DSP Matlab Code

- *Sine wave*

```
t = 0:1/1000:5/50
y1 = sin (2*pi*50*t)
subplot (2,1,1)
plot (t,y1)
subplot (2,1,2)
stem (t,y1)
```

- *Cos wave*

```
t = 0:1/500:5/50
y = cos (2*pi*50*t)
subplot (2,1,1)
plot (t,y)
subplot (2,1,2)
stem (t,y)
```

- *Unit-step*

```
t = -1:1/10:5
unitstep = t>=0;
subplot (2,1,1)
plot (t,unitstep)
subplot (2,1,2)
stem (t,unitstep)
```

- *Ramp*

```
t = -1:1/5:5
unitstep = t>=0;
ramp = t*unitstep
subplot (2,1,1)
plot (t,ramp)
subplot (2,1,2)
stem (t,ramp)
```

- LAB-3

```
figure(1);

%show img
im1=imread('image.jpg');
subplot(2,1,1)
```

```
imshow(im1)
title("Image")

%rgb2gray
gray=rgb2gray(im1);
subplot(2,1,2)
imshow(gray)
title("RGB to Gray")

figure(2);

% image resize
resized= imresize(gray,[256,256])
imshow(resized)
title("Resized")


figure(3);
% apply guassion filter
sigma=3
filtered=imgaussfilt(gray,sigma)
imshow(filtered)
title("filtered")

figure(4);
%edge detection
edges=edge(gray,'sobel')
imshow(edges)
title("edges")
```

- linear convolution

```
%linear convolution

clc;
clear all;
close all;
x=input ('Enter first sequence')
h=input ('Enter second sequence')
y=conv(x,h)
subplot(3,1,1)

stem(x)
xlabel('time')
```

```
ylabel('amplitude')
title('first input sequence')
subplot(3,1,2)

stem(h)
xlabel('time')
ylabel('amplitude')
title('second input sequence')
subplot(3,1,3)

stem(h)
xlabel('time')
ylabel('amplitude')
title('linear convolution')
```

- **Filters**

```
%filter

Fs=1000;
N=4;
Wc_lp=200/(Fs/2); %cutoff frequency for low pass filter
Wc_hp=400/(Fs/2); %cutoff frequency for high pass filter
W1_bp=200/(Fs/2); %lower cutoff freq for band pass filter
W2_bp=400/(Fs/2); %uper cutoff freq for band pass filter
W1_bs=200/(Fs/2); %lower cutoff freq for band stop filter
W2_bs=400/(Fs/2); %uper cutoff freq for band stop filter
Rp=1;
phi_ap = pi/2; %phase shift for all pass filter

%low pass filter

[b,a]=butter(N,Wc_lp,'low');
[h,w]=freqz (b,a);
figure;

subplot(2,1,1);
plot(w/(2*pi)*Fs, 20*log10(abs(h)));
title('Magnitude Response of Low-Pass Filter');
xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)')

subplot(2,1,2);
plot(w/(2*pi)*Fs,angle(h));
title('Phase Response of low-Pass Filter');
```

```
xlabel('Frequency (Hz)');
ylabel('phase (rad)')

%high pass filter

[b,a]=butter(N,Wc_hp,'high');
[h,w]=freqz (b,a);
figure;

subplot(2,1,1);
plot(w/(2*pi)*Fs, 20*log10(abs(h)));
title('Magnitude Response of high-Pass Filter');
xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)')

subplot(2,1,2);
plot(w/(2*pi)*Fs,angle(h));
title('Phase Response of high-Pass Filter');
xlabel('Frequency (Hz)');
ylabel('phase (rad)')

%Band pass filter

[b,a]=butter(N,[W1_bp,W2_bp],'bandpass');
[h,w]=freqz (b,a);
figure;

subplot(2,1,1);
plot(w/(2*pi)*Fs, 20*log10(abs(h)));
title('Magnitude Response of Band-Pass Filter');
xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)')

subplot(2,1,2);
plot(w/(2*pi)*Fs,angle(h));
title('Phase Response of Band-Pass Filter');
xlabel('Frequency (Hz)');
ylabel('phase (rad)')

%Band Stop Filter

[b,a]=butter(N,[W1_bp,W2_bp],'stop');
[h,w]=freqz (b,a);
figure;
```

```
subplot(2,1,1);
plot(w/(2*pi)*Fs, 20*log10(abs(h)));
title('Magnitude Response of Band-stop Filter');
xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)')

subplot(2,1,2);
plot(w/(2*pi)*Fs,angle(h));
title('Phase Response of Band-stop Filter');
xlabel('Frequency (Hz)');
ylabel('phase (rad)')
```

