

ELG3125B

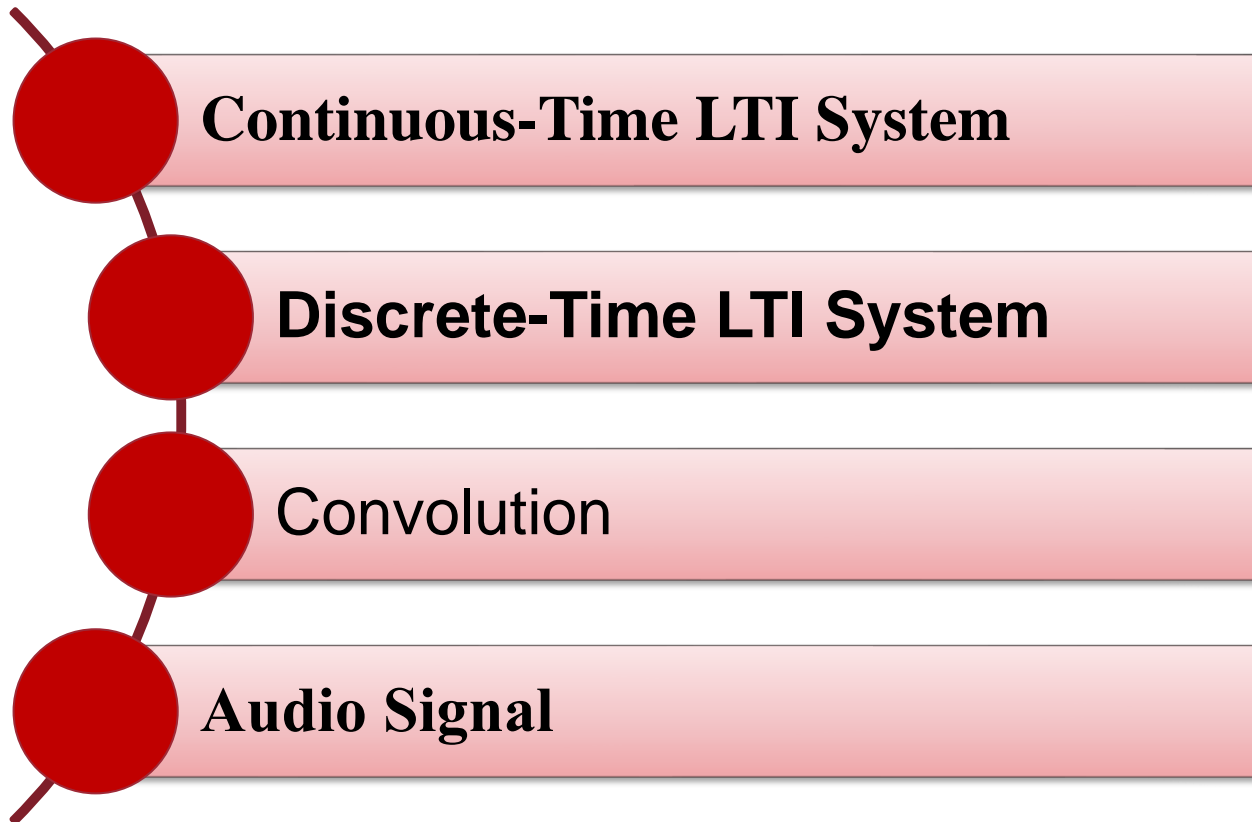
Signal and System Analysis Lab

• **Lab3: LTI Systems and Convolution Sum**

By: Hitham Jleed

hjleed@uottawa.ca

Outline



Continuous-Time LTI System

- N -th order linear constant-coefficient differential equation:

$$\sum_{k=0}^N a_k \frac{d^k y(t)}{dt^k} = \sum_{k=0}^M b_k \frac{d^k x(t)}{dt^k}$$

