

Digital Signal Processing

Lab5: audio effects

Instructor: Eng\ Samar Shaaban

E-mail: ssa10@fayoum.edu.eg

Github Repo: <https://github.com/SamarShabanCS/DSP>

Slack workspace: <https://fayoum-university-fci.slack.com>

Quiz

$$X[n-2] * \delta[n] = ? ?$$

Create audio

```
%% Creation of audio file
clc;close all;clear;

fs=44100;
time=2;
recObj1 = audiorecorder(fs,16,1); %define object of recorder
disp ('Start speaking.')
recordblocking(recObj1, time); %stop program to record
disp('End of Recording.');
```

x = getaudiodata(recObj1); %to catch the recorded data

audiowrite('test.wav', x, fs); %to store recorded data named as test.wav

% to listen: two ways

```
play(recObj1)
sound(x,fs)
```

Mono v.s Stereo

MONO



speaker 1



speaker 2



Signal

STEREO



speaker 1



speaker 2



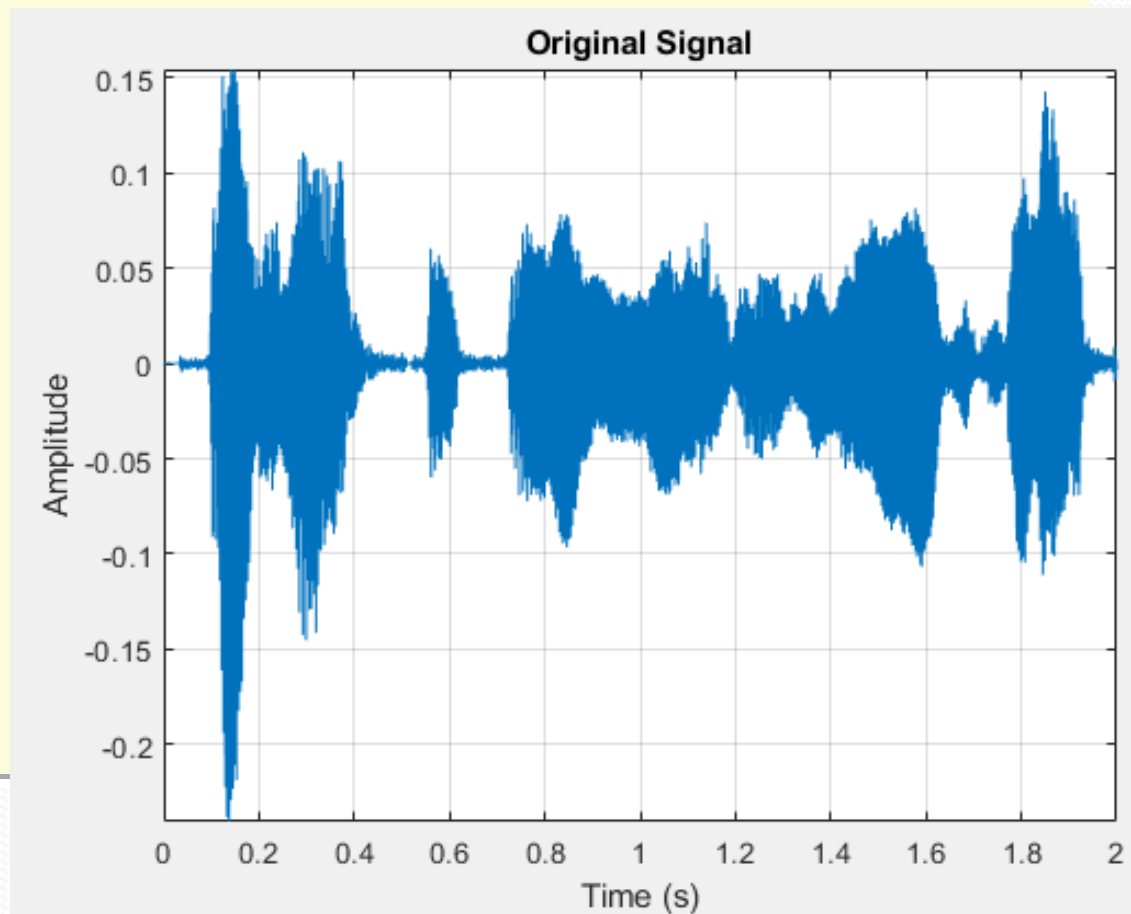
Signal 1



Signal 2

```
%% plotting the Original audio signal
```

```
N=fs*time;  
t=linspace(0,time,N);  
plot(t,x)  
xlabel('Time (s)')  
ylabel('Amplitude')  
title('Original Signal')  
axis tight  
grid on
```



Read audio file

- Read audio file
- Test sample frequency with different values (funny apps).
- Increase/decrease audio volume

