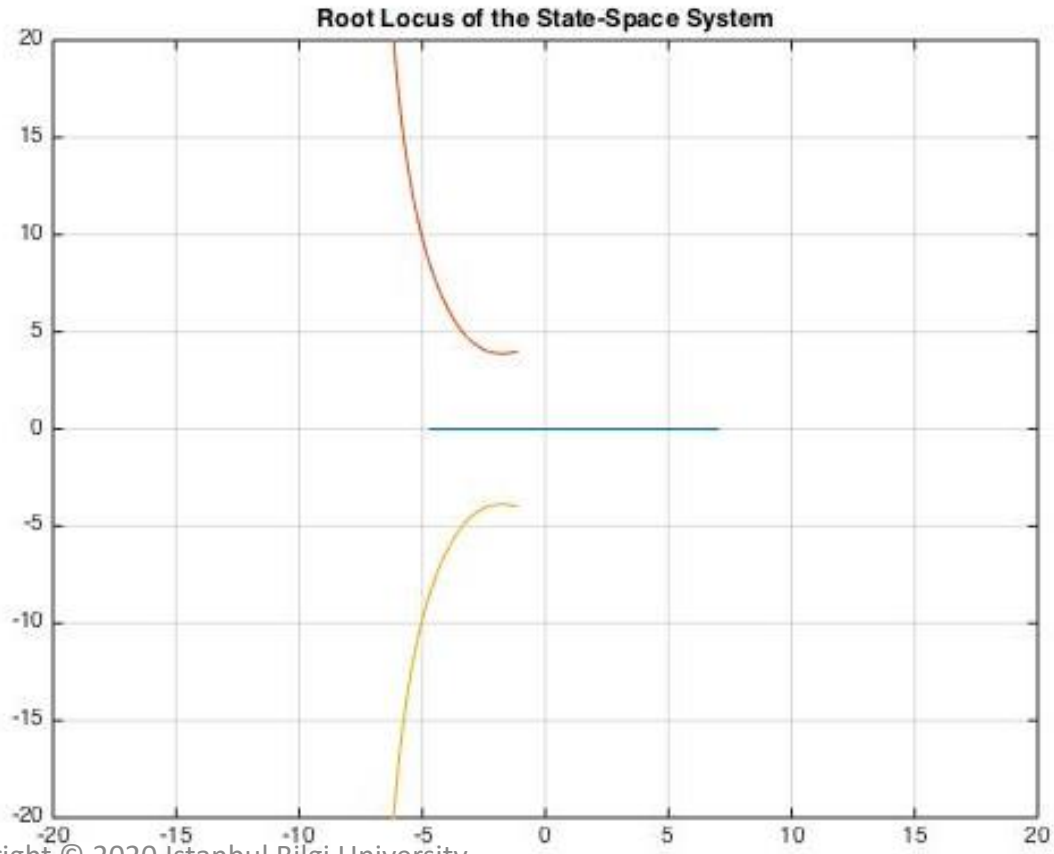
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -80 & -28 & -7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ -14 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

The root locus command

rlocus (A, B, C, D)

```
%Root Locus
%System is represented in State-Space
%**** Enter matrices A,B,C and D
A=[0 1 0;0 0 1;-80 -28 -7];
B=[0; 1;-14];
C=[1 0 0];
D=[0];
%K varies from 0 to 400
K=0:0.1:400;
% plot the root locus
r=rlocus(A,B,C,D);
plot(r,'-');v=[-20 20 -20 20];axis(v)
grid
title('Root Locus of the State-Space System')
```



Plotting the Nyquist diagram

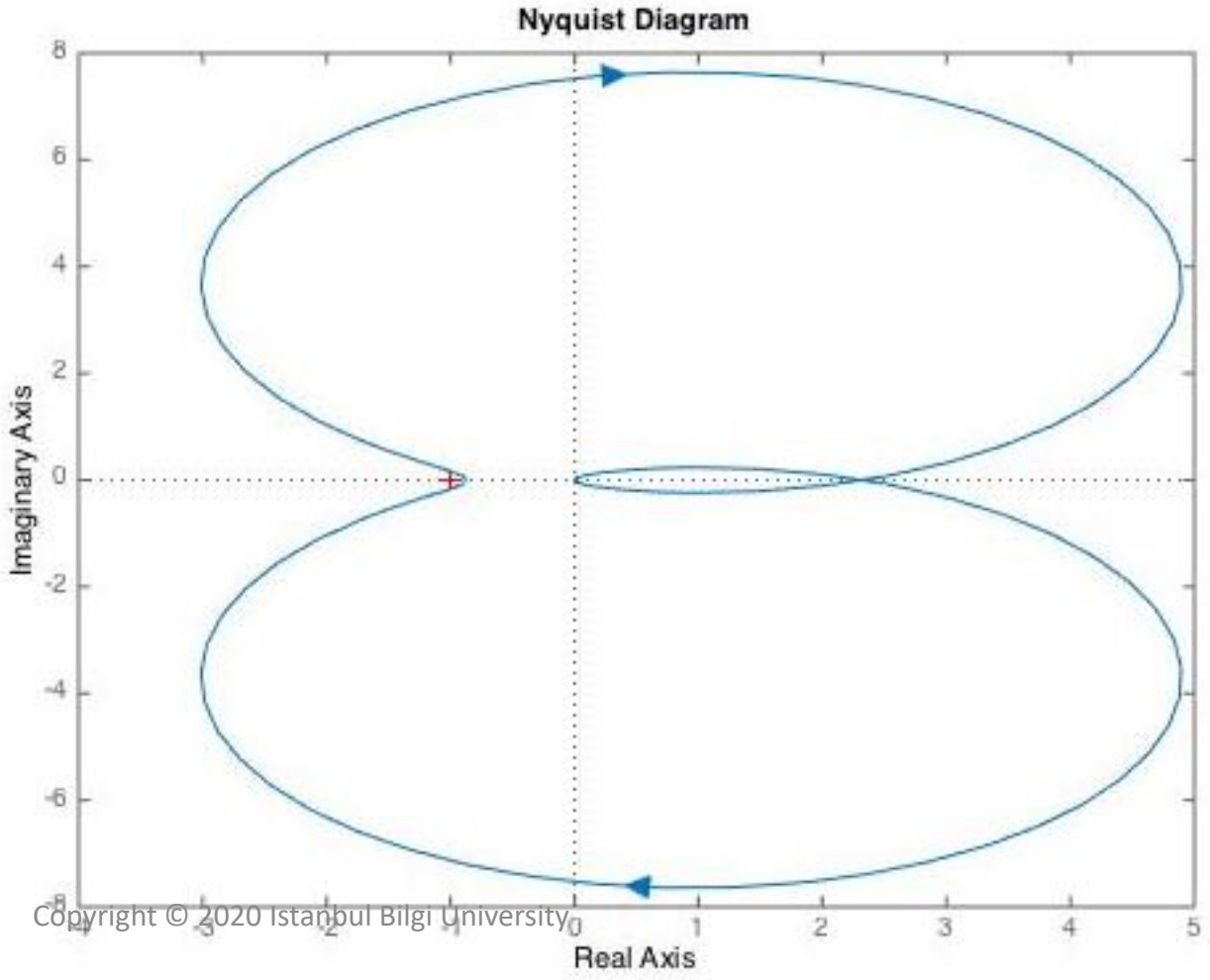
Plotting the Nyquist Diagram

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -8 & -2 & -7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ -14 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

