

ECEN/MAE 3723 – Systems I

MATLAB Lecture 3

Lecture Overview



- Building Models for LTI System
 - Continuous Time Models
 - Discrete Time Models
- Combining Models
- Transient Response Analysis
- Frequency Response Analysis
- Stability Analysis Based on Frequency Response
- Other Information

Building Models for LTI System

- Control System Toolbox supports continuous time models and discrete time models of the following types*:
 - Transfer Function
 - Zero-pole-gain
 - State Space

Continuous Time Transfer Function(1)

Function: Use **tf** function create transfer function of following form:

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

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

Matlab Output

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

Continuous Time Transfer Function(2)

Include **delay** to continuous time Transfer Function

Example $H(s) = e^{-2s} \frac{2s + 1}{s^2 + 3s + 2}$

```
>>num = [2 1];  
>>den = [1 3 2];  
>>H=tf(num,den,'inputdelay',2)
```

Matlab Output

Transfer function:

$$\exp(-2*s) * \frac{2 s + 1}{s^2 + 3 s + 2}$$

Continuous Time Transfer Function(3)

Function: Use **zpk** function to create transfer function of following form:

$$\text{Example } H(s) = \frac{2s+1}{s^2+3s+2} = 2 \frac{s+0.5}{(s+1)(s+2)}$$

```
>>num = [-0.5];  
>>den = [-1 -2];  
>>k = 2;  
>>H=zpk(num,den,k)
```

Matlab Output

```
Zero/pole/gain:  
  2 (s+0.5)  
-----  
(s+1) (s+2)
```

Continuous Time State Space Models(1)

State Space Model for dynamic system

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

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

Matrices: $\mathbf{A}$ is state matrix; $\mathbf{B}$ is input matrix; $\mathbf{C}$ is output matrix; and $\mathbf{D}$ is direct transmission matrix

Vectors: $\mathbf{x}$ is state vector; $\mathbf{u}$ is input vector; and $\mathbf{y}$ is output vector

Note: Only apply to system that is linear and time invariant

Continuous Time State Space Models(2)

Function: Use **ss** function creates state space models. For example:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ -5 & -2 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 3 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} 0 \end{bmatrix}$$

```
>>A = [0 1;-5 -2];  
>>B = [0;3];  
>>C = [0 1];  
>>D = [0];  
>>sys=ss(A,B,C,D)
```

Matlab Output

a =			b =		
	x1	x2		u1	
x1	0	1	x1	0	
x2	-5	-2	x2	3	
c =			d =		
	x1	x2		u1	
y1	0	1	y1	0	

Conversion between different models

Converting From	Converting to	Matlab function
Transfer Function	Zero-pole-gain	$[z, p, k] = \text{tf2zp}(\text{num}, \text{den})$
Transfer Function	State Space	$[A, B, C, D] = \text{tf2ss}(\text{num}, \text{den})$
Zero-pole-gain	Transfer Function	$[\text{num}, \text{den}] = \text{zp2tf}(z, p, k)$
Zero-pole-gain	State Space	$[A, B, C, D] = \text{zp2ss}(z, p, k)$
State Space	Transfer Function	$[\text{num}, \text{den}] = \text{ss2tf}(A, B, C, D)$
State Space	Zero-pole-gain	$[z, p, k] = \text{ss2zp}(A, B, C, D)$

Lecture Overview



- Building Models for LTI System
 - Continuous Time Models
 - Discrete Time Models
- Combining Models
- Transient Response Analysis
- Frequency Response Analysis
- Stability Analysis Based on Frequency Response
- Other Information

Discrete Time Transfer Function(1)

Function: Use **tf** function create transfer function of following form:

Example: $H(z) = \frac{2z + 1}{z^2 + 3z + 2}$ with sampling time 0.4

```
>>num = [2 1];  
>>den = [1 3 2];  
>>Ts=0.4;  
>>H=tf(num,den,Ts)
```

Matlab Output

Transfer function:

2 z + 1

z^2 + 3 z + 2

Sampling time: 0.4

Discrete Time Transfer Function(2)

Function: Use **zpk** function to create transfer function of following form:

Example: $H(z) = 2 \frac{z + 0.5}{(z + 1)(z + 2)}$ with sampling time 0.4

```
>>num = [-0.5];  
>>den = [-1 -2];  
>>k = 2;  
>>Ts=0.4;  
>>H=zpk(num,den,k,Ts)
```

Matlab Output

Zero/pole/gain:

2 (z+0.5)

(z+1) (z+2)

Sampling time: 0.4

Discrete Time State Space Models(1)

- State Space Model for dynamic system

$$\mathbf{x}[n + 1] = \mathbf{A}\mathbf{x}[n] + \mathbf{B}\mathbf{u}[n]$$

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

Matrices: **A** is state matrix; **B** is input matrix; **C** is output matrix; and **D** is direct transmission matrix

Vectors: **x** is state vector; **u** is input vector; and **y** is output vector

n is the discrete-time or time-index

Note: Only apply to system that is linear and time invariant

Discrete Time State Space Models(2)

Function: Use **ss** function creates state space models. For example:

$$\mathbf{x}[n] = \begin{bmatrix} x_1[n] \\ x_2[n] \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ -5 & -2 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 3 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} 0 \end{bmatrix}$$

```
>>A = [0 1; -5 -2];  
>>B = [0;3];  
>>C = [0 1];  
>>D = [0];  
>>Ts= [0.4];  
>>sys=ss(A,B,C,D,Ts)
```

