**Matlab仿真与工程应用**

**课程设计报告**

项目名称：**基于matlab simulink的信号调制解调GUI设计**

专业：**电子科学与技术**

学号：**1711143**

姓名：**王忠汉**

指导教师：**程如岐**

目 录

第1章 不同调制解调方式的simulink实现

1.1 AM调制解调

1.1.1 AM相干调制解调

1.1.2 AM包络调制解调

1.2 DSB调制解调

1.3 SSB调制解调

1.3.1 SSB滤波法

1.3.2 SSB相移法

1.3.3 SSB解调

第2章 调制解调GUI界面的设计

2.1 调制解调方式的选择

2.1.1 Listbox的设置

2.1.2 Pushbutton的设置

2.1.3 效果展示

2.2 坐标轴与图形绘制

2.2.1 坐标轴的设置

2.2.3 绘制曲线

2.2.3 效果展示

2.3 信号频率的设置

2.3.1 slider的设置

2.3.2 可编辑文本框的设置

2.3.3 效果展示

2.4 频谱曲线的绘制

2.5 功率谱曲线的绘制

第3章 GUI总体效果

第4章 模型优缺点与实验反思

4.1 模型优点

4.2 模型缺点与改进

4.3 实验反思

附录A Simulink程序框图与参数设置

附录B GUI界面

**摘 要**

信号的调制就是用基带信号去控制载波信号的某个或几个参量的变化，将信息荷载在其上形成已调信号传输，而解调是调制的反过程，通过具体的方法从已调信号的参量变化中将恢复原始的基带信号。本实验采用的调制解调方法有：AM调制解调（AM相干、AM包络）、DSB调制解调、USB滤波调制解调、USB相移调制解调、LSB滤波调制解调、LSB相移调制解调。

本实验首先使用simulink搭建不同方式下的正弦信源调制解调模型，并使用scope测量调制信号、解调信号和信源信号。然后本文设计了可在不同调制解调方式之间切换的GUI界面，通过调用已搭建完成的相关simulink模型在坐标轴上显示出对应的曲线。GUI界面同时允许用户对基波和载波频率进行调节，并为四个环节的曲线增设频谱和功率谱分析选项，以供用户进一步研究。

本文在最后展示了默认条件下不同调制解调方式下的正弦信源调制解调信号及其频谱和功率谱，并针对本次matlab GUI设计加以反思，给出了优缺点。

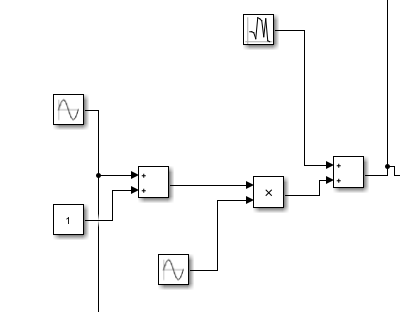
关键词：调制解调 simulink GUI

第1章 不同调制解调方式的simulink实现

**1.1 AM调制解调**

AM调制为幅度调制，指载波信号的幅度随信源信号的波形比例变化而载波频率保持不变的一种电子通信调制方式。具体的调制过程为：信源信号与直流信号叠加，生成的信号与载波信号相乘，即可生成调制信号。为确保真实性，本实验在调制信号的基础上叠加了平均值为0，方差为0.01的随机噪声。

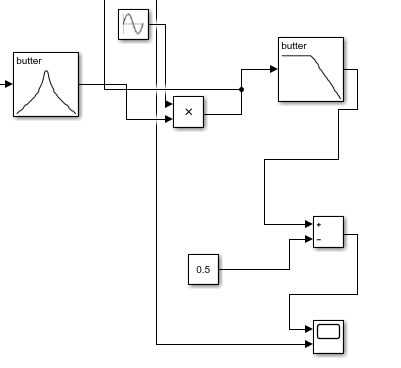
AM调制的simulink模型如下：其中载波频率为100Hz。



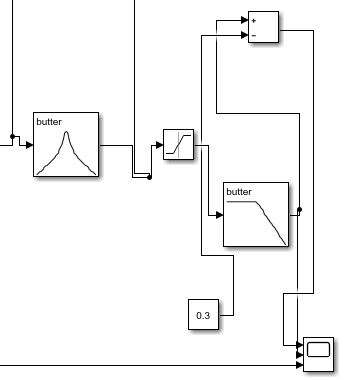
**1.1.1 AM相干解调**

AM相干解调指的是通过设置一个与载波信号同频同相的解调信号进行解调的方式，适用于所有线性调制信号。解调过程如下：通过带通滤波器后，设置与载波信号频率完全相同的解调信号，与已调信号相乘，然后经低通滤波器取出混合信号中的低频信号，为确保与信源信号近似相同，本实验在解调信号的基础上增加了适当的直流分量，使最终的信号中轴线大致为x轴。

AM相干解调的simulink模型如下：其中输入信号为已调信号，正弦信号与载波信号完全相同。

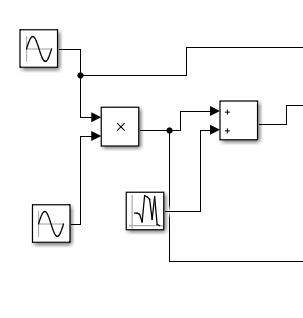
**1.1.2 AM包络检波**

包络检波为非相干解调，不需要增设与载波信号同频同相的信号，可直接从已调信号中提取信源信号。解调过程如下：信号通过带通滤波器之后，利用saturation的包络检波模块过滤掉下边带，然后经过低通滤波器取出低频信号，最后叠加直流分量进行最后的调整。

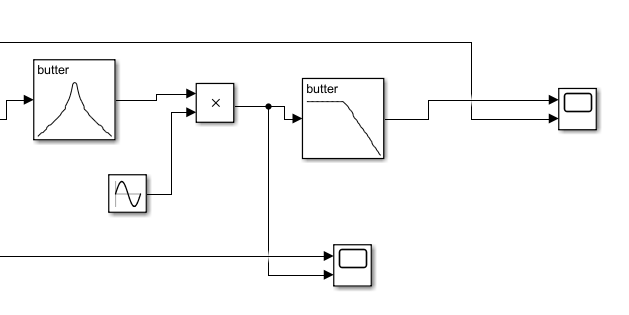
AM包络检波的simulink模型如下：其中输入信号为已调信号，saturation为包络检波模块。

**1.2 DSB调制解调**

与AM调制解调不同，DSB为抑制载波双边带信号，不含直流分量，调制效率更高，具体调制过程如下：首先信源信号与载波信号直接相乘，而不同于AM调制的叠加直流分量，产生后的混合信号叠加均值为0，方差为0.01的随机噪声。

DSB调制的simulink模型如下：其中信源信号频率为10Hz，载波信号频率为100Hz。

DSB解调方式采用相干解调，与AM相干解调类似：信号经过带通滤波器之后，与一个等同于载波信号的信号相乘，经过低通滤波器之后可直接得到解调信号，由于DSB信号不含直流分量，因此滤波后的信号往往无需直流处理。

