**第五章Matlab作业报告**

2015010912014 耿瑞旭

[M5\_2: 1](#_Toc482018791)

[1. 设计思路 1](#_Toc482018792)

[2. ABC的具体实现 1](#_Toc482018793)

[3. D(GUI)的实现 2](#_Toc482018794)

[4. 运行结果 2](#_Toc482018795)

[5. 结论 3](#_Toc482018796)

[M5\_10 3](#_Toc482018797)

[0. 注：因做题需要，将题目中”例5.13”改为”例“5.14” 3](#_Toc482018798)

[1. 实现思路 3](#_Toc482018799)

[2. 代码实现 3](#_Toc482018800)

[3. 输出结果： 4](#_Toc482018801)

[M5\_11 5](#_Toc482018802)

[1. 设计思路 5](#_Toc482018803)

[2. 代码实现 5](#_Toc482018804)

[3. 实验结果 7](#_Toc482018805)

[附录1 9](#_Toc482018806)

[M5\_2修改后代码 9](#_Toc482018807)

[M5\_2 GUI代码 10](#_Toc482018808)

[M5\_10.m 14](#_Toc482018809)

[conv\_save.m 15](#_Toc482018810)

[chapter5.m 16](#_Toc482018811)

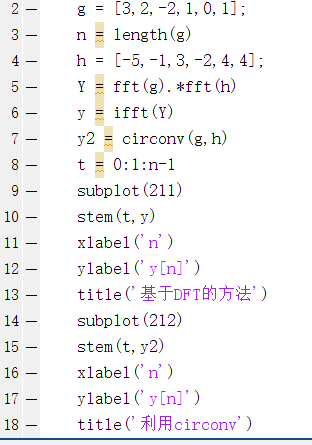
M5\_2:

1. 设计思路

该题目的完成可分为如下步骤：

1. 时域圆周卷积在频域上相当于两序列的DFT 的相乘,而计算DFT 可以采用它的 快速算法——快速傅立叶变换(FFT)，因此可用两次fft和一次ifft计算出圆周卷积
2. 将每一问用DFT计算出的圆周卷积画出来
3. 将每一问用circonv得到的圆周卷积画出来
4. 设计GUI界面，使得两者结果的比较更加直观
5. ABC的具体实现