- N -th order linear constant-coefficient differential equation:

$$A = [a_N, a_{N-1}, \dots, a_0], \quad B = [b_M, b_{M-1}, \dots, b_0]$$

- To find the impulse response $h(t)$ of the system:

$$\text{impulse}(B, A, t)$$

- To compute the output given the input $x(t)$:

$$\text{lsim}(B, A, x, t)$$

Example: Continuous-Time LTI System

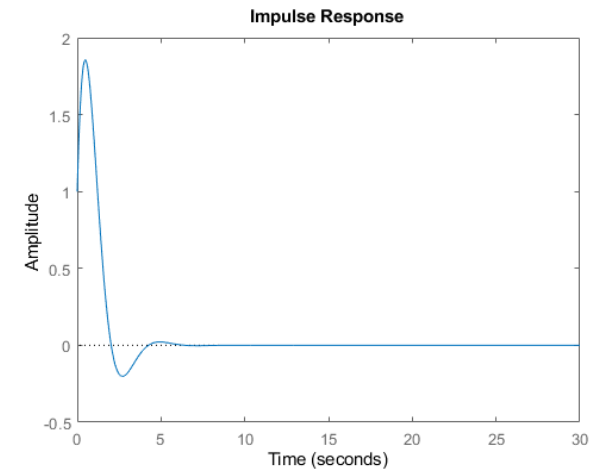
- Second order LTI system

$$\frac{d^2 y(t)}{dt^2} + 2 \frac{dy(t)}{dt} + 3y(t) = \frac{dx(t)}{dt} + 6x(t)$$

$$A = [1 \ 2 \ 3] \quad B = [0 \ 1 \ 6]$$

- Coefficients are highest-order first from left to right in the vector.
- The size of A and B should be the same.

```
A=[1 2 3];  
B=[0 1 6];  
t=0:0.01:30;  
impulse(B,A,t)
```



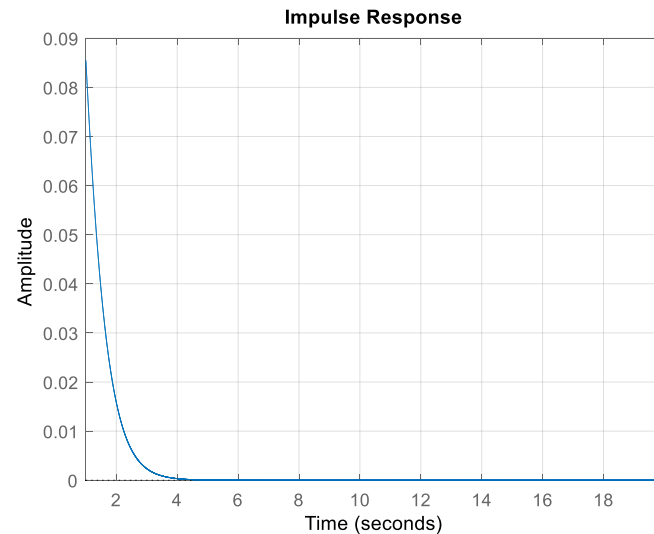
Assignment – Question 1

1. For the following continuous time LTI system:

$$\frac{d^2 y(t)}{dt^2} + 5 \frac{dy(t)}{dt} + 6y(t) = x(t) \quad (\text{system initially at rest})$$

a) Find and plot using Matlab the impulse response $h(t)$ of the system, using the function *impulse(b,a,t)* where b and a correspond to the coefficients in the differential equation (i.e. $a=[1, 5, 6]$ and $b=[0 \ 0 \ 1]$).

