LABORATORY WORK BOOK

For Academic Session _______

Semester _____

SIGNALS AND SYSTEMS

(TC-204)

For

SE (TC)

Name: ____

Roll Number: ____

Batch: ____

Department: ____

Year/Semester: ____

Department of Electronic Engineering
NED University of Engineering & Technology, Karachi

LABORATORY WORK BOOK

For The Course

TC-204 SIGNALS AND SYSTEMS

Prepared By:

Ms. Saima Athar (Lecturer)

Ms. Nida Nasir (Assistant Prof.)

Reviewed By

Dr. Irfan Ahmed (Associate Professor)

Approved By:

Board of Studies of Department of Electronic Engineering

INTRODUCTION

This laboratory manual contains exercises based on MATLAB and EV kits. The purpose of these exercises is to help reestablish what is and how to points of view on signals and systems. The exercises integrate the basic concepts for both continuous-time and discrete-time signals and systems. This laboratory manual focuses on an imperative style, where signals and systems are constructed procedurally.

MATLAB distributed by The Math Works, Inc., is chosen as the basis for these exercises because it is widely used by practitioners in the field, and because it is capable of realizing interesting systems.

The exercises in this manual cover many of the properties of linear time-invariant (LTI) systems. It provides an introduction to the basic concepts involved in using MATLAB to represent signals and systems. Also, the necessary tools for dealing with both numerical and symbolic expressions are learned.

The manual covers a variety of exercises that includes signal and system representations for both time and frequency domains, basic properties of signals and systems, the processing of signals by linear systems, Fourier series and transforms, discrete-time processing of continuous-time signals.

The lab exercises introduced the fundamental ideas of signal and system analysis that will help the students to understand the engineering systems in many diverse areas, including seismic data processing, communications, speech processing, image processing, and defense electronics.

CONTENTS

LAB SESSION 01

OBJECT:-

Simulate the signals (sinusoidal, impulse, ramp and step signals) in Matlab.

THEORY:-

Sinusoids are the building block of analog signal processing. All real world signals can be represented as an infinite sum of sinusoidal functions via a Fourier series. A sinusoidal function can be represented in terms of an exponential by the application of Euler's Formula.

An impulse (Dirac delta function) is defined as a signal that has an infinite magnitude and an infinitesimally narrow width with an area under it of one, centered at zero. An impulse can be represented as an infinite sum of sinusoids that includes all possible frequencies. It is not, in reality, possible to generate such a signal, but it can be sufficiently approximated with large amplitude, narrow pulse, to produce the theoretical impulse response in a network to a high degree of accuracy. The symbol for an impulse is δ(t). If an impulse is used as an input to a system, the output is known as the impulse response. The impulse response defines the system because all possible frequencies are represented in the input

A unit step function is a signal that has a magnitude of zero before zero and a magnitude of one after zero. The symbol for a unit step is u(t). If a step is used as the input to a system, the output is called the step response. The step response shows how a system responds to a sudden input, similar to turning on a switch. The period before the output stabilizes is called the transient part of a signal. The step response can be multiplied with other signals to show how the system responds when an input is suddenly turned on.

The unit step function is related to the Dirac delta function by;

$$\mathrm{u}(t) = \int_{-\infty}^{t} \delta(s)ds$$

PROCEDURE & OBSERVATION:-

Matlab code for discrete sinusoidal signal generation:

```
clc;
clear all;
close all;
disp('Sinusoidal Signal generation');
N=input('Enter no of samples: ');
n=0 : 0.1 : N;
x=sin(n);
figure, stem(n,x);
xlabel('Samples');
ylabel('Amplitude');
title('Sinusoidal Signal');
```

The sin(n) function returns an array which corresponds to sine value of the array ‘n’.

Matlab code for unit impulse signal generation:

```
clc;
clear all;
close all;
disp('Unit Impulse Signal Generation');
N = input('Enter no of samples: ');
n = -N : 1 : N;
x = [zeros(1,N),1,zeros(1,N)];
stem(n,x);
xlabel('Sample');
ylabel('Amplitude');
title('Unit Impulse Signal');
```

In this code, the impulse is generated by using ZEROS(x,y) function, which produces an array of size X,Y with all elements as ZERO.

