

西安邮电大学

通信与信息工程学院

信息对抗专业课程设计 I 报告

(2018/2019 学年 第 一学期)

实验名称：_____信息对抗专业课程设计 I_____

学生姓名：_____郝希烜_____

专 业：_____信息对抗技术_____

学 号：_____03166063_____

指导教师：_____吴昊_____

摘要

本次课程设计使我较全面地了解 MATLAB 常用方法指令，能够运用 MATLAB 进行仿真设计，进行模拟调制信号与解调的仿真、线性调频信号以及脉冲压缩性能的仿真、用窗函数设计数字滤波器、二进制数字调制与解调信号的仿真、用 MATLAB 解微分方程等等。实践能力检测包括了矩阵运算、方程求解、绘制函数和数据、添加标注等内容，体现了 MATLAB 软件在数值计算方面首屈一指的强大功能，

关键词：MATLAB；仿真；通信；数值计算；

目录

1. 绪论	1
2. MATLAB 软件的安装	2
3. MATLAB 工作界面的简要介绍	9
3.1 MATLAB 工作界面图解	9
3.2 MATLAB 工作界面介绍	9
4. 课堂作业	10
4.1 求商品的实际销售价格	10
4.2 求平均数之和	12
5. 模拟调制信号与解调的仿真	13
5.1 AM	13
5.2 DSB	14
5.3 SSB	19
6. 线性调频信号以及脉冲压缩性能的仿真	24
6.1 线性调频	24
6.2 脉冲匹配滤波压缩	25
7. 用窗函数设计数字滤波器	27
7.1 题目	27
7.2 程序	27
7.3 结果	29
8. 二进制数字调制与解调信号的仿真	30
8.1 2ASK	30

8.2 2FSK.....	35
8.3 2PSK.....	41
8.4 2DPSK.....	47
9. 用 MATLAB 解微分方程.....	53
10. MATLAB 实践能力检测.....	55
11. MATLAB 实验心得	68
参考文献.....	69

1. 绪论

课程实验目的

通过实验巩固和加深对 MATLAB 语法、数值计算、绘图等功能的理解。并通过一系列仿真加深对通信原理相关知识的掌握。

实验课授课宗旨

- 讲授 MATLAB 的通用功能。
- 寓教于例，由浅入深。
- 关于科学计算，着重强调理论概念、算法和实际计算三者之间的关系。

课程设计要求

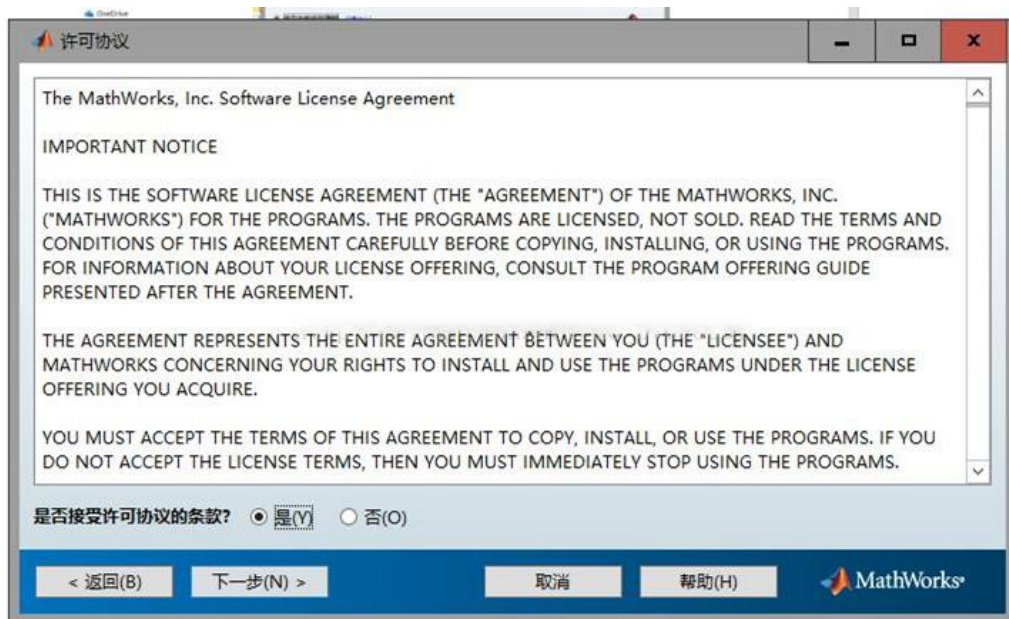
1. MATLAB 软件的安装；
2. MATLAB 及工作界面操作；
3. 两题课堂作业；
4. 模拟调制信号与解调的仿真；
5. 线性调频信号以及脉冲压缩性能的仿真（科研文献 chirp 信号类）；
6. 用窗函数设计数字滤波器；
7. 二进制数字调制与解调信号的仿真；
8. 用 MATLAB 解微分方程；
9. MATLAB 实践能力检测作答；
10. 撰写 MATLAB 实验心得。

2. MATLAB 软件的安装

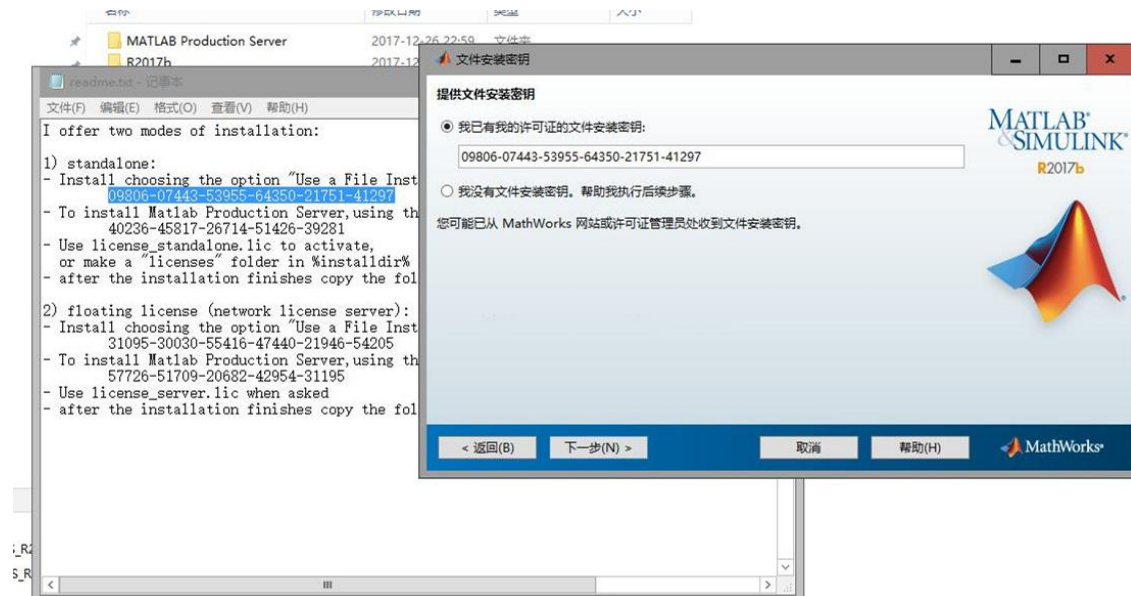
2.1 下载安装包，第一步为解压安装文件，找到文件安装包里的 dvd1和 dvd2，格式为 iso，同时解压。或解压 dvd1，安装到一半会显示请插入 dvd2，这时再点击 dvd2。两者效果一样。但不可只解压 dvd1。



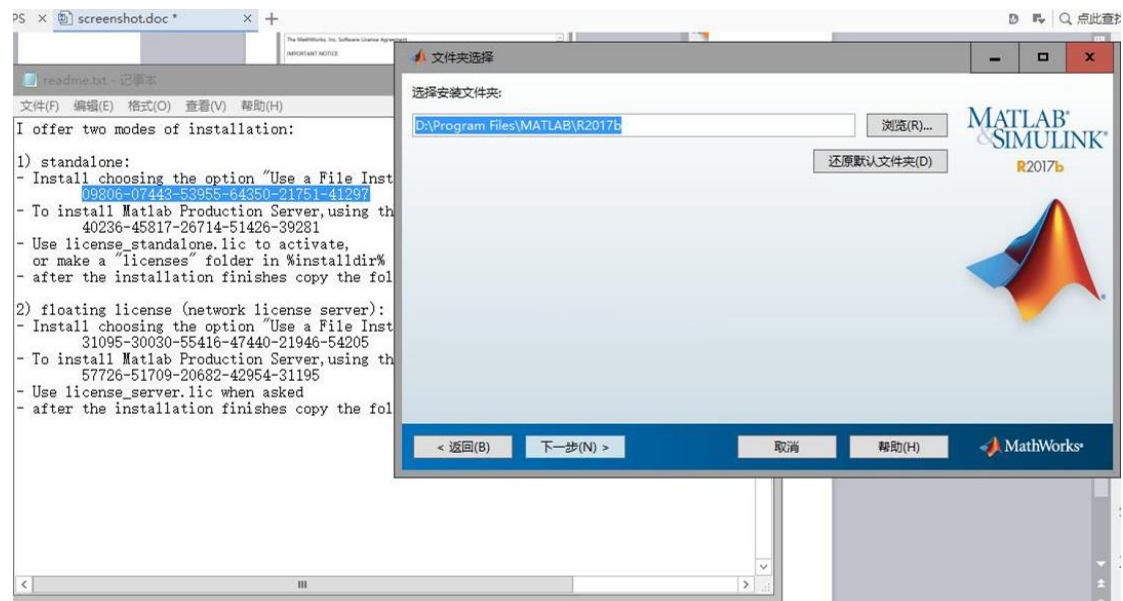
点击接受许可



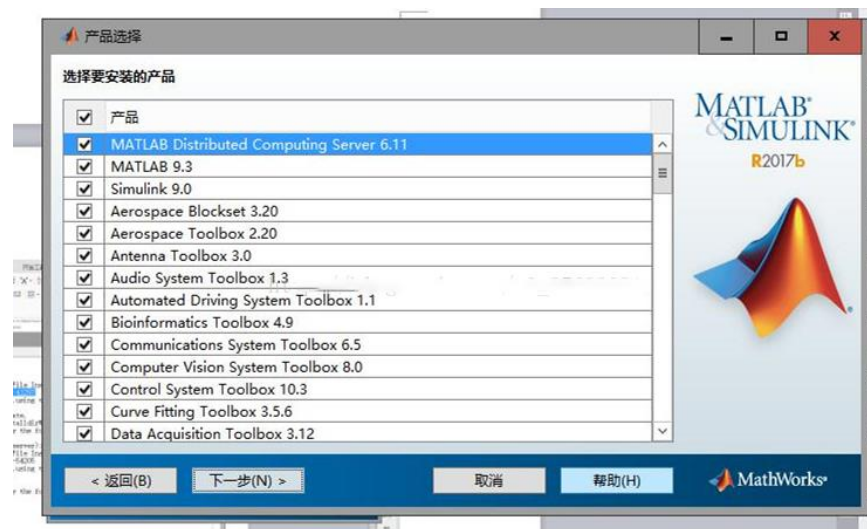
输入安装密钥09806-07443-53955-64350-21751-41297，安装密钥可在 txt 文件中找到



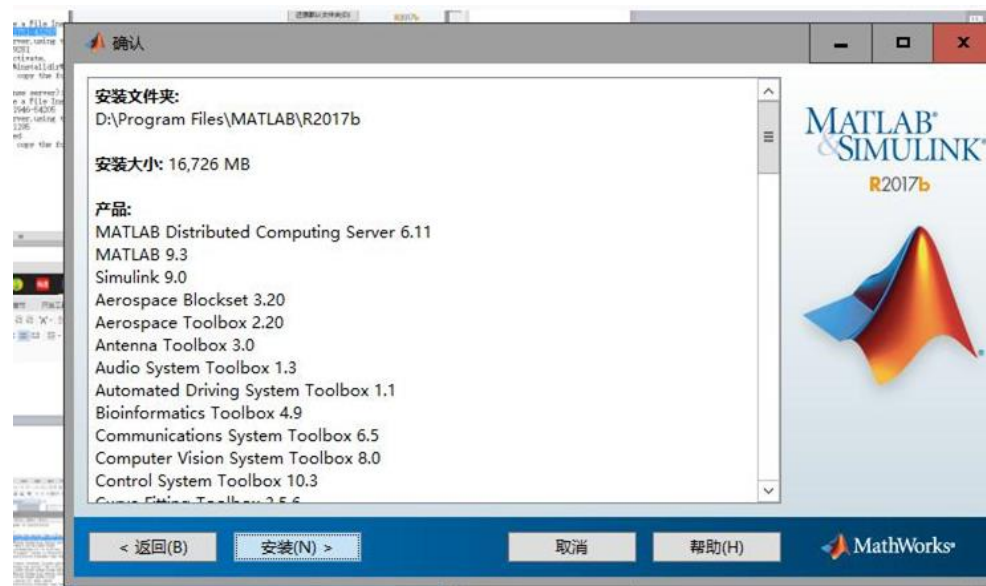
2.3 选择 matlab 安装位置，默认为 C 盘，笔者自己选择了其他文件夹，由于文件很大，所以推荐不要在系统盘



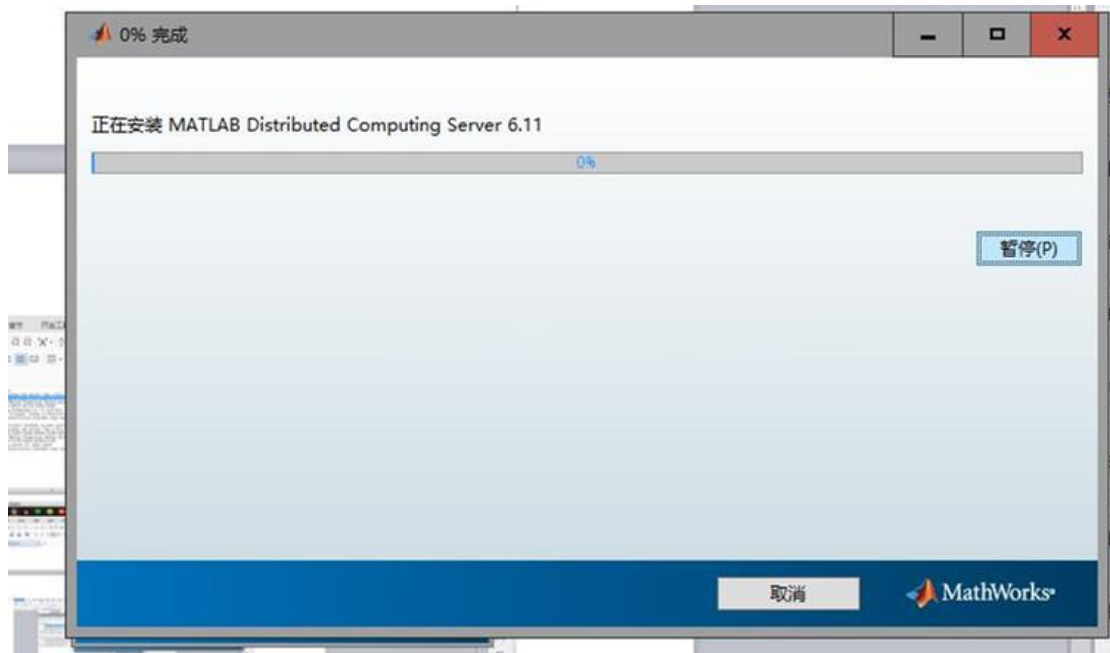
选择安装的产品，默认且推荐全选，2017的优势之处就在于多了很多新的功能，如神经网络安装包等



