**大连理工大学实验报告**

学院（系）： 电子信息与电气工程学部 专业： 电子信息工程 班级： 电信1801

姓 名： 杨题鸣 学号： 201883016 组： \_\_\_

实验时间： 2020/5/16 实验室： 实验台：

指导教师签字： 成绩：

**实验名称**

一、实验题目和结果

学会利用MATLAB进行周期信号的傅里叶展开，频谱分析，非周期信号傅里叶变换及性质

代码如下：

function varargout = main(varargin)

% MAIN MATLAB code for main.fig

% MAIN, by itself, creates a new MAIN or raises the existing

% singleton\*.

%

% H = MAIN returns the handle to a new MAIN or the handle to

% the existing singleton\*.

%

% MAIN('CALLBACK',hObject,eventData,handles,...) calls the local

% function named CALLBACK in MAIN.M with the given input arguments.

%

% MAIN('Property','Value',...) creates a new MAIN or raises the

% existing singleton\*. Starting from the left, property value pairs are

% applied to the GUI before main\_OpeningFcn gets called. An

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to main\_OpeningFcn via varargin.

%

% \*See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one

% instance to run (singleton)".

%

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help main

% Last Modified by GUIDE v2.5 29-Dec-2018 21:42:40

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @main\_OpeningFcn, ...

'gui\_OutputFcn', @main\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

% End initialization code - DO NOT EDIT

% --- Executes just before main is made visible.

function main\_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% varargin command line arguments to main (see VARARGIN)

% Choose default command line output for main

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes main wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = main\_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

varargout{1} = handles.output;

% --- Executes on selection change in popupmenu1.

function popupmenu1\_Callback(hObject, eventdata, handles)

% hObject handle to popupmenu1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents as cell array

% contents{get(hObject,'Value')} returns selected item from popupmenu1

choice=get(handles.popupmenu1,'Value');

fs=500;

dt=1/fs;

t=0:dt:2;

N=length(t);

switch choice

case 1

%请产生10Hz正弦波，时间为上面定义的t，信号存储在变量x中

x=sin(2\*pi\*10\*t);

case 2

%请产生周期为1的方波，时间为上面定义的t，信号存储在变量x中

x=square(2\*pi\*t,50);

case 3

x=sawtooth(2\*pi\*t);

case 4

x=heaviside(t);

end

plot(handles.wave,t,x);

set(handles.wave,'YLim',[-1.5 1.5]);

title(handles.wave,'波形');

%==================================

%请在下面输入对信号x进行傅里叶变换的代码，并显示变换后的幅度谱和相位谱

%角频率为变量W，傅里叶变换后的结果存放在F中

k = -N:N;

W = 2\*pi\*k/((2\*N+1)\*dt);

F = dt \* x\*exp(-j\*t'\*W);

%==========================================

plot(handles.magnitude,W,abs(F));

title(handles.magnitude,'幅度谱');

plot(handles.phase,W,angle(F));

title(handles.phase,'相位谱');

% --- Executes during object creation, after setting all properties.

function popupmenu1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to popupmenu1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

% --- Executes on button press in pushbutton2.

function pushbutton2\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

set(handles.panel,'Visible','on');

set(handles.panel1,'Visible','off');

set(handles.panel2,'Visible','off');

% --- Executes on button press in pushbutton3.

function pushbutton3\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

set(handles.panel,'Visible','off');

set(handles.panel1,'Visible','on');

set(handles.panel2,'Visible','off');

% --- Executes on button press in pushbutton4.

function pushbutton4\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton4 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

set(handles.panel,'Visible','off');

set(handles.panel1,'Visible','off');

set(handles.panel2,'Visible','on');

function edit1\_Callback(hObject, eventdata, handles)

% hObject handle to edit1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text

% str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.

function edit1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

% --- Executes on button press in button1.

function button1\_Callback(hObject, eventdata, handles)