Matlab code for unit ramp signal generation:

```
clc;
clear all;
close all;
disp('Unit Ramp Signal Generation');
N = input('Enter no of samples: ');
a = input(' Max Amplitude: ');
n = -N : 1 : N;
x =  a*n/N;
stem(n,x);
xlabel('Sample');
ylabel('Amplitude');
title('Unit Ramp Signal');
```

Matlab code for unit step (delayed step) signal generation:

```
clc;
clear all;
close all;
disp('Delayed Unit Step Signal Generation');
N = input('Enter no of samples: ');
d = input('Enter  delay  value: ');
n = -N : 1 : N;
x = [zeros(1,N+d),ones(1,N-d+1)];
stem(n,x);
xlabel('Sample');
ylabel('Amplitude');
title('Delayed Unit Step Signal');
```

LAB SESSION 02

OBJECT:-

Fourier synthesis of square wave in Matlab

THEORY:-

A square wave spectrum is made of the sum of all the harmonics being odd of the fundamental with decreasing amplitude according to the law of trigonometric Fourier series. In other words the square wave shown in fig 2.1 can be obtained by summing up the infinite sine waves as per the following relation:

$$S(t) = \sin(2\Pi Ft)/1 + \sin(2\Pi 3Ft)/3 + \sin(2\Pi 5Ft)/5 + \sin(2\Pi 7Ft)/7 + \sin(2\Pi 9Ft)/9 + \ldots\ldots..$$

PROCEDURE AND OBSERVATIONS:-

```
% Fourier Synthesis of Square Wave.

tt=5000; %Total Simulation Run
T=500;   %Time period of sine component
out=zeros(1,tt);

% Square wave synthesis equation. You can add and delete harmonics in this equation.

for t=1:1:tt
s= sin(2*pi*t/T)+(1/3)*(sin(2*3*pi*t/T))+
(1/5)*(sin(2*5*pi*t/T))+(1/7)*(sin(2*7*pi*t/T))+(1/9)*(sin(2*9*pi*t/T));
out(t)=s;
end

plot(1:tt,out);
xlabel('Time')
ylabel('Amplitude')
title('Fourier Synthesis of Square Wave')
```

LAB SESSION 03

OBJECT:-

Fourier synthesis of a triangular wave in Matlab

THEORY:-

A triangular wave spectrum is made of the sum of all the harmonics being odd of the fundamental with decreasing amplitude according to the law of trigonometric Fourier series. In other words the triangular wave can be obtained by summing up the infinite sine waves as per the following relation:

$S(t) = \cos(2\Pi Ft)/1 + \cos(2\Pi 3Ft)/3^2 + \cos(2\Pi 5Ft)/5^2 + \cos(2\Pi 7Ft)/7^2 + \cos(2\Pi 9Ft)/9^2 + \ldots\ldots$

PROCEDURE AND OBSERVATIONS:-

```
% Fourier Synthesis of Triangular Wave.

tt=5000; %Total time simulation run
T=500;   %Time period of sine component
out=zeros(1,tt);

% Triangular wave synthesis equation. You can add and delete harmonics in this    equation.

for t=1:1:tt
s= cos(2*pi*t/T)+((1/3)^2)*(cos(2*3*pi*t/T))+…      ((1/5)^2)*(cos(2*5*pi*t/T))+((1/7)^2)*…
(cos(2*7*pi*t/T))+((1/9)^2)*(cos(2*9*pi*t/T));
out(t)=s;
end

plot(1:tt,out);
xlabel('Time')
ylabel('Amplitude')
title('Fourier Synthesis of Triangular Wave')
```

LAB SESSION 04

OBJECT:-
To carry out the Fourier synthesis of square wave

EQUIPMENT:-

1 Modules T10H.
1 +/- 12Vdc Supply
1 Oscilloscope.

THEORY:-

A square wave spectrum is made of the sum of all the harmonics being odd of the fundamental with decreasing amplitude according to the law of trigonometric Fourier series. In other words the square wave shown in fig 2.1 can be obtained by summing up the infinite sine waves as per the following relation:

$S(t) = \sin(2\Pi Ft)/1 + \sin(2\Pi 3Ft)/3 + \sin(2\Pi 5Ft)/5 + \sin(2\Pi 7Ft)/7 + \sin(2\Pi 9Ft)/9 + \ldots\ldots..$

PROCEDURE AND OBSERVATIONS:-

1- Odd harmonics (1, 3, 5, 7, 9): two way switches -/0/+ on + and two way switches sin/cos on sin.
2- Even harmonics (2, 4, 6, 8): two way switches -/0/+ on 0.
3- Connect the oscilloscope with the amplifier output of the fundamental (1st) and adjust the amplitude at 10Vp-p.

4- Connect the oscilloscope with the output of the third harmonic amplifier (3RD) and adjust the amplitude at 10/3 ≈ 303Vp-p.