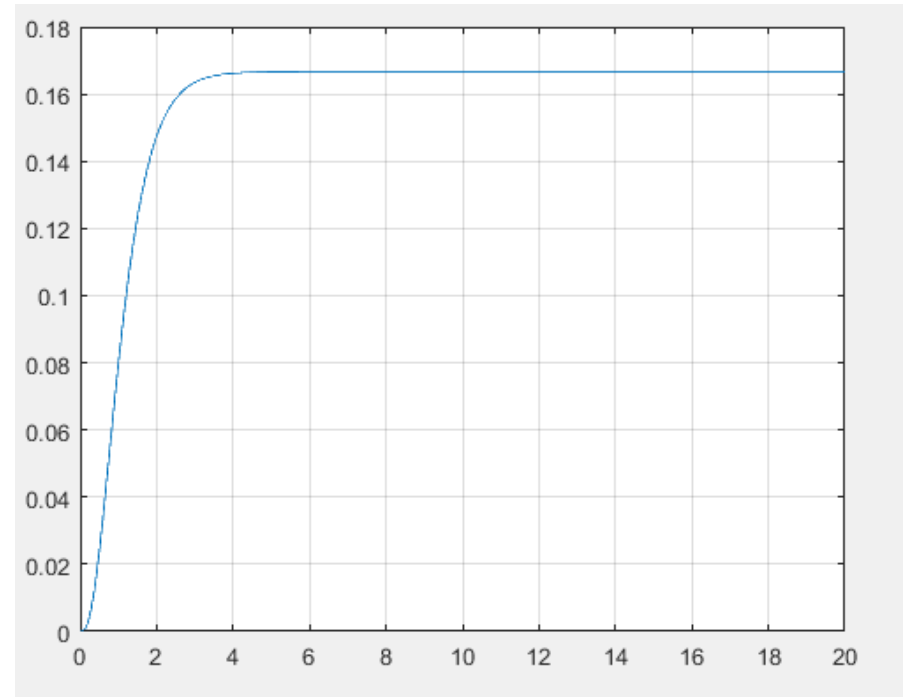
```
t=0:0.001:20;  
a=[1 5 6];  
b=[0 0 1];  
impulse(b,a,t);  
grid on;
```



Assignment ...

b). For an input signal defined as: $x(t) = (1 - e^{-3t})u(t)$
 compute the output $y(t)$ of that system, using the function $lsim(b,a,x,t)$.
 Use a time range long enough to see the $y(t)$ signal stabilize.

```
t=0:0.001:20;
a=[1, 5, 6];
b=[0, 0, 1];
x=1-exp(-3*t);
y=lsim(b,a,x,t);
plot(t,y);
grid on;
```



$$\frac{d^2 y(t)}{dt^2} + 5 \frac{dy(t)}{dt} + 6y(t) = x(t)$$

Discrete-Time LTI System

- N th-order linear constant-coefficient difference equation

$$\sum_{k=0}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k]$$

- Describe the system in Matlab

$$A = [a_0, a_1, \dots, a_N], B = [b_0, b_1, \dots, b_M]$$

- To find the impulse response $h[n]$ of the system:

$$\text{impz}(B,A,n)$$

- To compute the output given the input $x[n]$:

$$\text{filter}(B,A,x)$$



Exercise: Discrete-Time LTI System

Consider the system given by:

$$y[n] - 0.4 y[n-1] + 0.75 y[n-2] = 2.2403x[n] + 2.4908x[n-1] + 2.2403 x[n-2].$$

The MATLAB command `impz(B,A,n)` can be used to compute the first N samples of the impulse response of the causal LTI discrete-time system

Start Now ...