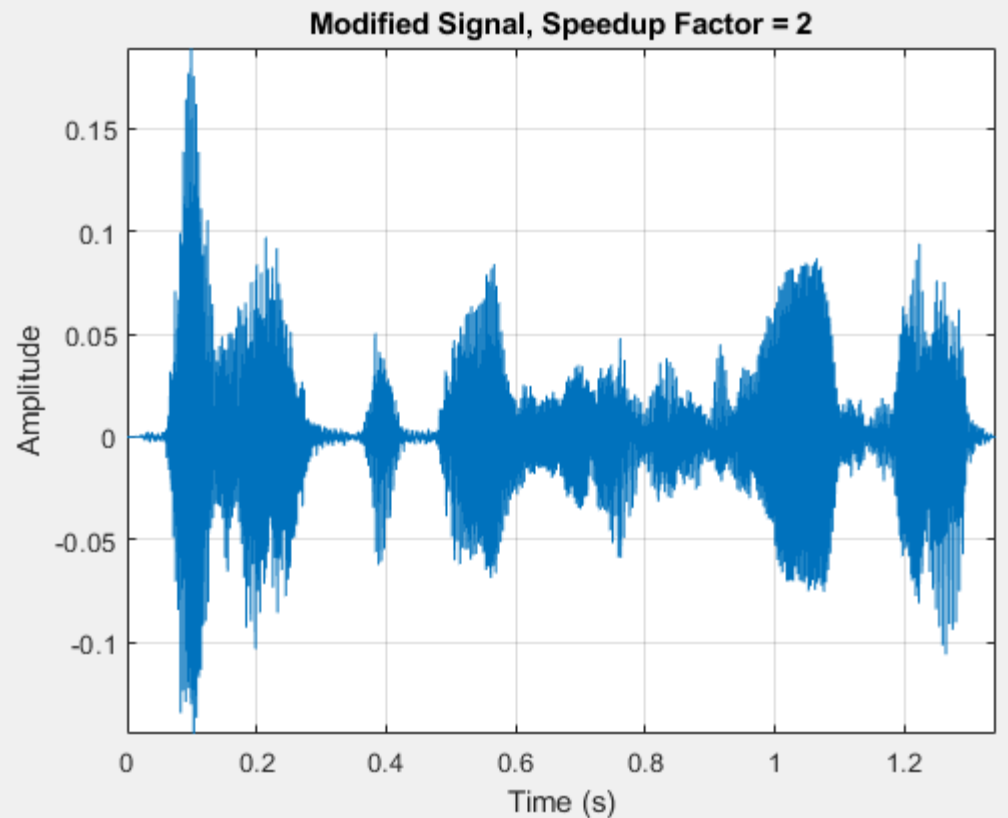
```
%% read signal
clc;close all;clear;
[x,fs]=audioread('test.wav');
sound(3*x,fs);  %increase or decrease volume
sound(5*x,1.5*fs); % speed signal using fs, try 1.5,0.7
```

```
%% play audio file
player = audioplayer(y,fs);
play(player); % comment this out it will play.
stop(player); % comment this out it will stop.
```

```

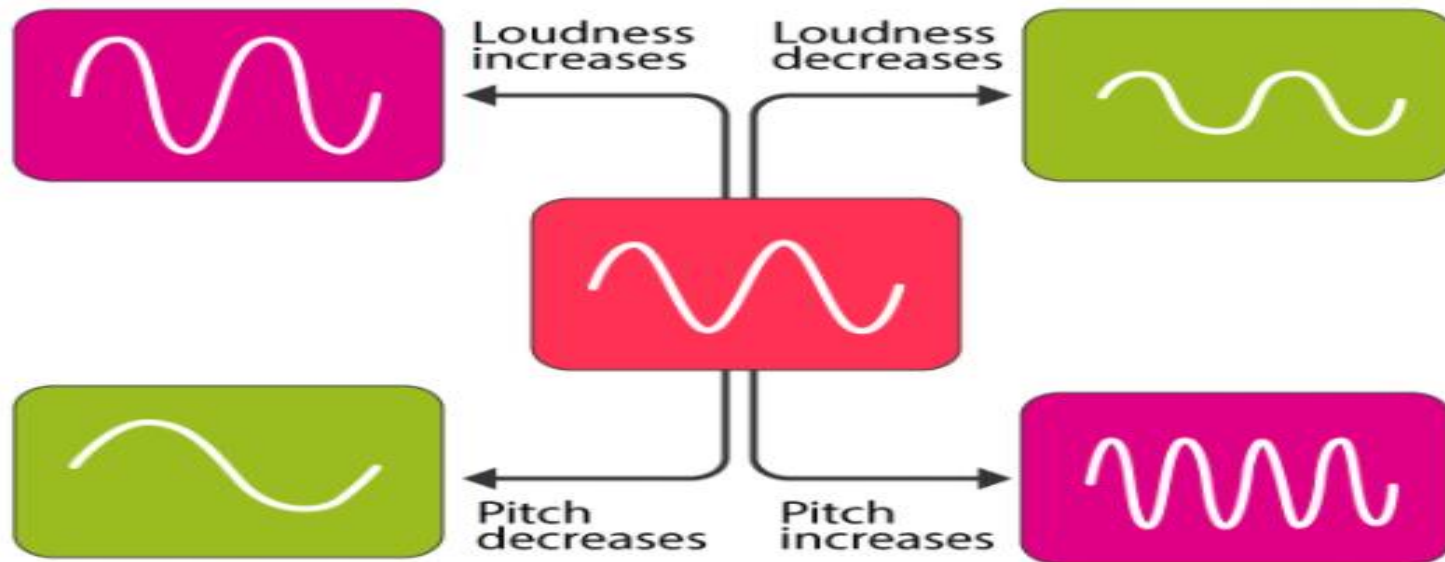
%% speed audio without change pitch
speed_factor=1.5;    %try 0.75,1.5
y = stretchAudio(x,speed_factor);
sound(y,fs);
t = (0:size(y,1)-1)/fs;
plot(t,y)
xlabel('Time (s)')
ylabel('Amplitude')
title('Modified Signal, Speedup Factor = 2')
axis tight
grid on

```



Audio time stretching v.s pitch scaling

- why every person has a different voice?
- Time stretching is the process of changing the speed/ duration of audio signal without affecting its pitch.
- phase vocoder is a way of stretching the length of a signal without affecting the pitch .