5- Connect the oscilloscope with the output of the 5TH harmonic amplifier (5TH) and adjust the amplitude at 10/5 = 2Vp-p.

6- Connect the oscilloscope with the output of the seventh harmonic amplifier (7TH) and adjust the amplitude at 10/7 ≈ 1.4Vp-p.

7- Connect the oscilloscope with the output of the 9th harmonic amplifier (9TH) and adjust the amplitude at 10/9 ≈ 1.1Vp-p
----

8- Connect the oscilloscope with OUT and check that there is the signal corresponding to the components sum.

9-Remove some harmonics (put the relating two way switch **on 0**) and check the o/p signal.

10- Prove the Fourier series of square wave by using formula

$$f(t) = a0 + \sum_{n=1}^{\infty}(an \cos nwt + bn \sin nwt)$$

LAB SESSION 05

OBJECT:-

To carry out the Fourier synthesis of a triangular wave

EQUIPMENT:-

1 Modules T10H.
1 +/- 12Vdc Supply
1 Oscilloscope.

THEORY:-

A triangular wave spectrum is made of the sum of all the harmonics being odd of the fundamental with decreasing amplitude according to the law of trigonometric Fourier series. In other words the triangular wave can be obtained by summing up the infinite sine waves as per the following relation:

$S(t) = \cos(2\Pi Ft)/1 + \cos(2\Pi 3Ft)/3^2 + \cos(2\Pi 5Ft)/5^2 + \cos(2\Pi 7Ft)/7^2 + \cos(2\Pi 9Ft)/9^2 + \ldots..$

PROCEDURE AND OBSERVATIONS:-

1- Odd harmonics (1, 3, 5, 7, 9): two way switches -/0/+ on + and two way switches sin/cos on cos.
2- Even harmonics (2, 4, 6, 8): two way switches -/0/+ on 0.
3- Connect the oscilloscope with the amplifier output of the fundamental (1_{st}) and adjust the amplitude at 10Vp-p.

4- Connect the oscilloscope with the output of the third harmonic amplifier (3_{RD}) and adjust the amplitude at $10/3^2$

5- Connect the oscilloscope with the output of the 5_{TH} harmonic amplifier (5_{TH}) and adjust the amplitude at $10/5^2$

6- Connect the oscilloscope with the output of the seventh harmonic amplifier (7_{TH}) and adjust the amplitude at $10/7^2$

7- Connect the oscilloscope with the output of the 9_{th} harmonic amplifier (9_{TH}) and adjust the amplitude at $10/9^2$

8- Connect the oscilloscope with OUT and check that there is the signal corresponding to the components sum.

9-Remove some harmonics (put the relating two way switch **on 0**) and check the o/p signal.

10- Prove the Fourier series of triangular wave by using formula

$$f(t) = a0 + \sum_{n=1}^{\infty}(an \cos nwt + bn \sin nwt)$$

LAB SESSION 06

OBJECT:-

To verify the properties of LTI (Linear Time Invariant) systems

THEORY:-

The defining properties of any LTI system are *linearity* and *time invariance*.

- ***Linearity*** means that the relationship between the input and the output of the system is a linear map: If input $x_1(t)$ produces response $y_1(t)$, and input $x_2(t)$ produces response $y_2(t)$, then the *scaled* and *summed* input $a_1x_1(t) + a_2x_2(t)$ produces the scaled and summed response $a_1y_1(t) + a_2y_2(t)$ where a_1 and a_2 are real scalars. It follows that this can be extended to an arbitrary number of terms, and so for real numbers $c_1, c_2, \ldots, c_k$,

Input $\sum_k c_k\, x_k(t)$ produces output $\sum_k c_k\, y_k(t).$

In particular,

Input $\int_{-\infty}^{\infty} c_\omega\, x_\omega(t)\, \mathrm{d}\omega$ produces output $\int_{-\infty}^{\infty} c_\omega\, y_\omega(t)\, \mathrm{d}\omega$

where c_ω and x_ω are scalars and inputs that vary over a continuum indexed by ω. Thus if an input function can be represented by a continuum of input functions, combined "linearly", as shown, then the corresponding output function can be represented by the corresponding continuum of output functions, *scaled* and *summed* in the same way.

- ***Time invariance*** means that whether we apply an input to the system now or *T* seconds from now, the output will be identical except for a time delay of the *T* seconds. That is, if the output due to input $x(t)$ is $y(t)$, then the output due to input $x(t-T)$ is $y(t-T)$. Hence, the system is time invariant because the output does not depend on the particular time the input is applied.

