ESTD:2008

NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

(Approved by AICTE, New Delhi & Permanently Affiliated to JNTUGV, Vizianagaram)

Recognized under Section 2(f) & 12(B) of the UGC Act, 1956| Accredited by NAAC with 'A' Grade

SONTYAM, ANANDAPURAM, VISAKHAPATNAM - 531173

Department of Electronics and Communication Engineering

DIGITAL SIGNAL PROCESSING
STUDENT LABORATORY RECORD

NAME OF THE STUDENT____________________________________

ROLL.NO____

BRANCH____

YEAR____________________ SEMESTER________________________

ESTD

NSRIT

NS RAJU INSTITUTE OF TECHNOLOGY

AUTONOMOUS

SONTYAM, PENDURTI - ANANDAPURAM HIGHWAY, VISAKHAPATNAM - 531 173

(Apporved by AICTE, New Delhi & Affiliated to JNTUK, Kakinada)

(AN ISO 9001 : 2008 Certified Institution)

CERTIFICATE

This is certify that is bonafied record of Practical work done in the Department of ....B.Tech................ Year...............Semester during the academic year -20 by

Name :..

Roll No...........................Branch..................Subject..

Staff Member **Head of the Department**

Record Submitted for the Practical Examination

Held on :..Batch No...

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ESTD:2008
NSRIT NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)
(Approved by AICTE, New Delhi & Permanently Affiliated to JNTUGV, Vizianagaram)
Recognized under Section 2(f) & 12(B) of the UGC Act, 1956| Accredited by NAAC with 'A' Grade

III Year – II SEMESTER

T	P	C
0	3	1.5

DIGITAL SIGNAL PROCESSING LAB

List of Experiments –

1. Generation of discrete time signals for discrete signals
2. To verify the Linear Convolution
    a. Using MATLAB
    b. b) Using Code Composer Studio (CCS)
3. To verify the Circular Convolution for discrete signals
    a. Using MATLAB
    b. b) Using Code Composer Studio (CCS)
4. To Find the addition of Sinusoidal Signals
5. To verify Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT)
    a. Using MATLAB
    b. Using Code Composer Studio (CCS)
6. Transfer Function Stability Analysis: using pole-zero plot, bode plot, Nyquist plot, z-plane plot.
7. Frequency Response of IIR low pass Butterworth Filter
8. Frequency Response of IIR high pass Butterworth Filter
9. Frequency Response of IIR low pass Chebyshev Filter
10. Frequency Response of IIR high pass Chebyshev Filter
11. Frequency Response of FIR low pass Filter using Rectangle Window
12. Frequency Response of FIR low pass Filter using Triangle Window

Equipment's & Software required:

Software:

i.) Computer Systems with latest specifications
ii) Connected in LAN (Optional)
iii) Operating system (Windows XP)
iv) Simulations software (Simulink &MATLAB, Code Composer Studio (CCS))

ESTD:2008

NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)
(Approved by AICTE, New Delhi & Permanently Affiliated to JNTUGV, Vizianagaram)
Recognized under Section 2(f) & 12(B) of the UGC Act, 1956| Accredited by NAAC with 'A' Grade

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DIGITAL SIGNAL PROCESSINGLABORATORY

INDEX

S.No	Date	Name of the Experiment	Page. No	Marks	Sign
1		Generation of discrete time signals for discrete signals			
2		To verify the Linear Convolution Using MATLAB			
3		To verify the Circular Convolution for discrete signals Using MATLAB			
4		To Find the addition of Sinusoidal Signals			
5		To verify Discrete Fourier Transform (DFT) And Inverse Discrete FourierTransform (IDFT) Using MATLAB			
6		To verify the Linear Convolution Using Code Composer Studio (CCS)			
7		To verify the Circular Convolution for discrete signals Using Code Composer Studio (CCS)			
8		Frequency Response of IIR low pass and high pass using Butterworth Filter			
9		Frequency Response of IIR low pass and High pass using Chebyshev Filter			
10		Frequency Response of FIR low pass Filter using Rectangle Window			
11		Frequency Response of FIR low pass Filter using Triangle Window			
12		FFT of 1-D Signal (Add on Experiment)			

ESTD:2008
NSRIT NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)
(Approved by AICTE, New Delhi & Permanently Affiliated to JNTUGV, Vizianagaram)
Recognized under Section 2(f) & 12(B) of the UGC Act, 1956| Accredited by NAAC with 'A' Grade

Department of Electronics and Communication Engineering

Program Educational Objectives (PEOs)

PEO #1: Demonstrate the application of domain knowledge in solving real time problems and provide research based sustainable solutions in different specializations of Electronics and Communication Engineering or allied branch of engineering and technology and lead a satisfactory job employment with 21st century skills

PEO #2: Involve themselves in life-long learning by enriching his/her competency in the chosen field of interest through professional experience, advanced studies leading to research, learning new age skills that demands dynamism for a continued better prospect to accomplish their professional and career goals

PEO #3: Demonstrate the skill sets that are very much essential to work successfully for a rewarding career in an interdisciplinary environment.

Programme Outcomes (POs)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering Fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the Engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Programme Specific Outcomes (PSOs)

PSO 1: To demonstrate the ability to design and develop complex systems in the areas of next generation Communication Systems, IoT based Embedded Systems, Advanced Signal and Image Processing, latest Semiconductor technologies, RF and Power Systems

PSO 2: To demonstrate the ability to solve complex Electronics and Communication Engineering problems using latest hardware and software tools along with analytical skills to contribute to useful, frugal and eco-friendly solutions

ESTD:2008
NSRIT NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)
(Approved by AICTE, New Delhi & Permanently Affiliated to JNTUGV, Vizianagaram)
Recognized under Section 2(f) & 12(B) of the UGC Act, 1956| Accredited by NAAC with 'A' Grade
SONTYAM, Pendurthi - Anandapuram Highway, Visakhapatnam - 531173, Ph : 9885824167, 8099464546, www.nsrit.edu.in

CO-PO Mapping

DIGITAL SIGNAL PROCESSING LABORATORY

At the end of the course, students will be able to

Code	Course Outcomes	Mapping with POs		
		PO4	PO5	PSO1
20EC607.1	Develop various DSP Algorithms using MATLAB Software	3	2	3
20EC607.2	Analyze and Observe Magnitude and phase characteristics (Frequency response Characteristics) of digital IIR-Butterworth, Chebyshev filters	3	2	3
20EC607.3	Analyze and Observe Frequency response Characteristics of digital FIR filters using window techniques	3	2	3
20EC607.4	Develop and Implement DSP algorithms in software using a computer language such as C with TMS320C6713 Processor	3	3	3

1. Weakly Contributing | 2. Moderately Contributing | 3. Strongly Contributing, for the attainment of respective POs

ESTD:2008

NSRIT NADIMPALLI SATYANARAYANA RAJU INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

(Approved by AICTE, New Delhi & Permanently Affiliated to JNTUGV, Vizianagaram)

Recognized under Section 2(f) & 12(B) of the UGC Act, 1956| Accredited by NAAC with 'A' Grade

SONTYAM, Pendurthi - Anandapuram Highway, Visakhapatnam - 531173, Ph : 9885824167, 8099464546, www.nsrit.edu.in

Department of Electronics and Communication Engineering

Course Title: DIGITAL SIGNAL PROCESSING LABORATORY

Year/Sem: III/II **Regulation: AR20**

S.No	List of Experiments	POs		
		PO4	PO5	PSO 1
1	Generation of discrete time signals for discrete signals	3	2	3
2	To verify the Linear Convolution Using MATLAB	3	2	3
3	To verify the Circular Convolution for discrete signals Using MATLAB	3	3	3
4	To Find the addition of Sinusoidal Signals	3	2	3
5	To verify Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT) Using MATLAB	3	3	3
6	To verify the Linear Convolution Using Code Composer Studio (CCS)	3	2	3
7	To verify the Circular Convolution for discrete signals Using Code Composer Studio (CCS)	3	2	3
8	Frequency Response of IIR low pass and high pass using Butterworth Filter	3	2	3
9	Frequency Response of IIR low pass and High pass using Chebyshev Filter	3	2	3
10	Frequency Response of FIR low pass Filter using Rectangle Window	3	2	3
11	Frequency Response of FIR low pass Filter using Triangle Window	3	2	3
12	FFT of 1-D Signal (Add on Experiment)	3	2	3

PART-1
SIGNALS

NSRIT

Internal Quality Assurance Cell (IQAC)
Rubrics for Digital Signal Processing Laboratory Course (20EC607)

Preamble: This rubric is specifically designed to assess the performance of the students in the laboratory courses where the practical exercises are being involved. The assessment broadly covers "Conduct investigations of complex problems" and more particularly "the ability" to perform experiments in a laboratory set-up (identification of equipment, initial settings for performing the required tests, perform experiment, taking necessary readings, documentation, synthesis and interpretation of results to provide solutions with valid justifications).

	Criteria	Unsatisfactory (1)	Developing (2)	Satisfactory (3)	Good (4)	Exemplary (5)	Score
A. Preparation & Observation	Criterion #1 Software Menu Identification and Procedural Programming of given Digital Signal Processing experiment.	Unable to understand and use Software menu and understanding of procedural programming techniques	Able to understand and use Software menu operations and use procedural programming techniques for Coding given algorithm	Able to understand and use Software menu operations correct recognition and application of procedural programming techniques but performs crucial errors for the given processing scheme	Reasonable understanding of software menu operation, makes no major mistakes and correctly recognizes and uses procedural programming techniques with no errors but unable to run processing scheme successfully	Demonstrates command over software menu usage with frequent use of advance menu options and Correctly recognizes and uses procedural programming techniques without errors and runs processing successfully	
B. Implementation	Criterion #2 Identifying and rectifying Errors and Graphical Visualization	Unable to check and identify the error messages and indications in software and understand and utilize visualization or plotting features	Able to identify error messages and indications in software but no understanding of detecting those errors and their types and understand and utilize visualization or plotting features with frequent errors	Able to identify error messages and indications in software as well as understanding of detecting some of those errors and their types understand and utilize visualization or plotting features but unable to compare and analyze them	Able to identify error messages in software as well as understanding of detecting all of those errors and their types but unable to rectify the, and to understand and utilize visualization and plotting features successfully, partially able to compare and analyze them	Able to find error messages in software along with the understanding to detect and rectify all errors, and to understand and utilize visualization and plotting features successfully, also able to compare and analyse them	
C.Observations & Results	Criterion #2 Simulation Observations Discussion and Conclusion:	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	
D. Viva Voce	Criterion #4 Demonstrate the ability to effectively respond to questions	Completely unable understand the pre-lab and post-lab questions.	Unable to provide an appropriate answer, or is unable to answer questions.	Misunderstand the questions and does not respond appropriately to the teacher, or has some trouble in answering questions	Able to listen carefully and respond to questions appropriately	Able to listen carefully and respond to questions appropriately; is able to explain and interpret results to the teacher	
Signature of the Lab Instructor				**Total Score Secured by the Student [(A + B + C+D)/20] x 10**			

EXPERIMENT 1

GENERATION OF DISCRETE TIME SIGNALS FOR DISCRETE SIGNALS

AIM: To generate discrete time signals for discrete signals Using MATLAB Software.

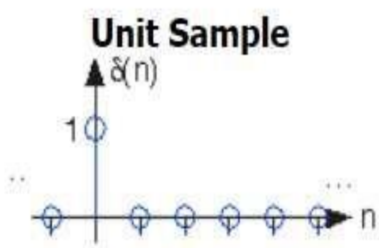
APPARATUS: MATLAB software, PC.

THEORY: There are several elementary signals which play vital role in the study of signals and systems. These elementary signals serve as basic building blocks for the construction of more complex signals. In fact, these elementary signals may be used to model a large number of physical signals which occur in nature. These elementary signals are also called standard signals.

Unit impulse function:

The unit sample function, often referred to as the unit impulse or delta function, is the function that defines the idea of a unit impulse in discrete time. There are not nearly as many intricacies involved in its definition as there are in the definition of the Dirac delta function, the continuous time impulse function. The unit sample function simply takes a value of one at n=0 and a value of zero elsewhere. The impulse function is often written as δ[n].

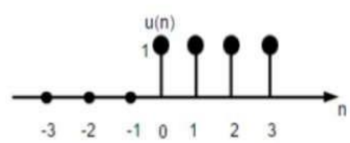
$$\delta[n] = \begin{cases} 1 \text{ if } n = 0 \\ 0 \text{ otherwise} \end{cases}$$



Unit Step Signal:

Discrete time unit step signal is defined as

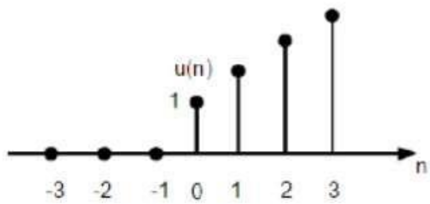
$$U(n) = \begin{cases} 1, & for \quad n \geq 0 \\ 0, & for \quad n < 0 \end{cases}$$



Unit Ramp Function:

A discrete unit ramp function can be defined as

$$r(n) = \begin{cases} n, & for \quad n \geq 0 \\ 0, & for \quad n < 0 \end{cases}$$



Sinusoidal Signal Generation:

The sine wave or sinusoid is a mathematical function that describes a smooth repetitive Oscillation. It occurs often in pure mathematics, as well as physics, signal processing, electrical engineering and many other fields. It's most basic form as a function of time (t) is:

where:

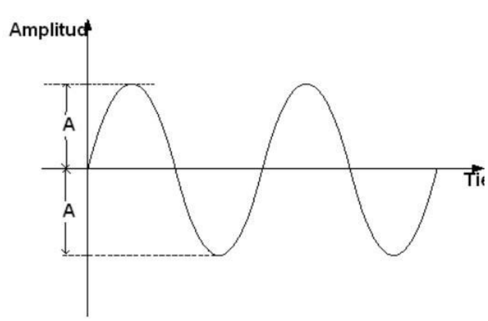
• A, the amplitude, is the peak deviation of the function from its centre position.

• ω, the angular frequency, specifies how many oscillations occur in a unit time interval, in radians per second

• υ, the phase, specifies where in its cycle the oscillation begins at t = 0.

A sampled sinusoid may be written as:

Where f is the signal frequency, fs is the sampling frequency, θ is the phase and A is the amplitude of the signal. The PROGRAM and its OUTPUT is shown below: Note that there are 64 samples with sampling frequency of 8000Hz or sampling time of 0.125 mS (i.e. 1/8000). Hence the record length of the signal is 64x0.125=8mS. There are exactly 8 cycles of sine wave, indicating that the period of one cycle is 1mS which means that the signal frequency is 1 KHz.