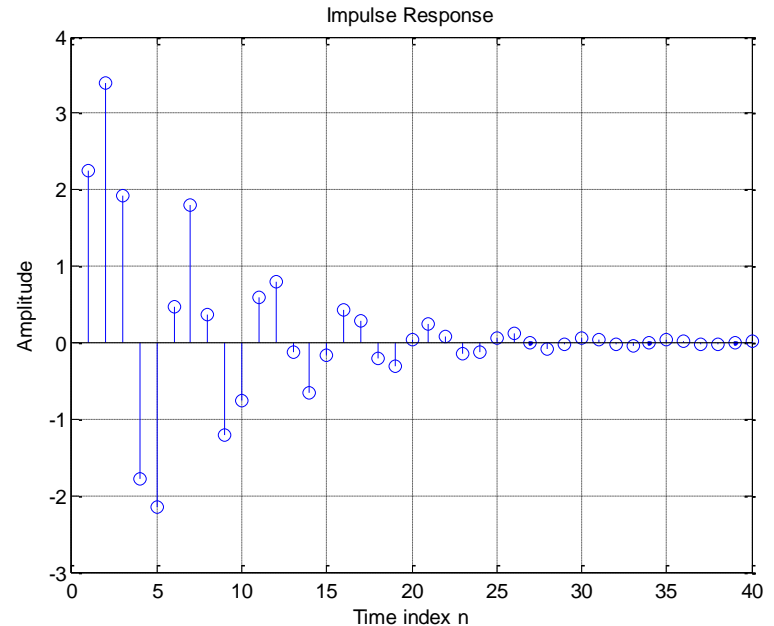
Answer: Discrete-Time LTI System

Consider the system given by:

$$y[n] - 0.4 y[n-1] + 0.75 y[n-2] = 2.2403x[n] + 2.4908x[n-1] + 2.2403 x[n-2].$$

The MATLAB command `h = impz(B,A,N)` can be used to compute the first N samples of the impulse response of the causal LTI discrete-time system

```
% Compute the impulse response y
clf;
N = 40;
B = [2.2403 2.4908 2.2403];
A = [1 -0.4 0.75];
h = impz(B,A,N);
% Plot the impulse response
stem(h);
xlabel('Time index n'); ylabel('Amplitude');
title('Impulse Response'); grid;
```



Convolution

- Convoluting two signals is very simple using MATLAB as follows. If it is required to convolute any two signals, you can use the *conv* instruction directly but you should care for the limits of the independent variable of the result as it will be the sum of the length of the independent variable of the convoluted signals.
- Note that the reverse operation is called deconvolution in which you provide the result of the convolution and one of the convoluted signals and will return the other convoluted signal.
- Type:
- `>> help conv`

Convolution sum

- Mathematical formula

$$y[n] = \sum_{-\infty}^{+\infty} x[n]h[n - k] = x[n] * h[n] = h[n] * x[n]$$

- Matlab function: `y=conv(x,h)`
- Length of signal `x[n]`: `length(x)`

Exercise: Illustration of the Filtering Concept

Consider the following two discrete-time systems characterized by the difference equations:

- **System No. 1**

$$y[n] = 0.5 x[n] + 0.27 x[n - 1] + 0.77 x[n - 2],$$

- **System No. 2**

$$y[n] = 0.45 x[n] + 0.5 x[n - 1] + 0.45 x[n - 2] + 0.53 y[n - 1] - 0.46 y[n - 2].$$

used the following input to compute the outputs of the above two systems.

$$x[n] = \cos\left(\frac{20\pi n}{256}\right) + \cos\left(\frac{200\pi n}{256}\right), \quad \text{with } 0 \leq n \leq 299.$$