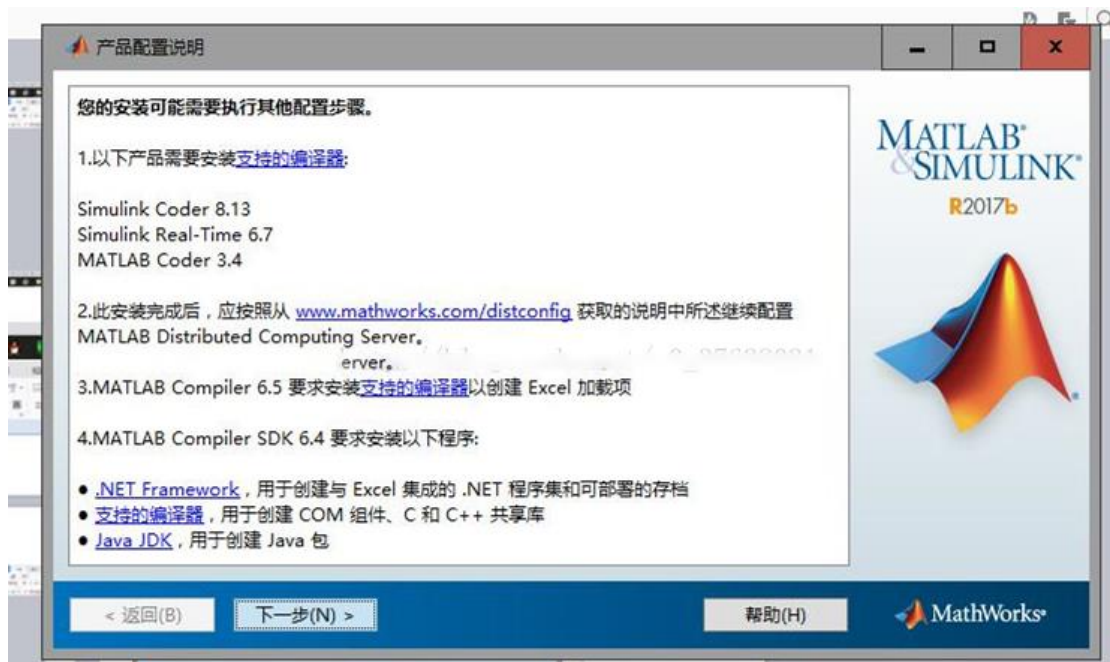
下一步之后，点击安装



安装过程大约会持续半个小时-1个小时



安装完成后会自动弹出配置，点击下一步

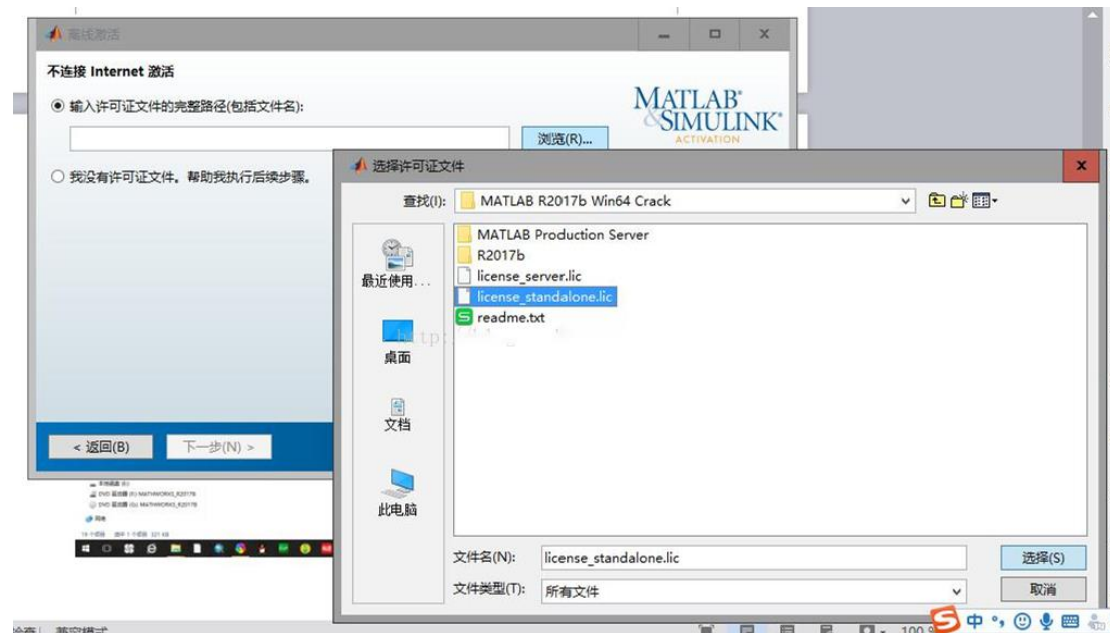


一定要点击 不适用网络激活，点击下一步

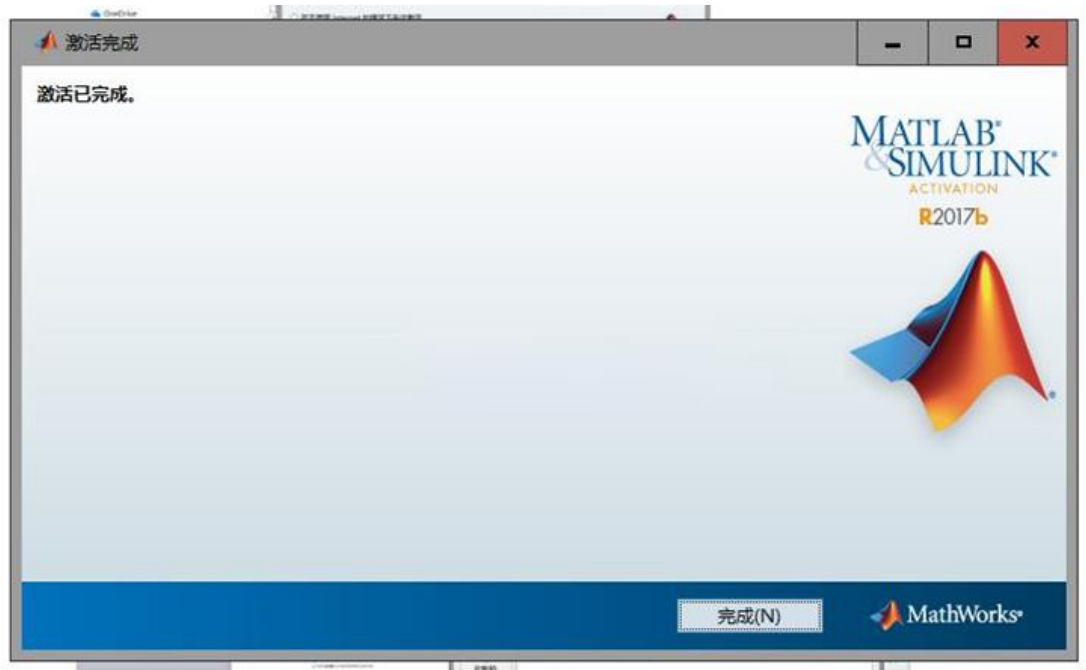


2.4 在输入文件的路径中选择破解文件中的 license_standalone.lic

如果你在前一步安装中输入的是 floating license 中的密钥，就点击 license_server.lic。需要一一对应，基本没有区别。只不过联网的需要登录自己的 mathwork 账号，没有的话比较麻烦，需要在官网去申请

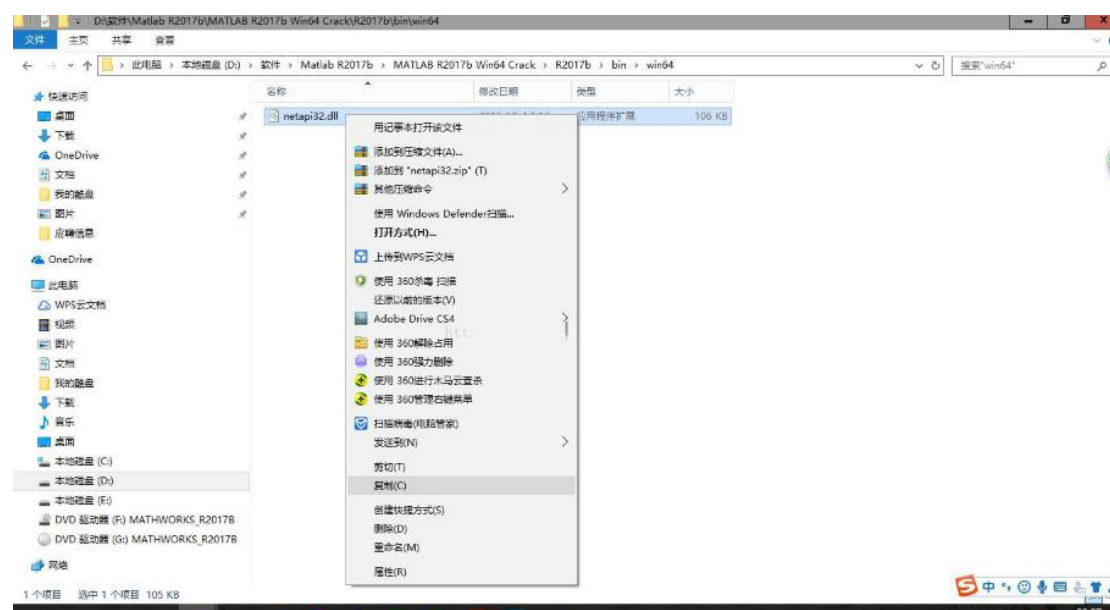


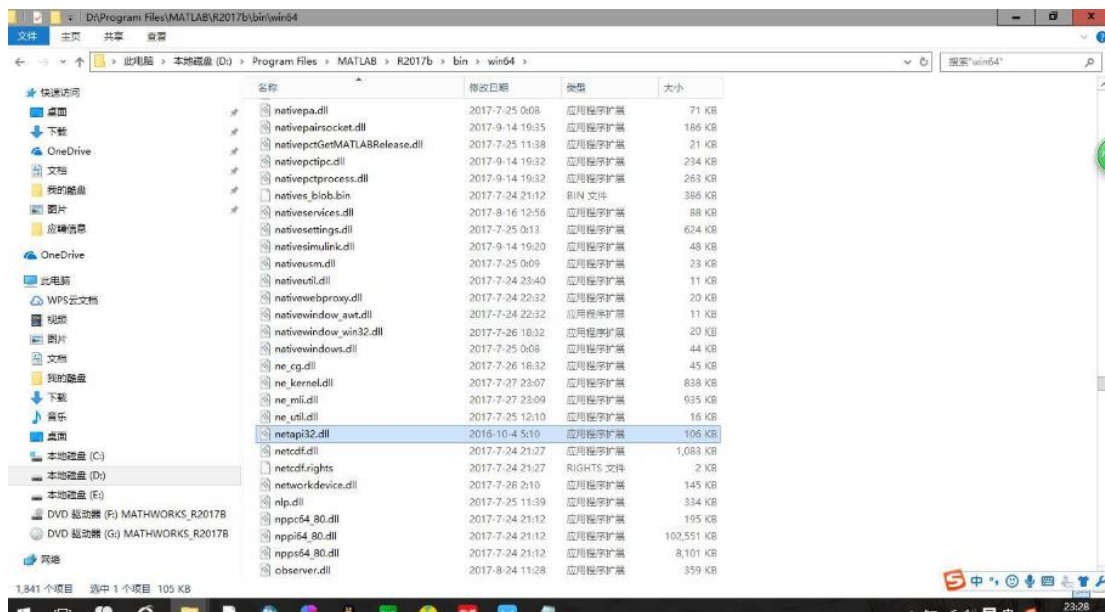
这样会显示激活完成



点击完成退出。但是需要注意的是，这里的激活完成并不意味着激活完全成功，这样软件会打不开，需要进行下一步操作

2.5 将破解文件中的 netapi32.dll 复制到已安装好的对应文件夹中

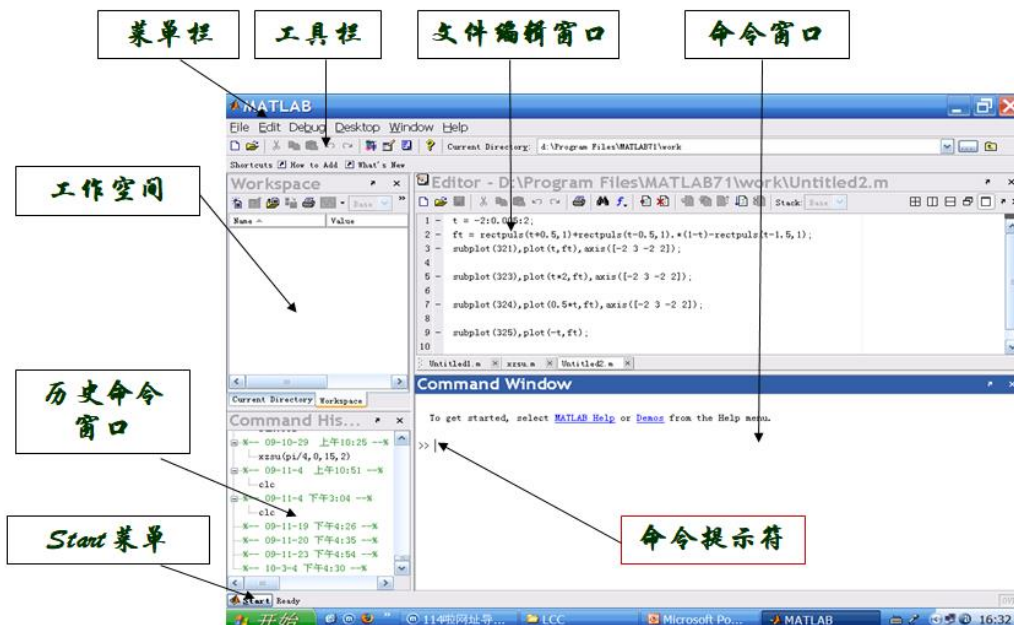




这样，可以在 bin 文件夹中找到 matlab，可发送至桌面为快捷方式，以此打开

3. MATLAB 工作界面的简要介绍

3.1 MATLAB 工作界面图解



3.2 MATLAB 工作界面介绍

3.1.1 功能栏。

在此栏有很多操作设置，可以新建 m 文件(m 文件便于填写大量代码编译，可以保存代码。如果只是一个函数调用，建议用 Command window)，还有个很好用的是 APPS,那里放着很多有用的软件，这些软件都是很实用的，例如图像处理需要用到的分类器训练，就可用 Training Image Labeler 软件划出感兴趣区域等等，更多的软件自己去慢慢探索。

3.2.2 搜索栏。

右上角那里有个“？”按钮，点击那里即可进入文档的主页，又或者在旁边的搜索栏输入自己想要查找的函数，也可以直接在 Command window 窗口输入 help。

3.2.3 文件路径。

这里很重要的是，要操作的图片一定要放在此路径内，不然会出现找不到图片的错误。（我一开始出 BUG 就是在此），因为读取图片是从当前的文件路径读取的，所以图片必须放在这里。当然，你可以修改当前路径，转到你放图片的路径上，还有一种方法就是你可以在 imread（）这里填上图片的路径。

3.2.4 命令窗口。

这是我们经常要用到的窗口，Matlab 之所以强大也是在此，逐行编译。也就是你写 50 行代码，第 50 行出错，它也会编译到 49 行。这对于我们找 BUG，太方便了。至于里面的填写内容，格式等等将会在后续的代码中介绍。

3.2.5 Workspace。

这个窗口存放着图片的数组信息。我们都知道，我们看到的是图片，而电脑看到的是，一堆数字，也就是矩阵。其实一切形式都可以化为矩阵的形式，矩阵是以数组形式的方式。即使一个数字也可看做 1*1 矩阵，不是吗？当你读取图片时，图片的像素就在此保存，点击可查看，这个后续再介绍。

4. 课堂作业

4.1 求商品的实际销售价格

4.1.1 题目

某商场对顾客所购买的商品实行打折销售，标准如下：

price <200	没有折扣
200<=price<500	3%折扣
500<=price<1000	5%折扣
1000<=price<2500	8%折扣
2500<=price<5000	10%折扣
5000<=price	14%折扣

输入所售商品的价格，求其实际销售价格。

4.1.2 程序

```
price=input('please input price:');
ans=price;
switch price
    case num2cell(200:499)
        ans=price*(1-0.03);
    case num2cell(500,999)
        ans=price*(1-0.05);
    case num2cell(1000,2499)
        ans=price*(1-0.08);
    case num2cell(2500,4999)
        ans=price*(1-0.1);
    otherwise
        if(price>=5000)
            ans=price*(1-0.14);
        end
end
disp(ans);
```

4.1.3 运行结果

>>

```
please input price:1000
    920
```

>>

```
please input price:2300
    2300
```

>>

```
please input price:4900
    4900
```

4.2 求平均数之和

4.2.1 题目

从键盘输入若干个数，当输入 0 时结束输入，求这些数的平均值和

4.2.2 程序

```
x=input('please input a number:');
num=0;
while(x~=0)
    num=num+1;
    a(num)=x;
    x=input('please input a number:');
end
sum=0;
if(num==0)
    sum
    ave=0
else
    for i=1:num
        sum=sum+a(i);
    end
    ave=sum/num;
sum
ave
end
```

4.3.3 结果

```
>>
```

```
please input a number:10
```

```
please input a number:435
```

```
please input a number:77
```

```
please input a number:114
```

```
please input a number:0
```

```
sum =
```

```
636
```


ave =

159

5. 模拟调制信号与解调的仿真

5.1 AM

5.1.1 AM 调制解调原理

首先讨论单频信号的调制情况。如果设单频调制信号 $U_o = U_{om}(t)\cos\omega_o t$ ，载波 $u_c = U_{cm}\cos\omega_c t$ ，那么调幅信号（已调波）可表示为

$$u_{AM} = U_{AM}(t) \cos\omega_c t$$

式中， $U_{AM}(t)$ 为已调波的瞬时振幅值。由于调幅信号的瞬时振幅与调制信号成线性关系，即有

$$\begin{aligned} U_{AM}(t) &= U_{cm} + k_a U_{om} \cos\omega_o t \\ &= U_{cm} \left(1 + \frac{k_a U_{om}}{U_{cm}} \right) = U_{cm} (1 + m_a \cos\omega_o t) \end{aligned}$$

由以上两式可得 $u_{AM} = U_{cm}(1 + m_a \cos\omega_o t) \cos\omega_c t$

包络检波是指检波器的输出电压直接反应输入高频调幅波包络变化规律的一种检波方式。由于 AM 信号的包络与调制信号成正比，因此包络检波只适用与 AM 波的解调，其原理方框图如图 1：

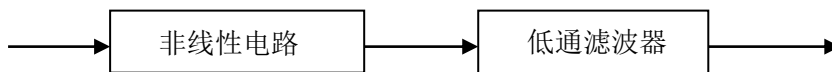


图 1

包络检波器的输入信号为振幅调制信号 $U_i = U_{im}(1 + m_a \cos\omega_o t) \cos\omega_c t$ ，其频谱由载频 ω_c 和边频 $\omega_c + \omega_o$ ， $\omega_c - \omega_o$ 组成，载频与上下边频之差就是 ω_o 。因而它含有调制信号的信息。