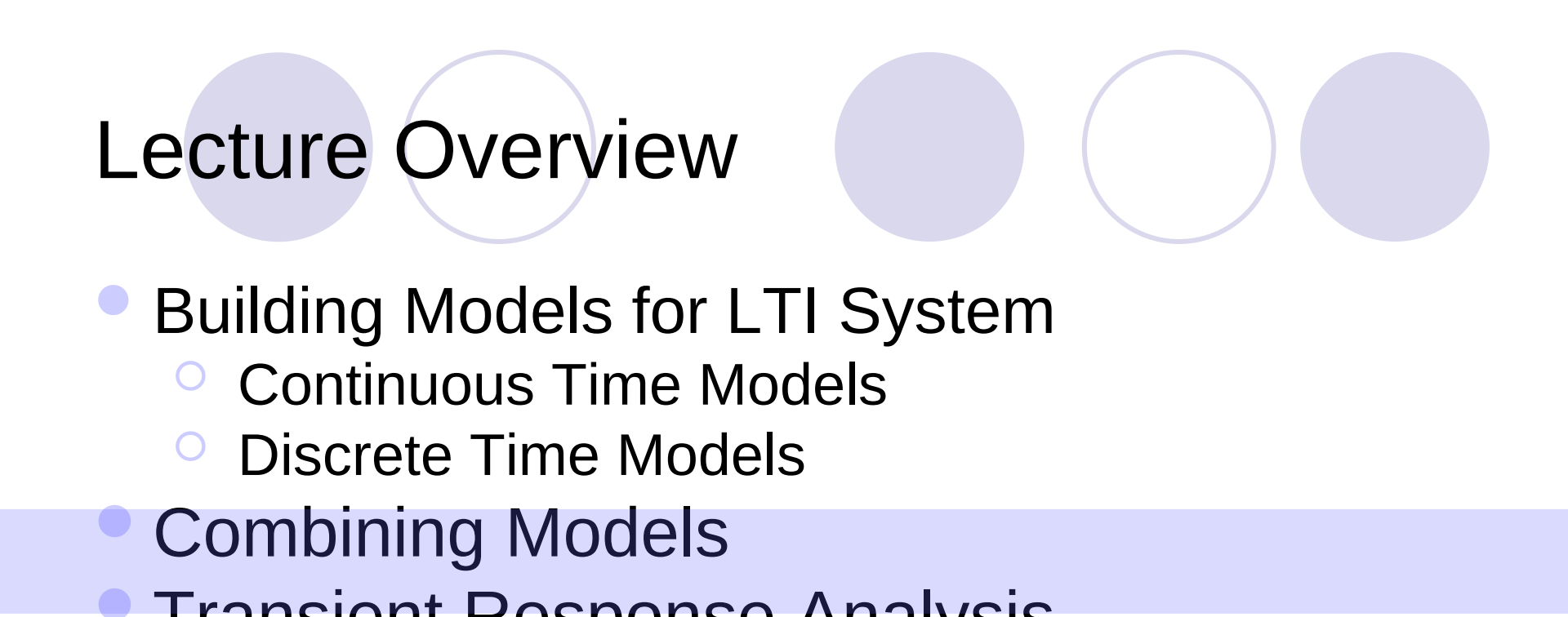
Matlab Output

a =		b =			
		x1	x2		u1
x1	0	1		x1	0
x2	-5	-2		x2	3

c =		d =			
		x1	x2		u1
y1	0	1		y1	0

Sampling time: 0.4

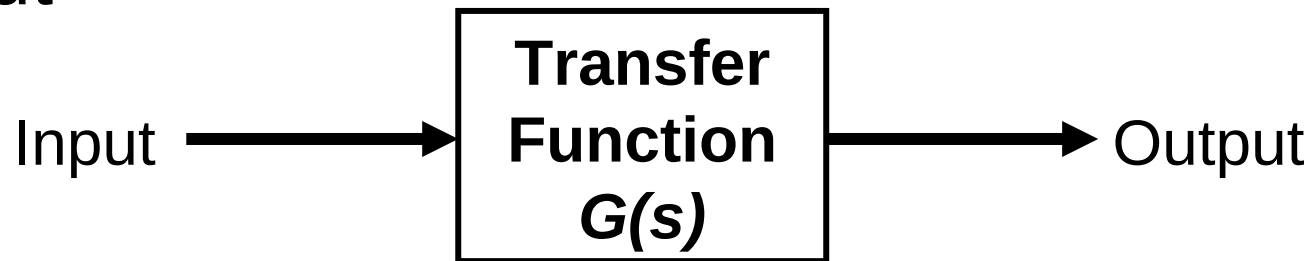
Lecture Overview

A decorative graphic at the top of the slide features five circles arranged horizontally. The first, third, and fifth circles are solid light purple, while the second and fourth are hollow with a light purple outline. Below these circles is a solid light purple horizontal bar that spans the width of the slide.

- Building Models for LTI System
 - Continuous Time Models
 - Discrete Time Models
- Combining Models
- Transient Response Analysis
- Frequency Response Analysis
- Stability Analysis Based on Frequency Response
- Other Information

Combining Models(1)

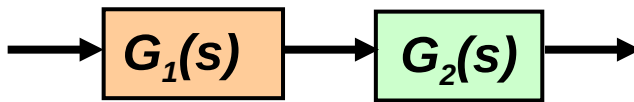
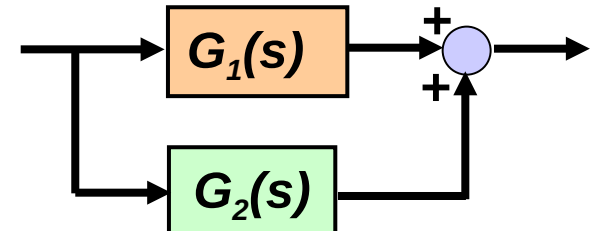
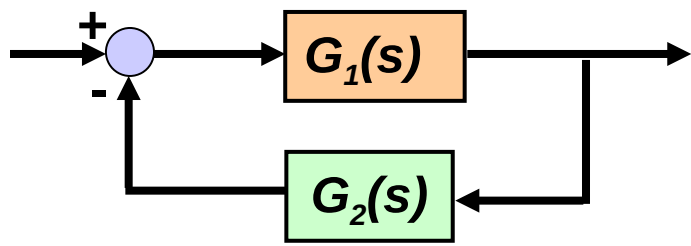
- A model can be thought of as a block with inputs and outputs (block diagram) and containing a transfer function or a state-space model inside it
- A symbol for the mathematical operations on the input signal to the block that produces the output