实现第一问的代码如下所示，思路如上所述



将这个代码扩展至3问

具体代码请见附录

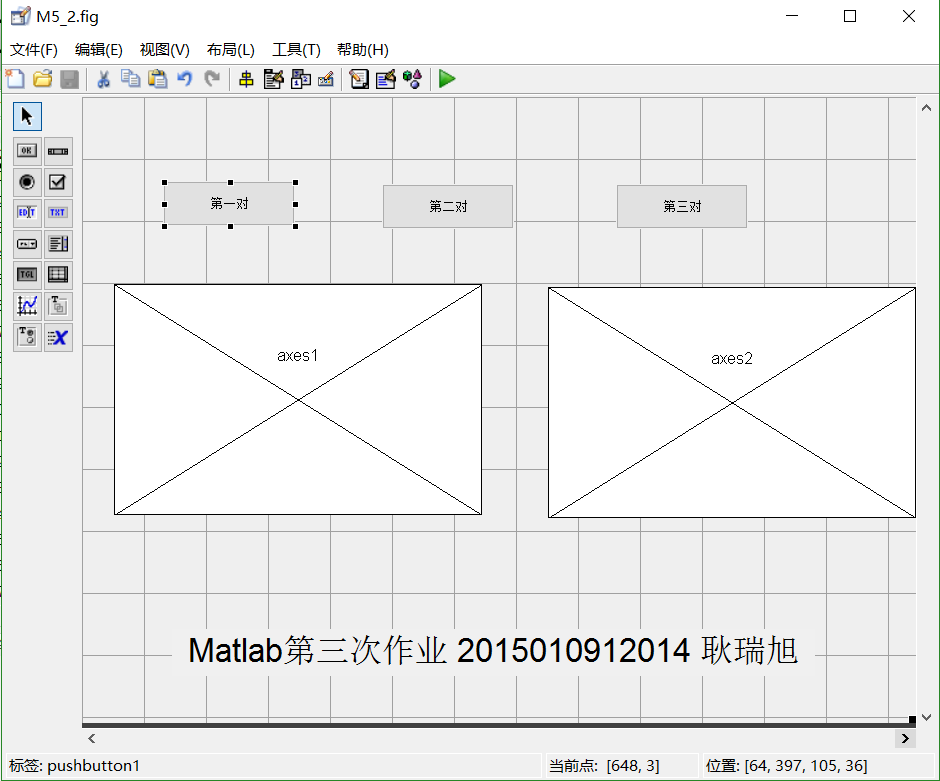
1. D(GUI)的实现

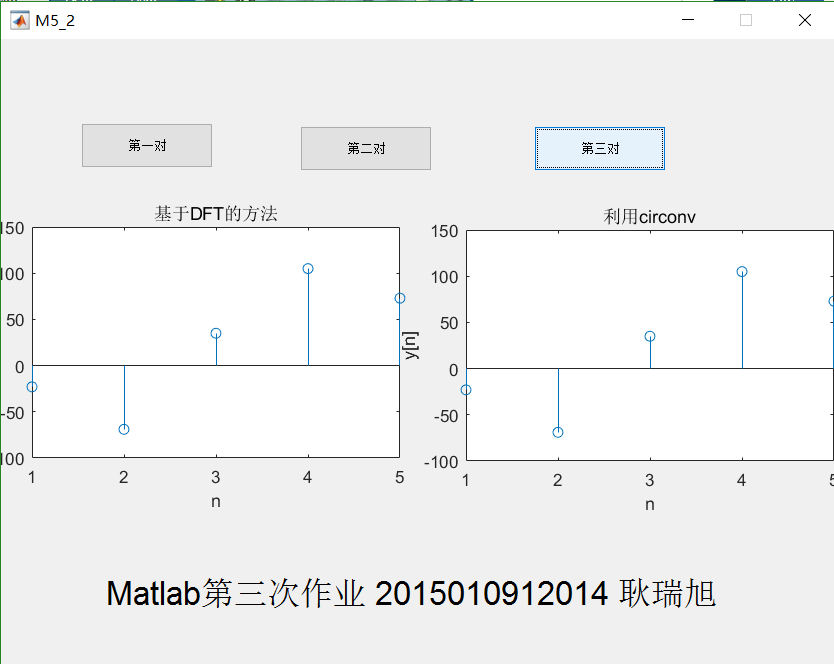
设计合理的GUI布置，仿照abc设置回掉函数即可（更加方便）。具体代码请见附录

1. 运行结果
2. 利用非GUI实现的结果如下：



1. 利用GUI实现的结果如下所示：





具体实现代码请见附录

1. 结论

两种方法计算的结果是一致的

M5\_10

1. 注：因做题需要，将题目中”例5.13”改为”例“5.14”
2. 实现思路

程序大致分为以下几部分：

1. 产生原序列，并作图
2. 产生高频序列，并作图
3. 产生混合序列，做fft，将50~206置0，再ifft，得到滤波后的序列,再绘制。

但是，在实现B时，遇到了问题。根据题目要求，此处的高频序列，应该是特定，dft的50~256项不为0，其他项都为0的高频分量。因此，若直接使用awgn函数加入高斯白噪声，无法很好地得到复原信号。

1. 代码实现

n = 0:1:255

x = zeros(1,256)

for m = 0:1:255

x(1,m+1) = 0.1\*m\*exp(-0.03\*m)

end

noise\_fft = [zeros(1,50) 10\*rand(1,156)-10 zeros(1,50)]

noise = 5\*(ifft(y\_fft,256)) % 利用随机数+ifft产生特定频率范围的随机信号

subplot(412)

plot(n,noise)

xlabel('n')

ylabel('noise')

title('高频噪音')

subplot(411)

plot(n,x)

xlabel('n')

ylabel('x[n]')

title('原信号')

subplot(413)

y = noise + x

plot(n,y)

xlabel('n')

ylabel('y[n]')

title('加入噪音的信号')

z = fft(y,256)

for m = 50:1:206

z(1,m) = 0 %滤波

end

recover = ifft(z,256)

subplot(414)

plot(n,recover)

xlabel('n')

ylabel('recover')

title('复原后的信号')

1. 输出结果：



由此可见，原信号得到了恢复，滤波效果良好。

M5\_11

1. 设计思路

与利用filter实现滑动平均滤波类似

本题目利用重叠保留法的线性卷积实现滤波

其中M由用户输入确定

为了保证N大于M，直接设定N=M+2

其余思路与书上所述重叠保留法求线性卷积相同

1. 代码实现

函数代码如下：

%%重叠保留法，具体见课本169页

function conv1 = conv\_save(x,h)