PROCEDURE:

- Open MATLAB
- Open new M-file
- Type the program
- Save in current directory
- Compile and Run the program
- For the output see command window\ Figure window

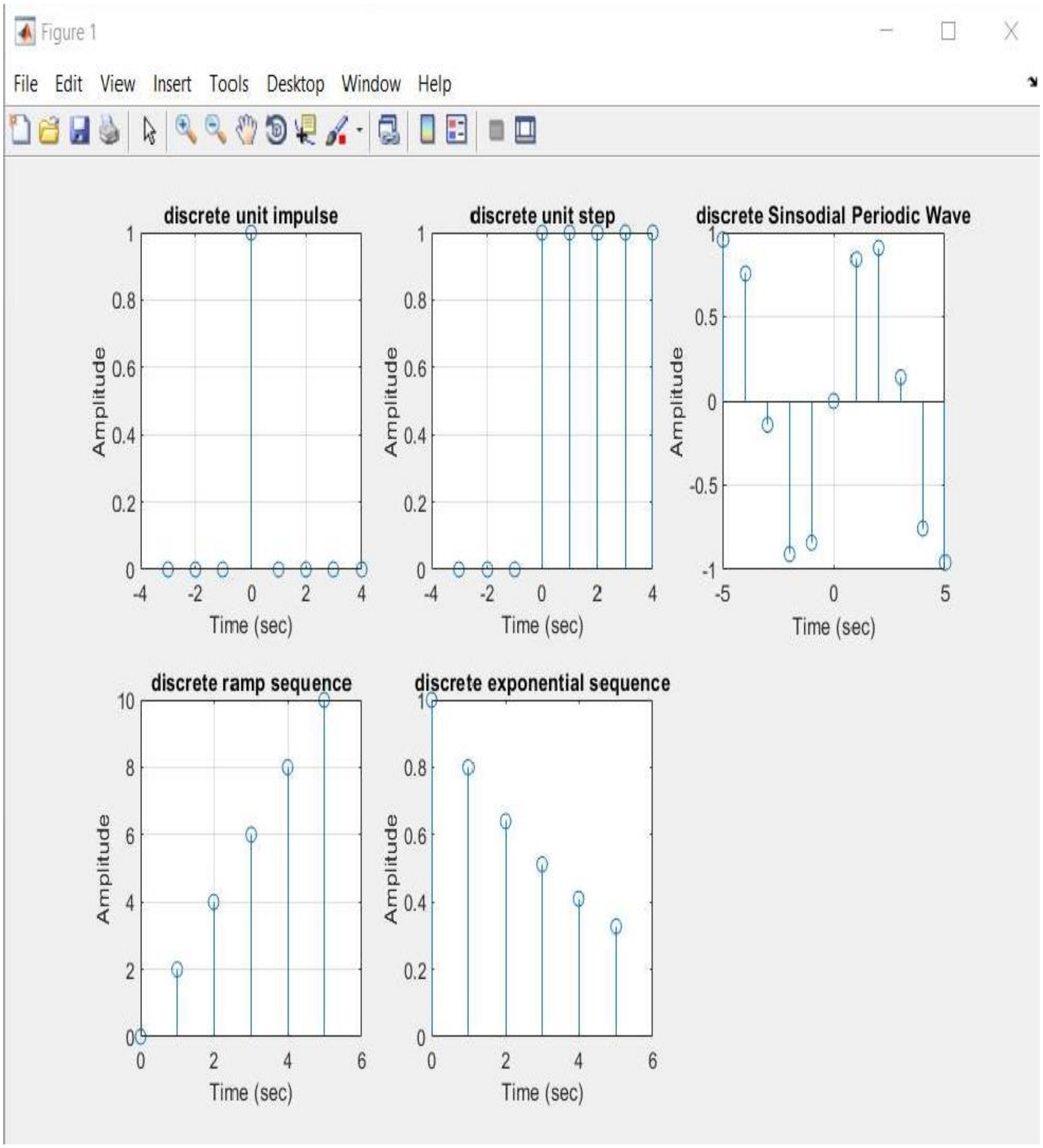
Program:

```
%discrete unit impulse sequence generation
 clc;
close all;
n=-3:4;
x=[n==0];
subplot(2,3,1);
stem (n,x);
xlabel('Time (sec)');
ylabel('Amplitude');
title('discrete unit impulse');
grid;
% discrete unit step sequence generation
n=-3:4;
y=[n>=0];
subplot(2,3,2);
stem(n,y);
xlabel('Time (sec)');
ylabel('Amplitude');
title('discrete unit step');
grid;
% discrete sinsodial sequence generator
```

```
n = -5:5;
x = sin(n);
subplot(2,3,3);
stem(n,x);
xlabel('Time (sec)');

ylabel('Amplitude');
title('discrete Sinsodial Periodic Wave');
grid;
% discrete ramp sequence generator
n = 0:5;
x=2*n;
subplot(2,3,4);
stem(n,x);
xlabel('Time (sec)');
ylabel('Amplitude');
title('discrete ramp sequence');
grid;
% generate a exponential signal
n = 0:5;
x =.8.^(n);
subplot(2,3,5);
stem(n,x);
xlabel('Time (sec)');
ylabel('Amplitude');
title('exponential signal');
```

OUTPUT:



RESULTS:

NSRIT

Internal Quality Assurance Cell (IQAC)
Rubrics for Digital Signal Processing Laboratory Course (20EC607)

Preamble: This rubric is specifically designed to assess the performance of the students in the laboratory courses where the practical exercises are being involved. The assessment broadly covers "Conduct investigations of complex problems" and more particularly "the ability" to perform experiments in a laboratory set-up (identification of equipment, initial settings for performing the required tests, perform experiment, taking necessary readings, documentation, synthesis and interpretation of results to provide solutions with valid justifications).

	Criteria	Unsatisfactory (1)	Developing (2)	Satisfactory (3)	Good (4)	Exemplary (5)	Score
A. Preparation & Observation	Criterion #1 Software Menu Identification and Procedural Programming of given Digital Signal Processing experiment.	Unable to understand and use Software menu and understanding of procedural programming techniques	Able to understand and use Software menu operations and use procedural programming techniques for Coding given algorithm	Able to understand and use Software menu operations correct recognition and application of procedural programming techniques but performs crucial errors for the given processing scheme	Reasonable understanding of software menu operation, makes no major mistakes and correctly recognizes and uses procedural programming techniques with no errors but unable to run processing scheme successfully	Demonstrates command over software menu usage with frequent use of advance menu options and Correctly recognizes and uses procedural programming techniques without errors and runs processing successfully	
B. Implementation	Criterion #2 Identifying and rectifying Errors and Graphical Visualization	Unable to check and identify the error messages and indications in software and understand and utilize visualization or plotting features	Able to identify error messages and indications in software but no understanding of detecting those errors and their types and understand and utilize visualization or plotting features with frequent errors	Able to identify error messages and indications in software as well as understanding of detecting some of those errors and their types understand and utilize visualization or plotting features but unable to compare and analyze them	Able to identify error messages in software as well as understanding of detecting all of those errors and their types but unable to rectify the, and to understand and utilize visualization and plotting features successfully, partially able to compare and analyze them	Able to find error messages in software along with the understanding to detect and rectify all errors, and to understand and utilize visualization and plotting features successfully, also able to compare and analyse them	
C.Observations & Results	Criterion #2 Simulation Observations Discussion and Conclusion:	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	
D. Viva Voce	Criterion #4 Demonstrate the ability to effectively respond to questions	Completely unable understand the pre-lab and post-lab questions.	Unable to provide an appropriate answer, or is unable to answer questions.	Misunderstand the questions and does not respond appropriately to the teacher, or has some trouble in answering questions	Able to listen carefully and respond to questions appropriately	Able to listen carefully and respond to questions appropriately; is able to explain and interpret results to the teacher	
Signature of the Lab Instructor				**Total Score Secured by the Student [(A + B + C+D)/20] x 10**			

EXPERIMENT 2

TO VERIFY THE LINEAR CONVOLUTION USING MATLAB

AIM: To verify the linear convolution of two sequences

APPARATUS: MATLAB software, PC.

THEORY: Communication is an integral concatenation of two signals. it has many applications in numerous areas of signal processing the most popular application is the determination of the output signal of a linear time invariant system by convoluting the input signal, with the impulse response of the system. We know that convoluting two signals is equivalent to multiply the Fourier transform of two signals.

MATHEMATICAL EXPRESSION: For discrete time signals x(n) and h(n) the integration is replaced by summation.

$$\mathbf{Y(n) = x(n)*h(n)} = \sum_{k=-\infty}^{\infty} \boldsymbol{x(k)h(n-k)}$$

where x(k) = input signal sample

h(n-k) = impulse response coefficient

y(n) = convolution output

GRAPHICAL REPRESENTATION:

1. Reflection of h(k) resulting in h(-k)
2. Shifting of h(-k) resulting in h(n-k)
3. Element wise multiplication of the sequence x(k) and h(n – k).
4. Summation of the product sequence x(k)h(n-k) resulting in the convolution value fory(n).

ALGORITHM:

1. Get the sequence x(n) and h(n). Calculate their lengths.
2. Calculate the linear convolution of the two sequences x(n) and y(n).
3. Plot the graph for the convolution sum of x(n)*h(n) versus 'n' i.e. length of the sequence.

PROCEDURE:

1. Open MATLAB window and create a new program in M – file.
2. SAVE THE PROGRAM.
3. COMPILE THE PROGRAM TO CHECK FOR ERRORS.
4. RUN THE PROGRAM AND VIEW THE OUTPUT GRAPH ON THE SCREEN.

PROGRAM:

```
clc;
clear all;
x=[1,1,1,2,1,1];
h=[1,1,2,1];
nx=length(x);
nh=length(h);
y=[]
for n=1:nx+nh-1
sum=0;
yh=0;
for  m=1:nx
if n-m+1<1
break
elseif n-m+1<=nh
yh=x(m)*h(n-m+1);
end
sum=sum+yh;
end
y=[y sum];
end
n=0:1:nx+nh-2;
stem(n,y);
xlabel('n');
ylabel('y(n)');
title('linear convolution');
```

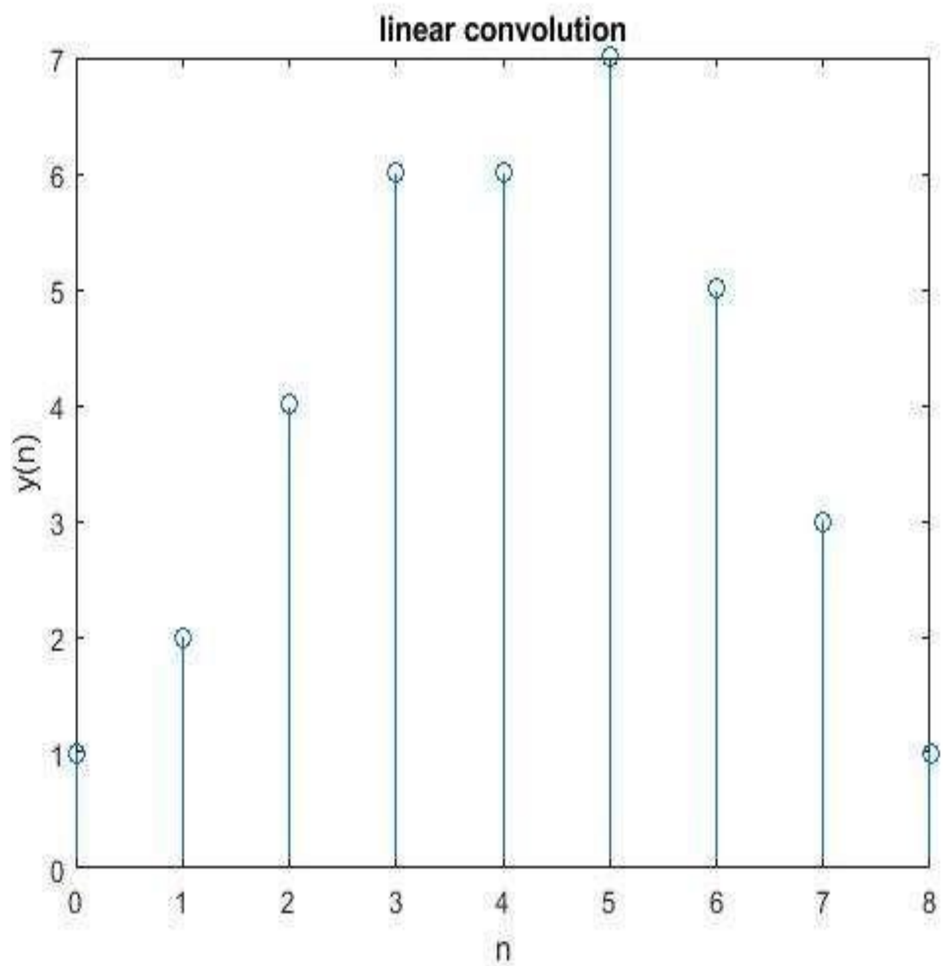
Work Sheet:

x=[1,1,1,2,1,1]

h=[1,1,2,1]

y=[1,2,4,6,6,7,5,3,1]

OUTPUT:



Linear Convolution

RESULT:

Internal Quality Assurance Cell (IQAC)
Rubrics for Digital Signal Processing Laboratory Course (20EC607)

Preamble: This rubric is specifically designed to assess the performance of the students in the laboratory courses where the practical exercises are being involved. The assessment broadly covers "Conduct investigations of complex problems" and more particularly "the ability" to perform experiments in a laboratory set-up (identification of equipment, initial settings for performing the required tests, perform experiment, taking necessary readings, documentation, synthesis and interpretation of results to provide solutions with valid justifications).

	Criteria	Unsatisfactory (1)	Developing (2)	Satisfactory (3)	Good (4)	Exemplary (5)	Score
A. Preparation & Observation	Criterion #1 Software Menu Identification and Procedural Programming of given Digital Signal Processing experiment.	Unable to understand and use Software menu and understanding of procedural programming techniques	Able to understand and use Software menu operations and use procedural programming techniques for Coding given algorithm	Able to understand and use Software menu operations correct recognition and application of procedural programming techniques but performs crucial errors for the given processing scheme	Reasonable understanding of software menu operation, makes no major mistakes and correctly recognizes and uses procedural programming techniques with no errors but unable to run processing scheme successfully	Demonstrates command over software menu usage with frequent use of advance menu options and Correctly recognizes and uses procedural programming techniques without errors and runs processing successfully	
B. Implementation	Criterion #2 Identifying and rectifying Errors and Graphical Visualization	Unable to check and identify the error messages and indications in software and understand and utilize visualization or plotting features	Able to identify error messages and indications in software but no understanding of detecting those errors and their types and understand and utilize visualization or plotting features with frequent errors	Able to identify error messages and indications in software as well as understanding of detecting some of those errors and their types understand and utilize visualization or plotting features but unable to compare and analyze them	Able to identify error messages in software as well as understanding of detecting all of those errors and their types but unable to rectify the, and to understand and utilize visualization and plotting features successfully, partially able to compare and analyze them	Able to find error messages in software along with the understanding to detect and rectify all errors, and to understand and utilize visualization and plotting features successfully, also able to compare and analyse them	
C.Observations & Results	Criterion #2 Simulation Observations Discussion and Conclusion:	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	
D. Viva Voce	Criterion #4 Demonstrate the ability to effectively respond to questions	Completely unable understand the pre-lab and post-lab questions.	Unable to provide an appropriate answer, or is unable to answer questions.	Misunderstand the questions and does not respond appropriately to the teacher, or has some trouble in answering questions	Able to listen carefully and respond to questions appropriately	Able to listen carefully and respond to questions appropriately; is able to explain and interpret results to the teacher	
Signature of the Lab Instructor				**Total Score Secured by the Student [(A + B + C+D)/20] x 10**			

EXPERIMENT 3

TO VERIFY THE CIRCULAR CONVOLUTION FOR DISCRETE SIGNALS USING MATLAB

AIM: To verify circular convolution for discrete signals Using MATLAB

APPARATUS: MATLAB software, PC.

THEORY: In the case of circular convolution, if x(n) contains 'L' number of samples and h(n) has 'm' number of samples and that L > m; then we perform circular convolution between the two using N=MAX(L,M),by adding L – m number of zero samples to the sequence h(n), so that both sequences are periodic with N.

Circular convolution cannot be used to find the response of a linear filter without zero padding.

X3(k) = x1(k) x2(k)

X3(k) = DFT[X1(n)©X2(n)]

$X(k) = \sum_{n=0}^{N-1} x(n)e^{-2\pi n\ /N}$ k = 0,1,2,3,............... N – 1

ALGORITHM:

Step1: start

Step2: read the first sequence

Step3: read the second sequence

Step4: find the length of the first sequence

Step5: find the length of the second sequence

Step6: if lengths are different perform zeropadding

Step7:perform circular convolution for both sequences

Step8:plot the sequence

Step 9: display the sequence

Step10: stop

PROCEDURE:

1. Open MATLAB window and create a new program in M – file

2. Save the program
3. Compile the program to check for errors.
4. Run the program and view the output graph on the screen

PROGRAM:

```
clc;
clear all;
close all;
g=input('Enter the sequence 1:');
h=input('Enter the sequence 2:');
N1=length(g);
N2=length(h);
N=max(N1,N2);
N3=N1-N2;
if(N3>0)
   h=[h,zeros(1,N3)];
else
   g=[g,zeros(1,-N3)];
end
for n=1:N;
   y(n)=0;
for i=1:N;
      j=n-i+1;
if(j<=0)
         j=N+j;
end
      y(n)=[y(n)+(g(i)*h(j))];
end
end
disp('The resultant is');y
stem(y);
xlabel('N->');
ylabel('Amplititude->');
```

Work Sheet:

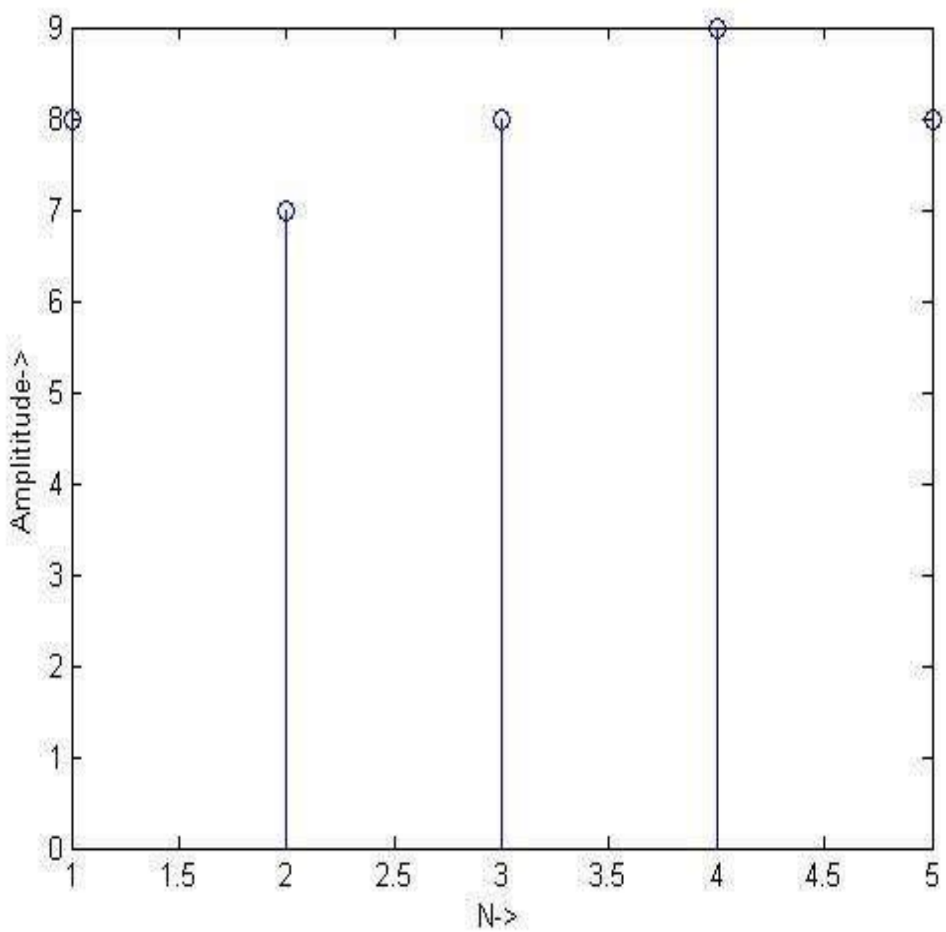
Enter the sequence 1:[1 2 3 1 3]

Enter the sequence 2:[1 2 1]

The resultant is

y = [8 7 8 9 8]

PLOT



Circular Convolution

RESULT:

NSRIT

Internal Quality Assurance Cell (IQAC)
Rubrics for Digital Signal Processing Laboratory Course (20EC607)

Preamble: This rubric is specifically designed to assess the performance of the students in the laboratory courses where the practical exercises are being involved. The assessment broadly covers "Conduct investigations of complex problems" and more particularly "the ability" to perform experiments in a laboratory set-up (identification of equipment, initial settings for performing the required tests, perform experiment, taking necessary readings, documentation, synthesis and interpretation of results to provide solutions with valid justifications).