5.1.2 程序

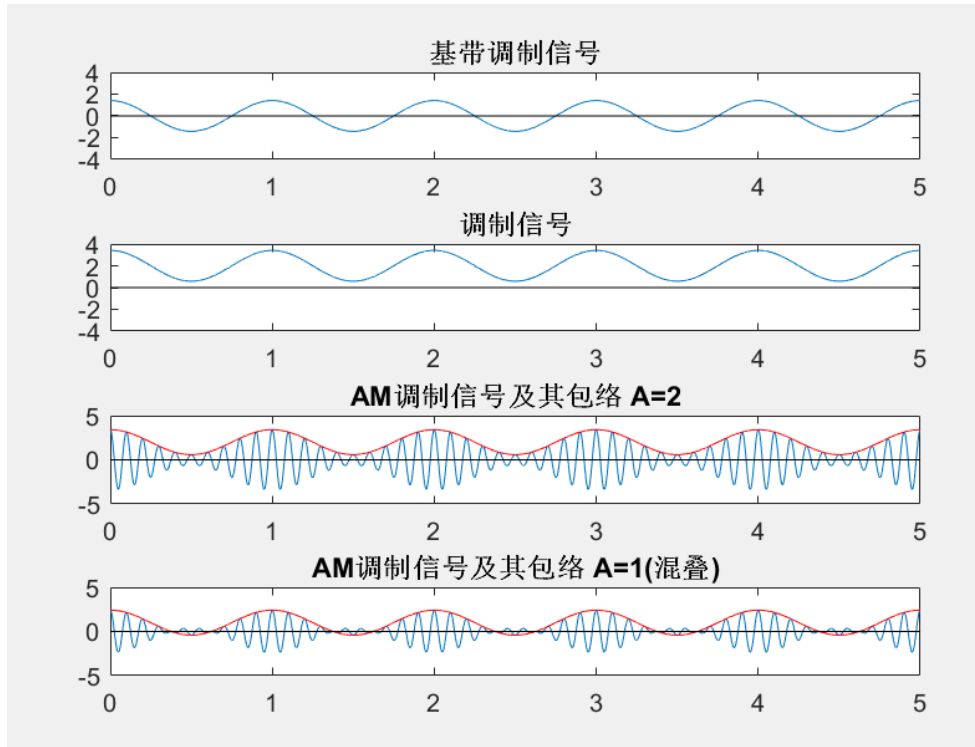
%AM调制信号的MATLAB实现

```

dt=0.0001;          %时间采样频谱
fc=10;              %载波中心频率
T=5;                %信号时长
N=T/dt;             %采样点个数
t=[0:N-1]*dt;       %采样点的时间序列
wc=2*pi*fc;
mt=sqrt(2)*cos(2*pi*t); %信源
subplot(411);
plot(t,mt);
title('基带调制信号');
axis([0 5 -4 4]);
line([0,5],[0,0],'color','k');
%mt的最大值是sqrt(2)
A=2;
subplot(412);
plot(t,A+mt);
title('调制信号');
axis([0 5 -4 4]);
line([0,5],[0,0],'color','k');
sam=(A+mt).*cos(wc*t);
subplot(413);
plot(t,sam);
hold on;           %画出AM信号波形
plot(t,A+mt,'r-');
title('AM调制信号及其包络 A=2');
line([0,5],[0,0],'color','k');
A=1;
sam=(A+mt).*cos(wc*t);
subplot(414);
plot(t,sam);
hold on;           %画出AM信号波形
plot(t,A+mt,'r-');
title('AM调制信号及其包络 A=1(混叠)');
line([0,5],[0,0],'color','k');

```

5. 1. 3 结果



5.2 DSB

5.2.1 DSB 调制与解调原理

在 AM 调制过程中，如果将载波分量抑制掉，就可形成抑制载波双边带信号。双边带信号可以用载波和调制信号直接相乘得到，即

$$U_{DSB} = k u_{\Omega}(t) u_{\omega_c} t$$

式中，常数 k 为相乘电路的相乘系数。

如果调制信号为单频信号 $u_{\omega_{\Omega}} = U_{\Omega} \cos \omega_{\Omega} t$ ，载波 $u_{\omega_c} = U_{cm} \cos \omega_c t$ ，则

$$\begin{aligned} U_{DSB} &= k U_{\omega_{\Omega}} U_{cm} \cos \Omega t \cos \omega_c t \\ &= \frac{1}{2} k U_{\Omega m} U_{cm} [\cos(\omega_c + \Omega) t + \cos(\omega_c - \Omega) t] \end{aligned}$$

同步检波分为乘积型与叠加型两种方式，这两种检波方式都需要接收端恢复载波支持。乘积型同步检波是直接把本地回复的借条载波和接收信号相乘，然后用低通滤波器将低频信号提取出来。在这种检波器中，要求本地的解调载波和发送端的调制载波同频同相。如果其频率或相位有一定的偏差，将会使恢复出来的调制信号产生失真。

设输入已调波信号 $u_i = U_{im} \cos \omega_{\Omega} t \cos \omega_c t$ ，本地解调载波 $u_o = U_{om} \cos(\omega_{\Omega} t + \varphi)$ ，则两信号相乘后的输出为

$$u_i u_o = k U_{im} U_{om} \cos \Omega t \cos \omega_c t \cos(\omega_{\Omega} t + \varphi)$$

$$= \frac{1}{2} k U_{im} U_{om} \cos \Omega t \{ \cos[(w_c + w_o) + \varphi] + \cos[(w_c - w_o) + \varphi] \}$$

式中，k 为乘法器的相乘系数。令 $w_c - w_o = \Delta w_o$ ，且低通滤波器的传输系数为 1，则经低通滤波器后的输出信号为

$$U_{\Omega} = \frac{1}{2} k U_{im} U_{om} \cos \Omega t \cos(\Delta w_o + \varphi) = U_{\Omega} \cos(\Delta w_o t + \varphi) \cos \Omega t$$

$$= U_{\Omega}(t) \cos \Omega t$$

当恢复的本地载波与发射端的调制载波同步（同频，同相），即 $\Delta w_o = 0$ ， $\varphi = 0$

时，有 $u_{\Omega} = U_{\Omega} \cos(\Omega t)$ 。即表明同步检波器能无失真地将调制信号恢复出来。

5. 2. 2. 程序

```
dt=0.001; %时间采样间隔
fm=1; %信源最高频率
fc=10; %载波中心频率
T=5; %信号时长
t=0:dt:T;
mt=sqrt(2)*cos(2*pi*fm*t); %信源
figure(1)
subplot(311);
plot(t,mt);
title('调制信号')
coss=cos(2*pi*fc*t);
subplot(312);
plot(t,coss);
title('载波信号')
%N0=0.01; %白噪声单边功率谱密度
%DSB调制
s_dsb=mt.*cos(2*pi*fc*t);
B=2*fm;
%noise=noise_nb(fc,B,N0,t);
%s_dsb=s_dsb+noise;
subplot(313)
plot(t,s_dsb); %画出DSB信号波形
hold on
plot(t,mt,'r--'); %标出m(t)波形
hold on
plot(t,-mt,'r--');
```

```

title('DSB调制信号');
%DSB相干解调
rt=s_dsb.*cos(2*pi*fc*t);
figure(2);
subplot(311);
plot(t,rt);
title('DSB调制信号与载波信号相乘')
[f,rf]=T2F(t,rt);%傅里叶变换
[t,rt]=lpf(f,rf,fb);%低通滤波
subplot(312)
plot(t,rt);
title('经过低通滤波的相干解调信号波形');
rt=rt-mean(rt);
subplot(313)
[f,sf]=T2F(t,s_dsb);%傅里叶变换
psf=(abs(sf).^2)/T;
plot(f,psf);
axis([-2*fc 2*fc 0 max(psf)]);
title('DSB信号功率谱');

```

用到的函数

①T2F.m

```
function [f,sf]= T2F(t,st)
```

%利用FFT计算信号的频谱并与信号的真实频谱的抽样比较。

%脚本文件T2F.m定义了函数T2F，计算信号的傅立叶变换。

%This is a function using the FFT function to calculate a signal Fourier

%Translation

%Input is the time and the signal vectors,the length of time must greater

%than 2

%Output is the frequency and the signal spectrum

```
dt = t(2)-t(1);
```

```
T=t(end);
```

```
df = 1/T;
```

```
N = length(st);
```

```
f=-N/2*df : df : N/2*df-df;
```

```
sf = fft(st);
```

```
sf = T/N*fftshift(sf);
```

②lpf.m

```
function [t,st]=lpf(f,sf,B)
```

%This function filter an input data using a lowpass filter

%Inputs: f: frequency samples

% sf: input data spectrum samples

% B: lowpass bandwidth with a rectangle lowpass

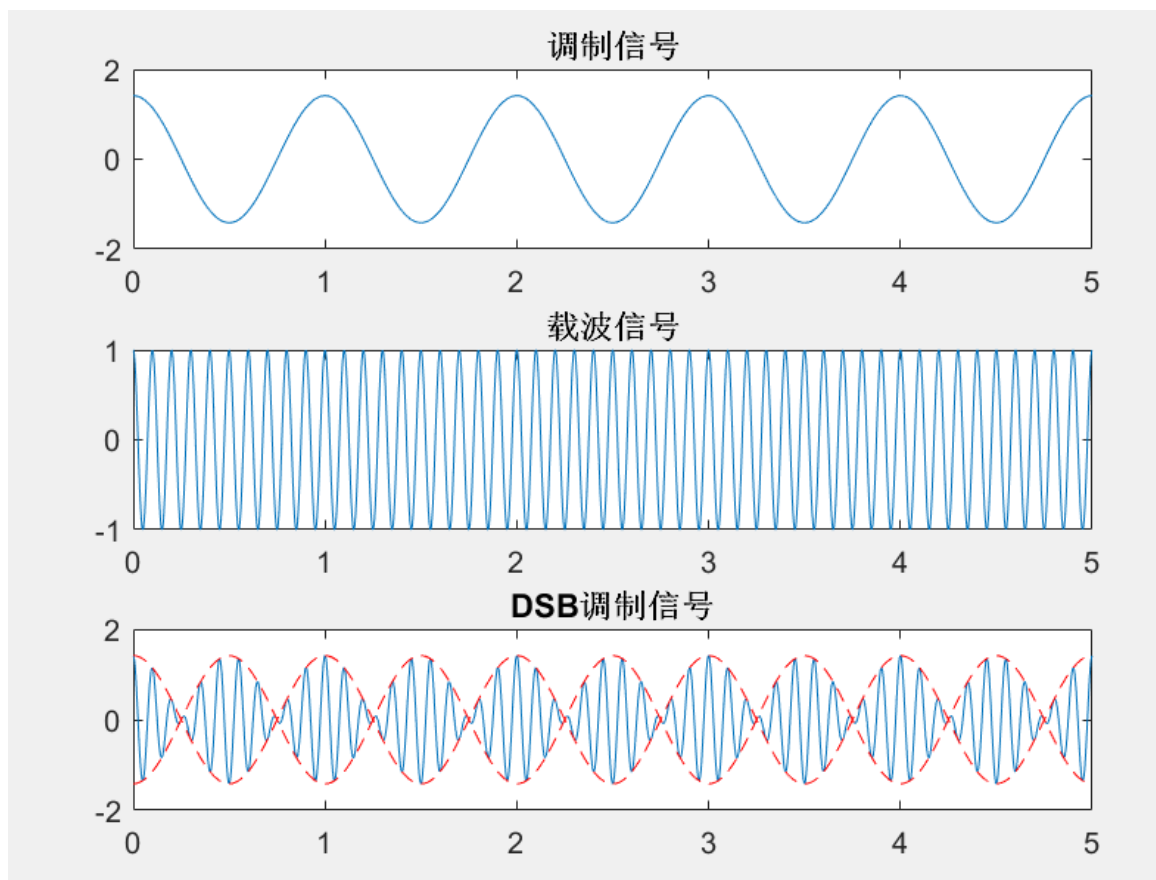
%Outputs: t: time samples

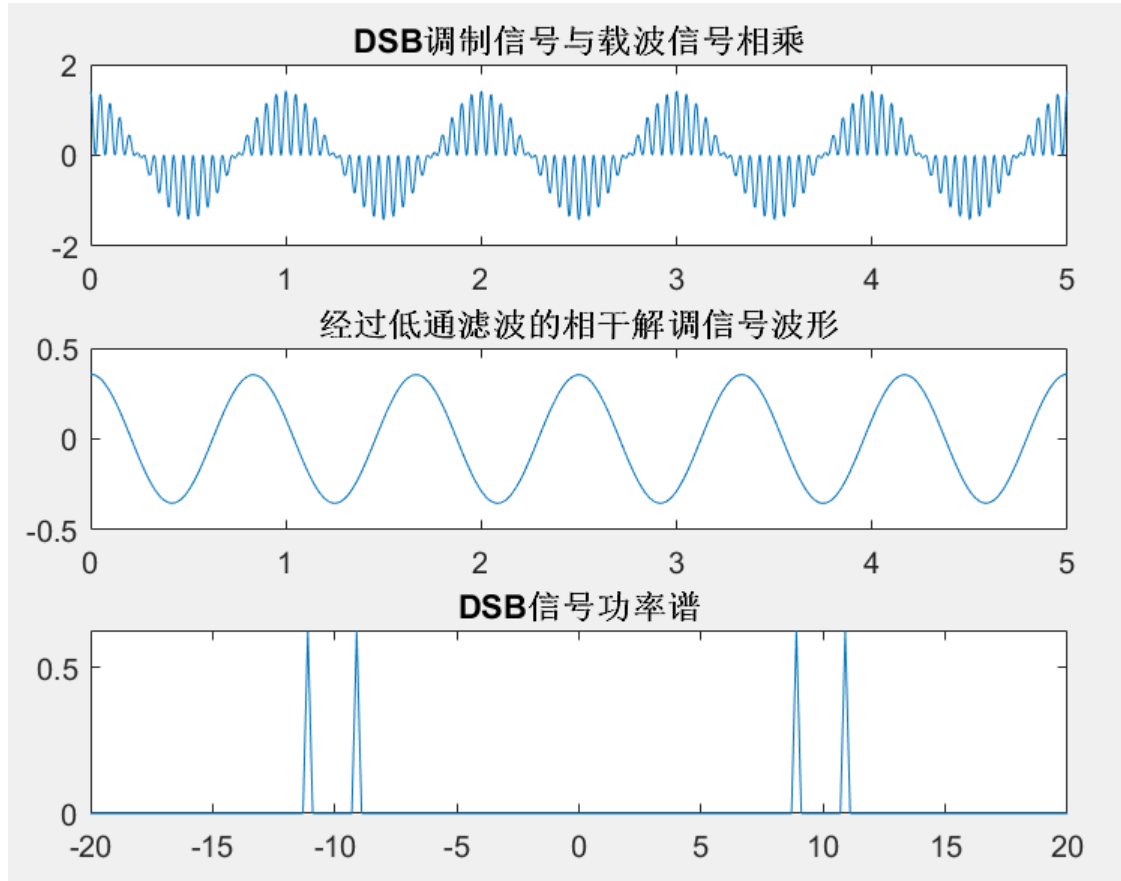
```

% st: output data time samples
df = f(2)-f(1);
T = 1/df;
hf = zeros(1,length(f));%全零矩阵
bf = [-floor( B/df ): floor( B/df )] + floor( length(f)/2 );
hf(bf)=1;
yf=hf.*sf;
[t,st]=F2T(f,yf);
st = real(st);

```

5.2.3 结果





5.3 SSB

5.3.1 SSB 调制与解调原理

对双边带调幅信号，只要取出其中的任一个边带部分，即可成为单边带调幅信号。其单频调制时的表示式为

$$\text{上边带信号} \quad U_{SSBU}(t) = \frac{1}{2} k U_{\Omega m} U_{cm} \cos(\omega_c + \Omega) t$$

$$\text{下边带信号} \quad U_{SSBL}(t) = \frac{1}{2} k U_{\Omega m} U_{cm} \cos(\omega_c - \Omega) t$$

单边带信号的频谱宽度 $B_{SSB} = \Omega_{max}$ ，仅为双边带振幅信号的一半，从而提高了频带使用率。由于只发射一个频带，因此大大节省了发射功率。

本文选用下边带信号进行解调，采用乘积型同步检波方式。设输入已调波信号为 $u_i = U_{im} \cos(\omega_c - \Omega) t$ ，本地解调载波 $u_o = U_{om} \cos(\omega_o t + \varphi)$ ，则两信号相乘后的输出为

$$u_o u_i = k U_{im} U_{om} \cos(\omega_c - \Omega) t \cos(\omega_o t + \varphi)$$

$$= \frac{1}{2} k U_{im} U_{om} \{ \cos[(\omega_c - \Omega) t + \omega_o t + \varphi] + \cos[(\omega_c - \Omega) t - \omega_o t - \varphi] \}$$

式中， k 为乘法器的相乘系数。令 $\omega_c - \omega_o = \Delta\omega_o$ ，且低通滤波器的传输系数为 1，则经低通滤波器后的输出信号为

$$U_{\Omega} = \frac{1}{2} k U_{im} U_{om} \cos[(\omega_c - \Omega)t - \omega_o t - \varphi] = U_{\Omega} \cos(\Delta\omega_o t - \Omega t - \varphi)$$

当恢复的本地载波与发射端的调制载波同步（同频，同相），即 $\Delta\omega_o = 0$ ， $\varphi = 0$ 时，有 $u_{\Omega} = U_{\Omega} \cos(\Omega t)$ 。即表明同步检波器能无失真地将调制信号恢复出来。

