NED University of Engineering & Technology

Department of Electrical Engineering

LAB MANUAL

For the course

Digital Signal Processing
(EE-395) For T.E.(EE)
(EE-493) For T.E.(BM)

Instructor name:______________________________________

Student name:____

Roll no:____________________ Batch:_____________________

Semester:_______________________ Year:__________________

LAB MANUAL

For the course

Digital Signal Processing
(EE-395) For T.E.(EE)
(EE-493) For T.E.(BM)

Content Revision Team:

Mr. Muhammad Omar & Mr. Nabeel Fayyaz

Approved By

The Board of Studies of Department of Electrical Engineering

_________________________ _________________________

_________________________ _________________________

_________________________ _________________________

Summary of Rubric Sheets for Data Entry in OBE MIS System [to be entered as they are on OBE portal with no scaling]

Lab #	Extent of achievement	Lab #	Extent of achievement	Lab #	Extent of achievement
	Total out of 32		Total out of 32		Total out of 32

Lab #	Extent of achievement	Lab #	Extent of achievement	Lab #	Extent of achievement
	Total out of 32		Total out of 32		Total out of 32

Add six total scores from above to get total Rubric score Out of 192	R=
Scaled to 25 marks (R/192)*25	S=
Lab Attendance percentage from portal:	A=
Attendance scaled to 5 marks: (A/100)*5	P=
For entry on NED exam portal Total lab sessionals out of 30: S + P	T=

PBL/OEL	Extent of achievement	Final	Extent of achievement
	Total out of 32		Total out of 32

Add two total scores from above to get total final Rubric Score Out of 64	Z=
For entry on NED exam portal Total Final Lab Marks scaled to 20 marks (Z/64)*20	F=

CONTENTS

LAB SESSION 01

OBJECTIVE:

To study the relationship between discrete-time and continuous time signals by examining sampling and aliasing.

THEORY:

Signals are physical quantities that carry information in their patterns of variation. Continuous-time signals are continuous functions of time, while discrete-time signals are sequences of numbers. If the values of a sequence are chosen from a finite set of numbers, the sequence is known as a digital signal. Continuous-time, continuous-amplitude signals are also known as analog signals.

Analog phenomenon is continuous – like a human speech of speaker, or a continuously rotating disc attached to the shaft of motor etc. With analog phenomena, there is no clear separation between one point and the next; in fact, between any two points, an infinite number of other points exist. **Discrete phenomenon**, on the other hand, is clearly separated. There's a point (in time or space), and then there's a neighboring point, and there's nothing between the two.

Signal processing is concerned with the acquisition, representation, manipulation, transformation, and extraction of information from signals. In analog signal processing these operations are implemented using analog electronic circuits. Converting the continuous phenomena of images, sound, and motion into a discrete representation that can be handled by a computer is called analog-to-digital conversion. Digital signal processing involves the conversion of analog signals into digital, processing the obtained sequence of finite precision numbers using a digital signal processor or general purpose computer, and, if necessary, converting the resulting sequence back into analog form. When stored in a digital computer, the numbers are held in memory locations, so they would be indexed by memory address. Regardless of the medium (either sound or an image), analog-to-digital conversion requires the same two steps: **Sampling and Quantization.**

Sampling: This operation chooses discrete (finite) points at which to measure a continuous phenomenon (which we will also call a **signal**). In the case of sound, the sample points are evenly separated in time. In the case of images, the sample points are evenly separated in space.

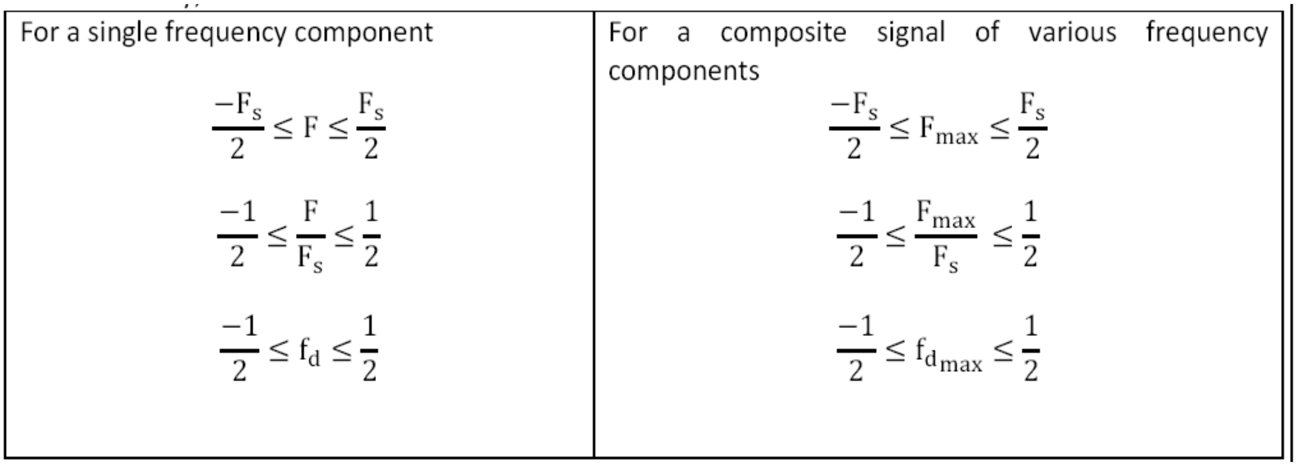
Sampling Rate: The number of samples taken per unit time or unit space is called the sampling rate. The frequency of sampled/discrete phenomenon (signal) can be calculated as

f_d = F /Fs (cycles/sec)/(samples/sec) = cycles/ samples

Where, F = Frequency of analog or continuous phenomenon (signal). [Unit: cycles/sec]
Fs = Sampling frequency or sampling rate [Unit: samples/sec]
fd = Frequency of Discrete phenomenon (signal). [Unit: cycles/sample]

Sampling Theorem: A continuous time phenomenon or signal like x(t) can be reconstructed exactly from its samples x(n) = x(nTs), if the samples are taken at a rate Fs = 1/Ts that is greater than twice the frequency of the signal being sampled i.e. $Fs \geq 2 * F$.

Mathematically,



For a single frequency component	For a composite signal of various frequency components
$\frac{-F_s}{2} \leq F \leq \frac{F_s}{2}$ $\frac{-1}{2} \leq \frac{F}{F_s} \leq \frac{1}{2}$ $\frac{-1}{2} \leq f_d \leq \frac{1}{2}$	$\frac{-F_s}{2} \leq F_{max} \leq \frac{F_s}{2}$ $\frac{-1}{2} \leq \frac{F_{max}}{F_s} \leq \frac{1}{2}$ $\frac{-1}{2} \leq f_{d_{max}} \leq \frac{1}{2}$

Aliasing: A common problem that arises when sampling a continuous signal is aliasing, where a sampled signal has replications of its sinusoidal components which can interfere with other components. It is an effect that causes two discrete time signals to become indistinct due to improper sampling (fd>1/2 cycles/sample).

<u>PROCEDURE:</u>

1. Simulate and plot two CT signals of 10 Hz and 110 Hz for 0 < t < 0.2 secs.
2. Sample at Fs = 100 Hz and plot them in discrete form.
3. Observe and note the *aliasing* effects.
4. Explore and learn.

<u>STEPS:</u>

1. Make a folder at desktop and name it as your current directory within MATLAB.
2. Open M-file editor and type the following code:

```
clear all;close all;clc;

F1 = 10;
F2 = 110;
Fs = 100;
Ts = 1/Fs;
t = [0 : 0.0005 : 0.2];
x1t = cos(2*pi*F1*t);
x2t = cos(2*pi*F2*t);
```

```
figure,
plot(t,x1t,t,x2t, 'LineWidth',2);
xlabel('cont time (sec)');
ylabel('Amp');
xlim([0 0.1]);
grid on;
legend('10Hz','110Hz');
title('Two CTCV sinusoids plotted');
```

3. Save the file as P011.m in your current directory and 'run' it, either using F5 key or writing the file name at the command window.
   (Check for the correctness of the time periods of both sinusoids.)

Now add the following bit of code at the bottom of your P011.m file and save.

```
nTs = [0 :Ts : 0.2];
n = [1 : length(nTs)-1 ];

x1n = cos(2*pi*F1*nTs);
x2n = cos(2*pi*F2*nTs);

figure,
subplot(2,1,1),
stem(nTs,x1n,'LineWidth',2);
grid on;
xlabel('discrete time (sec)');
ylabel('Amp');
xlim([0 0.1]);

subplot(2,1,2)
stem(nTs,x2n,'LineWidth',2);
grid on;
title('110Hz sampled')
xlabel('discrete time(sec)');
ylabel('Amp');
xlim([0 0.1]);
```

1. Before hitting the 'run', just try to understand what the code is doing and try to link it with what we have studied in classes regarding concepts of frequency for DT signals.
2. Now 'run' the file and observe both plots.

To see what is really happening, type the following code at the bottom of your existing P011.m file and run again.

```
 figure,
plot(t,x1t,t,x2t);
hold;
stem(nTs,x1n,'r','LineWidth',2);

xlabel('time (sec)');
ylabel('Amp');
xlim([0 0.05]);

 legend('10Hz','110Hz');
```

3. Observe the plots.

<u>RESULT:</u>

Explain (write) in your own words the cause and effects of what you just saw.

<u>LAB TASKS:</u>

1. Consider the following CT signal:
x(t) = sin (2 pi F0 t). The sampled version will be:
x(n) = sin (2 pi F0/Fs n),
where n is a set of integers and sampling interval Ts=1/Fs.
Plot the signal x(n) for n = 0 to 99
for Fs = 5 kHz and F1 = 0.5, 2, 3 and 4.5 kHz. Explain the similarities and differences among various plots.

2. Generate a tone in MATLAB with varying frequency f = 1000,2000,3000,4000, 5000, 6000, 8000, 9000, 25000,-1000,-2000,-3000 Hz with Fs = 8000 samples/sec.
Listen to the tones,and observe at Sounds like what frequency?
Also Specify whether Aliasing is happening or not.

3. Record a sentence in your voice.(you may use Simulink /audacity to record).Change Fs =44100, 22050, 11025, 8192, 4096 , 2048 , 1024
and observe

a) Voice quality during playback [Excellent/Good/OK/Bad]
b) File size in kilobytes
c) Aliasing happening or not?

NED University of Engineering & Technology
Department of ____________________ Engineering

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Software Menu Identification and Usage: Ability to initialise, configure and *operate* software environment *under supervision*, using menus, shortcuts, instructions etc.	Unable to understand and use software menu	Little ability and understanding of software menu operation, makes many mistake	Moderate ability and understanding of software menu operation, makes lesser mistakes	Reasonable understanding of software menu operation, makes no major mistakes	Demonstrates command over software menu usage with frequent use of advance menu options
Procedural Programming of given model: *Practice* procedural programming techniques, in order to code specific model	Little to no understanding of procedural programming techniques	Slight ability to use procedural programming techniques for coding given algorithm	Mostly correct recognition and application of procedural programming techniques but makes crucial errors for the given model	Correctly recognises and uses procedural programming techniques with no errors but unable to run model successfully	Correctly recognises and uses procedural programming techniques with no errors and runs model successfully
Relating Theoretical Concepts, Equations and Transforms to Code: *Recognise* relation between model concepts and written code and *manipulate* the code in accordance of requirements	Completely unable to relate between model concepts and written code, unable to do manipulations	Able to recognise some relation between model concepts and written code, unable to do manipulations	Able to recognise relation between model concepts and written code, unable to do manipulations	Able to recognise relation between model concepts and written code, able to do some manipulations	Able to recognise relation between model concepts and written code, able to completely manipulate code in line with theoretical concepts
Detecting and Removing Errors: *Detect* Errors/Exceptions and in simulation and *manipulate* code to rectify the simulation	Unable to check and detect error messages and indications in software	Able to find error messages and indications in software but no understanding of detecting those errors and their types	Able to find error messages and indications in software as well as understanding of detecting some of those errors and their types	Able to find error messages in software as well as understanding of detecting all of those errors and their types	Able to find error messages in software along with the understanding to detect and rectify them

Graphical Visualisation and Comparison of model **Parameters:** *Manipulate* given simulation under supervision, in order to produce graphs/plots for measuring and comparing model parameters	Unable to understand and utilise visualisation or plotting features	Ability to understand and utilise visualisation and plotting features with frequent errors	Ability to understand and utilise visualisation and plotting features successfully but unable to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, partially able to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, also able to compare and analyse them
Following step-by-step procedure to complete lab work: *Observe, imitate and operate* software to complete the provided sequence of steps	Inability to recognise and perform given lab procedures	Able to recognise given lab procedures and perform them but could not follow the prescribed order of steps	Able to recognise given lab procedures and perform them by following prescribed order of steps, with frequent mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with occasional mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with no mistakes
Recording Simulation Observations: *Observe and copy* prescribed or required simulation results in accordance with lab manual instructions	Inability to recognise prescribed or required simulation measurements	Able to recognise prescribed or required simulation measurements but does not record according to given instructions	—	Able to recognise prescribed or required simulation measurements but records them incompletely	Able to recognise prescribed or required simulation measurements and records them completely, in tabular form
Discussion and Conclusion: *Demonstrate* discussion capacity on the recorded observations and draw conclusions from it, relating them to theoretical principles/concepts	Complete inability to discuss recorded observations and draw conclusions	Slight ability to discuss recorded observations and draw conclusions	Moderate ability to discuss recorded observations and draw conclusions	Reasonable ability to discuss recorded observations and draw conclusions	Full ability to discuss recorded observations and draw conclusions

LAB SESSION 02

OBJECTIVE:

To observe the quantization effects on sampled signals and to understand how quantization leads to quantization error. In this lab, we will investigate the influence of the number of quantization levels on the quality of digitized signal. Method of selection of ADC is also a part of this lab session.

THEORY:

Everything stored on a computer is discrete time discrete valued signal. Because computer has finite number of registers and each register is a finite length register. We take too many samples to give the 'effect' of continuous time signals. But actually they are discrete time. We also take very fine resolution of amplitude axis to give the effect of continuous valued signal but due to finite word length of the computer register, the stored variables are already quantized. This lab aims to explain the quantization effects in a computer.

Regardless of the medium (audio or image), the digitization of real world analog signal usually involves two stages: **sampling**, i.e. the measurement of signal at discretely spaced time intervals, and **quantization**, i.e. the transformation of the measurements (amplitudes) into finite-precision numbers (allowed discrete levels), such that they can be represented in computer memory. **Quantization** is a matter of representing the amplitude of individual samples as integers expressed in binary. The fact that integers are used forces the samples to be measured in a finite number of bits (discrete levels). The range of the integers possible is determined by the **bit depth**, the number of bits used per sample. The **bit depth** limits the precision with which each sample can be represented.

Bit Depth:

Within digital hardware, numbers are represented by binary digits known as bits—in fact, the term bit originated from the words Binary digit. A single bit can be in only one of two possible states: either a one or a zero. When samples are taken, the amplitude at that moment in time must be converted to integers in binary representation. The number of bits used to represent each sample, called **the bit depth (bits/sample) or sample size,** determines the precision with which the sample amplitudes can be represented. Each bit in a binary number holds either a 1 or a 0. In digital sound, bit depth affects how much you have to round off the amplitude of the wave when it is sampled at various points in time

The number of different values that can be represented with b-bit is 2^b .The largest decimal number that can be represented with an b-bit binary number is 2^b - 1. For example, the decimal values that can be represented with an 8-bit binary number range from 0 to 255, so there are 256 different values (levels of ADC). A bit depth of 8 allows 2^8=256 different discrete levels at which samples can be approximated or recorded. Eight bits together constitute one byte. A bit depth of 16 allows 2^{16} = 65,536 discrete levels, which in turn provides much higher precision than a bit depth of 8.