	Criteria	Unsatisfactory (1)	Developing (2)	Satisfactory (3)	Good (4)	Exemplary (5)	Score
A. Preparation & Observation	Criterion #1 Software Menu Identification and Procedural Programming of given Digital Signal Processing experiment.	Unable to understand and use Software menu and understanding of procedural programming techniques	Able to understand and use Software menu operations and use procedural programming techniques for Coding given algorithm	Able to understand and use Software menu operations correct recognition and application of procedural programming techniques but performs crucial errors for the given processing scheme	Reasonable understanding of software menu operation, makes no major mistakes and correctly recognizes and uses procedural programming techniques with no errors but unable to run processing scheme successfully	Demonstrates command over software menu usage with frequent use of advance menu options and Correctly recognizes and uses procedural programming techniques without errors and runs processing successfully	
B. Implementation	Criterion #2 Identifying and rectifying Errors and Graphical Visualization	Unable to check and identify the error messages and indications in software and understand and utilize visualization or plotting features	Able to identify error messages and indications in software but no understanding of detecting those errors and their types and understand and utilize visualization or plotting features with frequent errors	Able to identify error messages and indications in software as well as understanding of detecting some of those errors and their types understand and utilize visualization or plotting features but unable to compare and analyze them	Able to identify error messages in software as well as understanding of detecting all of those errors and their types but unable to rectify the, and to understand and utilize visualization and plotting features successfully, partially able to compare and analyze them	Able to find error messages in software along with the understanding to detect and rectify all errors, and to understand and utilize visualization and plotting features successfully, also able to compare and analyse them	
C.Observations & Results	Criterion #2 Simulation Observations Discussion and Conclusion:	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	
D. Viva Voce	Criterion #4 Demonstrate the ability to effectively respond to questions	Completely unable understand the pre-lab and post-lab questions.	Unable to provide an appropriate answer, or is unable to answer questions.	Misunderstand the questions and does not respond appropriately to the teacher, or has some trouble in answering questions	Able to listen carefully and respond to questions appropriately	Able to listen carefully and respond to questions appropriately; is able to explain and interpret results to the teacher	
Signature of the Lab Instructor				**Total Score Secured by the Student [(A + B + C+D)/20] x 10**			

EXPERIMENT 4

TO FIND THE ADDITION OF SINUSOIDAL SIGNALS

AIM: To generate the sum of two sine waves using MATLAB software.

APPARATUS: MATLAB software, PC.

THEORY: A continuous time signal is given by

$$x(t) = Asin\,(\Omega t + Q)$$

Where A= amplitude

Ω = Frequency in radius per second

Q = Phase angle in radians

The signals in periodic satisfying the condition x(t+ĩ) = x(t), where, t is the fundamental period for different values of frequencies. The continuous time sinusoidal signals are themselves difference. The discrete time sinusoidal signal is given by

$$X(n)=A\,Cos(w_0n+\phi)$$

Where $w_{0=}$ Frequency

Φ = Phase angle

Using Euler's Identity.

$$A\ Cos\,(w_0n+\phi) = \frac{A}{2} e^{j\emptyset} e^{jw} + \frac{A}{2} e^{-j\emptyset} e^{-jw0n}$$

ALGORITHM:

1. Take input number of samples (N) signal frequency (F) and sampling Frequency (fs).
2. Compute discrete time signal frequency F=f/fs.
3. Generate discrete time sine wave using sine function.
4. Display the sine wave.
5. Similarly generate discrete time, cosine wave using cosine function and display the cosine wave.

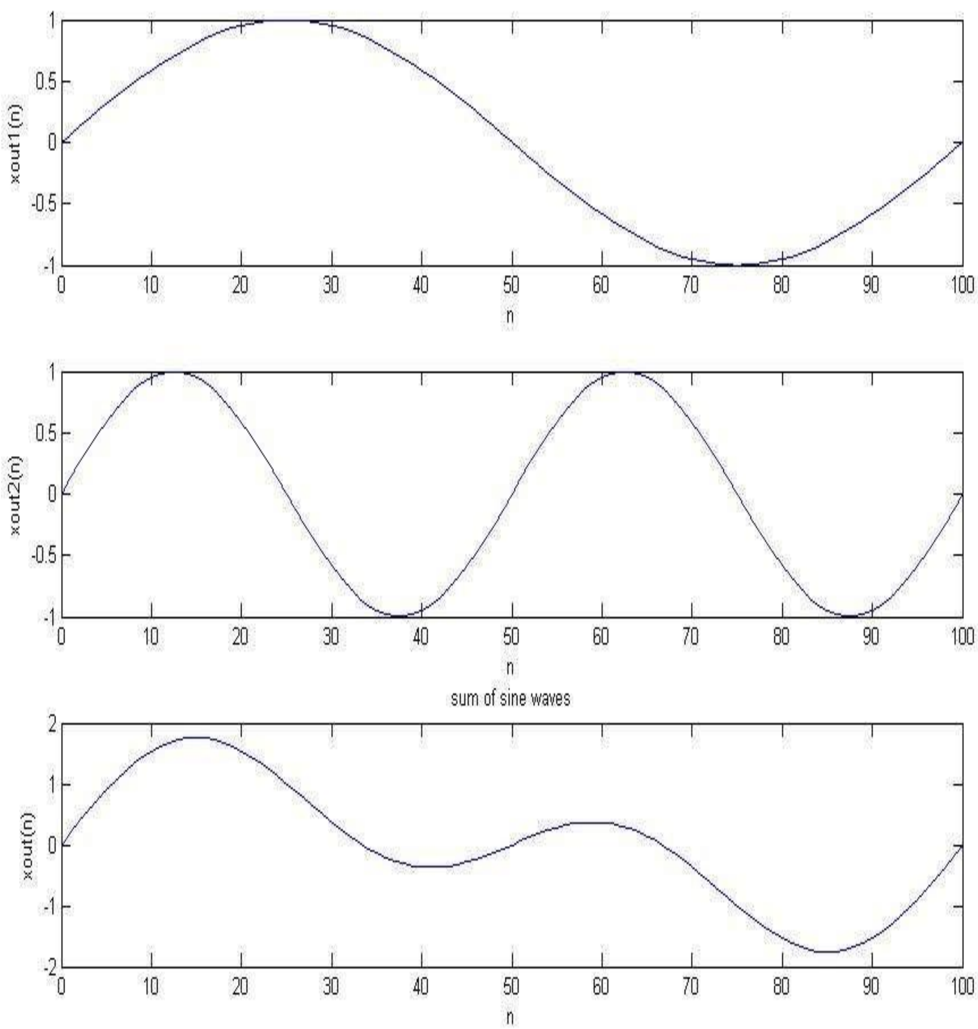
PROCEDURE:

1. Open MATLAB window and create a new program in M – file.
2. Save the program.
3. Compile the program to check for errors.
4. Run the program and view the output graph on the screen.

PROGRAM:

```
clc;
clear all;
close all;
%generation of sinewave1
N=101;
n=0:1:N-1;
xout1=sin(0.02*pi*n);
subplot(3,1,1);
plot(n,xout1);
xlabel('n');
ylabel('xout1(n)');
%generation of sinewave2
n=0:1:N-1;
xout2=sin(0.04*pi*n);
subplot(3,1,2);
plot(n,xout2);
xlabel('n');
ylabel('xout2(n)');
%generation of sum of sinewaves 1and2
n=0:1:N-1;
xout3=xout1+xout2;
subplot(3,1,3);
plot(n,xout3);
xlabel('n');
ylabel('xout3(n)');
title('sum of sinewaves');
```

PLOTS



RESULT:

NSRIT

Internal Quality Assurance Cell (IQAC)
Rubrics for Digital Signal Processing Laboratory Course (20EC607)

Preamble: This rubric is specifically designed to assess the performance of the students in the laboratory courses where the practical exercises are being involved. The assessment broadly covers "Conduct investigations of complex problems" and more particularly "the ability" to perform experiments in a laboratory set-up (identification of equipment, initial settings for performing the required tests, perform experiment, taking necessary readings, documentation, synthesis and interpretation of results to provide solutions with valid justifications).

	Criteria	Unsatisfactory (1)	Developing (2)	Satisfactory (3)	Good (4)	Exemplary (5)	Score
A. Preparation & Observation	Criterion #1 Software Menu Identification and Procedural Programming of given Digital Signal Processing experiment.	Unable to understand and use Software menu and understanding of procedural programming techniques	Able to understand and use Software menu operations and use procedural programming techniques for Coding given algorithm	Able to understand and use Software menu operations correct recognition and application of procedural programming techniques but performs crucial errors for the given processing scheme	Reasonable understanding of software menu operation, makes no major mistakes and correctly recognizes and uses procedural programming techniques with no errors but unable to run processing scheme successfully	Demonstrates command over software menu usage with frequent use of advance menu options and Correctly recognizes and uses procedural programming techniques without errors and runs processing successfully	
B. Implementation	Criterion #2 Identifying and rectifying Errors and Graphical Visualization	Unable to check and identify the error messages and indications in software and understand and utilize visualization or plotting features	Able to identify error messages and indications in software but no understanding of detecting those errors and their types and understand and utilize visualization or plotting features with frequent errors	Able to identify error messages and indications in software as well as understanding of detecting some of those errors and their types understand and utilize visualization or plotting features but unable to compare and analyze them	Able to identify error messages in software as well as understanding of detecting all of those errors and their types but unable to rectify the, and to understand and utilize visualization and plotting features successfully, partially able to compare and analyze them	Able to find error messages in software along with the understanding to detect and rectify all errors, and to understand and utilize visualization and plotting features successfully, also able to compare and analyse them	
C.Observations & Results	Criterion #2 Simulation Observations Discussion and Conclusion:	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	
D. Viva Voce	Criterion #4 Demonstrate the ability to effectively respond to questions	Completely unable understand the pre-lab and post-lab questions.	Unable to provide an appropriate answer, or is unable to answer questions.	Misunderstand the questions and does not respond appropriately to the teacher, or has some trouble in answering questions	Able to listen carefully and respond to questions appropriately	Able to listen carefully and respond to questions appropriately; is able to explain and interpret results to the teacher	
Signature of the Lab Instructor				**Total Score Secured by the Student [(A + B + C+D)/20] x 10**			

EXPERIMENT 5

TO VERIFY DISCRETE FOURIER TRANSFORM (DFT) AND INVERSE DISCRETE FOURIER TRANSFORM (IDFT) USING MATLAB

AIM: To verify Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT).

Apparatus: MATLAB software, PC

THEORY: The DFT is the most important discrete transform, used to perform Fourier analysis in many practical applications. Basic equation to find the DFT of a sequence is given below.

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{kn}{N}}$$

Basic equation to find the IDFT of a sequence is given below.

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi \frac{kn}{N}}$$

Algorithm:

Step I: Get the input sequence.

Step II: Find the DFT of the input sequence using direct equation of DFT.

Step III: Find the IDFT using the direct equation.

Step IV: Plot DFT and IDFT of the given sequence using MATLAB command stem.

Step V: Display the above outputs.

PROGRAM:

```
clc;
close all;
clear all;
xn=input('Enter the sequence x(n)'); %Get the sequence from user
ln=length(xn); %find the length of the sequence
xk=zeros(1,ln); %initialize an array of same size as that of input sequence
ixk=zeros(1,ln); %initialize an array of same size as that of input sequence
%DFT of the sequence
%-----------------------------------------------------------------------
for k=0:ln-1
for n=0:ln-1
xk(k+1)=xk(k+1)+(xn(n+1)*exp((-i)*2*pi*k*n/ln));
end
```

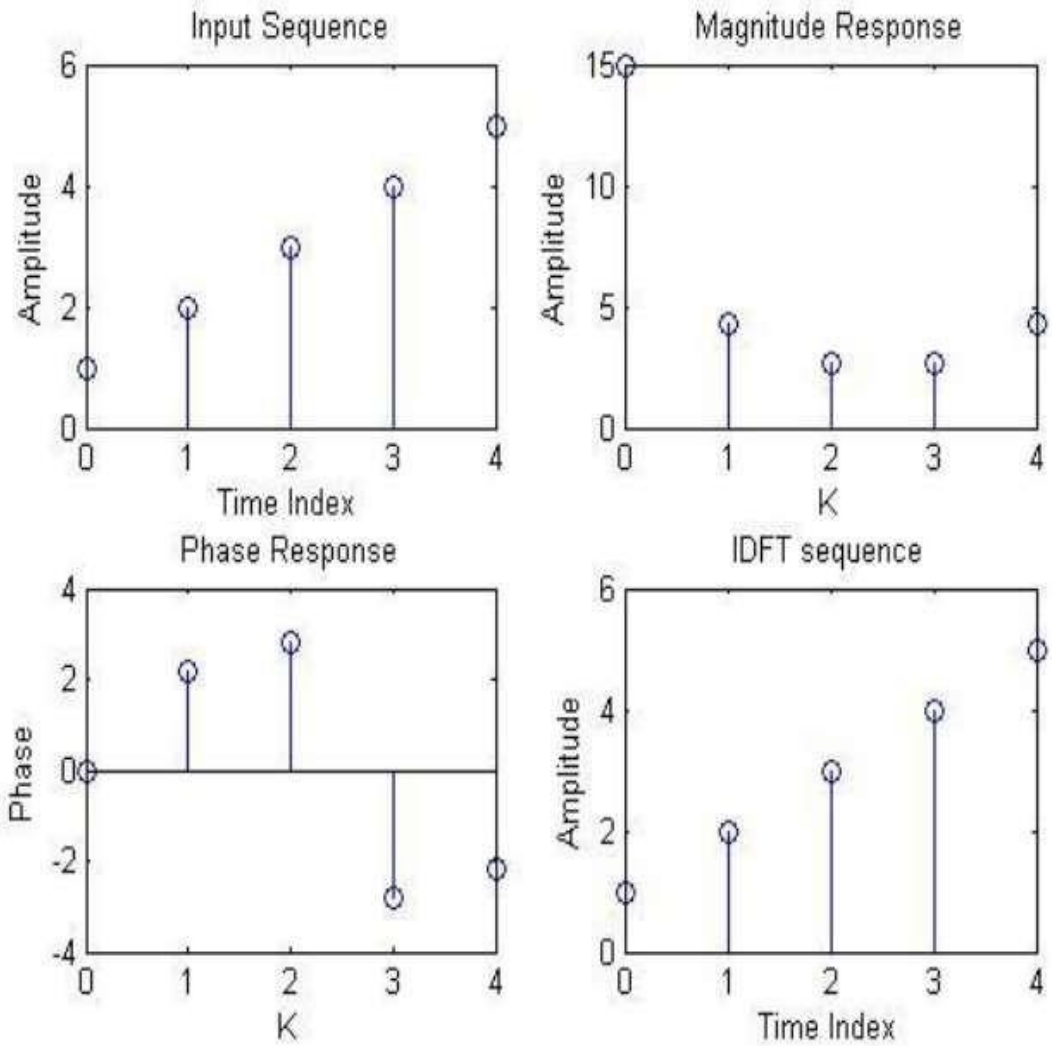
```
end
%________________________________________________________________
%Plotting input sequence
%_____________________________________________________________-
t=0:ln-1;
subplot(221);
stem(t,xn);
ylabel ('Amplitude');
xlabel ('Time Index');
title('Input Sequence');
%__________________________________________________________________________
magnitude=abs(xk); % Find the magnitudes of individual DFT points
% plot the magnitude response
%______________________________________________________________________
t=0:ln-1;
subplot(222);
stem(t,magnitude);
ylabel ('Amplitude');
xlabel ('K');
title('Magnitude Response');
%______________________________________________________________________
phase=angle(xk); % Find the phases of individual DFT points % plot the magnitudesequence
%______________________________________________________________________
t=0:ln-1;
subplot(223);
stem(t,phase);
ylabel ('Phase');
xlabel ('K');
title ('Phase Response');
%______________________________________________________________________
%IDFT of the sequence
%______________________________________________________________________
for n=0:ln-1
for k=0:ln-1
ixk(n+1)=ixk(n+1)+(xk(k+1)*exp((i)*2*pi*k*n/ln));
end
end
ixk=ixk./ln;
%______________________________________________________________________
%code block to plot the input sequence
%______________________________________________________________________
t=0:ln-1;
subplot(224);
stem(t,ixk);
ylabel ('Amplitude');
xlabel ('Time Index');
title ('IDFT sequence');
%________________________________________________________________
```