% x = ones(1,200) %用于测试

% for m = 1:1:200

% x(m) = 8\*rand(1)-4

% end

% % x = [2,-3,4,5,-6,7,8,-9,-10,11,-12,-13,-14]

% h = [1,-2,-3] %用于测试

%x的长度应远远大于h的长度

h = h'

len\_x = length(x);

len\_h = length(h);

M = len\_h

len\_conv = len\_x + len\_h -1

% temp = ceil(log2(len\_h));

% N = 2^(temp); %171页下部的M和N

N = M+2

num = floor(len\_x/M) +1 %对x切片的数量

x\_m = zeros(num,N) %对x切的片，每一行都是一个片

cir\_m = zeros(num,len\_conv) %x的每个切片和y[n]相卷，存入一个矩阵

%N点圆周卷积，而M<=N,故需要视情况对h进行补0操作

if(N > len\_h)

for i = len\_h+1:1:N

h(1,i) = 0;

end

end

%对x进行切片，每片长度为N

iN = 1; %第iN个切片

iM = 1; %切片中的第

% for m = 1:N:num\*N

x\_m(1,:) = [zeros(1,N-M) x(1,1:M)]

x\_m(num,:) = [x(1,len\_x-N+M-mod(len\_x,M)+1:len\_x) zeros(1,M-mod(len\_x,M)) ]

for iN = 2:1:num-1

for iM = 1:1:N

x\_m(iN,iM) = x(1,(iN-1)\*M+(iM-N+M))

end

% cir\_m(iN,1+(iN-1)\*(N-M+1):1+(iN-1)\*(N-M+1)+(N-1)) = circonv(x\_m(iN,:),h)

end

for iN=1:1:num

for iM=1:1:N

if 1+(iN-1)\*(N-M+1)+(N-1) < len\_conv

cir\_m(iN,1+(iN-1)\*(N-M+1):1+(iN-1)\*(N-M+1)+(N-1)) = circonv(x\_m(iN,:),h)

end

end

end

% circonv()

% aban = M-1 %开头不正确的需要被丢弃

%去掉cir\_m每一行的前(M-1)个元素

cir\_m\_del = cir\_m;

for m =1:1:num

cir\_m\_del(m,1+(m-1)\*(N-M+1):1+(m-1)\*(N-M+1)+(M-1-1)) = 0

end

%将cir\_m\_del合并成为一个行向量，可得到线性卷积序列

% conv = zeros(1,len\_conv)

% conv = zeros(1,len\_x)

conv2 = sum(cir\_m\_del)

conv1 = conv(x,h)