The number of bits in a data word is a key consideration. The more bits used in the word, the better the resolution of the number, and the larger the maximum value that can be represented. Some computers use 64-bit words. Now, 2^{64} is approximately equal to 1.8 x 10^{19}—that's a pretty large number. So large, in fact, that if we started incrementing a 64-bit counter once per second at the beginning of the universe (≈20 billion years ago), the most significant four bits of this counter would still be all zeros today.

To simplify the explanation, take an example of ADC with a bit depth of 3, 2^3 = 8 quantization levels ranging from -4 to 3 are possible in signed magnitude representation. For bipolar ADCs (or signed magnitude representation), by convention, half of the quantization levels are below the horizontal axis (that is $\mathbf{2^1}$, of the quantization levels). One level is the horizontal axis itself (level 0), and $\mathbf{2^{b-1} - 1}$levels are above the horizontal axis.Note that since one bit is used for the signed bit (in 2-complementformat), the largest magnitude corresponds to 2^(b -1). (not $\mathbf{2^b}$). When a sound is sampled, each sample must be scaled to one of the 8 discrete levels. However, the samples in reality might not fall neatly onto these levels. They have to be rounded up or down by some consistent convention.

QUANTIZATION ERROR:

The samples, which are taken at evenly-spaced points in time, can take on the values only at the discrete quantization levels to store on our computer. Therefore quantization leads to a loss in the signal quality, because it introduces a **“Quantization error**”. Quantization error is sometimes referred to as "'**Quantization noise**". **Noise** can be broadly defined as part of an audio signal that isn’t supposed to be there. However, some sources would argue that a better term for quantization error is "**distortion**", defining distortion as an unwanted part of an audio signal that is related to the true signal.

The difference between the quantized samples and the original samples constitutes quantization error or rounding error (if round-off method is used). $\mathbf{Xe(n) = X_q(n) - x(n).}$ The lower the bit depth, the more values potentially must be approximated (rounded), resulting in greater quantization error

To calculate the required bit depth of ADC i.e. bits/sample, there are two important points which we must have to consider:

a) How much noise is already present in the analog signal?

b) How much more noise can be tolerated in the digital signal? Signal-to -noise-ratio- SNR (of analog signal)

Before looking at SNR specifically in the context of digital imaging and sound, let's consider the general definition. Signal-to-noise ratio can generally be defined as the ratio of the meaningful content of a signal versus the associated background noise.

SNR = 10log$_{10}$ (P$_x$ /P$_e$)

Where, **P$_x$**and **P$_e$** are average power of the analog signal and noise signal respectively.

A signal is any type of communication – something a person says to you, a digital signal sending an image file across a network, a message posted on an electronic bulletin board, a piece of audio being played from a cassette, etc. The noise is the part of the message that is not meaningful; in fact, it gets in the way of the message intended in the communication. You could use the term signal-to-noise ratio to describe communications in your everyday life. If you know someone who talks a lot but doesn't really convey a lot of meaning, you could say that he or she has a low signal-to-noise ratio. Web-based bulletin board and chat groups are sometimes described as having a low SNR – there may be quite a few postings, but very much meaningful content. In these first two examples, the noise consists of all the empty "filler" words. In the case of a digital signal sent across a network, the noise is the electronic degradation of the signal. On a piece of audio played from cassette, the noise could be caused by damage to the tape or mechanical imperfections in the cassette player.

In analog data communication (analog signals), the signal-to-noise ratio is defined as the ratio of the average power in the signal versus the power in the noise level. In this context, think of a signal being sent over a network connection compared to the extent to which the signal is corrupted. This is related to the general usage of the term described above. This usage of the term SNR applies to analog signals.

SIGNAL-TO-QUANTIZATION-NOISE-RATIO- SQNR (OF ADC):

Using finite word lengths prevents us from representing values with infinite precision, increases the background noise in our spectral estimation techniques etc. The amount of error implicit in a chosen bit depth can be measured in terms of the signal-to-noise ratio (SNR).

For a digitized image or sound, the signal-to-noise ratio is defined as the ratio of the maximum sample value versus the maximum quantization error. In this usage of the term, the ratio depends on the bit depth chosen for the signal. Any signal encoded with a given bit depth will have the same ratio. This can also be called signal-toquantization-noise ratio (SQNR), but you should be aware that in many sources the term signal-to-noise ratio is used with this meaning as well. (Henceforth, we'll use the term SQNR to distinguish this measurement from SNR.)

Practical A/D converters are constrained to have binary output words of finite length. Commercial A/D converters are categorized by their output word lengths, which are normally in the range from 8 to 16 bits. There is no infinite bit ADC. So whenever we will digitize our signal, we will always have a quantization error. Quantization error represents the quality of quantization process but the total error may also turn out to be zero, so signal-toquantization-noise-ratio (SQNR) is used to describe the quality of quantization process and it can be defined as

SQNR$_{A/D}$ **= 10 log** $_{10}$**(Px /P**$_e$**)**

Where, **Px**~and **P**$_e$ are average power of the DTCV (sampled) signal and quantization error signal respectively.

$$P_x = \frac{1}{N}\sum_{n=0}^{N-1} \left|x(n)\right|^2$$

$$P_e = \frac{1}{N}\sum_{n=0}^{N-1} \left|x_e(n)\right|^2 => P_e = \frac{1}{N}\sum_{n=0}^{N-1} \left|x_q(n) - x(n)\right|^2$$

PROCEDURE:

1. Simulate a DTCV sinusoid of 1/50 cycles/sample with length of the signal be 500.
2. Choose the no. of significant digits for round-off and apply to the signal generated above.
3. Compute the error signals and SQNR
4. Explore and observe.

STEPS:

1. Make a folder at desktop and name it as your current directory within MATLAB.
2. Open M-file editor and write the following code:

```
clear all;
close all;
clc;
fd1 = 1/50;
n = [0 : 499 ];


q=input('No. of Digits after decimal points to be retained (0-9): ');
x1 = cos(2*pi*fd1*n);
Px1 = sum(abs(x1).^2)/length(x1);
x1q = round(x1*10^q)/10^q;
x1e = x1 -x1q;
Pe1 = sum(abs(x1e).^2)/length(x1e);



SQNR = 10*log10(Px1/Pe1);
```

```
disp(['The Signal to Quantization Noise Ratio is: ' num2str(SQNR) '
dB.' ]);
figure,
subplot(2,1,1);
plot(n,x1,n,x1q);
xlabel('indices');
ylabel('Amp');
xlim([0 49]);
ylim([-1.1 1.1]);
legend('DTCV','DTDV');


subplot(2,1,2);
plot(n,x1e);
xlabel('indices');
ylabel('Error');
xlim([0 49]);
```

3. Save the file as **P021.m** in your current directory and run it.

Explore and take notes.
Now modify the above code as follows and save as another file P022.m.

```
clear all;
close all;
clc;
fd1 = 1/50;
n = [0 : 499 ];
q = [0 : 10];

% No. of Digits after decimal points to be retained for num = 1 :
length(q)
 x1 = cos(2*pi*fd1*n);
 Px1 = sum(abs(x1).^2)/length(x1);
 x1q = round(x1*10^q(num))/10^q(num);
 x1e = x1 -x1q;
Pe1 = sum(abs(x1e).^2)/length(x1e);
SQNR(num) = 10*log10(Px1/Pe1);
end

figure,
plot(q,SQNR);
xlabel('Significant Digits');
ylabel('SQNR (dB)');
xlim([q(1) q(end)]);
```

1. Before hitting the ‘run’, just try to understand what the code is doing and try to link it with

the previous code.

2. Now ‘run’ the file and observe the results.

RESULT:

Explain (write) in your own words the cause and effects of what you just saw.

LAB TASKS:

1. Effects of Quantization with variable precision levels

Simulate a DTCV sampled composite signal of $fd1$=125 samples/sec and $fd2$=150 samples/sec with length of the signal be 250 samples.

Take the desired number of significant digits from user as an input. Then choose the method of Quantization (round-off, floor & ceil) and apply to the signal generated above.

Compute the quantization error signals and SQNR.

2. **Simple sinusoid quantized to various bits per sample**

Generate a 100 Hz sinusoid sampled at 10000 samples/sec and quantized at 1_bit/sample.

Now increase the bit depth for various numbers of bits per sample (2, 3, 4, 5, 6, 7, 8) and attach plots. You can use two column format for plotting (but the diagrams should be visible).

3. Audio signal quantization to various bits per sample

Use your recorded voice in last session and quantize it at 1 bit /sample.

Change bit depth to 2,3,4 and then listen and take notes of your observations.

Decide no. of bits for audio until quality stops improving.

NED University of Engineering & Technology
Department of ____________________ Engineering

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Software Menu Identification and Usage: Ability to initialise, configure and *operate* software environment *under supervision*, using menus, shortcuts, instructions etc.	Unable to understand and use software menu	Little ability and understanding of software menu operation, makes many mistake	Moderate ability and understanding of software menu operation, makes lesser mistakes	Reasonable understanding of software menu operation, makes no major mistakes	Demonstrates command over software menu usage with frequent use of advance menu options
Procedural Programming of given model: *Practice* procedural programming techniques, in order to code specific model	Little to no understanding of procedural programming techniques	Slight ability to use procedural programming techniques for coding given algorithm	Mostly correct recognition and application of procedural programming techniques but makes crucial errors for the given model	Correctly recognises and uses procedural programming techniques with no errors but unable to run model successfully	Correctly recognises and uses procedural programming techniques with no errors and runs model successfully
Relating Theoretical Concepts, Equations and Transforms to Code: *Recognise* relation between model concepts and written code and *manipulate* the code in accordance of requirements	Completely unable to relate between model concepts and written code, unable to do manipulations	Able to recognise some relation between model concepts and written code, unable to do manipulations	Able to recognise relation between model concepts and written code, unable to do manipulations	Able to recognise relation between model concepts and written code, able to do some manipulations	Able to recognise relation between model concepts and written code, able to completely manipulate code in line with theoretical concepts
Detecting and Removing Errors: *Detect* Errors/Exceptions and in simulation and *manipulate* code to rectify the simulation	Unable to check and detect error messages and indications in software	Able to find error messages and indications in software but no understanding of detecting those errors and their types	Able to find error messages and indications in software as well as understanding of detecting some of those errors and their types	Able to find error messages in software as well as understanding of detecting all of those errors and their types	Able to find error messages in software along with the understanding to detect and rectify them

Graphical Visualisation and Comparison of model **Parameters:** *Manipulate* given simulation under supervision, in order to produce graphs/plots for measuring and comparing model parameters	Unable to understand and utilise visualisation or plotting features	Ability to understand and utilise visualisation and plotting features with frequent errors	Ability to understand and utilise visualisation and plotting features successfully but unable to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, partially able to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, also able to compare and analyse them
Following step-by-step procedure to complete lab work: *Observe, imitate and operate* software to complete the provided sequence of steps	Inability to recognise and perform given lab procedures	Able to recognise given lab procedures and perform them but could not follow the prescribed order of steps	Able to recognise given lab procedures and perform them by following prescribed order of steps, with frequent mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with occasional mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with no mistakes
Recording Simulation Observations: *Observe and copy* prescribed or required simulation results in accordance with lab manual instructions	Inability to recognise prescribed or required simulation measurements	Able to recognise prescribed or required simulation measurements but does not record according to given instructions	—	Able to recognise prescribed or required simulation measurements but records them incompletely	Able to recognise prescribed or required simulation measurements and records them completely, in tabular form
Discussion and Conclusion: *Demonstrate* discussion capacity on the recorded observations and draw conclusions from it, relating them to theoretical principles/concepts	Complete inability to discuss recorded observations and draw conclusions	Slight ability to discuss recorded observations and draw conclusions	Moderate ability to discuss recorded observations and draw conclusions	Reasonable ability to discuss recorded observations and draw conclusions	Full ability to discuss recorded observations and draw conclusions

LAB SESSION 03

OBJECTIVE:

To study impulse response, observe convolution technique in signal processing, and verify different properties like causality, commutative, distributive and associative properties.

THEORY:

1. Convolution is given as :y(n) = x(n)*h(n) =

$$= \sum_{k=-\infty}^{\infty} x(k)h(n-k) = \sum_{k=-\infty}^{\infty} h(k)x(n-k).$$

   i.e.one can compute the output y(n) to a certain input x(n) when impulse response h(n) of that system is known. Convolution holds commutative property.

2. The length of the resulting convolution sequence is N+M-1,where N and M are the lengths of two convolved signals respectively.

3. In causal system, the outputs only depend on the past and/or present values of inputs and NOT on future values. This means that the impulse response h(n) of a causal system will always exist only for n≥ 0.

PROCEDURE:

1. We have the impulse response of a system as h(n) = { 3,2,1,-2,1,0,-4,0,3}
   ↑
2. For x(n)={1,-2,3,-4,3,2,1}
   ↑

STEPS:

1. Make a folder at desktop and name it as your current directory within MATLAB.
2. Open M-file editor and write the following code:

```
clear all;
close all;
 clc;
h = [3 2 1 -2 1 0 -4 0 3]; % impulse response
org_h = 1; % Sample number where origin exists
nh = [0 : length(h)-1]- org_h + 1;
x = [1 -2 3 -4 3 2 1]; % input sequence
org_x = 1; % Sample number where origin exists
nx = [0 : length(x)-1]- org_x + 1;
```

```
y = conv(h,x);

ny = [nh(1)+ nx(1) : nh(end)+nx(end)];
figure,

subplot(3,1,1),
stem(nh,h);
xlabel('Time index n');
ylabel('Amplitude');
xlim([nh(1)-1 nh(end)+1]);
title('Impulse Response h(n)');
grid;

subplot(3,1,2),
stem(nx,x);
xlabel('Time index n');
ylabel('Amplitude');
xlim([nx(1)-1 nx(end)+1]);
title('Input Signal x(n)');
grid;

subplot(3,1,3)
stem(ny,y);
xlabel('Time index n');
ylabel('Amplitude');
xlim([ny(1)-1 ny(end)+1]);
title('Output Obtained by Convolution');
grid;
```

1. Save the file as **P031.m** in your current directory and ‘run’ it.
2. Calculate the length of input signal (N) and impulse response (M) used in above task?
3. Calculate the length of the output sequence and verify the result with N+M-1
4. Try to learn, explore the code and make notes.
5. Now modify the above code such that h(n)= {3,2, 1, -2,1,0,-4,0,3}(origin is shifted) and check for causality.