Echo effect

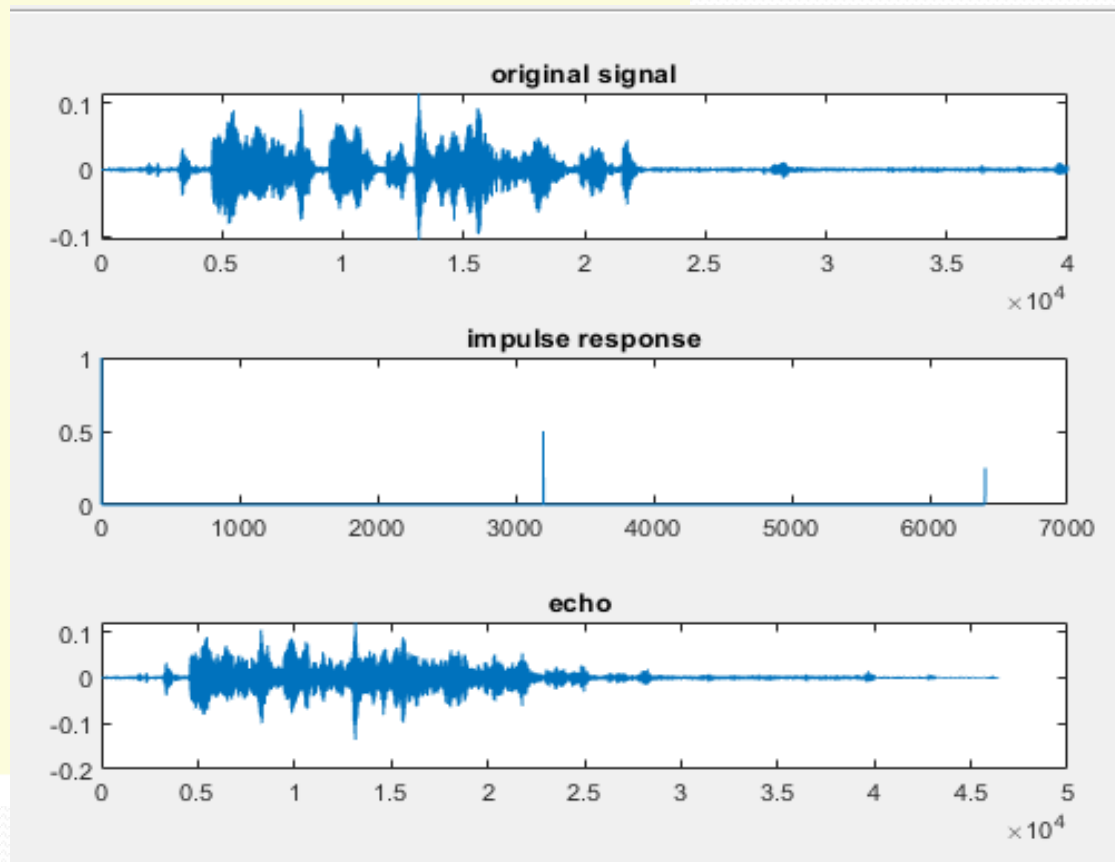
```
%% echo effect using convolution
clear; close all; clc;
%to read specific voice note
[y,fs]=audioread('H:\samar partition\DSP\matlab_current folder\Q2.wav');
h=[1,zeros(1,0.4*fs),0.5,zeros(1,0.4*fs),0.25];
%h=[1,zeros(1,0.4*fs),0.5];
echo=conv(y,h);

figure;
subplot(3,1,1);
plot(y);
title('original signal');

subplot(3,1,2);
plot(h);
title('impulse response');

subplot(3,1,3);
plot(echo);
title('echo ');

sound(echo,fs)
```



Electrocardiogram (ECG)

- Electrocardiography is the interpretation of the electrical activity of the heart over a period of time, as detected by electrodes attached to the surface of the skin and recorded by a device external to the body. The recording produced by this noninvasive procedure is an electrocardiogram (ECG).

Figure 1.1: A typical ECG tracing of the cardiac cycle consists of a P wave, a QRS complex, a T wave and a U wave.