for iJ = 1:1:len\_conv-2

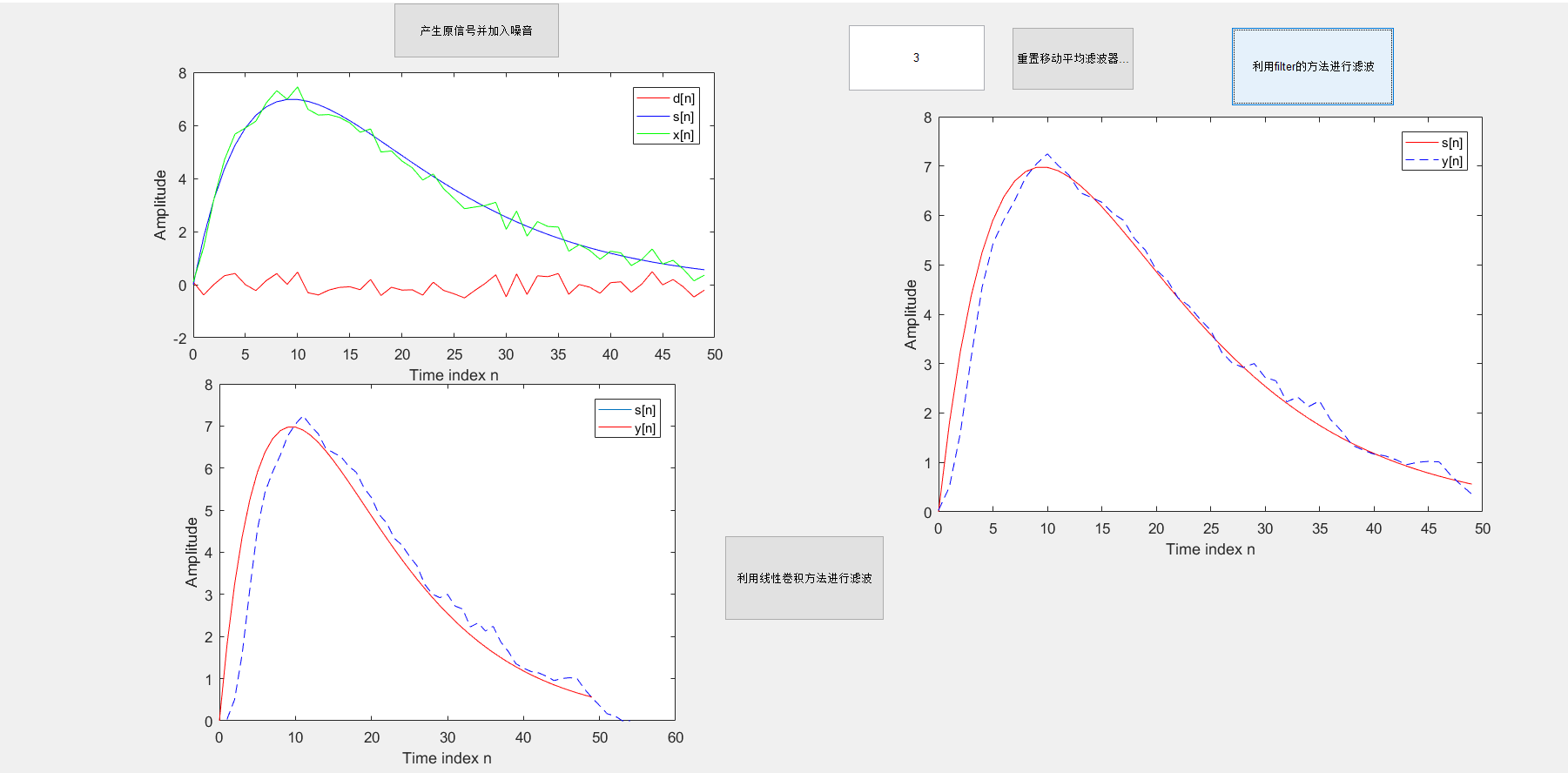
conv2(iJ) = conv2(iJ+2);

end

1. 实验结果

如下图所示：

(为了对比更加清晰，直接使用GUI界面显示)

