



# Autonomous Mobile Robot Design

## The very basics of control theory

Dr. Kostas Alexis (CSE)

# 1. Modeling of Dynamic Systems

- ▶ Generic form of dynamic system:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$$

- ▶  $\mathbf{x}$  is the state vector,  $\mathbf{u}$  is the input vector.
- ▶ Generic linear state space form:

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

- ▶ And given specific output vector:

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$$

# State Space Representation

- ▶ Generic linear (time invariant) state space form:

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

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$$

- ▶ **A** is the system matrix ( $n \times n$ ), **B** is the input matrix ( $n \times p$ ), **C** is the output matrix ( $q \times n$ ), **D** is the feedforward matrix ( $q \times p$ ).
- ▶ In general a MIMO system (Multi-Input, Multi-Output)
- ▶ In fact a Linear Time Invariant system (LTI). More complex representations exist.

# Transfer Function Representation

- ▶ Linear time invariant can be represented in the frequency domain using the Laplace Transform.
- ▶ Laplace Transform:

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^{\infty} e^{-st} f(t) dt$$

- ▶ Transfer function form to represent Single Input Single Output (SISO) relationships:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}$$

- ▶ And equivalently:

$$G(s) = K \frac{(s - z_1)(s - z_2) \dots (s - z_{m-1})(s - z_m)}{(s - p_1)(s - p_2) \dots (s - p_{n-1})(s - p_n)}$$

- ▶  $p_1 \dots p_n$  correspond to the poles and  $z_1 \dots z_m$  to the zeros.

# State Space to Transfer Function

- ▶ Transformation from state space to transfer function:

$$G(s) = \frac{Y(s)}{U(s)} = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$$

- ▶ Possibly leads to multiple transfer functions

# Dynamic system example

- System ordinary differential equation:

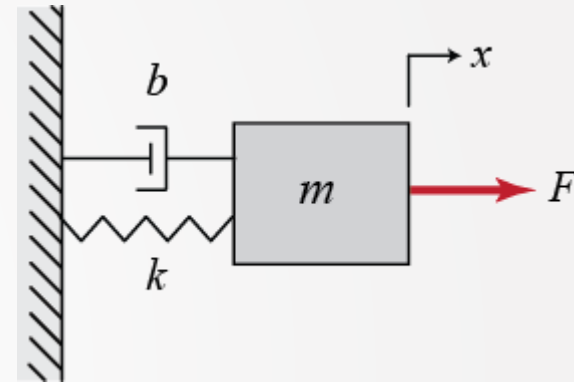
$$m\ddot{x} = \sum F$$
$$F - b\dot{x} - kx = m\ddot{x}$$

- Or in state space form:

$$\mathbf{x} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, \quad \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} F$$

- Or in transfer function form:

$$F(s) - bsX(s) - kX(s) = ms^2X(s) \Rightarrow$$
$$\frac{X(s)}{F(s)} = \frac{1}{ms^2 + bs + k}$$





## 2. Analysis of Dynamic Systems

- ▶ Time-Domain Response of an LTI system
- ▶ Frequency-Domain Response of an LTI system
- ▶ Stability of an LTI system



# Time-Domain Response

- ▶ The time response of a linear dynamic system consists of the **sum of the transient response** which depends on the initial conditions **and the steady-state response** which depends on the system input.
- ▶ These correspond to the free (homogeneous or zero input) and the forced (inhomogeneous or non-zero input) solutions of the governing differential equations respectively.

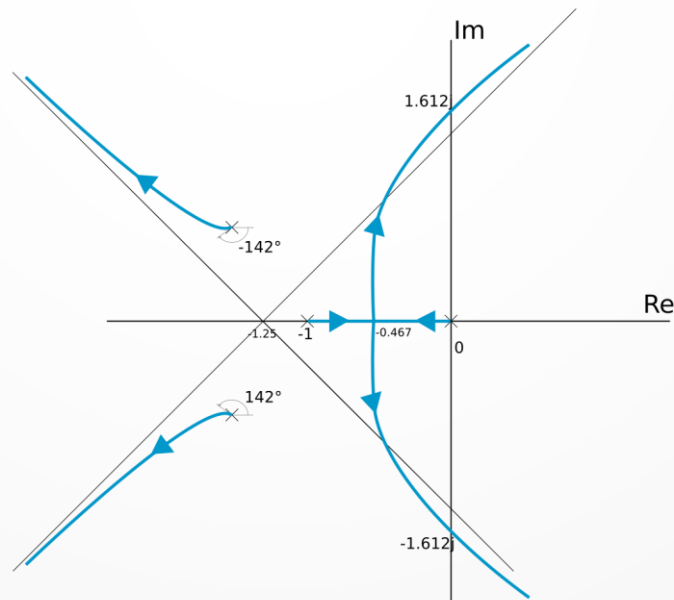


# Frequency-Domain Response

- ▶ All the examples presented in this tutorial are modeled by linear constant coefficient differential equations and are thus linear time-invariant (LTI). LTI systems have the extremely important property that if the input to the system is sinusoidal, then the steady-state output will also be sinusoidal at the same frequency but in general with different magnitude and phase. These magnitude and phase differences as a function of frequency comprise the **frequency response** of the system.
- ▶ The frequency response of a system can be found from the transfer function in the following way: create a vector of frequencies (varying between zero or "DC" to infinity) and compute the value of the plant transfer function at those frequencies. If  $G(s)$  is the open-loop transfer function of a system and  $\omega$  is the frequency vector, we then plot  $G(j\omega)$  versus  $\omega$ . Since  $G(j\omega)$  is a complex number, we can plot both its magnitude and phase (the Bode Plot) or its position in the complex plane (the Nyquist Diagram). Both methods display the same information in different ways.

# Stability

- ▶ Bounded Input Bounded Output (BIBO) sense: a system is stable if the output remains bounded for all bounded (finite) inputs. Practically, this means that the system will not “blow up” while in operation.
- ▶ Using transfer functions: the system is stable if ALL of the poles are in the left complex plane. If any of them is on the right then the system is unstable. If any of them has zero real component, the system is critically stable.