WORK SPACE:

Xn=[1 2 3 4 5]

Xk = 15,-2.50+3.44i,-2.50+0.81i,-2.49-0.81i,-2.49-3.44i

PLOTS



RESULT:

NSRIT

Internal Quality Assurance Cell (IQAC)
Rubrics for Digital Signal Processing Laboratory Course (20EC607)

Preamble: This rubric is specifically designed to assess the performance of the students in the laboratory courses where the practical exercises are being involved. The assessment broadly covers "Conduct investigations of complex problems" and more particularly "the ability" to perform experiments in a laboratory set-up (identification of equipment, initial settings for performing the required tests, perform experiment, taking necessary readings, documentation, synthesis and interpretation of results to provide solutions with valid justifications).

	Criteria	Unsatisfactory (1)	Developing (2)	Satisfactory (3)	Good (4)	Exemplary (5)	Score
A. Preparation & Observation	Criterion #1 Software Menu Identification and Procedural Programming of given Digital Signal Processing experiment.	Unable to understand and use Software menu and understanding of procedural programming techniques	Able to understand and use Software menu operations and use procedural programming techniques for Coding given algorithm	Able to understand and use Software menu operations correct recognition and application of procedural programming techniques but performs crucial errors for the given processing scheme	Reasonable understanding of software menu operation, makes no major mistakes and correctly recognizes and uses procedural programming techniques with no errors but unable to run processing scheme successfully	Demonstrates command over software menu usage with frequent use of advance menu options and Correctly recognizes and uses procedural programming techniques without errors and runs processing successfully	
B. Implementation	Criterion #2 Identifying and rectifying Errors and Graphical Visualization	Unable to check and identify the error messages and indications in software and understand and utilize visualization or plotting features	Able to identify error messages and indications in software but no understanding of detecting those errors and their types and understand and utilize visualization or plotting features with frequent errors	Able to identify error messages and indications in software as well as understanding of detecting some of those errors and their types understand and utilize visualization or plotting features but unable to compare and analyze them	Able to identify error messages in software as well as understanding of detecting all of those errors and their types but unable to rectify the, and to understand and utilize visualization and plotting features successfully, partially able to compare and analyze them	Able to find error messages in software along with the understanding to detect and rectify all errors, and to understand and utilize visualization and plotting features successfully, also able to compare and analyse them	
C.Observations & Results	Criterion #2 Simulation Observations Discussion and Conclusion:	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	
D. Viva Voce	Criterion #4 Demonstrate the ability to effectively respond to questions	Completely unable understand the pre-lab and post-lab questions.	Unable to provide an appropriate answer, or is unable to answer questions.	Misunderstand the questions and does not respond appropriately to the teacher, or has some trouble in answering questions	Able to listen carefully and respond to questions appropriately	Able to listen carefully and respond to questions appropriately; is able to explain and interpret results to the teacher	
Signature of the Lab Instructor				**Total Score Secured by the Student [(A + B + C+D)/20] x 10**			

EXPERIMENT 6

TO VERIFY THE LINEAR CONVOLUTION USING CODE COMPOSER STUDIO (CCS)

AIM: Verify Linear Convolution on DSP Processor.

APPARATUS: DSP Trainer Kit TMS320C6713

CC Studio, PC.

Procedure to Work on Code Composer Studio

To create the New Project

Project→ New (File Name. pjt, Eg: Vectors.pjt)

To create a Source file

File →New→ Type the code (Save & give file name, Eg: sum.c).

To Add Source files to Project

Project→ Add files to Project→ sum.c

To Add rts.lib file & Hello.cmd:

Project→ Add files to Project→ rts6700.lib

Library files: rts6700.lib (Path: c:\ti\c6000\cgtools\lib\ rts6700.lib)

Note: Select Object& Library in (*.o,*.l) in Type of files

Project→ Add files to Project→hello.cmd

CMD file- Which is common for all non-real time programs.

(Path: c:\ti \ tutorial\dsk6713\hello1\hello.cmd)

Note: Select Linker Command file (*.cmd) in Type of files

Compile:-

To Compile: Project→Compile

To rebuild: project → rebuild,

Which will create the final .out executable file. (Eg.Vectors.out).

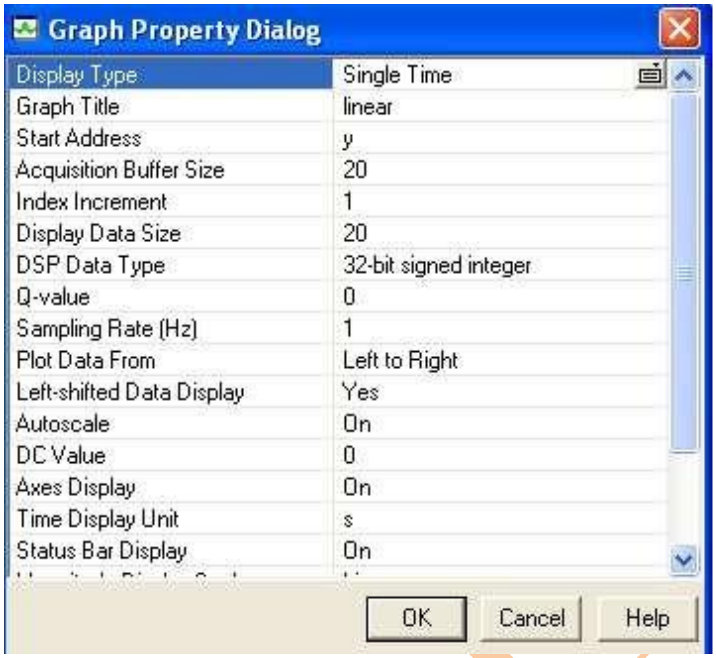
Procedure to Lode and Run program:

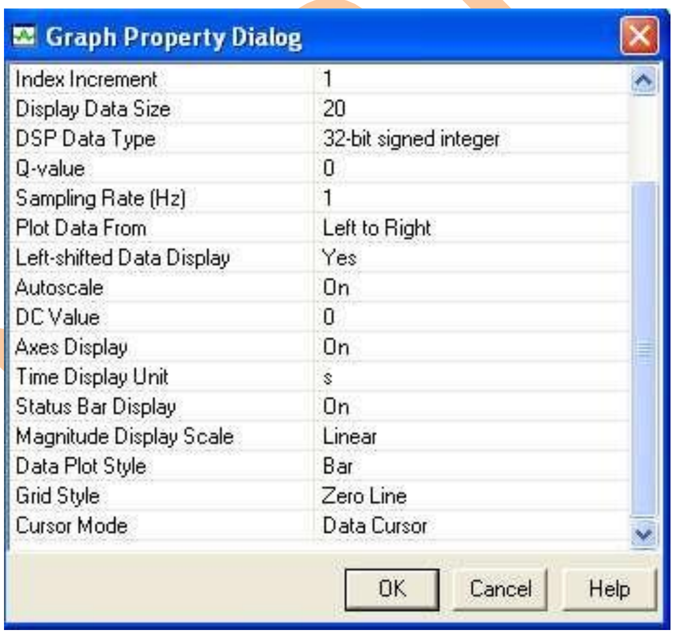
Load the Program to DSK: File→ Load program →Vectors.out

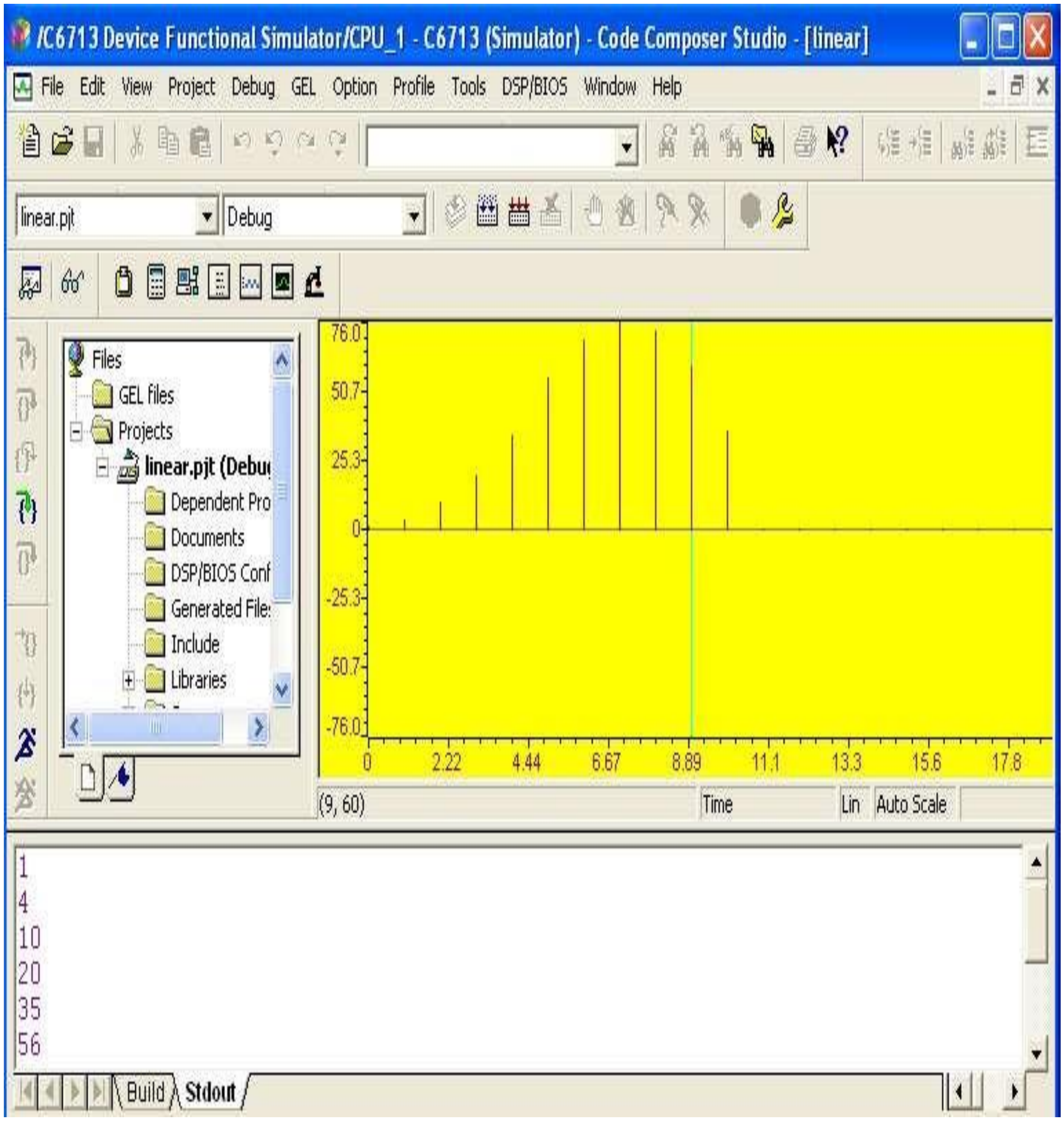
To execute project: Debug → Run

PROGRAM:- LINEAR CONVOLUTION

```
#include<stdio.h>
int m=6;
      int n=6;
      int i=0,j;
      int  x[15]={1,2,3,4,5,6,0,0,0,0,0,0};
int  h[15]={1,2,3,4,5,6,0,0,0,0,0,0};
      int y[20];
main()
{
int i;
for(i=0;i<m+n-1;i++)
{
y[i]=0;
for(j=0;j<=i;j++)
                  y[i]+=x[j]*h[i-j];
}
   for(i=0;i<m+n-1;i++)
    printf("%d \n",y[i]);
}
```


Graph Property Dialog
Display Type Single Time
Graph Title linear
Start Address y
Acquisition Buffer Size 20
Index Increment 1
Display Data Size 20
DSP Data Type 32-bit signed integer
Q-value 0
Sampling Rate (Hz) 1
Plot Data From Left to Right
Left-shifted Data Display Yes
Autoscale On
DC Value 0
Axes Display On
Time Display Unit s
Status Bar Display On
OK
Cancel
Help


Graph Property Dialog
Index Increment 1
Display Data Size 20
DSP Data Type 32-bit signed integer
Q-value 0
Sampling Rate (Hz) 1
Plot Data From Left to Right
Left-shifted Data Display Yes
Autoscale On
DC Value 0
Axes Display On
Time Display Unit s
Status Bar Display On
Magnitude Display Scale Linear
Data Plot Style Bar
Grid Style Zero Line
Cursor Mode Data Cursor
OK
Cancel
Help



OUTPUT:-

1 4 10 20 35 56 70 76 73 60 36

RESULT:

NSRIT

Internal Quality Assurance Cell (IQAC)
Rubrics for Digital Signal Processing Laboratory Course (20EC607)

Preamble: This rubric is specifically designed to assess the performance of the students in the laboratory courses where the practical exercises are being involved. The assessment broadly covers "Conduct investigations of complex problems" and more particularly "the ability" to perform experiments in a laboratory set-up (identification of equipment, initial settings for performing the required tests, perform experiment, taking necessary readings, documentation, synthesis and interpretation of results to provide solutions with valid justifications).

	Criteria	Unsatisfactory (1)	Developing (2)	Satisfactory (3)	Good (4)	Exemplary (5)	Score
A. Preparation & Observation	Criterion #1 Software Menu Identification and Procedural Programming of given Digital Signal Processing experiment.	Unable to understand and use Software menu and understanding of procedural programming techniques	Able to understand and use Software menu operations and use procedural programming techniques for Coding given algorithm	Able to understand and use Software menu operations correct recognition and application of procedural programming techniques but performs crucial errors for the given processing scheme	Reasonable understanding of software menu operation, makes no major mistakes and correctly recognizes and uses procedural programming techniques with no errors but unable to run processing scheme successfully	Demonstrates command over software menu usage with frequent use of advance menu options and Correctly recognizes and uses procedural programming techniques without errors and runs processing successfully	
B. Implementation	Criterion #2 Identifying and rectifying Errors and Graphical Visualization	Unable to check and identify the error messages and indications in software and understand and utilize visualization or plotting features	Able to identify error messages and indications in software but no understanding of detecting those errors and their types and understand and utilize visualization or plotting features with frequent errors	Able to identify error messages and indications in software as well as understanding of detecting some of those errors and their types understand and utilize visualization or plotting features but unable to compare and analyze them	Able to identify error messages in software as well as understanding of detecting all of those errors and their types but unable to rectify the, and to understand and utilize visualization and plotting features successfully, partially able to compare and analyze them	Able to find error messages in software along with the understanding to detect and rectify all errors, and to understand and utilize visualization and plotting features successfully, also able to compare and analyse them	
C.Observations & Results	Criterion #2 Simulation Observations Discussion and Conclusion:	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	
D. Viva Voce	Criterion #4 Demonstrate the ability to effectively respond to questions	Completely unable understand the pre-lab and post-lab questions.	Unable to provide an appropriate answer, or is unable to answer questions.	Misunderstand the questions and does not respond appropriately to the teacher, or has some trouble in answering questions	Able to listen carefully and respond to questions appropriately	Able to listen carefully and respond to questions appropriately; is able to explain and interpret results to the teacher	
Signature of the Lab Instructor				**Total Score Secured by the Student [(A + B + C+D)/20] x 10**			

EXPERIMENT 7

TO VERIFY THE CIRCULAR CONVOLUTION FOR DISCRETE SIGNALS USING CODE COMPOSER STUDIO (CCS)

AIM: Verify Circular Convolution on DSP Processor.