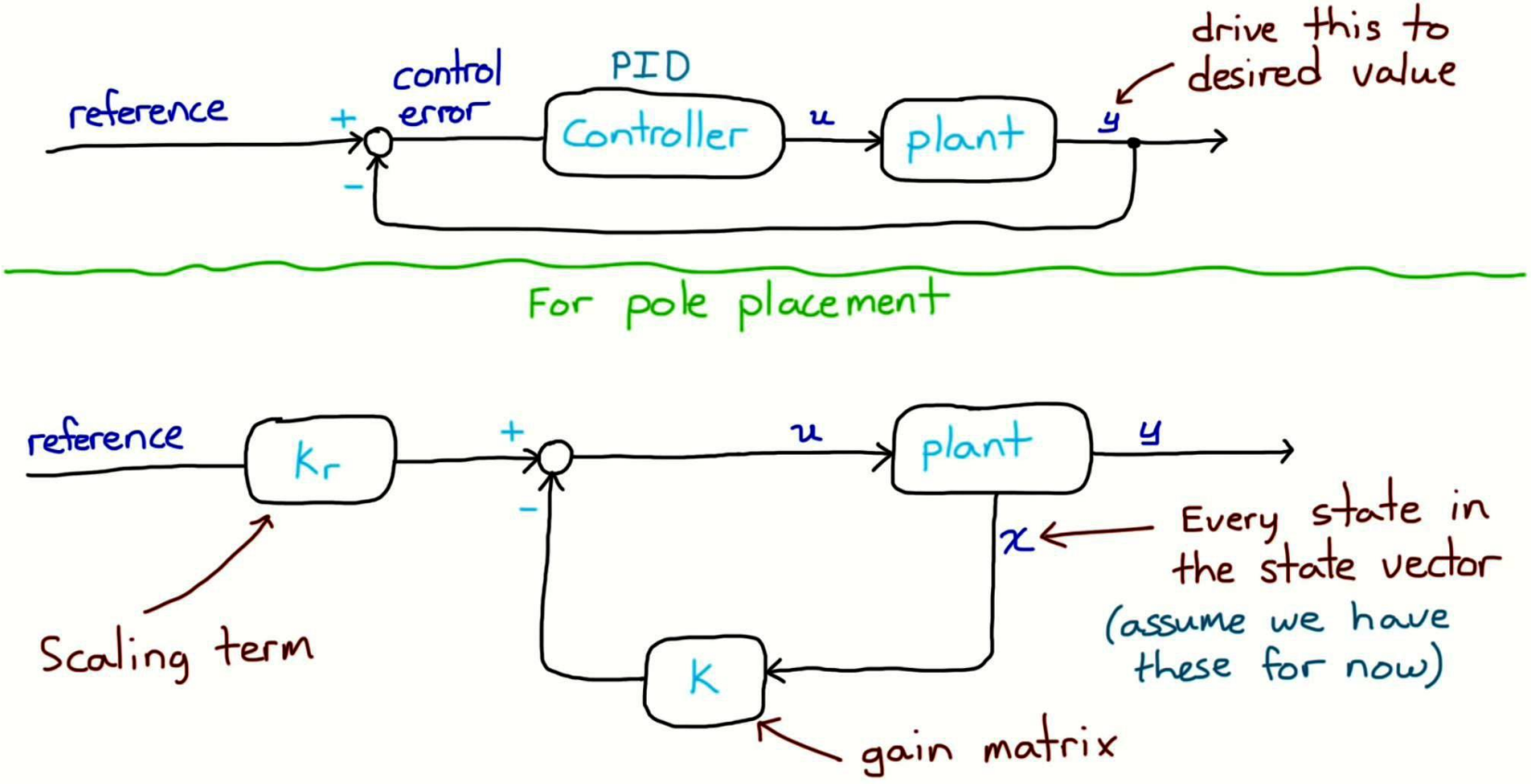
$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

nyquist (A, B, C, D)

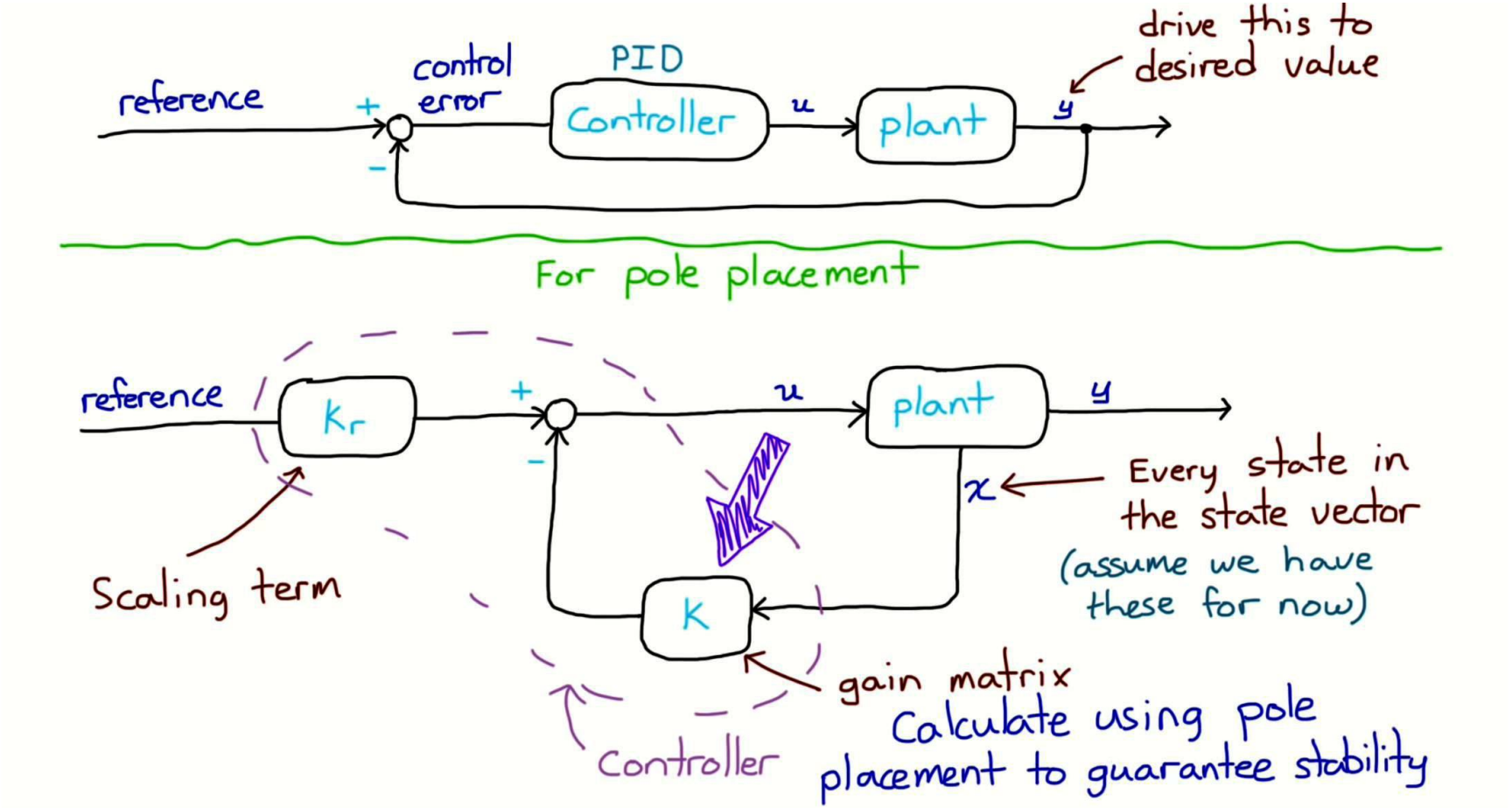
```
%Finding the transfer function
%System is represented in State-Space
%**** Enter matrices A,B,C and D
A=[0 1 0;0 0 1;-8 -2 -7];
B=[0; 1;-14];
C=[1 0 0];
D=[0];
nyquist(A,B,C,D)
```



Design by Pole Placement



Design by Pole Placement



Design by Pole Placement

Dynamics captured here

How the system responds to inputs

$$\dot{x} = Ax + Bu$$

Special? Yes!