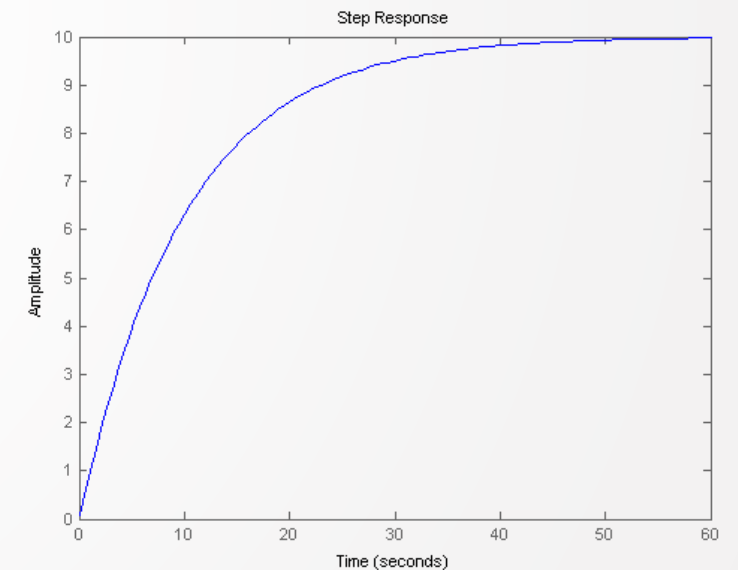


# 1<sup>st</sup> Order Systems

- ▶ First order system representation:

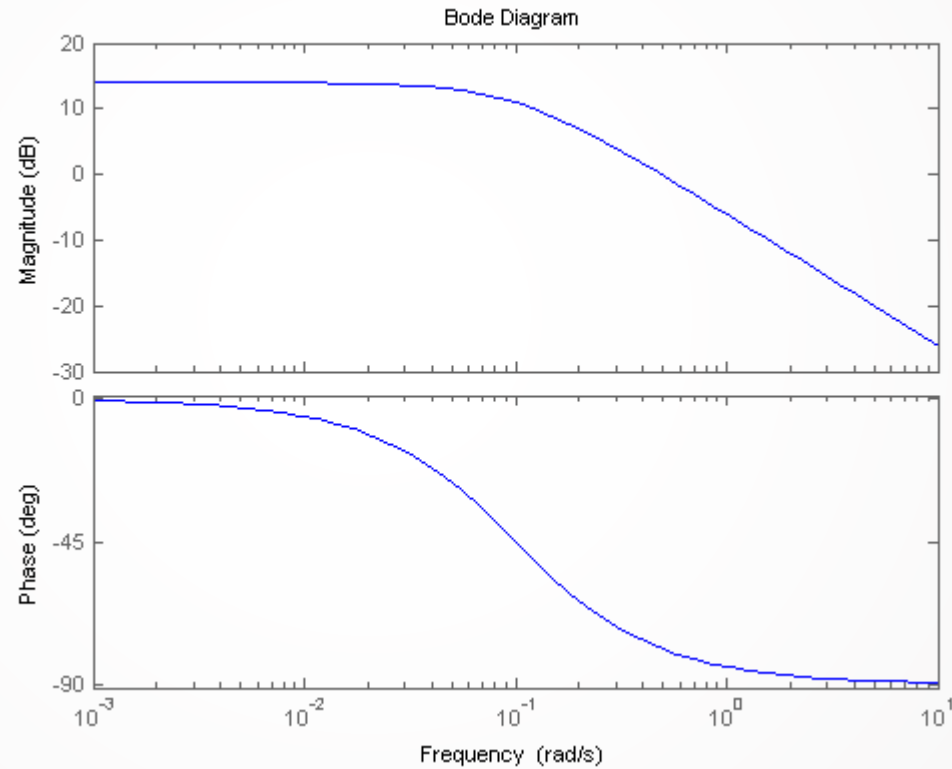
$$\dot{y} + ay = bu \Rightarrow \tau\dot{y} + y = k_{dc}u$$
$$\Rightarrow G(s) = \frac{b}{s + a} = \frac{k_{dc}}{\tau s + 1}$$

- ▶ With pole  $-a$ , time constant  $\tau$  and dcgain  $k_{dc}$
- ▶ Time constant is the value that the system reaches 63% of its steady-state value.



# 1<sup>st</sup> Order Systems

➤ Bode Response:



# 2<sup>nd</sup> Order Systems

- ▶ First order system representation:

$$m\ddot{y} + b\dot{y} + ky = f(t) \quad \Rightarrow \quad \ddot{y} + 2\zeta\omega_n\dot{y} + \omega_n^2y = k_{dc}\omega_n^2u$$

- ▶ Or in transfer function form:

$$G(s) = \frac{1}{ms^2 + bs + k} = \frac{k_{dc}\omega_n^2}{s^2 + 2\zeta\omega_ns + \omega_n^2}$$

- ▶ dcgain:  $k_{dc} = \frac{1}{k}$

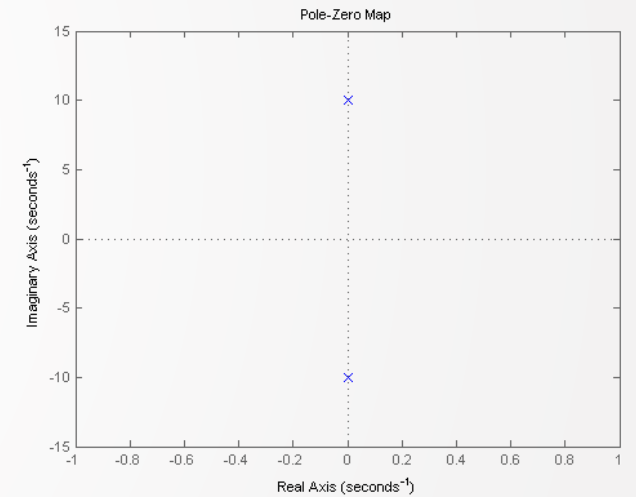
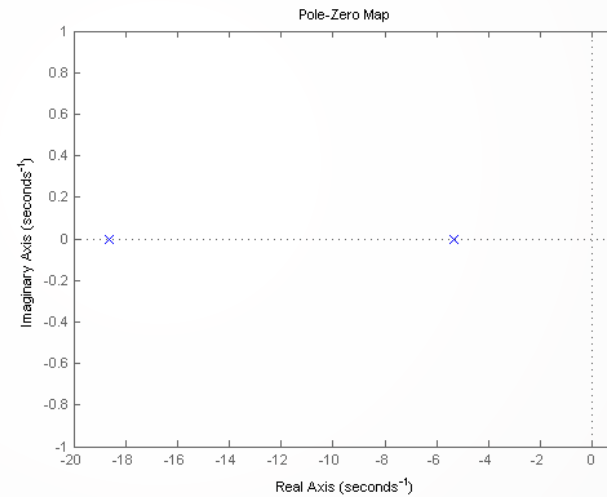
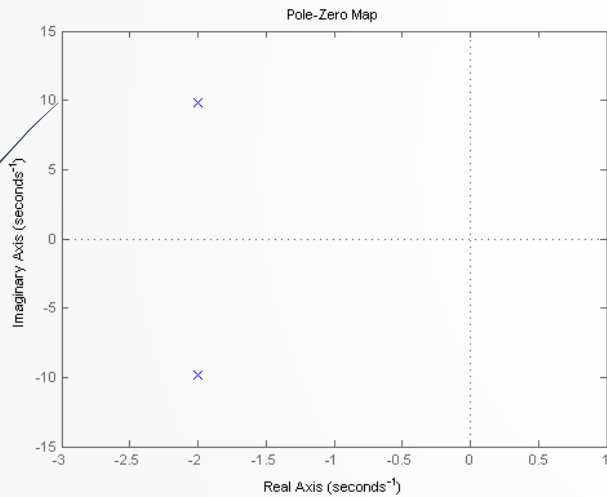
- ▶ damping ratio:  $\zeta = \frac{b}{2\sqrt{k/m}}$  (if  $\zeta < 1$  system is underdamped)