↑

RESULT:

EXERCISE:

1. What will happen if we input $x(n)=\{0,0,\underset{\uparrow}{1},0,0\}$ into the above system.
2. Can you prove the commutative property of the convolution?
3. Modify the code to prove Associative and Distributed properties of the convolution.
4. Convolve your recorded sound with drumloop.wav. Note your observation

   a) Plot the output.
   b) Listen the output

NED University of Engineering & Technology
Department of ______________________ Engineering

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Software Menu Identification and Usage: Ability to initialise, configure and *operate* software environment *under supervision*, using menus, shortcuts, instructions etc.	Unable to understand and use software menu	Little ability and understanding of software menu operation, makes many mistake	Moderate ability and understanding of software menu operation, makes lesser mistakes	Reasonable understanding of software menu operation, makes no major mistakes	Demonstrates command over software menu usage with frequent use of advance menu options
Procedural Programming of given model: *Practice* procedural programming techniques, in order to code specific model	Little to no understanding of procedural programming techniques	Slight ability to use procedural programming techniques for coding given algorithm	Mostly correct recognition and application of procedural programming techniques but makes crucial errors for the given model	Correctly recognises and uses procedural programming techniques with no errors but unable to run model successfully	Correctly recognises and uses procedural programming techniques with no errors and runs model successfully
Relating Theoretical Concepts, Equations and Transforms to Code: *Recognise* relation between model concepts and written code and *manipulate* the code in accordance of requirements	Completely unable to relate between model concepts and written code, unable to do manipulations	Able to recognise some relation between model concepts and written code, unable to do manipulations	Able to recognise relation between model concepts and written code, unable to do manipulations	Able to recognise relation between model concepts and written code, able to do some manipulations	Able to recognise relation between model concepts and written code, able to completely manipulate code in line with theoretical concepts
Detecting and Removing Errors: *Detect* Errors/Exceptions and in simulation and *manipulate* code to rectify the simulation	Unable to check and detect error messages and indications in software	Able to find error messages and indications in software but no understanding of detecting those errors and their types	Able to find error messages and indications in software as well as understanding of detecting some of those errors and their types	Able to find error messages in software as well as understanding of detecting all of those errors and their types	Able to find error messages in software along with the understanding to detect and rectify them

Graphical Visualisation and Comparison of model **Parameters:** *Manipulate* given simulation under supervision, in order to produce graphs/plots for measuring and comparing model parameters	Unable to understand and utilise visualisation or plotting features	Ability to understand and utilise visualisation and plotting features with frequent errors	Ability to understand and utilise visualisation and plotting features successfully but unable to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, partially able to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, also able to compare and analyse them
Following step-by-step procedure to complete lab work: *Observe, imitate and operate* software to complete the provided sequence of steps	Inability to recognise and perform given lab procedures	Able to recognise given lab procedures and perform them but could not follow the prescribed order of steps	Able to recognise given lab procedures and perform them by following prescribed order of steps, with frequent mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with occasional mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with no mistakes
Recording Simulation Observations: *Observe and copy* prescribed or required simulation results in accordance with lab manual instructions	Inability to recognise prescribed or required simulation measurements	Able to recognise prescribed or required simulation measurements but does not record according to given instructions	—	Able to recognise prescribed or required simulation measurements but records them incompletely	Able to recognise prescribed or required simulation measurements and records them completely, in tabular form
Discussion and Conclusion: *Demonstrate* discussion capacity on the recorded observations and draw conclusions from it, relating them to theoretical principles/concepts	Complete inability to discuss recorded observations and draw conclusions	Slight ability to discuss recorded observations and draw conclusions	Moderate ability to discuss recorded observations and draw conclusions	Reasonable ability to discuss recorded observations and draw conclusions	Full ability to discuss recorded observations and draw conclusions

LAB SESSION 04

OBJECTIVE:

To study discrete time correlation and apply it to real data to observe the correlation between two signals.

THEORY:

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n)y(n-l) = \sum_{n=-\infty}^{\infty} x(n+l)y(n)$$

1. **Correlation** is given as where ‘*l*’ is the lag. This is called cross-correlation and it gives the magniyude and location of similarity between two signals. The correlation between x(n) and y(n) . It is given as:

$$r_{yx}(l) = \sum_{n=-\infty}^{\infty} y(n)x(n-l) = \sum_{n=-\infty}^{\infty} y(n+l)x(n).$$

2. Generally r*xy*(l) = r_{yx}(*l*). These two are the same when x(n) and y(n) are the same signals or when x(n) and y(n) are even symmetric signals .

3. The length of the resulting correlation sequence is N+M-1, where N and M are the lengths of the two signals.

4. Correlation may also be computed using convolution algorithm with a modification that we need to fold one of the signals before applying convolution**.**
   Mathematically, r_{xy}(n)= x(n) * y(-n)

STEPS:

1. Generate two sinusoids of length 10 and fd = 0.1 with variable phase.
2. Apply correlation and check for certain properties such as magnitude and location of maximum correlation with varying phases.

PROCEDURE:

1.Make a folder at desktop and name it as your current directory within MATLAB. -
2.Open M-file editor and write the following code:)

```
clear all;
close all;
```

```
clc;
n = [0:9];
ph1 = 0;
ph2 = 0;
x = sin(2*pi*0.1*n + ph1);
org_x = 1;
nx = [0 : length(x)-1]- org_x + 1;

y = sin(2*pi*0.1*n + ph2);
org_y = 1;
ny = [0 : length(y)-1]- org_y + 1;

rxy = xcorr(x,y);
nr = [nx(1)-ny(end) : nx(end)-ny(1)];

[maxR indR] = max(rxy);

disp(['The correlation at lag zero is: ' num2str(rxy(find(nr==0)))
'.']);
disp(['The maximum correlation is at lag ' num2str(nr(indR)) '.']);
|
figure,
subplot(3,1,1),
stem(nx,x);
xlabel('Time index n');
ylabel('Amplitude');
xlim([nx(1)-1 nx(end)+1]);
title('Signal x(n)');
grid;

subplot(3,1,2),
stem(ny,y);
xlabel('Time index n');
ylabel('Amplitude');
xlim([ny(1)-1 ny(end)+1]);
title('Signal y(n)');
grid;

subplot(3,1,3)
stem(nr,rxy);
xlabel('Time index n');
ylabel('Amplitude');
xlim([nr(1)-1 nr(end)+1]);
title('Cross Correlation');
grid;
```

Save the file as **P041.m** in your current directory and 'run' it.

Learn the specific logical bits of the code and make notes

Now modify the phase of the second signal to pi/2 (it will make it cosine) and observe the correlation at lag zero. Modify the phase again to 'pi' and observe.

1. Check for auto-correlation (ph1 = ph2) that the lag zero value gives the energy of the Signal.
2. Observe that the commutative property does not hold.

RESULT:

Please write in exercise book.

EXERCISE:

1. Now modify the phase of the second signal to pi/2 (it will make it cosine)and observe the correlation at lag zero.
2. Modify the phase again to 'pi' and observe.
3. Check for auto-correlation (ph1 = ph2) that the lag zero value gives the m energy of the signal.
4. Observe that the commutative property does not hold.
5. Modify the code, such that the correlation is obtained using convolution command.
6. Calculate correlation between voltages of any two phases of a 10HP motor Using the data given below. First use Ms. Excel to copy data and then calculate correlation.

Voltage A Min	Voltage B Min	Voltage C Min
189.358	153.917	195.735
189.175	159.719	201.877
188.783	161.575	186.718
188.757	172.186	187.659
176.995	173.206	205.876
180.472	176.865	204.831
180.524	176.917	192.494
180.262	189.28	199.839
181.778	189.828	211.887
179.975	189.462	211.94
178.642	189.253	212.462
180.315	188.94	193.749
180.707	190.377	200.492
180.262	190.194	201.433
180.628	190.064	202.635
180.315	189.907	200.701
179.635	189.541	203.289
179.243	189.567	202.635
179.4	189.619	200.989
180.576	189.044	197.591
180.837	189.123	199.865
180.184	189.332	201.093
180.08	189.097	201.041
177.675	189.044	199.656
175.297	189.018	198.558
173.99	189.123	204.595

LAB SESSION 05

OBJECTIVE:

To study the computer implementation of Discrete FourierT transform and Inverse Fourier Transform using Twiddle factor.

THEORY:

The formulas for the DFT and IDFT are given as

X(K) = $\sum_{n=0}^{N-1} x(n)\, W_N^{kn}$; k=0,1,……N-1

X(n) = $\frac{1}{N} \sum_{n=0}^{N-1} X(k)\, W_N^{-kn}$; k=0,1,……N-1

Where by definition $W_N = e^{\frac{-j2\Pi}{N}}$

Which is an Nth root of unity. Where W_N is called Twiddle Factor , also

$[W_N] = [e^{-j2\Pi/N}]^{kn}$

$W_N^{kn} = e^{-j2\Pi kn/N}$

DFT analysis equation in matrix form is

$X_N = [\, W_N^{kn}\,] x_N$

DFT synthesis equation in matrix form is

$x_N = [W_N^{kn}]^{-1} X_N$

PROCEDURE:

TASK

Compute 4 point DFT of *x*(n)= (1,2,3,0).

STEPS

1.Generate given sequence in Matlab .
2.Take N-=4 to calculate 4-point DFT.
3.Define 0: N-1 point vector for time and frequency samples.
4.Define W matrix and then use DFT analysis equation to compute DFT.

```
close all,
clear all;
clc;
x=[1 ,2 ,3 ,0];
N=4;
n=[0:1:N-1];
k=[0:1:N-1];

WN=exp(-j*2*pi/N);

nk=n'*k;

WNnk=WN.^nk;

Xk=x*WNnk
```

LAB TASK

Prove DFT synthesis equation using DFT output generated from lab task.

LAB SESSION 06

OBJECTIVE:

To observe/find different frequency components in an audio signal and plot it with different x_ axes .

THEORY:

- DF analysis equation: $X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}$, $k = 0,1,\dots,N-1$
- DFT synthesis eq: $x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N}$, $n = 0,1,\dots,N-1$
- Frequency Resolution is given by $\Delta F = \frac{F_s}{N}$.
- Sample frequencies are given by $F_k = \Delta F \times k$, $k = 0,1,\dots,N-1$.

PROCEDURE:

1. Load an audio file ‘noisy.wav’ into Matlab.
2. There is a tone added to the speech in this file. The objective is to find the frequency of this tone.
3. Computing the DFT of this signal;
4. Generating frequency vector in Hz.
5. Displaying the DFT and observing the frequency of the added tone.

STEPS

1. Make a folder at desktop and name it as your current directory within MATLAB.
2. Copy the audio file ‘noisy.wav’ into your current directory.
3. Open M file editor and write the following code:

```
clear all; clc; close all;
[y,Fs,bits] = wavread('noisy.wav');
 Ts = 1/Fs;
n = [0:length(y)-1];
t = n.*Ts; k = n;
Df = Fs./length(y);
F = k.*Df;
Y = fft(y);
magY = abs(Y);
sound(y,Fs);
```

```
figure,

subplot(2,1,1);
plot(F,magY);
grid on;
xlim([0 Fs/2]);
xlabel('Frequency (Hz)');
ylabel('DFT Magnitude');
title('Discrete Fourier Transform');

subplot(2,1,2);
plot(F,magY);
grid on;

xlim([0 2000]);
xlabel('Frequency (Hz)');
ylabel('DFT Magnitude');
title('Discrete Fourier
Transform');
```

4. Save the file as **P081.m** in your current directory and run it.

RESULT:

Explore and take notes.

EXERCISE:

Use recorded data,
1. Plot different frequencies present in it with
a) x-axis as time
b)x-axis as frequency. (Take FFT and plot).
2. Calculate the amount of energy present in fundamental frequency.
3. Calculate the amount of energy present in different harmonics.

NED University of Engineering & Technology
Department of ______________________ Engineering

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Software Menu Identification and Usage: Ability to initialise, configure and *operate* software environment *under supervision*, using menus, shortcuts, instructions etc.	Unable to understand and use software menu	Little ability and understanding of software menu operation, makes many mistake	Moderate ability and understanding of software menu operation, makes lesser mistakes	Reasonable understanding of software menu operation, makes no major mistakes	Demonstrates command over software menu usage with frequent use of advance menu options
Procedural Programming of given model: *Practice* procedural programming techniques, in order to code specific model	Little to no understanding of procedural programming techniques	Slight ability to use procedural programming techniques for coding given algorithm	Mostly correct recognition and application of procedural programming techniques but makes crucial errors for the given model	Correctly recognises and uses procedural programming techniques with no errors but unable to run model successfully	Correctly recognises and uses procedural programming techniques with no errors and runs model successfully
Relating Theoretical Concepts, Equations and Transforms to Code: *Recognise* relation between model concepts and written code and *manipulate* the code in accordance of requirements	Completely unable to relate between model concepts and written code, unable to do manipulations	Able to recognise some relation between model concepts and written code, unable to do manipulations	Able to recognise relation between model concepts and written code, unable to do manipulations	Able to recognise relation between model concepts and written code, able to do some manipulations	Able to recognise relation between model concepts and written code, able to completely manipulate code in line with theoretical concepts
Detecting and Removing Errors: *Detect* Errors/Exceptions and in simulation and *manipulate* code to rectify the simulation	Unable to check and detect error messages and indications in software	Able to find error messages and indications in software but no understanding of detecting those errors and their types	Able to find error messages and indications in software as well as understanding of detecting some of those errors and their types	Able to find error messages in software as well as understanding of detecting all of those errors and their types	Able to find error messages in software along with the understanding to detect and rectify them

Graphical Visualisation and Comparison of model **Parameters:** *Manipulate* given simulation under supervision, in order to produce graphs/plots for measuring and comparing model parameters	Unable to understand and utilise visualisation or plotting features	Ability to understand and utilise visualisation and plotting features with frequent errors	Ability to understand and utilise visualisation and plotting features successfully but unable to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, partially able to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, also able to compare and analyse them
Following step-by-step procedure to complete lab work: *Observe, imitate and operate* software to complete the provided sequence of steps	Inability to recognise and perform given lab procedures	Able to recognise given lab procedures and perform them but could not follow the prescribed order of steps	Able to recognise given lab procedures and perform them by following prescribed order of steps, with frequent mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with occasional mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with no mistakes
Recording Simulation Observations: *Observe and copy* prescribed or required simulation results in accordance with lab manual instructions	Inability to recognise prescribed or required simulation measurements	Able to recognise prescribed or required simulation measurements but does not record according to given instructions	—	Able to recognise prescribed or required simulation measurements but records them incompletely	Able to recognise prescribed or required simulation measurements and records them completely, in tabular form
Discussion and Conclusion: *Demonstrate* discussion capacity on the recorded observations and draw conclusions from it, relating them to theoretical principles/concepts	Complete inability to discuss recorded observations and draw conclusions	Slight ability to discuss recorded observations and draw conclusions	Moderate ability to discuss recorded observations and draw conclusions	Reasonable ability to discuss recorded observations and draw conclusions	Full ability to discuss recorded observations and draw conclusions

LAB SESSION 07

OBJECTIVE:

To study s-plane and plot impulse and frequency response for different pole zero location in s-plane. Also to determine weather system is FIR or IIR.

THEORY:

The Laplace Transform of a general continuous time signal x (t) is defined as;

$$X(S) = \int x(t)\, e^{-st}\, dt.$$

Where the complex variable s=δ+ j w, with δ and w the real and imaginary parts. CTFT is a subset of Laplace when δ =0. Since 'δ' information is not present in CTFT, therefore information about stability can only be obtained from Laplace. If pole lies on L.H.S of s-plane, system is stable. If pole lies on R.H.S of s-plane, system is unstable. If pole lies on y(jw)-axis, system is marginally stable or oscillatory. If system has FIR, it is stable. If system is IIR, it can be stable or unstable.

PROCEDURE:

Generate pole zero constellation in s plane.

1. Plot corresponding Frequency (Bode magnitude) response.
2. Plot impulse response and determine that the system is FIR or IIR.
3. Modify location of poles in s plane to observe the corresponding change in frequency and impulse response.

STEPS.

1. Make a folder at desktop and name it as your current directory within MATLAB.
2. Open M-file editor and write the following code:

```
clear all;
close all;
clc;
Num = poly([(0-(i*(pi/2))),(0+(i*(pi/2)))]);
Zeros=roots(Num)
Den = poly([-1,-1]);
poles=roots(Den) sys=tf(Num,Den)

figure;
subplot(3,1,1);
pzmap(sys);
xlim([-2 2]);
ylim([-4 4]);
```

```
subplot(3,1,2);
[mag phase w]=bode(sys);
mag=squeeze(mag);
plot(w,mag);

subplot(3,1,3);
impulse(sys);
H=dfilt.df1(Num,Den);A=isfir(H)
```

3. Save the file as **P091.m** in your current directory and ‘run’ it.

RESULT:

1. Learn the specific logical bits of the code and make notes.
2. Observe the plots.
3. Now, explain (write) in your own words the cause and effects of what you just saw.

EXERCISE:

Change the location of poles from L.H.S of s-plane to y axis first, and then to R.H.S of s-plane and observe the effects.

LAB SESSION 08

OBJECTIVE:

To study z-plane and plot impulse and frequency response for different pole zero location in z-plane.Also to determine weather system is FIR or IIR.