APPARATUS: DSP Trainer Kit TMS320C6713

CC Studio, PC.

Procedure to Work on Code Composer Studio

To create the New Project

Project→ New (File Name. pjt, Eg: Vectors.pjt)

To create a Source file

File →New→ Type the code (Save & give file name, Eg: sum.c).

To Add Source files to Project

Project→ Add files to Project→ sum.c

To Add rts.lib file & Hello.cmd:

Project→ Add files to Project→ rts6700.lib

Library files: rts6700.lib (Path: c:\ti\c6000\cgtools\lib\ rts6700.lib)

Note: Select Object& Library in (*.o,*.l) in Type of files

Project→ Add files to Project→hello.cmd

CMD file- Which is common for all non real time programs.

(Path: c:\ti \ tutorial\dsk6713\hello1\hello.cmd)

Note: Select Linker Command file (*.cmd) in Type of files

Compile:-

To Compile: Project→Compile

To Rebuild: project → rebuild,

Which will create the final .out executable file. (Eg.Vectors.out).

Procedure to Lode and Run program:

Load the Program to DSK: File→ Load program →Vectors.out

To Execute project: Debug → Run

PROGRAM:-

```
#include<stdio.h>
int m,n,x[30],h[30],y[30],i,j,temp[30],k,x2[30],a[30];
void main()
{
printf("enter the length of the first sequence\n");
scanf("%d",&m);
printf("enter the length of the second sequence\n");
scanf("%d",&n);
printf("enter the first sequence\n");
for(i=0;i<m;i++)
scanf("%d",&x[i]);
printf("enter the second sequence\n");
for(j=0;j<n;j++)
scanf("%d",&h[j]);
if(m-n!=0)
{
if(m>n)
{
for(i=n;i<m;i++)
h[i]=0;
n=m;
}
for(i=m;i<n;i++)
x[i]=0;
m=n;
}
y[0]=0;
```

```
a[0]=h[0];
for(j=1;j<n;j++)
a[j]=h[n-j];
for(i=0;i<n;i++)
y[0]+=x[i]*a[i];
for(k=1;k<n;k++)
{
y[k]=0;
for(j=1;j<n;j++)
x2[j]=a[j-1];
x2[0]=a[n-1];
for(i=0;i<n;i++)
{
a[i]=x2[i];
y[k]+=x[i]*x2[i];
  }
  }
  printf("the circular convolution is\n");
for(i=0;i<n;i++)
printf("%d\t",y[i]);
}
```

INPUT:-

enter the length of the 1st sequence5

enter the length of the second sequence5

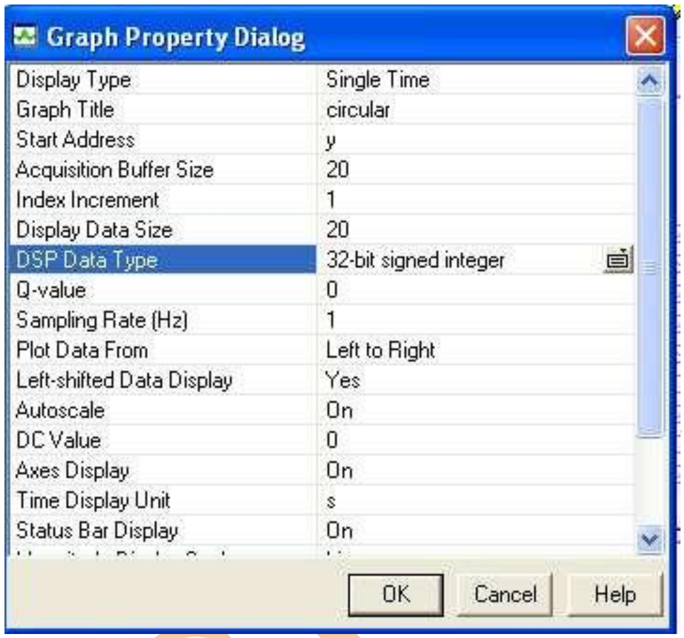
enter the 1st sequence[1 2 3 4 5]

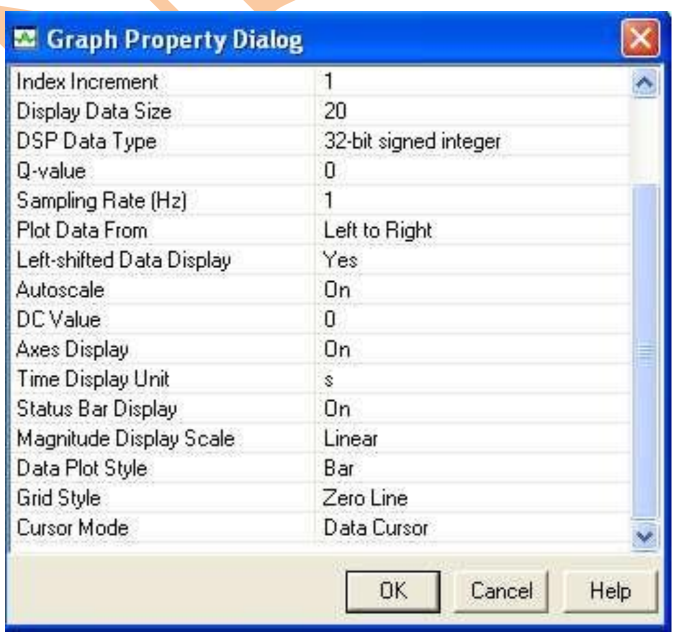
enter the second sequence [1 2 3 4 5]

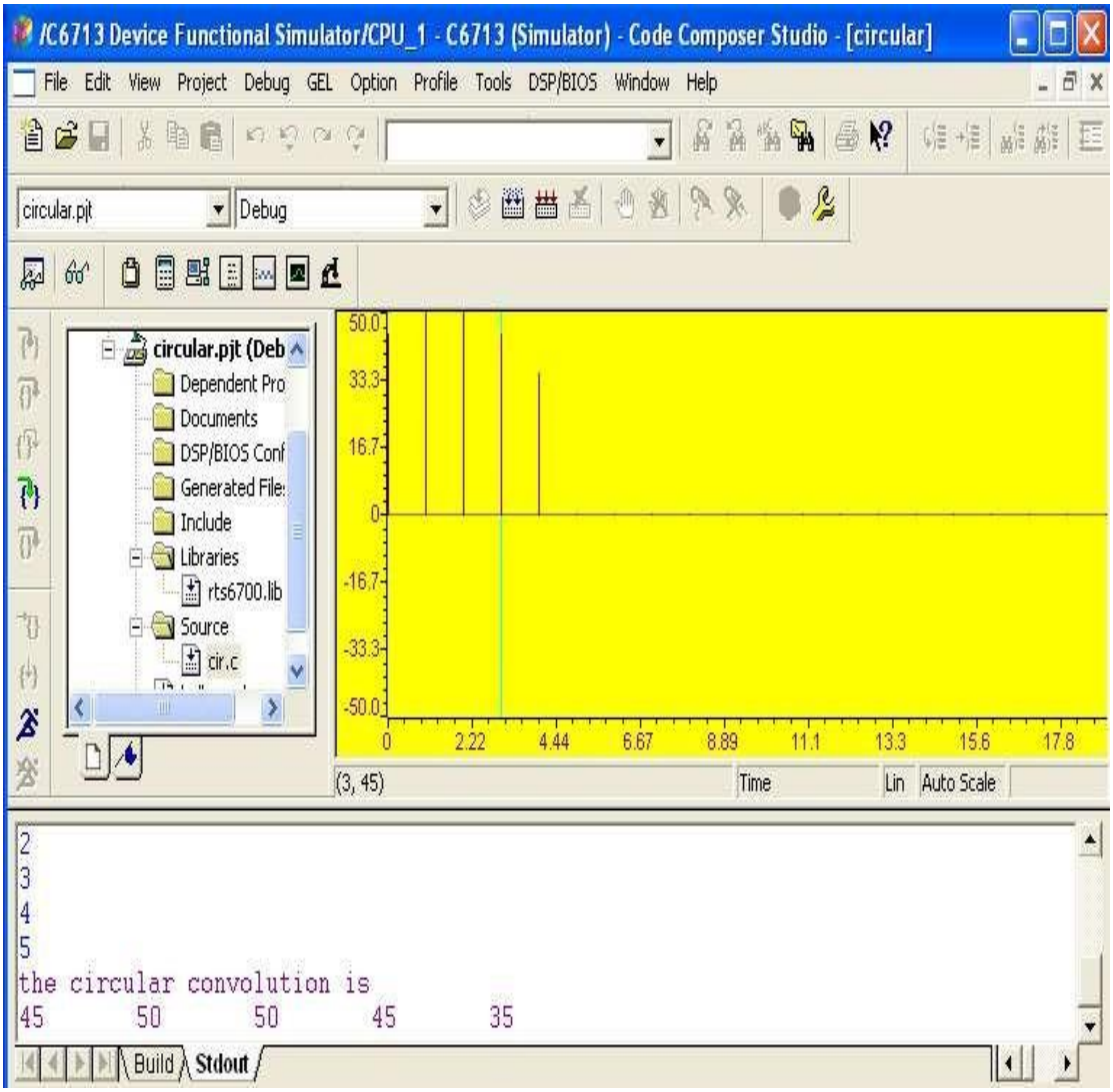
OUTPUT:-

the circular convolution is

45 50 50 45 35


Graph Property Dialog
Display Type Single Time
Graph Title circular
Start Address y
Acquisition Buffer Size 20
Index Increment 1
Display Data Size 20
DSP Data Type 32-bit signed integer
Q-value 0
Sampling Rate (Hz) 1
Plot Data From Left to Right
Left-shifted Data Display Yes
Autoscale On
DC Value 0
Axes Display On
Time Display Unit s
Status Bar Display On
OK
Cancel
Help


Graph Property Dialog
Index Increment 1
Display Data Size 20
DSP Data Type 32-bit signed integer
Q-value 0
Sampling Rate (Hz) 1
Plot Data From Left to Right
Left-shifted Data Display Yes
Autoscale On
DC Value 0
Axes Display On
Time Display Unit s
Status Bar Display On
Magnitude Display Scale Linear
Data Plot Style Bar
Grid Style Zero Line
Cursor Mode Data Cursor
OK
Cancel
Help


/C6713 Device Functional Simulator/CPU_1 - C6713 (Simulator) - Code Composer Studio - [circular]
File Edit View Project Debug GEL Option Profile Tools DSP/BIOS Window Help
circular.pjt
Debug
circular.pjt (Deb
Dependent Pro
Documents
DSP/BIOS Conf
Generated File:
Include
Libraries
rts6700.lib
Source
cir.c
50.0
33.3
16.7
0
-16.7
-33.3
-50.0
0 2.22 4.44 6.67 8.89 11.1 13.3 15.6 17.8
(3, 45)
Time
Lin
Auto Scale
2
3
4
5
the circular convolution is
45 50 50 45 35
Build
Stdout

RESULTS: -

PART-2
FILTERS

NSRIT

Internal Quality Assurance Cell (IQAC)
Rubrics for Digital Signal Processing Laboratory Course (20EC607)

Preamble: This rubric is specifically designed to assess the performance of the students in the laboratory courses where the practical exercises are being involved. The assessment broadly covers "Conduct investigations of complex problems" and more particularly "the ability" to perform experiments in a laboratory set-up (identification of equipment, initial settings for performing the required tests, perform experiment, taking necessary readings, documentation, synthesis and interpretation of results to provide solutions with valid justifications).

	Criteria	Unsatisfactory (1)	Developing (2)	Satisfactory (3)	Good (4)	Exemplary (5)	Score
A. Preparation & Observation	Criterion #1 Software Menu Identification and Procedural Programming of given Digital Signal Processing experiment.	Unable to understand and use Software menu and understanding of procedural programming techniques	Able to understand and use Software menu operations and use procedural programming techniques for Coding given algorithm	Able to understand and use Software menu operations correct recognition and application of procedural programming techniques but performs crucial errors for the given processing scheme	Reasonable understanding of software menu operation, makes no major mistakes and correctly recognizes and uses procedural programming techniques with no errors but unable to run processing scheme successfully	Demonstrates command over software menu usage with frequent use of advance menu options and Correctly recognizes and uses procedural programming techniques without errors and runs processing successfully	
B. Implementation	Criterion #2 Identifying and rectifying Errors and Graphical Visualization	Unable to check and identify the error messages and indications in software and understand and utilize visualization or plotting features	Able to identify error messages and indications in software but no understanding of detecting those errors and their types and understand and utilize visualization or plotting features with frequent errors	Able to identify error messages and indications in software as well as understanding of detecting some of those errors and their types understand and utilize visualization or plotting features but unable to compare and analyze them	Able to identify error messages in software as well as understanding of detecting all of those errors and their types but unable to rectify the, and to understand and utilize visualization and plotting features successfully, partially able to compare and analyze them	Able to find error messages in software along with the understanding to detect and rectify all errors, and to understand and utilize visualization and plotting features successfully, also able to compare and analyse them	
C.Observations & Results	Criterion #2 Simulation Observations Discussion and Conclusion:	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	
D. Viva Voce	Criterion #4 Demonstrate the ability to effectively respond to questions	Completely unable understand the pre-lab and post-lab questions.	Unable to provide an appropriate answer, or is unable to answer questions.	Misunderstand the questions and does not respond appropriately to the teacher, or has some trouble in answering questions	Able to listen carefully and respond to questions appropriately	Able to listen carefully and respond to questions appropriately; is able to explain and interpret results to the teacher	
Signature of the Lab Instructor				**Total Score Secured by the Student [(A + B + C+D)/20] x 10**			

EXPERIMENT 8

FREQUENCY RESPONSE OF IIR LOW PASS AND HIGH PASS USING BUTTERWORTHFILTER

IIR LOW PASS BUTTERWORTH FILTER

AIM: To find the frequency response of IIR Low pass Butterworth filter.

APPARATUS: MATLAB software, PC.

THEORY: IIR filters are digital filters with infinite impulse response. Unlike FIR filters, they have the feedback (a recursive part of a filter) and are known as recursive digital filters therefore. For this reason, IIR filters have much better frequency response than FIR filters of the same order. Unlike FIR filters, their phase characteristic is not linear which can cause a problem to the systems which need phase linearity. For this reason, it is not preferable to use IIR filters in digital signal processing when the phase is of the essence. Otherwise, when the linear phase characteristic is not important, the use of IIR filters is an excellent solution. There is one problem known as a potential instability that is typical of IIR filters only. FIR filters do not have such a problem as they do not have the feedback. For this reason, it is always necessary to check after the design process whether the resulting IIR filter is stable or not.

IIR FILTER DESIGN: For the given specifications to Design a digital IIR filter, first we need to design analog filter (Butterworth or Chebyshev). The resultant analog filter is transformed to digital filter by using either "Bilinear transformation or Impulse Invariant transformation". If all the infinite samples of impulse response are considered for designing of filters, then it is known as infinite impulse response filter. The differential equation of the infinite impulse response system is given by

$$\mathrm{Y(n)} = \sum_{\mathrm{K=1}}^{\mathrm{N}} AKY(n-k) + \sum_{\mathrm{k=0}}^{\mathrm{M}} Bkx(n-k)$$

$$\mathrm{H(Z)} = \mathrm{Y(Z)/X(Z)} = \sum_{\mathrm{K=0}}^{\mathrm{M}} B_{\mathrm{K}} Z^{-\mathrm{K}} \,/\, 1 - \sum_{\mathrm{K=1}}^{\mathrm{N}} A_{\mathrm{K}} Z^{-\mathrm{K}}$$

Algorithm:

Step I : Enter the pass band ripple (rp) and stop band ripple (rs).

Step II : Enter the pass band frequency (wp) and stop band frequency (ws).

Step III : Get the sampling frequency (fs).