- ▶ natural frequency:  $\omega_n = \sqrt{\frac{k}{m}}$

- ▶ poles/zeros:  $s_p = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$

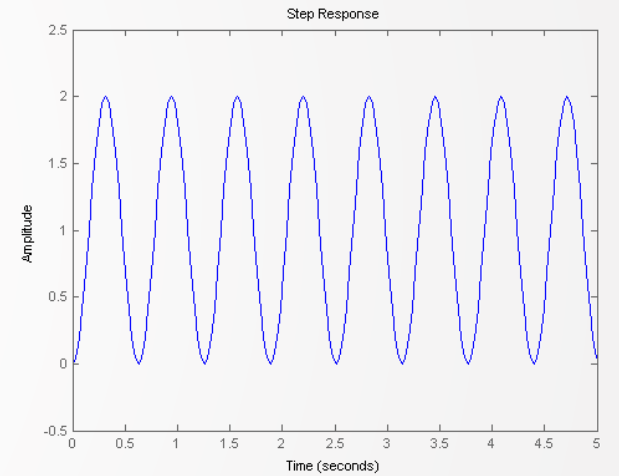
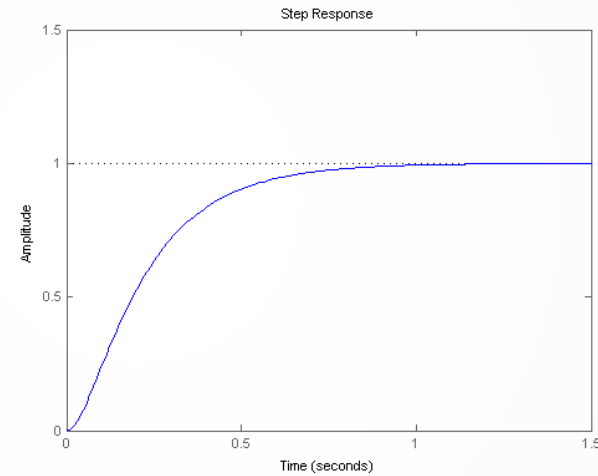
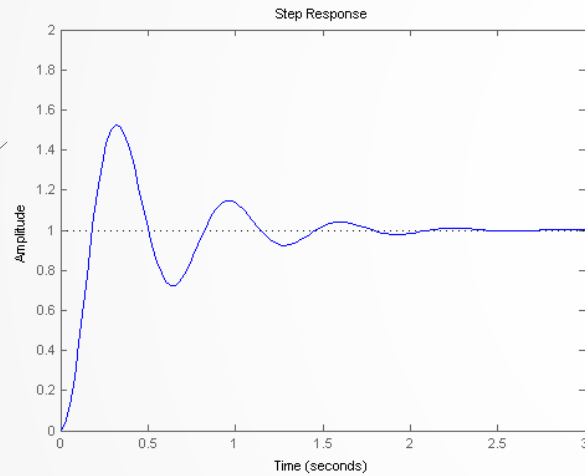
# 2<sup>nd</sup> Order Systems

## ➤ Pole map



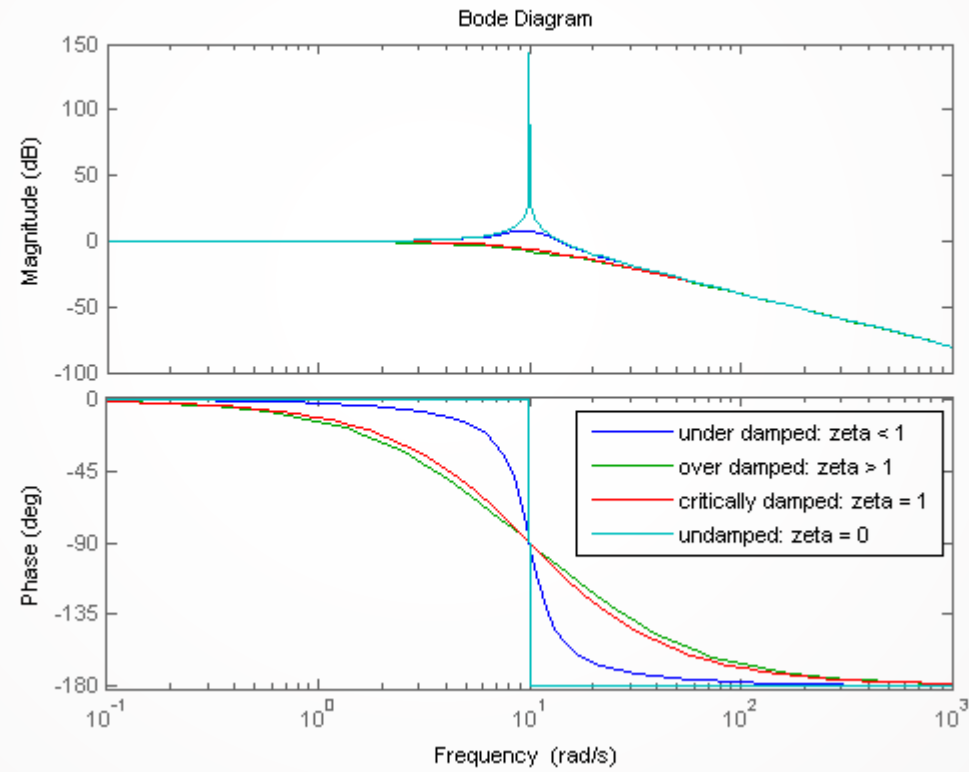
# 2<sup>nd</sup> Order Systems

## Time response



# 2<sup>nd</sup> Order Systems

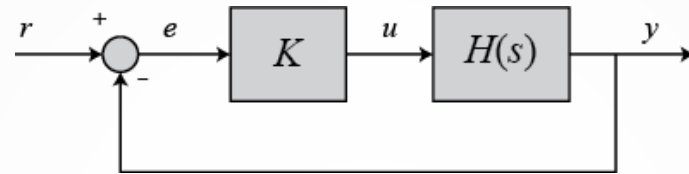
➤ Bode plot





# 3. The Root Locus Method

- ▶ Consider a transfer function and a gain  $K$ :