Any LTI system can be characterized entirely by a single function called the system's impulse response. The output of the system is simply the convolution of the input to the system with the system's impulse response. This method of analysis is often called the time domain point-of-view.

The same result is true of discrete-time linear shift-invariant systems in which signals are discrete-time samples, and convolution is defined on sequences.

The response of linear time invariant system can be computed using the convolution integral by convolving the input *x*(*t*) with the impulse response *h*(*t*) to generate the response or output *y*(*t*).

PROCEDURE & OBSERVATION:-

```
% Proof of LTI system properties

% Linearity

t=-1.2:0.0001:1.2;
x=zeros(size(t));
h=zeros(size(t));
x(t>=-1 & t<=1)=1;
subplot(6,1,1);
plot(t,x);ylabel('x(t)');
h(t>=-1 & t<=1)=1;
subplot(6,1,2);
plot(t,h);
ylabel('h(t)');
y=conv(x,h);
tt=t(1)+t(1):0.0001:t(end)+t(end);
subplot(6,1,3);
plot(tt,y*0.0001);
ylabel('y(t)');

% Addition property

x2=x+x;
ya=conv(x2,h);
tt=t(1)+t(1):0.0001:t(end)+t(end);
subplot(6,1,4);
plot(tt,ya*0.0001);
yb=y+y;
hold on
plot(tt,yb*0.0001,'r+');
ylabel('Addition property');

% Scaling property

xs=2*x;
ya=conv(xs,h);
tt=t(1)+t(1):0.0001:t(end)+t(end);
subplot(6,1,5);
plot(tt,ya*0.0001);
yb=2*y;
hold on
plot(tt,ya*0.0001,'r+');
ylabel('Scaling property');
```

```
% Time-invariance

t=-1.2:0.0001:1.2;
x1=zeros(size(t));
x1(t>=-0.8 & t<=1.2)=1;
y1=conv(x1,h);
tt=t(1)+t(1):0.0001:t(end)+t(end);
subplot(6,1,6);
plot((1:length(y1))*0.0001,y1*0.0001);
hold on;
pad=zeros(1,0.2/0.0001);
y2=[pad y];
plot((1:length(y2))*0.0001,y2*0.0001,'r+');
ylabel('Time-invariance property');
```

LAB SESSION 07

OBJECT:- Computation of Fourier transform.

THEORY:-

The Fourier transform is one of the most useful mathematical tools for many fields of science and engineering. The Fourier transform has applications in signal processing, physics, communications, geology, astronomy, optics, and many other fields. This technique transforms a function or set of data from the time or sample domain to the frequency domain. This means that the Fourier transform can display the frequency components within a time series of data. The Discrete Fourier Transform (DFT) transforms discrete data from the sample domain to the frequency domain. The Fast Fourier Transform (FFT) is an efficient way to do the DFT, and there are many different algorithms to accomplish the FFT. Matlab uses the FFT to find the frequency components of a discrete signal.

This lab shows how to use the FFT to analyze an audio file in Matlab. The audio file is provided with the lab. This shows how the Fourier transform works and how to implement the technique in Matlab.

PROCEDURE & OBSERVATION:-

```
% Fourier Transform of Audio File
% Load File
file = 'C:\MATLAB\work\audio_file';
[y,Fs,bits] = wavread(file);
Nsamps = length(y);
t = (1/Fs)*(1:Nsamps)          %Prepare time data for plot

% Do Fourier Transform
y_fft = abs(fft(y));            %Retain Magnitude
y_fft = y_fft(1:Nsamps/2);      %Discard Half of Points
f = Fs*(0:Nsamps/2-1)/Nsamps;   %Prepare freq data for plot

% Plot Audio File in Time Domain
figure
plot(t, y)
xlabel('Time (s)')
ylabel('Amplitude')
title('audio_file')

% Plot Audio File in Frequency Domain
figure
plot(f, y_fft)
xlim([0 1000])
xlabel('Frequency (Hz)')
ylabel('Amplitude')
title('Frequency Response of audio_file')
```

The audio file is opened using the wavread function, which returns the sampled data from the file, the sampling frequency, and the number of bits used in the A/D converter. Note that the file extension ".wav" does not have to be specified in the function call. The sampling frequency is important for interpreting the data.