A controller has to modify the A matrix to change dynamics

eigenvalues (A) = poles of the system

location dictates stability

Moving the poles $\Rightarrow$ choose system stability

Design by Pole Placement

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 2 & -1 \end{bmatrix} x \quad \leftarrow A$$

can be written as

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = 2x_1 - x_2$$

transform using

```
>> A=[0 1;2 -1];
>> eig(A)

ans =

     1
    -2

>> [X,D]=eig(A)

X =

    0.7071   -0.4472
    0.7071    0.8944


D =

     1     0
     0    -2
```

to

$$\dot{z} = \tilde{A} z$$

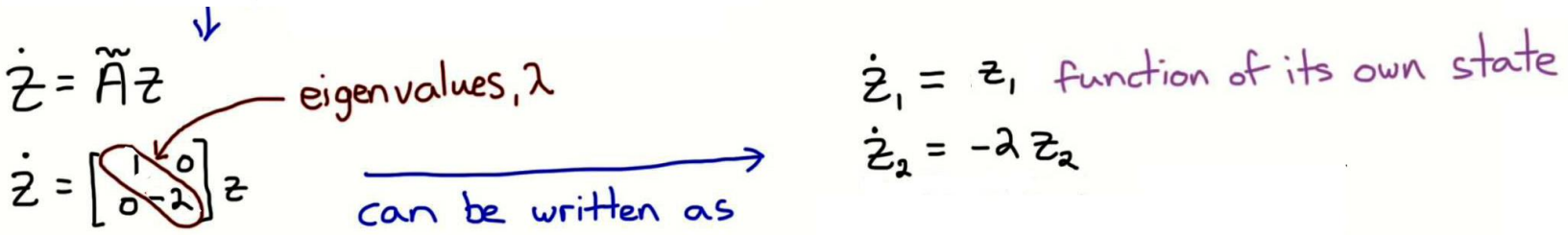
eigenvalues, λ

$$\dot{z} = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix} z$$

can be written as

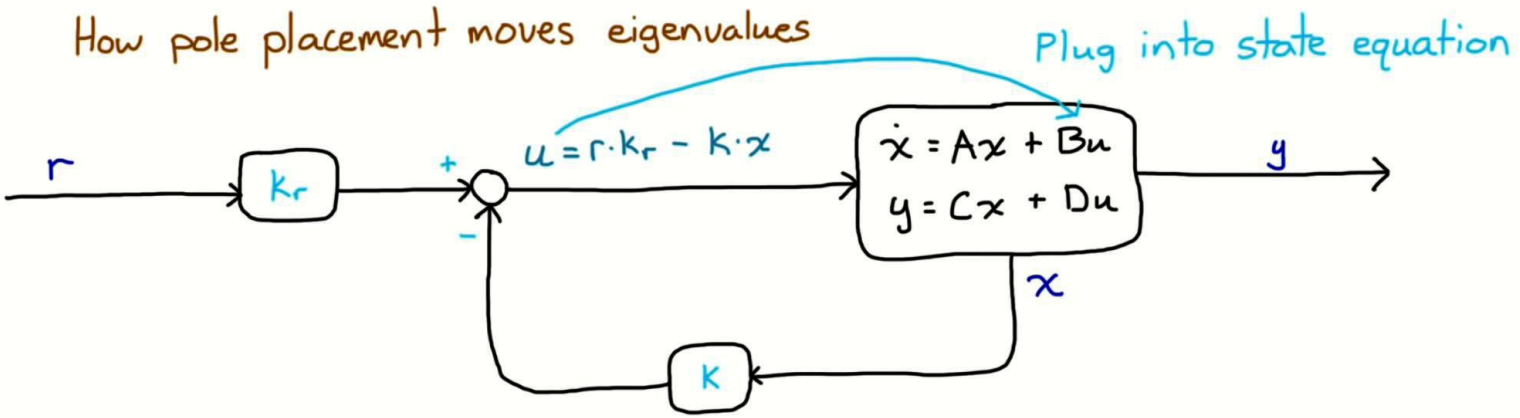
$$\dot{z}_1 = z_1$$ function of its own state
$$\dot{z}_2 = -2 z_2$$