THEORY:

The z - Transform of a general discrete time signal x(n) is defined as;

$$X\ (z) = \sum_{n=o}^{\infty} x(n)\ z^{-n}$$

Where the complex variable z=r $\angle w$, with r the radius and w the angle. DTFT is a subset of z transform when r =1. Since ‘r’ information is not present in DTFT, therefore information about stability in discrete time can only be obtained from z transform. If pole lies inside the unit circle, system is stable. If pole lies outside the unit circle, system is unstable. If pole lies at the unit circle, system is marginally stable or oscillatory. If system has FIR, it is stable. If system is IIR, it can be stable or unstable.

PROCEDURE:

1. Generate pole zero constellation in z plane.
2. Plot corresponding Frequency (Bode magnitude) response.
3. Plot impulse response and determine that the system is FIR or IIR.
4. Modify location of poles in z plane to observe the corresponding change in frequency and impulse response.

STEPS:

1. Make a folder at desktop and name it as your current directory within MATLAB.
2. Open M-file editor and write the following code:

```
clear all;
close all;
clc;
Num = poly([(0-(i*(pi/2))),(0+(i*(pi/2)))]);


Den = poly([-1,-1]);
Num1 = poly([j,-j]);
Den1 = poly([exp(-1),exp(-1)]);
sys1=tf(Num1,Den1,1)
```

```
figure;
subplot(3,1,1);
pzmap(sys1);
xlim([-2 2]);
ylim([-4 4]);
subplot(3,1,2);
[mag phase w]=bode(sys1);

mag=squeeze(mag);
plot(w,mag);
xlim([0 100])
subplot(3,1,3);
impulse(sys1);
H=dfilt.df1(Num,Den);
A=isfir(H)
figure;
pzmap(sys1)
grid on;
```

3. Save the file as **P010.m** in your current directory and 'run' it.

RESULT:

1 Learn the specific logical bits of the code and make notes.
2 Observe the plots.
3 Now, explain (write) in your own words the cause and effects of what you just saw.

EXERCISE:

Change the location of poles from inside the unit circle to outside and at the unit circle and observe and note the changes.

LAB SESSION 09

OBJECTIVE:

Object of this lab is introduction to digital filters and its types, design FIR filter and study how it performs filtering on a signal. Further truncate different types of FIR filter like Low Pass, High Pass, Band Pass using different windows like rectangular, Kaiser Etc. and compare the results obtained from different windows.

THEORY:

The process of deriving a realizable transfer function of a digital filter by considering given frequency response specifications is known as digital filter design. The digital filter can be classified as:

1. Finite –duration impulse response (FIR) filter
2. Infinite –duration impulse response (IIR) filter

In MATLAB, there are built in functions which can be used to design digital filter like IIR and FIR.

The different types of FIR filters are listed as follows:

- Window techniques based FIR filter.
  1. Rectangular windows.
  2. Hamming window.
  3. Hanning window.
  4. Blackman window.
  5. Barlett window.
  6. Kaiser window.
- Equiripple linear phase FIR filter.
- Least square error FIR filters.

The different types of IIR filters are listed as follows:

- Butterworth filter
- Chebyshev Type I filter
- Chebyshev Type II filter
- Elliptic filter

FIR digital filter operates on digital sample values. It uses current and past input samples to produce a current output sample. It does not use previous output samples. There are various types of FIR filter based on need viz. low pass, high pass, band pass and band stop, Low pass filter.

Following points are usually considered to design FIR filter other the window type.

INPUT:

- Window Type
- Passband and stopband ripples
- passband and stopband edge frequencies
- sampling frequency
- order of the filter
- window coefficients

OUTPUT:

- magnitude and phase responses

COMPARISON OF FIR AND IIR FILTERS

1. FIR filters are Finite Impulse Response filters with no feedback, whereas IIR contains feedback.
2. Transfer function of FIR filter does not contain any non-trivial poles. Their frequency response is solely dependent on zero locations. IIR filters contain poles as well as zeros.
3. As there are no poles, FIR filters cannot become unstable; however, IIR filters can become unstable if any pole lies outside the unit circle in z-plane.
4. More number of coefficients is needed to design the desired filter in FIR than IIR.

PROCEDURE:

TASK-1

1. Create a signal vector containing two frequencies as:
   i) 100 Hz. and ii) 150 Hz. with Fs = 1000 Hz.
2. Design two band pass FIR filters with 64 coefficients and with pass bands as i) 125 to 175 Hz. and ii) 75 to 125 Hz.
3. Use both filters on the created signal and observe their outputs.
4. Plot frequency responses and pole-zero constellations of both filters and note observations.

```
close all; clear all; clc; % Frequencies in Hz.



F1 = 100;  F2 = 150;

% Sampling Frequency in samples/sec.
```

```
Fs = 1000;

t = [0 : 1/Fs : 1]; % Time Vector
F = Fs*[0:length(t)-1]/length(t); % Frequency Vector

x = exp(j*2*pi*F1*t)+2*exp(j*2*pi*F2*t); % Signal Vector

bh = fir1( 64 , [125  175]/500); % filter coeffs.
     bl = fir1( 64 , [75  125]/500); % filter coeffs.
     [hh,wh]=freqz(bh,1,length(t),'whole'); % Frequency
     response for filter 1
     [hl,wl]=freqz(bl,1,length(t),'whole'); % Frequency
     response for filter 2
     % Filter operation - see filtfilt in help to learn what it
     does

     yh = filtfilt(bh,1,x);
     yl = filtfilt(bl,1,x);

     % Plotting

     figure, subplot(5,1,1),

     plot(F,abs(fft(x)));

     xlim([0 Fs/2]);
     title('FFT of original signal');

     subplot(5,1,2),
     plot(F,abs(hh));
     xlim([0 Fs/2]);
     title('Frequency response of Filter One');

     subplot(5,1,3),
     plot(F,abs(fft(yh)));
     xlim([0 Fs/2]);
     title('FFT of filtered signal from filter one');

     subplot(5,1,4),
     plot(F,abs(hl));
     xlim([0 Fs/2]);
     title('Frequency response of Filter Two');

     subplot(5,1,5),
     plot(F,abs(fft(yl)));
     xlim([0 Fs/2]);
     title('FFT of filtered signal from filter two');
     xlabel('Hz.')
     % Pole Zero Constellations

     [bh,ah] = eqtflength(bh,1);
     [zh,ph,kh] = tf2zp(bh,ah);
     [bl,al] = eqtflength(bl,1);
     [zl,pl,kl] = tf2zp(bl,al);

     figure,
     subplot(1,2,1),
     pzplane(bh,ah);
     xlim([-1.5 1.5]);
     ylim([-1.5 1.5]);
     title('Filter_One');
     subplot(1,2,2),
     pzplane(bl,al);
     xlim([-1.5 1.5]);
     ylim([-1.5 1.5]);
     title('Filter Two');
```

TASK -2

Write a program to design a FIR filter using Hanning windows,take inputs from user for design values of filter.

```
close all;
clear all;
clc;

fp=input('Enter the pass band frequency');
fs=input('Enter the stop band frequency');
rp=input(' Enter the pass band attenuation');
rs=input('Enter the stop band attenuation');
f=input(' Enter the sampling frequency');

% Calculating filter order

num=-20*log10(sqrt(rp*rs))-13;
dem=14.6*(fs-fp)/f;
n=ceil(num/dem);
n=abs(n);

% Normalizing the frequencies

wp=2*fp/f;
ws=2*fs/f;
wn=(ws+wp)/2;
```

```
%Adjusting the filter order. The order of window must be an odd
number
%and the order of filter must be one less than that of the
window

if (rem(n,2)==0)
    m=n+1;
else
        m=n;
        n=n-1;
end

%Window sequence calculation

w=hann(m);

%Calculation of filter coefficients

b=fir1(n,wn,'low',w);

%Plotting the filter response

freqz(b,1,500,3000);
TITLE('Magnitude and Phase response');
```

TASK-3

Write a program for FIR(Finite Impulse Response) filter like Low pass FIR filter, High pass FIR filter, Band pass FIR filter and Band stop FIR filter using Rectangular window using MATLAB .

ALGORITHM:
LOW PASS FILTER:
Step 1: Read the input sequence
Step 2: Perform low pass filter calculations
Step 3: Plot the output sequences

HIGH PASS FILTER:
Step 1: Read the input sequence
Step 2: Perform high pass filter calculations
Step 3: Plot the output sequences

BAND PASS FILTER:
Step 1: Read the input sequence
Step 2: Perform band pass filter calculations

Step 3: Plot the output sequences

BAND STOP FILTER:
Step 1: Read the input sequence
Step 2: Perform band stop filter calculations
Step 3: Plot the output sequences

PROGRAM:

```
clc;
clear all;
close all;
rp=input('Enter the passband ripple(rp):');
rs=input('Enter the stopband ripple(rs):');
fp=input('Enter the passband frequency(fp):');
fs=input('Enter the stopband frequency(fs):');
f=input('Enter the sampling frequency(f):');
wp=2*fp/f;
ws=2*fs/f;
num=-20*log10(sqrt(rp*rs))-13;
dem=14.6*(fs-fp)/f;
n=ceil(num/dem);
n1=n+1;
if(rem(n,2)~=0)
    n1=n;
    n=n-1;
end
y=boxcar(n1);
%Low pass filter
b=fir1(n,wp,y);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));
subplot(2,2,1);
plot(m);
ylabel('Gain(db)->');
xlabel('(a)Normalised frequency->');
%High pass filter
b=fir1(n,wp,'high',y);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));

subplot(2,2,2);
plot(m);
ylabel('Gain(db)');
xlabel('(b)Normalised frequency');
%Band pass filter
```

```
wn=[wp*ws];
b=fir1(n,wn,y);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));
subplot(2,2,3);
plot(m);
ylabel('Gain(db)');
xlabel('(c)Normalised frequency');
%Band stop filter=============
wn=[wp*ws];
b=fir1(n,wn,'stop',y);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));
subplot(2,2,4);
plot(m);
ylabel('Gain(db)');
xlabel('(d)Normalised frequency-');
```

EXERCISE:

Q1. Perform Q3.using Hamming and Kaiser Window.
Compare results of designed filters using three different windows on a single plot.

NED University of Engineering & Technology
Department of ____________________ Engineering

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Software Menu Identification and Usage: Ability to initialise, configure and *operate* software environment *under supervision*, using menus, shortcuts, instructions etc.	Unable to understand and use software menu	Little ability and understanding of software menu operation, makes many mistake	Moderate ability and understanding of software menu operation, makes lesser mistakes	Reasonable understanding of software menu operation, makes no major mistakes	Demonstrates command over software menu usage with frequent use of advance menu options
Procedural Programming of given model: *Practice* procedural programming techniques, in order to code specific model	Little to no understanding of procedural programming techniques	Slight ability to use procedural programming techniques for coding given algorithm	Mostly correct recognition and application of procedural programming techniques but makes crucial errors for the given model	Correctly recognises and uses procedural programming techniques with no errors but unable to run model successfully	Correctly recognises and uses procedural programming techniques with no errors and runs model successfully
Relating Theoretical Concepts, Equations and Transforms to Code: *Recognise* relation between model concepts and written code and *manipulate* the code in accordance of requirements	Completely unable to relate between model concepts and written code, unable to do manipulations	Able to recognise some relation between model concepts and written code, unable to do manipulations	Able to recognise relation between model concepts and written code, unable to do manipulations	Able to recognise relation between model concepts and written code, able to do some manipulations	Able to recognise relation between model concepts and written code, able to completely manipulate code in line with theoretical concepts
Detecting and Removing Errors: *Detect* Errors/Exceptions and in simulation and *manipulate* code to rectify the simulation	Unable to check and detect error messages and indications in software	Able to find error messages and indications in software but no understanding of detecting those errors and their types	Able to find error messages and indications in software as well as understanding of detecting some of those errors and their types	Able to find error messages in software as well as understanding of detecting all of those errors and their types	Able to find error messages in software along with the understanding to detect and rectify them

Graphical Visualisation and Comparison of model **Parameters:** *Manipulate* given simulation under supervision, in order to produce graphs/plots for measuring and comparing model parameters	Unable to understand and utilise visualisation or plotting features	Ability to understand and utilise visualisation and plotting features with frequent errors	Ability to understand and utilise visualisation and plotting features successfully but unable to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, partially able to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, also able to compare and analyse them
Following step-by-step procedure to complete lab work: *Observe, imitate and operate* software to complete the provided sequence of steps	Inability to recognise and perform given lab procedures	Able to recognise given lab procedures and perform them but could not follow the prescribed order of steps	Able to recognise given lab procedures and perform them by following prescribed order of steps, with frequent mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with occasional mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with no mistakes
Recording Simulation Observations: *Observe and copy* prescribed or required simulation results in accordance with lab manual instructions	Inability to recognise prescribed or required simulation measurements	Able to recognise prescribed or required simulation measurements but does not record according to given instructions	—	Able to recognise prescribed or required simulation measurements but records them incompletely	Able to recognise prescribed or required simulation measurements and records them completely, in tabular form
Discussion and Conclusion: *Demonstrate* discussion capacity on the recorded observations and draw conclusions from it, relating them to theoretical principles/concepts	Complete inability to discuss recorded observations and draw conclusions	Slight ability to discuss recorded observations and draw conclusions	Moderate ability to discuss recorded observations and draw conclusions	Reasonable ability to discuss recorded observations and draw conclusions	Full ability to discuss recorded observations and draw conclusions

LAB SESSION 10

OBJECTIVE:

Object of this lab is to design different IIR filter using FDA tool.

THEORY:

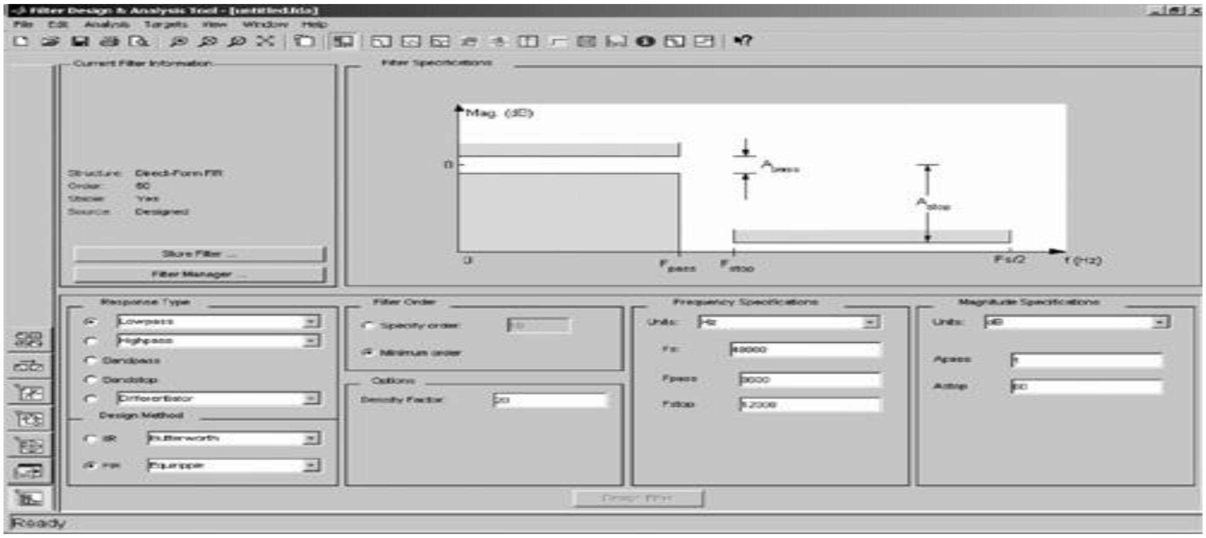
Filter Design and Analysis Tool (FDA Tool) is Graphic User Interface for designing and analyzing filters. It is used to design FIR and IIR filters by entering the desired filter specifications, or by importing filter from MATLAB workspace or by adding, moving or deleting poles and zeros. After designing a filter, the response can be viewed and analyses in other Graphic User Interface tool named Filter Visualization Tool (FV Tool) linked with FDA Tool. The different types of responses that can be viewed are listed below:

- Magnitude response
- Phase response
- Group delay
- Phase delay
- Impulse response
- Step response
- Pole-zero plot
- Zero-phase plot

OPENING FDA TOOL WINDOW:

FDA Tool can be opened using command:

```
fdatool
```


Filter Design & Analysis Tool - [untitled.fda]
File Edit Analysis Targets View Window Help
Current Filter Information
Filter Specifications
Mag. (dB)
0
Apass
Astop
0
Fpass
Fstop
Fs/2
f (Hz)
Store Filter ...
Filter Manager ...
Design Filter
Ready

Structure: Direct-Form FIR
Order: 60
Stable: Yes
Source: Designed

Response Type
● Lowpass
○ Highpass
○ Bandpass
○ Bandstop
○ Differentiator

Design Method
○ IIR Butterworth
● FIR Equiripple

Filter Order
○ Specify order: 10
● Minimum order

Options
Density Factor: 20

Frequency Specifications
Units: Hz
Fs: 48000
Fpass: 9600
Fstop: 12000

Magnitude Specifications
Units: dB
Apass: 1
Astop: 80

Figure A

The different steps involved in designing a filter using FDA Tool can be listed as:

1. Selection of type of response required
2. Selection of type of filter design
3. Specifying the filter order
4. Entering the filter specifications
5. Entering the magnitude specifications

After providing the information listed above, filter can be designed and its response can be viewed and analysed.