Step IV : Calculate normalized pass band frequency, and normalized stop band frequency w1 and w2 respectively. w1 = 2 * wp /fs w2 = 2 * ws /fs

Step V : Make use of the following function to calculate order of filter Butterworth filter order [n,wn]=buttord(w1,w2,rp,rs)

Step VI : Design an nth order digital low pass Butterworth using the following statements. Butterworth filter [b, a]=butter (n, wn)

Step VII : Find the digital frequency response of the filter by using 'freqz( )' function

Step VIII : Calculate the magnitude of the frequency response in decibels (dB) mag=20*log10 (abs (H))

Step IX : Plot the magnitude response [magnitude in dB Vs normalized frequency]

Step X : Calculate the phase response using angle (H)

Step XI : Plot the phase response [phase in radians Vs normalized frequency (Hz)].

PROCEDURE:

1. Open MATLAB window and create a new program in M – file.
2. Save the program
3. Compile the program to check for address
4. Run the program and view the output graph on the screen.

PROGRAM:

```
clc;
clear all;
rp=input('enter passband ripple');
rs=input('enter stopband ripple');
wp=input('enter passband frequency');
ws=input('enter stopband frequency');
fs=input('enter sampling frequency');
w1=2*wp/fs;
w2=2*ws/fs;
[n,wn]=buttord(w1,w2,rp,rs);
[b,a]=butter(n,wn);
w=0:0.01:pi;
[h,om]=freqz(b,a,w);
m=20*log10(abs(h));
an=angle(h);
subplot(2,1,1);
plot(om/pi,m);
ylabel('gainin db');grid;
```

```
xlabel('normalised frequency');
subplot(2,1,2);
plot(om/pi,an);
ylabel('phase in radians');grid;
xlabel('normalisedfrequency');
```

WORKSPACE:

enter passband ripple0.15

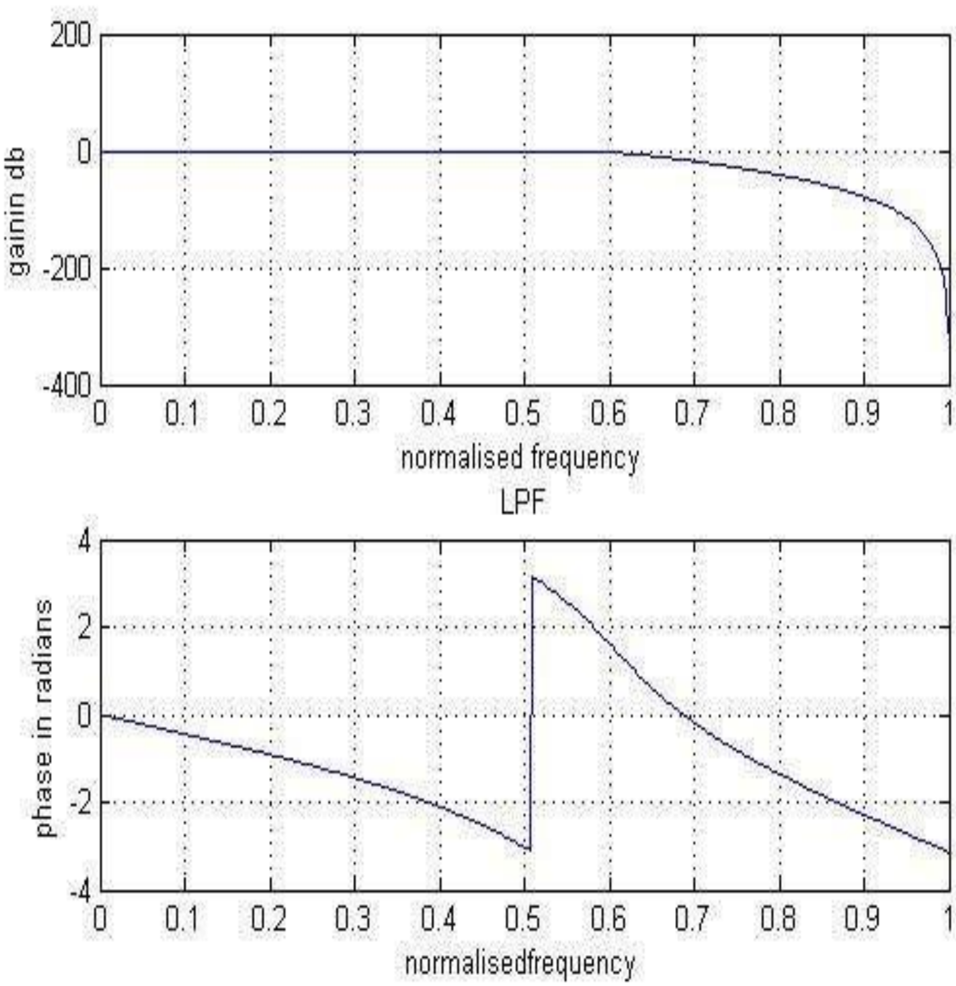
enter stopband ripple60

enter passband frequency1500

enter stopband frequency3000

enter sampling frequency7000

PLOTS:



RESULT:

IIR HIGH PASS BUTTERWORTH FILTER

AIM: To find the frequency response of IIR High pass Butterworth filter.

APPARATUS: MATLAB software,PC.

ALGORITHM:

Step I : Enter the pass band ripple (rp) and stop band ripple (rs).

Step II : Enter the pass band frequency (wp) and stop band frequency (ws).

Step III : Get the sampling frequency (fs).

Step IV : Calculate normalized pass band frequency, and normalized stop band frequency w1 and w2 respectively. w1 = 2 * wp /fs w2 = 2 * ws /fs

Step V : Make use of the following function to calculate order of filter Butterworth filter order [n,wn]=buttord(w1,w2,rp,rs)

Step VI : Design an nth order digital low pass Butterworth using the following statements. Butterworth filter [b, a]=butter (n, wn,'high')

Step VII : Find the digital frequency response of the filter by using 'freqz( )' function

Step VIII : Calculate the magnitude of the frequency response in decibels (dB)

mag=20*log10 (abs (H))

Step IX : Plot the magnitude response [magnitude in dB Vs normalized frequency]

Step X : Calculate the phase response using angle (H)

Step XI : Plot the phase response [phase in radians Vs normalized frequency (Hz)].

PROCEDURE:

1. Open MATLAB window and create a new program in M – file.
2. Save the program
3. Compile the program to check for address
4. Run the program and view the output graph on the screen.

PROGRAM:

```
clc;
clear all;
rp=input('enter passband ripple');
rs=input('enter stopband ripple');
```

```
wp=input('enter passband frequency');

ws=input('enter stopband frequency');
fs=input('enter sampling frequency');
w1=2*wp/fs;
w2=2*ws/fs;
[n,wn]=buttord(w1,w2,rp,rs);
[b,a]=butter(n,wn,'high');
w=0:0.01:pi;
[h,om]=freqz(b,a,w);
m=20*log10(abs(h));
an=angle(h);
subplot(2,1,1);
plot(om/pi,m);
ylabel('gain in db');grid;
xlabel('normalised frequency');
subplot(2,1,2);
plot(om/pi,an);
ylabel('phase in radians');grid;
xlabel('normalisedfrequency');
```

Work Sheet:

enter passband ripple0.2

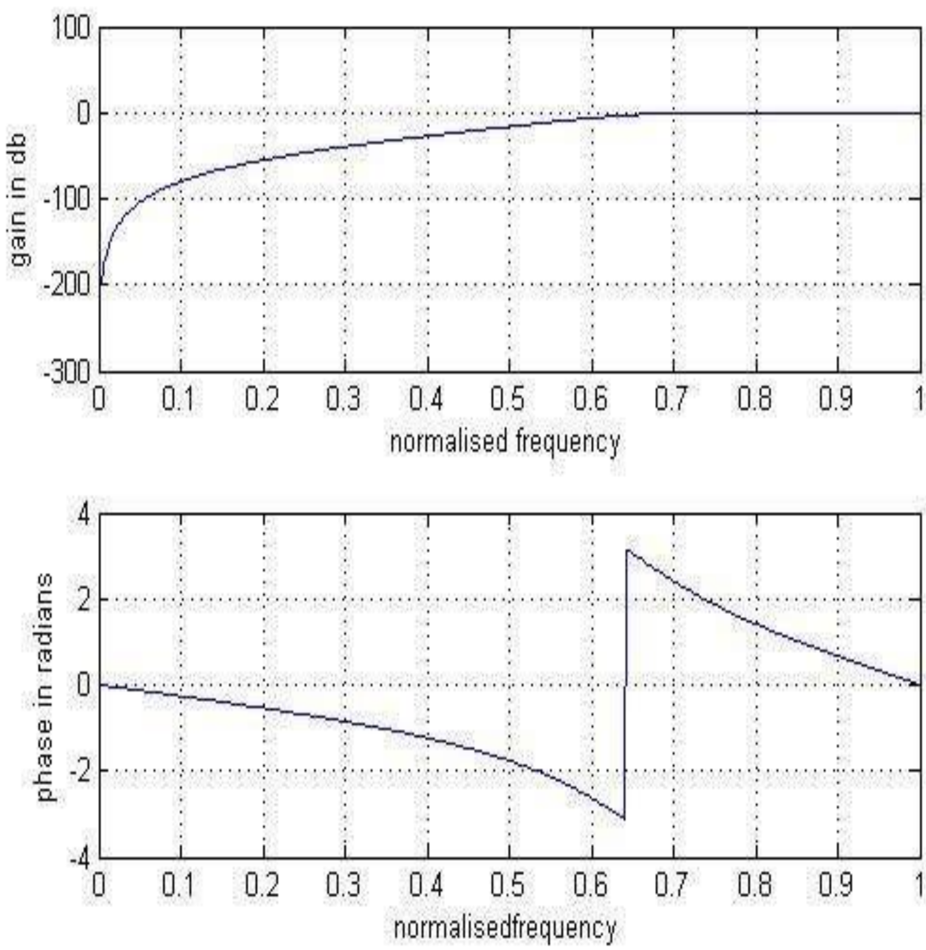
enter stopband ripple40

enter passband frequency2000

enter stopband frequency3500

enter sampling frequency8000

PLOTS:



RESULT:

NSRIT

Internal Quality Assurance Cell (IQAC)
Rubrics for Digital Signal Processing Laboratory Course (20EC607)

Preamble: This rubric is specifically designed to assess the performance of the students in the laboratory courses where the practical exercises are being involved. The assessment broadly covers "Conduct investigations of complex problems" and more particularly "the ability" to perform experiments in a laboratory set-up (identification of equipment, initial settings for performing the required tests, perform experiment, taking necessary readings, documentation, synthesis and interpretation of results to provide solutions with valid justifications).

	Criteria	Unsatisfactory (1)	Developing (2)	Satisfactory (3)	Good (4)	Exemplary (5)	Score
A. Preparation & Observation	Criterion #1 Software Menu Identification and Procedural Programming of given Digital Signal Processing experiment.	Unable to understand and use Software menu and understanding of procedural programming techniques	Able to understand and use Software menu operations and use procedural programming techniques for Coding given algorithm	Able to understand and use Software menu operations correct recognition and application of procedural programming techniques but performs crucial errors for the given processing scheme	Reasonable understanding of software menu operation, makes no major mistakes and correctly recognizes and uses procedural programming techniques with no errors but unable to run processing scheme successfully	Demonstrates command over software menu usage with frequent use of advance menu options and Correctly recognizes and uses procedural programming techniques without errors and runs processing successfully	
B. Implementation	Criterion #2 Identifying and rectifying Errors and Graphical Visualization	Unable to check and identify the error messages and indications in software and understand and utilize visualization or plotting features	Able to identify error messages and indications in software but no understanding of detecting those errors and their types and understand and utilize visualization or plotting features with frequent errors	Able to identify error messages and indications in software as well as understanding of detecting some of those errors and their types understand and utilize visualization or plotting features but unable to compare and analyze them	Able to identify error messages in software as well as understanding of detecting all of those errors and their types but unable to rectify the, and to understand and utilize visualization and plotting features successfully, partially able to compare and analyze them	Able to find error messages in software along with the understanding to detect and rectify all errors, and to understand and utilize visualization and plotting features successfully, also able to compare and analyse them	
C.Observations & Results	Criterion #2 Simulation Observations Discussion and Conclusion:	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	
D. Viva Voce	Criterion #4 Demonstrate the ability to effectively respond to questions	Completely unable understand the pre-lab and post-lab questions.	Unable to provide an appropriate answer, or is unable to answer questions.	Misunderstand the questions and does not respond appropriately to the teacher, or has some trouble in answering questions	Able to listen carefully and respond to questions appropriately	Able to listen carefully and respond to questions appropriately; is able to explain and interpret results to the teacher	
Signature of the Lab Instructor				**Total Score Secured by the Student [(A + B + C+D)/20] x 10**			

EXPERIMENT 9

FREQUENCY RESPONSE OF IIR LOW PASS AND HIGH PASS USING CHEBYSHEV FILTER

IIR LOWPASS CHEBYSHEV FILTER

AIM: To find the frequency response of IIR Low pass Chebyshev filter.

APPARATUS: MATLAB software,PC.

Algorithm:

Step I : Enter the pass band ripple (rp) and stop band ripple (rs).

Step II : Enter the pass band frequency (wp) and stop band frequency (ws).

Step III : Get the sampling frequency (fs).

Step IV : Calculate normalized pass band frequency, and normalized stop band frequency w1 and w2 respectively. w1 = 2 * wp /fs w2 = 2 * ws /fs

Step V : Make use of the following function to calculate order of filter Chebyshev filter order [n,wn]=cheb1ord(w1,w2,rp,rs)

Step VI : Design an nth order digital low pass Chebyshev filter using the following statements. [b,a]=cheby1 (n, 0.5, wn)

Step VII : Find the digital frequency response of the filter by using 'freqz( )' function

Step VIII : Calculate the magnitude of the frequency response in decibels (dB) mag=20*log10 (abs (H))

Step IX : Plot the magnitude response [magnitude in dB Vs normalized frequency]

Step X : Calculate the phase response using angle (H)

Step XI : Plot the phase response [phase in radians Vs normalized frequency (Hz)].**PROCEDURE:**

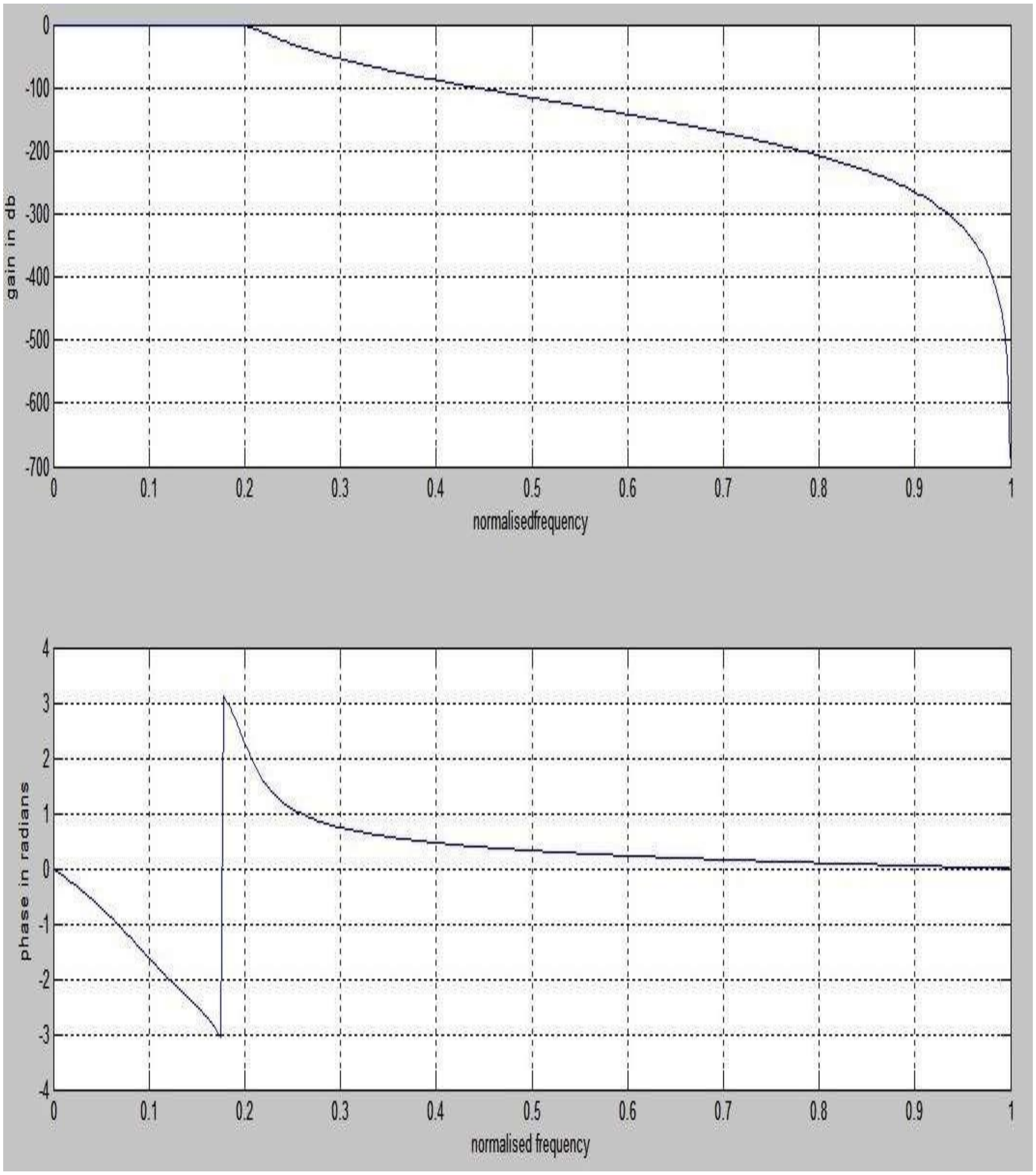
1. Open MATLAB window and create a new program in M – file.
2. Save the program
3. Compile the program
4. Run the program and view the output graph on the screen.