5.3.2 程序

```
fm=10;fc=40;
am=sqrt(2);
Fs=300;      %采样频率 Fs，载波频率 fc，信号频率 fm
wc=2*pi*fc;
wm=fm*2*pi;
N=300;
n=0:N-1;
t=n/Fs;      %时间序列
f=n*Fs/N;
%基带信号时域
sm=am*cos(wm*t);
figure(1);
subplot(211);
plot(t,sm);
title('基带信号');
xlabel('t');
axis([0 1 -2 2]);
grid on
%基带信号频域
S=fft(sm,300);%300 点的 fft
SG=abs(S);
```



```

subplot(212);
plot(f(1:N/2),SG(1:N/2));      %SSB 信号频域波形
xlabel('Frequency(HZ)');
title('基带信号频域波形 ');
grid on;
%SSB 调制信号时域
s=modulate(sm,fc,Fs,'amssb');    %对调制信号进行调制
S=fft(s,300);
SG=abs(S);
figure(2);
subplot(211);
plot(t,s);                      %SSB 信号时域波形
title('SSB 信号时域波形 ');
xlabel('t');
subplot(212);
plot(f(1:N/2),SG(1:N/2));      %SSB 信号频域波形
xlabel('Frequency(HZ)');
title('SSB 信号频域波形 ');
grid on;
%-----
%解调
fm=10;%信号频率 fm
fc=40;%载波频率 fc
am=sqrt(2);
Fs=300;      %采样频率 Fs
wc=2*pi*fc;
wm=fm*2*pi;
N=300;
n=0:N-1;

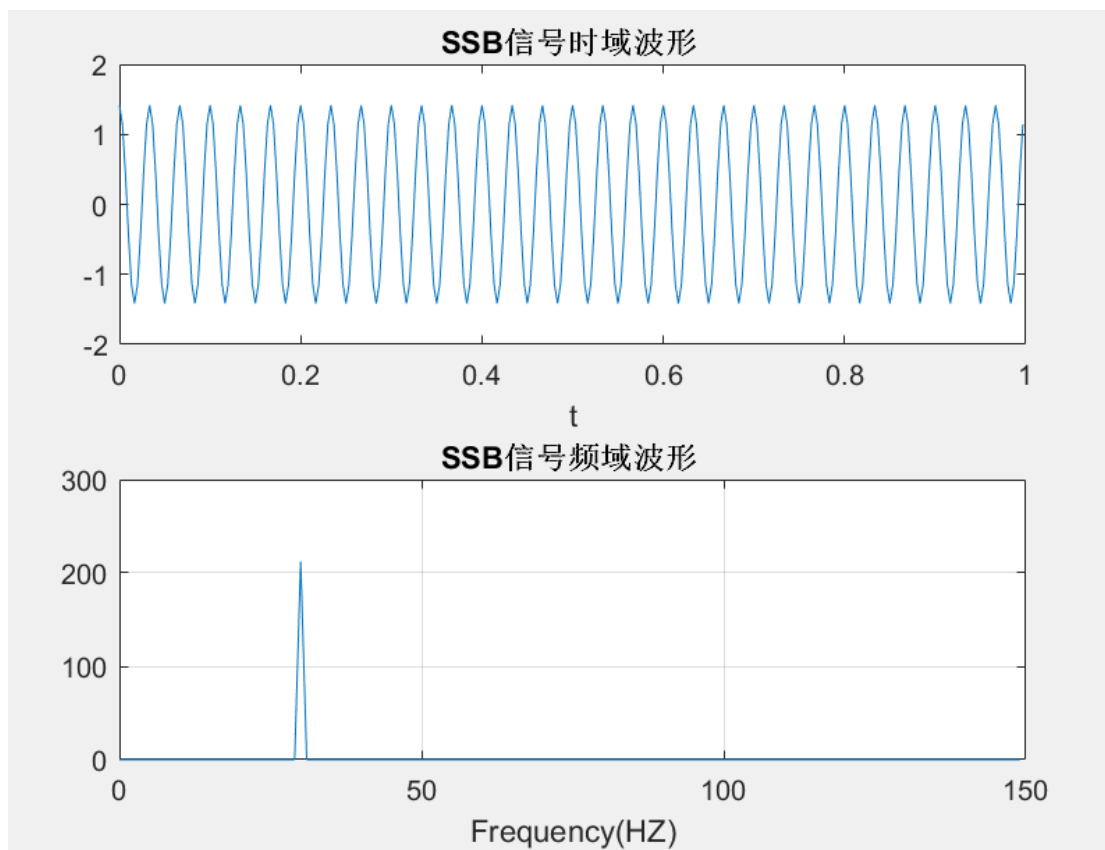
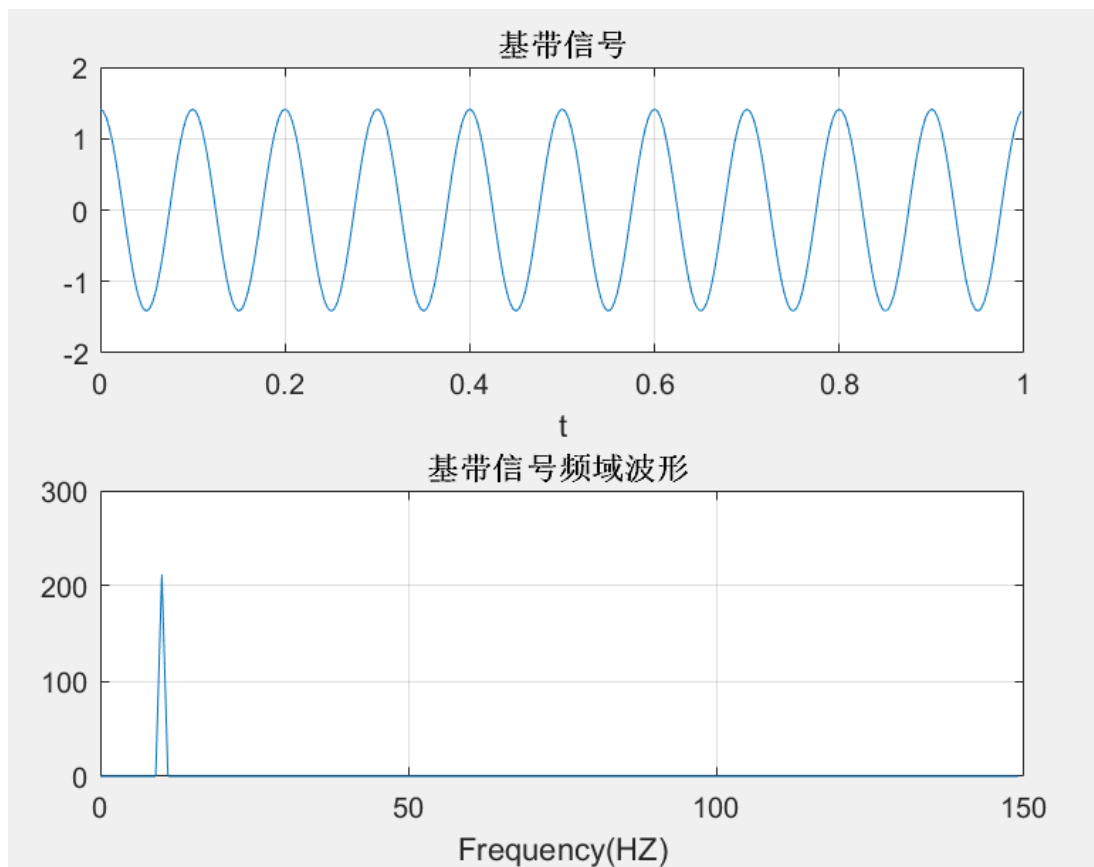
```

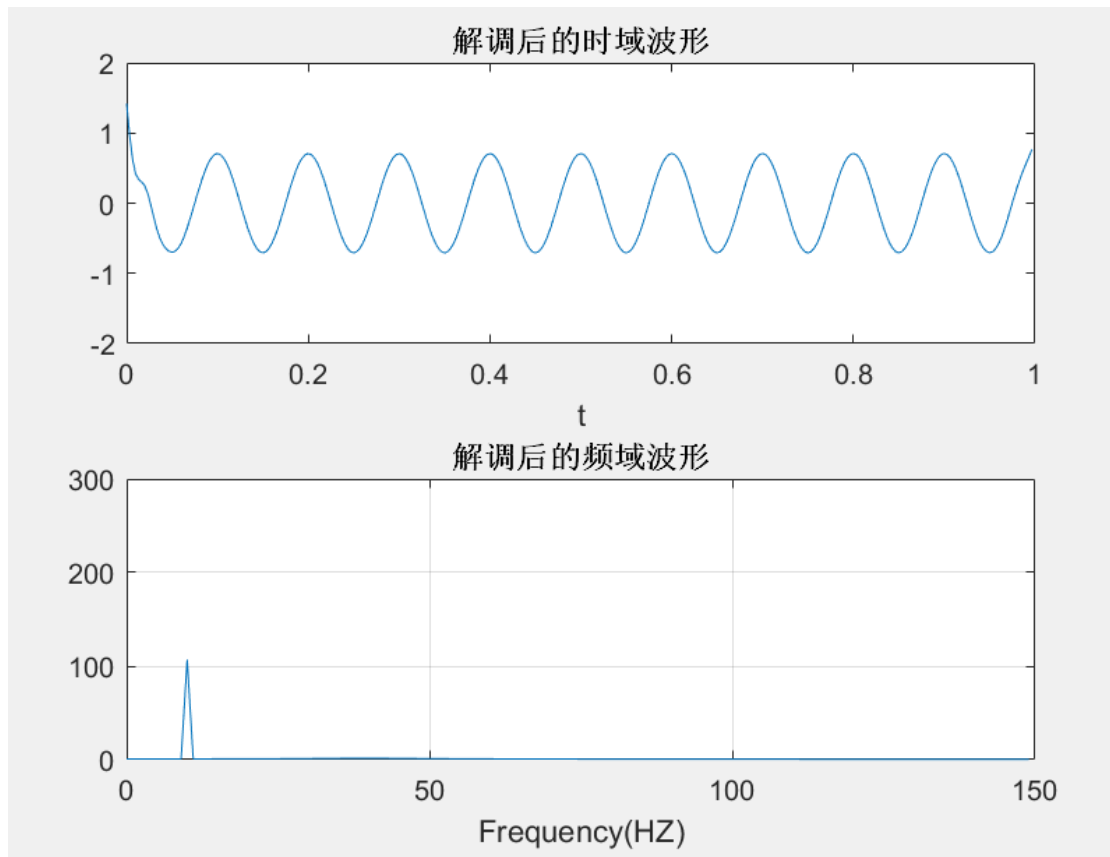
```

t=n/Fs;      %时间序列
f=n*Fs/N;
sm=am*cos(wm*t);
s=modulate(sm,fc,Fs,'amssb');
sd=demod(s,fc,Fs,'amssb');      %对 SSB 信号进行解调
SD=fft(sd,300);
SDG=abs(SD);
figure(3);
subplot(2,1,1);
plot(t,sd);                      %解调后的时域波形
title('解调后的时域波形');
xlabel('t');
axis([0 1 -2 2]);
subplot(2,1,2);
plot(f(1:N/2),SDG(1:N/2));      %解调后的频域波形
title('解调后的频域波形');
xlabel('Frequency(HZ)');
axis([0 150 0 300]);
grid on;

```

5. 3. 3 结果





6. 线性调频信号以及脉冲压缩性能的仿真

6.1 线性调频

6.1.1 程序

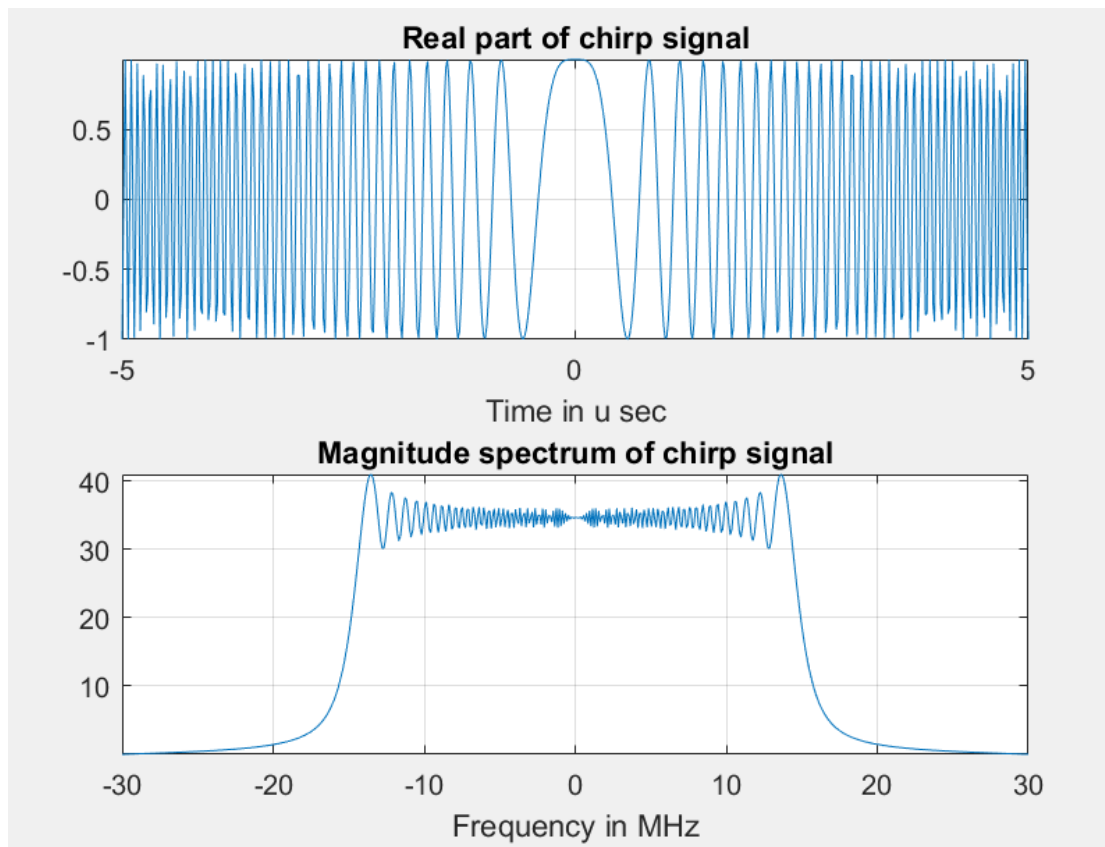
```
%%demo of chirp signal
T=10e-6;                                %pulse duration 10us
B=30e6;                                  %chirp frequency modulation
bandwidth 30MHz
K=B/T;                                   %chirp slope
Fs=2*B;Ts=1/Fs;                          %sampling frequency and sample
spacing
N=T/Ts;
t=linspace(-T/2,T/2,N);
St=exp(j*pi*K*t.^2);                    %generate chirp signal
subplot(211)
plot(t*1e6,real(St));
xlabel('Time in u sec');
title('Real part of chirp signal');
grid on;axis tight;
subplot(212)
```

```

freq=linspace(-Fs/2,Fs/2,N);
plot(freq*1e-6,fftshift(abs(fft(St))));
xlabel('Frequency in MHz');
title('Magnitude spectrum of chirp signal');
grid on;axis tight;

```

6.1.2 结果



6.2 脉冲匹配滤波压缩

6.2.1 程序

```

%%demo of chirp signal after matched filter
T=10e-6;                                %pulse duration10us
B=30e6;                                  %chirp frequency modulation
bandwidth 30MHz
K=B/T;                                    %chirp slope
Fs=10*B;Ts=1/Fs;                         %sampling frequency and sample
spacing
N=T/Ts;
t=linspace(-T/2,T/2,N);
St=exp(j*pi*K*t.^2);                     %chirp signal
Ht=exp(-j*pi*K*t.^2);                    %matched filter
Sot=conv(St,Ht);                         %chirp signal after matched filter

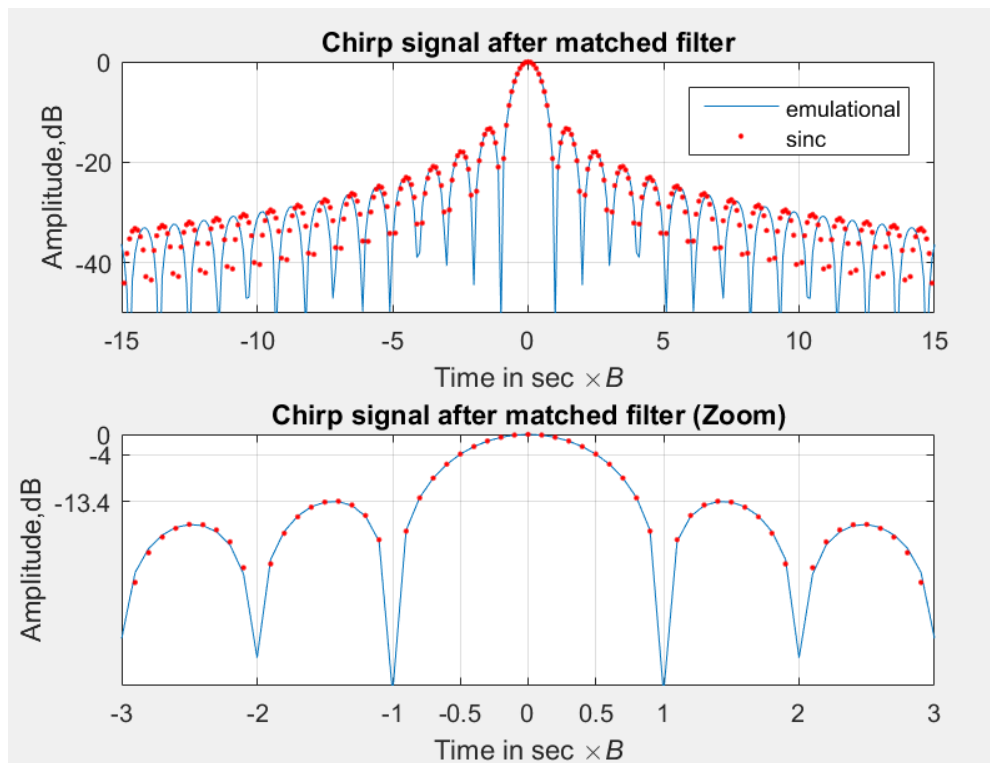
```

```

subplot(211)
L=2*N-1;
t1=linspace(-T,T,L);
Z=abs(Sot);Z=Z/max(Z);           %normalize
Z=20*log10(Z+1e-6);
Z1=abs(sinc(B.*t1));           %sinc function
Z1=20*log10(Z1+1e-6);
t1=t1*B;
plot(t1,Z,t1,Z1,'r. ');
axis([-15,15,-50,inf]);grid on;
legend('emulational','sinc');
xlabel('Time in sec \times \it B');
ylabel('Amplitude,dB');
title('Chirp signal after matched filter');
subplot(212)                    %zoom
N0=3*Fs/B;
t2=-N0*Ts:Ts:N0*Ts;
t2=B*t2;
plot(t2,Z(N-N0:N+N0),t2,Z1(N-N0:N+N0),'r. ');
axis([-inf,inf,-50,inf]);grid on;
set(gca,'Ytick',[-13.4,-4,0],'Xtick',[-3,-2,-1,-0.5,0,0.5,1,2,3]);
xlabel('Time in sec \times \it B');
ylabel('Amplitude,dB');
title('Chirp signal after matched filter (Zoom)');