Elements of a Block Diagram

Combining Models(2)

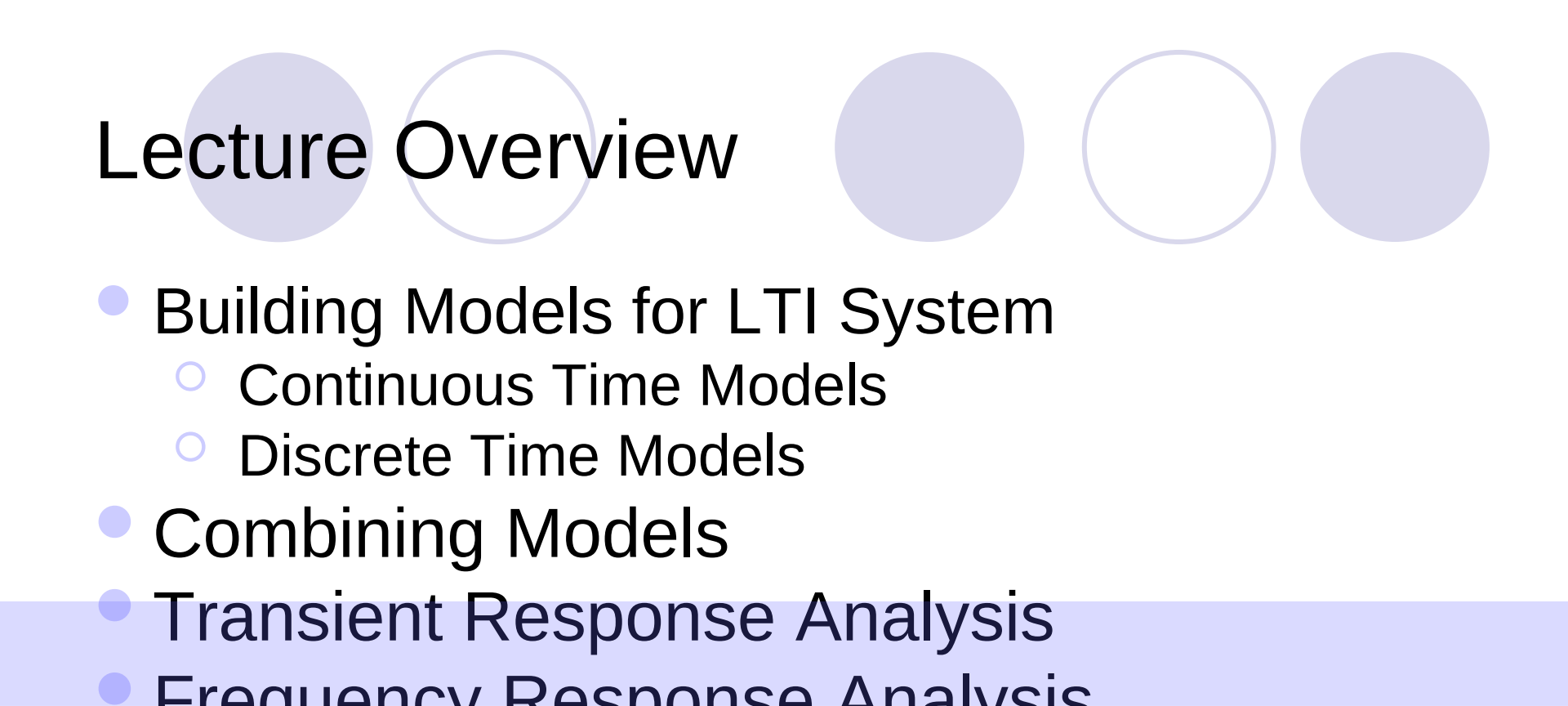
The Following Matlab functions can be used to perform basic block diagram manipulation

Combination	Matlab Command
 A block diagram showing two blocks, $G_1(s)$ (orange) and $G_2(s)$ (green), connected in series. An input arrow enters $G_1(s)$, and an output arrow exits $G_2(s)$.	<code>sys = series(G1, G2)</code>
 A block diagram showing two parallel paths. An input arrow splits into two branches. The top branch contains block $G_1(s)$ (orange), and the bottom branch contains block $G_2(s)$ (green). The outputs of both blocks are summed at a summing junction (blue circle with two '+' signs), and the result is the final output arrow.	<code>sys = parallel(G1, G2)</code>
 A block diagram showing a feedback loop. An input arrow enters a summing junction (blue circle with a '+' sign for the input and a '-' sign for the feedback). The output of the summing junction goes into block $G_1(s)$ (orange). The output of $G_1(s)$ is the system output and also branches off to go into block $G_2(s)$ (green). The output of $G_2(s)$ is fed back to the negative input of the summing junction.	<code>sys = feedback(G1, G2)</code>

Basic arithmetic operations of Models

Arithmetic Operations	Matlab Code
Addition	<code>sys = G1+G2;</code>
Multiplication	<code>sys = G1*G2;</code>
Inversion	<code>sys = inv(G1);</code>

Lecture Overview

A decorative graphic at the top of the slide features five circles arranged horizontally. The first, third, and fifth circles are solid light purple, while the second and fourth are hollow with a light purple outline. Below these circles is a horizontal light purple bar that spans the width of the slide, partially obscuring the text of the fourth and fifth items in the list.