The FFT is performed using the "fft" function. Matlab has no "dft" function, as the FFT computes the DFT exactly. Only the magnitude of the FFT is saved, although the phase of the FFT is useful is some applications. The "fft" function allows the number of points outputted by the FFT to be specified, but in this lab, we will use the same number of input and output points. In the next line, half of the points in the FFT are discarded. This is done for the purposes of this lab, but for many applications, the entire spectrum is interesting. In the following line, the data that will be used for the abscissa is prepared by using the sampling frequency and the number of samples in the time domain. This step is important to determine the actual frequencies contained in the audio data.

Next, the original data are plotted in the time domain and the FFT of the data is plotted. The x-axis is limited to the range [0, 1000] in the plot to show more detail at the peak frequency.

The Fourier transform is a useful tool in many different fields. Two-dimensional Fourier transforms are often used for images as well. Try the code above for yourself.

LAB SESSION 08

OBJECT:-

Representation of LTI systems in Matlab.

THEORY:-

Any LTI system can be characterized entirely by a single function called the system's impulse response. The output of the system is simply the convolution of the input to the system with the system's impulse response. This method of analysis is often called the *time domain* point-of-view. The same result is true of discrete-time linear shift-invariant systems in which signals are discrete-time samples, and convolution is defined on sequences.

Equivalently, any LTI system can be characterized in the *frequency domain* by the system's transfer function, which is the Laplace transform of the system's impulse response (or Z transform in the case of discrete-time systems). As a result of the properties of these transforms, the output of the system in the frequency domain is the product of the transfer function and the transform of the input. In other words, convolution in the time domain is equivalent to multiplication in the frequency domain.

LTI system theory is good at describing many important systems. Most LTI systems are considered "easy" to analyze, at least compared to the time-varying and/or nonlinear case. Any system that can be modeled as a linear homogeneous differential equation with constant coefficients is an LTI system. Examples of such systems are electrical circuits made up of resistors, inductors, and capacitors (RLC circuits). Ideal spring–mass–damper systems are also LTI systems, and are mathematically equivalent to RLC circuits.

PROCEDURE & OBSERVATION:-

```
% This simulation show time response of first order circuit
R = 10e3;
C = 1e-6;
num = 1;
den = [R*C 1];
[z,p,k] = tf2zp(num,den)
[mag,phase]=bode(sys);
sys=tf(num,den);
[mag,phase]=bode(sys);

% IR response of circuit
ir_system=impulse(sys);
plot(ir_systm)
t = 0:.001:.1;
Vin=t<0.05;

% Response of circuit for square wave in time domain
lsim(num,den,Vin,t);
```

LAB SESSION 09

OBJECT:-

Compute and plot the frequency response of an LTI system using the Laplace transform.

THEORY:-

The Laplace transform is used for solving differential and integral equations. In physics and engineering it is used for analysis of linear time-invariant systems such as electrical circuits, harmonic oscillators, optical devices, and mechanical systems. In such analyses, the Laplace transform is often interpreted as a transformation from the time-domain, in which inputs and outputs are functions of time, to the frequency-domain, where the same inputs and outputs are functions of complex angular frequency, in radians per unit time. Given a simple mathematical or functional description of an input or output to a system, the Laplace transform provides an alternative functional description that often simplifies the process of analyzing the behavior of the system, or in synthesizing a new system based on a set of specifications.

PROCEDURE & OBSERVATION:-

```
clear all;
close all;

% a numerator coefficients of transfer function
% b denominator coefficients of transfer function
% z is zero
% p is pole
% k is gain

% First Order High pass
b=[1 10];
a=[1 1000];
w = logspace(-1,4);
h = freqs(b,a,w);
mag = abs(h);
phase = angle(h);

subplot(2,1,1), loglog(w,mag)
subplot(2,1,2), semilogx(w,phase)

% First Order Low pass
b=[1 1000];
a=[1 10];
w = logspace(-1,4);
h = freqs(b,a,w);
mag = abs(h);
phase = angle(h);
figure
subplot(2,1,1), loglog(w,mag)
subplot(2,1,2), semilogx(w,phase)
```

```
% Second order low pass transfer function
b=[0 -10];
a=[1 1/20 100];
w = logspace(-1,4);
h = freqs(b,a,w);
mag = abs(h);
phase = angle(h);
subplot(2,1,1), loglog(w,mag)
subplot(2,1,2), semilogx(w,phase)

% Second order High pass transfer function
b=[1 0 0];
a=[1 1/20 100];
w = logspace(-1,4);
h = freqs(b,a,w);
mag = abs(h);
phase = angle(h);
subplot(2,1,1), loglog(w,mag)
subplot(2,1,2), semilogx(w,phase)

% Second order Band pass transfer function
b=[-1 0];
a=[1 1/20 100];
w = logspace(-1,4);
h = freqs(b,a,w);
mag = abs(h);
phase = angle(h);
subplot(2,1,1), loglog(w,mag)
subplot(2,1,2), semilogx(w,phase)

% Finding Zero and poles from transfer function
b=[-1 0];
a=[1 1/20 100];
[z,p,k] = tf2zp(b,a);

z
p
k

% Finding Transfer function from zero and pole

k=1;
z=[-10 -1000]';
p=[-100 -10000]';
[b,a] = zp2tf(z,p,k);

b
a
```