```

6. 2. 2 结果



7. 用窗函数设计数字滤波器

7.1 题目

用窗函数法设计一线性相位 FIR 低通滤波器，设计指标为

$$W_p=0.3\pi, W_s=0.5\pi, R_p=0.25\text{dB}, R_s=50\text{dB}$$

7.2 程序

①hamming 窗

```
%design a digital FIR lowpass filter with Hamming window
wp=0.3*pi;
ws=0.5*pi;
tr_width=ws-wp;
N=ceil(6.6*pi/tr_width)+1; %N=34
%N=ceil(6.6*pi/tr_width);
n=0:1:N-1;
wc=(ws+wp)/2;
hd=ideallp(wc,N);
w_ham=hamming(N)';
h=hd.*w_ham;
[dB,mag,pha,grd,w]=freqz_m(h,[1]);
delta_w=2*pi/1000;
Rp=-(min(dB(1:1:wp/delta_w+1)));
As=-round(max(dB(round(ws/delta_w+1):1:501)));
%plots
subplot(221);
stem(n,hd);
title('理想脉冲响应');
axis([0,N-1,-0.09,0.5]);
xlabel('n');
set(gca,'YTickMode','manual','YTick',[0,0.1,0.2,0.3]);
subplot(222);
stem(n,w_ham);
title('Hamming 窗');
axis([0,N-1,0,1.1]);
xlabel('n');
subplot(223);
stem(n,h);
title('实际脉冲响应');
axis([0,N-1,-0.09,0.5]);
xlabel('n');
set(gca,'YTickMode','manual','YTick',[0,0.1,0.2,0.3]);
```

```

subplot(224);
plot(w/pi,dB);
title('幅频响应:dB');
grid;
axis([0,1,-100,0]);
xlabel('pi');
ylabel('dB');
%set(gca,'YTickMode','manual','XTick',[0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]);
%set(gca,'YTickMode','manual','YTick',[-100,-90,-80,-70,-60,-50,-40,-30,-20,-10,0]);

```

②blackman 窗

%design a digital FIR lowpass filter with Hamming window

```

wp=0.3*pi;
ws=0.5*pi;
tr_width=ws-wp;
N=ceil(6.6*pi/tr_width)+1; %N=34
%N=ceil(6.6*pi/tr_width);
n=0:1:N-1;
wc=(ws+wp)/2;
hd=ideallp(wc,N);
w_ham=blackman(N);
h=hd.*w_ham;
[dB,mag,pha,grd,w]=freqz_m(h,[1]);
delta_w=2*pi/1000;
Rp=-(min(dB(1:1:wp/delta_w+1)));
As=-round(max(dB(round(ws/delta_w+1):1:501)));
%plots
subplot(221);
stem(n,hd);
title('理想脉冲响应');
axis([0,N-1,-0.09,0.5]);
xlabel('n');
set(gca,'YTickMode','manual','YTick',[0,0.1,0.2,0.3]);
subplot(222);
stem(n,w_ham);
title('blackman 窗');
axis([0,N-1,0,1.1]);
xlabel('n');
subplot(223);
stem(n,h);
title('实际脉冲响应');

```

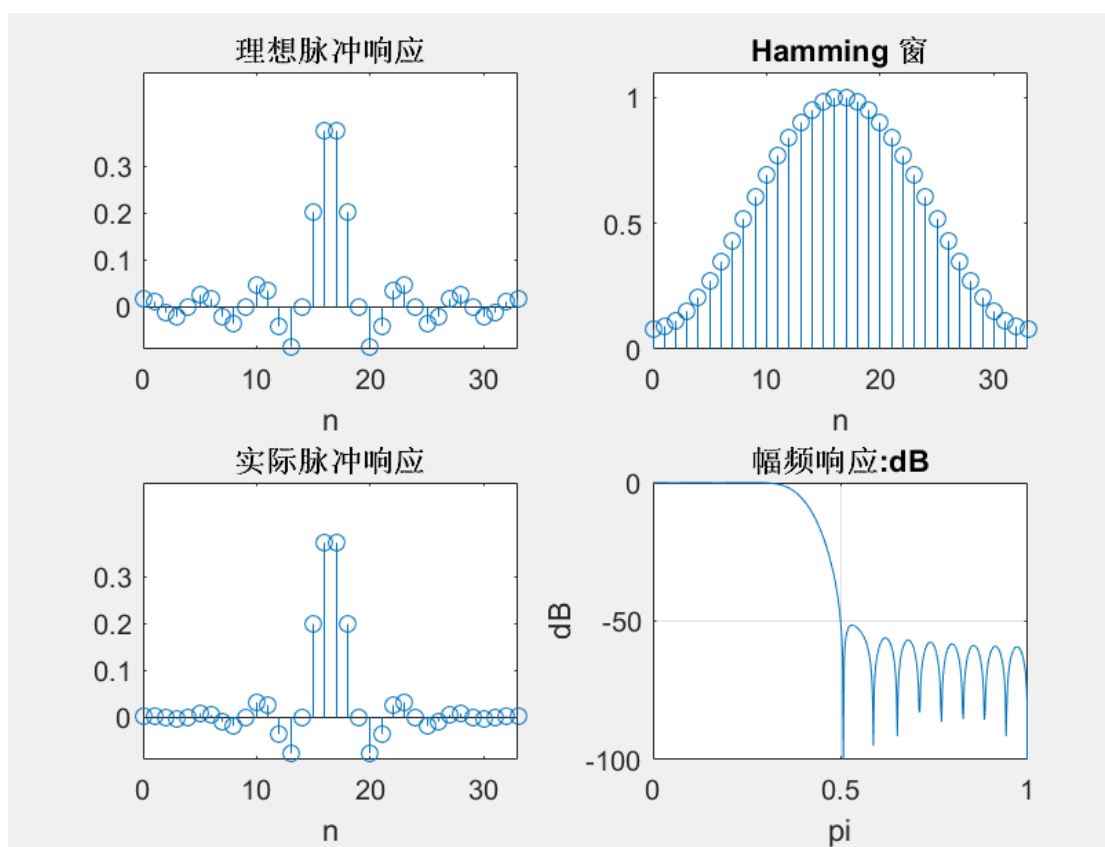


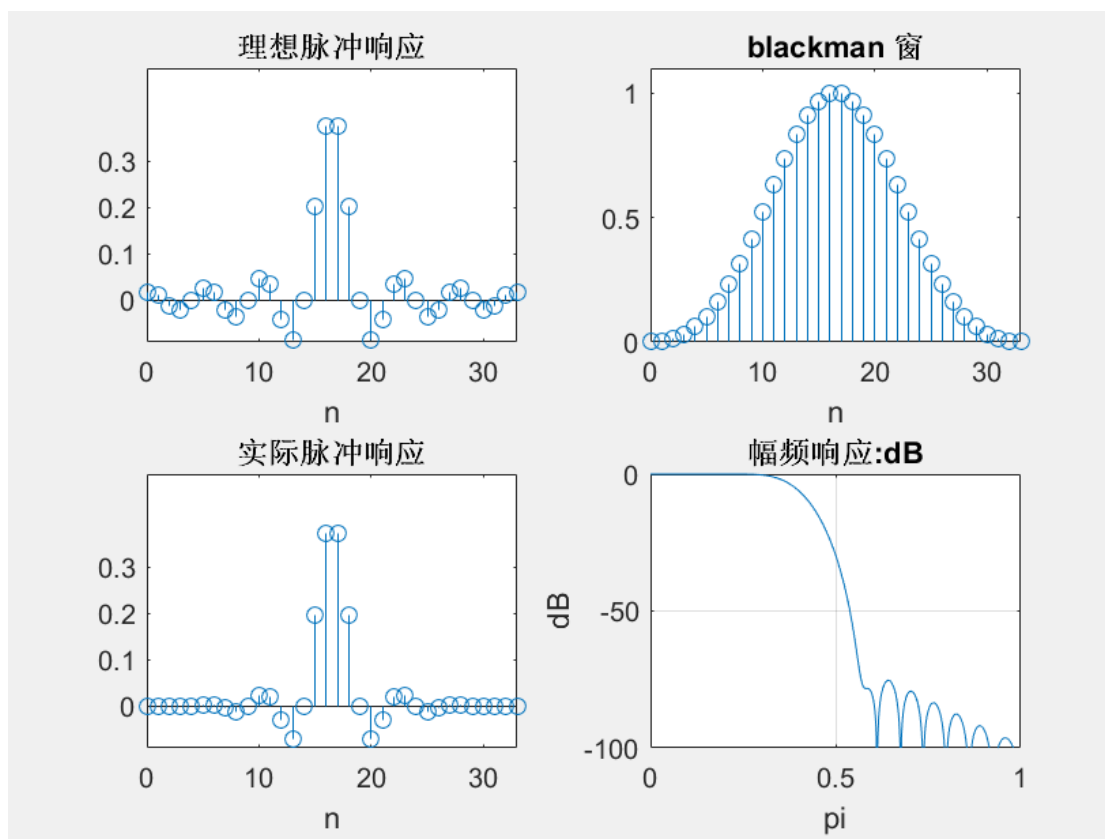
```

axis([0,N-1,-0.09,0.5]);
xlabel('n');
set(gca,'YTickMode','manual','YTick',[0,0.1,0.2,0.3]);
subplot(224);
plot(w/pi,dB);
title('幅频响应:dB');
grid;
axis([0,1,-100,0]);
xlabel('pi');
ylabel('dB');
%set(gca,'YTickMode','manual','XTick',[0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]);
%set(gca,'YTickMode','manual','YTick',[-100,-90,-80,-70,-60,-50,-40,-30,-20,-10,0]);

```

7.3 结果





8. 二进制数字调制与解调信号的仿真

8.1 2ASK

8.1.1 2ASK 原理

频移键控是利用载波的幅度变化来传递数字信息，而其频率和初始相位保持不变。在 2ASK 中，载波的幅度只有两种变化状态，分别对应二进制信息“0”或“1”。二进制振幅键控的表达式为：

$$s(t) = A(t) \cos(w_0 + \theta) \quad 0 < t \leq T$$

式中， $w_0 = 2\pi f_0$ 为载波的角频率； $A(t)$ 是随基带调制信号变化的时变振幅，即

$$A(t) = \begin{cases} A & \text{当发送“1”时} \\ 0 & \text{当发送“0”时} \end{cases}$$

典型波形如图 1 所示：

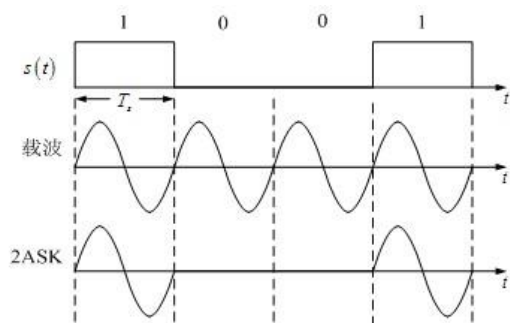


图 1

2ASK 信号的产生方法通常有两种：相乘法 and 开关法，相应的调制器如图 2。图 2 (a) 就是一般的模拟幅度调制的方法，用乘法器实现；图 2 (b) 是一种数字键控法，其中的开关电路受 $s(t)$ 控制。

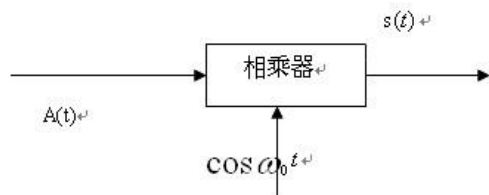


图 2 (a) 相乘法

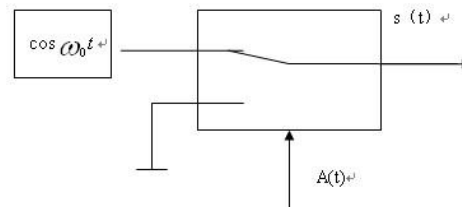


图 2 (b) 开关法

在接收端，2ASK 有两种基本的解调方法：非相干解调（包络检波法）和相干解调（同步检测法），相应的接收系统方框图如图：

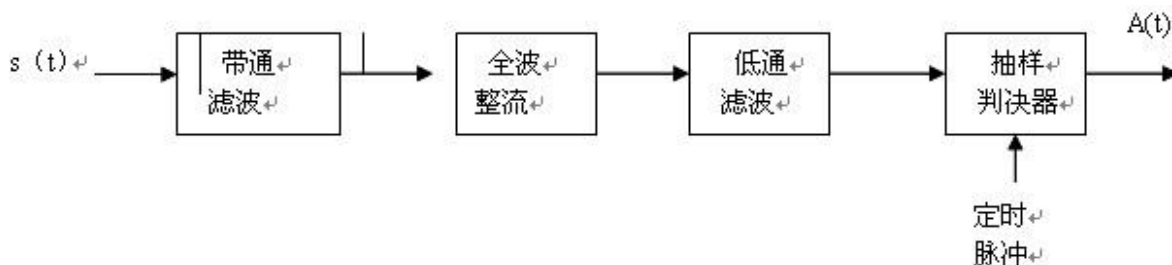


图 3 (a) 非相干解调（包络检波）

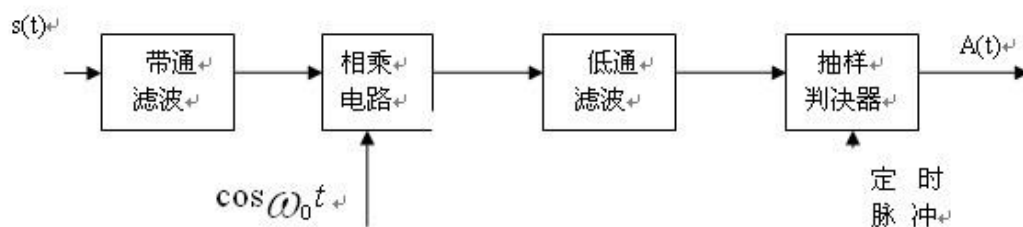


图 3 (b) 相干解调

8.1.2 程序

```
i=10;%10 个码元
j=5000;
t=linspace(0,5,j);%0-5 之间产生 5000 个点行矢量，即将[0,5]分成 5000 份
fc=10;%载波频率
fm=i/5;%码元速率
%产生基带信号
x=(rand(1,i));%rand 函数产生在 0-1 之间随机数，共 1-10 个
%figure(2);plot(x);
a=round(x);%随机序列，round 取最接近小数的整数
%>0.5 的值就为 1，<0.5 的值就为 0
%figure(3);stem(a);%火柴梗状图
st=t;
for n=1:10
    if a(n)<1
        disp(j/i*(n-1))
        for m=j/i*(n-1)+1:j/i*n %a(1)是 1 的话,就将 0-1 赋值为 1
            st(m)=0;
        end
    else
        for m=j/i*(n-1)+1:j/i*n
            st(m)=1;
        end
    end
end
figure(1);
subplot(421);
plot(t,st);
axis([0,5,-1,2]);
```

```

title('基带信号 st');
%载波
s1=cos(2*pi*fc*t);
subplot(422);
plot(s1);
title('载波信号 s1');
%调制
e_2ask=st.*s1;%st 是基带信号,s1 是载波
subplot(423);
plot(t,e_2ask);
title('已调信号');
noise =rand(1,j);
e_2ask=e_2ask+noise;%加入噪声
subplot(424);
plot(t,e_2ask);
title('加入噪声的信号');
%相干解调
at=e_2ask.*cos(2*pi*fc*t);%这里用的 cos 必须和载波 s1 完全同步
%subplot(428);plot(t,at);
at=at-mean(at);%因为是单极性波形，还有直流分量，应去掉
subplot(425);
plot(t,at);
title('与载波相乘后信号');
[f,af] = T2F(t,at);%通过低通滤波器
[t,at] = lpf(f,af,2*fm);
subplot(426);
plot(t,at);
title('相干解调后波形');
%抽样判决

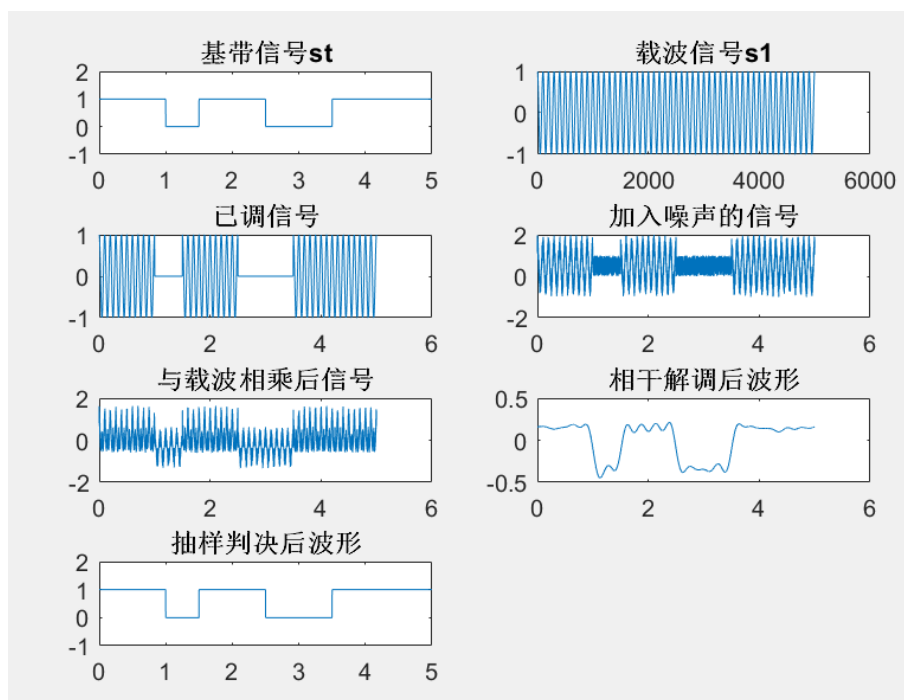
```

```

for m=0:i-1 %i=10    i 是码元个数
    if (at(1,m*500+250)+0.5)<0.5%500 是 1 个码元的长度，+250 就是正好每
        次都定位到每个码元中部
            for j=m*500+1:(m+1)*500%如果判决这位码元的值<0.5,那么这个码元
                判为 0
                    at(1,j)=0;
                end
            else
                for j=m*500+1:(m+1)*500
                    at(1,j)=1;%否则判为 1
                end
            end
        end
    end
end
subplot(4,2,7);
plot(t,at);
axis([0,5,-1,2]);
title('抽样判决后波形')