% hObject handle to button1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

%========请补充下面的代码=========================

%参考实验讲义，请在这里生成幅度为1，周期为1的方波信号，时间变量定义为t，方波信号为y

t=-1:0.001:1;%定义信号时间

y=square(2\*pi\*t,50); %生成方波

plot(t,y)

%============================================

%获取用户输入的谐波次数后，显示出最高次谐波的波形。

n\_max=1:1:str2num(get(handles.edit1,'String')); %项数

y\_m=sin(2\*pi\*max(n\_max)\*t);

plot(handles.C\_harmonic,t,y\_m);

title(handles.C\_harmonic,'最高次谐波')

%======请补充下面的代码======================

%请在这里写代码，完成前max次谐波对方波信号的合成

%合成后的结果存放在x变量中

m=length(n\_max); %需要循环几次

for k=1:m

n=1:2: n\_max(k); %获得奇数向量，n 是一个 1\*d 的向量

M=4./(pi\*n); %系数，M 是一个 1\*d 的向量

N=sin(2\*pi\*n'\*t); %谐波，如果将 t 看作是一个常数，N 是一个 d\*1 的向量

x=M\*N; %合成结果，x 是一个 1\*2001 的向量

% figure; %生成空白图像

% plot(t,y,'b',t, x,'r'); %画曲线图

end

%============================

plot(handles.synthetic,t,y,t,x);

set(handles.synthetic,'YLim',[-1.5 1.5]);

title(handles.synthetic,'多次合成后')

% --- Executes on button press in pushbutton6.

function pushbutton6\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton6 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

[filename,filepath,~]=uigetfile({'\*.mp3';'\*.mp4';'\*.wav'},...

'Select an audio file',...

'MultiSelect','off');

if(filename==0)

return

else

file=strcat(filepath,filename);

set(handles.ViewPath,'String',filename);

[tdata,fs]=audioread(file);%tdata是读入的音频信号，fs是音频采样频率

tdata=tdata(:,2);%取出右声道的数据进行处理，注意：也可以用左声道数据，或者两个声道都处理，都用

tdata=tdata';

t=0:1/fs:(length(tdata)-1)/fs;

plot(handles.wave1,t,tdata);

title(handles.wave1,'波形');

N=length(t);

fdata=abs(fft(tdata,N));

pdata=angle(fft(tdata,N));

df=fs/N;

f=(0:df:df\*(N-1))-fs/2;

plot(handles.magnitude1,f,fftshift(fdata));

title(handles.magnitude1,'幅度谱');

plot(handles.phase1,f,pdata);

title(handles.phase1,'相位谱');

end

% --- Executes on button press in play.

function play\_Callback(hObject, eventdata, handles)

% hObject handle to play (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

filename=get(handles.ViewPath,'String');

[y Fs]=audioread(filename);

sound(y,Fs);

% --- Executes during object creation, after setting all properties.

function C\_harmonic\_CreateFcn(hObject, eventdata, handles)