- Building Models for LTI System
 - Continuous Time Models
 - Discrete Time Models
- Combining Models
- Transient Response Analysis
- Frequency Response Analysis
- Stability Analysis Based on Frequency Response
- Other Information

Transient Response Analysis(1)

- Transient response refers to the process generated in going from the initial state to the final state
- Transient responses are used to investigate the time domain characteristics of dynamic systems
- Common responses: step response, impulse response, and ramp response

Transient Response Analysis(2)

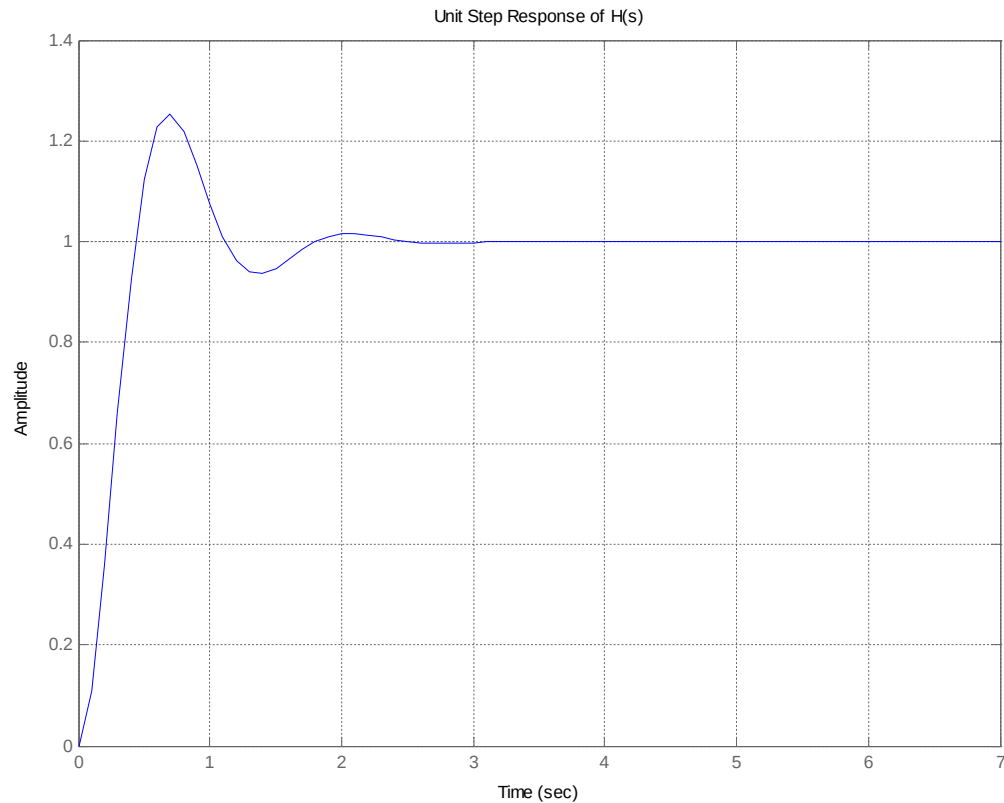
Unit step response of the transfer function system

Consider the system: $H(s) = \frac{25}{s^2 + 4s + 25}$

```
%*****Numerator & Denominator of H(s)
>>num = [0 0 25];den = [1 4 25];
%*****Specify the computing time
>>t=0:0.1:7;
>>step(num,den,t)
%*****Add grid & title of plot
>>grid
>>title('Unit Step Response of H(s)')
```

Transient Response Analysis(3)

Unit step response of $H(s)$



Transient Response Analysis(4)

Alternative way to generate Unit step response of the transfer function, $H(s)$

```
%*****Numerator & Denominator of H(s)
>>num = [0 0 25];den = [1 4 25];
%*****Create Model
>>H=tf(num,den);
>>step(H)
```

If step input is $10u(t)$, then step response is generated with the following command:

```
>>step(10*H)
```

Transient Response Analysis(5)

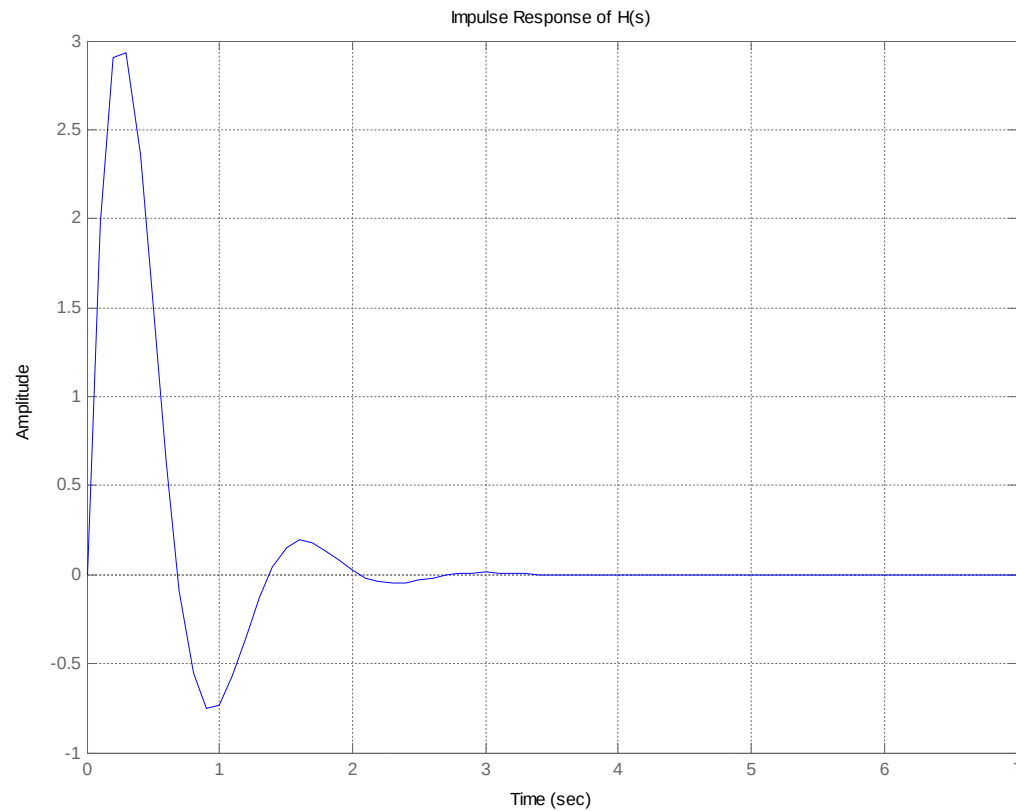
Impulse response of the transfer function system