DSB解调的simulink模型如下：其中正弦信号完全等同于载波信号。

**1.3 SSB调制解调**

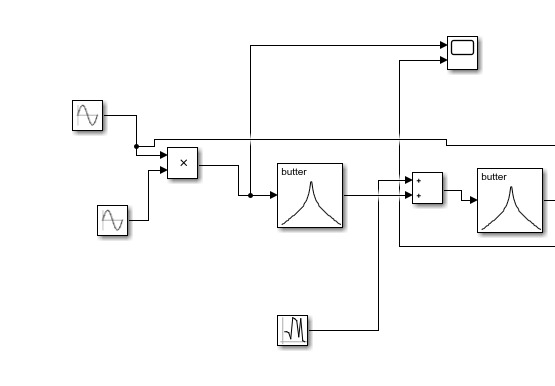
SSB为单边带调制，由于两个边带中只有一个包含了所有的频谱信息，因此相较于DSB调制解调其更加节省功率，可认为是DSB调制的简化版。

DSB信号转变为SSB信号需要滤掉其中一个边带，滤掉上边带保留下边带即为LSB，滤掉下边带保留上边带即为USB；根据调制方式不同又可以分为滤波法和相移法。

**1.3.1 SSB滤波法**

由于USB与LSB仅仅为边带保留的不同，因此这里不作区分。SSB滤波法的调制过程如下：信源信号与载波信号相乘之后，经过带通滤波器滤波，叠加均值为0，方差为0.01的随机噪声后再次经过带通滤波器，得到已调信号，由于涉及多次滤波，因此得名滤波法。

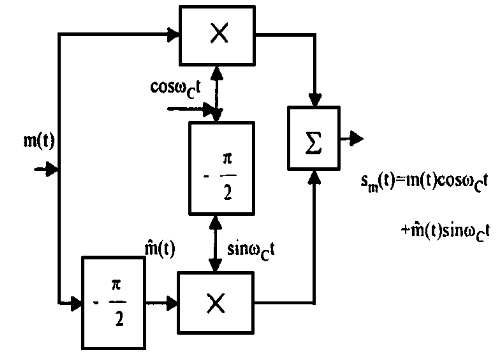
SSB滤波法调制的simulink模型如下：



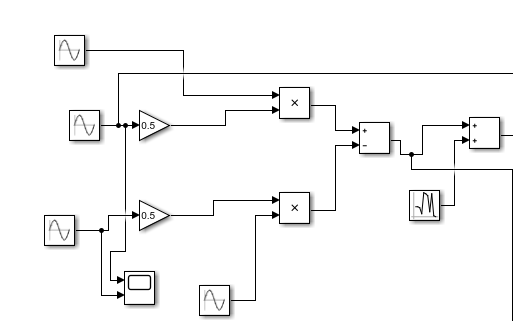
**1.3.2 SSB相移法**

SSB相移法模拟了一个与信源信号相同，但相位有所偏差的信号，两个信号与各自的载波信号相乘后进行减法运算，形成的混合信号再叠加随机噪声，滤波得调制信号。

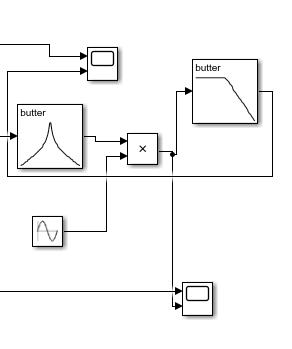
原理图大致如下：



SSB相移法的simulink模型如下：其中信源信号与模拟信号频率为10Hz，载波信号频率为100Hz。



**1.3.3 SSB解调**

SSB解调为相干解调，在已调信号的基础上叠加与载波信号同频同相的信号，滤波后得到解调信号。SSB解调的simulink模型如下：

第2章 调制解调GUI界面的设计

**2.1 调制解调方式的选择**

本实验使用Listbox、Pushbutton、可编辑文本框等GUI组件来实现调制解调方式的选择。

**2.1.1 Listbox的设置**

Listbox用来罗列供选择的调制解调方式，所含String依次为：’AM相干’ ’AM包络’ ‘DSB’ ‘USB滤波’ ‘USB相移’ ‘LSB滤波’ ‘LSB相移’，对应的被选中时value为1、2、3、4、5、6、7。

**2.1.2 Pushbutton 的设置**

本实验设置了两个pushbutton，分别用于调制与解调方式的选择，当Listbox对应的方式被选中时，按下相应的pushbutton可以在相应的调制解调文本框中显示对应的方式，对应的’String’分别为’选择调制方式’ ‘选择解调方式’。

为实现该功能，callback函数中增设了如下的程序（仅以选择调制方式为例，选择解调方式类似）：

index=get(handles.listbox1,'value');

%index为Listbox被选中时对应的value，可用于鉴别被选中的项目

str=get(handles.listbox1,'string');

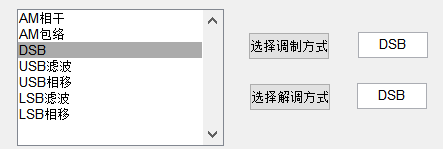
%str为Listbox的’String’数组

set(handles.edit7,'String',str{index});

%使用set函数使可编辑文本框edit7显示相应的String，使用value定为str中的选中项

**2.1.3 效果展示**

该阶段达成的效果如下：

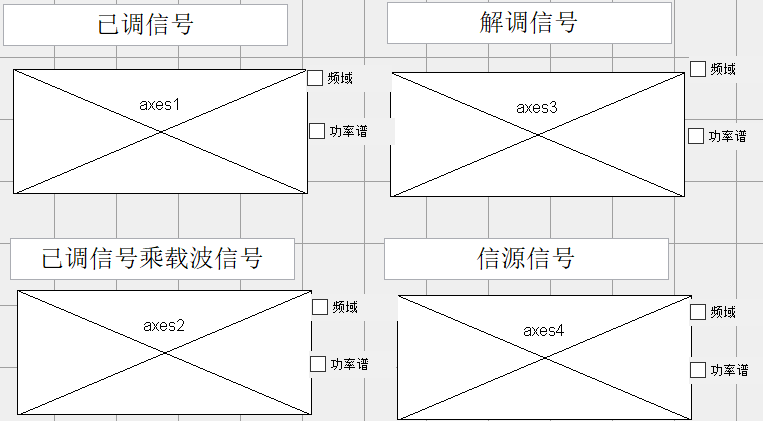


可以看出选中DSB之后，点击对应的pushbutton，可以在相应的文本框中显示选择的方式，调制与解调方式不一定相同，也可以选取不同的方式。

**2.2 坐标轴与图形绘制**

**2.2.1 坐标轴的设置**

本实验设置四个坐标轴，两个用于调制效果展示，两个用于解调效果展示，调整阶段主要表现调制信号以及调制信号与解调载波信号的乘积（由于AM包络检波不含解调载波信号，因此选取带通滤波后的已调信号取代）。

坐标轴下方配备有相应的文本框，以方便对照，整体效果如下：

**2.2.2 绘制曲线**

本实验设置两个pushbutton分别用于开始调制与解调，达到的效果是点击相应的pushbutton之后，相应的坐标轴上生成曲线。

为实现本目的，callback函数增加如下程序（以开始调制为例，开始解调类似）：

obj=get(handles.edit7,'String');

%obj为对应的文本框显示的字符

switch obj

%下面是针对不同的调制方式下的分支结构

case 'AM相干'

%选择了AM相干调制方式

open('AMxianggan');sim('AMxianggan');

%打开对应的simulink模型，并运行

set(handles.edit5,'String','已调信号乘载波信号');

%改变对应坐标轴下方的文本框显示字符

axes(handles.axes1);