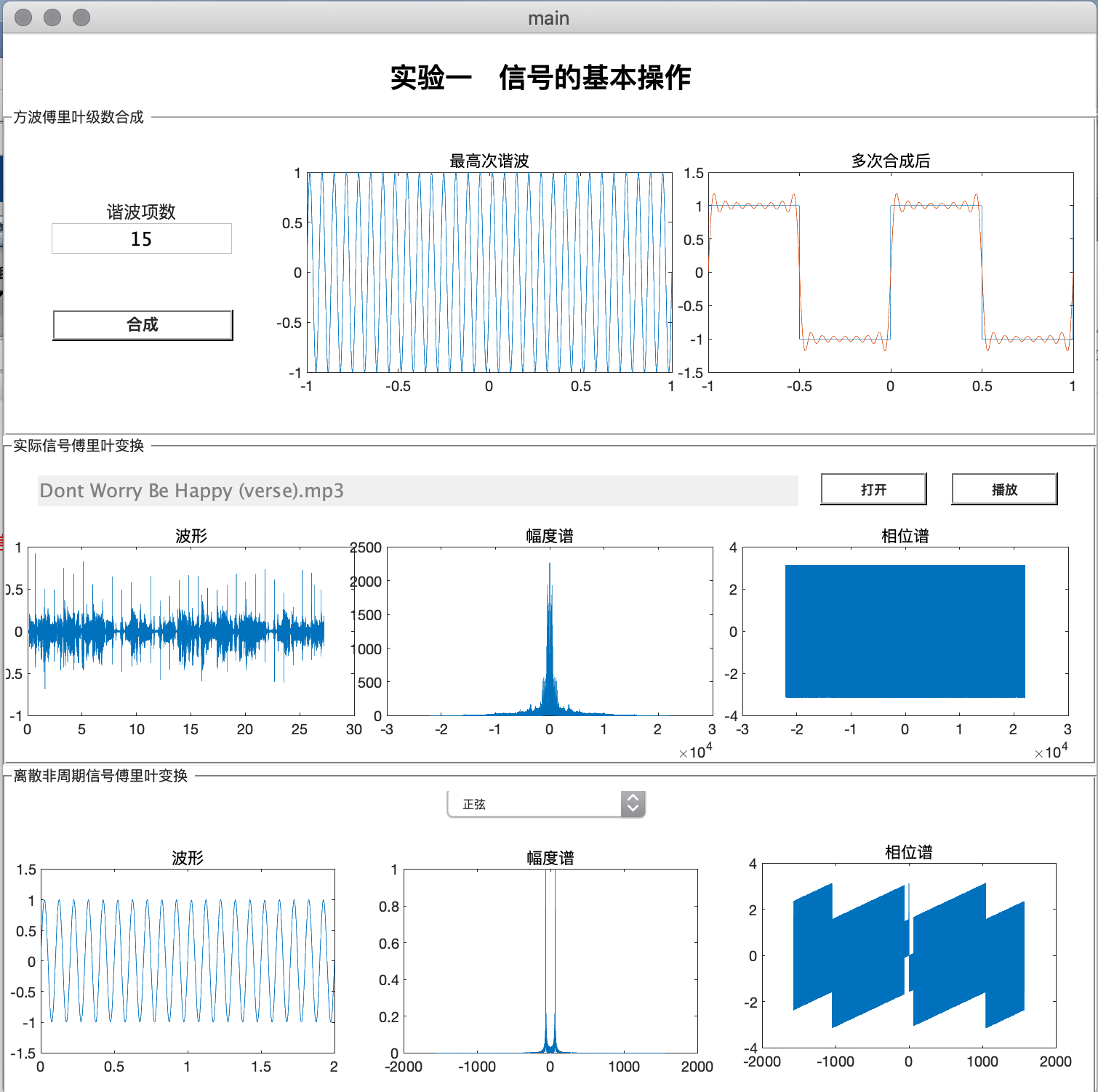
% hObject handle to C\_harmonic (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate C\_harmonic

**结果：**



二、实验总结

我是一名电信学生，今天写完信号与系统的试验报告，才发现我数电实验的报告格式不对。这一次实验改进了上次实验的报告格式问题。

这一次实验比较的简单，用了不到1个半小时就能做完，实验的讲义里面的很多代码，都能直接用到实验当中来。通过这次实验，我进一步的了解了傅立叶变换变换在实际当中的运用，还是有很多的成就感。之前接触过的图像处理，都会用到傅立叶变换，但是都是有现成的api来实现，自己也并没有亲自，一步一步的把波形和幅度谱和相位谱给画出来。实验能进一步的提高了我在不同编程语言之间的切换，能够提高整体的编程能力。

这几次的编程，都是基于一些简单的函数，如方波，正弦波等等来做的实验，如果能够与图像处理有关的那就更有意思了。