LAB SESSION 10

OBJECT:-

Determine the I/O characteristic of a low-pass system and calculate the cutting frequency.

EQUIPMENT:-

Modules C 17/EV
+/- 12 V dc Supply
Oscilloscope
Function generator
Digital multimeter

THEORY:-

The low-pass filter is carried out with an operational amplifier as active element and resistors and capacitors as passive elements.

The diagram of a generic filter is the one of figure 16.1 which does not specifies which are the impedance used.
The characteristics of the filter are determined by the impedances used and their value.

The input / output relation for the generic diagram of figure 16.1 is given by

$$\frac{Vo}{Vi} = \frac{Z2.Z4.Z5}{Z1.Z3.Z5 + Z1.Z2.Z5 + Z1.Z2.Z3 + Z2.Z3.Z5 - Z2.Z1.Z4}$$

Where Vo and Vi respectively indicate the output and input voltage.

This relation is obtained supposing that the input impedance of the operational amplifier is infinite and that its input is not crossed by current and that the amplification is infinite. This fact implies that the differential voltage is null and that the inverting input can be considered as virtually grounded.

With these hypothesis and applying the superimposition theorem you can obtain the above relation after a few simple operation.

This filter is low-pass if Z1, Z3 and Z5 must be resistors while Z2 and Z4 are capacitors.

Changing the generic impedances Z1 of the general formula with the characteristic impedance of the components used we obtain the following general relation for the low-pass filter

$$\frac{Vo}{Vi} = \frac{-R3 / (w^2.C1 .C2)}{R1.R2.R3 + \dfrac{R1.R2.R3 + R1.R3 + R1.R2. +R2.R3}{JwC1} + \dfrac{R1}{w^2.C1.C2}}$$

Where W is the Pulse of the input signal.

The cutting frequency of the filter is determined by the value of the passive components. For the value marked in figure 16.2, this frequency is about 66 Hz.

Fig. 16.1

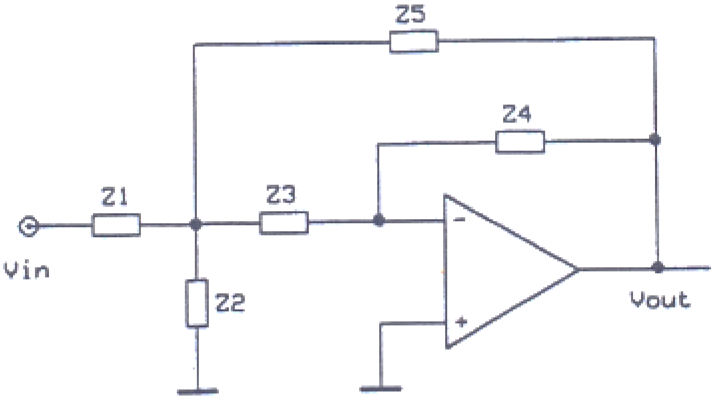
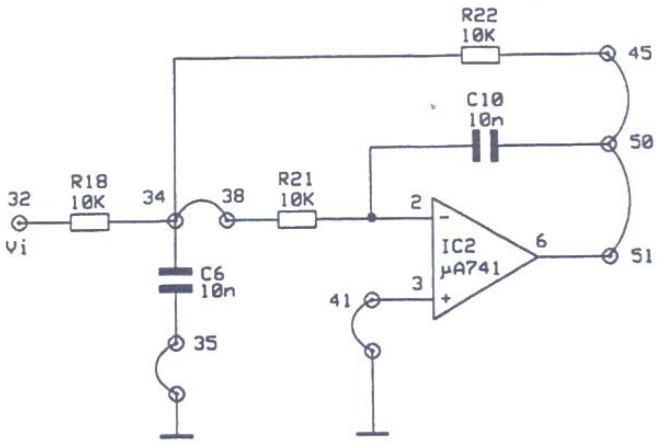


Fig. 16.2



PROCEDURE & OBSERVATION:-

- Carry out the circuit of figure 16.2
- Apply a sine voltage with amplitude of 1 Vpp and frequency of 10 Hz across the circuit.
- Connect one probe of the oscilloscope to the output of the filter and measure the signal amplitude
- Report this value in table 16.1
- Repeat the measurement of the output voltage (and report it in the proper spaces) for the frequencies reported in table 16.1

- Repeat all values obtained in the table and draw the input / output characteristic of the filter connecting all points.
- From the graph above, calculate the cutting frequency of the filter, defined as the frequency at which the amplification drops to 0.707 times the maximum value.