The complete description of the FDA Tool window and different steps required to design a filter are elaborated below:

1. **Selecting response type**: The desired response type is selected from the list of available options, i.e., lowpass, highpass, bandpass, bandstop, differentiation, multiband, peaking etc.

2. **Type of design method**: The design can be of FIR or IIR filter. Depending upon whether FIR Or IIR filter design is selected, further options are available in the dropdown menu. In IIR filter design, the different options available in dropdown menu are as given below:

    - Butterworth
    - Chebyshev type I
    - Chebyshev type II
    - Elliptic
    - Maximally flat
    - Least Pth-norm
    - Const least Pth-norm

    In FIR filter design the options available are listed as follows:

    - Equirriple
    - Least square
    - Window
    - Const least squares
    - Complex equiripple
    - Least Pth norm
    - Constrained equiripple
    - Generalized equiripple
    - Constrained band equirriple
    - Interpolated FIR

    The options available depend upon the selection of response type.

3. **Filter order**: Under this, two options available are as

    - User-defined order: Under this option user has to enter the order of the filter.
    - Minimum order: The other option available for selecting the filter order is minimum order. It is calculated by system itself.

4. **Filter specifications**: Depending upon the response type and design method selected, the graphical representation of generalized filter specifications appear in the display region of FDA Tool. These specifications are ‘Frequency Specifications’ and ‘Magnitude Specification’.

    These specifications are provided by the user, as per filter design requirement, in the appropriate blocks.

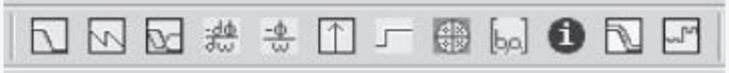
5. **Designing filter**: After all the requisite information is entered, a filter can be designed by clicking the ‘Design Filter’ button available at the bottom of the window. Filter | coefficients are calculated and magnitude response appears in the display region.

(Note: ‘Design Filter’ button will be disabled once the filter coefficients are computed. This button will be enabled again in case any changes are made in the filter specifications.)

6. **Displaying filter responses:** Once the filter coefficients are calcu

lated as per the specifications provided by the user, the display region will show magnitude response of the designed filter. The other filter response characteristics can be viewed in the display region or FV Tool. The response to be viewed can be selected from the different icons displayed on the toolbar shown in Figure below.

Figure : Different Response Icons on the Toolbar

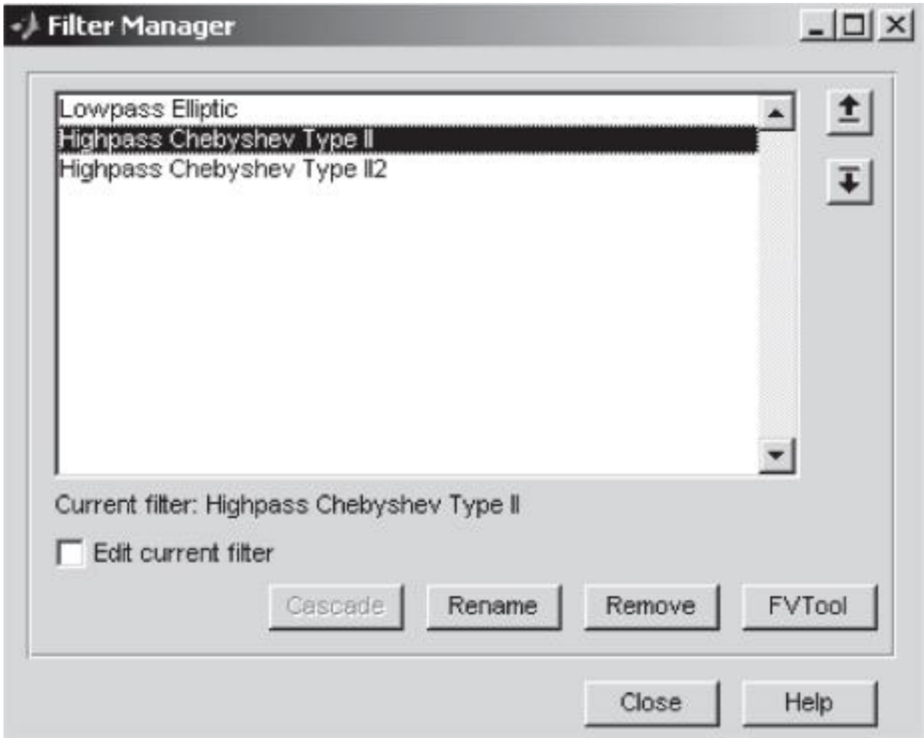


(NOTE: The different responses for display can also be selected from the ‘Analysis’ menu on menu bar.)

7. **Current filter information**: The information about the designed filter is given in the ‘Current Filter Information’ region of FDA Tool window as shown in Figure A The information provided is about the ‘structure’, ‘order’, ‘stability’ and ‘source’

- Storing a filter
  The designed filter is stored by clicking ‘Store Filter’ button in the ‘Current Filter Information’ region.
- Filter manager
  The ‘Filter Manager’ button opens up a new Filter Manager window (Figure B) showing the list of filters stored. This window also has options as: Edit current filter, Cascade, Rename, Remove and FV Tool.

  Figure B Filter Manager Window



Lowpass Elliptic
Highpass Chebyshev Type II
Highpass Chebyshev Type II2

Current filter: Highpass Chebyshev Type II

□ Edit current filter

To cascade two or more filters, highlight the designed filters and press 'Cascade' button. A new cascaded filter is added to the 'Filter Manager'.

8. Saving and opening filter design session:
The filter design session is saved as MAT-file and can be used later on. It can be saved by clicking save icon or selecting save session option in File menu and giving desired session name. Similarly, the saved session can be opened by clicking open icon or by selecting open option in file menu and selecting the previously saved filter design session.

FILTER VISUALIZATION TOOL:

The response characteristics can be viewed in a separate window by selecting the 'Filter Visualization Tool' (FV Tool) from 'view' menu or clicking the 'Full View Analysis' button on the toolbar. The FV Tool window is shown in Figure C

FV Tool has most of the menus on the menu bar and icons on the toolbar similar to that FDA Tool with some additional icons which are mainly used to work with representation of the responses.

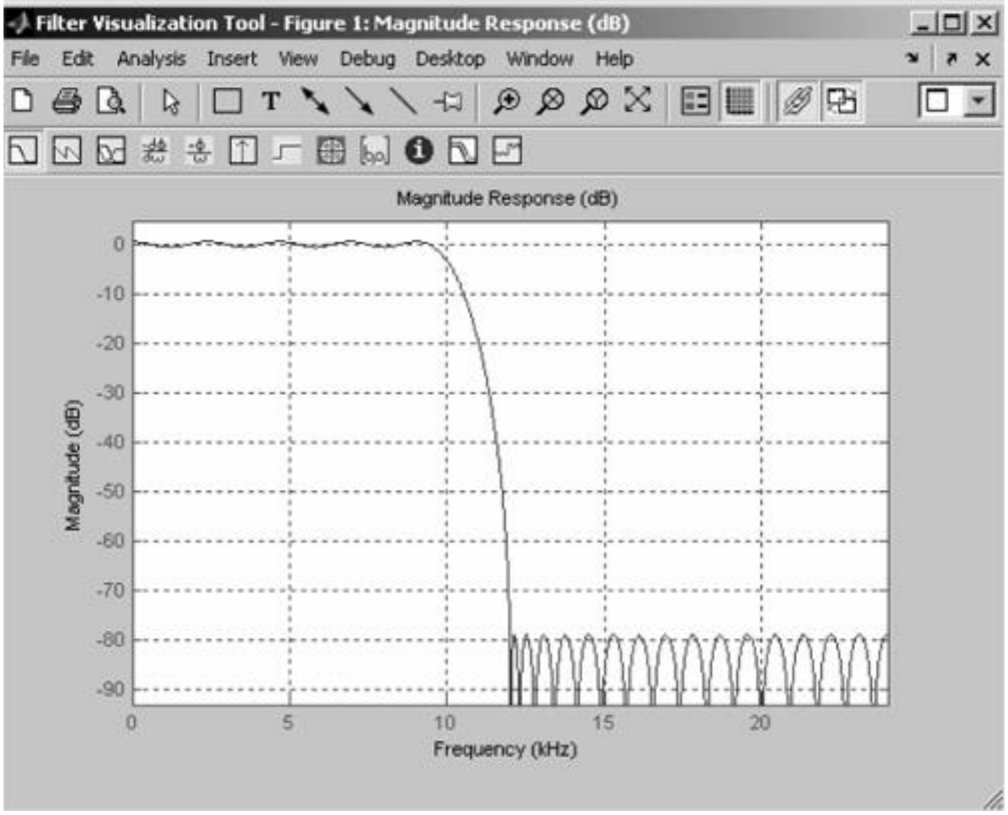

Filter Visualization Tool - Figure 1: Magnitude Response (dB)
File Edit Analysis Insert View Debug Desktop Window Help
Magnitude Response (dB)
Magnitude (dB)
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
0
5
10
15
20
Frequency (kHz)

Figure C Filter Visualization Tool(FV Tool) window

IIR FILTER DESIGN USING FDA TOOL

TASK-1

Design an IIR Butterworth band pass filter with the following specifications:

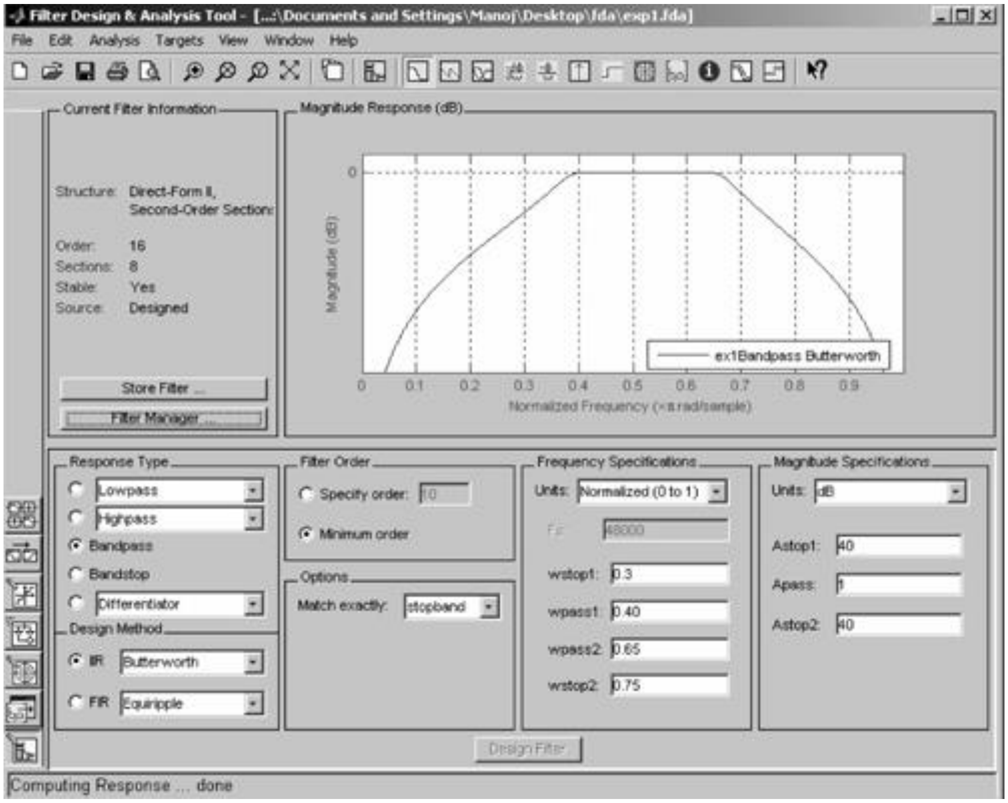
Normalized pass band edges at 0.40 and 0.65
Normalized stop band edges at 0.3 and 0.75
Pass band ripple 1 dB
Minimum stop band attenuation 40 dB

Show (i) Magnitude response (ii) Phase response (iii) Group delay (iv) Phase delay response.

Solution:

As per the given specifications, the requisite data is entered in new FDA Tool window as shown in Figure

Figure. FDA Tool Window Showing Specification Entered and Magnitude Response for Task-1.



Filter Design & Analysis Tool - [...\Documents and Settings\Manoj\Desktop\fda\exp1.fda]

Current Filter Information

Structure: Direct-Form II, Second-Order Sections

Order: 16
Sections: 8
Stable: Yes
Source: Designed

Response Type: ○ Lowpass ○ Highpass ● Bandpass ○ Bandstop ○ Differentiator

Design Method: ● IIR Butterworth ○ FIR Equiripple

Filter Order: ○ Specify order: 10 ● Minimum order

Options: Match exactly: stopband

Frequency Specifications

Units: Normalized (0 to 1)
Fs: 48000
wstop1: 0.3
wpass1: 0.40
wpass2: 0.65
wstop2: 0.75

Magnitude Specifications

Units: dB
Astop1: 40
Apass: 1
Astop2: 40

Computing Response ... done

The filter is designed for minimum order so as to reduce the complexity of the design. In case, it has to be designed for user defined order, then the order of the filter has to be calculated first by user using appropriate formulas or MATLAB function.