PROGRAM:

```
clc;
clear all;
alphap=1;
```

```
alphas=15;
wp=0.2*pi;
ws=0.3*pi;
[n,wn]=cheb1ord(wp/pi,ws/pi,alphap,alphas);
[b,a]=cheby1(n,alphap,wn);
w=0:0.01:pi;
[h,ph]=freqz(b,a,w);
m=20*log(abs(h));
an=angle(h);
subplot(2,1,1);
plot(ph/pi,m);
title('magnitude response of IIR LP filter is');
grid;
ylabel('gain in db');
xlabel('normalisedfrequency');
subplot(2,1,2);
plot(ph/pi,an);
title('Phase response of IIR LP filter is');
grid;
ylabel('phase in radians');
xlabel('normalised  frequency');
```

PLOTS:



RESULT:

IIR HIGHPASS CHEBYSHEV FILTER

AIM: To find the frequency response of IIR High pass Chebyshev filter.

APPARATUS: MATLAB software, PC.

Algorithm:

Step I : Enter the pass band ripple (rp) and stop band ripple (rs).

Step II : Enter the pass band frequency (wp) and stop band frequency (ws).

Step III : Get the sampling frequency (fs).

Step IV : Calculate normalized pass band frequency, and normalized stop band frequency w1 and w2 respectively. w1 = 2 * wp /fs w2 = 2 * ws /fs

Step V : Make use of the following function to calculate order of filter Chebyshev filter order [n,wn]=cheb1ord(w1,w2,rp,rs)

Step VI : Design an nth order digital low pass Chebyshev filter using the following statements. [b,a]=cheby1 (n, 0.5, wn)

Step VII : Find the digital frequency response of the filter by using 'freqz( )' function

Step VIII : Calculate the magnitude of the frequency response in decibels (dB) mag=20*log10 (abs (H))

Step IX : Plot the magnitude response [magnitude in dB Vs normalized frequency]

Step X : Calculate the phase response using angle (H)

Step XI : Plot the phase response [phase in radians Vs normalized frequency (Hz)].**PROCEDURE:**

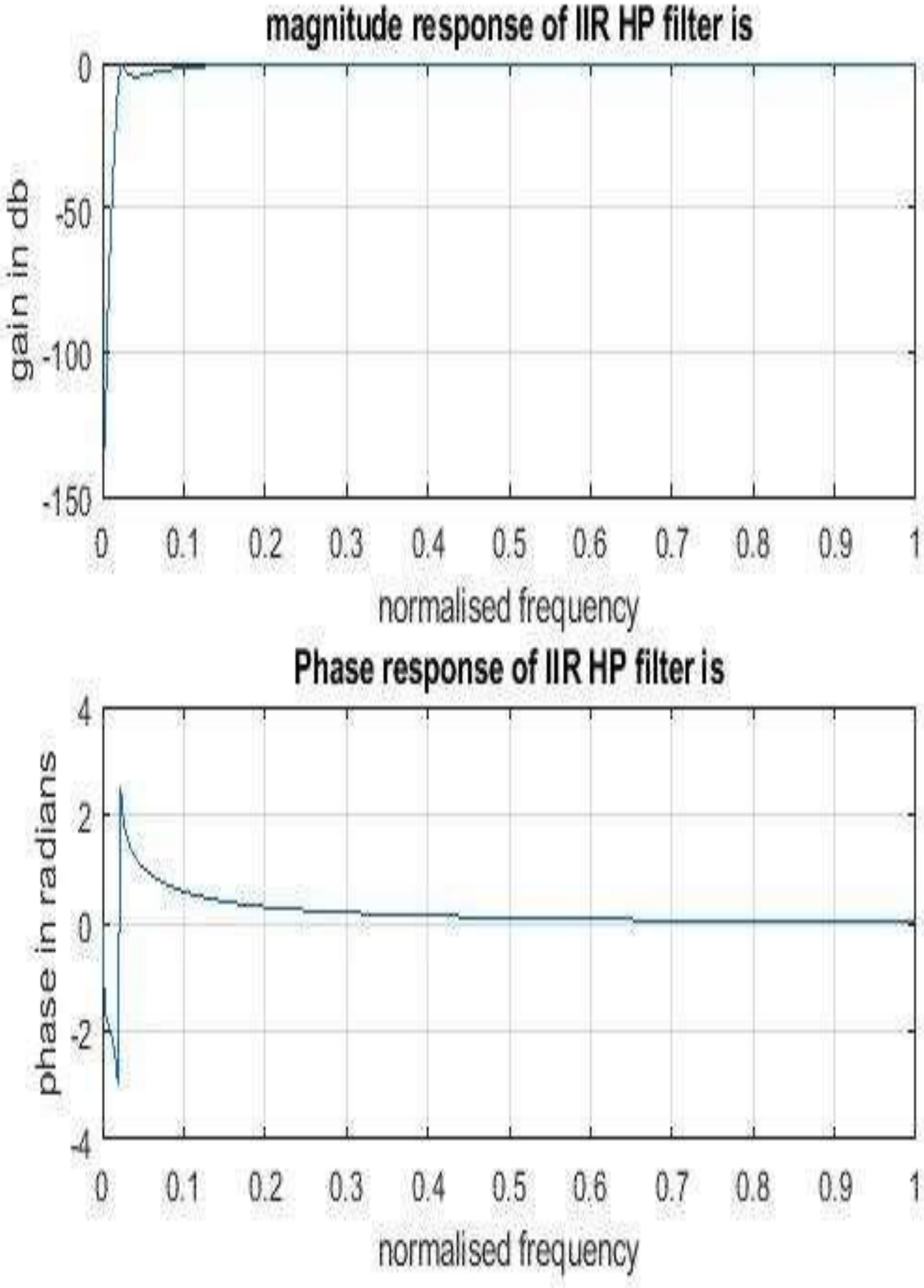
1. Open MATLAB window and create a new program in M – file.
2. Save the program
3. Compile the program
4. Run the program and view the output graph on the screen.

PROGRAM:

```
clc;
clear all;
alphap=2;
alphas=20;
wp=0.02
ws=0.01;
[n,wn]=cheb1ord(wp,ws,alphap,alphas);
```

```
[b,a]=cheby1(n,alphap,wn,'high');
w=0:0.01:pi;
[h,ph]=freqz(b,a,w);
m=20*log(abs(h));
an=angle(h);
subplot(2,1,1);
plot(ph/pi,m);
title('magnitude response of IIR HP filter is');
grid;
ylabel('gain in db');
xlabel('normalised frequency');
subplot(2,1,2);
plot(ph/pi,an);
title('Phase response of IIR HP filter is');
grid;
ylabel('phase in radians');
xlabel('normalised  frequency');
```

PLOTS:



RESULT:

Internal Quality Assurance Cell (IQAC)
Rubrics for Digital Signal Processing Laboratory Course (20EC607)

Preamble: This rubric is specifically designed to assess the performance of the students in the laboratory courses where the practical exercises are being involved. The assessment broadly covers "Conduct investigations of complex problems" and more particularly "the ability" to perform experiments in a laboratory set-up (identification of equipment, initial settings for performing the required tests, perform experiment, taking necessary readings, documentation, synthesis and interpretation of results to provide solutions with valid justifications).

	Criteria	Unsatisfactory (1)	Developing (2)	Satisfactory (3)	Good (4)	Exemplary (5)	Score
A. Preparation & Observation	Criterion #1 Software Menu Identification and Procedural Programming of given Digital Signal Processing experiment.	Unable to understand and use Software menu and understanding of procedural programming techniques	Able to understand and use Software menu operations and use procedural programming techniques for Coding given algorithm	Able to understand and use Software menu operations correct recognition and application of procedural programming techniques but performs crucial errors for the given processing scheme	Reasonable understanding of software menu operation, makes no major mistakes and correctly recognizes and uses procedural programming techniques with no errors but unable to run processing scheme successfully	Demonstrates command over software menu usage with frequent use of advance menu options and Correctly recognizes and uses procedural programming techniques without errors and runs processing successfully	
B. Implementation	Criterion #2 Identifying and rectifying Errors and Graphical Visualization	Unable to check and identify the error messages and indications in software and understand and utilize visualization or plotting features	Able to identify error messages and indications in software but no understanding of detecting those errors and their types and understand and utilize visualization or plotting features with frequent errors	Able to identify error messages and indications in software as well as understanding of detecting some of those errors and their types understand and utilize visualization or plotting features but unable to compare and analyze them	Able to identify error messages in software as well as understanding of detecting all of those errors and their types but unable to rectify the, and to understand and utilize visualization and plotting features successfully, partially able to compare and analyze them	Able to find error messages in software along with the understanding to detect and rectify all errors, and to understand and utilize visualization and plotting features successfully, also able to compare and analyse them	
C.Observations & Results	Criterion #2 Simulation Observations Discussion and Conclusion:	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	
D. Viva Voce	Criterion #4 Demonstrate the ability to effectively respond to questions	Completely unable understand the pre-lab and post-lab questions.	Unable to provide an appropriate answer, or is unable to answer questions.	Misunderstand the questions and does not respond appropriately to the teacher, or has some trouble in answering questions	Able to listen carefully and respond to questions appropriately	Able to listen carefully and respond to questions appropriately; is able to explain and interpret results to the teacher	
Signature of the Lab Instructor				**Total Score Secured by the Student [(A + B + C+D)/20] x 10**			

EXPERIMENT 10

FREQUENCY RESPONSE OF FIR LOW PASS FILTER USING RECTANGLE WINDOW

AIM : To design FIR low pass and high pass filters using rectangle window.

APPARATUS : MATLAB software,

Personal Computer.

THEORY :

- The rectangular window sequence is given by

$w_R(n) = 1$ for $-(N-1)/2 \leq n \leq (N-1)/2$

$= 0$ otherwise

- The spectrum of the Rectangular window is given by

$$w_R(e^{jw}) \quad \sum_{n=-(N-1)/2}^{(n-1)/2} \quad ^{-jwn} = \frac{\sin\frac{WN}{2}}{\sin\frac{W}{2}}$$

- The frequency response is real and it's zero occurs when $\frac{NW}{2} = 19\pi$ or $w = \frac{2k}{N}$ where k is an integer. The response for w between $2\pi N$ and - $2\pi N$ is called the "main lobe" of that and the other lobes are known as "side lobes". The main lobe of the response is that portion that lies between the first two zero crossing, the main lobe width for the rectangular window is equal to $\frac{4\pi}{N}$.
- The width of the transition region depends on the width of the main lobe.

ALGORITHM:

1. Get the pass band and stop band ripple.
2. Get the pass band and stop band edge frequencies.
3. Get the sampling frequency.
4. Calculate order of the filter.
5. Find window co – efficient using $w_R(n) = 1$ for $n \leq (N-1)/2$
   $= 0$ otherwise
6. Draw the magnitude and phase response.
7. Repeat the same for high pass filter.
8.

PROCEDURE:

1. Open MATLAB window and create a new program in M – file.
2. Save the program

3. Compile the program to check for address.
4. Run the program and view the output graph on the screen.

PROGRAM:

```
clc;
clear all;
format long;
rs=input('enter the stopband ripple');
rp=input('enter the passband ripple');
fs=input('enter the stopband frequency');
fp=input('enter the passband frequency');
f=input('enter the modulating frequency');
n=(-20*log10(sqrt(rs*rp))-13)/(14.6*(fs-fp)/f);
n=ceil(n);
n1=n+1;
wp=2*fp/f;
ws=2*fs/f;
if(rem(n,2)~=0)
n1=n;
n=n-1;
end
y=rectwin(n1);
subplot(3,1,1);
stem(y);
title('rectangular window')
b=fir1(n,wp,y);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));
subplot(3,1,2);
plot(o/pi,m);
xlabel('normalised frequency');grid;
ylabel('gain in db');
```

```
title('LPF')
b=fir1(n,wp,'high',y);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));
subplot(3,1,3);
plot(o/pi,m);
xlabel('normalised frequency');grid;
ylabel('gain in db');
title('HPF')
```

Work Sheet:

enter the stopband ripple0.002

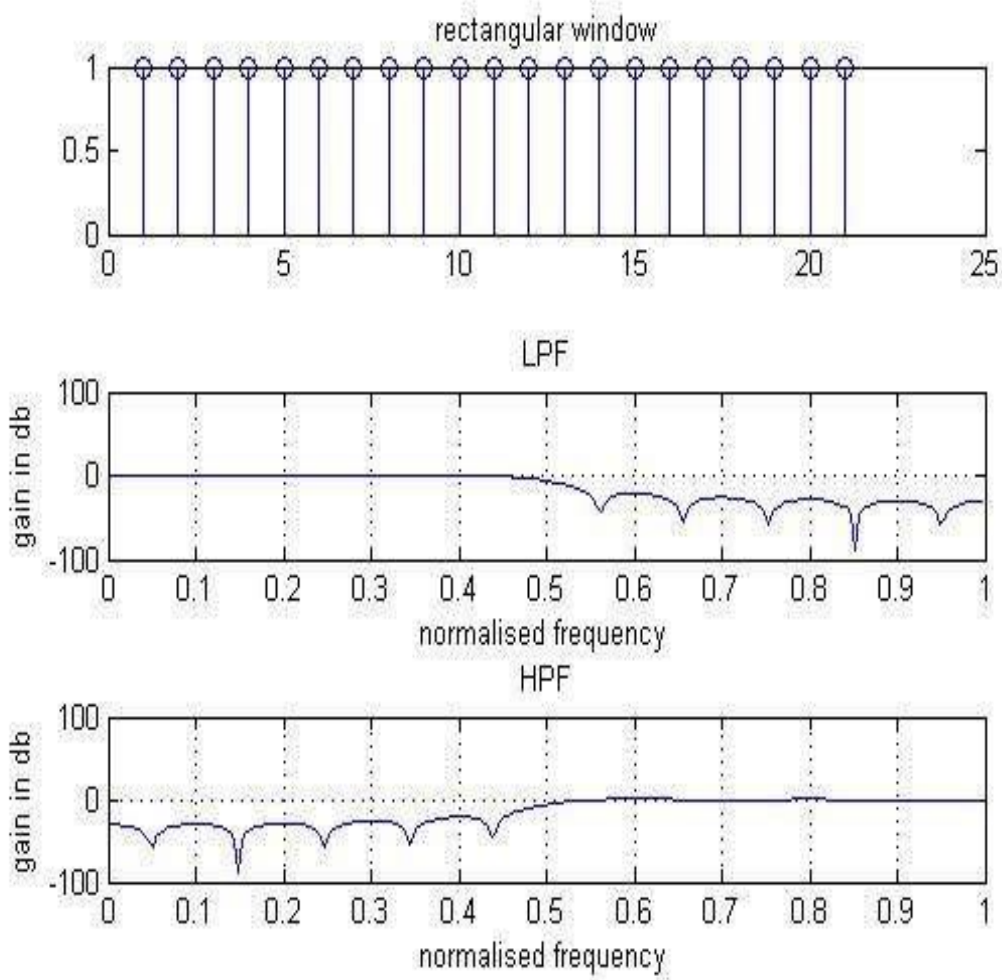
enter the passband ripple0.8

enter the stopband frequency600

enter the passband frequency500

enter the modulating frequency2000

PLOTS:



RESULT:

NSRIT

Internal Quality Assurance Cell (IQAC)
Rubrics for Digital Signal Processing Laboratory Course (20EC607)

Preamble: This rubric is specifically designed to assess the performance of the students in the laboratory courses where the practical exercises are being involved. The assessment broadly covers "Conduct investigations of complex problems" and more particularly "the ability" to perform experiments in a laboratory set-up (identification of equipment, initial settings for performing the required tests, perform experiment, taking necessary readings, documentation, synthesis and interpretation of results to provide solutions with valid justifications).