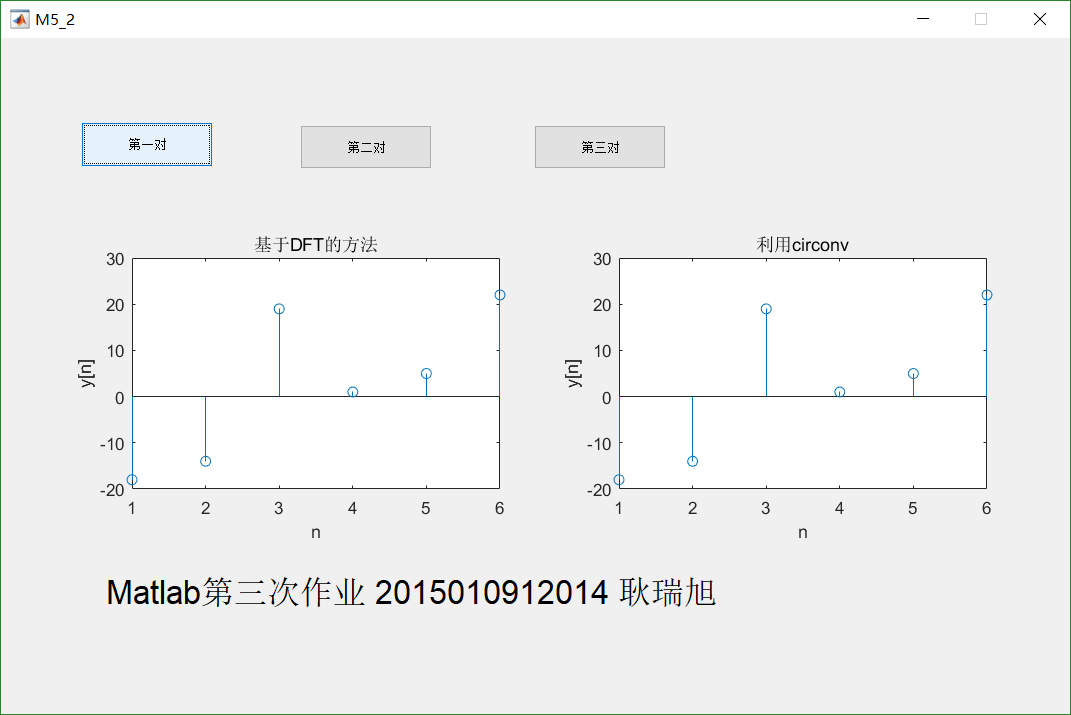


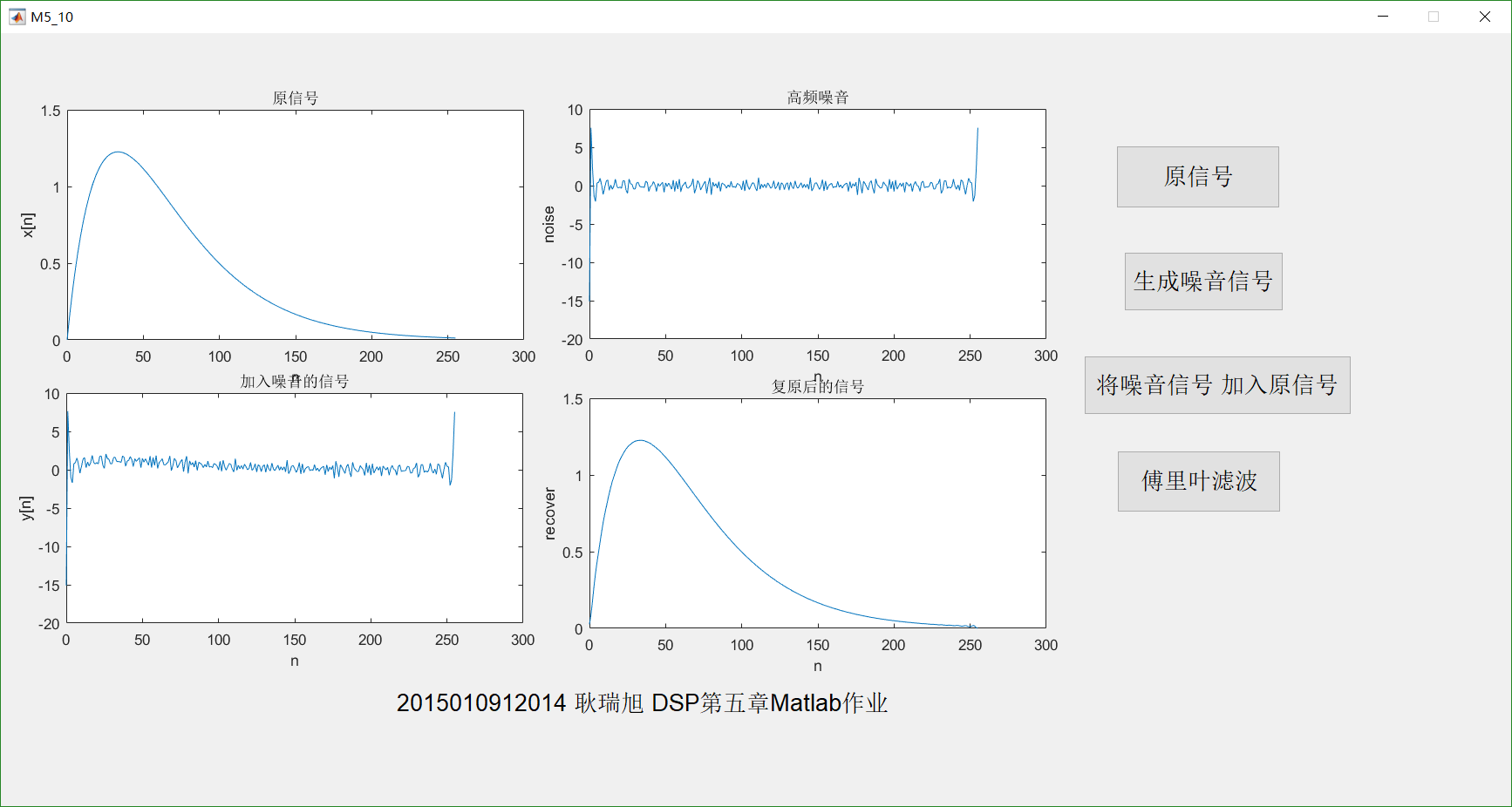
可见除了尾部由于重叠保留法会多出0之外，其余部分效果良好

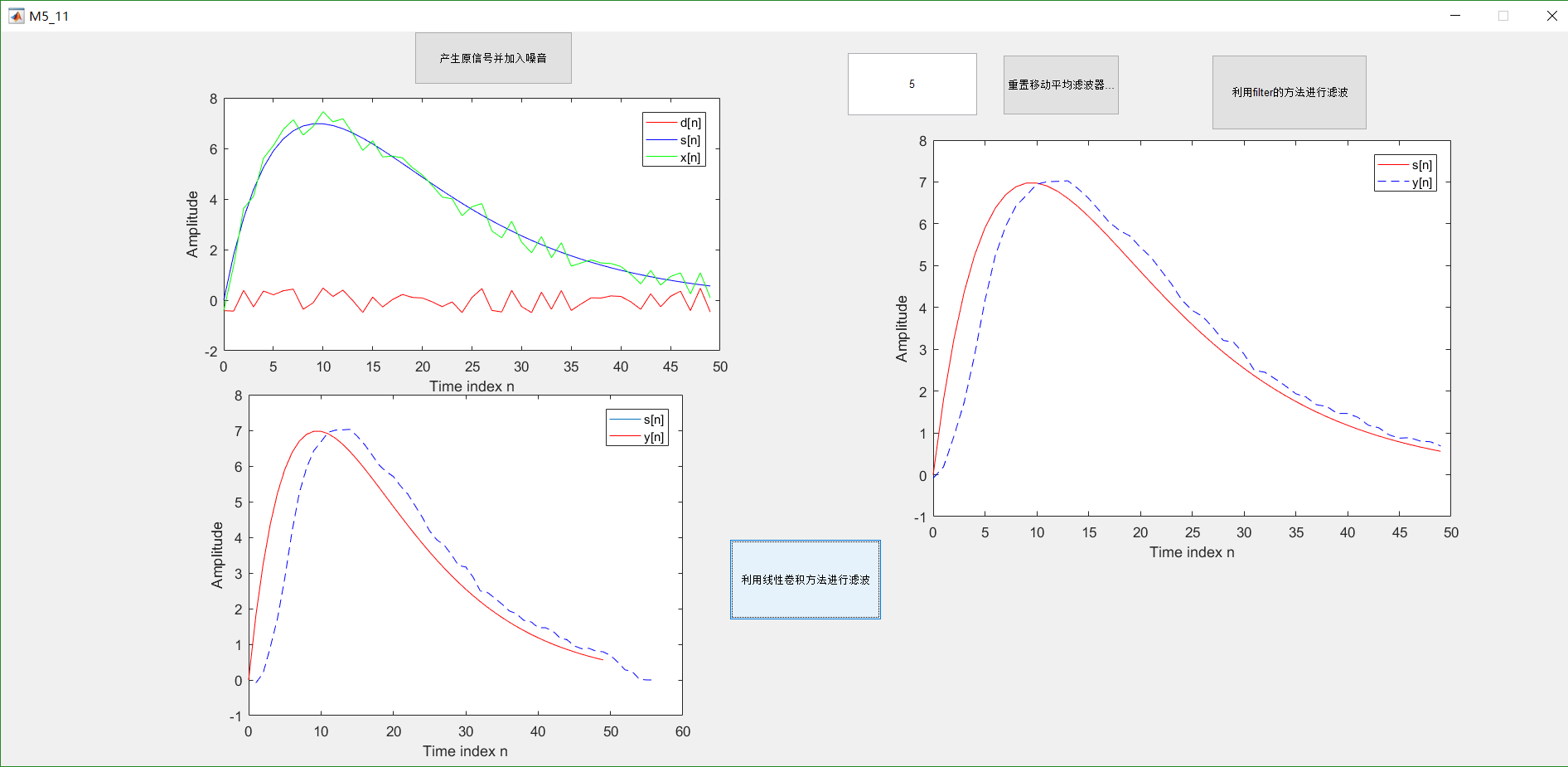
总体GUI效果

此次作业加入了整体的GUI设计，具体结果如下若干图所示：









附录1

M5\_2修改后代码

g = [3,2,-2,1,0,1];

g1 = [3-2i, 4-i, -2+3j, j, 0]

g2 = [cos(pi\*0/2),cos(pi\*1/2),cos(pi\*2/2),cos(pi\*3/2),cos(pi\*4/2)]

% n = length(g)

n1 = length(g1)

n2 = length(g2)

h = [-5,-1,3,-2,4,4];

h1 = [1-3j, -2-j, 2+2j, 3, -2+4j]

h2 = [1,3,9,27,81]

Y = fft(g).\*fft(h)

Y1 = fft(g1).\*fft(h1)