- ▶ The closed-loop transfer function takes the form:

$$\frac{Y(s)}{R(s)} = \frac{KH(s)}{1 + KH(s)}$$

- ▶ Thus the closed-loop poles are the solutions of:

$$1 + KH(s) = 0$$

- ▶ And if we write:

$$H(s) = b(s)/a(s)$$

- ▶ Then the poles are the solutions of:

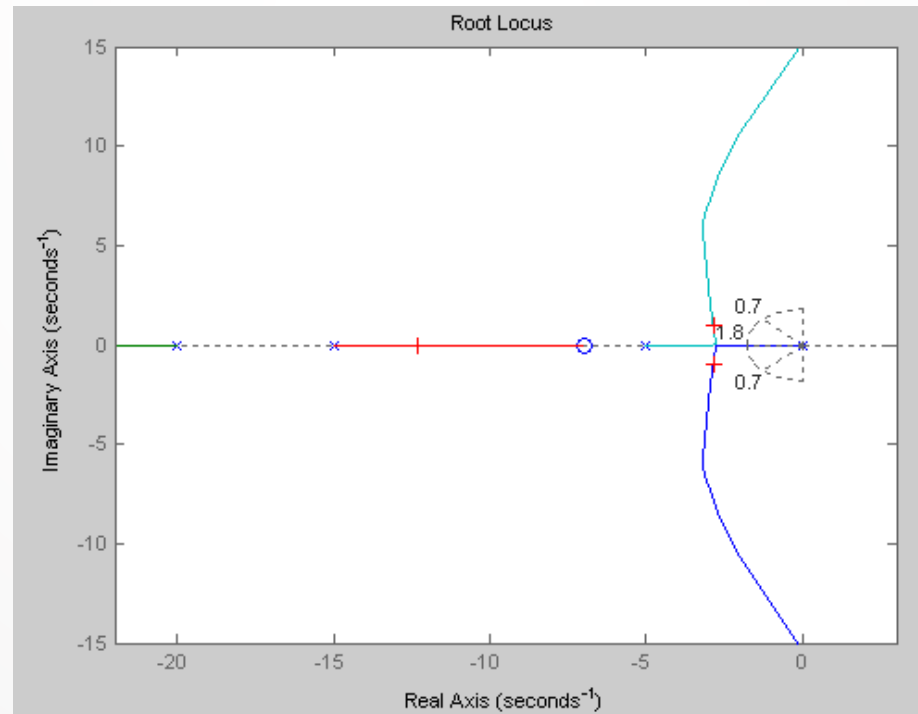
$$\frac{a(s)}{K} + b(s) = 0$$

# Root Locus Method example

- Consider the transfer function:

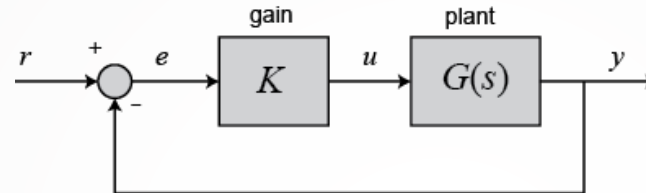
$$H(s) = \frac{Y(s)}{U(s)} = \frac{s + 7}{s(s + 5)(s + 15)(s + 20)}$$

- Root locus for a gain K:

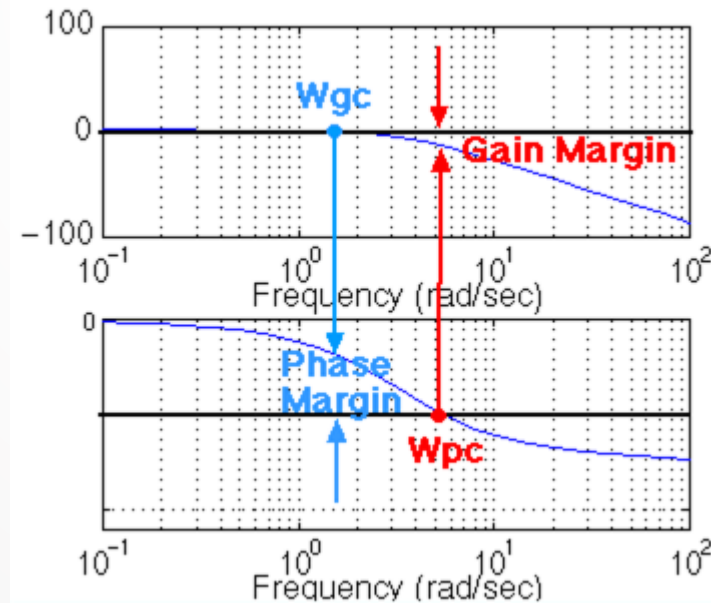


# 4. Frequency Domain Design

- Consider the following system:

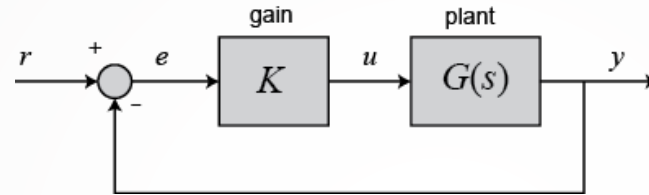


- We can use the bode diagram to identify how to tune the controller K based on the system Gain and Phase margins.