```

8.1.3 结果



8.2 2FSK

8.2.1 2FSK 原理

二进制频移键控信号码元的“1”和“0”分别用两个不同频率的正弦波形来传送，而其振幅和初始相位不变。故其表达式为：

$$s(t) = \begin{cases} A \cos(\omega_1 t + \varphi_n), & \text{发送“1”时} \\ A \cos(\omega_2 t + \varphi_n), & \text{发送“0”时} \end{cases}$$

典型波形：

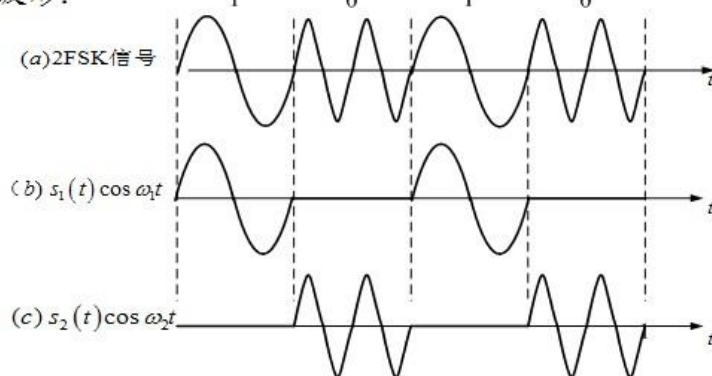


图 4 2FSK 信号时间波形

由图可见，2FSK 信号的波形 (a) 可以分解为波形 (b) 和波形 (c)，也就是说，一个 2FSK 信号可以看成是两个不同载频的 2ASK 信号的叠加。

2FSK 信号的调制方法主要有两种。第一种是用二进制基带矩形脉冲信号去调制一个调频器，使其能够输出两个不同频率的码元。第二种方法是用一个受基带脉冲控制的开关电路去选择两个独立频率源的振荡作为输出。

2FSK 信号的接收也分为相关和非相关接收两类。

相关接收根据已调信号由两个载波 f_1 、 f_2 调制而成，则先用两个分别对 f_1 、 f_2 带通的滤波器对已调信号进行滤波，然后再分别将滤波后的信号与相应的载波 f_1 、 f_2 相乘进行相干解调，再分别低通滤波、用抽样信号进行抽样判决器即可。原理图如下：

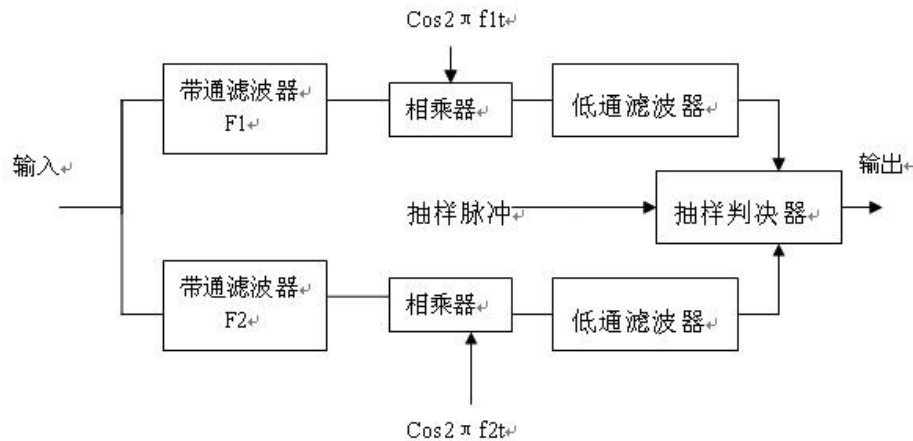


图 5 (a) 相干方式

非相关接收经过调制后的 2FSK 数字信号通过两个频率不同的带通滤波器 f_1 、 f_2 滤出不需要的信号，然后再将这两种经过滤波的信号分别通过包络检波器检波，最后将两种信号同时输入到抽样判决器同时外加抽样脉冲，最后解调出来的信号就是调制前的输入信号。其原理图如下图所示：

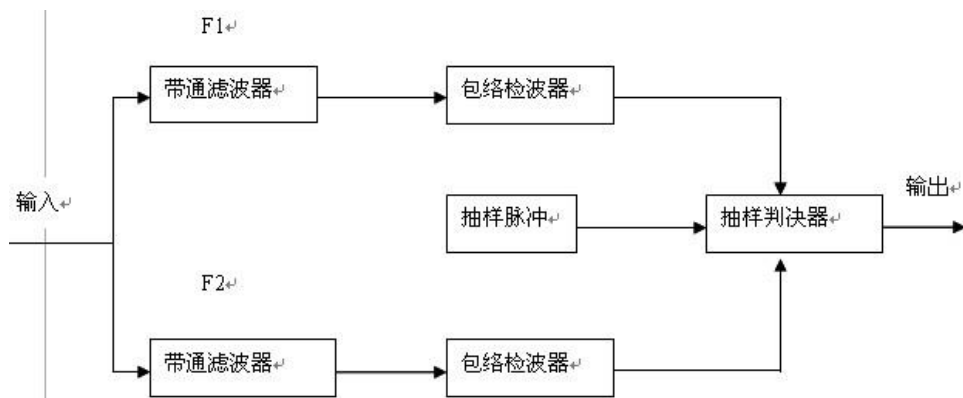


图 5 (b) 非相干方式

8.2.2 程序

```
i=10;%基带信号码元数
```

```
j=5000;
```

```
t=linspace(0,5,j);%0-5 之间产生 5000 个点行矢量，即将[0,5]分成 5000 份
```

```
f1=10;%载波 1 频率
```

```
f2=5;%载波 2 频率
```

```
fm=i/5;%基带信号频率    码元数是 10，而时域长度是 5，也就是一个单位 2 个码元
```

```
a=round(rand(1,i));%产生随机序列
```



```

%产生基带信号
st1=t;
for n=1:10
    if a(n)<1
        for m=j/i*(n-1)+1:j/i*n
            st1(m)=0;
        end
    else
        for m=j/i*(n-1)+1:j/i*n
            st1(m)=1;
        end
    end
end
end
figure(1);
subplot(411);
plot(t,st1);
title('基带信号 st1');
axis([0,5,-1,2]);
%基带信号求反
st2=t;
for n=1:j
    if st1(n)==1
        st2(n)=0;
    else
        st2(n)=1;
    end
end
end
subplot(412);
plot(t,st2);

```

```

title('基带信号反码 st2');
axis([0,5,-1,2]);
%载波信号
s1=cos(2*pi*f1*t);
s2=cos(2*pi*f2*t);
subplot(413),plot(s1);
title('载波信号 s1');
subplot(414),plot(s2);
title('载波信号 s2');
%调制
F1=st1.*s1;%加入载波 1
F2=st2.*s2;%加入载波 2
figure(2);
subplot(411);
plot(t,F1);
title('F1=s1*st1');
subplot(412);
plot(t,F2);
title('F2=s2*st2');
e_fsk=F1+F2;
subplot(413);
plot(t,e_fsk);
title('2FSK 信号');%键控法产生的信号在相邻码元之间相位不一定连续
%加噪
nosie=rand(1,j);
fsk=e_fsk+nosie;
subplot(414);
plot(t,fsk);
title('加噪声后信号')

```

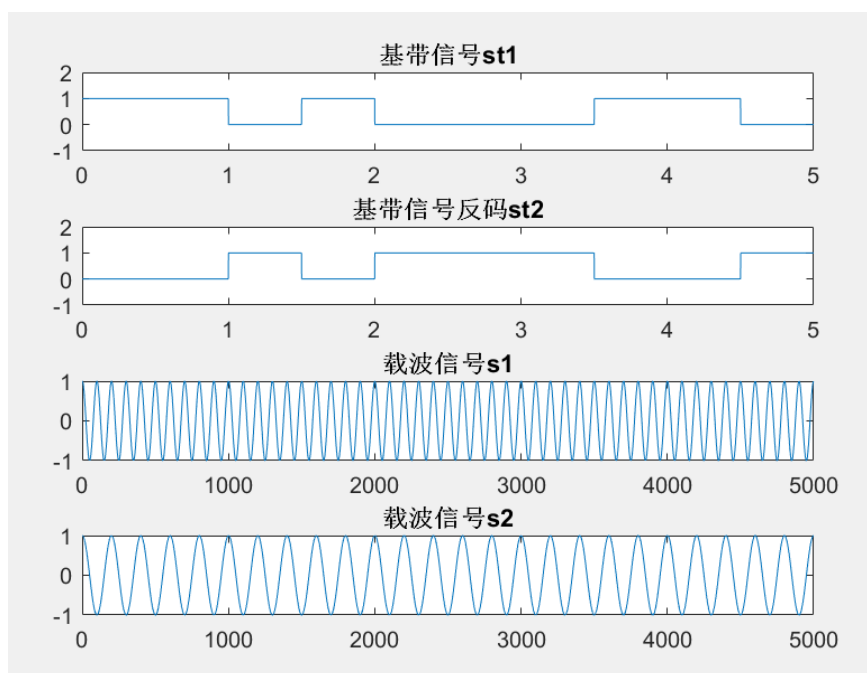
```

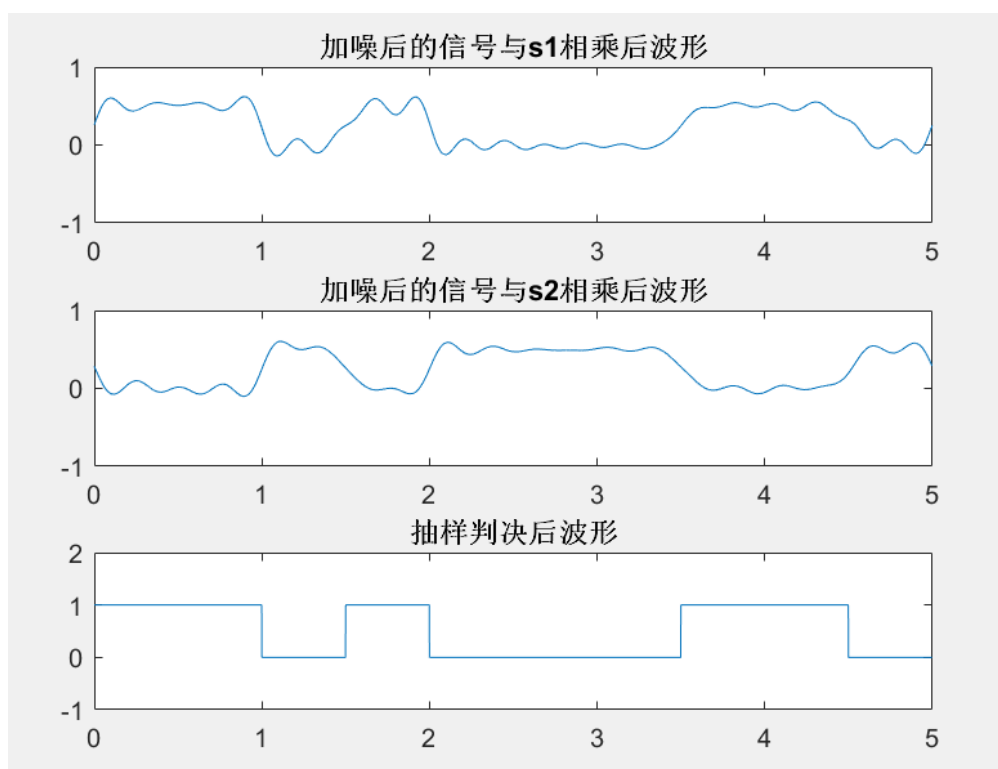
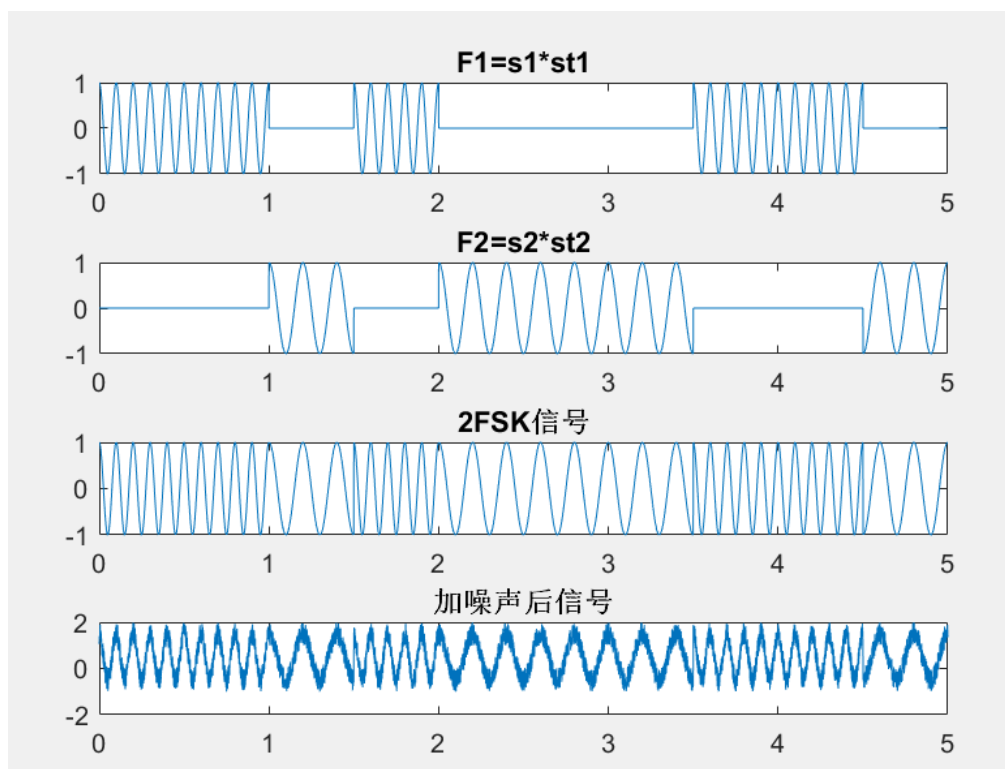
%相干解调
st1=fsk.*s1; %与载波 1 相乘
[f,sf1] = T2F(t,st1);%傅里叶变换
[t,st1] = lpf(f,sf1,2*fm);%通过低通滤波器
figure(3);
subplot(311);
plot(t,st1);
title('加噪后的信号与 s1 相乘后波形');
st2=fsk.*s2;%与载波 2 相乘
[f,sf2] = T2F(t,st2);%通过低通滤波器
[t,st2] = lpf(f,sf2,2*fm);
subplot(312);
plot(t,st2);
title('加噪后的信号与 s2 相乘后波形');
%抽样判决
for m=0:i-1
    if st1(1,m*500+250)>st2(1,m*500+250)
        for j=m*500+1:(m+1)*500
            at(1,j)=1;
        end
    else
        for j=m*500+1:(m+1)*500
            at(1,j)=0;
        end
    end
end
end
subplot(313);
plot(t,at);
axis([0,5,-1,2]);

```

title('抽样判决后波形')

8.2.3 结果





8.3 2PSK

8.3.1 2PSK 原理

由于 2PSK 的幅度是恒定的，必须进行相干解调。经过带通滤波的信号在相乘器中与本地载波相乘，然后用低通滤波器滤除高频分量，在进行抽样判决。判决器是按极性来判决的。即正抽样值判为 1，负抽样值判为 0。2PSK 信号的相干解调原理图如图 2-20 所示，各点的波形如图 2-21 所示。

由于 2PSK 信号的载波回复过程中存在着 180° 的相位模糊，即恢复的本地载波与所需相干载波可能相同，也可能相反，这种相位关系的不确定性将会造成解调出的数字基带信号与发送的基带信号正好相反，即“1”变成“0”吗“0”变成“1”，判决器输出数字信号全部出错。这种现象称为 2PSK 方式的“倒 π ”现象或“反相工作”。但在本次仿真中是直接给其同频同相的载波信号，所以不存在此问题