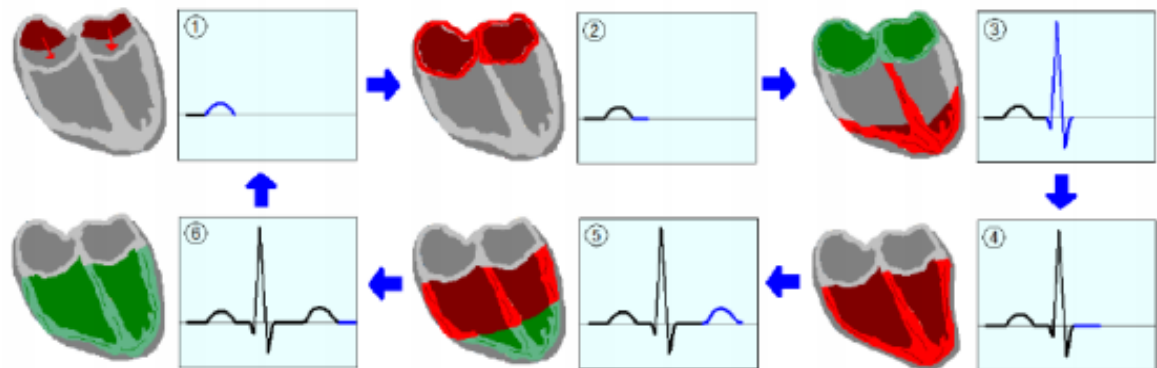
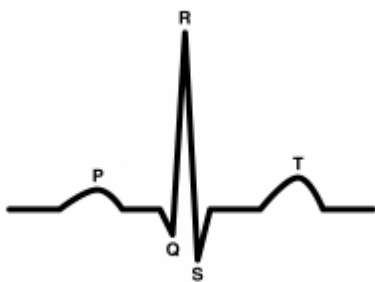
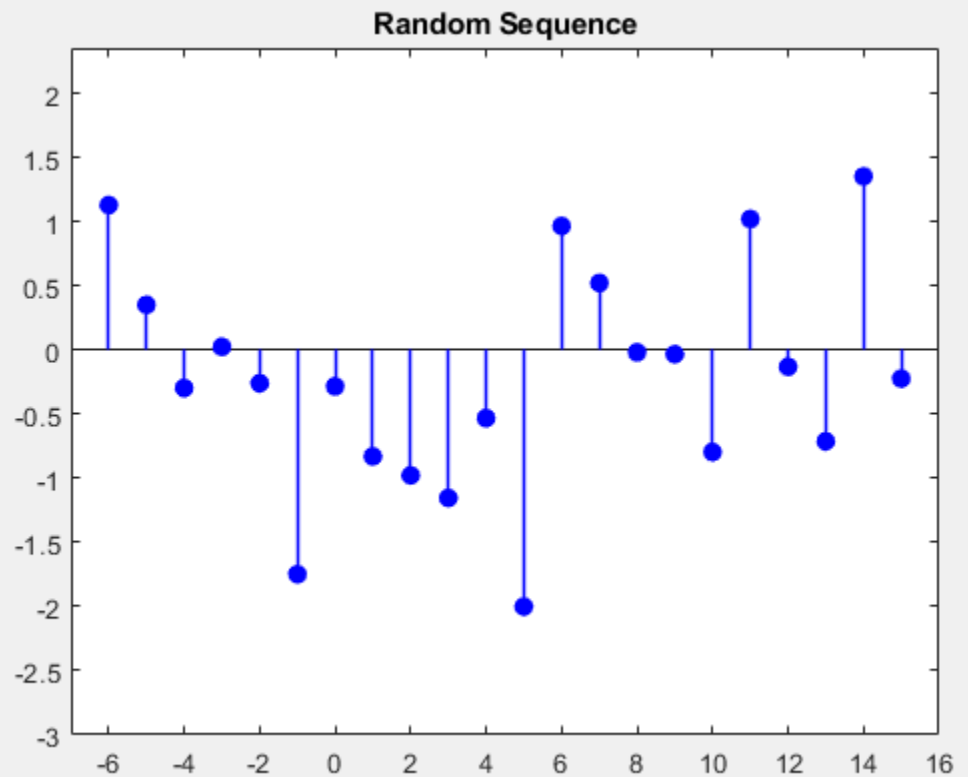