Consider the system: $H(s) = \frac{25}{s^2 + 4s + 25}$

```
%*****Numerator & Denominator of H(s)
>>num = [0 0 25];den = [1 4 25];
%*****Specify the computing time
>>t=0:0.1:7;
>>impulse(num,den,t)
%*****Add grid & title of plot
>>grid
>>title('Impulse Response of H(s)')
```

Transient Response Analysis(6)

Impulse response of $H(s)$



Transient Response Analysis(7)

- **Ramp** response of the transfer function system
- There's no ramp function in Matlab
- To obtain ramp response of $H(s)$, divide $H(s)$ by “s” and use **step** function

Consider the system: $H(s) = \frac{25}{s^2 + 4s + 25}$

For unit-ramp input, $U(s) = \frac{1}{s^2}$. Hence

$$Y(s) = \frac{1}{s^2} \times \left(\frac{25}{s^2 + 4s + 25} \right) = \frac{1}{s} \times \overbrace{\frac{25}{s(s^2 + 4s + 25)}}^{\text{NEW } H(s)}$$

Indicate Step response

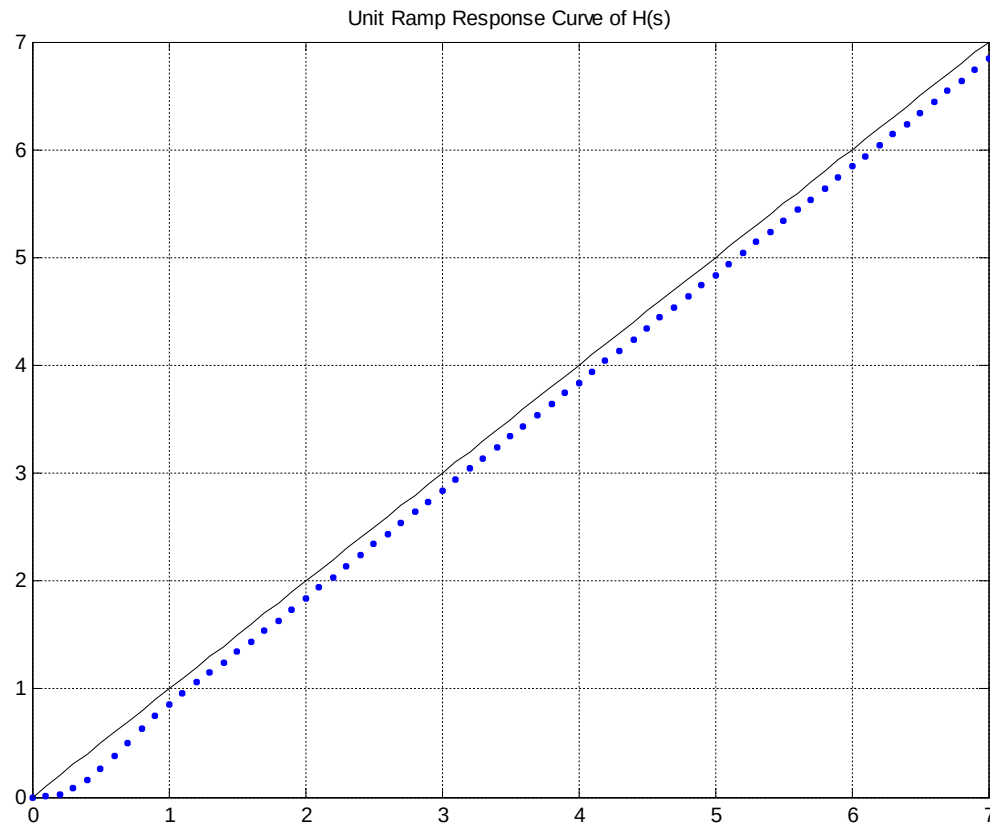
Transient Response Analysis(8)

Example: Matlab code for Unit Ramp Response

```
%*****Numerator & Denominator of NEW H(s)
>>num = [0 0 0 25];den = [1 4 25 0];
%*****Specify the computing time
>>t=0:0.1:7;
>>y=step(num,den,t);
%*****Plot input & the ramp response curve
>>plot(t,y, '.',t,t,'b-')
%*****Add grid & title of plot
>>grid
>>title('Unit Ramp Response Curve of H(s)')
```

Transient Response Analysis(9)

Unit Ramp response of $H(s)$



Lecture Overview



- Building Models for LTI System
 - Continuous Time Models
 - Discrete Time Models
- Combining Models
- Transient Response Analysis
- Frequency Response Analysis
- Stability Analysis Based on Frequency Response
- Other Information

Frequency Response Analysis(1)

- For Transient response analysis - hard to determine accurate model (due to noise or limited input signal size)
- Alternative: Use frequency response approach to characterize how the system behaves in the frequency domain
- Can adjust the frequency response characteristic of the system by tuning relevant parameters (design criteria) to obtain acceptable transient response characteristics of the system