%选定生成曲线的坐标轴

plot(AMxgcj.time,AMxgcj.signals(1).values);

%AMxgcj为AM调制阶段的示波器数据，signals（1）为调制信号，开始在坐标轴1上生成对应的曲线

axes(handles.axes2);

plot(AMxgcj.time,AMxgcj.signals(2).values);

%signals（2）为已调信号乘载波信号，并在坐标轴2上生成曲线

case 'AM包络'

%选择了AM包络方式，下同AM相干，不再解释

open('AMbaoluo');sim('AMbaoluo');

set(handles.edit5,'String','带通滤波后的已调信号');

axes(handles.axes1);

plot(AMbltz.time,AMbltz.signals(1).values\*gain);

axes(handles.axes2);

plot(AMbltz.time,AMbltz.signals(2).values\*gain);

case 'DSB'

open('DSB');sim('DSB');

set(handles.edit5,'String','已调信号乘载波信号');

axes(handles.axes1);

plot(DSBtiaozhi.time,DSBtiaozhi.signals(1).values\*gain);

axes(handles.axes2);

plot(DSBtiaozhi.time,DSBtiaozhi.signals(2).values\*gain);

case 'USB滤波'

open('USBlvbo');sim('USBlvbo');

set(handles.edit5,'String','已调信号乘载波信号');

axes(handles.axes1);

plot(USBlbtz.time,USBlbtz.signals(1).values\*gain);

axes(handles.axes2);

plot(USBlbtz.time,USBlbtz.signals(2).values\*gain);

case 'USB相移'

open('USBxiangyi');sim('USBxiangyi');

set(handles.edit5,'String','已调信号乘载波信号');

axes(handles.axes1);

plot(USB23.time,USB23.signals(1).values\*gain);

axes(handles.axes2);

plot(USB23.time,USB23.signals(2).values\*gain);

case 'LSB滤波'

open('LSBlvbo');sim('LSBlvbo');

set(handles.edit5,'String','已调信号乘载波信号');

axes(handles.axes1);

plot(LSBlbtz.time,LSBlbtz.signals(1).values\*gain);

axes(handles.axes2);

plot(LSBlbtz.time,LSBlbtz.signals(2).values\*gain);

case 'LSB相移'

open('LSBxiangyi');sim('LSBxiangyi');

set(handles.edit5,'String','已调信号乘载波信号');

axes(handles.axes1);

plot(LSB3.time,LSB3.signals(1).values\*gain);

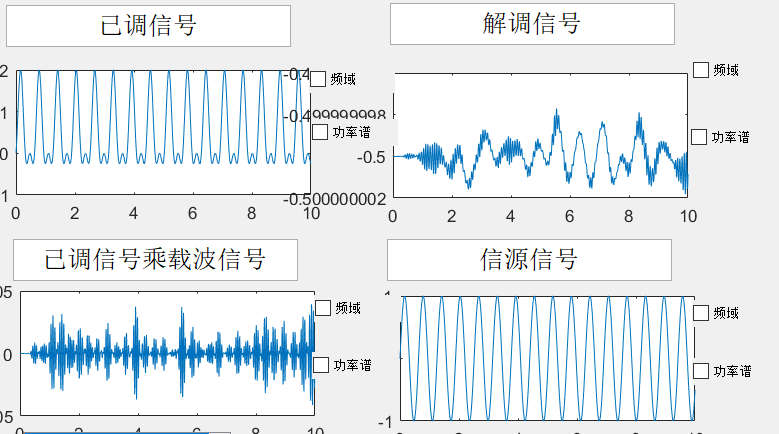
axes(handles.axes2);

plot(LSB3.time,LSB3.signals(3).values\*gain);

end

**2.2.3 效果展示**

该阶段达成的效果如下：



选中特定调制解调方式后，坐标轴会生成对应的曲线。

**2.3 信号频率的设置**

前面的实验信源幅度默认为10V，在此环节增设一个slider和一个checkbox，slider可连续调节信源信号的幅度，同时设置了可编辑文本框用于指定特定的信源幅度。

**2.3.1 slider的设置**

本实验设置一个slider，设定min为0，max为100，ListboxTop为100，以确保信源幅度可以在0至100之间连续变化。在未勾选checkbox的前提下，为确保调节slider之后，可编辑文本框可显示对应的信源幅度，在slider的callback中增设如下程序：

amplitude1=get(handles.slider5,'value');

set(handles.edit11,'String',num2str(amplitude1));

assignin('base','xy1',amplitude1);

%使用assigin函数设置信源信号的频率，xy1为simulink模型中的信源频率变量

amplitude2=get(handles.slider6,'value');

set(handles.edit12,'String',num2str(amplitude2));

assignin('base','zb1',amplitude2);

%使用assigin函数设置载波信号的频率，zb1为simulink模型中的载波频率变量

**2.3.2 可编辑文本框的设置**

有时我们需要特定的信源幅度，此时使用slider调节比较麻烦，因此我们设置在文本框中输入数字之后，可以使slider自定滑到对应位置，同时设定信源信号的幅度，callback函数增设如下程序：

check=get(handles.checkbox1,'value');

%获取checkbox的勾选情况

if check==0

%若未勾选

strr=get(handles.edit11,'String');

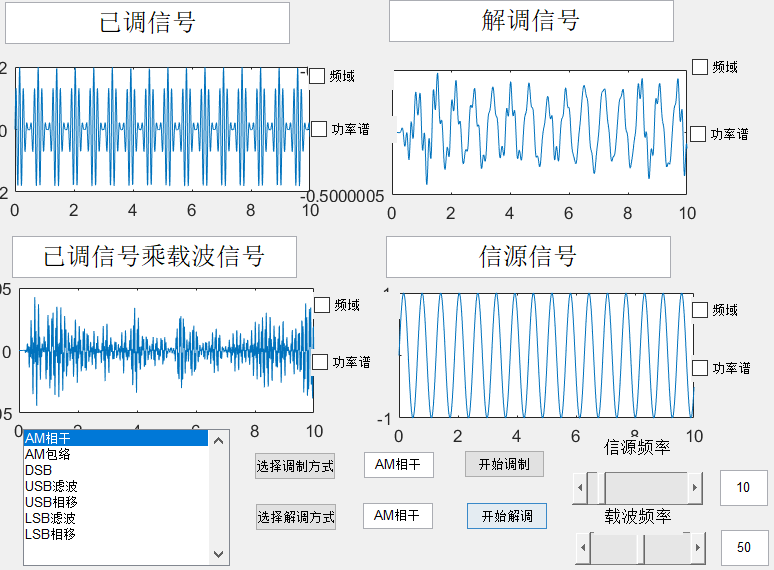
%获取可编辑文本框中的输入字符

set(handles.slider5,'value',str2num(strr));

%将slider的value设置为对应的数字

end

**2.3.3 效果展示**

如图所示，手动输入载波频率50Hz，或者滑动slider至50Hz后，调制与解调的效果如图。

**2.4 频谱曲线的绘制**

本实验采用快速傅里叶变换（FFT）对示波器信号进行处理，从而将时域曲线转变为频域曲线。在GUI界面上，用户可通过勾选频域复选框进行频域分析，设置该复选框选中时自动绘制曲线。