- **DFT**

```
xn=input('Input sequence: ');
N = input('Enter the number of points: ');
Xk=calcdft(xn,N);
disp('DFT X(k): ');
disp(Xk);
mgXk = abs(Xk);
phaseXk = angle(Xk);
k=0:N-1;
subplot (2,1,1);
stem(k,mgXk);
title ('DFT sequence: ');
xlabel('Frequency');
ylabel('Magnitude');
subplot(2,1,2);
stem(k,phaseXk);
title('Phase of the DFT sequence');
xlabel('Frequency');
ylabel('Phase');

function[Xk] = calcdft(xn,N)
    L=length(xn);
    if(N<L)
        error('N must be greater than or equal to L!!')
    end
    x1=[xn, zeros(1,N-L)];
    for k=0:1:N-1
```

```
        for n=0:1:N-1
            p=exp(-i*2*pi*n*k/N);
            W(k+1,n+1)=p;
        end
    end
    disp('Transformation matrix for DFT')
    disp(W);
    Xk=W*(x1.')
End
```

- **FFT**

```
% Sample sequence
sequence = [1, 2, 3, 4, 5, 6, 7, 8];

% Compute the FFT
fft_sequence = fft(sequence);

% Compute the magnitude response
magnitude = abs(fft_sequence);

% Compute the phase response
phase = angle(fft_sequence);

% Plotting the magnitude response
subplot(2, 1, 1);
stem(magnitude);
title('Magnitude Response');
xlabel('Frequency');
ylabel('Magnitude');

% Plotting the phase response
subplot(2, 1, 2);
stem(phase);
title('Phase Response');
xlabel('Frequency');
ylabel('Phase');

% Introduce aliasing error by downsampling the sequence
downsampled_sequence = downsample(sequence, 2);
```

```
% Compute the FFT of the downsampled sequence
fft_downsampled = fft(downsampled_sequence);

% Plotting the magnitude response of the downsampled sequence
figure;
stem(abs(fft_downsampled));
title('Magnitude Response of Downsampled Sequence');
xlabel('Frequency');
ylabel('Magnitude');
```

- **ECG**

```
% Generate synthetic ECG signal
fs = 1000; % Sampling frequency (Hz)
T = 1/fs; % Sampling period (s)
t = 0:T:1-T; % Time vector (s)
f1 = 1; % Frequency of the first component (Hz)
f2 = 50; % Frequency of the second component (Hz)
ecg_signal = sin(2*pi*f1*t) + 0.5*sin(2*pi*f2*t); % Synthetic ECG signal

% Apply DFT
dft = fft(ecg_signal);

% Apply FFT
fft_signal = fftshift(fft(ecg_signal));

% Frequency axis for plotting
N = length(ecg_signal); % Number of samples
f = (-N/2:N/2-1)*(fs/N);

% Calculate magnitude and phase spectra
magnitude = abs(fft_signal);
phase = angle(fft_signal);

% Plot the original ECG signal
figure;
subplot(3,1,1);
plot(t, ecg_signal);
xlabel('Time (s)');
ylabel('Amplitude');
title('Original ECG Signal');

% Plot the magnitude spectrum
subplot(3,1,2);
```

```
plot(f, magnitude);
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Magnitude Spectrum');

% Plot the phase spectrum
subplot(3,1,3);
plot(f, phase);
xlabel('Frequency (Hz)');
ylabel('Phase');
title('Phase Spectrum');

% Show the plots
sgtitle('DFT and FFT of ECG Signal');
```

- **Circular convolution**

```
a=input ('Enter first sequence')
b=input ('Enter second sequence')
c=cconv(a,b,4);
subplot(3,1,1)

stem(a)
xlabel('time')
ylabel('amplitude')
title('first sequence')
subplot(3,1,2)

stem(b)
xlabel('time')
ylabel('amplitude')
title('second sequence')
subplot(3,1,3)

stem(c)
xlabel('time')
ylabel('amplitude')
title('circular convolution')
```