Design by Pole Placement



Also,

Design by Pole Placement



$$\dot{x} = Ax + B(r \cdot k_r - k \cdot x)$$

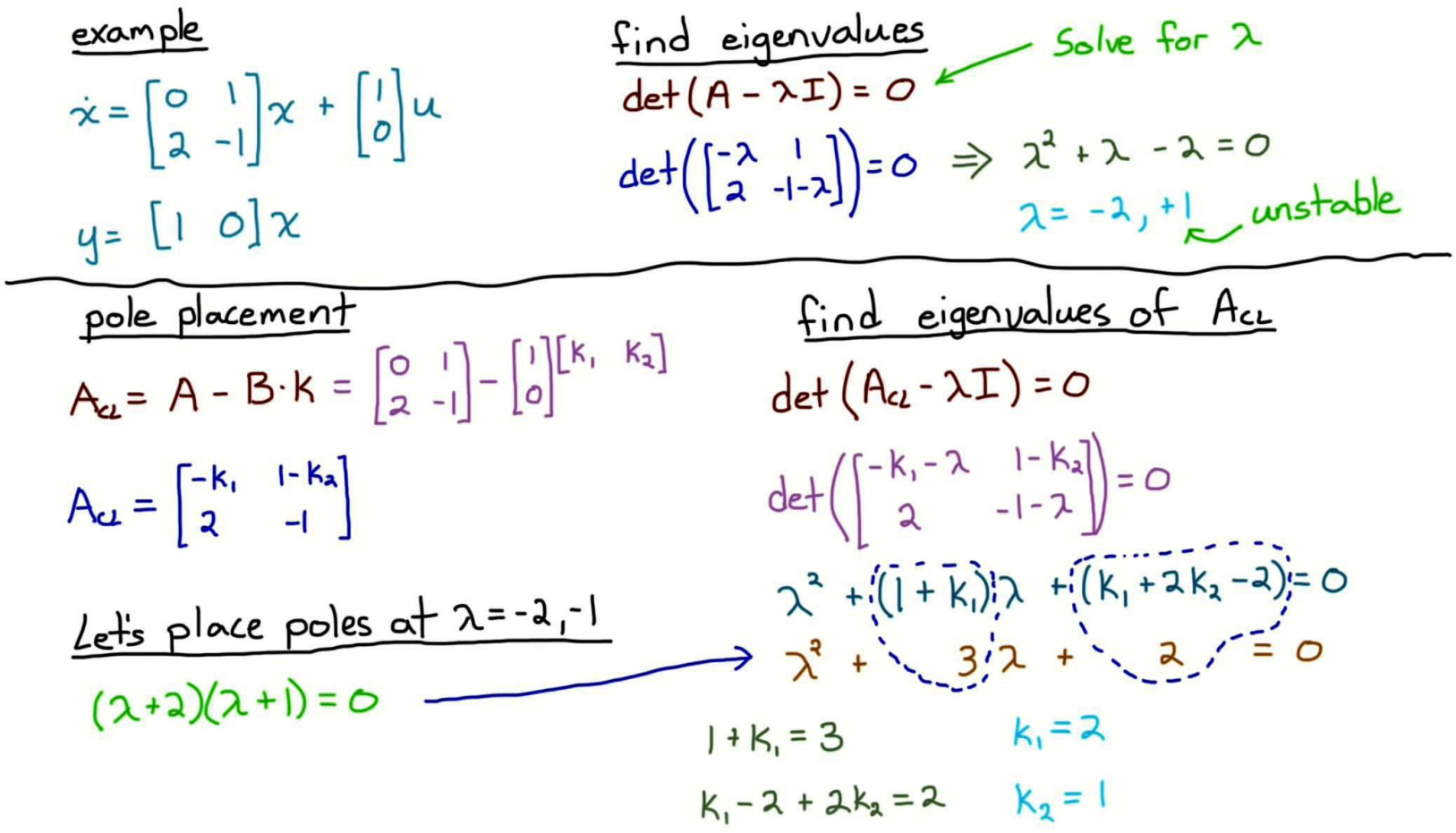
$$\dot{x} = Ax - Bkx + Br k_r$$

Combine

$$\dot{x} = (A - Bk)x + Br k_r$$

Closed loop A matrix

Design by Pole Placement



Summary

Designing a Control System by Pole Placement

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

We shall choose the control as

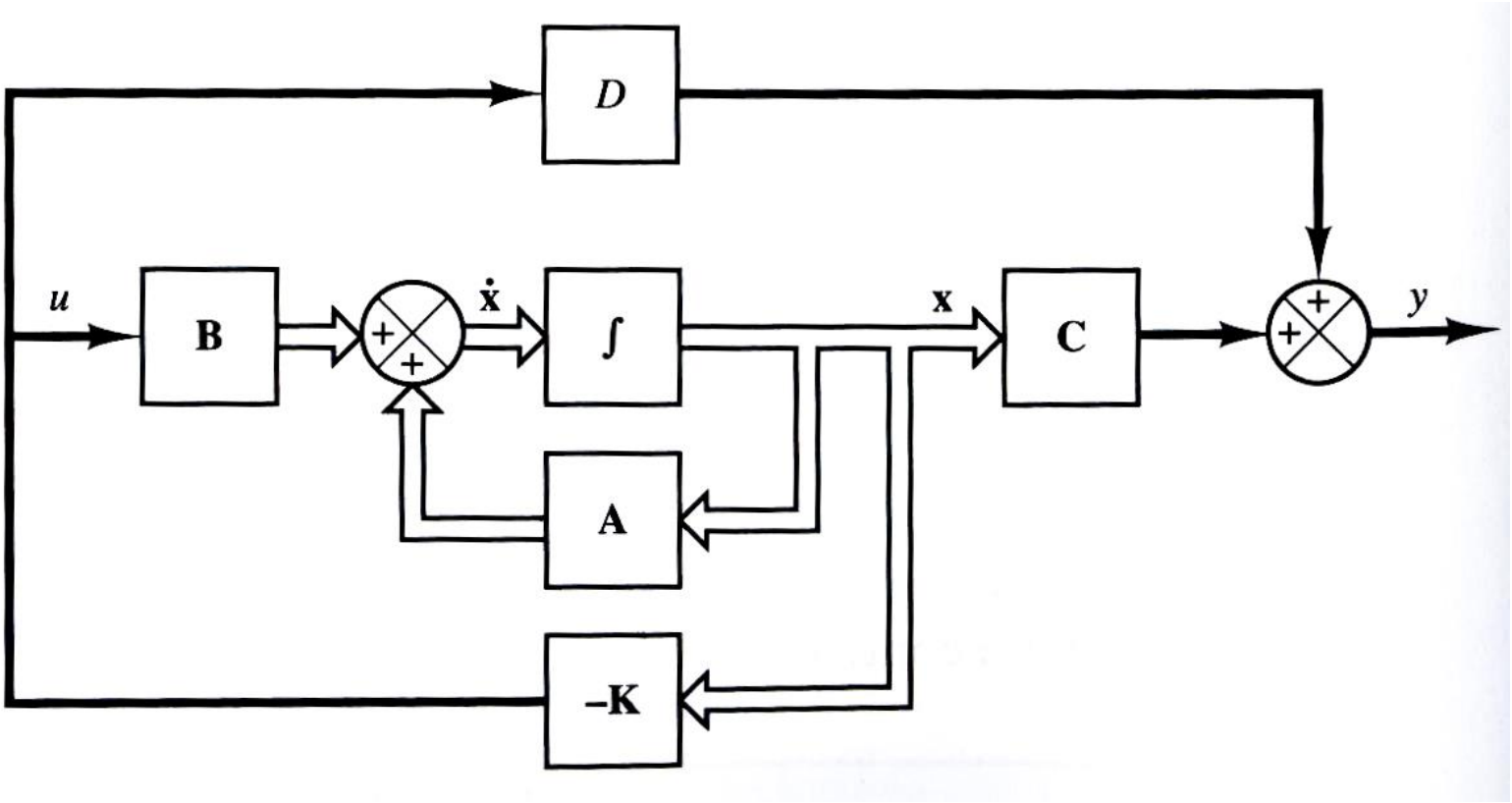
$$u = -Kx$$

$$\dot{x} = (A - BK)x$$

How should we choose K so that the system will have desired closed loop poles ?

This can be solved by using MATLAB

Design by Pole Placement



Design by Pole Placement

Example:

Consider the system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -5 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

$$u = -\begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

determine k_1, k_2, k_3

to have the closed loop poles at

$$P_1 = -2+j4, \quad P_2 = -2-j4, \quad P_3 = -10$$

```
J = [ P1  P2  P3 ]
K = place ( A, B, J )
```

Design by Pole Placement

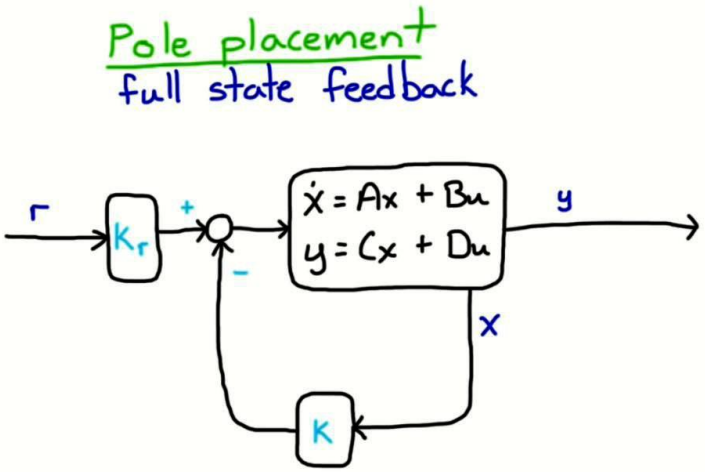
```
%Design by Pole_Placement
%Enter A, B and J
A=[0 1 0;0 0 1;-1 -5 -6];
B=[0; 0; 1];
J=[-2+j*4 -2-j*4 -10];
%Compute K
K=place(A,B,J)
```

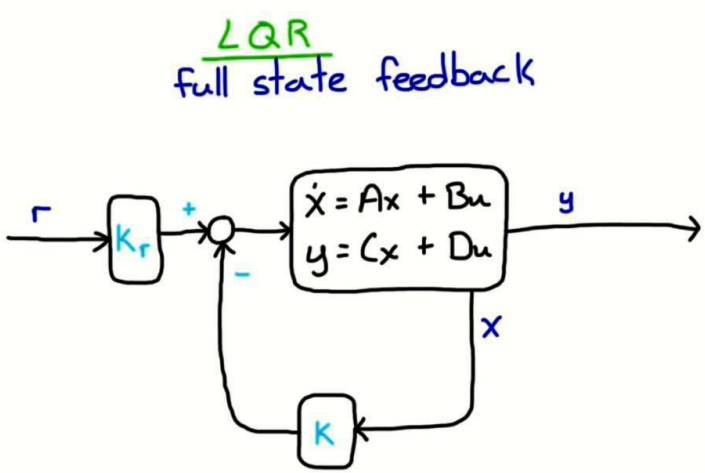
```
K =

  199.0000   55.0000    8.0000
```

Design by Pole Placement

Design by LQR

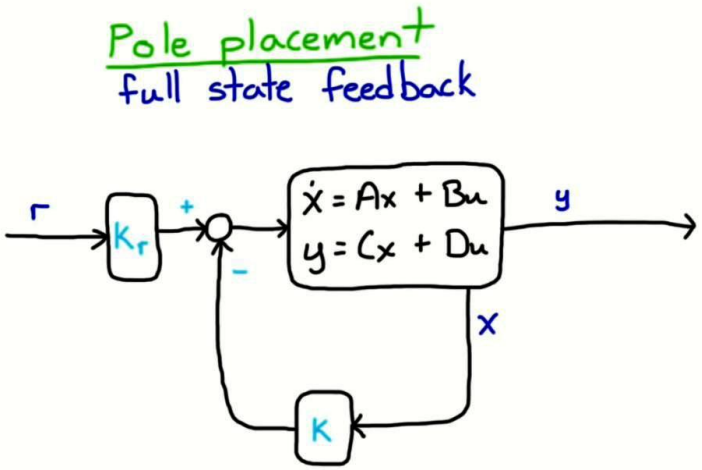


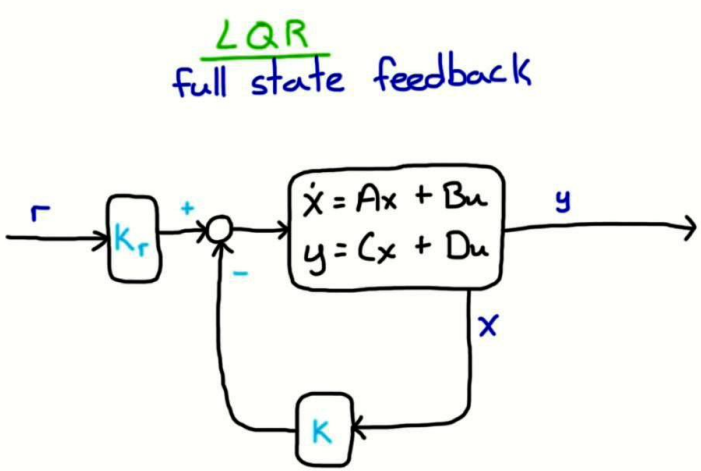


What is the Difference ?