F (Vin)	**Vin**	**Vo**	**Vo / Vin**
10Hz			
20Hz			
30Hz			
40Hz			
50Hz			
60 Hz			
70 Hz			
100Hz			
200Hz			
500Hz			
1 KHz			
2 KHz			

LAB SESSION 11

OBJECT:-

Determine the I/O characteristic of a high-pass system and calculate the cutting frequency.

EQUIPMENT:-

Modules C 17/EV
+/- 12 V dc Supply
Oscilloscope
Function generator
Digital multimeter

THEORY:-

The ideal high-pass filter is a circuit which has a zero amplification for all those signals which frequency is inferior to a certain frequency f' while it has an amplification different from zero and constant for all those signals which frequencies are higher than f'.
In this case as for the low-pass filter, the circuit which carries out the filter is composed by an operational amplifier and by resistors and capacitors.

Obviously, the real circuit which carries out the high-pass filter can have ideal characteristics: it is enough that the transfer function has a behavior near the one of the ideal filter.

In this case too, we start from the relation between input and output. Consider the relation written for the generic filter which here we write as

$$\frac{Vo}{Vi} = \frac{Z2 . Z4 . Z5}{Z1.Z3.Z5 + Z1.Z2.Z5 + Z1.Z2.Z3 + Z2.Z3.Z5 - Z2.Z1.Z4}$$

This filter is high-pass if Z1, Z3 and Z5are capacitors while Z2 and Z4 are resistors.

Changing the values of the general relation above described we obtain.

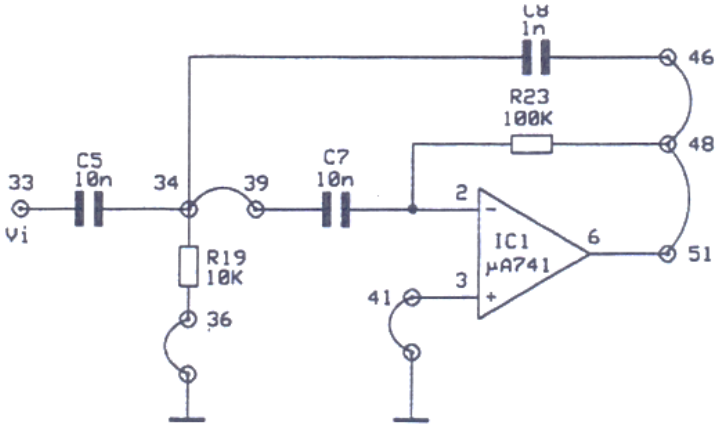
$$\frac{Vo}{Vi} = \frac{R1.R2 / (jwC3)}{\frac{1}{JwC1.jwC2.jwC3} + \frac{R1}{JwC1.jwC3} + \frac{R1}{JwC1.jwC2} + \frac{R1}{JwC2.jwC3} - \frac{R1.R2}{jWC1}}$$

Now the characteristic of the filter, i.e. the cutting frequency and the transfer function will be determined by the value of the passive components used.

PROCEDURE & OBSERVATION:-

1. Carry out the circuit of figure 17.1

Fig. 17.1



2. Apply a sine voltage with amplitude of 1 Vpp and frequency of 1 KHz across the input of the circuit.
3. Connect one probe of the oscilloscope to the output of the filter and measure the amplitude of the signal
4. Fill table which the value. Repeat the output voltage measurement (and report it in the proper spaces) for all the frequencies shown in table.
5. Report the obtain values of figure 17.2 and connect all the points to obtain a graphic of the variation of the output voltage with frequency.
6. From the graph above, calculate the cutting frequency of the filter, defined as the frequency at which the amplification drops to 0.707 times the maximum.

F (Vin)	Vin	Vo	Vo / Vin
1KHz			
2KHz			
5KHz			
10KHz			
20KHz			
50KHz			
100 KHz			
200 KHz			

LAB SESSION 12

OBJECT:-

Determine the I/O characteristic of a band-pass system and calculate the cutting frequency.