Frequency Response Analysis(2)

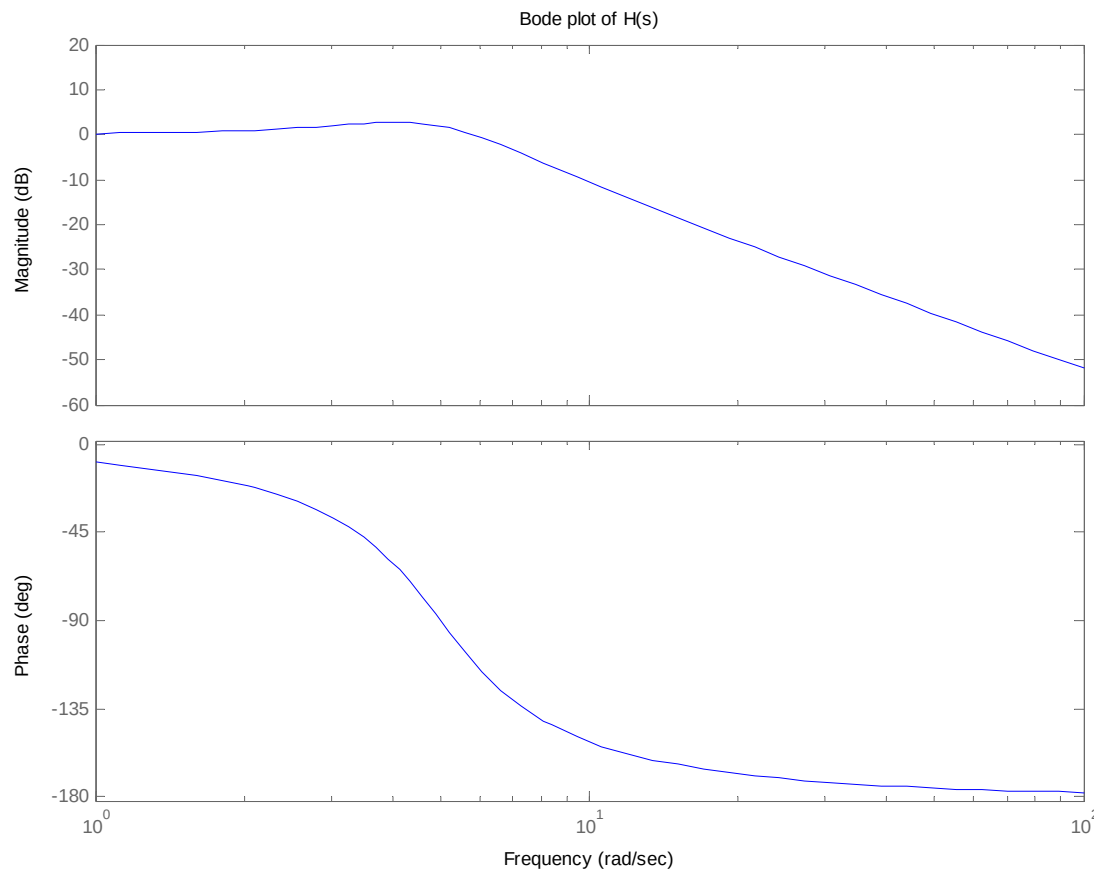
Bode Diagram Representation of Frequency Response

- Consists of two graphs:
 - Log-magnitude plot of the transfer function
 - Phase-angle plot (degree) of the transfer function
- Matlab function is known as 'bode'

```
%*****Numerator & Denominator of H(s)
>>num = [0 0 25];den = [1 4 25];
%*****Use 'bode' function
>>bode(num,den)
%*****Add title of plot
>>title('Bode plot of H(s)')
```

Frequency Response Analysis(3)

Example: Bode Diagram for $H(s) = \frac{25}{s^2 + 4s + 25}$



Bode magnitude plot

Bode phase plot

Lecture Overview



- Building Models for LTI System
 - Continuous Time Models
 - Discrete Time Models
- Combining Models
- Transient Response Analysis
- Frequency Response Analysis
- Stability Analysis Based on Frequency Response
- Other Information

Stability Analysis Based on Frequency Response(1)

- Stability analysis can also be performed using a Nyquist plot
- From Nyquist plot – determine if system is stable and also the degree of stability of a system
- Using the information to determine how stability may be improved
- Stability is determined based on the Nyquist Stability Criterion

Stability Analysis Based on Frequency Response(2)