Design by Pole Placement

Design by LQR



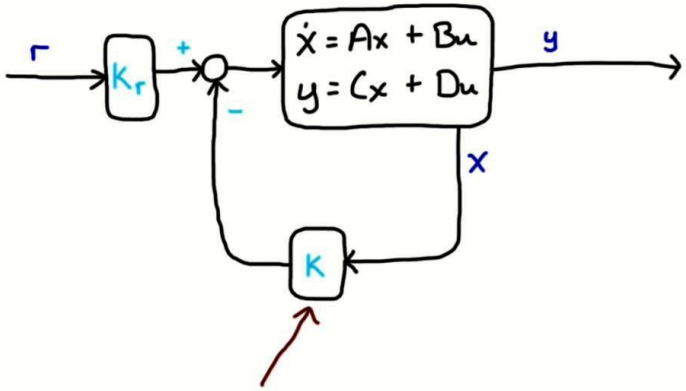


The implementation of K is the same

But how we choose K is different

Design by Pole Placement

Pole placement
full state feedback

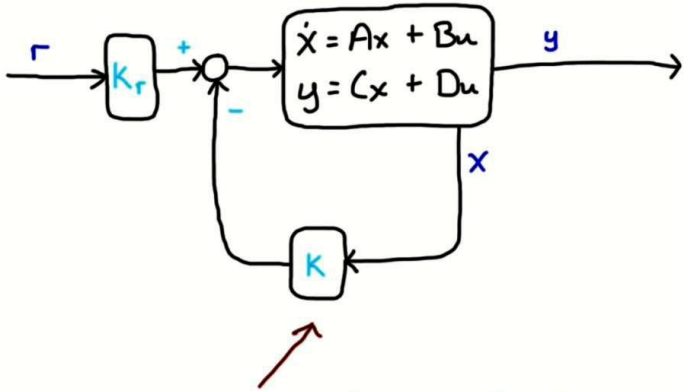


Solve for K by choosing pole locations

not terribly intuitive

Design by LQR

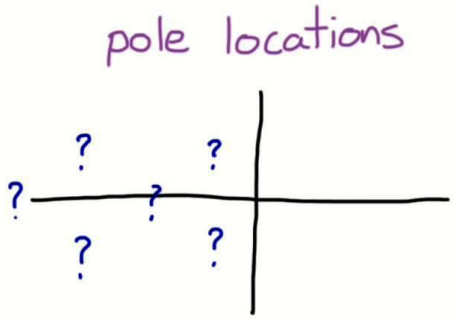
LQR
full state feedback



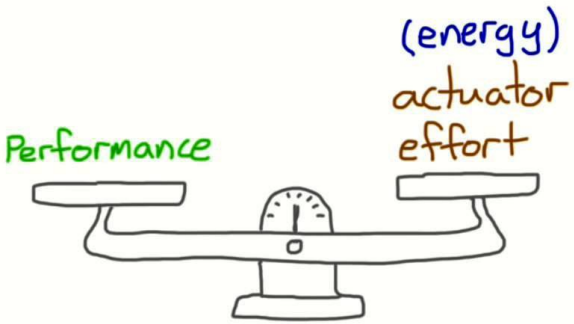
Find the optimal K by choosing characteristics

performance & effort

Design by Pole Placement

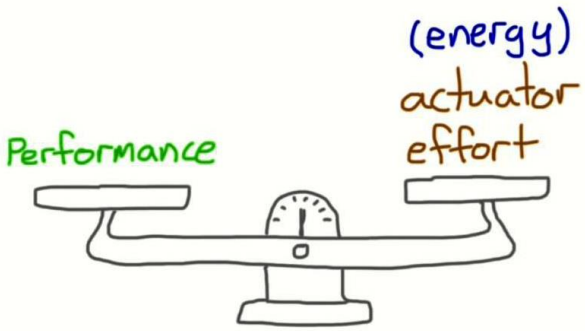


Design by LQR



This is how LQR approaches finding the optimal gain

Design by LQR

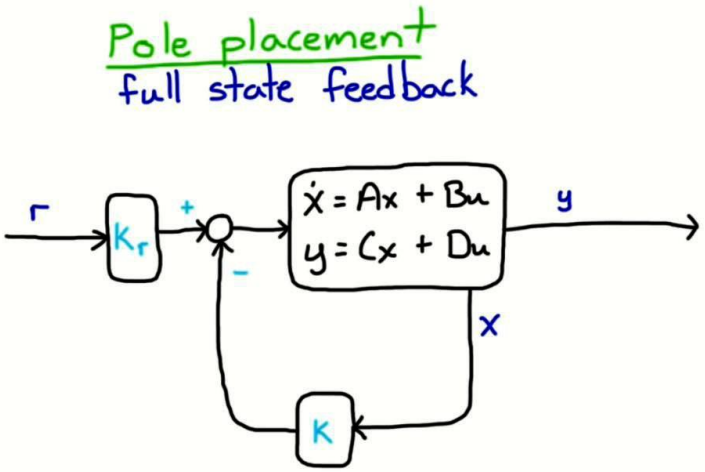


This is how LQR approaches finding the optimal gain

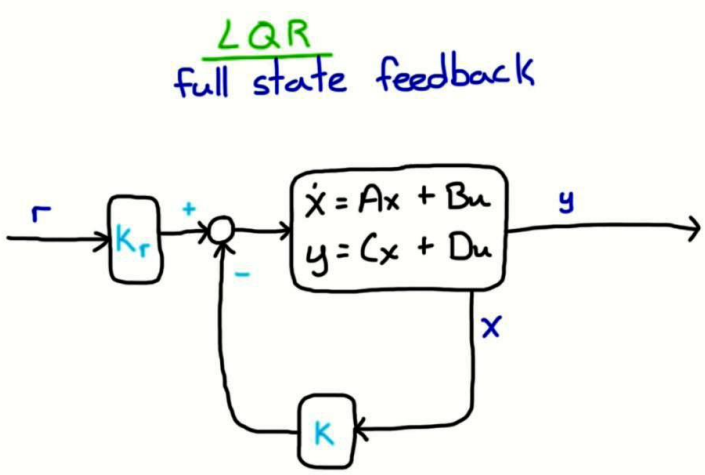
$$J = \int_0^\infty (x^T Q x + u^T R u)\, dt \Rightarrow u = -Kx \Rightarrow \begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