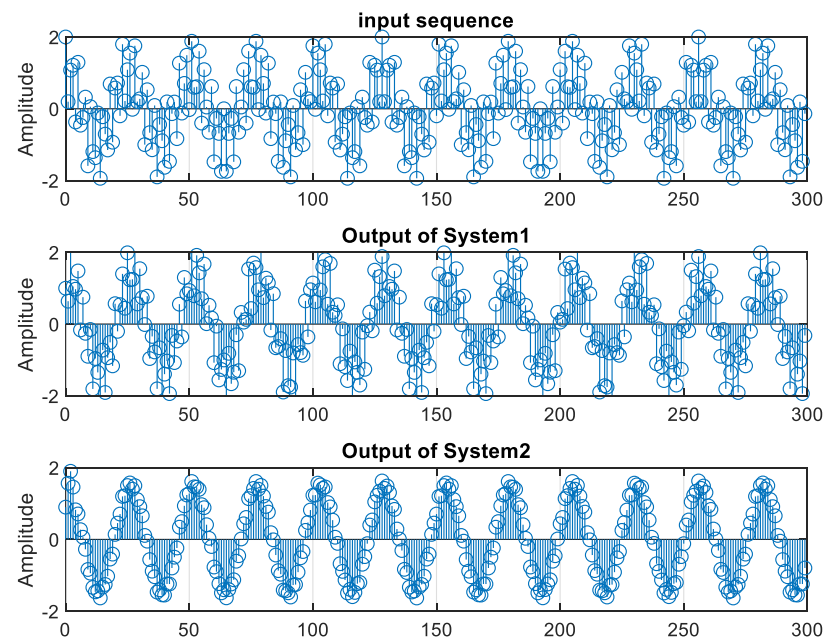
```
>> help filter
```

Start Now ...



```

% Generate the input sequence
close all;
n = 0:299;
x = cos(2*pi*10*n/256)+cos(2*pi*100*n/256);
% Compute the output sequences
B1 = [0.5 0.27 0.77];A1=[1 0 0];
y1 = filter(B1,A1,x); % Output of System
subplot(311);stem(n,x);axis([0 300 -2 2]);
ylabel('Amplitude');title('input sequence');
grid;
subplot(312); stem(n,y1);axis([0 300 -2 2]);
ylabel('Amplitude');title('Output of System1');
grid;
B2 = [0.45 0.5 0.45];A2 = [1 -0.53 0.46];
y2 = filter(B2,A2,x); % Output of System No. 2
% Plot the output sequences
subplot(313); stem(n,y2);axis([0 300 -2 2]);
ylabel('Amplitude');title('Output of System2');
grid;
    
```



System No. 1

$$y[n] = 0.5 x[n] + 0.27 x[n - 1] + 0.77 x[n - 2],$$

System No. 2

$$y[n] = 0.45 x[n] + 0.5 x[n - 1] + 0.45 x[n - 2] + 0.53 y[n - 1] - 0.46 y[n - 2].$$



$$y[n] - 0.53 y[n - 1] + 0.46 y[n - 2] = 0.45 x[n] + 0.5 x[n - 1] + 0.45 x[n - 2]$$



Learn How to Operate Music Files

- Read a music file in Matlab: *audioread()*
- Listen to the music: *Sound()*



Assignment – Question 3

- First obtain a discrete time input signal $x[n]$ by reading the file "Audio1.wav" from the course's website. This signal has a duration (length) of 190912 samples, with a sampling frequency of 16000 samples/sec. Visualize the resulting signal and listen to it.

```
>> help audioread
```

Start Now ...



Assignment – Question 3

```
[x,fs]=audioread('Audio1.wav');  
sound(x,fs); % Please don't execute the syntax unless you  
have headphones!
```

Download the [audio1.wav](#)



Assignment ...

$$h[n] = 0.1 \times (0.99)^n, 0 \leq n \leq 1000$$

$$y[n] = x[n] * h[n]$$

c) Use the function *conv()* to compute $y[n]$. Observe the size of the result $y[n]$, plot $y[n]$ and listen to it. What do you observe when directly comparing with $x[n]$? Does it sound different ?

Do it Now ...



```
x=audioread('Audio1.wav');  
n=0:1000;  
h=0.1*(0.99).^n;  
y=conv(x,h);  
stem(y);  
length(y) % has to be equal to length(x)+length(h)-1
```

Assignment ...

- d) Now use the function `filter` to compute $y[n]$, where the parameter `b` in the function `filter` corresponds to $h[n]$ and the parameter `a` is set to 1.0. Like the function `conv()`, the function `filter` also computes the result of a discrete time convolution in this case, but not over the same interval of output $y[n]$ values. Explain the difference.

```
x=audioread('Audio1.wav');  
n=0:1000;  
h=0.1*(0.99).^n;  
y=filter(h,1,x); % y=conv(h,x);  
stem(y);  
length(y) % has to be equal to length(x)
```

The END