EQUIPMENT:-

Modules T 10A-T 10B.
+/- 12 V dc Supply.
Oscilloscope.

THEORY:-

Ceramic Filter:-

A ceramic filter is a band pass filter using a piezoelectric ceramic material. Important parameters of ceramic filter are input & output impedance. The center frequency of ceramic is 455 KHz. The response curve can be obtained by applying a variable freq across i/p & detecting amplitude at o/p. The attenuation measured at different frequency is given by:

$A = Vo/Vi$

$A_{dB} = 20 \log (Vo/Vi)$

PROCEDURE & OBSERVATION:-

1- Supply the modules with dc supply. Carryout the following presetting:
VCO1: switch on 500 KHz, level about 2Vp-p, Freq.450KHz.

2- Apply a signal of 455 KHz corresponding to the central frequency of the ceramic filter.

3- If Vo and Vi are the peak-to-peak voltages measured across the output and the input of the filter. The attenuation A of the filter at 455 KHz is given by.
A=Vo/Vi and dB= 20 log (Vo/Vi)

4- Repeat the measurement carried out in the last step varying the frequency from 445 to 465 KHz at step of 1 KHz.

5- Calculate AdB in correspondence to each frequency and report all in the following table.

FREQUENCY (KHz)	OUT PUT VOLTAGE (Vo)	INPUT VOLTAGE (Vi)	AdB=20log (Vo/Vi)
445			
446			
447			
448			
449			
450			
451			

452			
453			
454			
455			
456			
457			
458			
459			
460			
461			
462			
463			
464			

6-With the data in the table plot a graph setting A_{dB} on the Y axis and frequency on the x-axis, you obtain the frequency response curve of the filter.

LAB SESSION 13

OBJECT:-

Simulate and plot the time response of LTI systems to arbitrary inputs.

THEORY:-

Any LTI system can be characterized entirely by a single function called the system's impulse response. The output of the system is simply the convolution of the input to the system with the system's impulse response. This method of analysis is often called the *time domain* point-of-view. The same result is true of discrete-time linear shift-invariant systems in which signals are discrete-time samples, and convolution is defined on sequences.

Equivalently, any LTI system can be characterized in the *frequency domain* by the system's transfer function, which is the Laplace transform of the system's impulse response (or Z transform in the case of discrete-time systems). As a result of the properties of these transforms, the output of the system in the frequency domain is the product of the transfer function and the transform of the input. In other words, convolution in the time domain is equivalent to multiplication in the frequency domain.

PROCEDURE & OBSERVATION:-

```
clear all;
close all;

num = [1 0 1];
den = [2 3 1];
sys = tf(num, den);

[u1, t1] = gensig('square', 2, 10, 0.1);
[u2, t2] = gensig('sin', 2, 10, 0.1);
[u3, t3] = gensig('pulse', 2, 10, 0.1);

lsim(sys, u1, t1);
lsim(sys, u2, t2);
lsim(sys, u3, t3);
```

It should be noted that in order to actually plot the time response, Matlab always converts the continuous-time output signal into discrete-time signal by sampling at the “appropriate” sampling frequency.
We can actually simulate the system for any input by using the lsim command, which plots both the input (gray color) and the output (blue).

LAB SESSION 14

OBJECT:-

Analysis of LTI systems using z-transform.

THEORY:-

The **Z-transform** converts a time domain signal, which is a sequence of real or complex numbers, into a complex frequency domain representation. It can be considered as a discrete-time equivalent of the Laplace transform.

Due to its convolution property, the z-transform is a powerful tool to analyze LTI systems.

$$y[n] = h[n] * x[n] \xrightarrow{Z} Y(z) = H(z)X(z)$$

PROCEDURE & OBSERVATION:-

```
% Computation of z-transform
 syms n;
f = n^4;
ztrans(f)
syms a z;
g = a^z;
ztrans(g)
syms a n w;
f = sin(a*n);
ztrans(f, w)

% For data sampled at 1000 Hz, plot the poles and zeros of a
% 4th-order elliptic low pass digital filter with cutoff frequency of 200 Hz,
% n3 dB of ripple in the pass band, and 30 dB of attenuation in the
% stop band.

[z,p,k] = ellip(4,3,30,200/500);
zplane(z,p);
title('4th-Order Elliptic Lowpass Digital Filter');

% To generate the same plot with a transfer function representation of the filter, use

[b,a] = ellip(4,3,30,200/500);    % Transfer function
zplane(b,a)
```