# 4. Frequency Domain Design

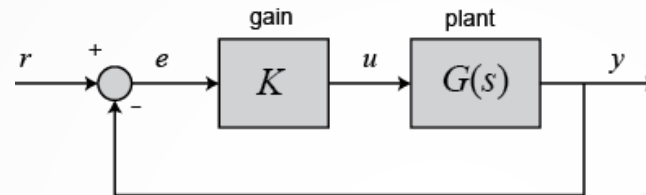
- ▶ Consider the following system:



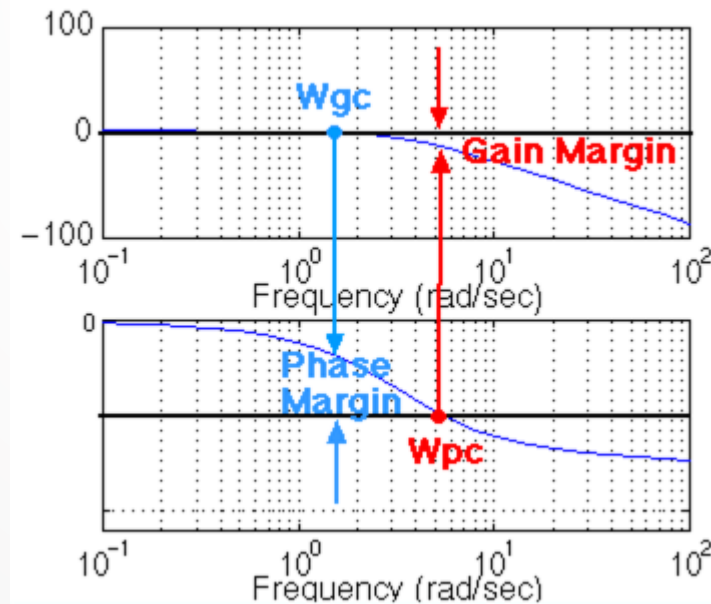
- ▶ We can use the bode diagram to identify how to tune the controller  $K$  based on the system Gain and Phase margins.
- ▶ The **gain margin** is defined as the change in open-loop gain required to make the system unstable. Systems with greater gain margins can withstand greater changes in system parameters before becoming unstable in closed-loop.
- ▶ The **phase margin** is defined as the change in open-loop phase shift required to make a closed-loop system unstable.

# Bode-based design example

- Consider the following system:



- We can use the bode diagram to identify how to tune the controller  $K$  based on the system Gain and Phase margins.

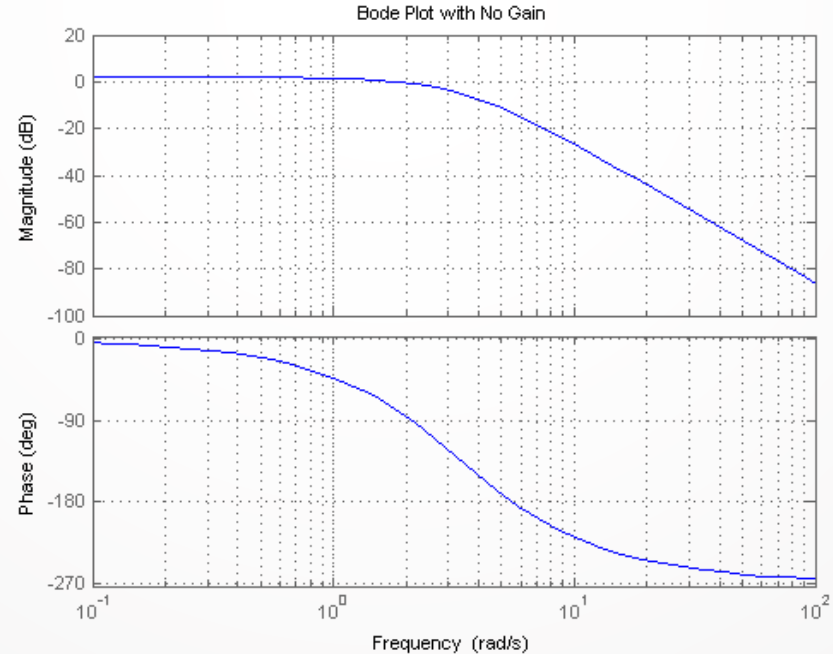


# Bode-based design example

- ▶ Consider the following transfer function:

$$H(s) = \frac{Y(s)}{U(s)} = \frac{50}{s^3 + 9s^2 + 30s + 40}$$

- ▶ Bode Plot Open-Loop:

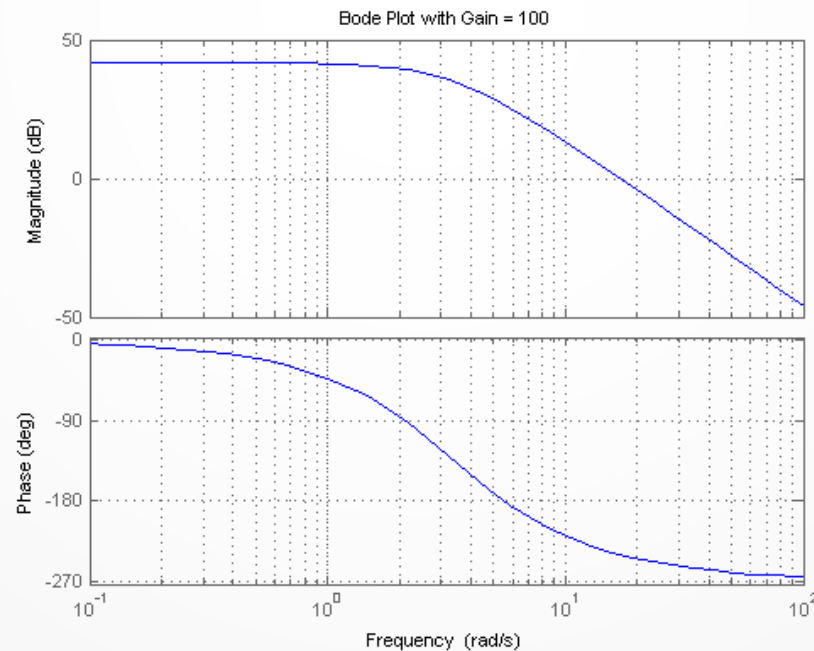


# Bode-based design example

- Consider the following transfer function:

