**PRACTICAL WORK BOOK**

**For Academic Session Fall 2018**

**Digital Signal Processing (EE-394)**

**For**

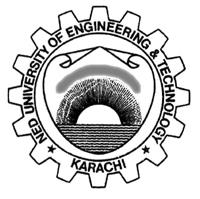
**TE Electrical**

Name: **Muhammad Umer**

Roll Number: **EE-17181** Year: **TE**

Section: **D**

Batch: **2017-18**



**Department of Electrical Engineering**

**N.E.D. University of Engineering & Technology, Karachi**

**CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lab.**  **No.** | **List of Experiments** | **Page No.** | **Remarks** |
| 1 | Introduction to Digital Signal Processing |  |  |
| 2 | Analog to Digital Conversion |  |  |
| 3 | Analog to Digital Conversion |  |  |
| 4 | Open Ended Lab 1 |  |  |
| 5 | Discrete time modelling of continuous time system |  |  |
| 6 | Convolution for Discrete Time Signals |  |  |
| 7 | Introduction to Continuous time Fourier series (CTFS) |  |  |
| 8 | Analysis and Synthesis of Signals through Discrete Fourier Transform (DFT) |  |  |
| 9 | To become familiar with practical constraints in calculating Fourier Transform of any real world signal |  |  |
| 10 | Open Ended Lab 02 |  |  |
| 11 | To understand the concept of window overlapping |  |  |

**Lab Rubrics:**

1. Properly formatted lab document (report writing skill)

2. Understanding of the concepts delivered.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No.** | **Rubric No.** | **Grading** | | | **Score** |
| 1 | 1 | **Exemplary (100%):**  Lab report was well written and properly formatted with good delivery of the required concepts. | **Adequate (70%):**  Lab report was written with some needed improvements | **Poor (40%):**  Poorly written lab content. |  |
| 2 | 2 | **Exemplary (100%):**  Clearly describes the objectives of the lab as well as the concepts learned. | **Adequate (70%):**  Describes the objective and concepts related to lab with some deficiency. | **Poor (40%):**  Inaccurate understanding of the concepts related to lab session. |  |

|  |
| --- |
| **Laboratory Session No. 01** |

**Objective:**

***To get introduced with fundamentals of Digital Signal Processing***

**Post Lab Exercises:**

**Question 1:**

What do you mean by the term digital? Explain it briefly.

**Answer:**

A **digital** signal can only have a set of finite values. If the **digital** signal is \*binary\*, it can only have 2 values, 0 or 1. A PWM signal is an example of a continuous signal (NOT **discrete**), and binary **digital** having just 2 values. A **discrete** signal is one that is defined only at specific time intervals.

**Question 2:**

Write some (at least three) applications of DSP related to electrical (power) engineering.

**Answer:**

* Digital Signal Processing in Power System Protection and Control bridges the gap between the theory of protection and control and the practical applications of protection equipment. Digital Signal Processing in Power System Protection and Control can be useful for protection engineers working in utilities at various levels of the electricity network
* Power conversion systems are composed of different electrical and electronic components that need to be managed. Moreover, such systems are designed to work under different conditions and states; thus, several control algorithms can be found in a single DSP
* Sonar is an application of digital signal processing (DSP); sonar uses sound propagation to navigate and communicate with or detect an object under the surface of the water. Generally, two types of technologies are used in sonar: passive sonar and active sonar technologies. Passive sonar is used to listen to the sound of the vessels; active sonar is used to release pulses of sounds and to listen to echoes. Sonar may also be used for acoustic measurements.

|  |
| --- |
| **Laboratory Session No. 02** |

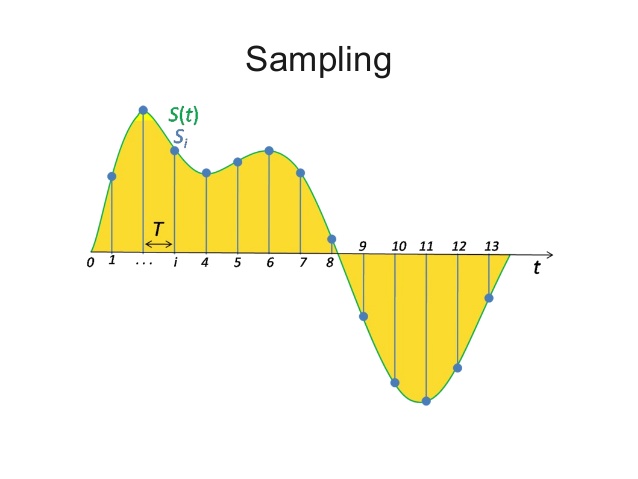
**Objective:**

***An Introduction to Analog to Digital Conversion (Sampling and Aliasing).***

**Post Lab Exercises:**

**Question 1:**

What do you mean by the term “Sampling”? Discuss it briefly with the help of figure.

****

**Answer:**

In digital signal processing, sampling is the reduction of a continuous-time signal to a discrete-time signal. A common example is the conversion of a sound wave (a continuous signal) to a sequence of samples (a discrete-time signal). A sample is a value or set of values at a point in time and/or space.

**Question 2:**

What is Sampling theorem? What do you mean by the term Aliasing?

**Answer:**

A continuous time signal can be represented in its samples and can be recovered back when sampling frequency fs is greater than or equal to the twice the highest frequency component of message signal. i. e.

fs≥2fm.

Aliasing is a phenomenon in which higher frequency signals are mapped into some lower frequency signals. Aliasing can be removed by following Nyquist criteria (Sampling Theorem).The sampling frequency is given by :

Fm – nfs = falias where n is the no. of aliasing

**Question 3:**

Human audible frequency ranges from **20 Hz** to **20 KHz**. Human voice frequency ranges from **4Hz** to **4 KHz**.

**Question 4:**

Record audio for 10sec and complete the following table. Show and verify the output file size through mathematical calculations.

(Hint: Check the Microphone ADC bits and use sampling frequency and the audio record time to evaluate the file size)

|  |  |  |  |
| --- | --- | --- | --- |
| S No | Sampling Frequency(Hz) | File Size  (MB) | Quality  Comment |
| 1 | 44100 | 1.68 | Quality of Sampling is Excellent to hear and there is no Aliasing in this Frequency |
| 2 | 22050 | 0.84 | Quality of Sampling is lesser than above Frequency and there is no Aliasing in this Frequency |
| 3 | 10000 | 0.38 | Quality of Sampling is good and easy to hear voice |
| 4 | 6000 | 0.22 | Quality of Sampling is little good and easy to understand voice |
| 5 | 4000 | 0.152 | Quality of Sampling is little poor |
| 6 | 2000 | 0.076 | Quality of Sampling is quite poor, aliasing is present and difficult to understand voice |
| 7 | 1000 | 0.038 | Quality of Sampling is very poor, aliasing is present and unable to understand voice |

**Answer:**

As we observe in above table that when we decrease the sample rate or frequency, we are difficult or unable to understand the voice. We are requiring lots of focus to understand the voice by decreasing the frequency. When we are unable to understand the voice that’s means that Aliasing is present.

**Calculation of File Size:**

File size can be calculated by the following formula:

𝐵𝑖𝑡𝑠 = 𝑆𝑎𝑚𝑝𝑙𝑖𝑛𝑔 𝐹𝑟𝑒𝑞𝑢𝑒𝑛𝑐𝑦 × 𝑆𝑎𝑚𝑝𝑙𝑒 𝑆𝑖𝑧𝑒 × 𝑇𝑖𝑚𝑒 × 𝐶ℎ𝑎𝑛𝑛𝑒l

|  |  |  |
| --- | --- | --- |
| 𝑭𝒔 = 𝟒𝟒𝟏𝟎𝟎𝑯𝒛  = 44100 × 16 × 10 × 2  = 14112000 𝑏𝑖𝑡𝑠  = 14112000 /8  = 1764000 𝐵  = 1764000/1024  = 1722.656 𝐾𝐵  = 1722.656 /1024  = 1.68 𝑀𝐵 | 𝑭𝒔 = 𝟐𝟐𝟎𝟓𝟎𝑯𝒛  = 22050 × 16 × 10 × 2  = 7056000 𝑏𝑖𝑡𝑠  = 7056000 /8  = 882000 𝐵  = 882000/1024  = 861.328 𝐾𝐵  = 861.328 /1024  = 0.8411 𝑀𝐵 | 𝑭𝒔 = 𝟏𝟎𝟎𝟎𝟎𝑯𝒛  = 10000 × 16 × 10 × 2  = 3200000 𝑏𝑖𝑡𝑠  = 3200000 /8  = 400000 𝐵  = 400000/1024  = 390.625 𝐾𝐵  = 390.625 /1024  = 0.3814 𝑀𝐵 |
| 𝑭𝒔 = 𝟔𝟎𝟎𝟎𝑯𝒛  = 6000 × 16 × 10 × 2  = 1920000 𝑏𝑖𝑡𝑠  = 1920000 /8  = 240000 𝐵  = 240000/1024  = 234.375 𝐾𝐵  = 234.375 /1024  = 0.2288 𝑀𝐵 | 𝑭𝒔 = 𝟒𝟎𝟎𝟎𝑯𝒛  = 4000 × 16 × 10 × 2  = 1280000 𝑏𝑖𝑡𝑠  = 1280000 /8  = 160000 𝐵  = 160000/1024  = 156.25 𝐾𝐵  = 156.25 /1024  = 0.1525𝑀𝐵 | 𝑭𝒔 = 𝟐𝟎𝟎𝟎𝑯𝒛  = 2000 × 16 × 10 × 2  = 640000 𝑏𝑖𝑡𝑠  = 640000 /8  = 80000 𝐵  = 80000/1024  = 78.125 𝐾𝐵  = 78.125 /1024  = 0.0762 𝑀𝐵 |

|  |
| --- |
| 𝑭𝒔 = 𝟏𝟎𝟎𝟎𝑯𝒛 |
| = 1000 × 16 × 10 × 2 |
| = 320000 𝑏𝑖𝑡𝑠 |
| = 320000 /8 |
| = 40000 𝐵 |
| = 40000/1024 |
| = 39.0625 𝐾𝐵 |
| = 39.0625/1024 |
| = 0.03814 𝑀𝐵 |

**Question 5:**

If an ADC has sampling frequency =1000 Hz and receive analog signals of the following frequencies what will be the frequency of a signal which is converted back to analog by a DAC converter?

|  |  |  |  |
| --- | --- | --- | --- |
| **S No** |  | **Frequency (Hz)** | **Frequency of output Signals (Hz)** |
|  | 1 | 100 | 100 |
|  | 2 | 750 | -250 |
|  | 3 | 1250 | 250 |
|  | 4 | 1900 | -100 |
|  | 5 | 2000 | 0 |
|  | 6 | 2500 | 500 |

**Answer:**

## Since 𝑭𝒔 ≥ 𝟐𝒇 OR 𝑭𝒔/𝟐 ≥ 𝒇

Frequencies less than equal to 500hz will remain unaliased while the rest of the signals which are greater than 500hz (750hz,1250hz,1900hz,2000hz,2500hz) will be aliased according Nyquist theorem.

𝑭𝒂𝒍𝒊𝒂𝒔 = 𝑭𝒎𝒂𝒙 − 𝒏𝑭𝒔𝒂𝒎𝒑𝒍𝒊𝒏𝒈

**Calculations: -**

1. No Aliasing in case of 100 Hz signal because 100 Hz < 500 Hz which is the maximum allowable frequency.
2. Falias = 750 − 1000 = −250 Hz
3. Falias = 1250 − 1000 = 250 Hz
4. Falias = 1900 − (2 ∗ 1000) = −100 Hz
5. Falias = 2000 − (2 ∗ 1000) = 0 Hz
6. Falias = 2500 − (2 ∗ 1000) = 500 Hz

|  |
| --- |
| **Laboratory Session No. 03** |

**Objective:**

***An Introduction to Analog to Digital Conversion (Quantization and Coding).***

**Matlab Code for the Analysis of Quantization:**

%% lab3

fprintf('\Lab #03, Quantization');

b=3; % Number of bits

N=120 ; % Number of samples in final signal

n=0:(N-1); %Index

%choose the input type

choice=questdlg('Choose input','Input','Sawtooth','Sine','Random','Random' );

fprintf('Bits = %g , levels = %g , signal= %s.\n',b,2^b,choice);

%Create the input data sequence

switch choice

case 'Sine'

x=sin(2\*pi\*n/N);

case 'Sawtooth'

x=sawtooth(2\*pi\*n/N);

case 'Random'

x=randn(1,N); %Random data

x=x/max(abs(x)); %scale to +/- 1

end

%signal is restricted to between -1 and +1

x(x>=+1)=(1-eps); %make signal from -1 to just less than 1

x(x<-1)=-1;

%Quantize a signal to "b" bits

xq=floor((x+1)\*2^(b-1)); %signal is one of 2^n int values (0 to 2^n-1)

xq=xq/(2^(b-1)); %signal is from 0 to 2 (quantized)

xq=xq-(2^(b)-1)/2^(b); %shift signal down (rounding)

xe=x-xq; %quantization error

stem(x,'b');

hold on;

stem(xq,'r');

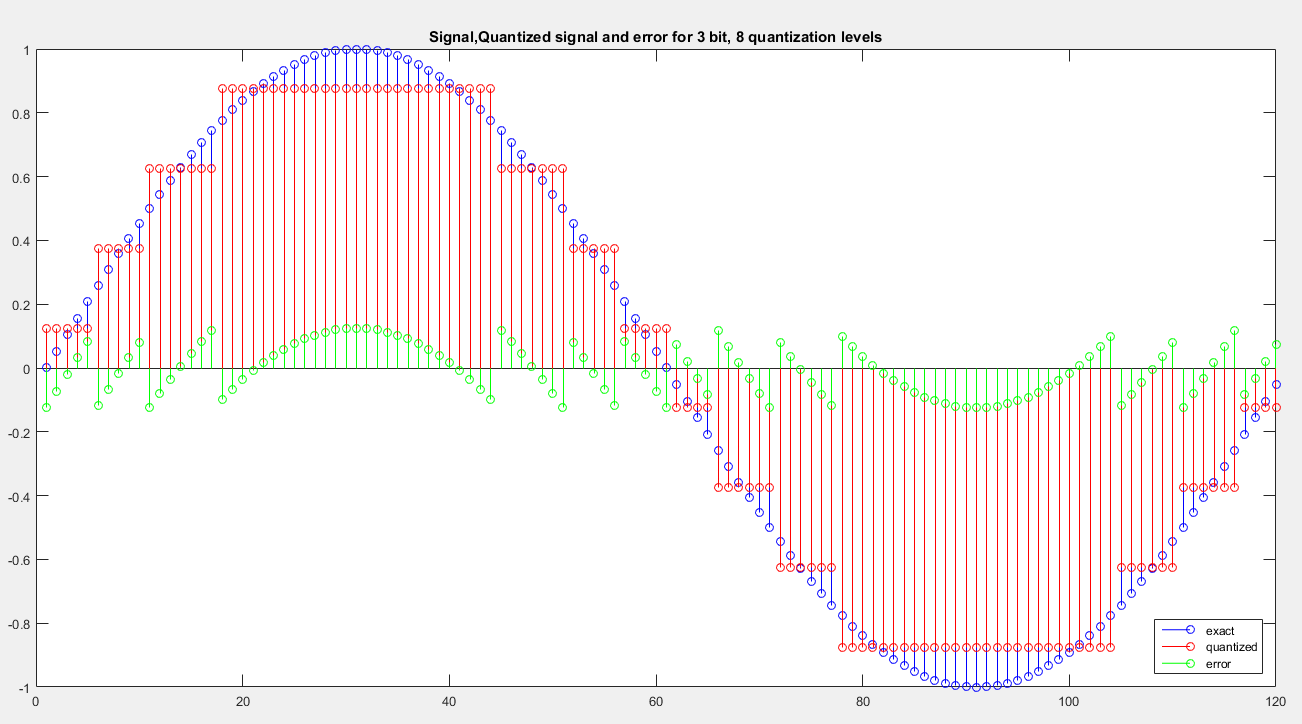
hold on

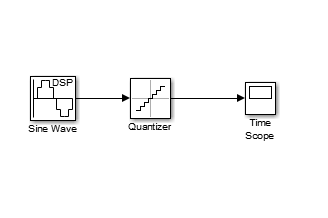
stem(xe,'g');

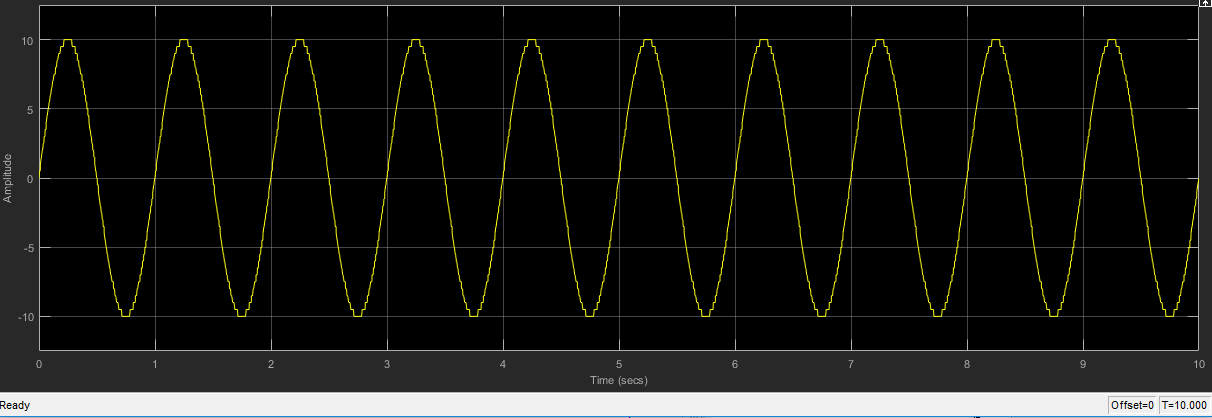
legend('exact','quantized','error','Location','Southeast')

title(sprintf('Signal,Quantized signal and error for %g bit, %g quantization levels',b,2^b));

hold off

**Output**

**Analysis of Quantization with DSP System toolbox in Simulink**



**Post Lab Exercises:**

**Question 1:**

Why Quantization is needed in Digital Signal Processing?

**Answer:**

Quantization, in mathematics and [digital signal processing](https://en.wikipedia.org/wiki/Digital_signal_processing), is the process of mapping input values from a large set (often a continuous set) to output values in a (countable) smaller set, often with a finite [number of elements](https://en.wikipedia.org/wiki/Cardinality).

The process of converting a discrete time continuous amplitude signal into a digital signal by expressing each sample value as a finite (instead of an infinite) number of digits is called quantization, The error induced in representing the continuous valued signal by a finite set of discrete value levels is called quantization error or quantization noise.

**Question 2:**

What is Anti-Aliasing Filter? Discuss it with some example.

**Answer:**

If we do not follow Nyquist criteria, our signal contains some aliased frequencies. These aliasing components are occurred due to the passing of unnecessary high frequency signal from a sampler. In order to avoid those high frequency signals we placed an anti-aliasing filter before the sampler which blocks the high frequency components.

For Example: This anti-aliasing filter is generally a low-pass filter which allows low frequency components and block the higher one.

**Question 3:**

Discuss the specifications of Arduino Uno ADC like number of bits, number of quantization levels, etc. Also, calculate the default Arduino Uno ADC resolution.

Link: <https://store.arduino.cc/usa/arduino-uno-rev3>

**Answer:**

The **Arduino Uno ADC** is of 10 bit **resolution** (so the integer values from (0-(2^10) 1023)) which means total quantization levels are 1024. This means that it will map input voltages between 0 and 5 volts into integer values between 0 and 1023. So for every (5/1023= 4.9mV) per unit.

**Question 4:**

A 12-bit ADC has input values in the range of 0 – 1 V. Calculate the resolution of ADC.

**Answer:**

Since No of levels (L) = 212 = 4096 levels

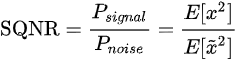
So resolution = 1-0/4095 = 0.24 mV

**Question 5:**

What do you mean by the term SQNR and ENOB? Discuss it briefly.

**Answer**

Signal-to-Quantization-Noise Ratio (SQNR or SNqR) is widely used quality measure in analysing [digitizing](https://en.wikipedia.org/wiki/Digitizing) schemes such as PCM ([pulse code modulation](https://en.wikipedia.org/wiki/Pulse-code_modulation)) and [multimedia codecs](https://en.wikipedia.org/wiki/Multimedia_codec). The SQNR reflects the relationship between the maximum nominal [signal strength](https://en.wikipedia.org/wiki/Signal_strength) and the [quantization error](https://en.wikipedia.org/wiki/Quantization_error) (also known as quantization noise) introduced in the [analog-to-digital conversion](https://en.wikipedia.org/wiki/Analog-to-digital_conversion). As SQNR, like SNR, is a ratio of signal power to some noise power, it can be calculated as:

{\displaystyle \mathrm {SQNR} ={\frac {P\_{signal}}{P\_{noise}}}={\frac {E[x^{2}]}{E[{\tilde {x}}^{2}]}}}

Effective number of bits (ENOB) is a measure of the [dynamic range](https://en.wikipedia.org/wiki/Dynamic_range) of an [analog-to-digital converter](https://en.wikipedia.org/wiki/Analog-to-digital_converter) (ADC), [digital-to-analog converter](https://en.wikipedia.org/wiki/Digital-to-analog_converter), or their associated circuitry. The resolution of an ADC is specified by the number of [bits](https://en.wikipedia.org/wiki/Bit) used to represent the analog value. Ideally, a 12-bit ADC will have an effective number of bits of almost 12. However, real signals have noise, and real circuits are imperfect and introduce additional [noise](https://en.wikipedia.org/wiki/Noise_(electronics)) and [distortion](https://en.wikipedia.org/wiki/Distortion). Those imperfections reduce the number of bits of accuracy in the ADC. The ENOB describes the effective resolution of the system in bits. An ADC may have 12-bit resolution, but the effective number of bits when used in a system may be 9.5.

|  |
| --- |
| **Laboratory Session No. 04** |

**Objective:**

***An Introduction to Time shifting, Reversal and scaling.***

**Post Lab Exercises:**

**Question 1:**

Let x(n)=[**1** 1 2 3 -1 -1 2 5 6 ]

Using Matlab plot the following signals.

1. x(n-3)
2. x(n+1)
3. x(-n)
4. x(2n)
5. x(n/3)
6. x(-n+1)
7. x(-n-1)

also show every step and mention operations i.e. time shifting, time scaling etc.

**Answer:**

%% lab3 shifting,flipping,scaling

x=[1 1 2 3 -1 -1 2 5 6 ];

l=length(x);

n=0:l-1;

subplot(4,2,1);

stem(n,x); %original signal

title('x(n)')

subplot(4,2,2);

stem(n+3,x); %shifting

title('x(n-3)')

subplot(4,2,3);

stem(n-1,x); %shifting

title('x(n+1)')

subplot(4,2,4);

xf1=flip(x); %flipping

stem(-flip(n),xf1);

title('x(-n)')

subplot(4,2,5);

n1=(0/2):(1/2):(l-1)/2; %scaling

stem(n1,x);

title('x(2n)')

subplot(4,2,6);

n2=(0\*3):1\*3:(l-1)\*3; %scaling

stem(n2,x);

title('x(n/3)')

subplot(4,2,7);

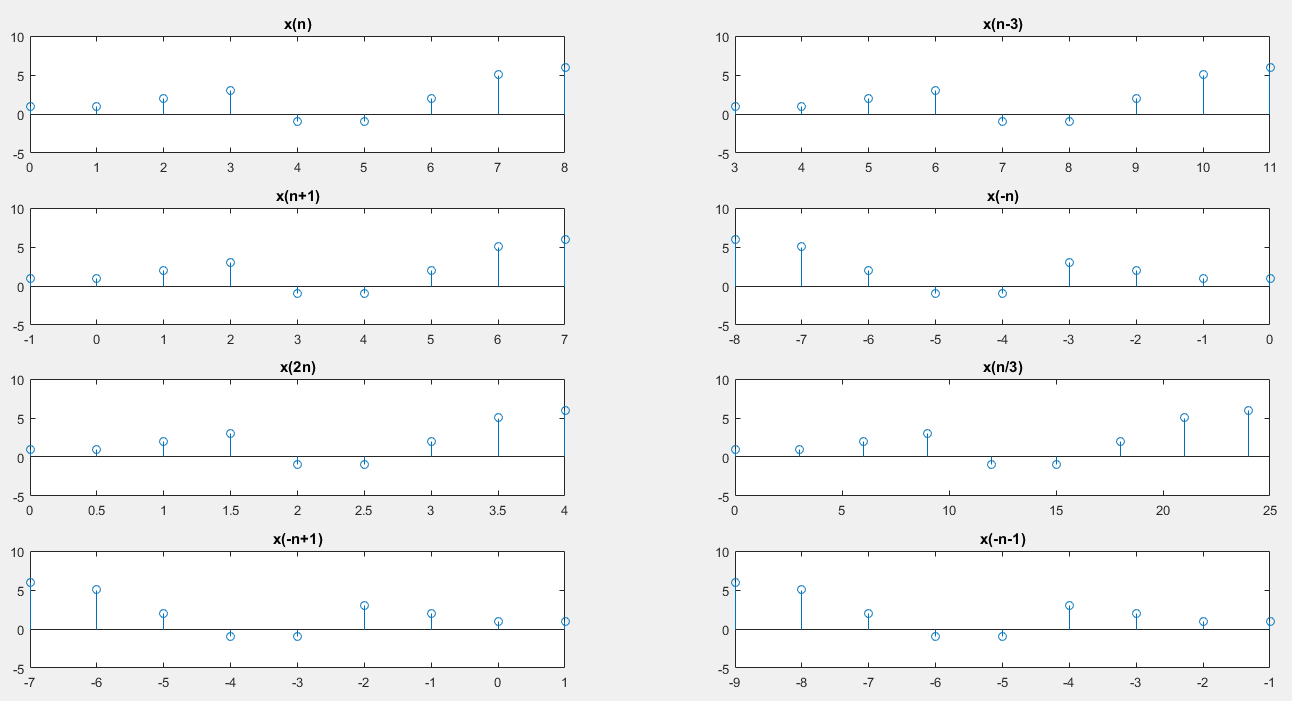
nf=-flip(n); %x(-n)

stem(nf+1,xf1); %x(-n+1)

title('x(-n+1)')

subplot(4,2,8);

stem(nf-1,xf1); %x(-n-1)

title('x(-n-1)');

**Question 2:**

let x(n)= [-2 2 1 -1 **3** 2 +2 -3 -1 5 0 1]

using Matlab plot the following signals.

1. x(n+2)
2. x(-n+2)
3. x(2n)
4. x(2n+3)

also show every step and mention operations i.e. time shifting, time scaling etc.

(Attach codes at the end of this lab)

**Answer:**

%% lab3 task2

x=[-2 2 1 -1 3 2 +2 -3 -1 5 0 1];

l=length(x);

n=-4:7;

subplot(2,3,1);

stem(n,x); grid on % original signal x(n)

title('x(n)')

subplot(2,3,2);

stem(n-2,x); grid on %x(n+2)

title('x(n+2)')

subplot(2,3,3);

xf=flip(x);

nf=-flip(n);

stem(nf,xf); grid on %x(-n)

title('x(-n)')

subplot(2,3,4);

stem(nf+2,xf); grid on %x(-n+2)

title('x(-n+2)')

subplot(2,3,5);

ns=n/2; %scaling by factor of 2

stem(ns,x); grid on %x(2n)

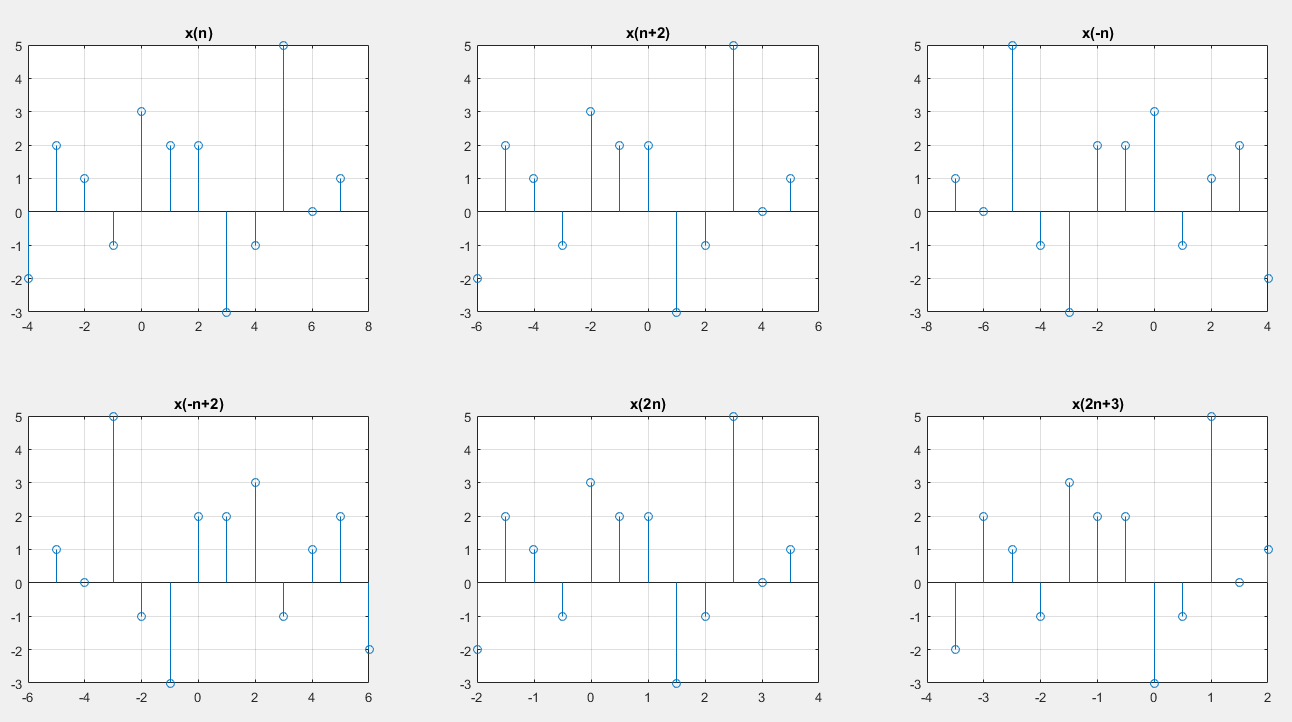
title('x(2n)')

subplot(2,3,6);

ns1=(n-3)/2;

stem(ns1,x); grid on %x(2n+3)

title('x(2n+3)')



|  |
| --- |
| **Laboratory Session No. 05**  **(Open Ended Lab)** |

**Objective:**

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

**Required Components:**

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

**Procedure:**

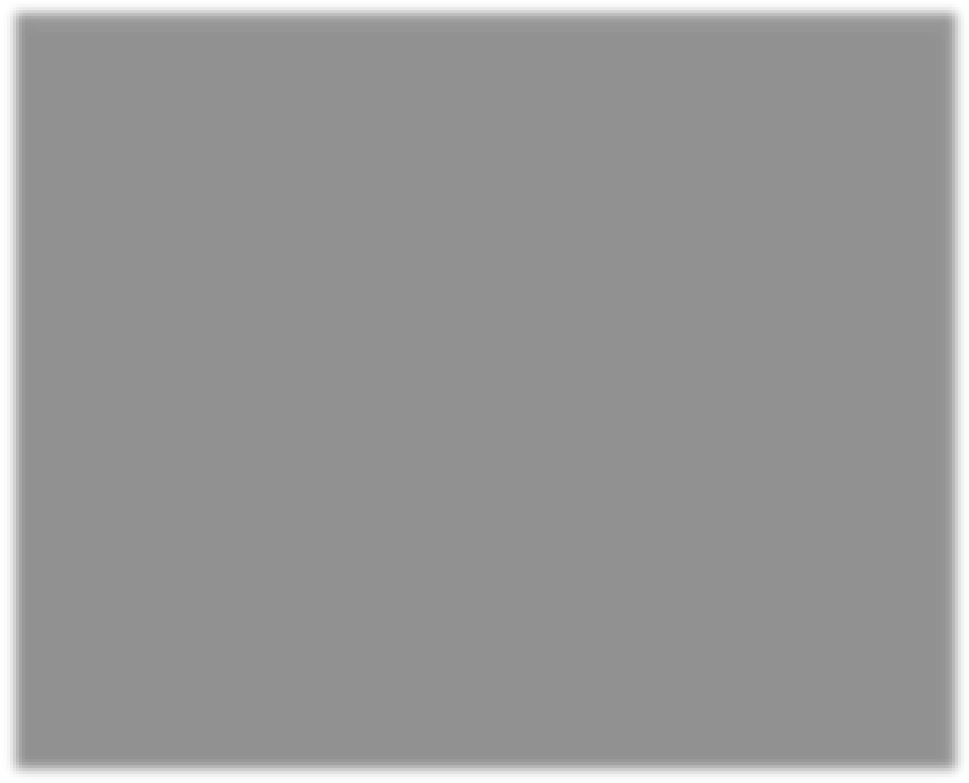
* Using Transformer convert 220VAC from mains into 12VAC.
* Using VDR convert 12VAC to a voltage compatible to audio card (show all the calculations of resistances with their power ratings).
* Set the sampling frequency of the audio card ADC in MATLAB Simulink environment with proper justification
*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

**Working**

* The basic concept of this lab was to observe and simulate the AC voltage on Matlab.
* For this purpose an external sound card is used.
* The voltage rating of the soundcard is 1V and its current rating is 1mA
* Voltage Measurements have been taken across the load Resistor. In order to do so, secondary of PT is connected to series potentiometer and load. Potentiometer is so adjusted that the voltage across the load will be in 1V range.
* One end of the aux cable is connected in parallel to load resistor and its other end is connected to the microphone input of the external sound card.
* For measuring current CT is placed in series with the load and its secondary is connected to the microphone
* The soundcard come with a built in ADC which converts the analog wave form to discrete wave form using a sampling frequency of 44.1 kHz.
* A sinusoid was observed but with amplitude different than the input applied from AC source.
* The gain block is then added on Simulink and the output is observed with parameters of wave almost similar to that observed practically in oscilloscope.
* A gain factor is applied to scale the wave form to the actual AC (i.e. 220 Vrms)

# SPECIFICATION OF COMPONENTS USED

**DIGITAL AUDIO SOUND CARD:**



* **Model number:** U237-001
* **Operating Temperature Range:** 32 to 104 F( 0 to 40 C)
* **Storage Temperature Range:** 14 to 131 F (0 to 40C)
* **Rated voltage:** 1V
* **Rated Current:** 1mA
* **Power Consumption (Watts) :** 0.5W
* **Ports :** 2
* **Side A- Connector 1:**USB A(MALE)
* **Side B- Connector 2:** (2)3.5mm (FEMALE)
* **Connector Plating:** Nickel

**CURRENT TRANSFORMER:**



**Model:** ZME1915007BF

**Application Range:** Measuring

**Kind :** Induction Cooker transformer current transformer

**Package Form:**  Plugin

**Winding form:** Multi-layer flat winding

**Magnetic properties:** Iron Core

**Core Shape:** E-shape

**Working Frequency:** Low frequency

**Instalation method:** Horizontal unsealed

**Quality Factor:**  98

**Allowable error:** +-20%

**Inductive XL:** 10 (Ω)

**Rated Current:** 5000 (mA)/5Amps

**Distributed capacitance** 10 (F)

**Input Voltage:**  220V (50/60Hz) (V)

**CALCULATIONS**

**V.D.R Used :**

**Series resitance = 65k ohms ; Load resitance = 1K ohms**

**No of bits** = 16 (**bits of sound card set in P.C )**

**Sampling Frequency** = 𝐹𝑠 = 44,100𝐻𝑧/44.1𝐾 𝐻𝑧

𝐍𝐎 𝐎𝐅 𝑳𝒆𝒗𝒆𝒍 = 𝑙 = 2# 𝑜𝑓 𝑏𝑖𝑡𝑠 = 216 = 65,536 𝑙𝑒𝑣𝑒𝑙𝑠 (𝑸𝒖𝒂𝒏𝒕𝒊𝒛𝒂𝒕𝒊𝒐𝒏 𝒍𝒆𝒗𝒆𝒍)

Total memory consumed: 𝑆𝑎𝑚𝑝𝑙𝑖𝑛𝑔 𝐹𝑟𝑒𝑞𝑢𝑒𝑛𝑐𝑦 × 𝑆𝑎𝑚𝑝𝑙𝑒 𝑆𝑖𝑧𝑒 × 𝑇𝑖𝑚𝑒×𝐶ℎ𝑎𝑛𝑛𝑒l

44100 × 16 × 10 × 2

= 14112000 𝑏𝑖𝑡𝑠

= 14112000 /8

= 1764000 𝐵

= 1764000/1024

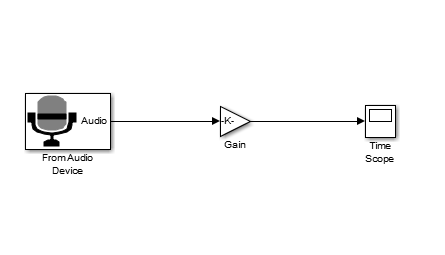
= 1722.656 𝐾𝐵

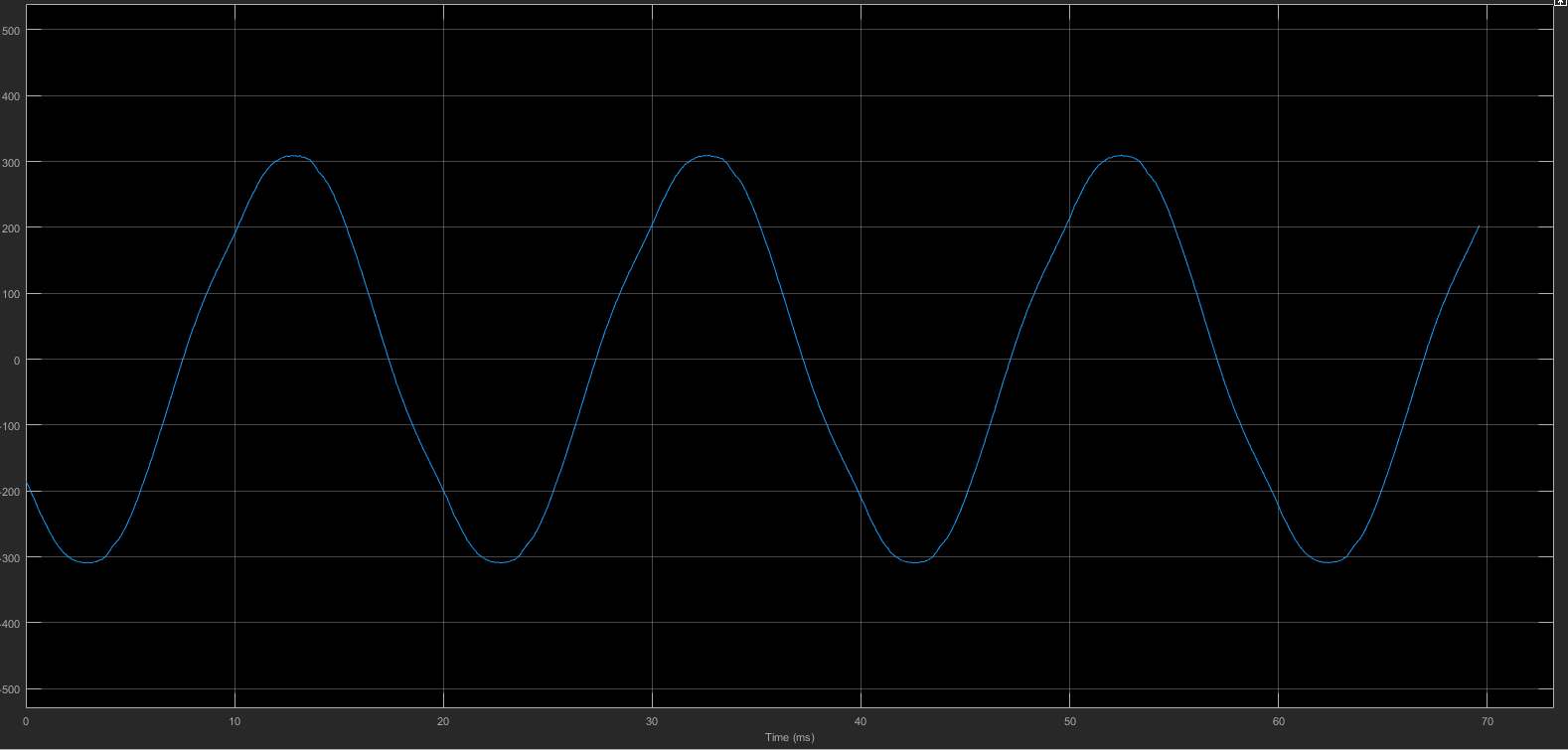
= 1722.656 /1024

= 1.68 𝑀𝐵

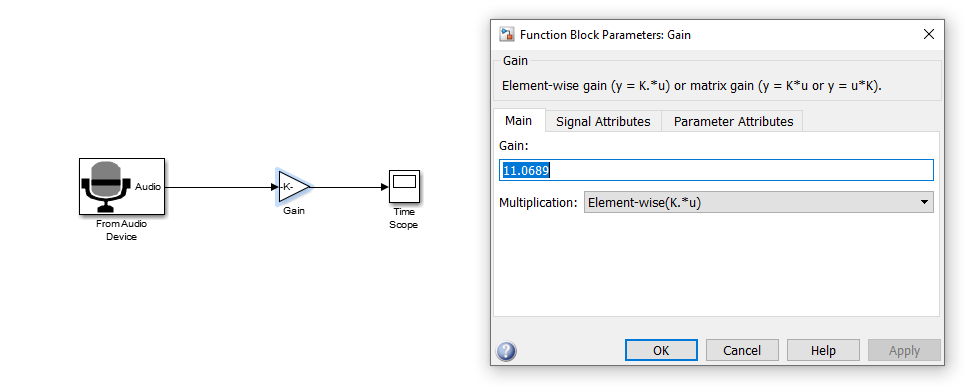
**MATLAB SIMULATION:**

**Voltage waveform**

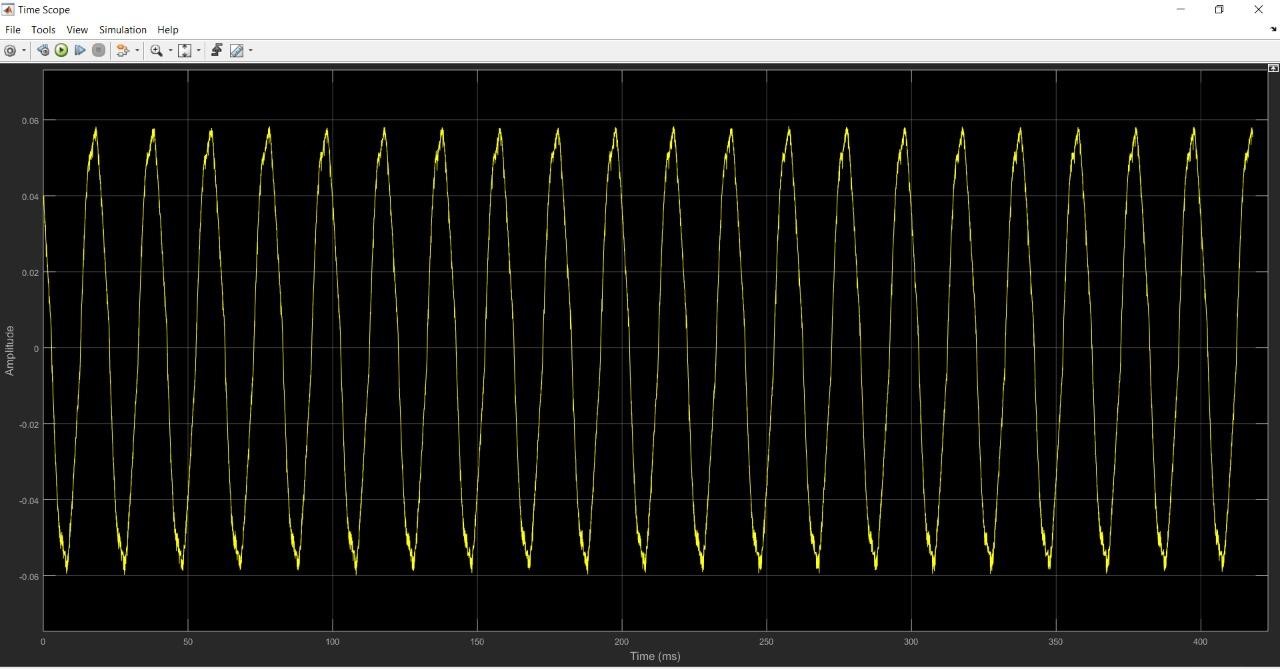
****

****

The input sinusoidal voltage signal having PK-PK value of 311V was simulated and digitized through an ADC (sound card) and observed on Matlab.

**Current Waveform**

**Conclusion:**



This lab demonstrate the conversion of Analog signal into Digital signal in the form of voltage and current, in simple words, we are converting Analog Electrical quantities into digital Electrical quantities because if we have to perform any processing techniques or want to do any mathematical calculations, we can do it in the form of software (digitized way) more simply than in the analog form by making our own hardware circuit which is quite expensive also as well as we can do data acquisition also which cannot be achieved in Analog domain

|  |
| --- |
| **Laboratory Session No. 06** |

**Objective:**

***To generate a square wave in time domain of specific time period and pulse width and to apply CTFS equation to compute the spectral coefficients using MATLAB.***

**Pre Lab Exercises:**

**Types of signals:**

• Continuous-Time Periodic Signals

• Discrete-Time Periodic Signals

• Continuous-Time aperiodic Signals

• Discrete-Time aperiodic Signals

**Mathematical tool for the analysis of Signals and Systems:**

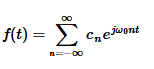
• Fourier series: *preferred for Signal Analysis*

• Laplace transform: *preferred for System Analysis*

• Z transform: *preferred for converting discrete time domain signal into discrete frequency domain signal.*

**Continuous Time Fourier series**

In this lab our objective is to study continous time fourier series. The continuous-time Fourier series expresses a periodic signal as a linear combination of harmonically related complex exponentials. Alternatively, it can be expressed in the form of a linear combination of sines and cosines or sinusoids of different phase angles.The synthesis equation fourier can be written as



Where as the analysis equation is given as



**In Lab Exercises:**

Continuous time Fourier series Analysis of a Sinusoidal signal

Ts=0.001;

Tp=2;

t=[0:Ts:Tp-Ts];

x=2+sin(2\*pi\*1/Tp\*t)+sin(2\*pi\*5/Tp\*t);

subplot(2,1,1), plot(t,x);

xlabel('time(EE-181)');

Fo=1/Tp;

for k=1:20

B=exp(-j\*2\*pi\*(k-1)\*Fo.\*t);

C(k)=sum(x.\*B)/(length(x));

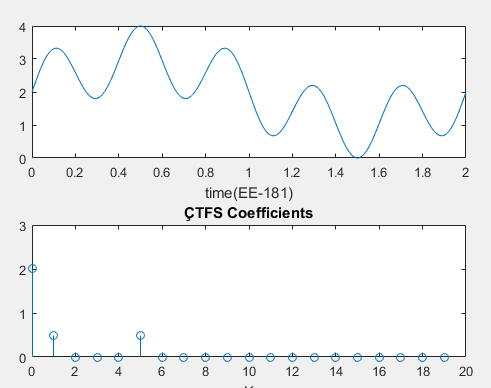
end

k=[0:k-1];

subplot(2,1,2), stem(k,abs(C));

title('ÇTFS Coefficients');

xlabel('K');

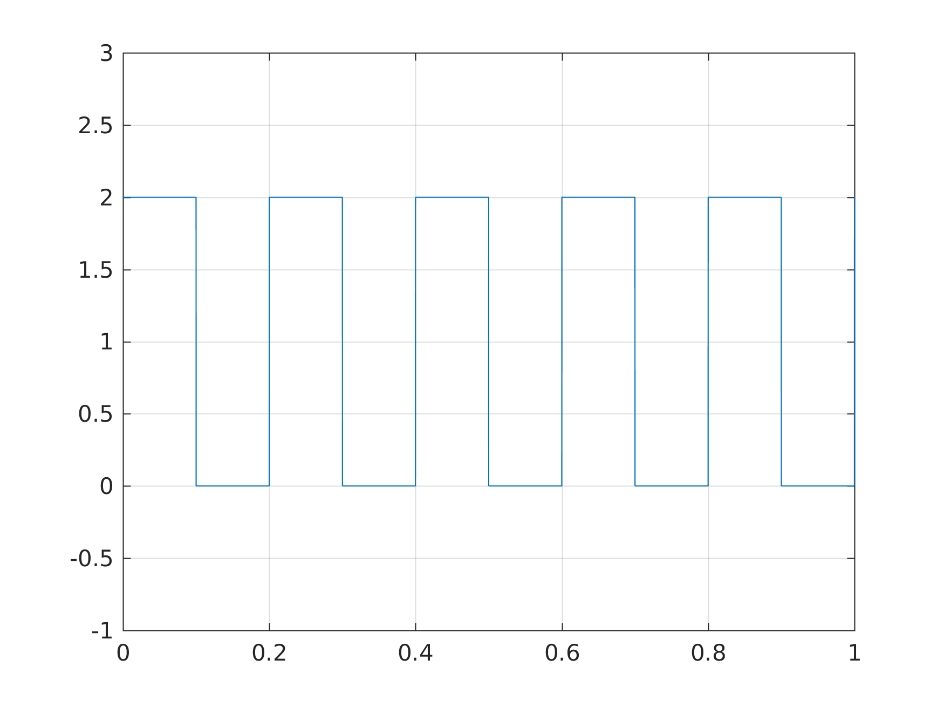


**Post Lab Exercises:**

**Question 1:**

Generate a square wave of magnitude 2 units with duty cycle of 50 % and have a frequency of 5 Hz using MATLAB *square()* function and then plot the frequency spectrum (magnitude vs frequency plot) of the resulting wave using MATLAB script (Evaluate five spectral coefficients). Also, verify your MATLAB results using hand calculations.

Note: Use MATLAB help to get the syntax of the *square()* function.



Ts=0.001;

Tp=1/5;

t=[0:Ts:Tp-Ts];

x=2\*square(2\*pi\*1/Tp\*t);

subplot(2,1,1), plot(t,x) ;

xlabel('time(EE-181)');

Fo=1/Tp;

for k=1:20

B=exp(-j\*2\*pi\*(k-1)\*Fo.\*t);

C(k)=sum(x.\*B)/(length(x));

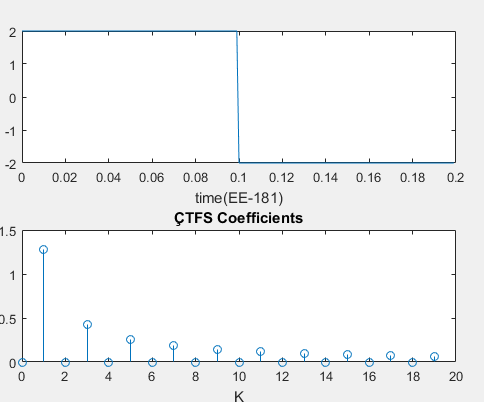
end

k=[0:k-1];

subplot(2,1,2), stem(k,abs(C));

title('ÇTFS Coefficients');

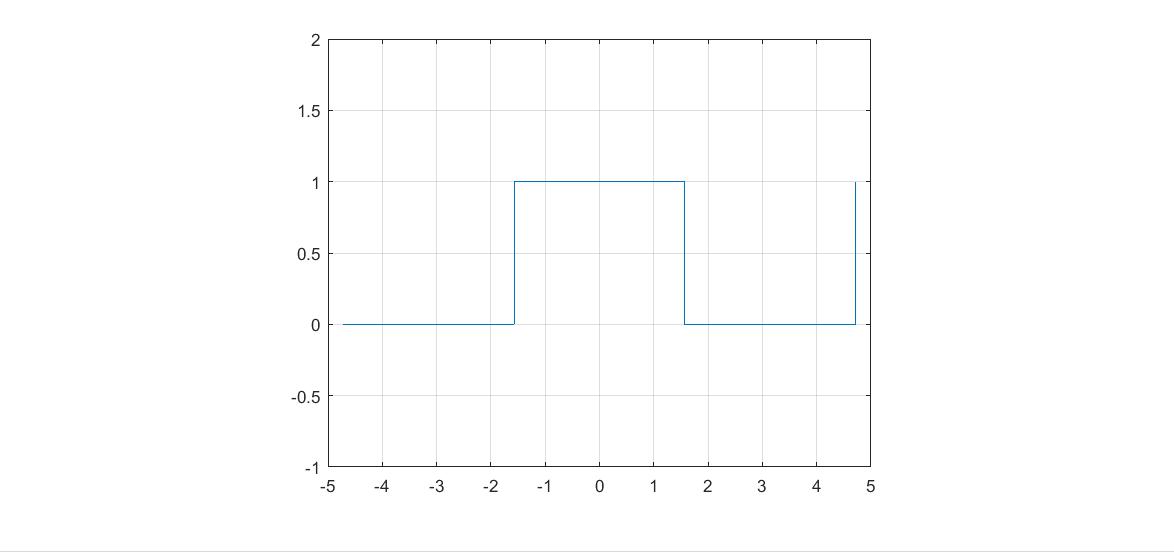
xlabel('K');



**Question 2:**

Verify Fourier series by adding harmonically related cosines and make square wave

Consider the given signal



This signal has time period of To=2 sec

And fundamental frequency is 2 rad/sec

After analyzing the above signal for one period , fourier analysis yields following result



Let's add individual component for one-period of time and see whether they are reproducing our actual time domain plot or not !

t=0:0.01:10;

X\_0=1/2;

X\_X=X\_0;

start=1;

for n=2:1000

if rem(n,2)==0

sign=1;

else

sign=-1;

end

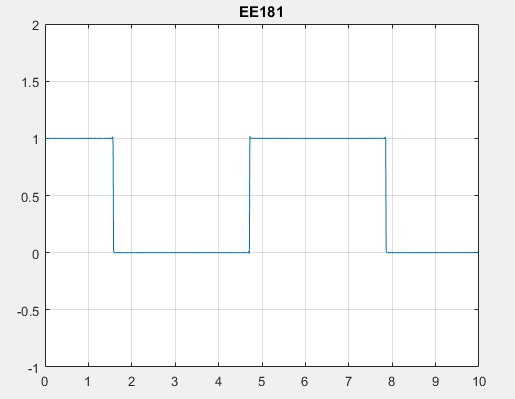
X\_X=X\_X+(2/pi)\*sign\*(1/start)\*cos(start\*t);

start=start+2;

plot(t,X\_X) ,ylim([-1,2]) , grid on;

title('EE181')

end



**Difference between Single Sided and Double Sided Spectrum**

Most frequency analysis instruments display only the positive half of the frequency spectrum because the spectrum of a real-world signal is symmetrical around DC. Thus, the negative frequency information is redundant. The two-sided results from the analysis functions include the positive half of the spectrum followed by the negative half of the spectrum.

A two-sided power spectrum displays half the energy at the positive frequency and half the energy at the negative frequency. Therefore, to convert a two-sided spectrum to a single-sided spectrum, you discard the second half of the array and multiply every point except for DC by two

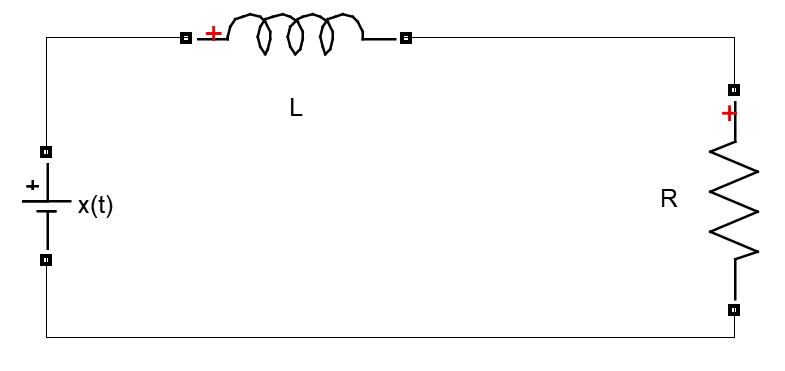
|  |
| --- |
| **Laboratory Session No. 06** |

**Objective:**

***To model a source free and sourced RL circuit in a discrete time domain and to verify it using MATLAB code and Simulink environment.***

**Theory:**

Consider a series RL circuit excited by a voltage source as shown below,



Applying KVL we get,

(1)

For a source a source free RL circuit the above expression will become,

(2)

By using first principle, the derivate of inductor current in a discrete time domain can be evaluated as,

(3)

or,

(4)

From eq. (2), the derivative at the instant can be evaluated as,

(5)

Equating eq. (4) and (5) we get,

(6)

By choosing sampling *T* close to zero we can evaluate the difference equation as,

(7)

Where, and are replaced by and respectively.

**Post Lab Exercise:**

**Question 1:**

For a series RL circuit excited by a voltage source , derive an expression for inductor current , in discrete time domain. Using the derived expression, develop a MATLAB code to plot the inductor current under following system parameters:

**Answer:**

|  |
| --- |
| **Laboratory Session No. 06** |

**Objective:**

***Convolution for Discrete Time Signals***

**Post Lab Exercises:**

**Question 1:**

We have an impulse response of a system h(n)= {**3**,2,1, 2,1,0, 4,0,3}, provided with an input

*x(n)*={**1** 2 3 5 4 6}, find and plot *y(n)*. Also check and comment on the causality property of the system.

**Answer:**

**Question 2:**

We have an impulse response of a system h(n)= {-3,1,2, 4,**1**,0, 4,0,3}, provided with an input

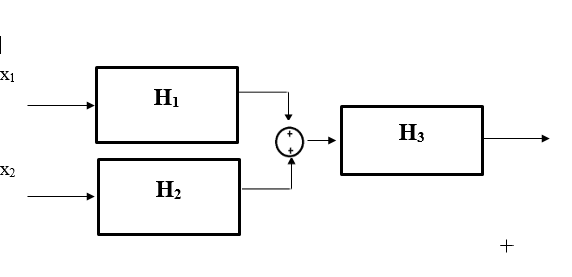
x(n)={**1** 2 3 5 4 6}, find and plot y(n). Also check and comment on the causality property of the system.

**Answer:**

**Question 3:**

If impulse response of the system h1(n)= {**1** 1 1}, h2(n)= {**0**,0, 1} and h3(n)= {1,**2**,0, 1} , Input

*x1(n)*={**1** 2 3 5 4 6} and *x2(n)*={**1** 2 3 5 4 6}, find and plot *y(n)*. Also check and comment on the causality property of the system.



**Answer:**

|  |
| --- |
| **Laboratory Session No. 07** |

**Objective:**

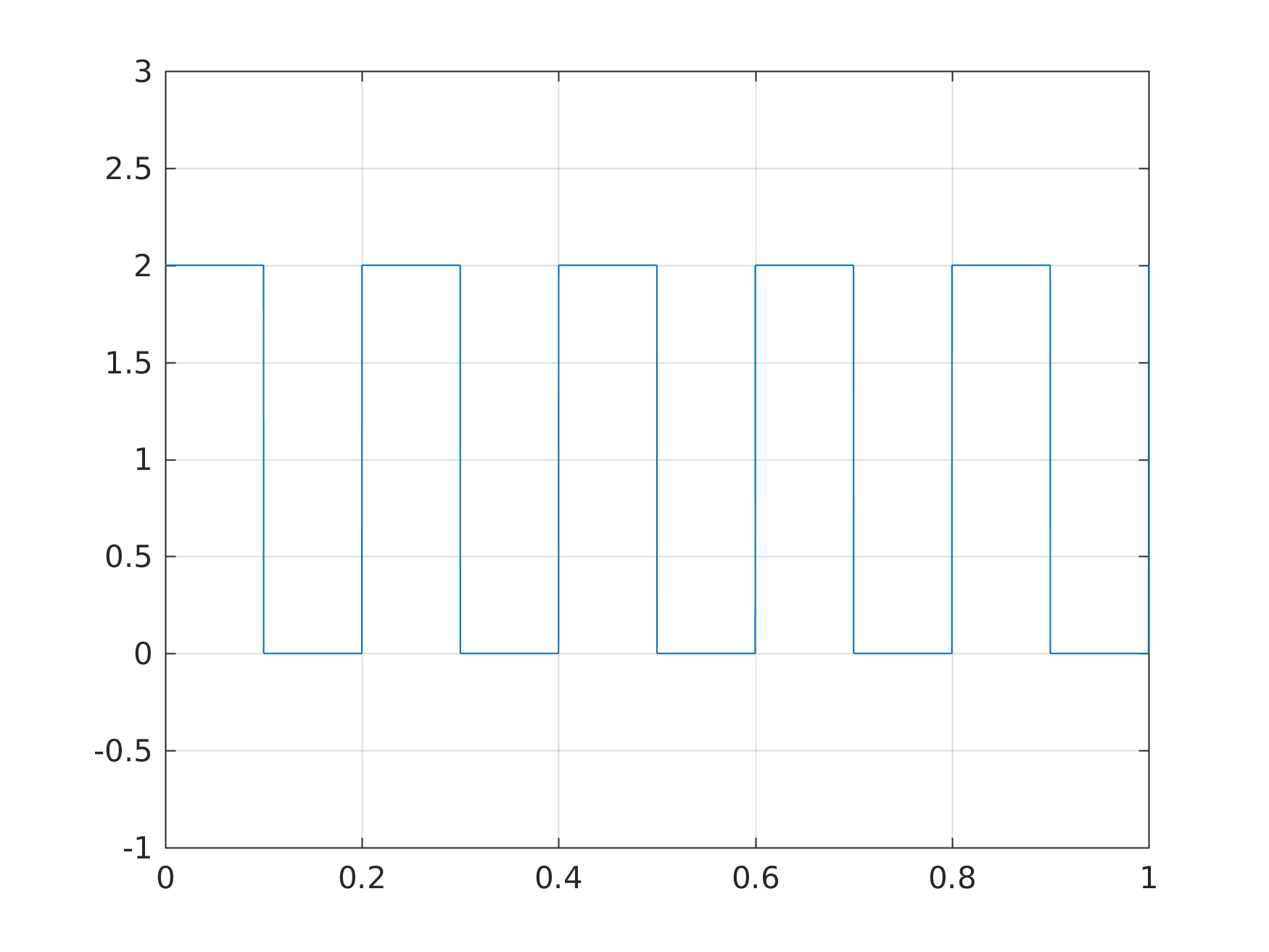
***To generate a square wave in time domain of specific time period and pulse width and to apply CTFS equation to compute the spectral coefficients using MATLAB.***

**Post Lab Exercises:**

**Question 1:**

Generate a square wave of magnitude 2 units with duty cycle of 50 % and have a frequency of 5 Hz using MATLAB *square()* function and then plot the frequency spectrum (magnitude vs frequency plot) of the resulting wave using MATLAB script (Evaluate five spectral coefficients). Also, verify your MATLAB results using hand calculations.

Note: Use MATLAB help to get the syntax of the *square()* function.



**Answer:**

|  |
| --- |
| **Laboratory Session No. 08** |

**Objective:**

***Analysis and Synthesis of Signals through Discrete Fourier Transform (DFT)***

**Post Lab Exercises:**

**Question 1:**

Discuss significance of “N” in N point DFT.

**Answer:**

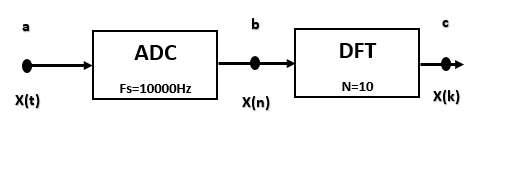
**Question 2:**

DFT can deal both \_\_\_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_\_\_ signals.

**Question 3:**

Consider the following figure, if this system receives input then, plot outputs at point *“b”* and *“c”*.

(Note: Both magnitude and phase plot is required at point *“c”*)



**Answer:**

**Question 4:**

Discuss at least three applications of FFT in Electrical power system Engineering.

**Answer:**

|  |
| --- |
| **Laboratory Session No. 09** |

**Objective:**

***To become familiar with practical constraints in calculating Fourier Transform of any real world signal****.*

**Post Lab Exercises:**

**Question 1:**

Briefly discuss the concept of spectral leakage.

**Answer:**

**Question 2:**

Discuss the solution of spectral leakage.

**Answer:**

**Question 3:**

Discuss the importance of window size (value of N) on frequency spectrum.

**Answer:**

**Question 4:**

Run the following MATLAB script, observe and discuss the resulting plots.

|  |
| --- |
| clear; close;  fs=4000;  N=400;  n=0:N-1;  y=10\*cos(2\*pi\*1500\*n/fs);  f=abs(fft(y));  k=0:N-1;  df=fs/N;  kf=k.\*df;  subplot(411);  stem(kf,f);  xlabel('Frequency Hz');  title('Magnitude Spectrum Without Spectral Leakage');  % If frequency bins are not available (Spectral Leakage) %  N=50;  n=0:N-1;  nfft=1024;  y=100\*cos(2\*pi\*1500\*n/fs);  f=abs(fft(y,nfft));  k=0:nfft-1;  df=fs/nfft;  kf=k.\*df;  subplot(412);  plot(kf,f);  xlabel('Frequency Hz');  title('Magnitude Spectrum With spectral leakage Leakage');    % Solution of Spectral leakage (Windowing)%  win=window(@hamming,N);  subplot(413);  stem(n,win);  title('Window');  y\_win=y.\*win';  f\_win=abs(fft(y\_win,nfft));  k=0:nfft-1;  df=fs/nfft;  kf=k.\*df;  subplot(414);  plot(kf,f\_win);  xlabel('Frequency Hz');  title('Magnitude Spectrum With Windowing'); |

**Answer:**

|  |
| --- |
| **Laboratory Session No. 10**  **(Open Ended Lab 02)** |

**Objective:**

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

**Required Components:**

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

**Procedure:**

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

**Calculations:**

**Results:**

|  |
| --- |
| **Laboratory Session No. 11** |

**Objective:**

***To understand the concept of window overlapping.***

**Post Lab Exercises:**

**Question 1:**

Run the following MATLAB script, attach and discuss the resulting plot.

|  |
| --- |
| clear all;  clc;  [y fs] = audioread('wavTones.com.unregistred.warble\_1000-2000Hz\_-6dBFS\_3s.wav');  n=0:5000;  win\_size = 40e-3\*fs;  frame\_rate = 20e-3\*fs;  Y = buffer(y,win\_size,frame\_rate);  [m n] = size(Y);  win = window(@hamming,win\_size);  YYF = [];  for i = 1:n  YY = Y(:,i).\*win;  YF = abs(fft(YY,fs));  YYF = [YYF YF];  end  figure;  for i = 1:n  clf;  plot(YYF(:,i));  drawnow;  end |

**Answer:**