下面以第一个坐标轴的复选框设置为例展开说明：

function checkbox2\_Callback(hObject, eventdata, handles)

check=get(handles.checkbox2,'Value');%复选框选中情况

if check==1

obj=get(handles.edit7,'String');

switch obj

case 'AM相干'

open('AMxianggan');sim('AMxianggan');

axes(handles.axes1);

f=fft(AMxgcj.signals(1).values);%对时域信息FFT

yy=abs(f);%对结果的复数取模

plot(yy,'r-');%绘制曲线

case 'AM包络'

open('AMbaoluo');sim('AMbaoluo');

axes(handles.axes1);

f=fft(AMbltz.signals(1).values);

yy=abs(f);

plot(yy,'r-');

case 'DSB'

open('DSB');sim('DSB');

axes(handles.axes1);

f=fft(DSBtiaozhi.signals(1).values);

yy=abs(f);

plot(yy,'r-');

case 'USB滤波'

open('USBlvbo');sim('USBlvbo');

axes(handles.axes1);

f=fft(USBlbtz.signals(1).values);

yy=abs(f);

plot(yy,'r-');

case 'USB相移'

open('USBxiangyi');sim('USBxiangyi');

axes(handles.axes1);

f=fft(USB23.signals(1).values);

yy=abs(f);

plot(yy,'r-');

case 'LSB滤波'

open('LSBlvbo');sim('LSBlvbo');

axes(handles.axes1);

f=fft(LSBlbtz.signals(1).values);

yy=abs(f);

plot(yy,'r-');

case 'LSB相移'

open('LSBxiangyi');sim('LSBxiangyi');

axes(handles.axes1);

f=fft(LSB3.signals(1).values);

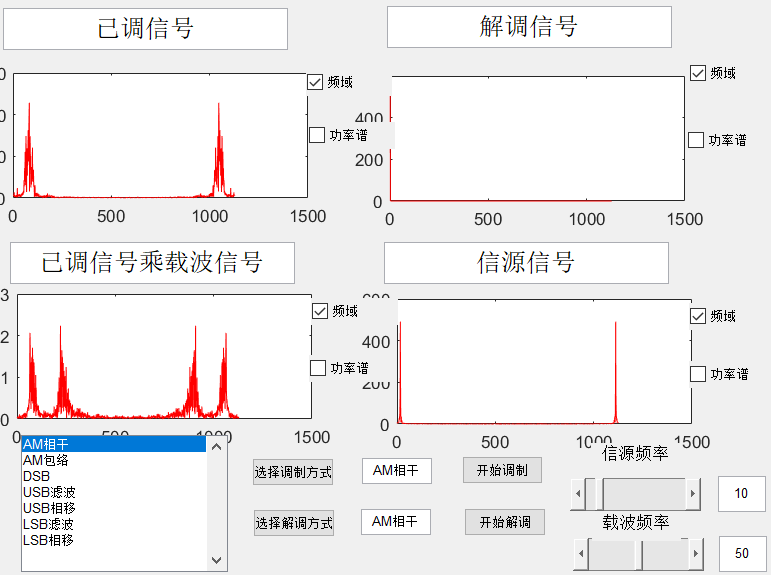
yy=abs(f);

plot(yy,'r-');

end

end

下面展示的是AM相干调节时四个环节频域分析：



**2.5 功率谱曲线的绘制**

本实验在时域信息的基础上，构建窗函数，并利用周期图法产生功率谱信息，以便用户进行功率谱分析。在GUI界面上，用户可通过勾选功率谱复选框进行频域分析，设置该复选框选中时自动绘制曲线。

下面以第一个坐标轴的复选框设置为例展开说明：

function checkbox6\_Callback(hObject, eventdata, handles)

% hObject handle to checkbox6 (see GCBO)

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

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

check2=get(handles.checkbox6,'Value');

if check2==1

nfft=1024;

Fs=1000;

obj=get(handles.edit7,'String');

switch obj

case 'AM相干'

open('AMxianggan');sim('AMxianggan');

window=boxcar(length(AMxgcj.signals(1).values));

%构建窗函数 [pxx,f]=periodogram(AMxgcj.signals(1).values,window,nfft,Fs);

%利用周期图法求解功率谱

axes(handles.axes1);

plot(f,10\*log10(pxx),'g-');

case 'AM包络'

open('AMbaoluo');sim('AMbaoluo');

window=boxcar(length(AMbltz.signals(1).values));

[pxx,f]=periodogram(AMbltz.signals(1).values,window,nfft,Fs);

axes(handles.axes1);

plot(f,10\*log10(pxx),'g-');

case 'DSB'

open('DSB');sim('DSB');

window=boxcar(length(DSBtiaozhi.signals(1).values));

[pxx,f]=periodogram(DSBtiaozhi.signals(1).values,window,nfft,Fs);

axes(handles.axes1);

plot(f,10\*log10(pxx),'g-');

case 'USB滤波'

open('USBlvbo');sim('USBlvbo');

window=boxcar(length(USBlbtz.signals(1).values));

[pxx,f]=periodogram(USBlbtz.signals(1).values,window,nfft,Fs);

axes(handles.axes1);

plot(f,10\*log10(pxx),'g-');

case 'USB相移'

open('USBxiangyi');sim('USBxiangyi');

window=boxcar(length(USB23.signals(1).values));

[pxx,f]=periodogram(USB23.signals(1).values,window,nfft,Fs);

axes(handles.axes1);

plot(f,10\*log10(pxx),'g-');

case 'LSB滤波'

open('LSBlvbo');sim('LSBlvbo');

window=boxcar(length(LSBlbtz.signals(1).values));

[pxx,f]=periodogram(LSBlbtz.signals(1).values,window,nfft,Fs);

axes(handles.axes1);

plot(f,10\*log10(pxx),'g-');

case 'LSB相移'

open('LSBxiangyi');sim('LSBxiangyi');

window=boxcar(length(LSB3.signals(1).values));

[pxx,f]=periodogram(LSB3.signals(1).values,window,nfft,Fs);

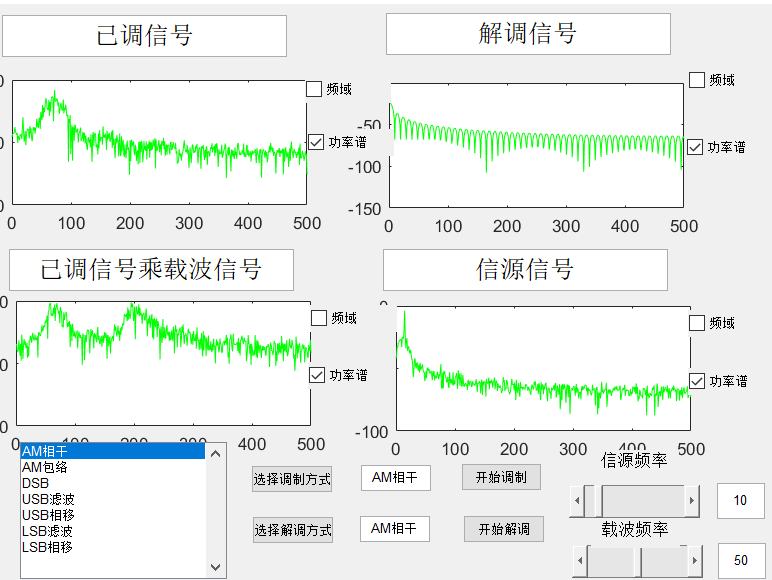
axes(handles.axes1);

plot(f,10\*log10(pxx),'g-');