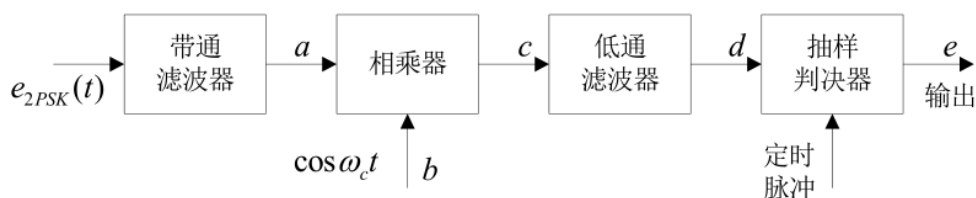


图 2-20 2PSK 的相干解调原理图

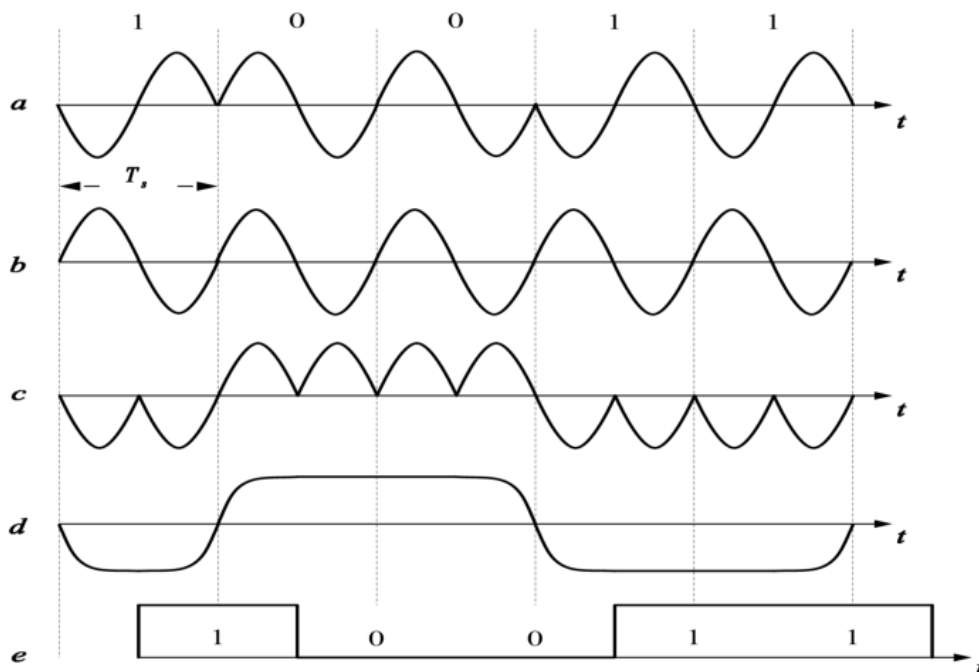


图 2-21 相干解调中各点波形图

8.3.2 程序

```
i=10;%基带信号码元数
j=5000;
t=linspace(0,5,j);%0-5 之间产生 5000 个点行矢量，即将[0,5]分成 5000 份
fc=5;%载波频率
fm=i/5;%码元速率
B=2*fm;%信号带宽
%产生基带信号
a=round(rand(1,i));%随机序列,基带信号
%figure(3);stem(a);
st1=t;
for n=1:10
    if a(n)<1
        for m=j/i*(n-1)+1:j/i*n
            st1(m)=0;
        end
    else
        for m=j/i*(n-1)+1:j/i*n
            st1(m)=1;
        end
    end
end
end
figure(1);
subplot(411);
plot(t,st1);
title('基带信号 st1');
axis([0,5,-1,2]);
%基带信号求反
```

%由于 PSK 中的是双极性信号，因此对上面所求单极性信号取反来与之一起构成双极性码

```
st2=t;
for k=1:j
    if st1(k)>=1
        st2(k)=0;
    else
        st2(k)=1;
    end
end
subplot(412);
plot(t,st2);
title('基带信号反码 st2');
axis([0,5,-1,2]);
st3=st1-st2;
subplot(413);
plot(t,st3);
title('双极性基带信号 st3');
axis([0,5,-2,2]);
%载波信号
s1=sin(2*pi*fc*t);
subplot(414);
plot(s1);
title('载波信号 s1');
%调制
e_psk=st3.*s1;
figure(2);
subplot(511);
plot(t,e_psk);
```



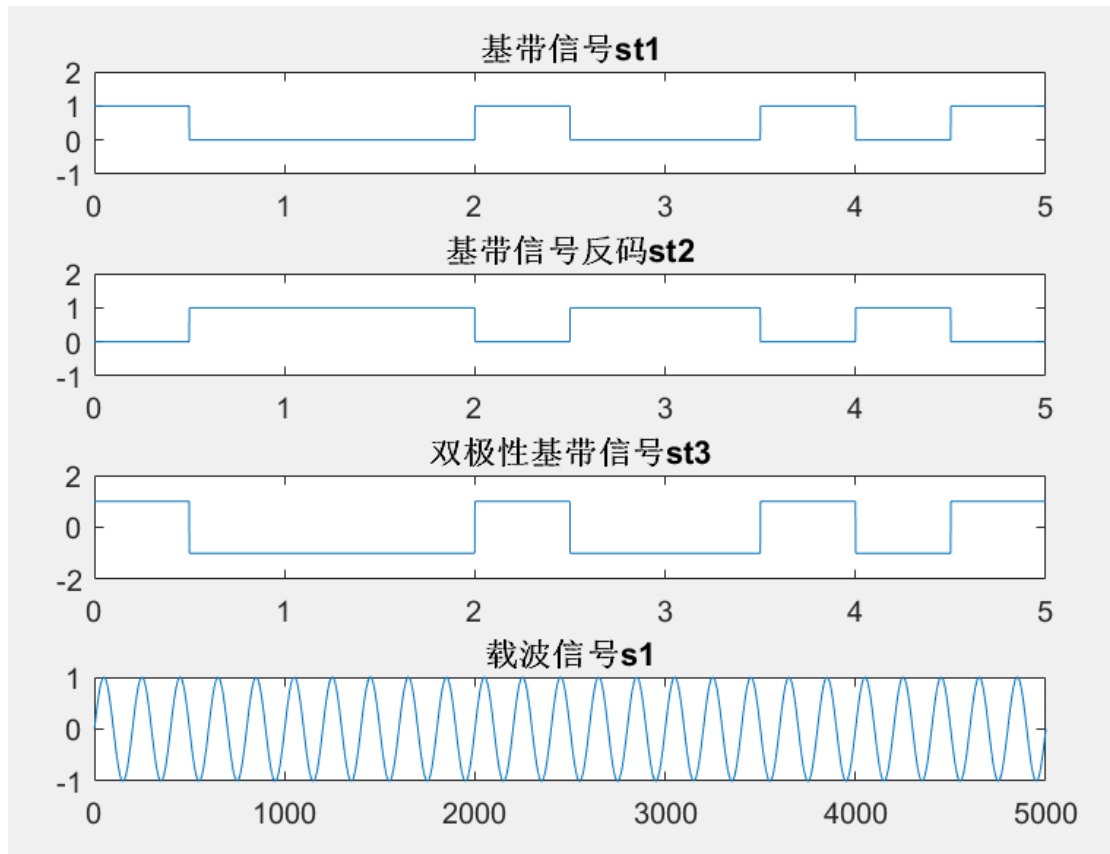
```

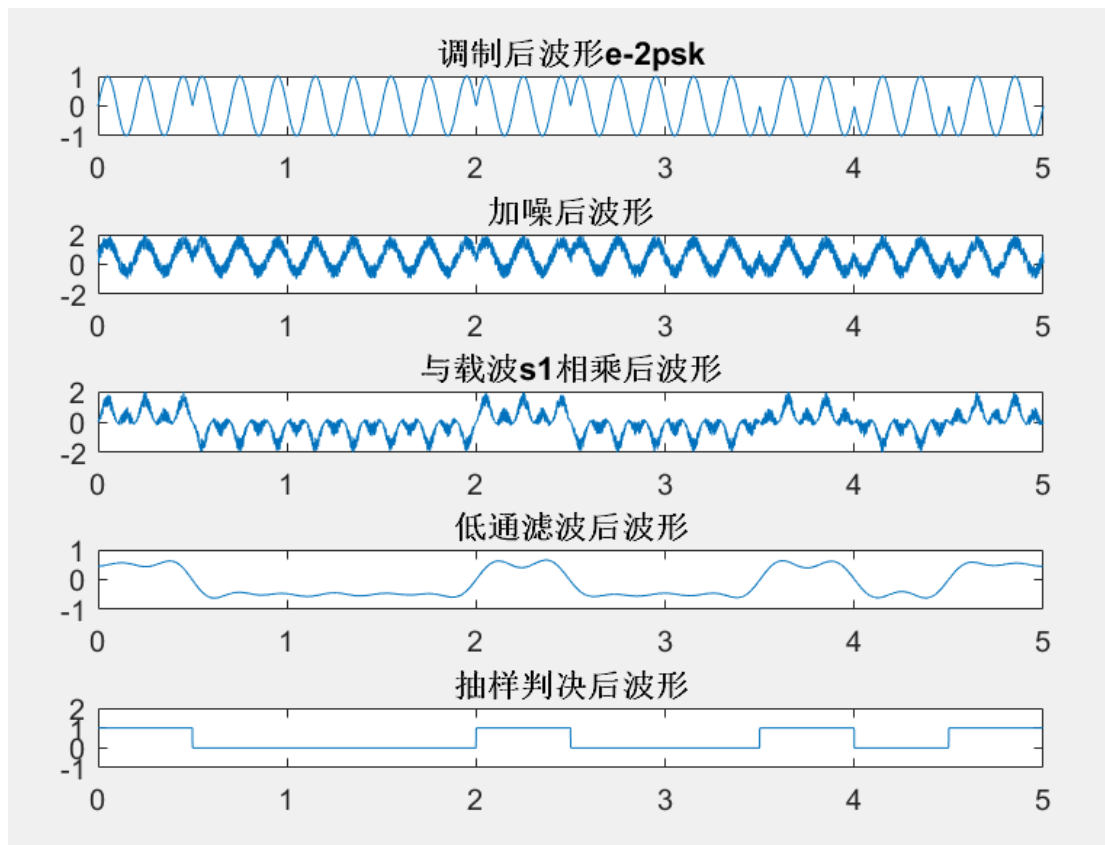
title('调制后波形 e-2psk');
%加噪
noise=rand(1,j);
psk=e_psk+noise;%加入噪声
subplot(512);
plot(t,psk);
title('加噪后波形');
%相干解调
psk=psk.*s1;%与载波相乘
subplot(513);
plot(t,psk);
title('与载波 s1 相乘后波形');
[f,af] = T2F(t,psk);%傅里叶变换
[t,psk] = lpf(f,af,B);%通过低通滤波器
subplot(514);
plot(t,psk);
title('低通滤波后波形');
%抽样判决
for m=0:i-1
    if psk(1,m*500+250)<0
        for j=m*500+1:(m+1)*500
            psk(1,j)=0;
        end
    else
        for j=m*500+1:(m+1)*500
            psk(1,j)=1;
        end
    end
end
end
subplot(515);

```

```
plot(t,psk);  
axis([0,5,-1,2]);  
title('抽样判决后波形');
```

8. 3. 2 结果





8.4 2DPSK

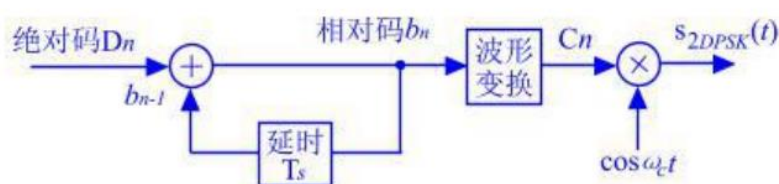
8.4.1 2DPSK 原理

二进制差分相移键控。2DPSK 方式是用前后相邻码元的载波相对相位变化来表示数字信息。假设前后相邻码元的载波相位差为 $\Delta\varphi$ ，可定义一种数字信息与 $\Delta\varphi$ 之间的关系为：

$$\Delta\varphi = \begin{cases} 0 & (\text{数字信息“0”}) \\ \pi & (\text{数字信息“1”}) \end{cases}$$

φ 为前一码元的相位。

实现二进制差分相移键控的最常用的方法是：先对二进制数字基带信号进行差分编码，然后对差分码进行绝对调 2DPSK 调制原理图所示。



变换出的
行绝对调
2DPSK 调
如图 2-25

8.4.2 程序

```
i=10;
j=5000;
t=linspace(0,5,j);%0-5 之间产生 5000 个点行矢量，即将[0,5]分成 5000 份
```

```

fc=5;%载波频率
fm=i/5;%码元速率
B=2*fm;%信号带宽
%产生基带信号
a=round(rand(1,i));
%figure(4);stem(a);
st1=t;
for n=1:10
    if a(n)<1;
        for m=j/i*(n-1)+1:j/i*n
            st1(m)=0;
        end
    else
        for m=j/i*(n-1)+1:j/i*n
            st1(m)=1;
        end
    end
end
figure(1);
subplot(321);
plot(t,st1);
title('绝对码');
axis([0,5,-1,2]);
%差分变换
%设 0 为参考位
b=zeros(1,i);%全零矩阵
if(a(1)==0)
    b(1)=0;
else
    b(1)=1;
end
for n=2:10
    if a(n)==b(n-1)
        b(n)=0;
    else
        b(n)=1;
    end
end
st1=t;
for n=1:10
    if b(n)==0
        for m=j/i*(n-1)+1:j/i*n
            st1(m)=0;
        end
    end
end

```

```

        else
            for m=j/i*(n-1)+1:j/i*n
                st1(m)=1;
            end
        end
    end
end
subplot(323);
plot(t,st1);
title('相对码 st1');
axis([0,5,-1,2]);
st2=t;
for k=1:j;
    if st1(k)==1;
        st2(k)=0;
    else
        st2(k)=1;
    end
end;
subplot(324);
plot(t,st2);
title('相对码反码 st2');
axis([0,5,-1,2]);
%载波信号
s1=sin(2*pi*fc*t);
subplot(325);
plot(s1);
title('载波信号 s1');
s2=sin(2*pi*fc*t+pi);%移了一个相位
subplot(326);
plot(s2);
title('载波信号 s2');
%信号调制
d1=st1.*s1;
d2=st2.*s2;
figure(2);
subplot(411);
plot(t,d1);
title('st1*s1');
subplot(412);
plot(t,d2);
title('st2*s2');
e_dpsk=d1+d2;
subplot(413);
plot(t,e_dpsk);

```

```

title('调制后波形');
%加噪
noise=rand(1,j);
dpsk=e_dpsk+noise;%加入噪声
subplot(414);
plot(t,dpsk);
title('加噪声后信号');
%相干解调
dpsk=dpsk.*s1;%与载波 s1 相乘
figure(3);
subplot(411);
plot(t,dpsk);
title('与载波 s1 相乘后波形');
[f,af]=T2F(t,dpsk);%傅里叶变换
[t,dpsk]=lpf(f,af,B);%通过低通滤波器,滤除部分噪声
subplot(412);
plot(t,dpsk);
title('低通滤波后波形');
%抽样判决
%正值判成 1, 负值判成 0
st=zeros(1,i);%%全零矩阵
for m=0:i-1
    if dpsk(1,m*500+250)<0
        st(m+1)=0;
        for j=m*500+1:(m+1)*500
            dpsk(1,j)=0;
        end
    else
        for j=m*500+1:(m+1)*500
            st(m+1)=1;
            dpsk(1,j)=1;
        end
    end
end
end
subplot(413);
plot(t,dpsk);
axis([0,5,-1,2]);
title('抽样判决后波形')
%码反变换 2DPSK 特有
dt=zeros(1,i);%%全零矩阵
dt(1)=st(1);
for n=2:10;
    % if (st(n)-st(n-1))<=0&&(st(n)-st(n-1))>-1;
    if (st(n)~=st(n-1))

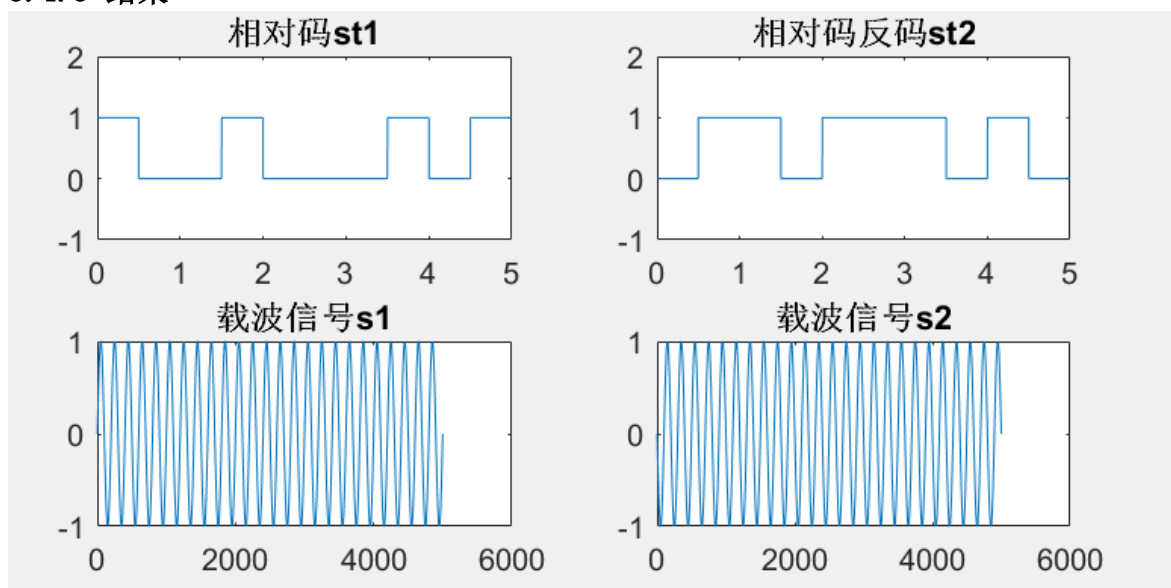
```