$$H(s) = \frac{Y(s)}{U(s)} = \frac{50}{s^3 + 9s^2 + 30s + 40}$$

- Bode Plot with Gain  $K = 100$

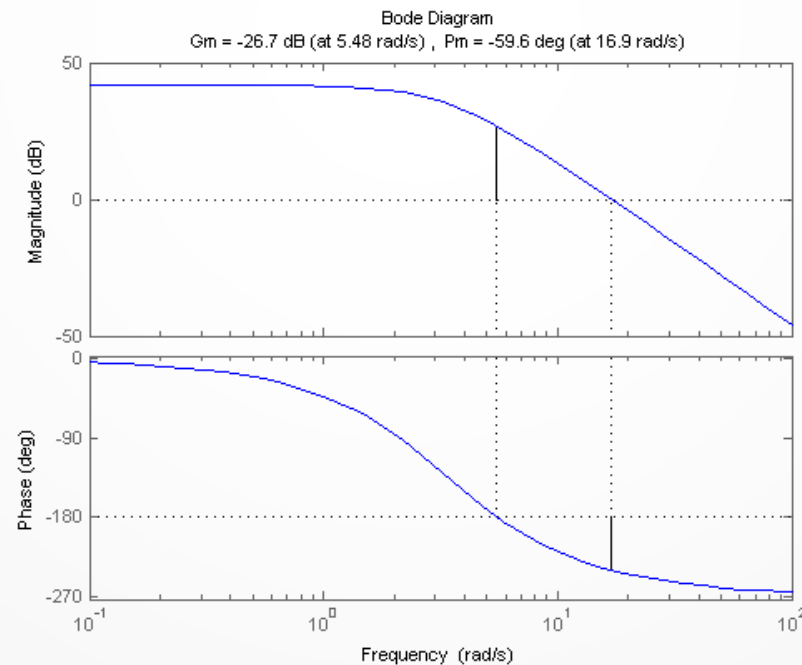


# Bode-based design example

- Consider the following transfer function:

$$H(s) = \frac{Y(s)}{U(s)} = \frac{50}{s^3 + 9s^2 + 30s + 40}$$

- Computation of Gain margin for  $K = 100$





# 5. State Space Control

- ▶ Consider the system:

$$\begin{aligned}\dot{\mathbf{x}} &= A\mathbf{x} + Bu \\ y &= C\mathbf{x} + Du\end{aligned}$$

- ▶ How do we define observability?
- ▶ How do we define controllability?
- ▶ How do we tune a simple controller to tune the dynamic response as desired? To place the poles where desired?

# State Space system Stability

- ▶ Consider the system:

$$\begin{aligned}\dot{\mathbf{x}} &= A\mathbf{x} + Bu \\ y &= C\mathbf{x} + Du\end{aligned}$$

- ▶ **Stability:** The system is stable if the eigenvalues of matrix  $A$  are all with negative real part.

# State Space system Observability

- Consider the system:

$$\begin{aligned}\dot{\mathbf{x}} &= A\mathbf{x} + Bu \\ y &= C\mathbf{x} + Du\end{aligned}$$

- Observability:** The system is observable if the initial state  $x_0$  can be determined from the system output,  $y(t)$ , over some finite time  $t_0 < t < t_f$ . For LTI systems, a system is observable if the observability matrix is of full rank.

$$OB = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

# State Space system Controllability

- ▶ Consider the system:

$$\begin{aligned}\dot{\mathbf{x}} &= A\mathbf{x} + Bu \\ y &= C\mathbf{x} + Du\end{aligned}$$

- ▶ **Controllability:** A system is controllable if there exists a control input,  $u(t)$ , that transfers any state of the system to zero in finite time. For LTI systems, a system is controllable if the controllability matrix is of full rank.

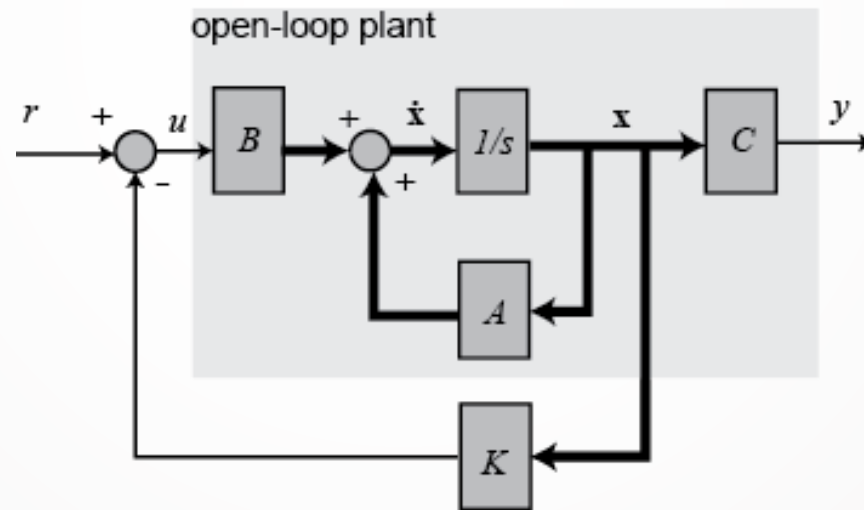
$$CO = [B|AB|A^2B|\dots|A^{n-1}B]$$

# Pole Placement

- ▶ Consider the system:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$
$$y = C\mathbf{x} + D\mathbf{u}$$

- ▶ Then the following structure is correct:



# Pole Placement

- ▶ Assuming zero reference:

$$u = -Kx$$

- ▶ The state space equations of the closed loop system take the form:

$$\begin{aligned}\dot{\mathbf{x}} &= (\mathbf{A} - \mathbf{BK})\mathbf{x} \\ \mathbf{y} &= (\mathbf{C} - \mathbf{DK})\mathbf{x}\end{aligned}$$

- ▶ The pole placement techniques is about how to find the gain matrix  $K$  such that the poles of the closed-loop system go to desired locations. This is under the assumption of an LTI system and infinite actuator dynamic range and bandwidth.