Figure 1.2: Electrical activity in myocardium

Generate ECG signal

- Random ECG signal

```
% random ECG signal  
  
clear; close all; clc;  
k = -6:15;  
y = randn(size(k));  
figure(); stem(k,y,'b','fill','LineWidth',1.2);  
axis([-7 16 min(y)-1 max(y)+1]);  
title('Random Sequence');
```

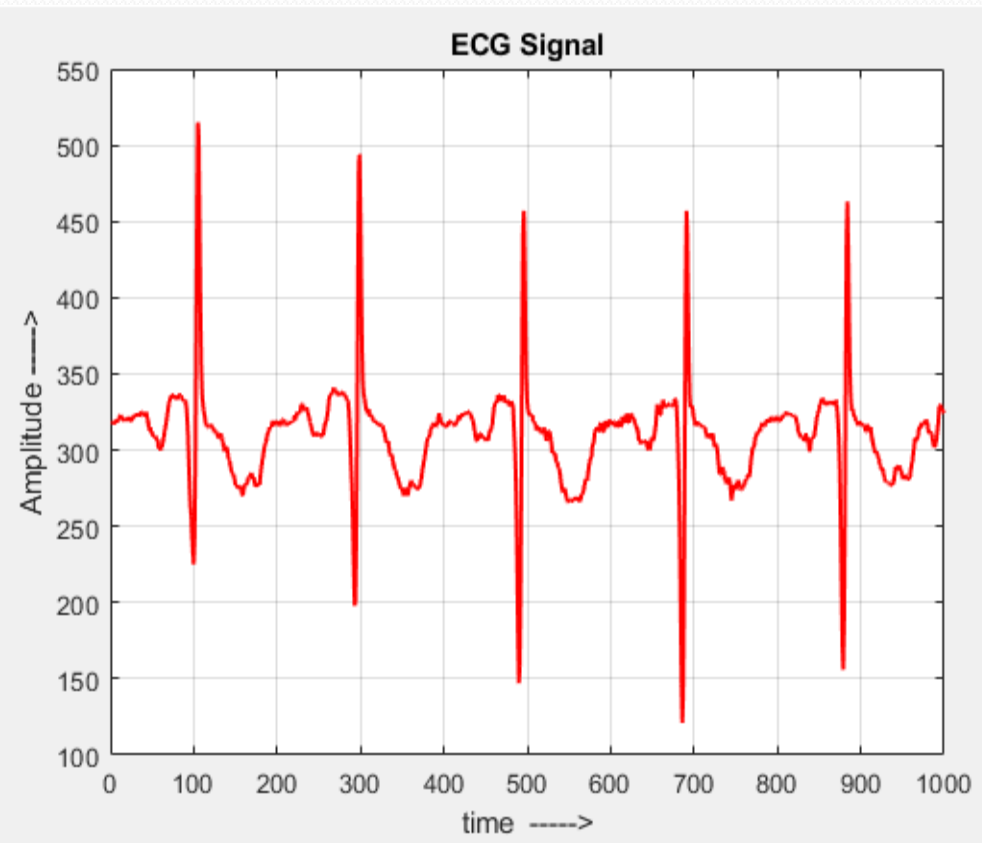


Load ECG signal

- Load ECG signal

```
%% load ECG signal

clear; close all; clc;
load ecg_data.mat
t = 1:1000;
figure(1); plot(t,x,'r','LineWidth',1.5);
xlabel('time ----->');ylabel('Amplitude ----->')
title('ECG Signal');grid on;
```



Z-transform

```
>> syms xx x y n  
>> xx=cos(n)
```

```
xx =
```

```
cos(n)
```

```
>> x=ztrans(xx)
```

```
x =
```

```
(z*(z - cos(1)))/(z^2 - 2*cos(1)*z + 1)
```

```
>> y=iztrans(x)
```