end

End

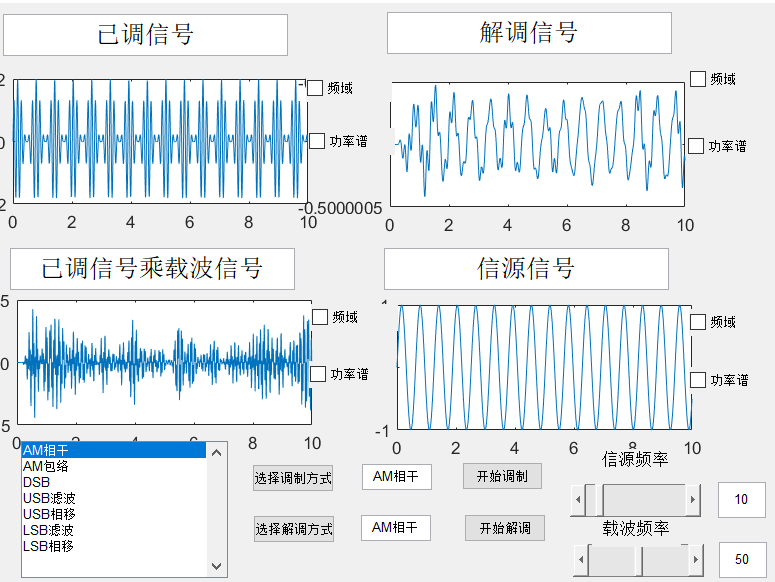
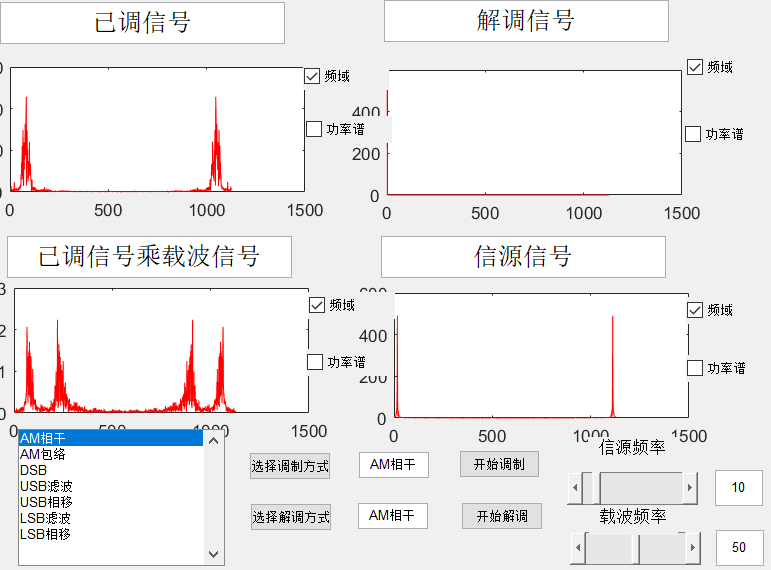
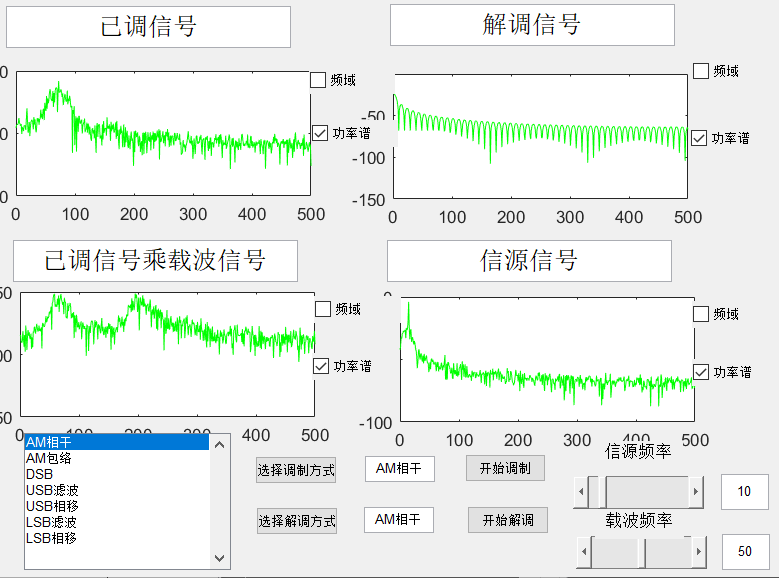
下图展示的是AM相干调制时的四个环节功率谱分析：

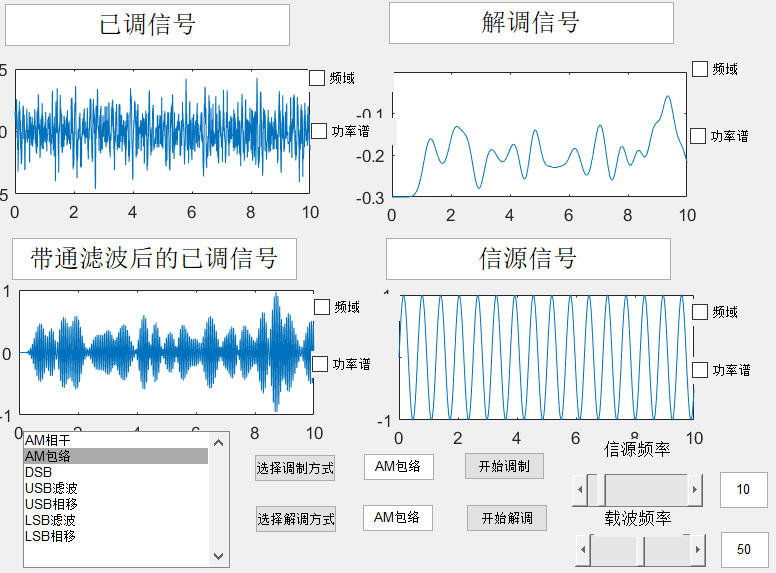
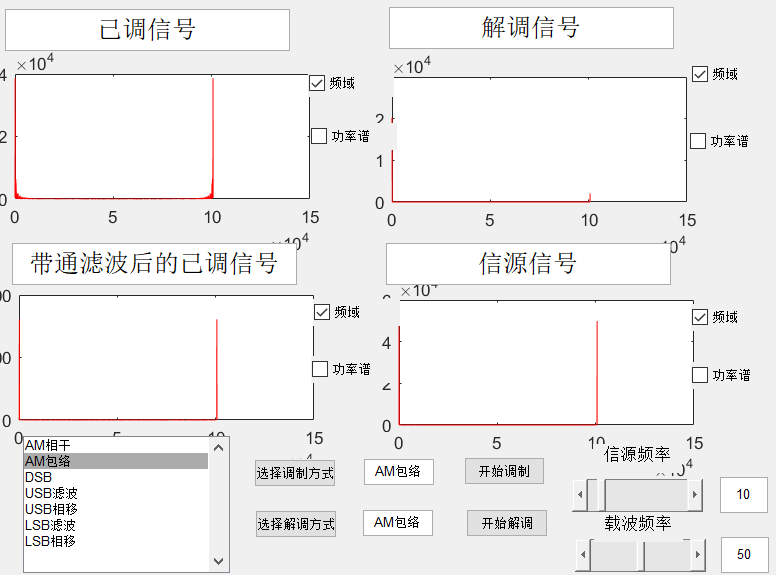