	Criteria	Unsatisfactory (1)	Developing (2)	Satisfactory (3)	Good (4)	Exemplary (5)	Score
A. Preparation & Observation	Criterion #1 Software Menu Identification and Procedural Programming of given Digital Signal Processing experiment.	Unable to understand and use Software menu and understanding of procedural programming techniques	Able to understand and use Software menu operations and use procedural programming techniques for Coding given algorithm	Able to understand and use Software menu operations correct recognition and application of procedural programming techniques but performs crucial errors for the given processing scheme	Reasonable understanding of software menu operation, makes no major mistakes and correctly recognizes and uses procedural programming techniques with no errors but unable to run processing scheme successfully	Demonstrates command over software menu usage with frequent use of advance menu options and Correctly recognizes and uses procedural programming techniques without errors and runs processing successfully	
B. Implementation	Criterion #2 Identifying and rectifying Errors and Graphical Visualization	Unable to check and identify the error messages and indications in software and understand and utilize visualization or plotting features	Able to identify error messages and indications in software but no understanding of detecting those errors and their types and understand and utilize visualization or plotting features with frequent errors	Able to identify error messages and indications in software as well as understanding of detecting some of those errors and their types understand and utilize visualization or plotting features but unable to compare and analyze them	Able to identify error messages in software as well as understanding of detecting all of those errors and their types but unable to rectify the, and to understand and utilize visualization and plotting features successfully, partially able to compare and analyze them	Able to find error messages in software along with the understanding to detect and rectify all errors, and to understand and utilize visualization and plotting features successfully, also able to compare and analyse them	
C.Observations & Results	Criterion #2 Simulation Observations Discussion and Conclusion:	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	
D. Viva Voce	Criterion #4 Demonstrate the ability to effectively respond to questions	Completely unable understand the pre-lab and post-lab questions.	Unable to provide an appropriate answer, or is unable to answer questions.	Misunderstand the questions and does not respond appropriately to the teacher, or has some trouble in answering questions	Able to listen carefully and respond to questions appropriately	Able to listen carefully and respond to questions appropriately; is able to explain and interpret results to the teacher	
Signature of the Lab Instructor				**Total Score Secured by the Student [(A + B + C+D)/20] x 10**			

EXPERIMENT 11

FREQUENCY RESPONSE OF FIR LOW PASS FILTER USING TRIANGLE WINDOW

AIM: to design the FIR low pass and high pass filter using Triangular window

APPARATUS: MATLAB software,PC.

THEORY: the N – point triangular window is given by

$$w_T(n) = 1 - \frac{2(n)}{N-1} \quad \text{for } \frac{-(N-1)}{2} \leq n \leq \frac{(N-1)}{2}$$

The Fourier transformation of the Triangular window is

$$w_T(e^{jw}) = [\ (\sin(n-1/4)w)/\ \sin(w/2)]^2$$

- The side band is smaller than that of Triangular window being reduced from -13db to -25 db. However the main lobe is now $\frac{8\pi}{N}$ or twice that of rectangular window.
- The disadvantage which compared to magnitude response obtained by using rectangular window is the transition region is more , the attenuation in stop band is less

ALGORITHM:

1. Get the pass band and stop band ripples.
2. Get the pass band and stop band edge frequencies
3. Get the sampling frequency.
4. Calculate the order of the filter.
5. Find the window coefficient using
   $w_R(n) = 1 - [2(n)/N-1]$ for $-(N-1)/2 \leq n \leq (N-1)/2$
   $= 0$ otherwise
6. Draw magnitude and phase response.
7. Repeat the same for high pas filter.

PROCEDURE:

1. Open MATLAB window and create a new program in M – file
2. Save the program
3. Compile the program to check for errors.
4. Run the program and view the output graph on the screen.

PROGRAM:

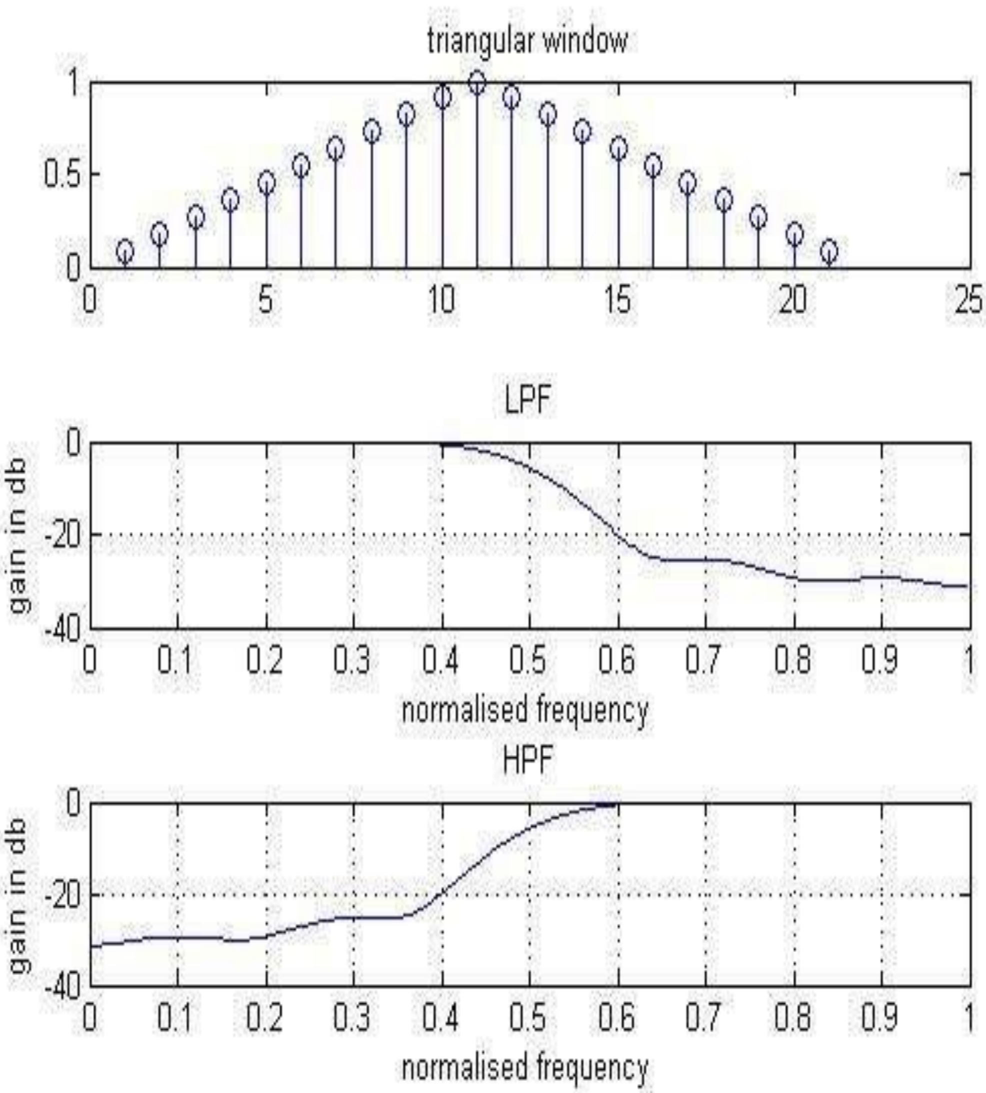
```
clc;
clear all;
close all;
format long;
rs=input('enter the stopband ripple');
rp=input('enter the passband ripple');
fs=input('enter the stopband frequency');
fp=input('enter the passband frequency');
f=input('enter the modulating frequency');
n=(-20*log10(sqrt(rs*rp))-13)/(14.6*(fs-fp)/f);
n=ceil(n);
n1=n+1;
wp=2*fp/f;
ws=2*fs/f;
if(rem(n,2)~=0)
   n1=n;
   n=n-1;
end
y=triang(n1)
subplot(3,1,1);
stem(y);
title('triangular window');
b=fir1(n,wp,y);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));
subplot(3,1,2);
plot(o/pi,m);
xlabel('normalised frequency');grid;
ylabel('gain in db');
title('LPF');
```

```
b=fir1(n,wp,'high',y);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));
subplot(3,1,3);
plot(o/pi,m);
xlabel('normalised frequency');grid;
ylabel('gain in db');
title(‘HPF’);
```

Work Sheet:

enter the stopband ripple0.002
enter the passband ripple0.8
enter the stopband frequency600
enter the passband frequency500
enter the modulating frequency2000

PLOTS:



RESULT:

ADDITIONAL EXPERIMENT

NSRIT

Internal Quality Assurance Cell (IQAC)
Rubrics for Digital Signal Processing Laboratory Course (20EC607)

Preamble: This rubric is specifically designed to assess the performance of the students in the laboratory courses where the practical exercises are being involved. The assessment broadly covers "Conduct investigations of complex problems" and more particularly "the ability" to perform experiments in a laboratory set-up (identification of equipment, initial settings for performing the required tests, perform experiment, taking necessary readings, documentation, synthesis and interpretation of results to provide solutions with valid justifications).

	Criteria	Unsatisfactory (1)	Developing (2)	Satisfactory (3)	Good (4)	Exemplary (5)	Score
A. Preparation & Observation	Criterion #1 Software Menu Identification and Procedural Programming of given Digital Signal Processing experiment.	Unable to understand and use Software menu and understanding of procedural programming techniques	Able to understand and use Software menu operations and use procedural programming techniques for Coding given algorithm	Able to understand and use Software menu operations correct recognition and application of procedural programming techniques but performs crucial errors for the given processing scheme	Reasonable understanding of software menu operation, makes no major mistakes and correctly recognizes and uses procedural programming techniques with no errors but unable to run processing scheme successfully	Demonstrates command over software menu usage with frequent use of advance menu options and Correctly recognizes and uses procedural programming techniques without errors and runs processing successfully	
B. Implementation	Criterion #2 Identifying and rectifying Errors and Graphical Visualization	Unable to check and identify the error messages and indications in software and understand and utilize visualization or plotting features	Able to identify error messages and indications in software but no understanding of detecting those errors and their types and understand and utilize visualization or plotting features with frequent errors	Able to identify error messages and indications in software as well as understanding of detecting some of those errors and their types understand and utilize visualization or plotting features but unable to compare and analyze them	Able to identify error messages in software as well as understanding of detecting all of those errors and their types but unable to rectify the, and to understand and utilize visualization and plotting features successfully, partially able to compare and analyze them	Able to find error messages in software along with the understanding to detect and rectify all errors, and to understand and utilize visualization and plotting features successfully, also able to compare and analyse them	
C.Observations & Results	Criterion #2 Simulation Observations Discussion and Conclusion:	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	
D. Viva Voce	Criterion #4 Demonstrate the ability to effectively respond to questions	Completely unable understand the pre-lab and post-lab questions.	Unable to provide an appropriate answer, or is unable to answer questions.	Misunderstand the questions and does not respond appropriately to the teacher, or has some trouble in answering questions	Able to listen carefully and respond to questions appropriately	Able to listen carefully and respond to questions appropriately; is able to explain and interpret results to the teacher	
Signature of the Lab Instructor				**Total Score Secured by the Student [(A + B + C+D)/20] x 10**			

EXPERIMENT 12

FFT OF 1-D SIGNAL

AIM: To compute FFT of a one-dimension signal.
APPARATUS: MATLAB Software, PC

THEORY:

The first Fourier transform algorithm explains two basic properties of twiddle factor and reduces number of complex multiplication required to perform DFT from N to N/2 log_2 N.In other words for N=1024 this implies about zero instead of 10^6 multiplication a reduction factor.

FFT algorithm are based on fundamental principle of decomposing computation of discrete FT of sequence of length N into successfully small discrete FT, there are basically two classes of FFT algorithm. They are:
1) Decimation in time.
2) Decimation in frequency.

ALGORITHM:
1) Enter the length of the ramp sequence
2) Assign ramp sequence to input sequence y(n).
3) Enter the length of FFT
4) Now apply FFT to given one dimensional ramp signal.
5) Plot the magnitude response and phase response corresponding FFT sequence

PROCEDURE:
1. Open MATLAB window and create a new program in M – file.
2. Save the program.

3.Run the program and view the output graph on the screen.

PROGRAM:

```
clc;
clear all;
close all;
s=input ('enter the length of Ramp sequence: ');
t=0:s;
y=t
subplot(3,1,1);
stem(t,y);
grid
```

```
input('y=');
disp(y);
title ('ramp Sequence');
xlabel ('time -->');
ylabel ('--> Amplitude');
xn=y;
N=input('enter the length of the FFT sequence: ');
xk=fft(xn,N);
magxk=abs(xk);
angxk=angle(xk);
k=0:N-1;
subplot(3,1,2);
stem(k,magxk);
grid
xlabel('k');
ylabel('|x(k)|');
subplot(3,1,3);
stem(k,angxk);
disp(xk);
grid
xlabel('k');
ylabel('ang(x(k))');
```

OUTPUT:

enter the length of Ramp sequence: 7

y = 0 1 2 3 4 5 6 7

y=[0 1 2 3 4 5 6 7]

0 1 2 3 4 5 6 7

enter the length of the FFT sequence: 8

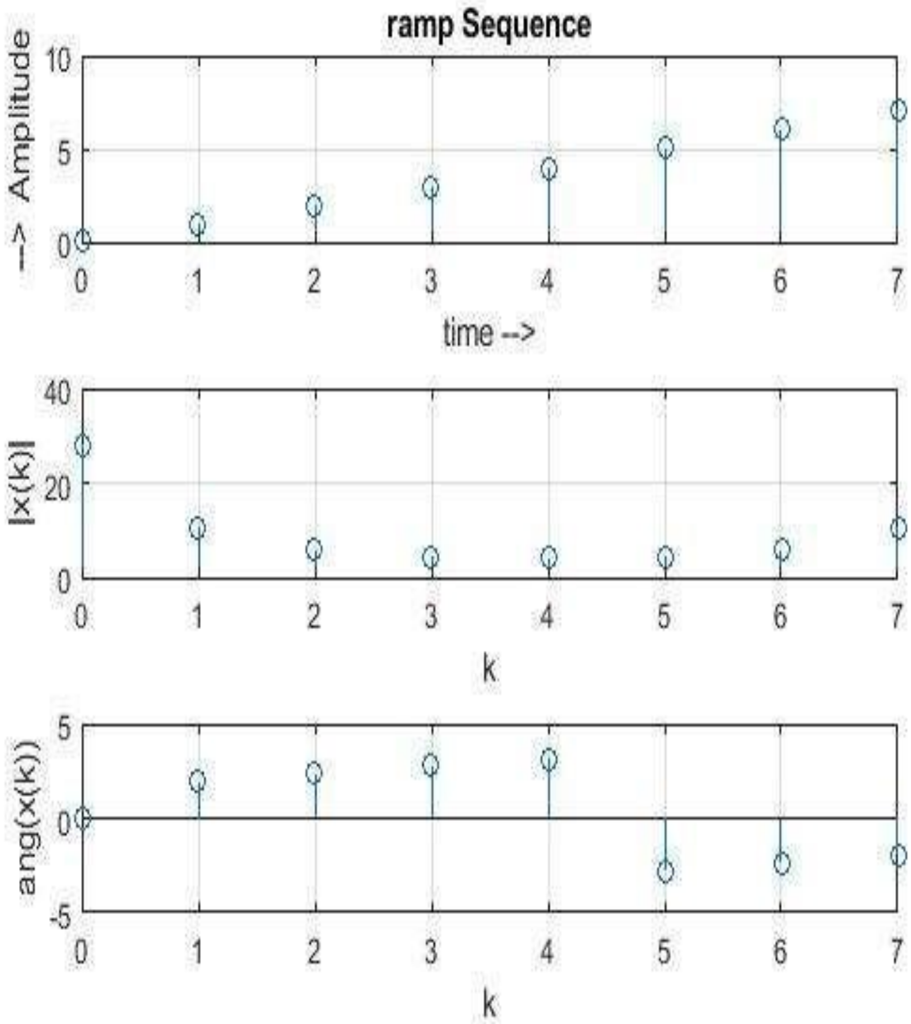
Columns 1 through 6

28.0000 + 0.0000i -4.0000 + 9.6569i -4.0000 + 4.0000i -4.0000 + 1.6569i -4.0000 + 0.0000i -4.0000 - 1.6569i

Columns 7 through 8

-4.0000 - 4.0000i -4.0000 - 9.6569i

Plot:


ramp Sequence
--> Amplitude
10
5
0
0 1 2 3 4 5 6 7
time -->
|x(k)|
40
20
0
0 1 2 3 4 5 6 7
k
ang(x(k))
5
0
-5
0 1 2 3 4 5 6 7
k

RESULT: