ETISALAT College Of Engineering

Department of Communication

3rd Year Project Report

Fourier Synthesizer
For Time Periodic Signals

Names of Student Members:

Sultan Rashid # 1158
Obaid Alshamsi # 1242
Humaid Alshamsi # 1223

Supervisor :Dr. R.Shubair

I.ABSTRACT

In this report we describe the Fourier synthesizer for periodic signal, and giving the derivation of the expression resulted from the function and the shape of the signal.

The report starts by introducing the definition of Fourier synthesizer and the relationship between it and the Fourier series.

Then the code of the function was implemented.

After that we introduce the graphical user interface (GUI), which shows the effect of harmonics.

The whole project has been simulated by MATLAB.

II. Acknowledgment

We would like to thank all the people who help us doing the project without those people the project want be done.

I can never give enough thanks to my project supervisor Dr. R. Shubair. His assistance, encouragement, and talent made this project possible. Moreover, he has adjusted my view of scientific projects.

In addition, We have to thank Dr.Shehab Jmaa , for his advise in preparing the presentations and writing the reports .

Special thanks to Mr.Hashim because he stands with us in many places where we need a support.

i.

Contents

Abstract

Acknowledgment

Chapter 1: Introduction

1. Introduction

    1.1. Definition of Fourier synthesizer
    1.2. Importance of Fourier synthesizer in the project
    1.3. Relationship between Fourier series & Fourier synthesizer
    1.4. Usage of Fourier synthesizer in our life
    1.5. Aim & Objective

Chapter 2: Theory

2.1 Theory of Fourier series
2.2 Attribute of Matlab

Chapter 3: Implementaion

3.1 System Functionality
3.1.1 Input…
3.1.2 Output…
3.2 Effect of Harmonics…
3.3 Example to describe the effect of harmonics
3.4 Examples of Different signals
3.5 Development of the program
3.6 Project timetable

Chapter 4: **Result**

4.1. Software code without GUI

4.2 Software code without GUI

Chapter 5: **Conclusion**

5.1 Conclusion and Recommendation for future work

5.2 Recommendations for re-design

- References
- Appendices

Chapter 1

1.0 Introduction:

This report has been written to represent the Fourier series to describes the basics of construct the Fourier synthesizer and to show the effect of harmonics .

A periodic signal can be described by a Fourier decomposition as Fourier series ,i.e. as a sum of sinusoidal and cosinusoidal oscillations.
By reversing this producer a periodic signal can be generated by superimposing sinusoidal and cosinusoidal waves.

Any periodic waveform can be expressed as an infinite sum of trigonometric function, consisting of sin or cosine function and its successive harmonics .

1.1 Definition of Fourier synthesizer

Fourier synthesis can be known as the inverse of Fourier analysis, in other words we can generate a signal by adding the fundamental frequency to the harmonic components.

In our life , Synthesis means to put together parts or elements so as to form the whole, according to Webster's Dictionary. That is the general concept behind a synthesizer: To put together parts and elements of a sound, from sound generating devices to sound processing devices, and form a whole sound.

1.2 Importance of Fourier synthesizer in the project

It's important because we want to achieve many type of signals such as square ,triangular ,saw tooth ,…… But in fact we can build any type of signal we want by combine the two main signals (cos & sin) and play with the number of harmonic components .

1.3 Relationship between Fourier series & Fourier synthesizer

The Fourier Series is a mathematical tool that allows the representation of any periodic signal as the sum of harmonically related sinusoids. This operation seems incredible. How, for instance, could a function that is not smooth be represented as the sum of sinusoids that are smooth? The answer lies in the number of sinusoids that are added to approximate the desired result. As more sinusoids are added, the output more closely resembles the desired output.

Fourier series turns out that any periodic signal (one that repeats itself at regular intervals) is equal to a sum of sines and cosines. The frequencies of the component sines and cosines are just the frequency of the original periodic signal (the fundamental frequency), plus the "overtones", i.e. the integer multiples of the fundamental frequency.

1.4 Usage of Fourier synthesizer in our life

With this piece of equipment, you can demonstrate and investigate this mathematical idea, adding a succession of sine wave harmonics and viewing the results waveform on an oscilloscope.

Synthesizers and electronic instruments have been in common use in today's music for at least 30 years, and have been used in serious compositional works for much longer than that. Today's compositions feature more synthesizers than ever - from the all-electronic blips found in today's techno music, to the styling and pads behind today's pop music.

1.5 Aim & Objective

The Aim of the project is to Design and implement a Matlab-based program for synthesizing different periodic signals. This involves:

1- Finding the Sinusoidal Fourier Series of commonly-used periodic signals.

2- Convert derived expressions into Matlab code.

3- Analysis of Matlab program to study effect of increasing harmonics.

4- Build a Graphical-User Interface to facilitate the use of the program.

Chapter 2

2.1 Theory Of Fourier Series

Fourier Series is defined as the decomposition which means that take the function and break it to it's component of a periodic signal into discrete frequencies (harmonics), each is a multiple of same basic frequency known as the fundamental frequency. Fourier series is valid only for periodic signals; this periodicity in time- domain forces the Fourier series coefficients to be discrete in the frequency-domain.
The Fourier series represents an infinite number of frequency components which added together yield the time domain, these frequency components constitute a discrete spectrum and the amplitudes of each discrete frequency are given by the coefficients an **bn.**

$$a_n = \quad 1/T \int_{-T/2}^{T/2} p(t).\sin(nw_o t)dt$$

$$b_n = \quad 1/T \int_{-T/2}^{T/2} p(t).\sin(nw_o t)dt$$

$$G(t) = \quad \sum_{n=1}^{\infty} a_n.\cos(nw_o t) + b_n.\sin(nw_o t)$$

2.2 Attributes of the MATLAB

MATLAB has many advantages ,such as integrates mathematical computing, visualization, and its a powerful language to provide a flexible environment for technical computing.

Advantages :

- Powerful
- Many functions and facilities
- User Friendly
- Strong Graphical User Interface tools.

Applications :

- MATLAB also includes tools for:
- Data acquisition Data analysis
- exploration Visualization
- image processing Algorithm prototyping
- development Modeling
- simulation Programming and
- application development

Chapter 3 (Implementation)

3.1 System functionality

3.1.1 Input & User Interface

The program will calculate the Fourier series constants **an** & **bn** of the periodic signals.
After that it will calculate the Fourier series summation of the periodic signals.
The user should Select the type of signal then the user should insert :

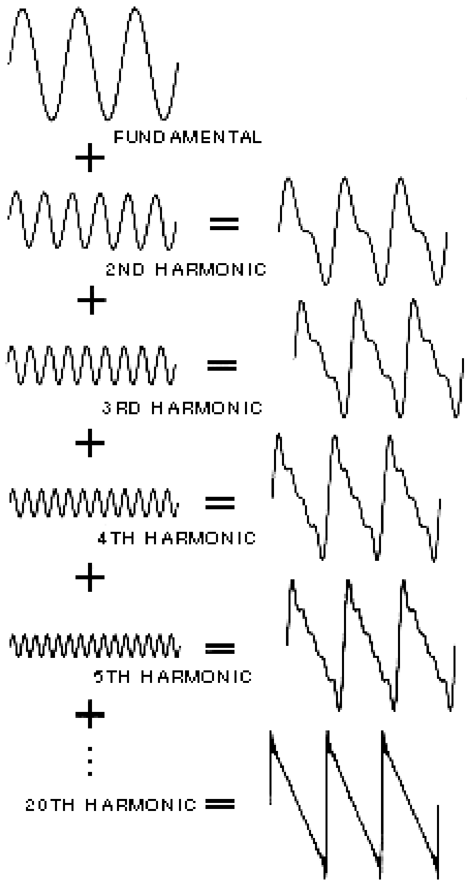
- number of harmonics
- Input frequency
- number of cycles

to specify the signal shape and the accuracy level.

3.1.2 Output

The Matlab plot the approximate shape of the entered signal ,The project includes also many methods to select the good harmonics and reject the others .

3.2 Effect of Harmonics

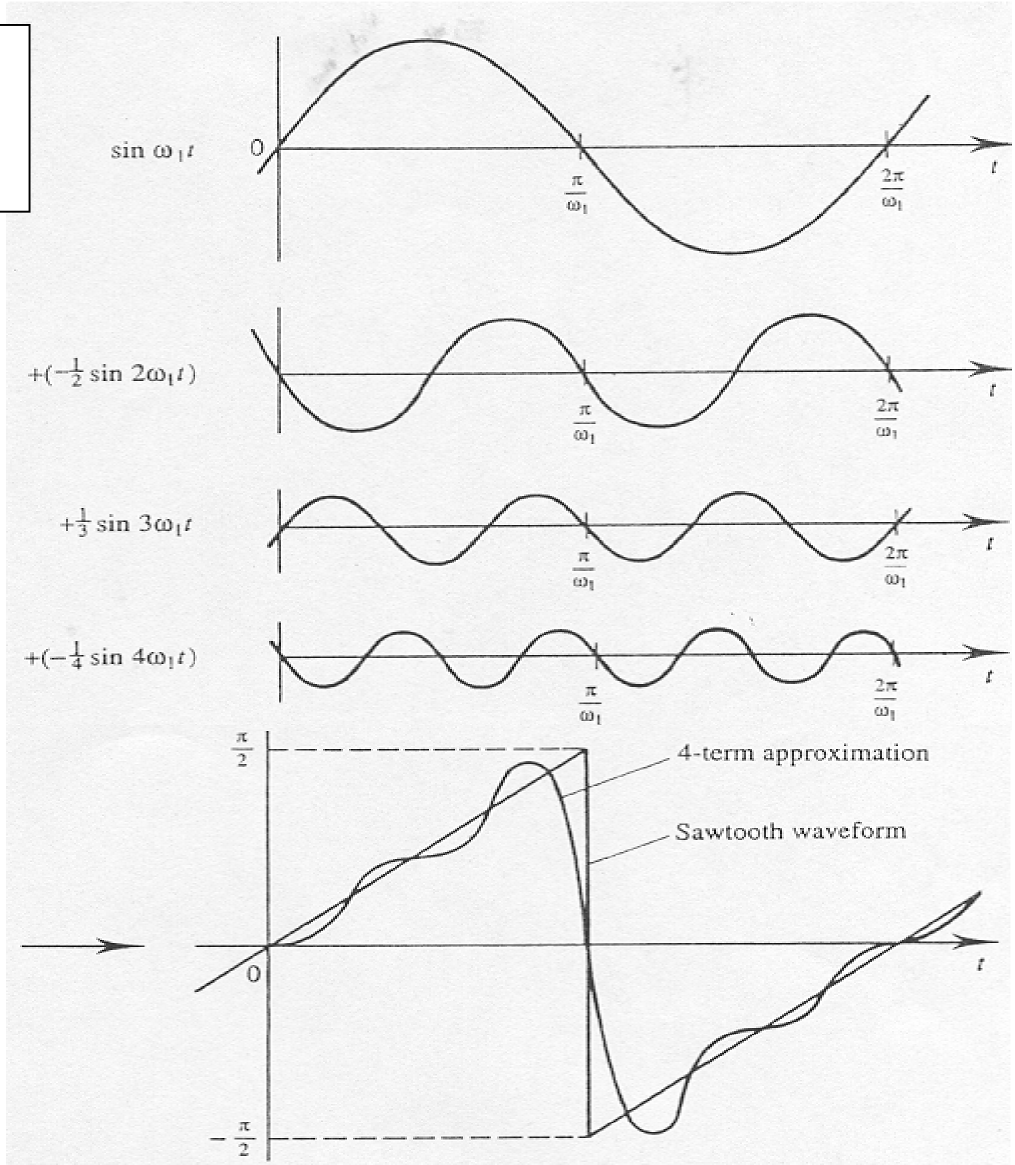


Here we see the fundamental frequency and We add more harmonic component to get the Required shape

As the number of terms or harmonics is Increased, the graph of the Output (the shape of the signal become More accurate .

3.3 Example to describe the effect of harmonics

This example gives
the Fourier series
of the Sawtooth wave:-



Almost any periodic signal can be approximated by adding together sinusoids with the correct freqs.Amplitudes &Phases .This e.q.for the sawtooth waveform approximated by a sum of sinusoids.

As the number of terms is increased, the graph gradually approaches the shape of the original waveform. This ripples increase in number and decrease in amplitude .For practical purposes , the 1st few terms normally suffice to give an accuracy of acceptable level .

3.4 *Examples of Different signals*

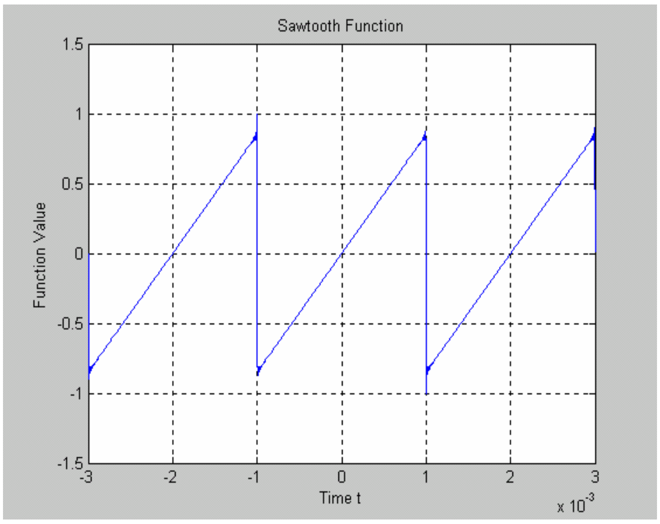
Example of Sawtooth signal:

The coefficients to be used and inserted by the user

$$a_0 = 0$$

$$a_n = 0$$

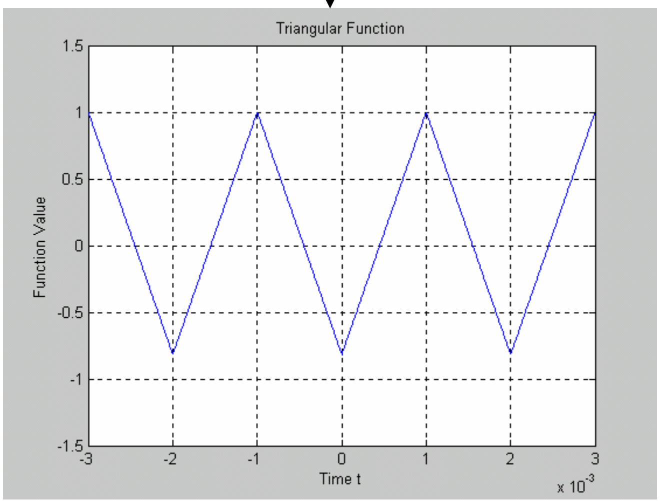
$$b_n = {2*\cos(n*\pi)}/{(n*\pi)}$$



Example of Triangular signal:

The shape of the signal

The coefficients to be used and inserted by the user



$$a_0 = 3*\pi$$

$$a_n = \frac{6}{n^2*\pi*(\cos(n*\pi)-1)}$$

$$b_n = 0$$

Example of Square signal :

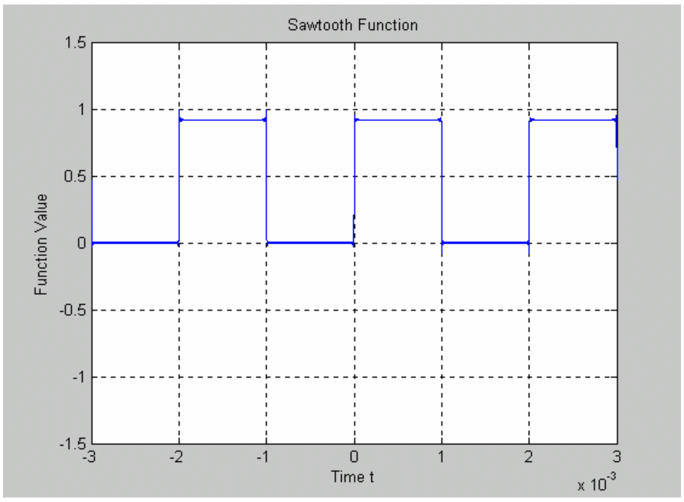
The shape of the signal

The coefficients to be used and inserted by the user

$$a_0 = 0$$

$$a_n = 0$$

$$b_n = \frac{1}{(n*\pi)*(1-\cos(n*\pi))}$$



Example of Bipolar signal :

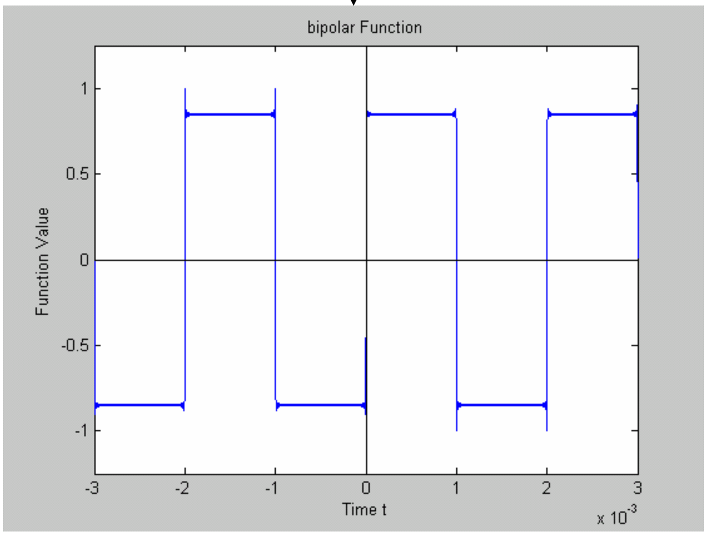
The shape of the signal

The coefficients to be used and inserted by the user

$$a_0 = 0$$

$$a_n = 0$$

$$b_n = \frac{2}{(n*\pi)*(1-\cos(n*\pi))}$$



Example of Rectified sin signal :-

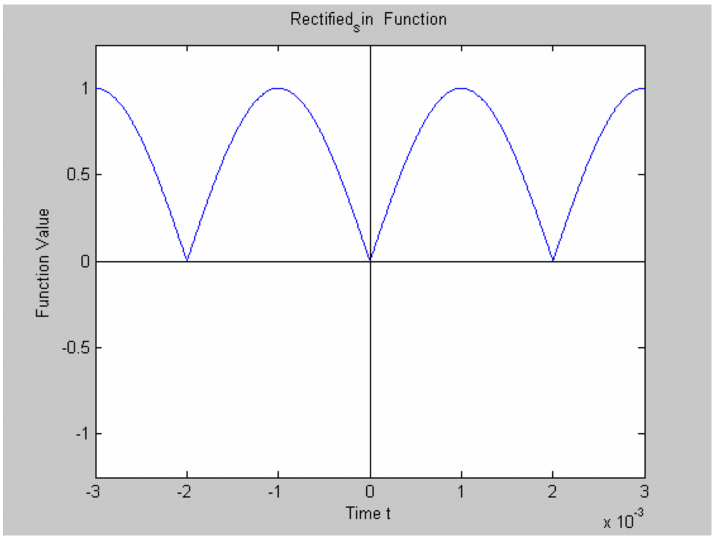
The coefficients to be used and
inserted by the user

$$a_0 = -4/\pi$$

$$a_n = \frac{-4}{(\pi * (n^2 - 1))}$$

$$b_n = 0$$

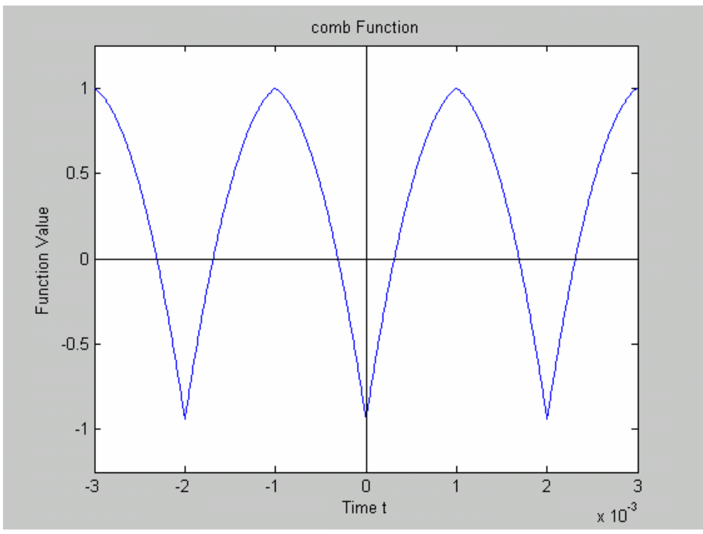
The shape of the signal



Example of comb signal :

The coefficients to be used and
inserted by the user

The shape of the signal



www.maknoon.com

$$a_0 = 1 + (\pi/2)$$

$$a_n = \frac{\cos(n*\pi)}{((n^2*\pi) - 1/(n^2*\pi)}$$

$$b_n = 0$$

3.5 Development Of The Program

Stage 1:

- Many books and references were read to know more about the Matlab and to learn how to use it .
- Collection of signals including many types of signals was define.
- Each signal and its expression was tabulated

Stage2:

Implement our initial program in the Matlab ,that is designed to get a periodic signal.
- the user should insert his function & open the program each time to enter frequency and harmonics then save it.

Stage 3:

Instead of writing the expression of the function,
The program use Fourier series to find the summation of the coefficients to draw the shape.

Stage 4:

We develop the program to ask the user automatically to insert
1.Frequency

2.Number of harmonics
3.Number of cycles

Rather than opening the program to change the coefficients and save it which is wasting the time .

Stage 5:

In this stage we search for the Fourier series constants an & bn have been obtained from a number of references. Emphasis has been on commonly-used periodic signals in communications applications. So we build many type of signals

Stage 6:

In this stage we work in the GUI to improve the project ,So that you can easily select the type of the signal and the three inputs (N,freq,cycles) from the window of the GUI , and also the output signal coming in the same window

3.6 *Project time table*

Week	Task
1	Introduction &distribution of Project list
2	Finalizing groups & Finalizing project selection
3	Learning how to plot in the MATLAB & Collecting equations for Fourier series
4	Preparation for first project Presentation
5	First Project Presentation
6	Complete Program for one type of signals
7	Generalize the program for more periodic signals
8	Test the program for different cases
9	Continue testing of program & Progress Presentation
10	Complete design of graphical interface
11	Writing Final project Report
12	Preparation For final project presentation Complete design of the Final system

Chapter 4 (Result)

4.1 Software code without GUI

You can see the software in Appendix (i) , that used before building the grapgical User Interface, This program is fixed to all of the type of signal we describes before in (Examples) , But only we change the coefficients of the function according to it's type .

We Put all functions we work with it , and the coefficients of it in the previous chapter.

Always when we run the program, It asks about the 3 things (Harmonics, frequency and cycles)

Then the figure coming in the other window ,

4.2 Software code with GUI

Without GUI we work in 3 windows , window of the program , other one for working in the command window , and also when we get the figure we must go to it's window to save it .
This Waste the user time specially when we put the 3 numbers .
So that we work in the graphical user interface then the usage of the program become more easier

As a benefit form this project we was learned more about the matlab and how to use it to right a programming code and generate a Graphical User Interface that important in our study in the future .

- The signals were listed in simple and easy way instead of putting each signal separately.

- We have a program , that is used GUI for any signal types

You can see it in Appendex II

- You can see the interface we plan to do it in the figure 1 in the Appendix

  You can select the type of the signal , add the values of the 3 number (N, frequency and cycles), To get the shape of the signal
  We finish this interface unless using of the options ,But in the future if any body want to improve the interface , he can do it by reading more and more in the Help of the MATLAB

- There is a program which has the designing of the interface named as (graphical)
It has the callback of the 6 functions to the interface and get the output signal to put it in this interface .

You can see it in Appendix III

- There is a program to recall the values of the variables (N, frequency and cycles)
  Named as (action)
  See Appendix IV

Chapter 5 (Conclusion)

5.1 Conclusion and Recommendation for future work

❖Project schedule has been followed without delays.

❖ . Different periodic signals have been synthesized individually.

❖ . Project tasks will be completed when the graphical user interface design is finished.

❖ Fourier Synthesizer theory is difficult to explain ,but with using the GUI its become clear and easy to learn.

❖ Fourier series describe the concept of the Fourier Synthesizer .

❖ . Sawtooth wave became like a sine wave but only have straight lines.

❖ . Square Wave moves sharply but evenly in amplitude.

❖ . Triangle Wave is like a sine wave but move up and down at a more abrupt rate

5.2 Recommendations for re-design

❖ As a benefit form this project we was learned more about the matlab and how to use it to right a programming code and generate a Graphical User Interface that important in our study in the future .

❖Our GUI is taking Time to Build it because we want to develop our program, and become easy to use so that the understanding of the project become more flexible .

❖ After finishing our project we can advise any teachers to use it to explain the theory of the Fourier series that is very easy to understand by explaining the Fourier synthesizer.

- ❖ In the future we can add other signal so that we get more waves .

References:

Books:

l.A.GLOVER &P.M.GRNT “Digital Communication”
PRENTICE HALL “1997,chp2,pp27

Simon Haykin “Communication System “
1978,chp2,pp12

M.J.Englefied “Mathematical method “
1987,chp16,pp414

Website:

http://www.math.ubc.ca/
http://mathworld.wolfram.com/

Appendix I. Software with out GUI

This is Matlab program is to generate many type of periodic signal , Just by choosing the values of the coefficient of the Fourier series then put it in this program to get the required signal :-

```
% This sample program calculate the Fourier series approximation of a Bipolar
%Function

clc;
clf;
clear all;
close all;

freq=input('Enter the frequency of the periodic signal to be generated freq=');
N=input('Enter the number of harmonics N=');
cycles=input('Enter the number of cycles to be plotted cycles=');

T=1/freq;                w0=2*pi/T;
t_initial=-T*cycles/2;
t_final=T*cycles/2;
step=(t_final-t_initial)/1000;
t=[t_initial:step:t_final];

a0=0;

for n=1:N
   a(n)=0;
   b(n)=2/(n*pi)*(1-cos(n*pi));
end;

for ii=1:length(t)
  func(ii)=a0/2;
  for n=1:N
    func(ii)=func(ii)+a(n)*cos(n*w0*t(ii))+b(n)*sin(n*w0*t(ii));
  end;
end;

maxvalue=max(func);
```

```
func=func/maxvalue;

x_time=[t_initial t_final];
y_time=[0 0];

x_amp=[0 0];
y_amp=[-1.25 1.25];

plot(t,func,x_time,y_time,'k',x_amp,y_amp,'k');

axis ([t_initial t_final -1.25 1.25]);
Title('bipolar Function');

xlabel('Time t');
ylabel('Function Value');

end;
```

Appendix II. Software with GUI

This software was fixed for whole function but only to change the coefficients .

```
% This program calculate the Fourier series approximation of a Square Function

%clc;
%clf;
%clear all;
%close all;



%freq=input('Enter the frequency of the periodic signal to be generated freq=');
%N=input('Enter the number of harmonics N=');
%cycles=input('Enter the number of cycles to be plotted cycles=');

freq=str2num(freq);
N=str2num(N);
cycles=str2num(cycles);

T=1/freq;                w0=2*pi/T;
t_initial=-T*cycles/2;
t_final=T*cycles/2;
step=(t_final-t_initial)/1000;
t=[t_initial:step:t_final];

a0=1;

for n=1:N
    a(n)=0;
    b(n)=1/(n*pi)*(1-cos(n*pi));
end;

for ii=1:length(t)
   func(ii)=a0/2;
   for n=1:N
```

```
      func(ii)=func(ii)+a(n)*cos(n*w0*t(ii))+b(n)*sin(n*w0*t(ii));
   end;
end;

maxvalue=max(func);
func=func/maxvalue;

plot(t,func);

grid on;
axis ([t_initial t_final -1.5 1.5]);
Title('Square Function');

xlabel('Time t');
ylabel('Function Value');

end;
```

Appendix III. Graphical Software

```
function fig = graphical()
% This is the machine-generated representation of a Handle
Graphics object
% and its children.  Note that handle values may change when
these objects
% are re-created. This may cause problems with any callbacks
written to
% depend on the value of the handle at the time the object was
saved.
% This problem is solved by saving the output as a FIG-file.
%
% To reopen this object, just type the name of the M-file at the
MATLAB
% prompt. The M-file and its associated MAT-file must be on your
path.
%
% NOTE: certain newer features in MATLAB may not have been saved
in this
% M-file due to limitations of this format, which has been
superseded by
% FIG-files.  Figures which have been annotated using the plot
editor tools
% are incompatible with the M-file/MAT-file format, and should be
saved as
% FIG-files.

load graphical

h0 = figure('Color',[0.8 0.8 0.8], ...
   'Colormap',mat0, ...
   'CreateFcn','plotedit(gcbf,''promoteoverlay''); ', ...
   'FileName','K:\Project\Year 3 - Project - Humaid Sultan
Obaid\graphical.m', ...
   'PaperPosition',[10 10 576 432], ...
   'PaperType','A4', ...
   'PaperUnits','points', ...
   'Position',[1 29 1024 702], ...
   'Tag','Fig1', ...
   'ToolBar','none');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
```

```
   'BackgroundColor',[0.752941176470588 0.752941176470588
0.752941176470588], ...
   'ListboxTop',0, ...
   'Position',[13.5 443.25 145.5 48], ...
   'String','Select the type of the signal', ...
   'Tag','Pushbutton1');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.752941176470588 0.752941176470588
0.752941176470588], ...
   'Callback','func1_sawtooth', ...
   'ListboxTop',0, ...
   'Position',[19.5 361.5 69.75 48], ...
   'String','Sawtooth', ...
   'Tag','Pushbutton2');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.752941176470588 0.752941176470588
0.752941176470588], ...
   'Callback','func2_square', ...
   'ListboxTop',0, ...
   'Position',[16.5 302.25 72 45.75], ...
   'String','Square', ...
   'Tag','Pushbutton2');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.752941176470588 0.752941176470588
0.752941176470588], ...
   'Callback','func3_bipolar', ...
   'ListboxTop',0, ...
   'Position',[16.5 242.25 74.25 45.75], ...
   'String','Bipolar', ...
   'Tag','Pushbutton2');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.752941176470588 0.752941176470588
0.752941176470588], ...
   'Callback','func5_triangular', ...
   'ListboxTop',0, ...
   'Position',[16.5 182.25 73.5 44.25], ...
   'String','Triangular', ...
   'Tag','Pushbutton2');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.752941176470588 0.752941176470588
0.752941176470588], ...
   'Callback','func4_rectified_sin', ...
   'ListboxTop',0, ...
   'Position',[18 123.75 73.5 45.75], ...
   'String','Rectified sin', ...
   'Tag','Pushbutton2');
h1 = uicontrol('Parent',h0, ...
```

```
    'Units','points', ...
    'BackgroundColor',[0.752941176470588 0.752941176470588
0.752941176470588], ...
    'Callback','func7_comb', ...
    'ListboxTop',0, ...
    'Position',[18.75 63 73.5 49.5], ...
    'String','Comb', ...
    'Tag','Pushbutton2');
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','action EditText1', ...
    'ListboxTop',0, ...
    'Position',[666 372.75 71.25 34.5], ...
    'Style','edit', ...
    'Tag','EditText1', ...
    'Value',77);
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','action EditText2', ...
    'CreateFcn','action EditText2', ...
    'ListboxTop',0, ...
    'Position',[670.5 260.25 69 36], ...
    'Style','edit', ...
    'Tag','EditText2', ...
    'UserData','[ ]', ...
    'Value',7);
h1 = uicontrol('Parent',h0, ...
    'Units','points', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','action EditText3', ...
    'ListboxTop',0, ...
    'Position',[668.25 162.75 70.5 33.75], ...
    'Style','edit', ...
    'Tag','EditText3', ...
    'Value',7);
h1 = axes('Parent',h0, ...
    'Units','pixels', ...
    'CameraUpVector',[0 1 0], ...
    'CameraUpVectorMode','manual', ...
    'Color',[1 1 1], ...
    'ColorOrder',mat1, ...
    'Position',[164 58 699 520], ...
    'Tag','Axes1', ...
    'XColor',[0 0 0], ...
    'YColor',[0 0 0], ...
    'ZColor',[0 0 0]);
h2 = text('Parent',h1, ...
    'Color',[0 0 0], ...
    'HandleVisibility','off', ...
    'HorizontalAlignment','center', ...
```

```
   'Position',[0.4985673352435529 -0.04624277456647397
9.160254037844386], ...
   'Tag','Axes1Text4', ...
   'VerticalAlignment','cap');
set(get(h2,'Parent'),'XLabel',h2);
h2 = text('Parent',h1, ...
   'Color',[0 0 0], ...
   'HandleVisibility','off', ...
   'HorizontalAlignment','center', ...
   'Position',[-0.04154727793696275 0.4971098265895955
9.160254037844386], ...
   'Rotation',90, ...
   'Tag','Axes1Text3', ...
   'VerticalAlignment','baseline');
set(get(h2,'Parent'),'YLabel',h2);
h2 = text('Parent',h1, ...
   'Color',[0 0 0], ...
   'HandleVisibility','off', ...
   'HorizontalAlignment','right', ...
   'Position',mat2, ...
   'Tag','Axes1Text2', ...
   'Visible','off');
set(get(h2,'Parent'),'ZLabel',h2);
h2 = text('Parent',h1, ...
   'Color',[0 0 0], ...
   'HandleVisibility','off', ...
   'HorizontalAlignment','center', ...
   'Position',[0.4985673352435529 1.013487475915222
9.160254037844386], ...
   'Tag','Axes1Text1', ...
   'VerticalAlignment','bottom');
set(get(h2,'Parent'),'Title',h2);
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.752941176470588 0.752941176470588
0.752941176470588], ...
   'FontSize',12, ...
   'ListboxTop',0, ...
   'Position',[244.5 459.75 294 32.25], ...
   'String','The shape of the signal', ...
   'Style','text', ...
   'Tag','StaticText1');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'ListboxTop',0, ...
   'Position',[672 406.5 60 29.25], ...
   'String','No. of Harmonics', ...
   'Style','text', ...
   'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'FontSize',12, ...
```

```
   'ListboxTop',0, ...
   'Position',[674.25 296.25 60 29.25], ...
   'String','frequency', ...
   'Style','text', ...
   'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.752941176470588 0.752941176470588
0.752941176470588], ...
   'FontSize',12, ...
   'ListboxTop',0, ...
   'Position',[672.75 195.75 60 29.25], ...
   'String','No. of cycles', ...
   'Style','text', ...
   'Tag','StaticText2');
h1 = uicontrol('Parent',h0, ...
   'Units','points', ...
   'BackgroundColor',[0.752941176470588 0.752941176470588
0.752941176470588], ...
   'Callback',mat3, ...
   'ListboxTop',0, ...
   'Position',[652.5 90 95.25 42], ...
   'String','Exit from MATLAB', ...
   'Tag','Pushbutton3');
if nargout > 0, fig = h0; end
```

Appendix IV   Action Software

%This software is to callback the values of the variables to the interface

```
function action(act)

global N freq cycles

if(act=='EditText1')
   x=(get(gcbo,'String'))
   N=x;
elseif(act=='EditText2')
   y=(get(gcbo,'String'))

   freq=y;

elseif (act=='EditText3')
   z=(get(gcbo,'String'))
   cycles=z;
end
```

Appendix V   Start Software

% This function to start the interface

global N freq cycles

graphical

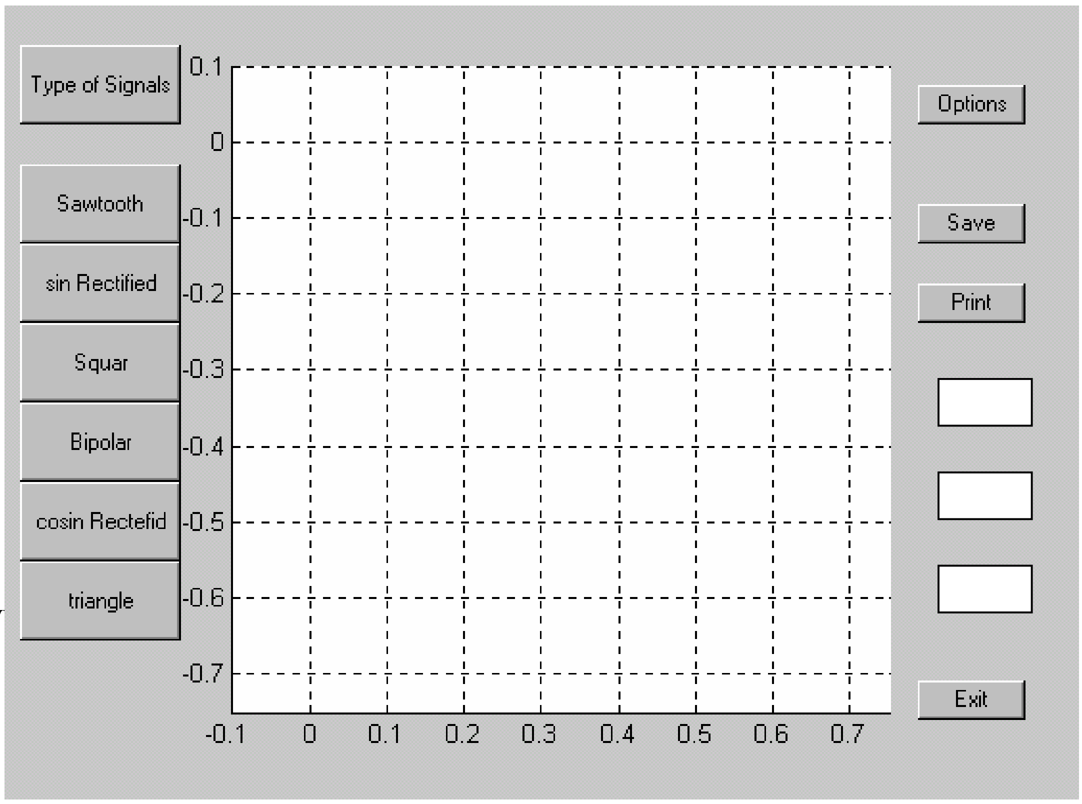
Appendix V   Exit Software

% This function to exit from the MATLAB

exit

Appendix Figure 1

Output signal



ww