# Pole Placement

## ► Pole Placement in MATLAB:

```
p1 = -10 + 10i;
```

```
p2 = -10 - 10i;
```

```
p3 = -50;
```

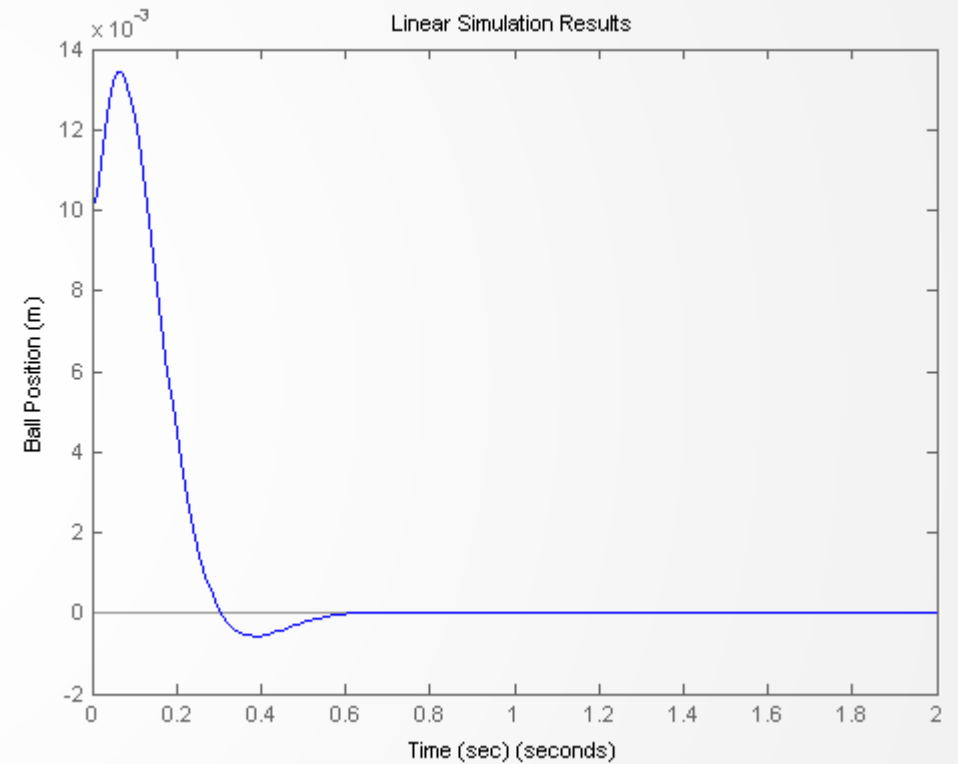
```
K = place(A,B,[p1 p2 p3]);
```

```
sys_cl = ss(A-B*K,B,C,0);
```

```
lsim(sys_cl,u,t,x0);
```

```
xlabel('Time (sec)')
```

```
ylabel('Ball Position (m)')
```



# Pole Placement

## ▶ Pole Placement in MATLAB:

```
p1 = -20 + 20i;
```

```
p2 = -20 - 20i;
```

```
p3 = -100;
```

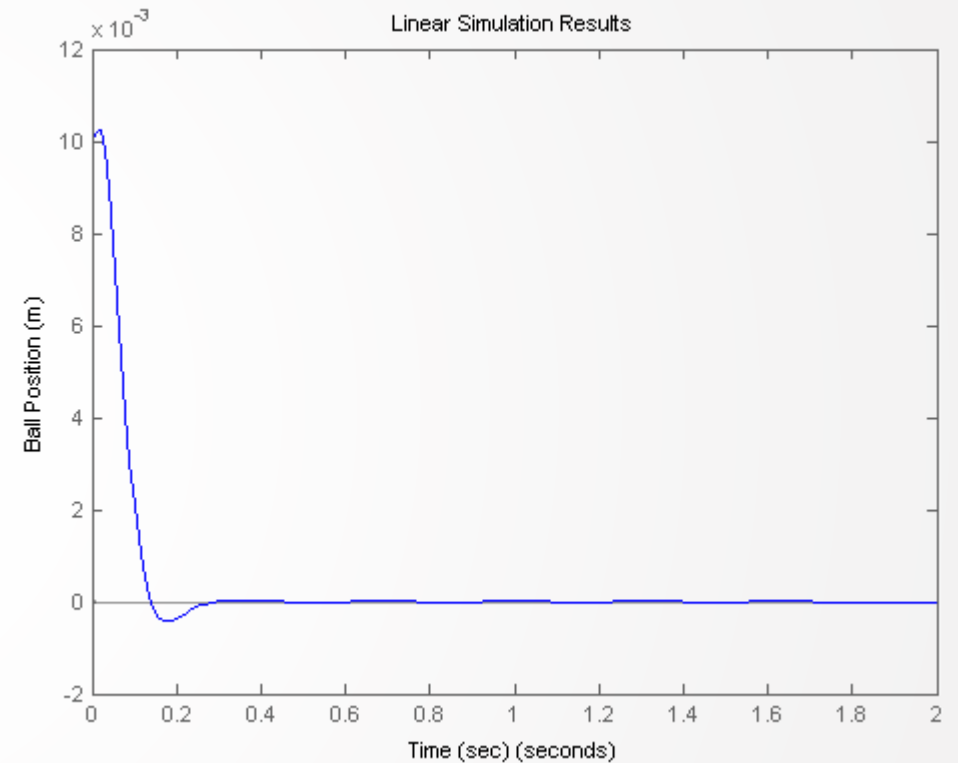
```
K = place(A,B,[p1 p2 p3]);
```

```
sys_cl = ss(A-B*K,B,C,0);
```

```
lsim(sys_cl,u,t,x0);
```

```
xlabel('Time (sec)')
```

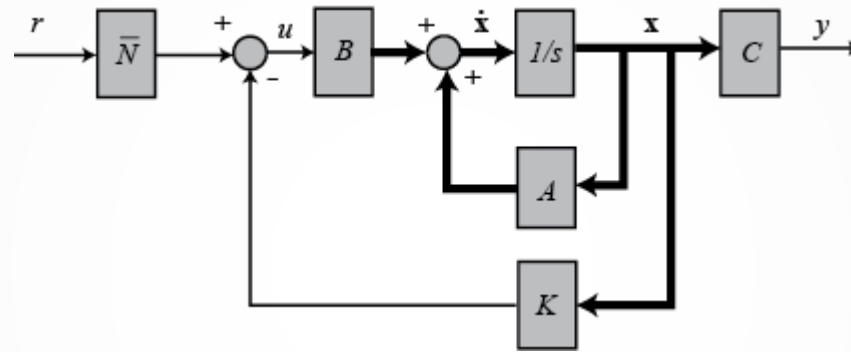
```
ylabel('Ball Position (m)')
```





# Pole Placement

- Introduce general reference input (not only zero)



$$Nbar = \text{rscale}(\text{sys}, K)$$

- `rscale` command: it does the reference scaling which will eliminate the steady-state error to a step reference for a continuous-time, single-input system with full-state feedback

# Aircraft Pitch example

- Consider the following system pitch representation:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -0.313 & 56.7 & 0 \\ -0.0139 & -0.426 & 0 \\ 0 & 56.7 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0.232 \\ 0.0203 \\ 0 \end{bmatrix} [\delta]$$
$$y = [0 \quad 0 \quad 1] \begin{bmatrix} \alpha \\ q \\ \theta \end{bmatrix} + [0][\delta]$$

- Which is of the general form:

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



# Aircraft Pitch example

- ▶ Control Design goals:
  - ▶ Overshoot:  $<10\%$
  - ▶ Rise time:  $< 2\text{sec}$
  - ▶ Setting time:  $<10\text{sec}$
  - ▶ Steady-state error:  $<2\%$