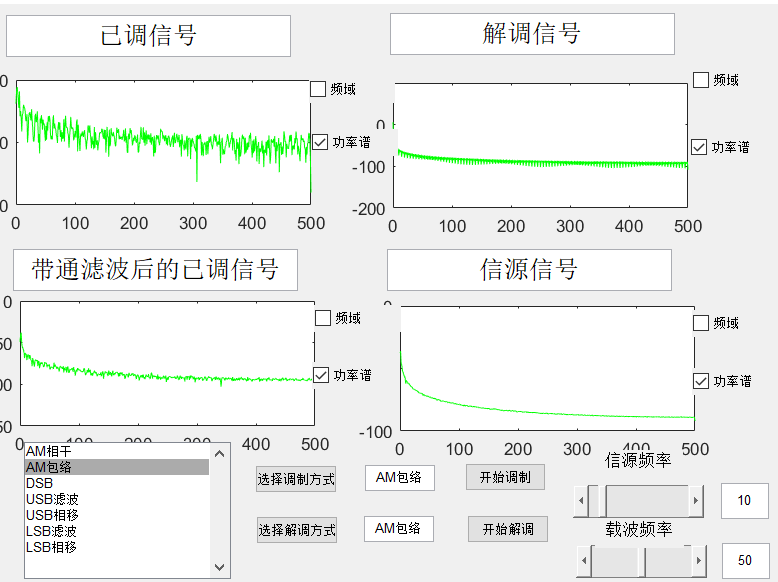
第3章 GUI总体效果

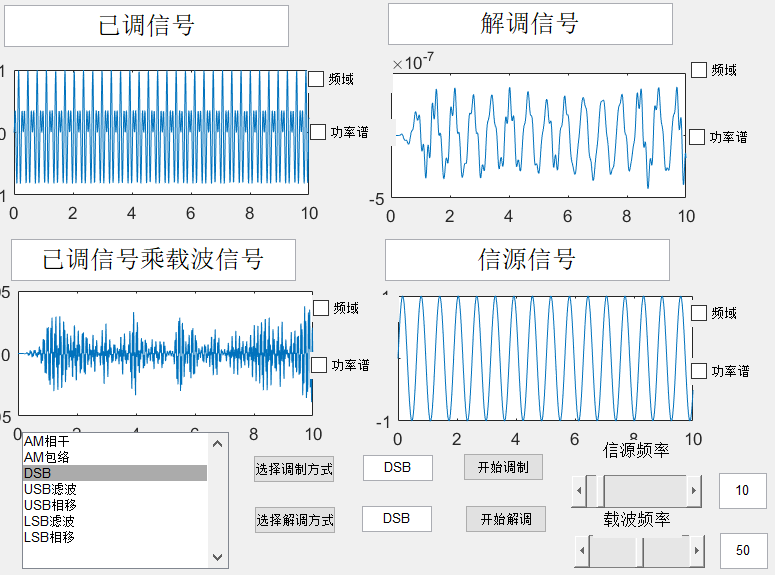
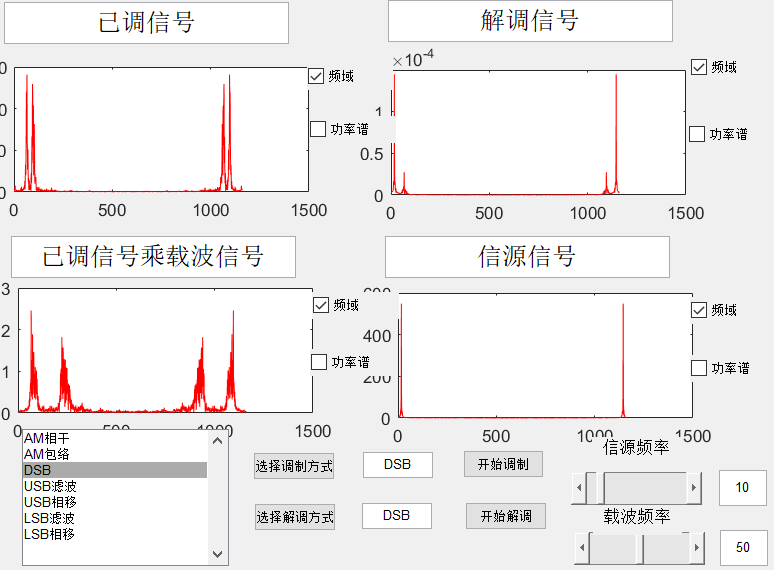
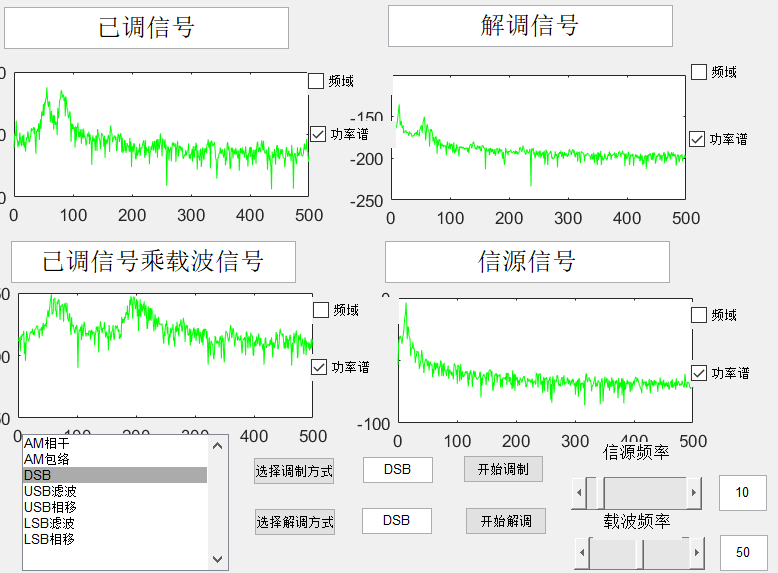
设置基波频率10Hz，载波频率50Hz，分别对AM相干调制、AM包络调制、DSB调制、USB滤波调制和USB相移调制进行分析（LSB调制与USB调制基本相同）。