performance, effort; find optimal K; when subject to these dynamics

Design by Pole Placement



Design by LQR

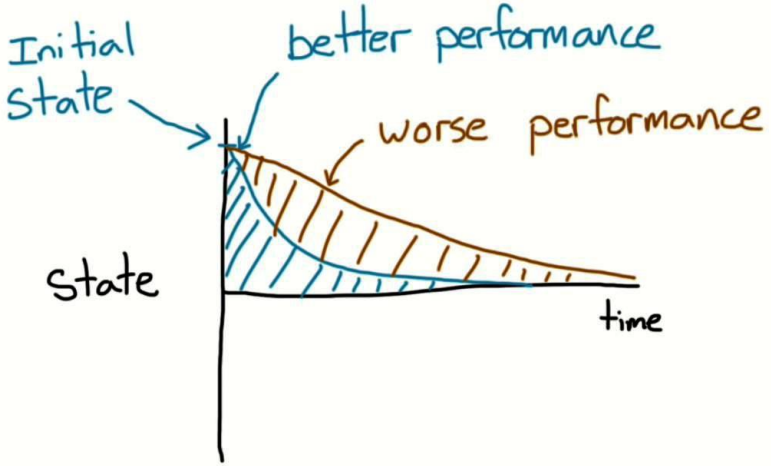


Design by LQR

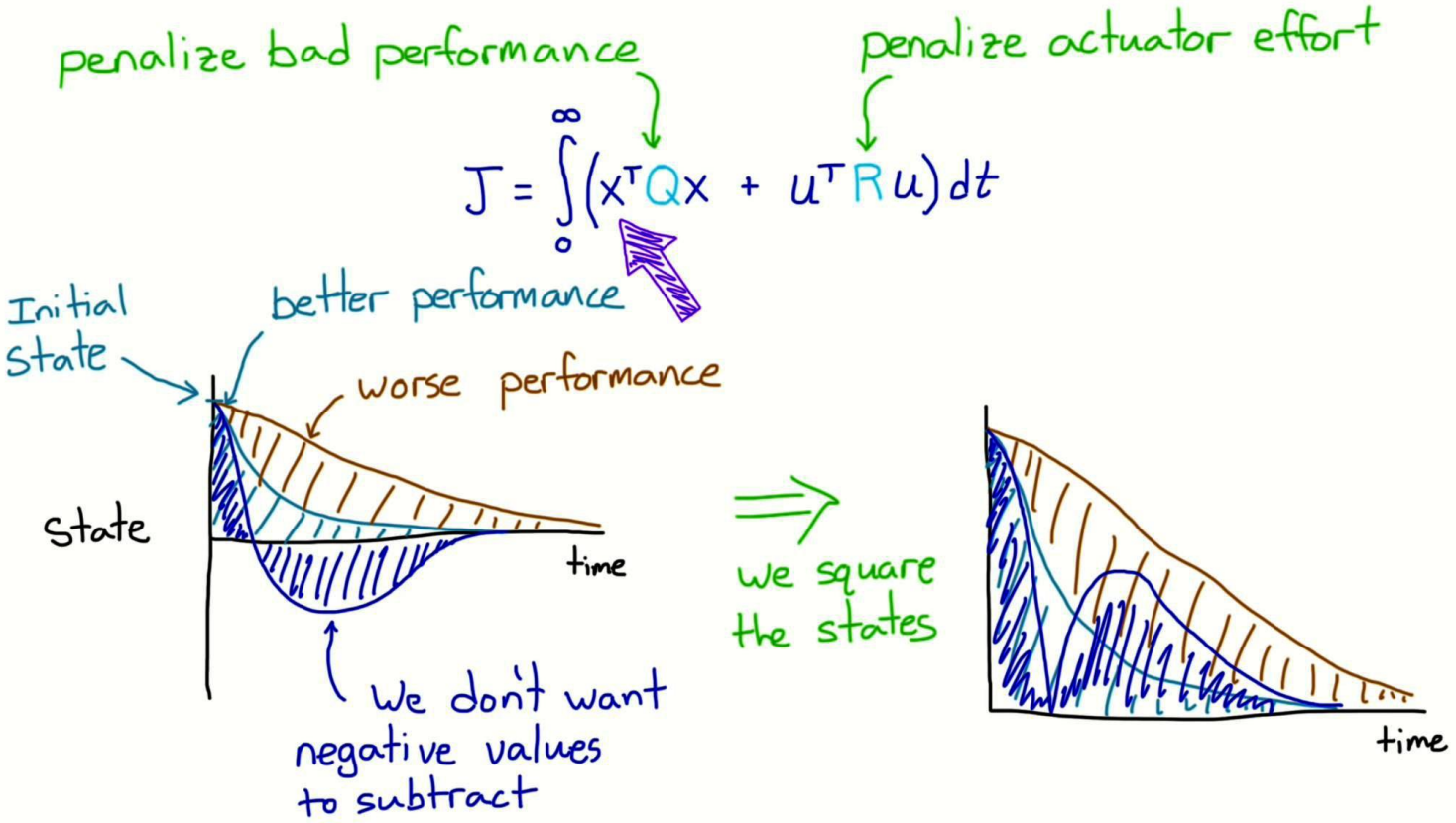
penalize bad performance

penalize actuator effort

$$J = \int_0^\infty (x^T Q x + u^T R u)\, dt$$



Design by LQR



Design by LQR

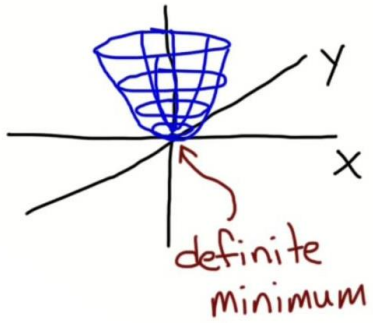
penalize bad performance

penalize actuator effort

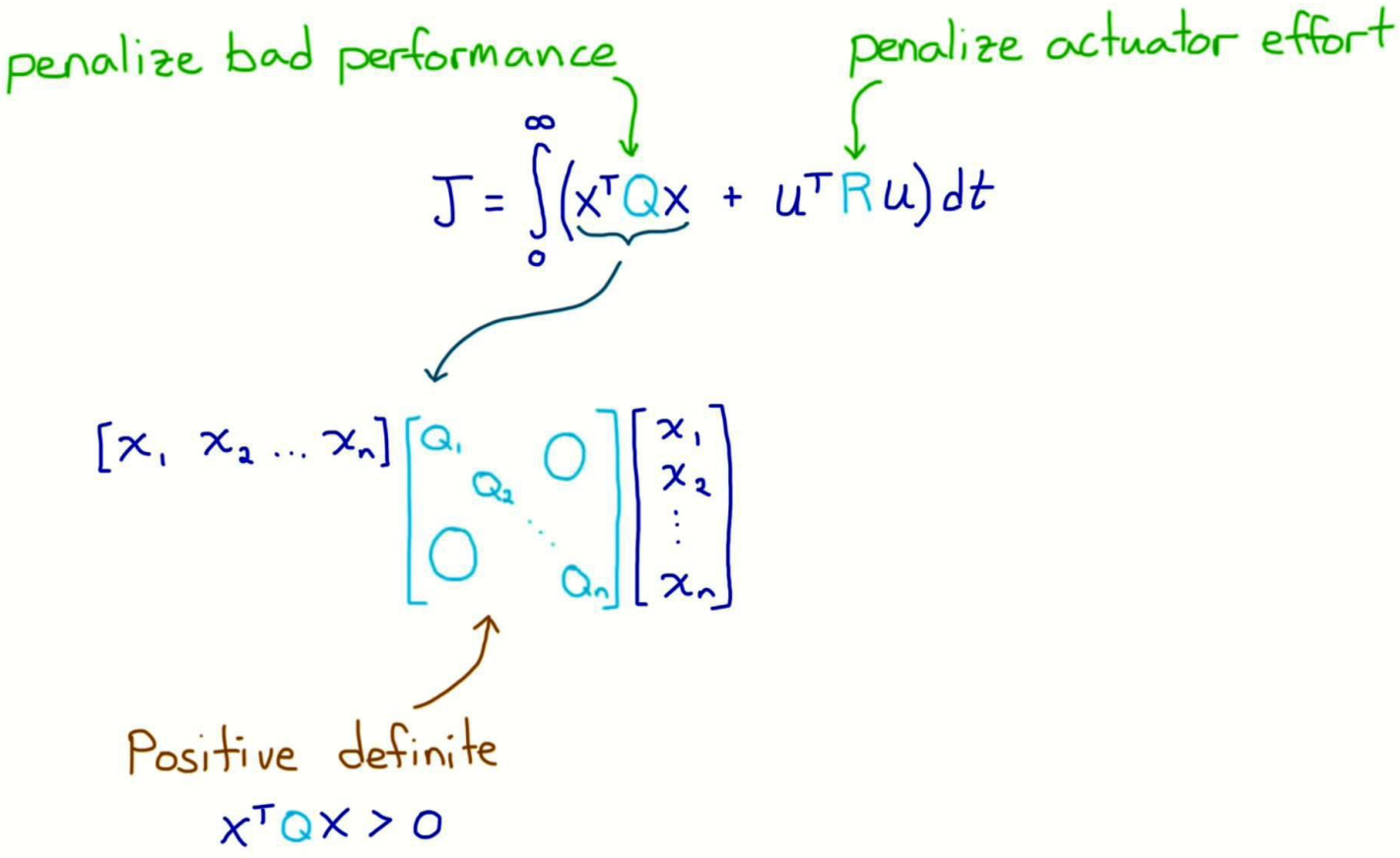
$$J = \int_0^\infty (x^T Q x + u^T R u)\, dt$$