Y2 = fft(g2).\*fft(h2)

y = ifft(Y)

y1 = ifft(Y1)

y2 = ifft(Y2)

yy = circonv(g,h)

yy1 = circonv(g1,h1)

yy2 = circonv(g2,h2)

% t = 0:1:n-1

subplot(321)

stem(y)

xlabel('n')

ylabel('y[n]')

title('基于DFT的方法')

subplot(322)

stem(t,yy)

xlabel('n')

ylabel('y[n]')

title('利用circonv')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

subplot(323)

stem(y1)

xlabel('n')

ylabel('y1[n]')

title('基于DFT的方法')

subplot(324)

stem(yy1)

xlabel('n')

ylabel('y1[n]')

title('利用circonv')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

subplot(325)

stem(y2)

xlabel('n')

ylabel('y2[n]')

title('基于DFT的方法')

subplot(326)

stem(yy2)

xlabel('n')

ylabel('y2[n]')

title('利用circonv')

M5\_2 GUI代码

function varargout = M5\_2(varargin)

% M5\_2 MATLAB code for M5\_2.fig

% M5\_2, by itself, creates a new M5\_2 or raises the existing

% singleton\*.

%

% H = M5\_2 returns the handle to a new M5\_2 or the handle to

% the existing singleton\*.

%

% M5\_2('CALLBACK',hObject,eventData,handles,...) calls the local

% function named CALLBACK in M5\_2.M with the given input arguments.

%

% M5\_2('Property','Value',...) creates a new M5\_2 or raises the

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

% applied to the GUI before M5\_2\_OpeningFcn gets called. An

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to M5\_2\_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 M5\_2

% Last Modified by GUIDE v2.5 01-May-2017 23:24:40

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

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

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

'gui\_OpeningFcn', @M5\_2\_OpeningFcn, ...

'gui\_OutputFcn', @M5\_2\_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 M5\_2 is made visible.

function M5\_2\_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 M5\_2 (see VARARGIN)

global g g1 g2 h h1 h2

g = [3,2,-2,1,0,1];

g1 = [3-2i, 4-i, -2+3j, j, 0]

g2 = [cos(pi\*0/2),cos(pi\*1/2),cos(pi\*2/2),cos(pi\*3/2),cos(pi\*4/2)]

% n = length(g)

n1 = length(g1)

n2 = length(g2)

h = [-5,-1,3,-2,4,4];

h1 = [1-3j, -2-j, 2+2j, 3, -2+4j]

h2 = [1,3,9,27,81]

% Choose default command line output for M5\_2

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes M5\_2 wait for user response (see UIRESUME)

% uiwait(handles.figure1);

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

function varargout = M5\_2\_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 button press in pushbutton1.

function pushbutton1\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton1 (see GCBO)

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

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

global g h

Y = fft(g).\*fft(h)

y = ifft(Y)

yy = circonv(g,h)

% t = 0:1:n-1

axes(handles.axes1)

stem(y)

xlabel('n')

ylabel('y[n]')

title('基于DFT的方法')

axes(handles.axes2)

stem(yy)

xlabel('n')

ylabel('y[n]')

title('利用circonv')

% --- 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)

global g1 h1

Y = fft(g1).\*fft(h1)

y = ifft(Y)

yy = circonv(g1,h1)

% t = 0:1:n-1

axes(handles.axes1)

stem(y)

xlabel('n')

ylabel('y[n]')

title('基于DFT的方法')

axes(handles.axes2)

stem(yy)

xlabel('n')

ylabel('y[n]')

title('利用circonv')

% --- 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)

global g2 h2

Y = fft(g2).\*fft(h2)

y = ifft(Y)

yy = circonv(g2,h2)

% t = 0:1:n-1

axes(handles.axes1)

stem(y)

xlabel('n')

ylabel('y[n]')

title('基于DFT的方法')

axes(handles.axes2)

stem(yy)

xlabel('n')

ylabel('y[n]')

title('利用circonv')

M5\_10.m

n = 0:1:255

x = zeros(1,256)

for m = 0:1:255

x(1,m+1) = 0.1\*m\*exp(-0.03\*m)

end

noise\_fft = [zeros(1,50) 10\*rand(1,156)-10 zeros(1,50)]

noise = 5\*(ifft(noise\_fft,256)) % 利用随机数+ifft产生特定频率范围的随机信号

subplot(412)

plot(n,noise)

xlabel('n')

ylabel('noise')

title('高频噪音')

subplot(411)

plot(n,x)

xlabel('n')

ylabel('x[n]')

title('原信号')

subplot(413)

y = noise + x

plot(n,y)

xlabel('n')

ylabel('y[n]')

title('加入噪音的信号')

z = fft(y,256)

for m = 50:1:206

z(1,m) = 0 %滤波

end

recover = ifft(z,256)

subplot(414)

plot(n,recover)

xlabel('n')

ylabel('recover')

title('复原后的信号')

conv\_save.m

%%重叠保留法，具体见课本169页

function conv1 = overlapsave(x,h)