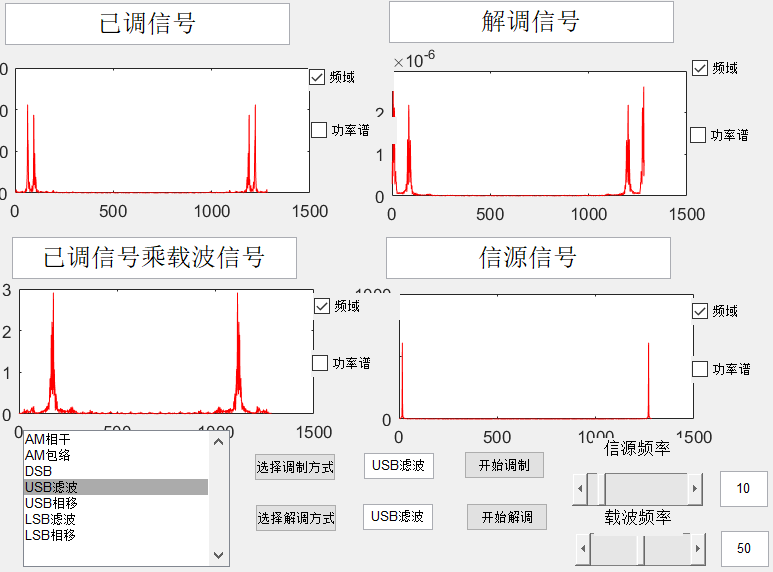
AM相干调制：

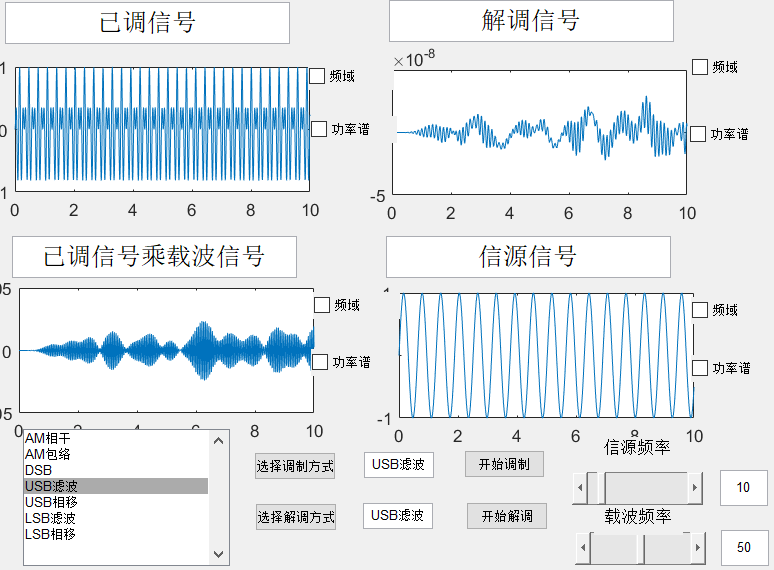
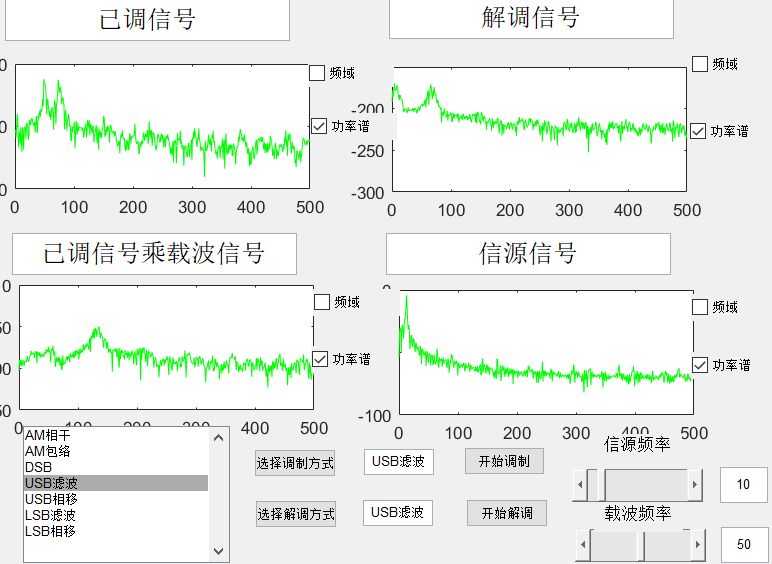


AM包络调制：

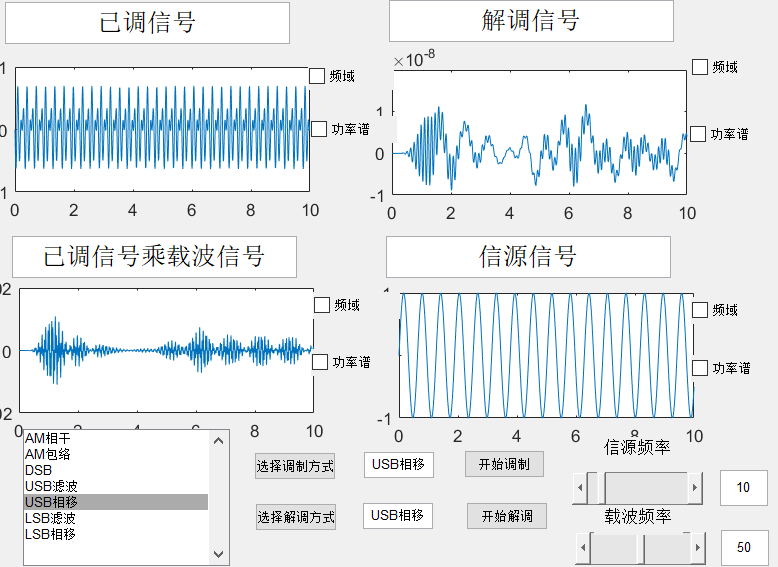
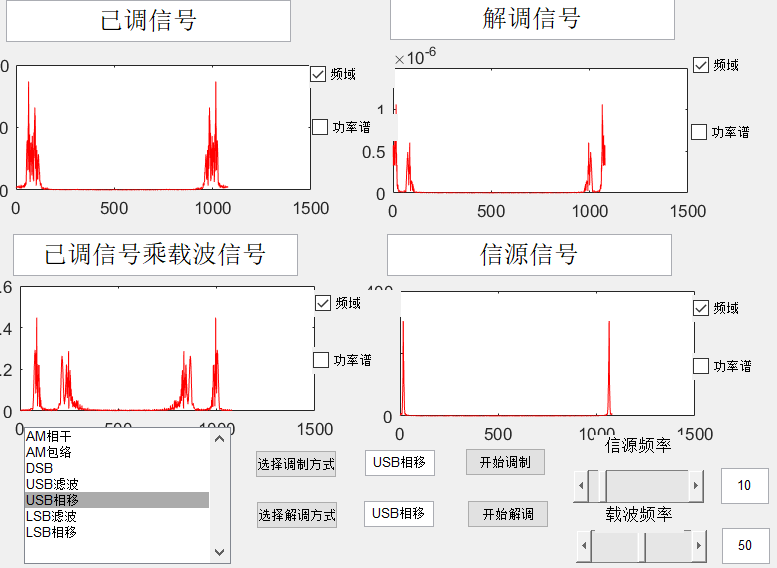
DSB调制：

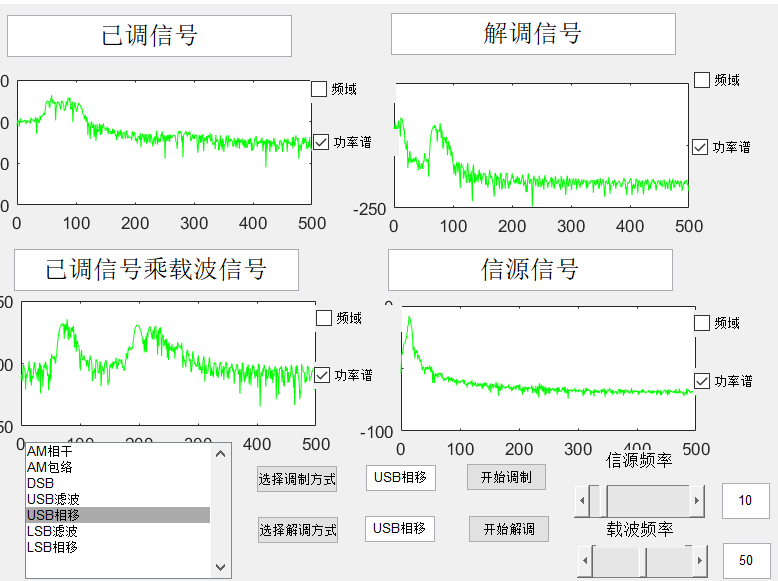


USB滤波调制：



USB相移调制：





第4章 模型优缺点与实验反思

**4.1 模型优点**

本模型选用了simulink后台工作的方式，基本刻画了正弦信源在不同的调制解调方式下的信号波形变化，比较全面、具体，前端则采用简洁直观的GUI界面，方便调节与仿真效果的呈现。