Quadratic Functions

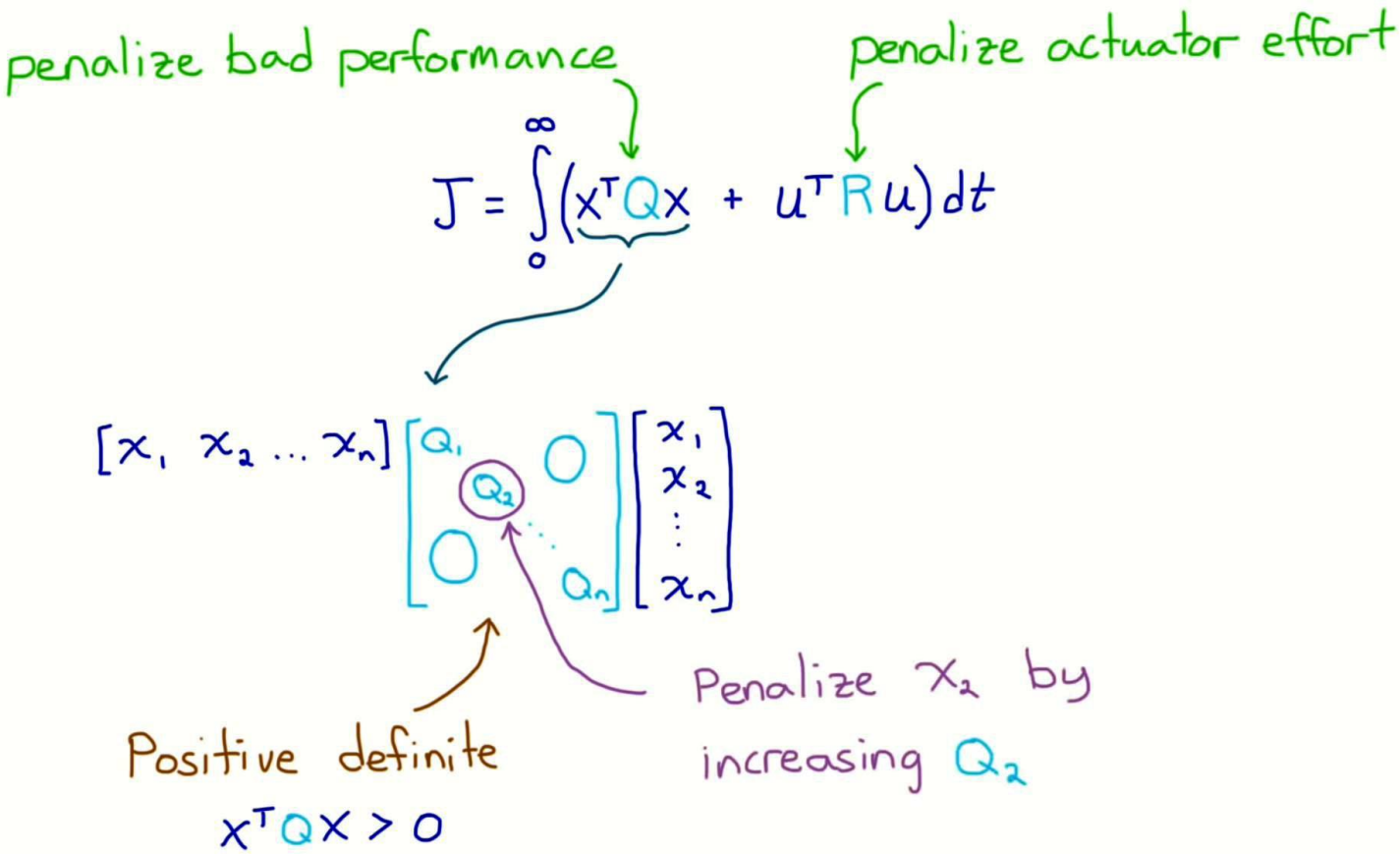
$$z = x^2 + y^2$$



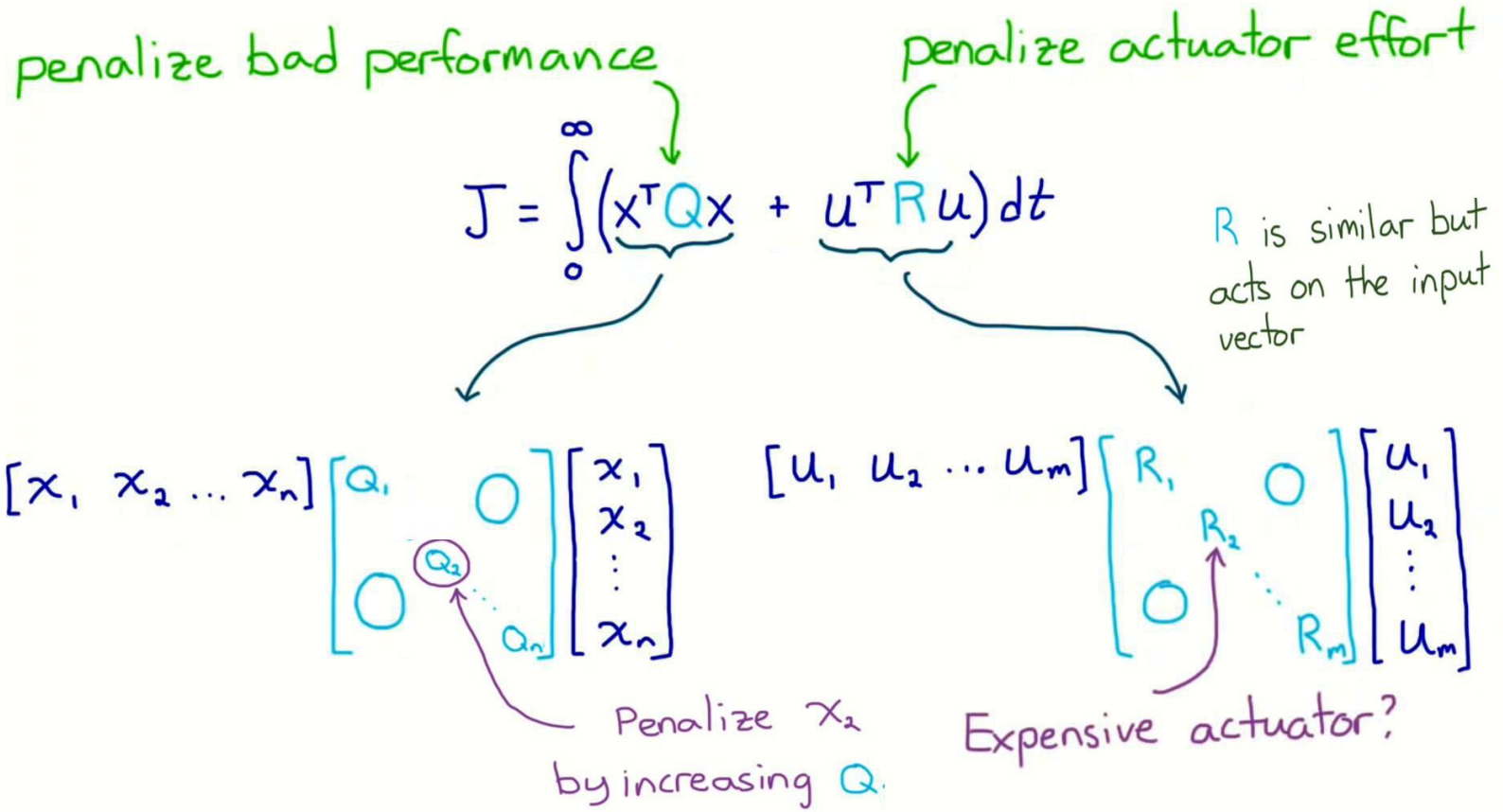
Design by LQR



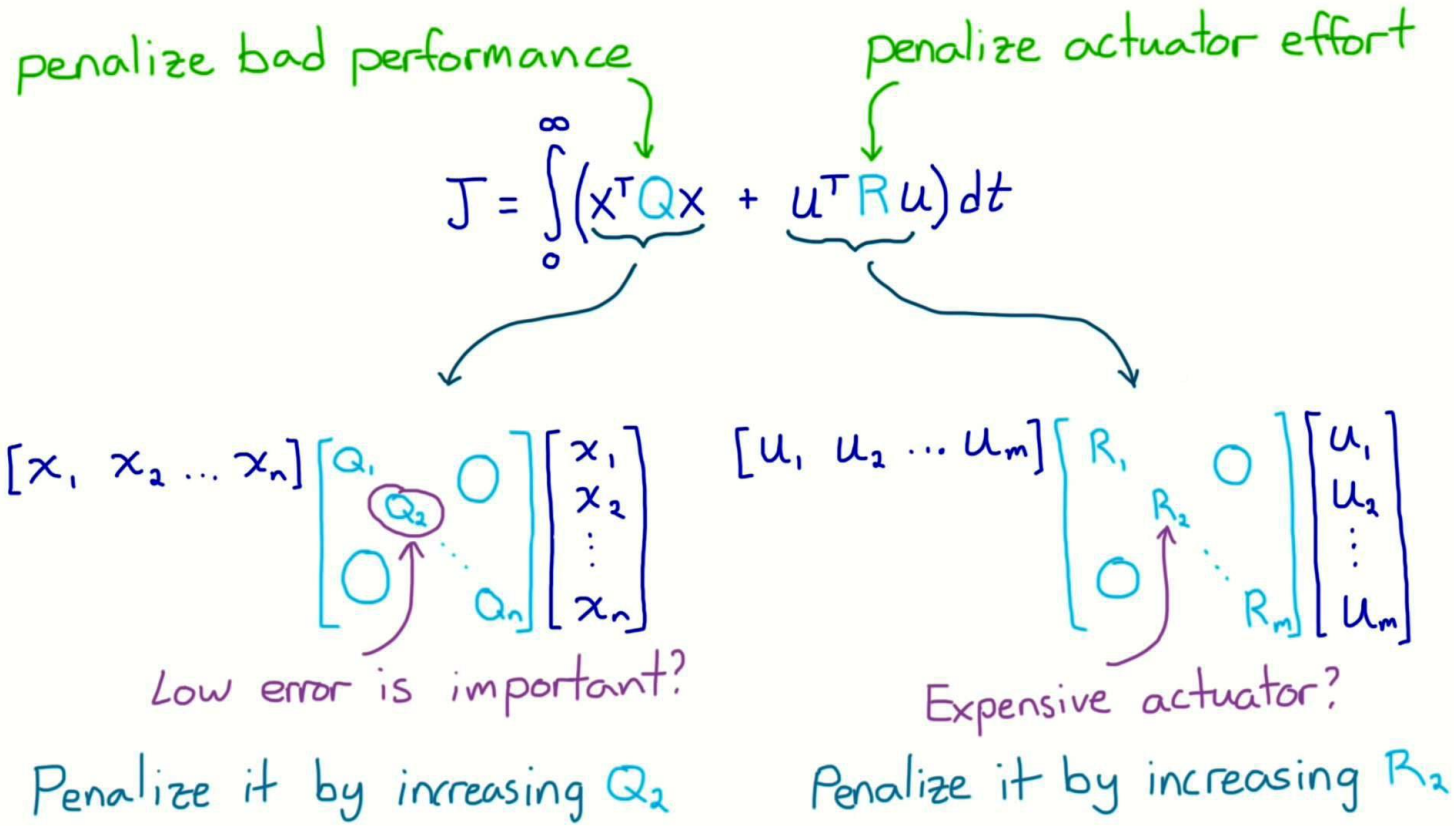
Design by LQR



Design by LQR



Design by LQR



Design by LQR

How do we solve this optimization problem?

$$J = \int_0^\infty (x^T Q x + u^T R u)\, dt \Rightarrow u = -Kx \Rightarrow \begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

performance: $x^T Q x$

effort: $u^T R u$

find optimal K

when subject to these dynamics

Design by LQR

Designing with LQR

- Develop a linear model

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

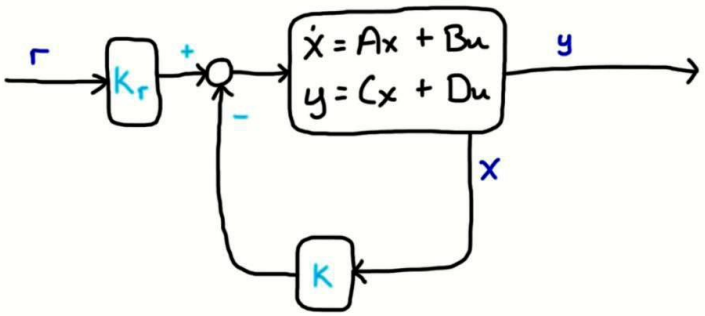
- Adjust Q and R

- Find the optimal gain set

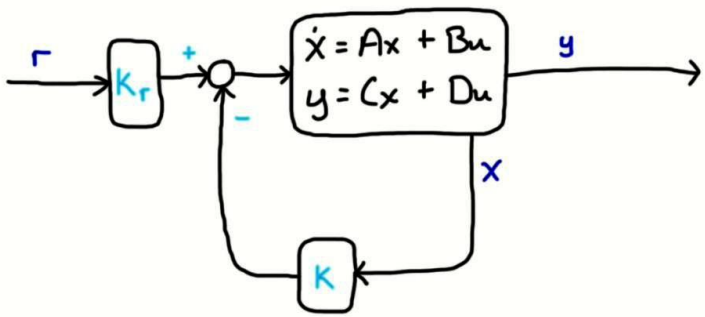
```
>> K = lqr(A,B,Q,R)
```

Design by LQR

Pole placement
full state feedback



LQR
full state feedback



Design of Optimal Control System

Optimal Control

Given the system equation

$$\dot{x} = Ax + Bu$$

Determine K

$$u(t) = -Kx(t)$$

so as to minimize the performance index

$$J = \int_0^\infty (x^T Q x + u^T R u)\, dt$$

This problem is solved by using the Matlab function

$$K = lqr(A, B, Q, R)$$

Design of Optimal Control System

Example

Consider the system

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -3 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad C = [1 \ 0 \ 0], \quad D = [0]$$

Determine the state-feedback gain matrix

$$K = [k_1 \quad k_2 \quad k_3]$$

Such that the following performance index is minimized

$$J = \int_0^\infty (x^T Q x + u^T R u)\, dt$$

where

$$Q = \begin{bmatrix} q_{11} & 0 & 0 \\ 0 & q_{22} & 0 \\ 0 & 0 & q_{33} \end{bmatrix}, \quad R = r_1$$

and

$$q_{11} = 100 \qquad q_{22} = q_{33} = 1 \qquad r_1 = 0.01$$

Design of Optimal Control System

```
%Design by Pole_Placement
%Enter A, B, Q and R
A=[0 1 0;0 0 1;0 -2 -3];
B=[0; 0; 1];
Q=[100 0 0;0 1 0;0 0 1];
R=[0.01];
%Compute K
K=lqr(A,B,Q,R)
```

```
K =

  100.0000   53.1200   11.6711
```

End of Lecture 6