

Run的回调函数：

function Run\_Callback(hObject, eventdata, handles)

% hObject handle to Run (see GCBO)

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

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

global x;

global y;global y2;

global T;

global Rxy;

Rxytemp = ifft(fft(y).\*conj(fft(y2)));

Rxy = fftshift(Rxytemp);

plot(handles.plot,x,y,'b','linewidth',1);

title(handles.plot,'y1');

xlim(handles.plot,[0,T]);

grid on;

plot(handles.plot\_2,x,y2,'b','linewidth',1);

title(handles.plot\_2,'y2');

xlim(handles.plot\_2,[0,T]);

grid on;

plot(handles.plot\_3,x,Rxy,'b','linewidth',1);

title(handles.plot\_3,'Rxy');

xlim(handles.plot\_3,[0,T]);

grid on;

这种情况对应的算法为：

1. 将信号y和y2分别进行傅里叶变换；
2. 将信号y的傅里叶变换与信号y2傅里叶变换的共轭相乘；
3. 对上述计算结果进行傅里叶反变换
4. 将零频点移到频谱中间

推导：



Self的回调函数：

function self\_xcorr\_Callback(hObject, eventdata, handles)

% hObject handle to self\_xcorr (see GCBO)

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

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

global x;

global y;global y2;

global T;

global Rxy;

Rxy = xcorr(y,y2);

N1=length(Rxy);

x1=linspace(-T,T,N1);

plot(handles.plot,x,y,'b','linewidth',1);

title(handles.plot,'y1');

xlim(handles.plot,[0,T]);

plot(handles.plot\_2,x,y2,'b','linewidth',1);

title(handles.plot\_2,'y2');

xlim(handles.plot\_2,[0,T]);

plot(handles.plot\_3,x1,Rxy,'b','linewidth',1);

title(handles.plot\_3,'Rxy');

xlim(handles.plot\_3,[-T,T]);

grid on;

self按钮直接调用MATLAB的API xcorr()

Loop回调函数：

global x;

global y; global y2;

global N;global T;

for k=1:N/2

z0=0; z1(k)=0; z2(k)=0;

for j=1:N

z0=z0+y(j)\*y2(j);

end

z0=z0/N;

for j=1:N-k

z1(k)=z1(k)+y(j)\*y2(j+k);

end

z1(k)=z1(k)/(N-k);

for j=k:N

z2(k)=z2(k)+y(j)\*y2(j-k+1);

end

z2(k)=z2(k)/(N-k);

end

zz=[z2 z0 z1];

N2=length(zz); x2=linspace(-T/2,T/2,N2);

plot(handles.plot,x,y,'b','linewidth',1);

title(handles.plot,'y1');

xlim(handles.plot,[0,T]);

grid on;

plot(handles.plot\_2,x,y2,'b','linewidth',1);

title(handles.plot\_2,'y2');

xlim(handles.plot\_2,[0,T]);

grid on;

plot(handles.plot\_3,x2,zz,'b','linewidth',1);

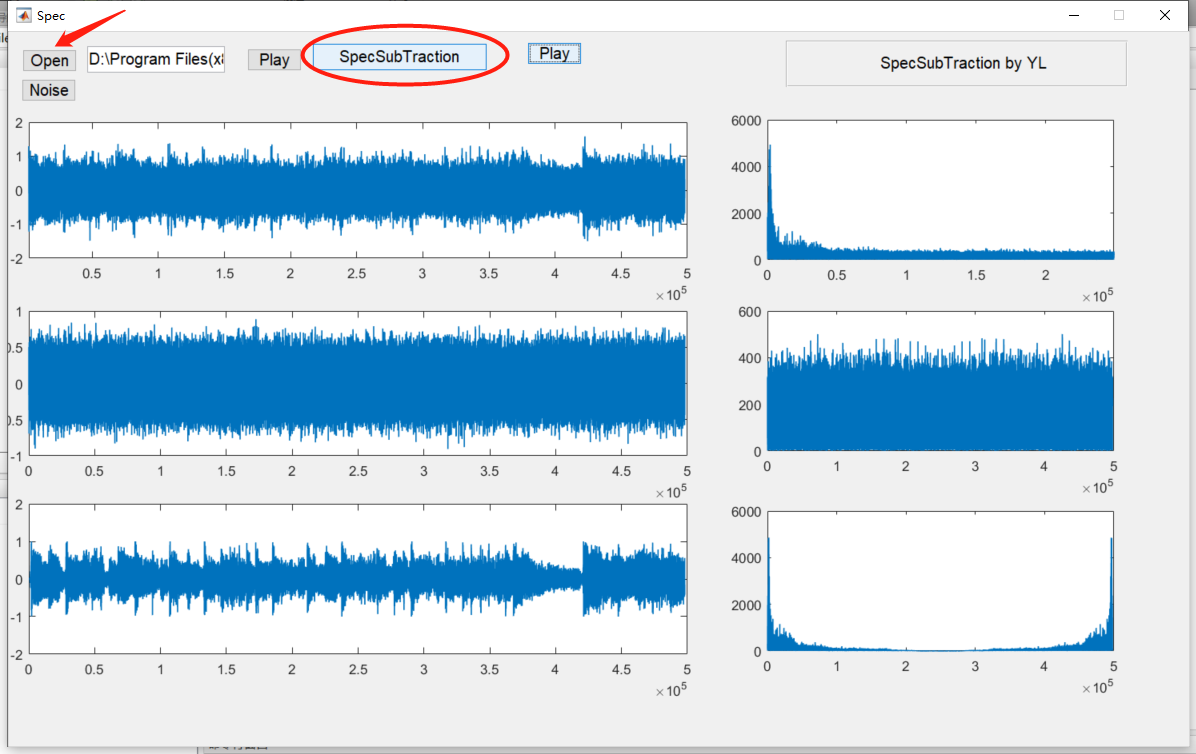
title(handles.plot\_3,'Rxy');

grid on;

使用两重循环，不将信号转换的频域，公式：

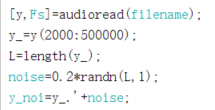


音频降噪：



实现方法：

1. 在下载的音频上加上随机噪声（下载的噪声音频频率有问题）



1. 降噪方法：将音频信号进行傅里叶变换，将噪声信号进行傅里叶变换，然后相减

% --- Executes on button press in spec.

function spec\_Callback(hObject, eventdata, handles)

% hObject handle to spec (see GCBO)

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

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

global y\_noi;

global fftnoise;

global music;

ffty = fft(y\_noi);

fftmusic = ffty - fftnoise;

plot(handles.axes6,abs(fftmusic),'linewidth',1);

music = ifft(fftmusic);

plot(handles.Mu,music,'linewidth',1);

公式：