在simulink后台工作模块中，带通滤波器、低通滤波器以及随机噪声的参数调节较为恰当，基本保证了波形的合理真实，同时善用多个scope进行阶段信号波形的测量，将实验结果分层次展现，更体现了信息传递的真实自然过程。

在GUI前端设计中，使用Listbox清晰明确地展现了不同调制方式，并使用多个可编辑文本框与静态文本框对信息加以说明，两组坐标轴将调试过程与解调过程分开展示，便于分析与改正。Slider的使用以及文本框的直接赋值则大大加强了对信源信号幅度改变的灵活性，加大了模型使用的舒适度。同时，GUI给出了每一个时域曲线的频谱分析和功率谱分析，加大了研究深度与仿真能力。

**4.2 模型缺点与改进**

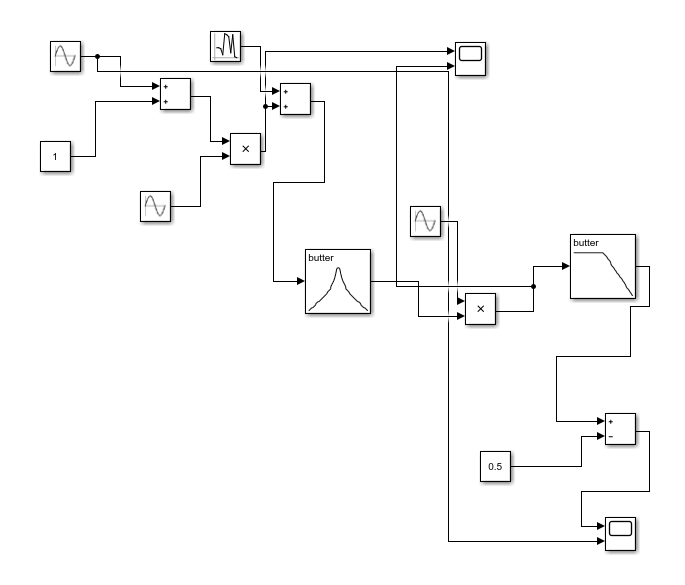
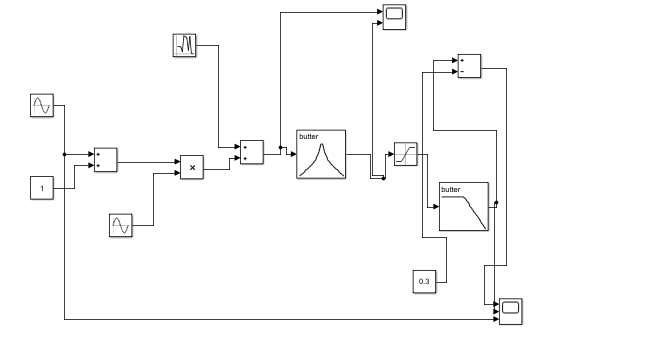
本实验仍存在一些缺点，以速度问题最为突出。由于本模型牵涉到simulink、GUI以及函数的参数传递与相互嵌套，因此初始化速度极慢，往往需要提前打开所有子程序之后再进行演示，同时增加了所需文件的个数，此模型演示需要simulink文件7个，GUI文件1个，GUI函数文件1个，缺一不可，因此在模型的改进与携带方面比较繁重。

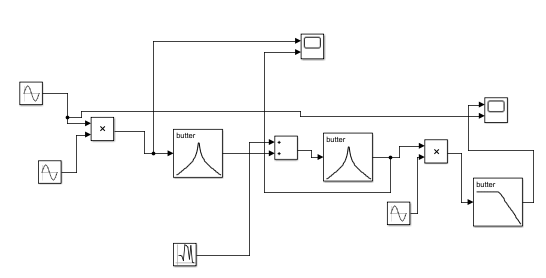
除此之外，在对于信号频率的调节与功率谱绘制上也有不足。本实验虽然可以调节信号的频率，但是用户选择可能过于自由，加之模拟滤波器的最佳滤波范围，合理的频率区间往往需要用户自己去定义。本实验的功率谱采用窗函数周期波法，估计的精度不能十分准确，因此生成的曲线与真实值难免有所偏差，仿真精度有待提高。

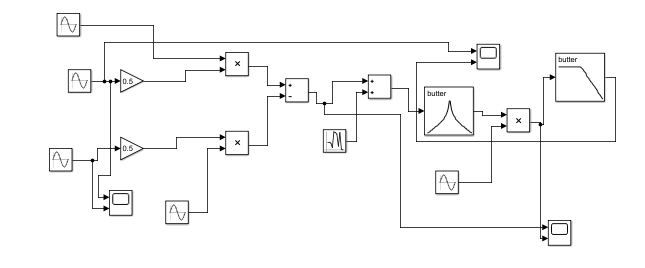
**4.3 实验反思**

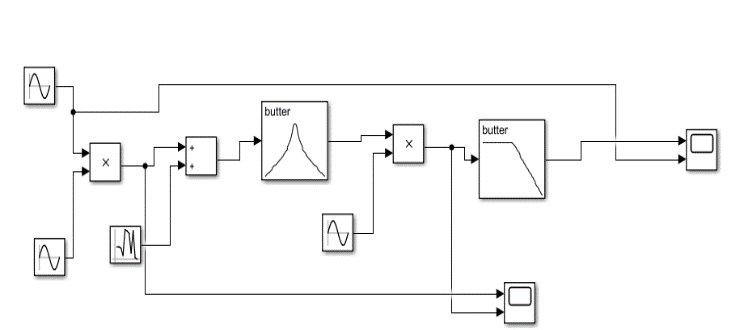
本次设计加大了我对matlab函数、simulink组件以及GUI界面设计的理解与应用能力，并在此基础上将三者联系起来，比如callback函数的编写与模型参数的更改，同时锻炼了我处理复杂问题的逻辑思考能力以及对专业相关的通信系统的理解，做到了学以致用的效果。

本次实验我仍有提高的地方，一是前面所述对信源和载波信号频率的改变，二是GUI界面的进一步优化与美化，三是程序运行速度的提高与代码复杂度的降低，这些需要我在以后的matlab工程仿真中留意与改正，希望我能运用matlab为今后的学习内容铺设道路，拉近自己与专业相关知识的距离。

附录A simulink程序框图与参数设置

AM相干调制解调框图 AM包络调制解调框图

SSB滤波调制解调框图 SSB相移调制解调框图



DSB调制解调框图

参数设置（对框图中未标注的参数进行说明）：

信源：幅度10V，相位0（相移法希尔伯特相位为3\*pi/2）

载波：幅度1V

带通滤波器：90-110Hz（USB：100-110Hz LSB：90-100Hz）

低通滤波器：10Hz

附录B GUI界面