The other responses can be viewed by clicking on the appropriate icon on the toolbar and responses obtained are shown in Figures below

Figure Magnitude Response in dB

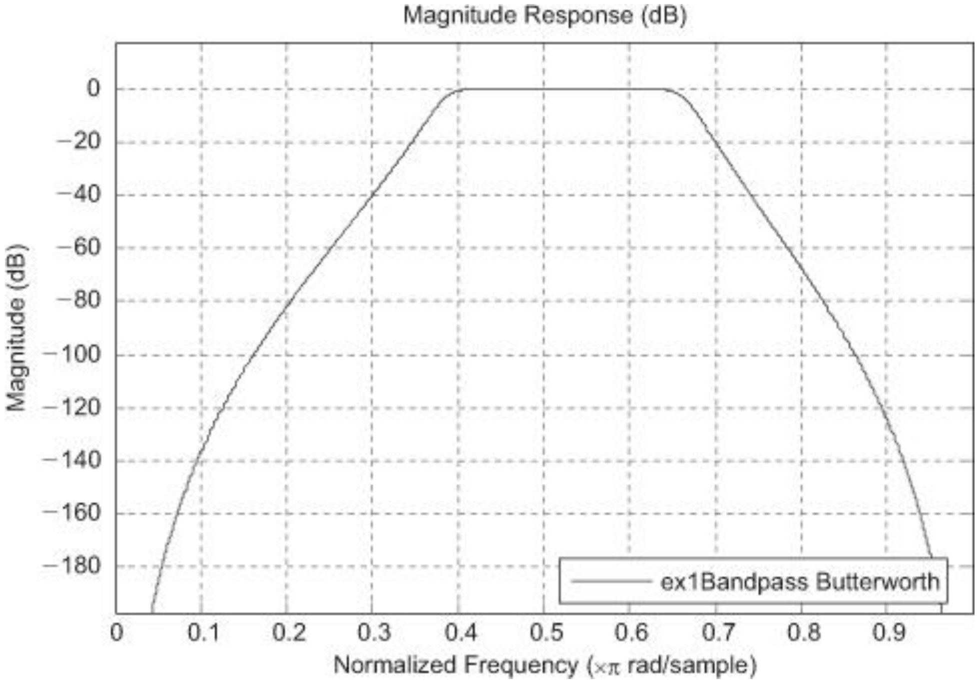


Figure Phase Response

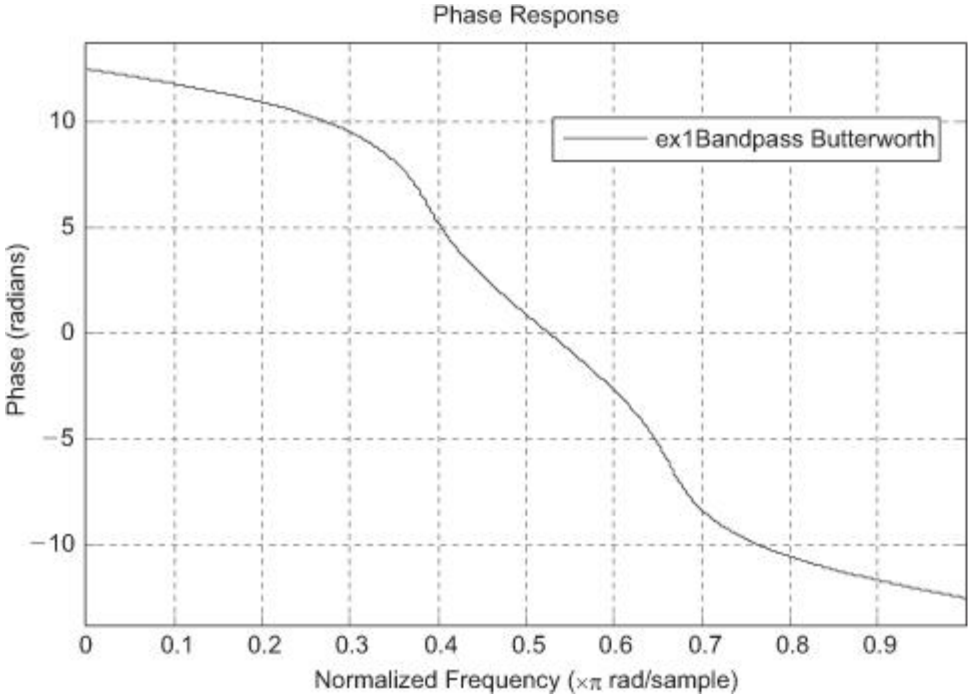


Figure Group Delay Response

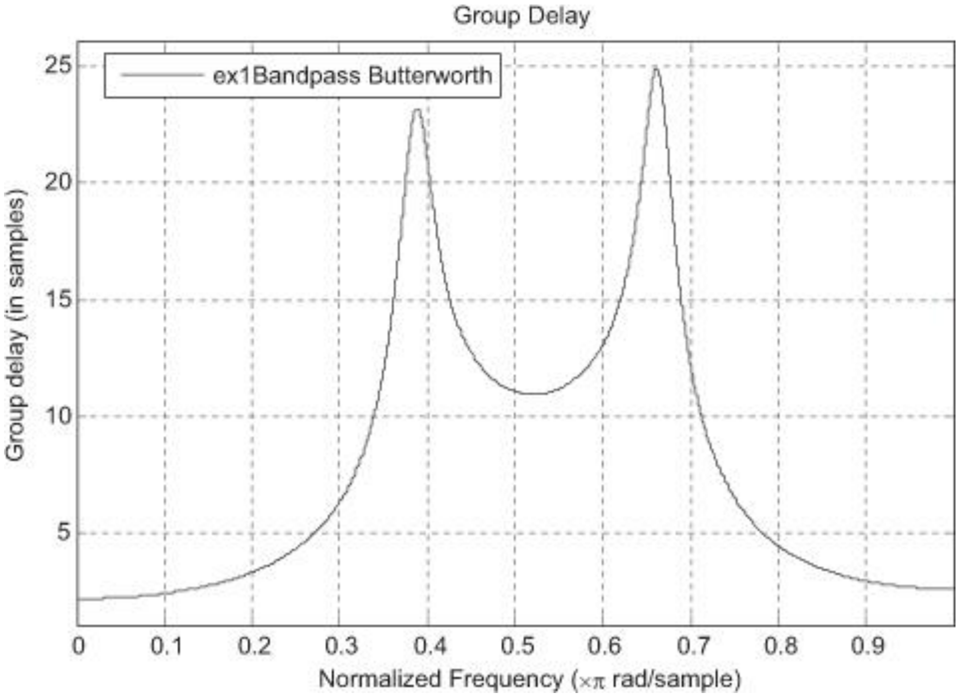
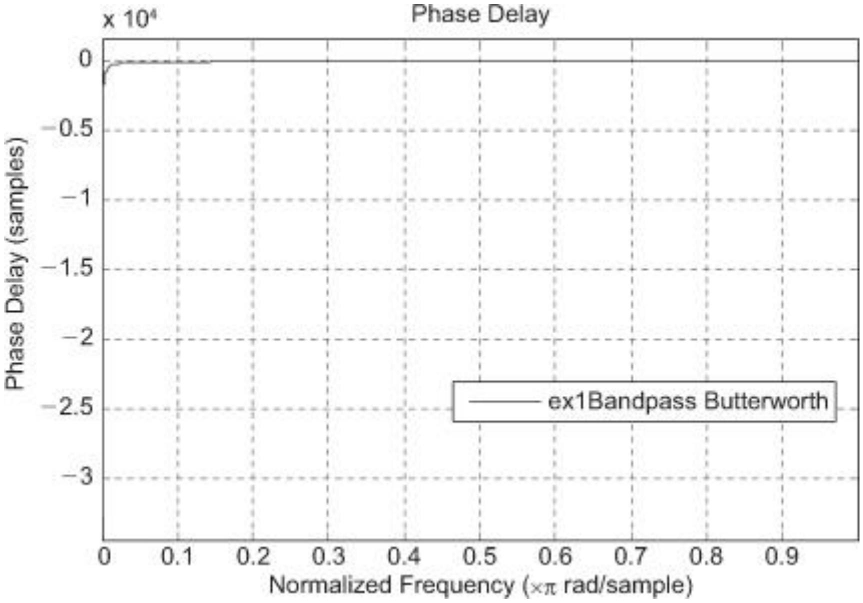


Figure. Phase Delay



These responses can also be viewed in FV Tool.

TASK-2

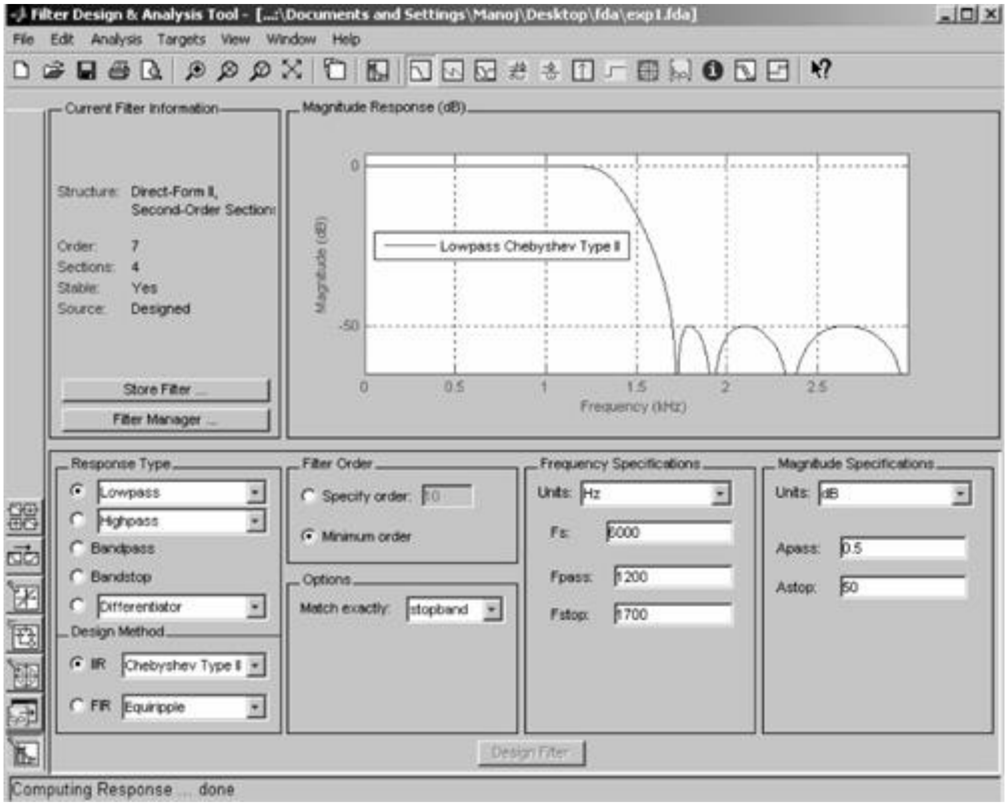
Design a Type II Chebyshev IIR lowpass filter with the following specifications:

Passband frequency	1,200 Hz
Stopband frequency	1,700 Hz
Sampling frequency	6,000 Hz
Passband ripple	0.50 dB

1. Show magnitude response.
2. Show pole/zero plot.

Solution: FDA Tool Window showing given specifications duly entered and magnitude response in response display region is shown in Figure.

Figure. FDA Tool Window for Example



Filter Design & Analysis Tool - [...:\Documents and Settings\Manoj\Desktop\fda\exp1.fda]

Current Filter Information

Structure: Direct-Form II, Second-Order Section:

Order: 7

Sections: 4

Stable: Yes

Source: Designed

Response Type: ● Lowpass

Design Method: ● IIR Chebyshev Type II

Filter Order: ● Minimum order

Options: Match exactly: stopband

Frequency Specifications: Units: Hz; Fs: 6000; Fpass: 1200; Fstop: 1700

Magnitude Specifications: Units: dB; Apass: 0.5; Astop: 50

Computing Response ... done

By using the FV Tool, the magnitude response and pole/zero plot is obtained as separate figures and is shown in Figures.

Figure. Magnitude Response for Task-2.

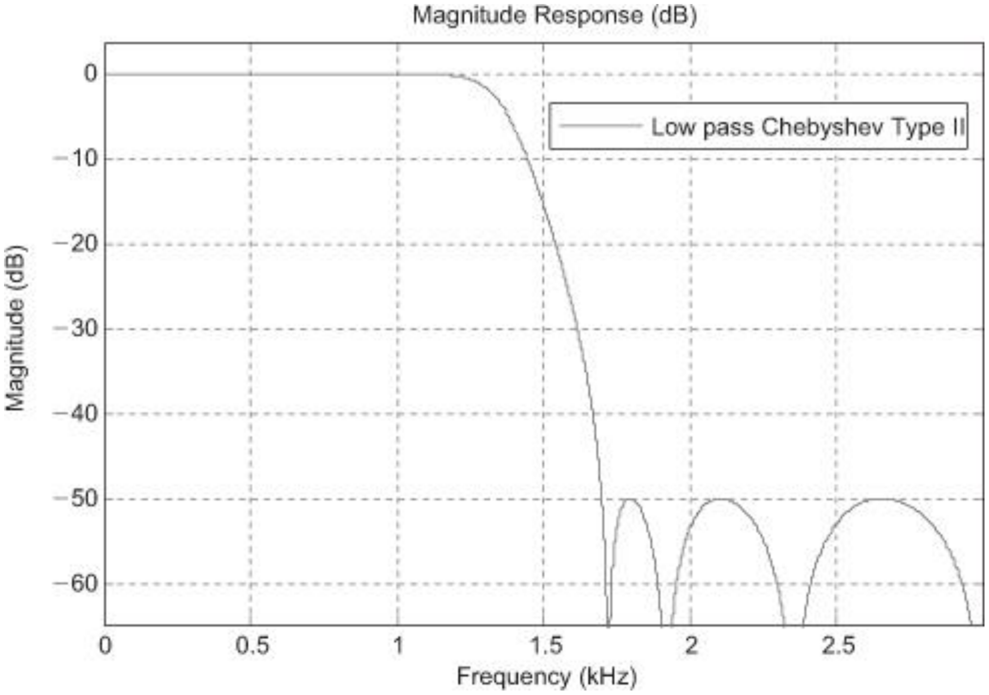
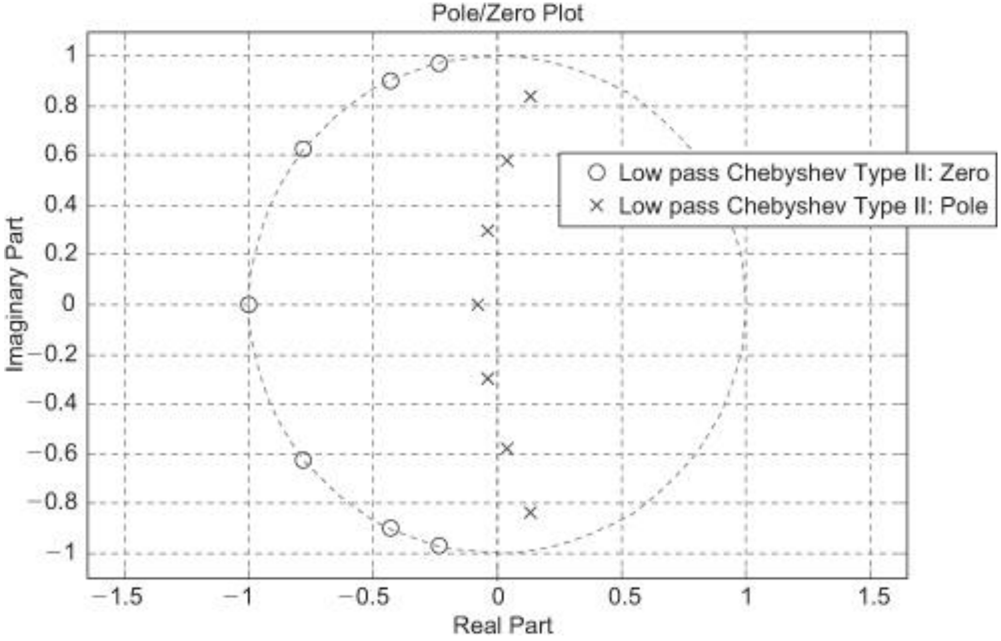


Figure Pole/zero Plot for Q2



TASK-3:

Design an elliptic IIR low pass filter with following specifications:

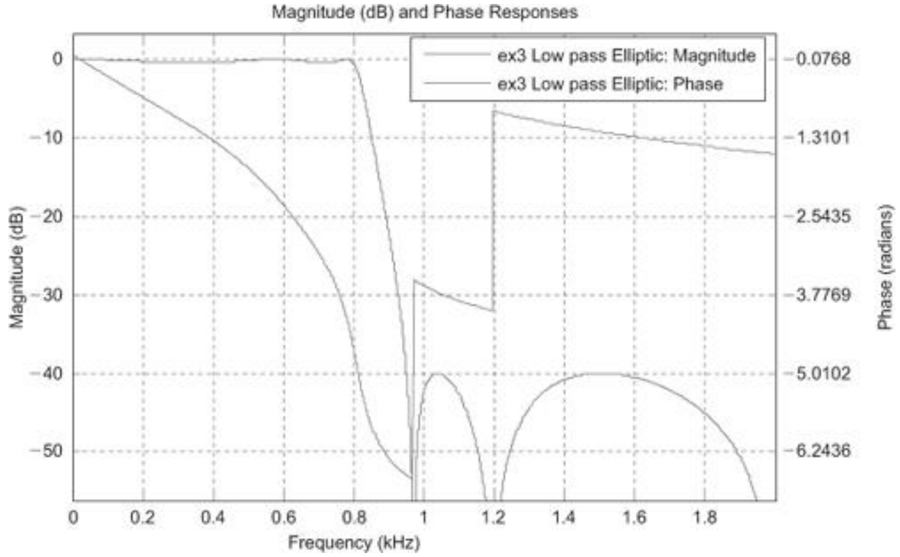
Pass band ripple	0.5 dB
Stop band attenuation	40 dB
Pass band frequency	800 Hz
Stop band frequency	1,000 Hz
Sampling frequency	4,000 Hz

1. Show magnitude and phase response in the same window.
2. Obtain filter information.

Solution:
As per given specifications, the requisite data is entered in FDA Tool window. By clicking appropriate icon on toolbar, the magnitude and phase responses are obtained in the same window.