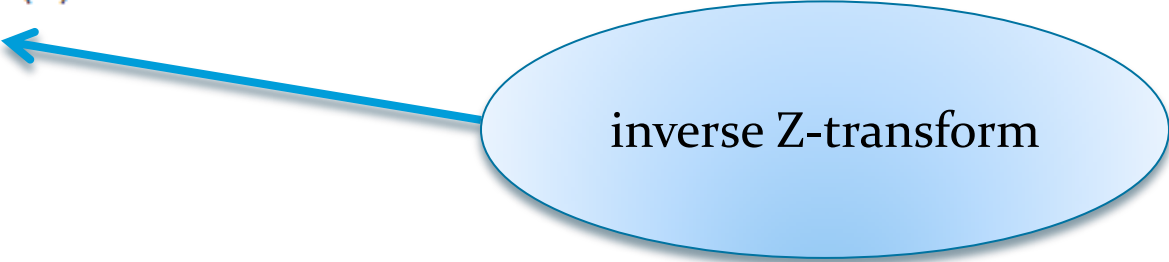
```
y =
```

```
cos(n)
```

Z-transform



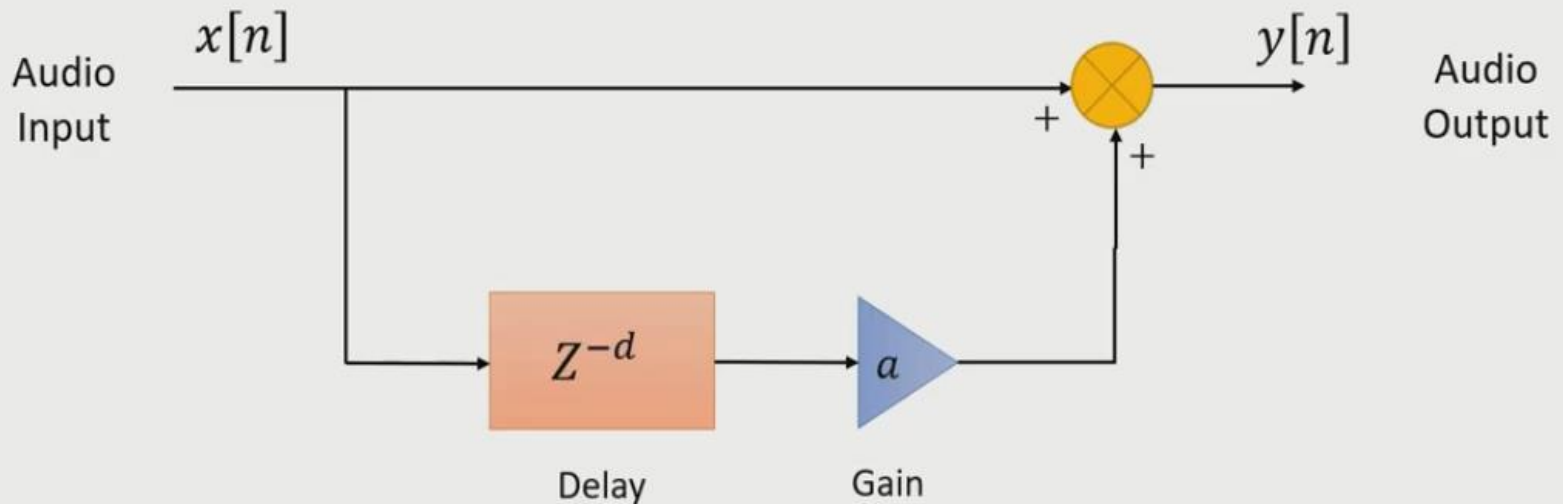
inverse Z-transform



```
%% z-transform
clc; close all; clear;
x=[9 5 4 2 1];
b=0;
n=length(x);
y=sym('z');
for i=1:n
    b=b+x(i)*y^(1-i);
end
disp(b);
```

Assignment

- Implement this :



$$y[n] = x[n] + a \cdot x[n - d]$$