# Aircraft Pitch example

- ▶ Is the system controllable?
  - ▶ Is  $\mathcal{C} = [B|AB|\dots|A^{n-1}B]$  of full rank?

```
A = [-0.313 56.7 0; -0.0139 -0.426 0; 0  
56.7 0];
```

```
B = [0.232; 0.0203; 0];
```

```
C = [0 0 1];
```

```
D = [0];
```

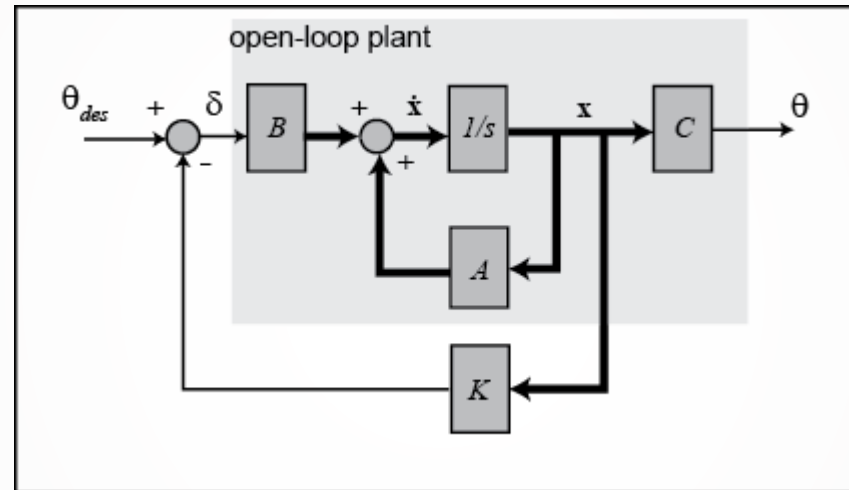
```
co = ctrb(A,B);
```

```
Controllability = rank(co)
```

- ▶ co=3 so yes

# Aircraft Pitch example

- Control design via Pole Placement



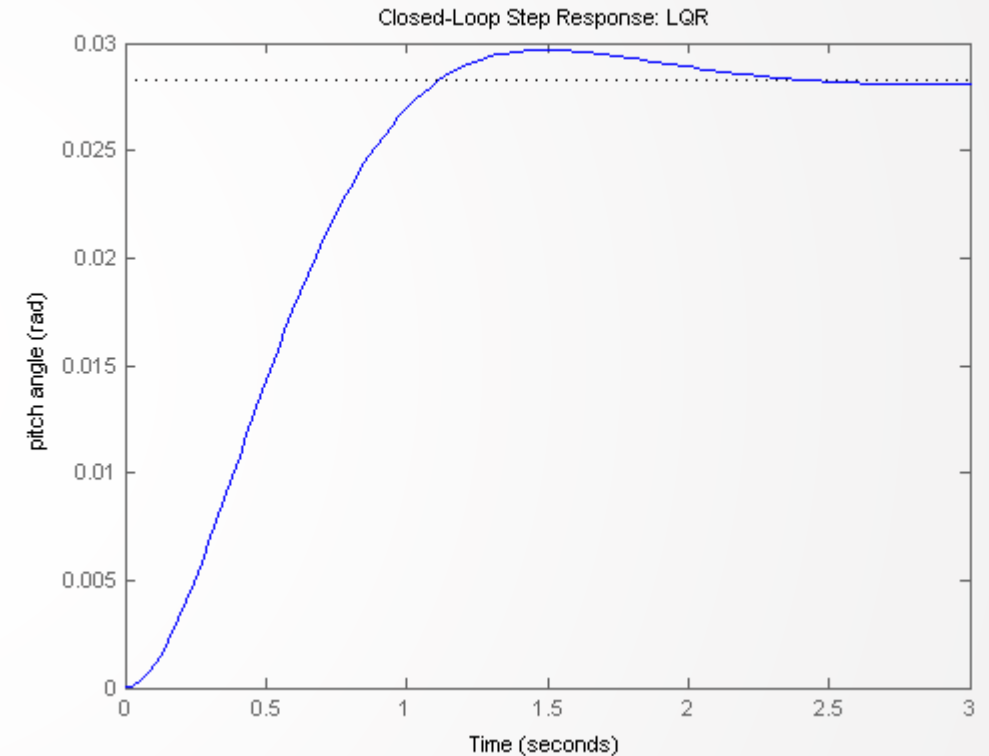
$$\mathbf{x} = [\alpha, q, \theta]^T$$
$$\delta = \theta_{des} - L\mathbf{x}$$

- We know from the above that we can place the closed-loop poles of the system anywhere we would like. The question then that is left is, where should we place them? If we have a standard first- or second-order system, we then have relationships that directly relate pole locations to characteristics of the step response and can use these relations to place the poles in order to meet our given requirements. This process becomes more difficult if we have a higher-order system or zeros. With a higher-order system, one approach is to place the higher-order poles 5-10 times farther to the left in the complex plane than the dominant poles, thereby leading them to have negligible contribution to the transient response. The effect of zeros is more difficult to address using a pole-placement approach to control. Another limitation of this pole-placement approach is that it doesn't explicitly take into account other factors like the amount of required control effort.

# Aircraft Pitch example

## ► LQR Synthesis

```
p = 50;  
Q = p*C'*C;  
R = 1;  
[K] = lqr(A,B,Q,R)  
sys_cl = ss(A-B*K, B, C, D);  
step(0.2*sys_cl)  
ylabel('pitch angle (rad)');  
title('Closed-Loop Step Response');
```



# Code Example

## ▶ MATLAB DC Motor Control Example

- ▶ [https://github.com/unr-arl/autonomous\\_mobile\\_robot\\_design\\_course/tree/master/matlab/control-systems/lqr](https://github.com/unr-arl/autonomous_mobile_robot_design_course/tree/master/matlab/control-systems/lqr)
- ▶ [https://github.com/unr-arl/autonomous\\_mobile\\_robot\\_design\\_course/tree/master/matlab/control-systems/pid](https://github.com/unr-arl/autonomous_mobile_robot_design_course/tree/master/matlab/control-systems/pid)
- ▶ [https://github.com/unr-arl/autonomous\\_mobile\\_robot\\_design\\_course/tree/master/matlab/control-systems/pid-cruise-control](https://github.com/unr-arl/autonomous_mobile_robot_design_course/tree/master/matlab/control-systems/pid-cruise-control)
- ▶ MATLAB 2016 Live note

A black and white photograph of a drone flying in the foreground. The drone is a quadcopter with a white protective cover over its camera. In the background, several construction cranes are visible, slightly out of focus. The overall scene is a construction site.

**Thank you!**

Please ask your question!