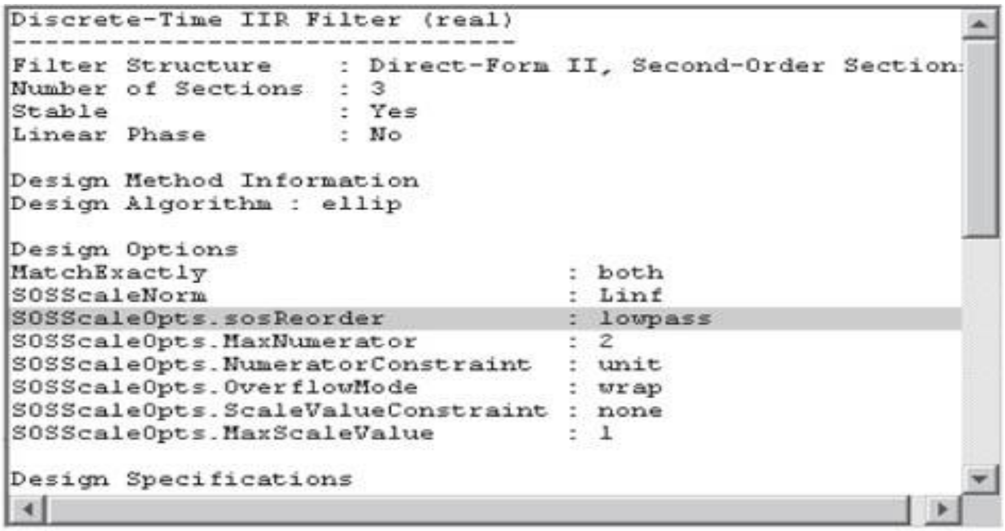
1. These magnitude and phase responses obtained are viewed in FV Tool window also and are shown in Figure .

Figure . Magnitude and Phase Response for Task-3.



2. To obtain the information about the filter ‘Filter Information’ icon on Toolbar of FDA Tool Window is clicked or ‘Filter Information’ option is selected from ‘Analysis’ menu. The detailed filter information appears in the display region as shown in Figure a, b and c.

Figure 'Filter information' for Task-2



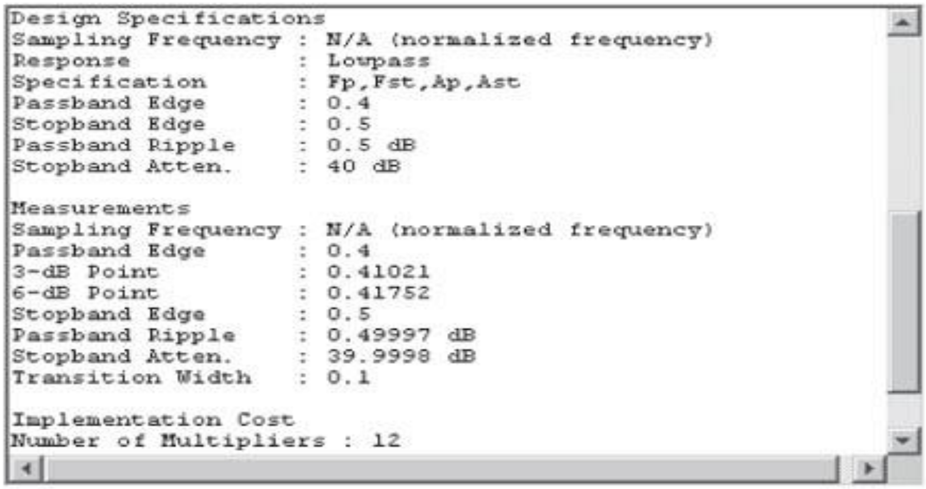
```
Discrete-Time IIR Filter (real)
-------------------------------
Filter Structure    : Direct-Form II, Second-Order Section:
Number of Sections  : 3
Stable              : Yes
Linear Phase        : No

Design Method Information
Design Algorithm : ellip

Design Options
MatchExactly                        : both
SOSScaleNorm                        : Linf
SOSScaleOpts.sosReorder             : lowpass
SOSScaleOpts.MaxNumerator           : 2
SOSScaleOpts.NumeratorConstraint    : unit
SOSScaleOpts.OverflowMode           : wrap
SOSScaleOpts.ScaleValueConstraint : none
SOSScaleOpts.MaxScaleValue          : 1

Design Specifications
```

(a)

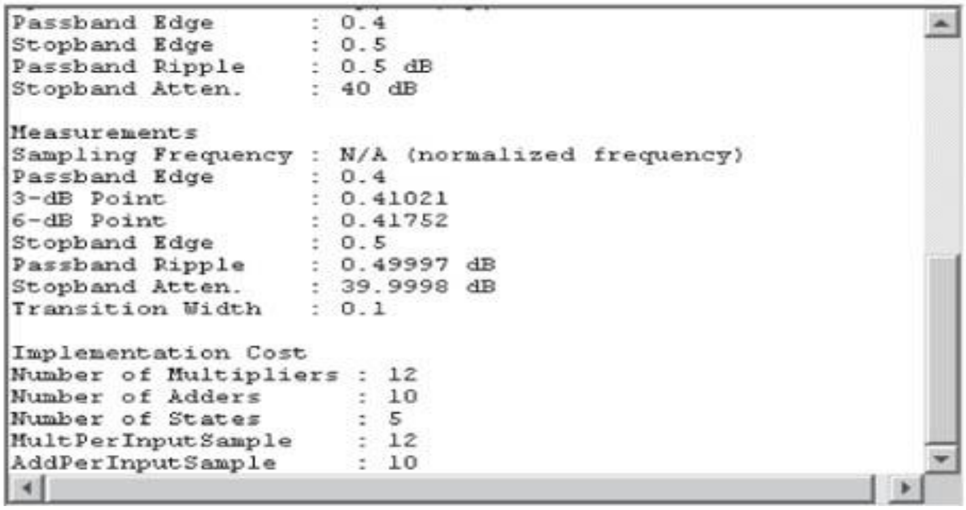


```
Design Specifications
Sampling Frequency : N/A (normalized frequency)
Response           : Lowpass
Specification      : Fp,Fst,Ap,Ast
Passband Edge      : 0.4
Stopband Edge      : 0.5
Passband Ripple    : 0.5 dB
Stopband Atten.    : 40 dB

Measurements
Sampling Frequency : N/A (normalized frequency)
Passband Edge      : 0.4
3-dB Point         : 0.41021
6-dB Point         : 0.41752
Stopband Edge      : 0.5
Passband Ripple    : 0.49997 dB
Stopband Atten.    : 39.9998 dB
Transition Width   : 0.1

Implementation Cost
Number of Multipliers : 12
```

(b)



```
Passband Edge      : 0.4
Stopband Edge      : 0.5
Passband Ripple    : 0.5 dB
Stopband Atten.    : 40 dB

Measurements
Sampling Frequency : N/A (normalized frequency)
Passband Edge      : 0.4
3-dB Point         : 0.41021
6-dB Point         : 0.41752
Stopband Edge      : 0.5
Passband Ripple    : 0.49997 dB
Stopband Atten.    : 39.9998 dB
Transition Width   : 0.1

Implementation Cost
Number of Multipliers : 12
Number of Adders      : 10
Number of States      : 5
MultPerInputSample    : 12
AddPerInputSample     : 10
```

(c)

The filter information is obtained by scrolling down the text in the window shown in Figure 15.41b.

EXERCISE:

Record Your Voice at home while turn any motor of your house ON.
Design a filter using FDA Tool.

Remove sound of motor from recorded signal.

Listen the output signal.

NED University of Engineering & Technology
Department of ____________________ Engineering

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Software Menu Identification and Usage: Ability to initialise, configure and *operate* software environment *under supervision*, using menus, shortcuts, instructions etc.	Unable to understand and use software menu	Little ability and understanding of software menu operation, makes many mistake	Moderate ability and understanding of software menu operation, makes lesser mistakes	Reasonable understanding of software menu operation, makes no major mistakes	Demonstrates command over software menu usage with frequent use of advance menu options
Procedural Programming of given model: *Practice* procedural programming techniques, in order to code specific model	Little to no understanding of procedural programming techniques	Slight ability to use procedural programming techniques for coding given algorithm	Mostly correct recognition and application of procedural programming techniques but makes crucial errors for the given model	Correctly recognises and uses procedural programming techniques with no errors but unable to run model successfully	Correctly recognises and uses procedural programming techniques with no errors and runs model successfully
Relating Theoretical Concepts, Equations and Transforms to Code: *Recognise* relation between model concepts and written code and *manipulate* the code in accordance of requirements	Completely unable to relate between model concepts and written code, unable to do manipulations	Able to recognise some relation between model concepts and written code, unable to do manipulations	Able to recognise relation between model concepts and written code, unable to do manipulations	Able to recognise relation between model concepts and written code, able to do some manipulations	Able to recognise relation between model concepts and written code, able to completely manipulate code in line with theoretical concepts
Detecting and Removing Errors: *Detect* Errors/Exceptions and in simulation and *manipulate* code to rectify the simulation	Unable to check and detect error messages and indications in software	Able to find error messages and indications in software but no understanding of detecting those errors and their types	Able to find error messages and indications in software as well as understanding of detecting some of those errors and their types	Able to find error messages in software as well as understanding of detecting all of those errors and their types	Able to find error messages in software along with the understanding to detect and rectify them

Graphical Visualisation and Comparison of model **Parameters:** *Manipulate* given simulation under supervision, in order to produce graphs/plots for measuring and comparing model parameters	Unable to understand and utilise visualisation or plotting features	Ability to understand and utilise visualisation and plotting features with frequent errors	Ability to understand and utilise visualisation and plotting features successfully but unable to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, partially able to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, also able to compare and analyse them
Following step-by-step procedure to complete lab work: *Observe, imitate and operate* software to complete the provided sequence of steps	Inability to recognise and perform given lab procedures	Able to recognise given lab procedures and perform them but could not follow the prescribed order of steps	Able to recognise given lab procedures and perform them by following prescribed order of steps, with frequent mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with occasional mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with no mistakes
Recording Simulation Observations: *Observe and copy* prescribed or required simulation results in accordance with lab manual instructions	Inability to recognise prescribed or required simulation measurements	Able to recognise prescribed or required simulation measurements but does not record according to given instructions	—	Able to recognise prescribed or required simulation measurements but records them incompletely	Able to recognise prescribed or required simulation measurements and records them completely, in tabular form
Discussion and Conclusion: *Demonstrate* discussion capacity on the recorded observations and draw conclusions from it, relating them to theoretical principles/concepts	Complete inability to discuss recorded observations and draw conclusions	Slight ability to discuss recorded observations and draw conclusions	Moderate ability to discuss recorded observations and draw conclusions	Reasonable ability to discuss recorded observations and draw conclusions	Full ability to discuss recorded observations and draw conclusions

(Open Ended Lab Part 01)

Objective:

To convert an analog (voltage & current) signal into digital signal using ADC (audio card) and display it on MATLAB Simulink environment.

Required Components:

1. Audio Card
2. Transformer (220V/12V)
3. Resistors (for VDR)
4. Veroboard
5. Audio jack
6. PC with MATLAB environment

Procedure:

- Using Transformer convert 220VAC from mains into 12VAC.
- Using VDR convert 12VAC to a voltage compatible to audio card (show all the calculations of resistances with their power ratings).
- Set the sampling frequency of the audio card ADC in MATLAB Simulink environment with proper justification
- Plot the acquired voltage waveform to Simulink scope.
- Mention the safe operating range of your equipment.

Project Summary: (Not more than one sheet)

Project Specification:

Calculations:

- VDR and Resistance power calculation
- Resolution of ADC
- Sampling Frequency
- Gain Factor

Attachments:

- Project Block Diagram
- Real Project Image
- Image of current and voltage plot (with proper labelling)

Course Code:	EE-394
Course Name:	Digital Signal Processing
Semester:	
Year:	
Section:	
Batch:	
Lab Instructor name:	
Submission deadline:	

OEL Statement:

To convert an analog (voltage & current) signal into digital signal using ADC and display it on MATLAB Simulink environment.

Deliverables:

1) A Prototype hardware that can read Analog signals (voltage & Current) and can display it on any digital environment.

2) A report containing details related to ADC and hardware specification.

Methodology:

1) Using Transformer convert 220VAC from mains into 12VAC.
2) Using VDR convert 12VAC to a voltage compatible to ADC (show all the calculations of resistances with their power ratings).
3) Set the sampling frequency of the ADC.
4) Plot the acquired voltage waveform to Simulink or any digital environment.
5) Mention the safe operating range of your equipment.

Guidelines:

Figures, Tables, Block Diagrams etc. should be clear with necessary details.