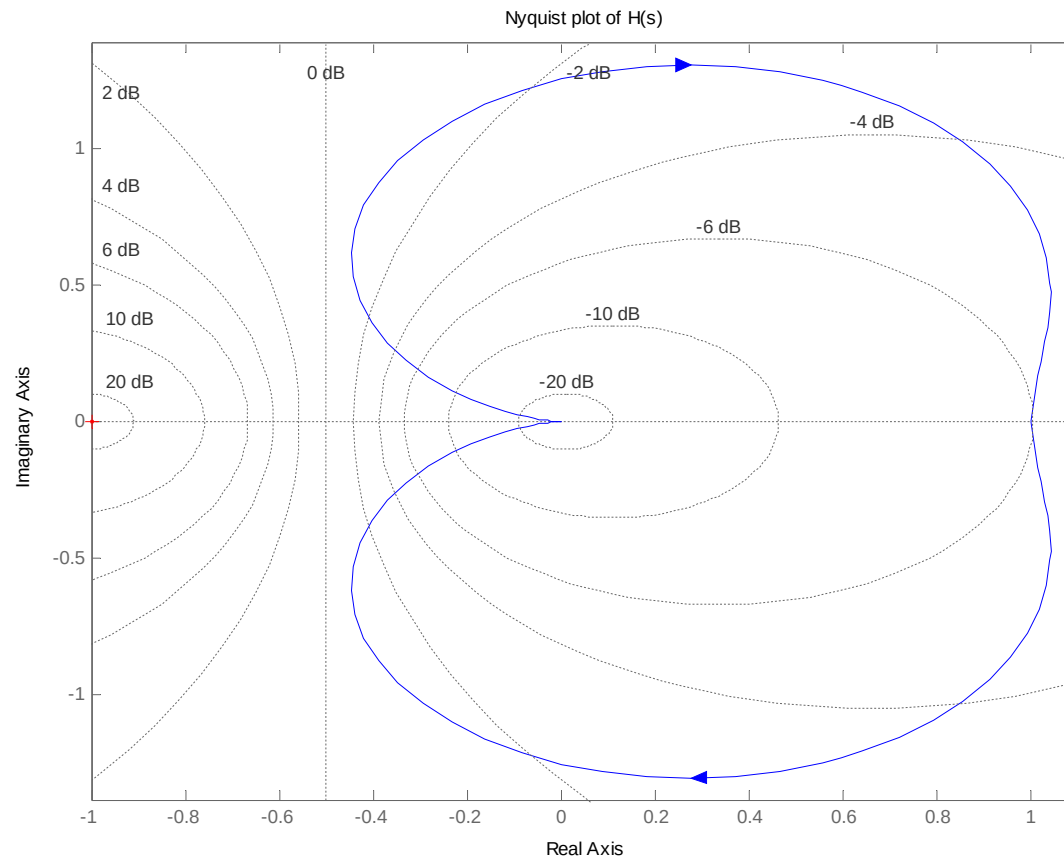
Example: Matlab code to draw a Nyquist Plot

Consider the system $H(s) = \frac{1}{s^2 + 0.8s + 1}$

```
%*****Numerator & Denominator of H(s)
>>num = [0 0 1];
>>den = [1 0.8 1];
%*****Draw Nyquist Plot
>>nyquist(num,den)
%*****Add grid & title of plot
>>grid
>>title('Nyquist Plot of H(s)')
```

Stability Analysis Based on Frequency Response(2)

The Nyquist Plot for $H(s) = \frac{1}{s^2 + 0.8s + 1}$



Lecture Overview

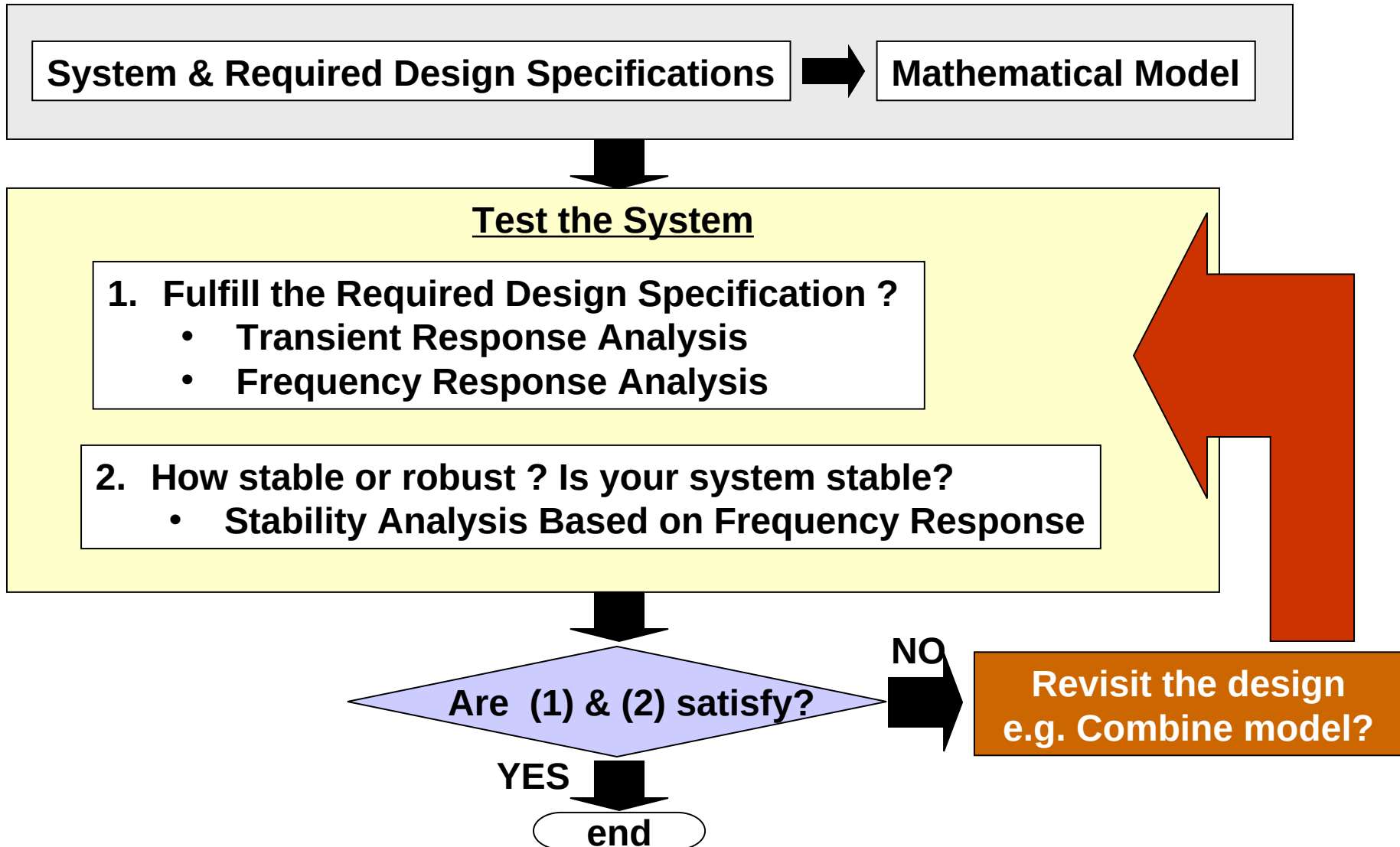


- Building Models for LTI System
 - Continuous Time Models
 - Discrete Time Models
- Combining Models
- Transient Response Analysis
- Frequency Response Analysis
- Stability Analysis Based on Frequency Response
- Other Information