% x = ones(1,200) %用于测试

% for m = 1:1:200

% x(m) = 8\*rand(1)-4

% end

% % x = [2,-3,4,5,-6,7,8,-9,-10,11,-12,-13,-14]

% h = [1,-2,-3] %用于测试

%x的长度应远远大于h的长度

h = h'

len\_x = length(x);

len\_h = length(h);

M = len\_h

len\_conv = len\_x + len\_h -1

% temp = ceil(log2(len\_h));

% N = 2^(temp); %171页下部的M和N

N = M+2

num = floor(len\_x/M) +1 %对x切片的数量

x\_m = zeros(num,N) %对x切的片，每一行都是一个片

cir\_m = zeros(num,len\_conv) %x的每个切片和y[n]相卷，存入一个矩阵

%N点圆周卷积，而M<=N,故需要视情况对h进行补0操作

if(N > len\_h)

for i = len\_h+1:1:N

h(1,i) = 0;

end

end

%对x进行切片，每片长度为N

iN = 1; %第iN个切片

iM = 1; %切片中的第

% for m = 1:N:num\*N

x\_m(1,:) = [zeros(1,N-M) x(1,1:M)]

x\_m(num,:) = [x(1,len\_x-N+M-mod(len\_x,M)+1:len\_x) zeros(1,M-mod(len\_x,M)) ]

for iN = 2:1:num-1

for iM = 1:1:N

x\_m(iN,iM) = x(1,(iN-1)\*M+(iM-N+M))

end

% cir\_m(iN,1+(iN-1)\*(N-M+1):1+(iN-1)\*(N-M+1)+(N-1)) = circonv(x\_m(iN,:),h)

end

for iN=1:1:num

for iM=1:1:N

if 1+(iN-1)\*(N-M+1)+(N-1) < len\_conv

cir\_m(iN,1+(iN-1)\*(N-M+1):1+(iN-1)\*(N-M+1)+(N-1)) = circonv(x\_m(iN,:),h)

end

end

end

% circonv()

% aban = M-1 %开头不正确的需要被丢弃

%去掉cir\_m每一行的前(M-1)个元素

cir\_m\_del = cir\_m;

for m =1:1:num

cir\_m\_del(m,1+(m-1)\*(N-M+1):1+(m-1)\*(N-M+1)+(M-1-1)) = 0

end

%将cir\_m\_del合并成为一个行向量，可得到线性卷积序列

% conv = zeros(1,len\_conv)

% conv = zeros(1,len\_x)

conv2 = sum(cir\_m\_del)

conv1 = conv(x,h)

for iJ = 1:1:len\_conv-2

conv2(iJ) = conv2(iJ+2);

end

chapter5.m

%总的GUI代码

function varargout = chapter5(varargin)

% CHAPTER5 MATLAB code for chapter5.fig

% CHAPTER5, by itself, creates a new CHAPTER5 or raises the existing

% singleton\*.

%

% H = CHAPTER5 returns the handle to a new CHAPTER5 or the handle to

% the existing singleton\*.

%

% CHAPTER5('CALLBACK',hObject,eventData,handles,...) calls the local

% function named CALLBACK in CHAPTER5.M with the given input arguments.

%

% CHAPTER5('Property','Value',...) creates a new CHAPTER5 or raises the

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

% applied to the GUI before chapter5\_OpeningFcn gets called. An

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to chapter5\_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 chapter5

% Last Modified by GUIDE v2.5 08-May-2017 14:36:46

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

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

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

'gui\_OpeningFcn', @chapter5\_OpeningFcn, ...

'gui\_OutputFcn', @chapter5\_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 chapter5 is made visible.

function chapter5\_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 chapter5 (see VARARGIN)

% Choose default command line output for chapter5

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes chapter5 wait for user response (see UIRESUME)

% uiwait(handles.figure1);

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

function varargout = chapter5\_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 button press in pushbutton1.

function pushbutton1\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton1 (see GCBO)

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

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

run('M5\_2.m')

% --- 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)

run('M5\_10.m')

% --- 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)

run('M5\_11.m')