Lethal voltages (above 50 V) must be properly insulated

Rubric Sheet

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	0	1	2	3	4
Equipment Identification Sensory skill to identify equipment and/or its component for a lab work.	Not able to identify the equipment.	--	--	--	Able to identify equipment as well as its components.
Equipment Use Sensory skills to describe the use of the equipment for the lab work.	Never describes the use of equipment.	Rarely able to describe the use of equipment.	Occasionally describe the use of equipment.	Often able to describe the use of equipment.	Frequently able to describe the use of equipment.
Procedural Skills Displays skills to act upon sequence of steps in lab work.	Not able to either learn or perform lab work procedure.	Able to slightly understand lab work procedure and perform lab work.	Able to somewhat understand lab work procedure and perform lab work.	Able to moderately understand lab work procedure and perform lab work.	Able to fully understand lab work procedure and perform lab work.
Response Ability to imitate the lab work on his/her own.	Not able to imitate the lab work.	Able to slightly imitate the lab work.	Able to somewhat imitate the lab work.	Able to moderately imitate the lab work.	Able to fully imitate the lab work.
Observation's Use Displays skills to perform related mathematical calculations using the observations from lab work.	Not able to use lab work observations into mathematical calculations.	Able to slightly use lab work observations into mathematical calculations.	Able to somewhat use lab work observations into mathematical calculations.	Able to moderately use lab work observations into mathematical calculations.	Able to fully use lab work observations into mathematical calculations.
Safety Adherence Adherence to safety procedures.	Doesn't adhere to safety procedures	Slightly adheres to safety procedures.	Somewhat adheres to safety procedures.	Moderately adheres to safety procedures.	Fully adheres to safety procedures.
Equipment Handling Equipment care during the use.	Doesn't handle equipment with required care.	Rarely handles equipment with required care.	Occasionally handles equipment with required care	Often handles equipment with required care.	Handles equipment with required care.
Group Work Contributes in a group-based lab work.	Never participates.	Rarely participates.	Occasionally participates and contributes.	Often participates and contributes.	Frequently participates and contributes.
Remarks					
Instructor's Signature					

NED University of Engineering & Technology
Department of _____________________ Engineering

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Software Menu Identification and Usage: Ability to initialise, configure and *operate* software environment *under supervision*, using menus, shortcuts, instructions etc.	Unable to understand and use software menu	Little ability and understanding of software menu operation, makes many mistake	Moderate ability and understanding of software menu operation, makes lesser mistakes	Reasonable understanding of software menu operation, makes no major mistakes	Demonstrates command over software menu usage with frequent use of advance menu options
Procedural Programming of given model: *Practice* procedural programming techniques, in order to code specific model	Little to no understanding of procedural programming techniques	Slight ability to use procedural programming techniques for coding given algorithm	Mostly correct recognition and application of procedural programming techniques but makes crucial errors for the given model	Correctly recognises and uses procedural programming techniques with no errors but unable to run model successfully	Correctly recognises and uses procedural programming techniques with no errors and runs model successfully
Relating Theoretical Concepts, Equations and Transforms to Code: *Recognise* relation between model concepts and written code and *manipulate* the code in accordance of requirements	Completely unable to relate between model concepts and written code, unable to do manipulations	Able to recognise some relation between model concepts and written code, unable to do manipulations	Able to recognise relation between model concepts and written code, unable to do manipulations	Able to recognise relation between model concepts and written code, able to do some manipulations	Able to recognise relation between model concepts and written code, able to completely manipulate code in line with theoretical concepts
Detecting and Removing Errors: *Detect* Errors/Exceptions and in simulation and *manipulate* code to rectify the simulation	Unable to check and detect error messages and indications in software	Able to find error messages and indications in software but no understanding of detecting those errors and their types	Able to find error messages and indications in software as well as understanding of detecting some of those errors and their types	Able to find error messages in software as well as understanding of detecting all of those errors and their types	Able to find error messages in software along with the understanding to detect and rectify them

Graphical Visualisation and Comparison of model **Parameters:** *Manipulate* given simulation under supervision, in order to produce graphs/plots for measuring and comparing model parameters	Unable to understand and utilise visualisation or plotting features	Ability to understand and utilise visualisation and plotting features with frequent errors	Ability to understand and utilise visualisation and plotting features successfully but unable to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, partially able to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, also able to compare and analyse them
Following step-by-step procedure to complete lab work: *Observe, imitate and operate* software to complete the provided sequence of steps	Inability to recognise and perform given lab procedures	Able to recognise given lab procedures and perform them but could not follow the prescribed order of steps	Able to recognise given lab procedures and perform them by following prescribed order of steps, with frequent mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with occasional mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with no mistakes
Recording Simulation Observations: *Observe and copy* prescribed or required simulation results in accordance with lab manual instructions	Inability to recognise prescribed or required simulation measurements	Able to recognise prescribed or required simulation measurements but does not record according to given instructions	—	Able to recognise prescribed or required simulation measurements but records them incompletely	Able to recognise prescribed or required simulation measurements and records them completely, in tabular form
Discussion and Conclusion: *Demonstrate* discussion capacity on the recorded observations and draw conclusions from it, relating them to theoretical principles/concepts	Complete inability to discuss recorded observations and draw conclusions	Slight ability to discuss recorded observations and draw conclusions	Moderate ability to discuss recorded observations and draw conclusions	Reasonable ability to discuss recorded observations and draw conclusions	Full ability to discuss recorded observations and draw conclusions

(Open Ended Lab Part 02)

Objective:

To convert analog (Voltage and current) signal into digital signal using ADC (audio card). Display it on MATLAB Simulink environment and perform Spectral Analysis of the resulting current signal.

Required Components:

1. Audio Card
2. Current Sensor (current sensing resistor / hall effect sensor / CT)
3. Vero board
4. Audio jack
5. Harmonic producing Load (Electronic devices)
6. PC with MATLAB environment

Procedure:

- Using current sensor, convert the current flowing through load into an equivalent voltage.
- If required, using VDR to convert the voltage as obtained from current sensor to a voltage compatible to audio card (show all the calculations of resistances with their power ratings).
- Set the sampling frequency of the audio card ADC in MATLAB Simulink environment with proper justification
- Plot the acquired current waveform to Simulink scope.
- Mention the safe operating range of your equipment.
- Plot the frequency spectrum of the obtained current and Voltage waveform. Use windowing function to reduce DFT leakage if required.
- Also, plot the frequency spectrum of the line voltage as obtained in open ended lab 01.

Project Summary:

Project Specification:

Attachments:

- Project Block Diagram
- Real Project Image

- Image of current and voltage plot and Spectrum(with proper labelling)

Results:

Course Code:	EE-394
Course Name:	Digital Signal Processing
Semester:	
Year:	
Section:	
Batch:	
Lab Instructor name:	
Submission deadline:	

OEL Statement:

To perform spectral analysis of the signals acquired in OEL 1.

Deliverables:

1) A Prototype hardware that can read Analog signals (voltage & Current) and can perform spectral analysis of the acquired signals

2) A report containing details related to spectral analysis of the acquired signals.

Methodology:

1) Plot the frequency spectrum of the obtained current and Voltage waveform.

2)Use windowing function to reduce DFT leakage if required.

Guidelines:

Figures, Tables, Block Diagrams etc. should be clear with necessary details.

Lethal voltages (above 50 V) must be properly insulated

Rubric Sheet

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	0	1	2	3	4
Equipment Identification Sensory skill to identify equipment and/or its component for a lab work.	Not able to identify the equipment.	--	--	--	Able to identify equipment as well as its components.
Equipment Use Sensory skills to describe the use of the equipment for the lab work.	Never describes the use of equipment.	Rarely able to describe the use of equipment.	Occasionally describe the use of equipment.	Often able to describe the use of equipment.	Frequently able to describe the use of equipment.
Procedural Skills Displays skills to act upon sequence of steps in lab work.	Not able to either learn or perform lab work procedure.	Able to slightly understand lab work procedure and perform lab work.	Able to somewhat understand lab work procedure and perform lab work.	Able to moderately understand lab work procedure and perform lab work.	Able to fully understand lab work procedure and perform lab work.
Response Ability to imitate the lab work on his/her own.	Not able to imitate the lab work.	Able to slightly imitate the lab work.	Able to somewhat imitate the lab work.	Able to moderately imitate the lab work.	Able to fully imitate the lab work.
Observation's Use Displays skills to perform related mathematical calculations using the observations from lab work.	Not able to use lab work observations into mathematical calculations.	Able to slightly use lab work observations into mathematical calculations.	Able to somewhat use lab work observations into mathematical calculations.	Able to moderately use lab work observations into mathematical calculations.	Able to fully use lab work observations into mathematical calculations.
Safety Adherence Adherence to safety procedures.	Doesn't adhere to safety procedures	Slightly adheres to safety procedures.	Somewhat adheres to safety procedures.	Moderately adheres to safety procedures.	Fully adheres to safety procedures.
Equipment Handling Equipment care during the use.	Doesn't handle equipment with required care.	Rarely handles equipment with required care.	Occasionally handles equipment with required care	Often handles equipment with required care.	Handles equipment with required care.
Group Work Contributes in a group-based lab work.	Never participates.	Rarely participates.	Occasionally participates and contributes.	Often participates and contributes.	Frequently participates and contributes.
Remarks					
Instructor's Signature					

NED University of Engineering & Technology
Department of ____________________ Engineering

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Software Menu Identification and Usage: Ability to initialise, configure and *operate* software environment *under supervision*, using menus, shortcuts, instructions etc.	Unable to understand and use software menu	Little ability and understanding of software menu operation, makes many mistake	Moderate ability and understanding of software menu operation, makes lesser mistakes	Reasonable understanding of software menu operation, makes no major mistakes	Demonstrates command over software menu usage with frequent use of advance menu options
Procedural Programming of given model: *Practice* procedural programming techniques, in order to code specific model	Little to no understanding of procedural programming techniques	Slight ability to use procedural programming techniques for coding given algorithm	Mostly correct recognition and application of procedural programming techniques but makes crucial errors for the given model	Correctly recognises and uses procedural programming techniques with no errors but unable to run model successfully	Correctly recognises and uses procedural programming techniques with no errors and runs model successfully
Relating Theoretical Concepts, Equations and Transforms to Code: *Recognise* relation between model concepts and written code and *manipulate* the code in accordance of requirements	Completely unable to relate between model concepts and written code, unable to do manipulations	Able to recognise some relation between model concepts and written code, unable to do manipulations	Able to recognise relation between model concepts and written code, unable to do manipulations	Able to recognise relation between model concepts and written code, able to do some manipulations	Able to recognise relation between model concepts and written code, able to completely manipulate code in line with theoretical concepts
Detecting and Removing Errors: *Detect* Errors/Exceptions and in simulation and *manipulate* code to rectify the simulation	Unable to check and detect error messages and indications in software	Able to find error messages and indications in software but no understanding of detecting those errors and their types	Able to find error messages and indications in software as well as understanding of detecting some of those errors and their types	Able to find error messages in software as well as understanding of detecting all of those errors and their types	Able to find error messages in software along with the understanding to detect and rectify them

Graphical Visualisation and Comparison of model **Parameters:** *Manipulate* given simulation under supervision, in order to produce graphs/plots for measuring and comparing model parameters	Unable to understand and utilise visualisation or plotting features	Ability to understand and utilise visualisation and plotting features with frequent errors	Ability to understand and utilise visualisation and plotting features successfully but unable to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, partially able to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, also able to compare and analyse them
Following step-by-step procedure to complete lab work: *Observe, imitate and operate* software to complete the provided sequence of steps	Inability to recognise and perform given lab procedures	Able to recognise given lab procedures and perform them but could not follow the prescribed order of steps	Able to recognise given lab procedures and perform them by following prescribed order of steps, with frequent mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with occasional mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with no mistakes
Recording Simulation Observations: *Observe and copy* prescribed or required simulation results in accordance with lab manual instructions	Inability to recognise prescribed or required simulation measurements	Able to recognise prescribed or required simulation measurements but does not record according to given instructions	—	Able to recognise prescribed or required simulation measurements but records them incompletely	Able to recognise prescribed or required simulation measurements and records them completely, in tabular form
Discussion and Conclusion: *Demonstrate* discussion capacity on the recorded observations and draw conclusions from it, relating them to theoretical principles/concepts	Complete inability to discuss recorded observations and draw conclusions	Slight ability to discuss recorded observations and draw conclusions	Moderate ability to discuss recorded observations and draw conclusions	Reasonable ability to discuss recorded observations and draw conclusions	Full ability to discuss recorded observations and draw conclusions

Rubric Sheet

Psychomotor Domain Assessment Rubric-Level P3					
Skill Sets	Extent of Achievement				
	0	1	2	3	4
Equipment Identification Sensory skill to identify equipment and/or its component for a lab work.	Not able to identify the equipment.	--	--	--	Able to identify equipment as well as its components.
Equipment Use Sensory skills to describe the use of the equipment for the lab work.	Never describes the use of equipment.	Rarely able to describe the use of equipment.	Occasionally describe the use of equipment.	Often able to describe the use of equipment.	Frequently able to describe the use of equipment.
Procedural Skills Displays skills to act upon sequence of steps in lab work.	Not able to either learn or perform lab work procedure.	Able to slightly understand lab work procedure and perform lab work.	Able to somewhat understand lab work procedure and perform lab work.	Able to moderately understand lab work procedure and perform lab work.	Able to fully understand lab work procedure and perform lab work.
Response Ability to imitate the lab work on his/her own.	Not able to imitate the lab work.	Able to slightly imitate the lab work.	Able to somewhat imitate the lab work.	Able to moderately imitate the lab work.	Able to fully imitate the lab work.
Observation's Use Displays skills to perform related mathematical calculations using the observations from lab work.	Not able to use lab work observations into mathematical calculations.	Able to slightly use lab work observations into mathematical calculations.	Able to somewhat use lab work observations into mathematical calculations.	Able to moderately use lab work observations into mathematical calculations.	Able to fully use lab work observations into mathematical calculations.
Safety Adherence Adherence to safety procedures.	Doesn't adhere to safety procedures	Slightly adheres to safety procedures.	Somewhat adheres to safety procedures.	Moderately adheres to safety procedures.	Fully adheres to safety procedures.
Equipment Handling Equipment care during the use.	Doesn't handle equipment with required care.	Rarely handles equipment with required care.	Occasionally handles equipment with required care	Often handles equipment with required care.	Handles equipment with required care.
Group Work Contributes in a group-based lab work.	Never participates.	Rarely participates.	Occasionally participates and contributes.	Often participates and contributes.	Frequently participates and contributes.
Remarks					
Instructor's Signature					

NED University of Engineering & Technology
Department of ______________________ Engineering

Psychomotor Domain Assessment Rubric for Laboratory (Level P3)					
Skill(s) to be assessed	Extent of Achievement				
	0	1	2	3	4
Software Menu Identification and Usage: Ability to initialise, configure and *operate* software environment *under supervision*, using menus, shortcuts, instructions etc.	Unable to understand and use software menu	Little ability and understanding of software menu operation, makes many mistake	Moderate ability and understanding of software menu operation, makes lesser mistakes	Reasonable understanding of software menu operation, makes no major mistakes	Demonstrates command over software menu usage with frequent use of advance menu options
Procedural Programming of given model: *Practice* procedural programming techniques, in order to code specific model	Little to no understanding of procedural programming techniques	Slight ability to use procedural programming techniques for coding given algorithm	Mostly correct recognition and application of procedural programming techniques but makes crucial errors for the given model	Correctly recognises and uses procedural programming techniques with no errors but unable to run model successfully	Correctly recognises and uses procedural programming techniques with no errors and runs model successfully
Relating Theoretical Concepts, Equations and Transforms to Code: *Recognise* relation between model concepts and written code and *manipulate* the code in accordance of requirements	Completely unable to relate between model concepts and written code, unable to do manipulations	Able to recognise some relation between model concepts and written code, unable to do manipulations	Able to recognise relation between model concepts and written code, unable to do manipulations	Able to recognise relation between model concepts and written code, able to do some manipulations	Able to recognise relation between model concepts and written code, able to completely manipulate code in line with theoretical concepts
Detecting and Removing Errors: *Detect* Errors/Exceptions and in simulation and *manipulate* code to rectify the simulation	Unable to check and detect error messages and indications in software	Able to find error messages and indications in software but no understanding of detecting those errors and their types	Able to find error messages and indications in software as well as understanding of detecting some of those errors and their types	Able to find error messages in software as well as understanding of detecting all of those errors and their types	Able to find error messages in software along with the understanding to detect and rectify them

Graphical Visualisation and Comparison of model **Parameters:** *Manipulate* given simulation under supervision, in order to produce graphs/plots for measuring and comparing model parameters	Unable to understand and utilise visualisation or plotting features	Ability to understand and utilise visualisation and plotting features with frequent errors	Ability to understand and utilise visualisation and plotting features successfully but unable to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, partially able to compare and analyse them	Ability to understand and utilise visualisation and plotting features successfully, also able to compare and analyse them
Following step-by-step procedure to complete lab work: *Observe, imitate and operate* software to complete the provided sequence of steps	Inability to recognise and perform given lab procedures	Able to recognise given lab procedures and perform them but could not follow the prescribed order of steps	Able to recognise given lab procedures and perform them by following prescribed order of steps, with frequent mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with occasional mistakes	Able to recognise given lab procedures and perform them by following prescribed order of steps, with no mistakes
Recording Simulation Observations: *Observe and copy* prescribed or required simulation results in accordance with lab manual instructions	Inability to recognise prescribed or required simulation measurements	Able to recognise prescribed or required simulation measurements but does not record according to given instructions	—	Able to recognise prescribed or required simulation measurements but records them incompletely	Able to recognise prescribed or required simulation measurements and records them completely, in tabular form
Discussion and Conclusion: *Demonstrate* discussion capacity on the recorded observations and draw conclusions from it, relating them to theoretical principles/concepts	Complete inability to discuss recorded observations and draw conclusions	Slight ability to discuss recorded observations and draw conclusions	Moderate ability to discuss recorded observations and draw conclusions	Reasonable ability to discuss recorded observations and draw conclusions	Full ability to discuss recorded observations and draw conclusions