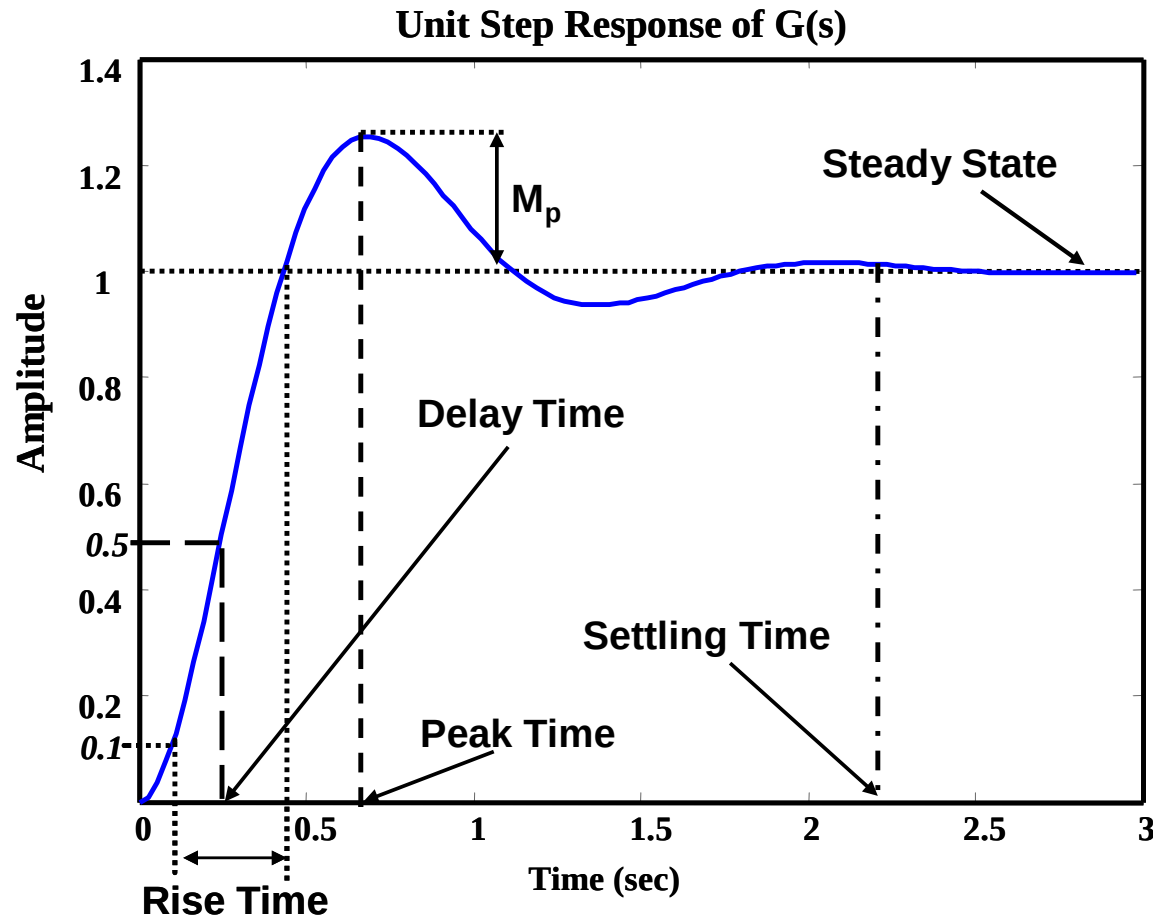
Other Information

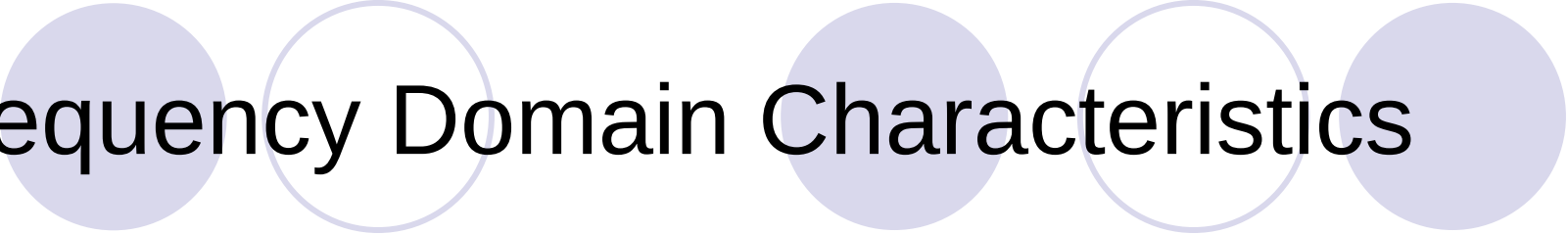
- Use help to find out more about the Matlab functions shown in this lecture
- Check out Control System Toolbox for other Matlab functions

Procedure of Designing a Control System



Transient response Specifications





Frequency Domain Characteristics

- What is the bandwidth of the system?
- What is the cutoff frequencies?
- What is the cutoff rate?
- Is the system sensitive to disturbance?

How the system behave in frequency domain?