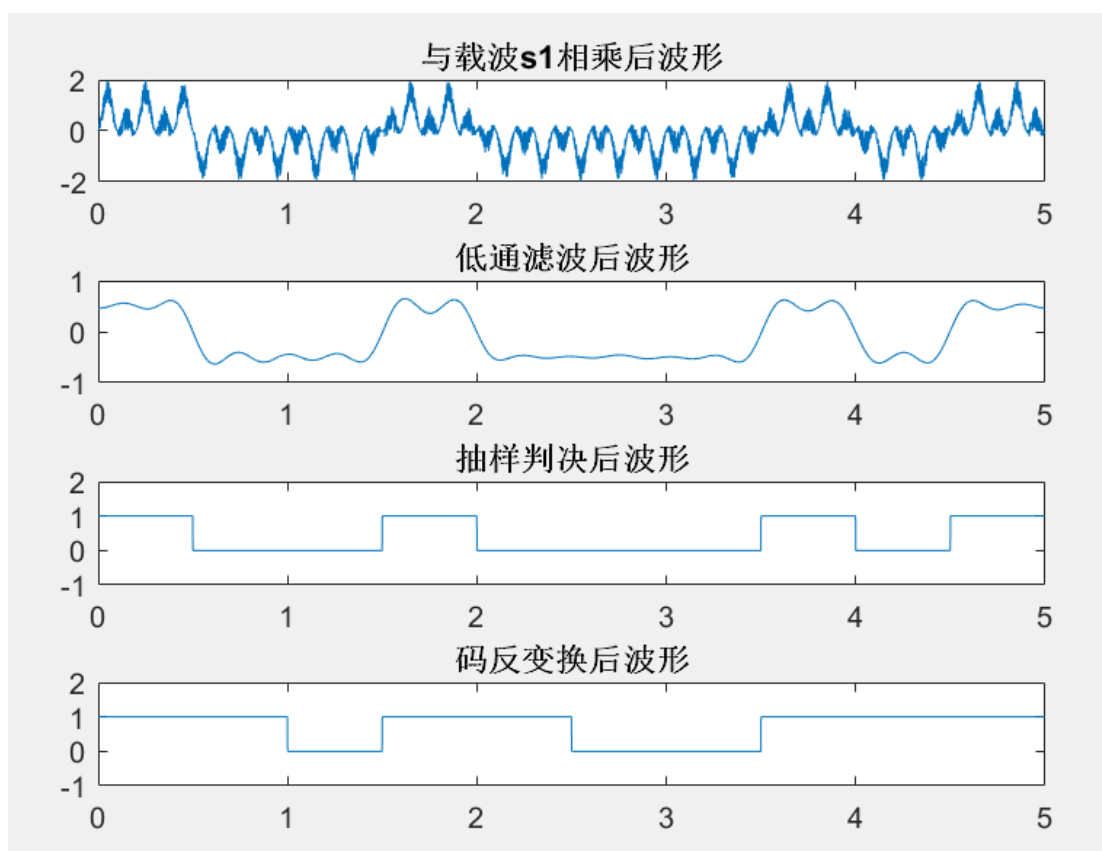
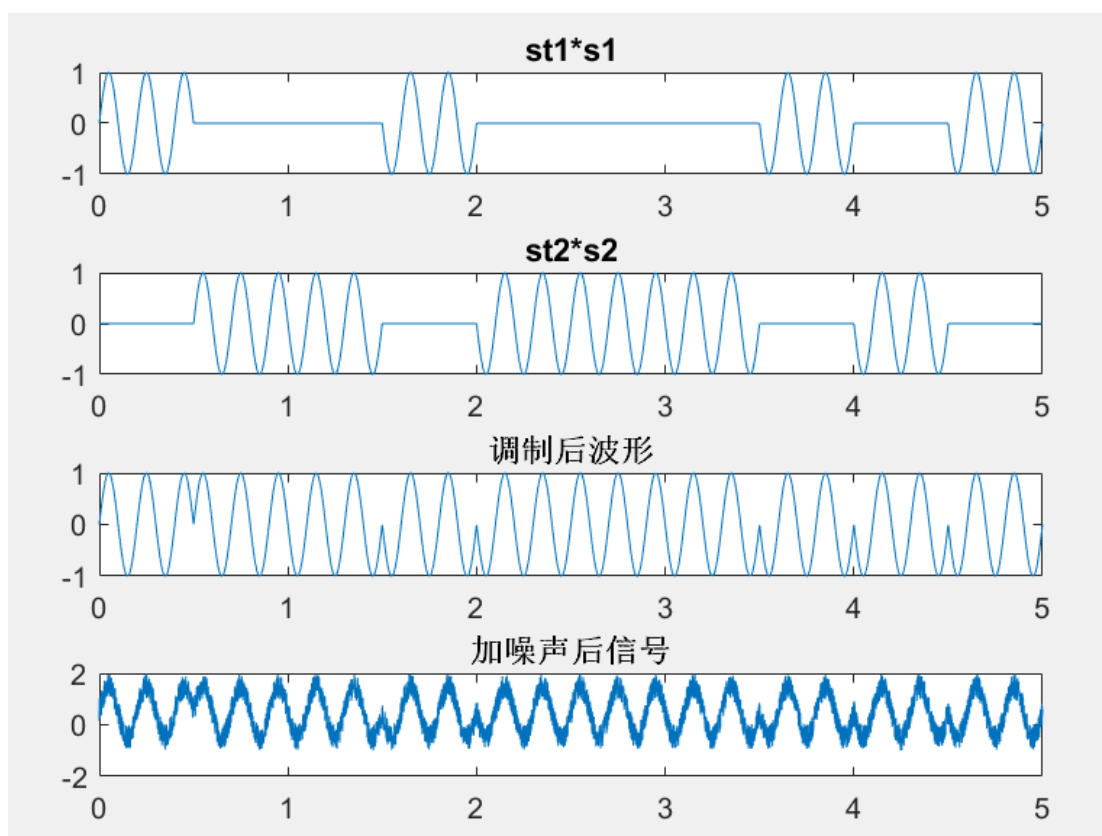
```

        dt(n)=1;
    else
        dt(n)=0;
    end
end
st=t;
for n=1:10
    if dt(n)<1;
        for m=j/i*(n-1)+1:j/i*n
            st(m)=0;
        end
    else
        for m=j/i*(n-1)+1:j/i*n
            st(m)=1;
        end
    end
end
end
subplot(414);
plot(t,st);
axis([0,5,-1,2]);
title('码反变换后波形');

```

8. 4. 3 结果





9. 用 MATLAB 解微分方程

9.1.1 求解微分方程 $\frac{dx}{dt} = 1 + x^2$

```
ans=dsolve('Dx=1+x^2')
```

结果:

```
>>
```

```
ans =
```

```
tan(C2 + t)
      li
      -li
```

9.1.2 求解微分方程 $y' + 2xy = xe^{-x^2}$

```
syms x y;
y=dsolve('Dy+2*x*y=x*exp(-x^2)', 'x')
```

结果:

```
>>
```

```
y =
C5*exp(-x^2) + (x^2*exp(-x^2))/2
```

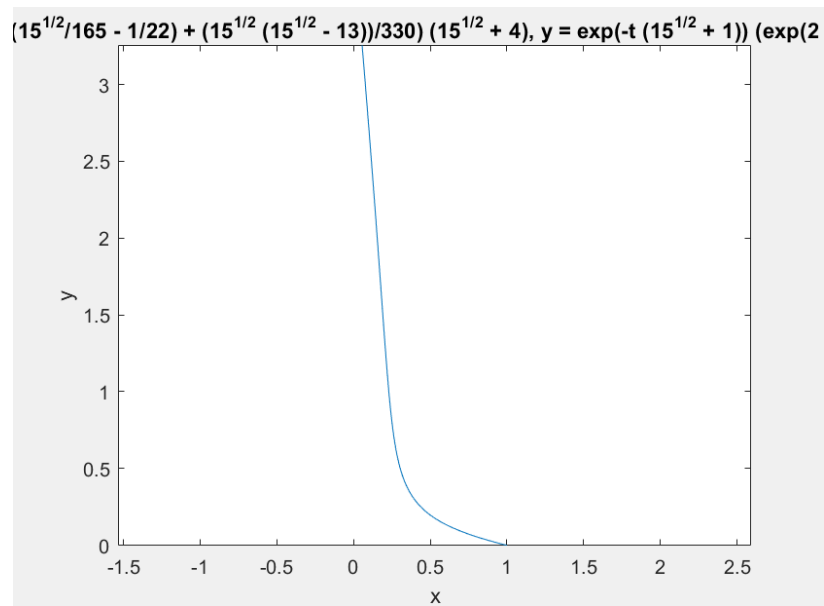
9.1.3 求

$$\begin{cases} \frac{dx}{dt} + 5x + y = e^t \\ \frac{dy}{dt} - 3x - 3y = 0 \\ x(0) = 1, y(0) = 0 \end{cases}$$

并画出函数图像

```
syms x y t;
[x,y]=dsolve('Dx+5*x+y=exp(t)', 'Dy-x-3*y=0', 'x(0)=1', 'y(0)=0', 't');
simplify(x);
simplify(y);
ezplot(x,y,[0 1]) ;
```

结果:

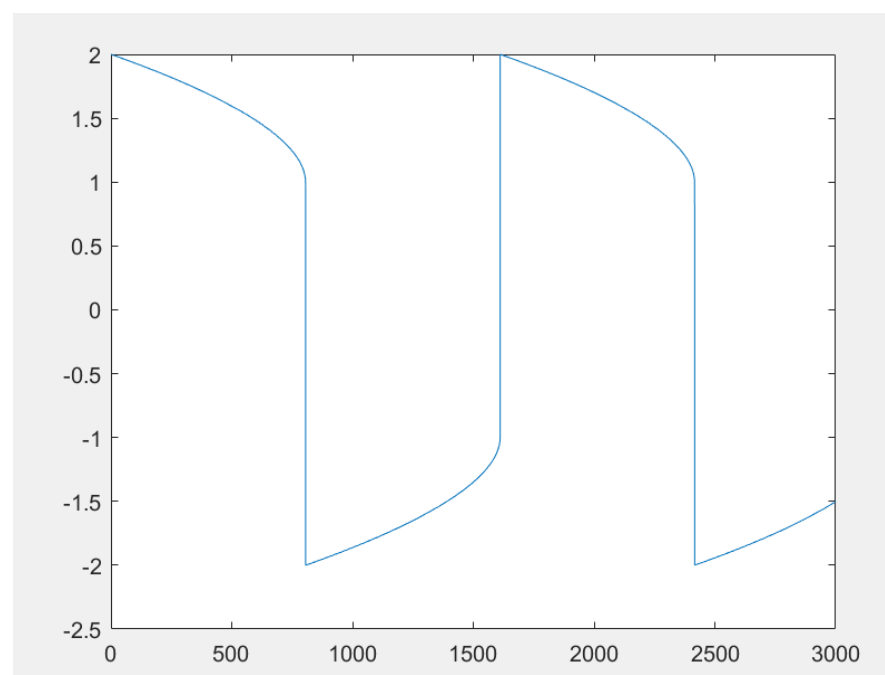


9.1.4 求

$$\begin{cases} \frac{d^2x}{dt^2} - 1000(1 - x^2) \frac{dx}{dt} = 0 \\ x(0) = 2; x'(0) = 0 \end{cases}$$

```
vdp=@(t,y)[y(2), 1000*(1-y(1)^2)*y(2)-y(1)]';
[t,y]=ode15s(vdp,[0 3000],[2 0]);
plot(t,y(:,1))
```

结果:



10. MATLAB 实践能力检测作答

10.1 在 MATLAB 中如何建立矩阵 $\begin{bmatrix} 5 & 7 & 3 \\ 4 & 9 & 1 \end{bmatrix}$ ，并将其赋予变量 a?

```
>> a=[5 7 3;4 9 1]
```

a =

```
5     7     3
4     9     1
```

10.2 计算矩阵 $\begin{bmatrix} 5 & 3 & 5 \\ 3 & 7 & 4 \\ 7 & 9 & 8 \end{bmatrix}$ 与 $\begin{bmatrix} 2 & 4 & 2 \\ 6 & 7 & 9 \\ 8 & 3 & 6 \end{bmatrix}$ 之和。

```
> a=[5 3 5;3 7 4;7 9 8];
```

```
b=[2 4 2;6 7 9;8 3 6];
```

```
c=a+b
```

c =

```
7     7     7
9    14    13
15    12    14
```

10.3 求 $x = \begin{bmatrix} 4+8i & 3+5i & 2-7i & 1+4i & 7-5i \\ 3+2i & 7-6i & 9+4i & 3-9i & 4+4i \end{bmatrix}$ 的共轭转置。

```
>> x=[4+8i 3+5i 2-7i 1+4i 7-5i;3+2i 7-6i 9+4i 3-9i 4+4i]
```

```
>> x'
```

ans =

```
4.0000 - 8.0000i    3.0000 - 2.0000i
3.0000 - 5.0000i    7.0000 + 6.0000i
2.0000 + 7.0000i    9.0000 - 4.0000i
1.0000 - 4.0000i    3.0000 + 9.0000i
7.0000 + 5.0000i    4.0000 - 4.0000i
```

10.4 计算 $a = \begin{bmatrix} 6 & 9 & 3 \\ 2 & 7 & 5 \end{bmatrix}$ 与 $b = \begin{bmatrix} 2 & 4 & 1 \\ 4 & 6 & 8 \end{bmatrix}$ 的数组乘积。

```
>> a=[6 9 3;2 7 5],b=[2 4 1;4 6 8]
```

```
>> a.*b
```

ans =

```
12    36     3
8     42    40
```

10.5 对于 $AX=B$ ，如果 $A=\begin{bmatrix} 4 & 9 & 2 \\ 7 & 6 & 4 \\ 3 & 5 & 7 \end{bmatrix}$ ， $B=\begin{bmatrix} 37 \\ 26 \\ 28 \end{bmatrix}$ ，求解 X 。

```
>> A=[4 9 2;7 6 4;3 5 7],B=[37;26;28],AX=B
```

```
>> A\B
```

```
ans =
```

```
-0.5118
```

```
4.0427
```

```
1.3318
```

10.6 已知： $a=\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ ，分别计算 a 的数组平方和矩阵平方，并观察其结果。

```
>> a=[1 2 3;4 5 6;7 8 9]
```

```
a =
```

```
1     2     3
```

```
4     5     6
```

```
7     8     9
```

```
>> a.*a
```

```
ans =
```

```
1     4     9
```

```
16    25    36
```

```
49    64    81
```

```
>> a*a
```

```
ans =
```

```
30    36    42
```

```
66    81    96
```

```
102   126   150
```

10.7 $a=\begin{bmatrix} 1 & 2 & 5 \\ 3 & 6 & -4 \end{bmatrix}$ ， $b=\begin{bmatrix} 8 & -7 & 4 \\ 3 & 6 & 2 \end{bmatrix}$ ，观察 a 与 b 之间的六种关系运算的结果。

```
>> a=[1 2 3;3 6 -4];
```

```
>> b=[8 -7 4;3 6 2];
```

```
>> a>b
```

```
ans =
```

```
    0    1    0
    0    0    0
```

```
>> a>=b
```

```
ans =
```

```
    0    1    0
    1    1    0
```

```
>> a<=b
```

```
ans =
```

```
    1    0    1
    1    1    1
```

```
>> a==b
```

```
ans =
```

```
    0    0    0
    1    1    0
```

```
>> a>b
```

```
ans =
```

```
    0    1    0
    0    0    0
```

```
>> a~=b
```

```
ans =
```

```
    1    1    1
    0    0    1
```

10.8 角度 $x=[30 \ 45 \ 60]$, 求 x 的正弦、余弦、正切和余切。

```
>> x=[30 45 60];
```

```
>> y=x/180*pi;
```

```
>> sin(y)
```

```
ans =
```

```
0.5000    0.7071    0.8660
```

```
>> cos(y)
```

```
ans =
```

```
0.8660    0.7071    0.5000
```

```
>> tan(x)
```

```
ans =
```

```
-6.4053    1.6198    0.3200
```

```
>> cot(y)
```

```
ans =
```

```
1.7321    1.0000    0.5774
```

10.9 用四舍五入的方法将数组[2.4568 6.3982 3.9375 8.5042]取整。

```
>> a=[2.4568 6.3982 3.9375 8.5042];
```

```
>> round(a)
```

```
ans =
```

```
2        6        4        9
```

10.10 矩阵 $a = \begin{bmatrix} 9 & 1 & 2 \\ 5 & 6 & 3 \\ 8 & 2 & 7 \end{bmatrix}$, 分别对 a 进行特征值分解、奇异值分解、LU 分解、QR

分解及 Cholllesky 分解。

```
>> a=[9 1 2;5 6 3;8 2 7];
```

```
>> [v,d]=eig(a)
```

```
v =
```

```
-0.4330    -0.2543    -0.1744  
-0.5657     0.9660    -0.6091  
-0.7018     0.0472     0.7736
```

d =

13.5482	0	0
0	4.8303	0
0	0	3.6216

>> [u,s,v]=svd(a)

u =

-0.5601	0.5320	-0.6350
-0.4762	-0.8340	-0.2788
-0.6779	0.1462	0.7204

s =

15.5234	0	0
0	4.5648	0
0	0	3.3446

v =

-0.8275	0.3917	-0.4023
-0.3075	-0.9156	-0.2592
-0.4699	-0.0907	0.8781

>> [l,u]=lu(a)

l =

1.0000	0	0
0.5556	1.0000	0
0.8889	0.2041	1.0000

u =

9.0000	1.0000	2.0000
0	5.4444	1.8889
0	0	4.8367

```
>> [q,r]=qr(a)
```

```
q =
```

```
-0.6903    0.3969   -0.6050  
-0.3835   -0.9097   -0.1592  
-0.6136    0.1221    0.7801
```

```
r =
```

```
-13.0384   -4.2183   -6.8260  
         0   -4.8172   -1.0807  
         0         0    3.7733
```

```
>> c=chol(a)
```

```
c =
```

```
3.0000    0.3333    0.6667  
         0    2.4267    1.1447  
         0         0    2.2903
```

10.11 求解多项式 $x^3-7x^2+2x+40$ 的根。

```
>> k=[1 -7 2 40];
```

```
>> x=roots(k)
```

```
x =
```

```
5.0000  
4.0000  
-2.0000
```

10.12 求解在 $x=8$ 时多项式 $(x-1)(x-2)(x-3)(x-4)$ 的值。

```
>> p=poly([1 2 3 4]);
```

```
>> polyvalm(p,8)
```

```
ans =
```

```
840
```

10.13 计算多项式乘法 $(x^2+2x+2)(x^2+5x+4)$ 。


```
>> conv([1 2 2],[1 5 4])
```

```
ans =
```

```
1      7     16     18     8
```

10.14 计算多项式除法 $(3x^3+13x^2+6x+8)/(x+4)$ 。

```
>> deconv([3 13 6 8],[1 4])
```

```
ans =
```

```
3      1     23
```

10.15 对下式进行部分分式展开：

$$\frac{3x^4 + 2x^3 + 5x^2 + 4x + 6}{x^5 + 3x^4 + 4x^3 + 2x^2 + 7x + 2}$$

```
>> b=[3 2 5 4 6];
```

```
>> a=[1 3 4 2 7 2];
```

```
>> [r,p,k]=residue(b,a)
```

```
r =
```

```
1.1274 + 1.1513i
```

```
1.1274 - 1.1513i
```

```
-0.0232 - 0.0722i
```

```
-0.0232 + 0.0722i
```

```
0.7916 + 0.0000i
```

```
p =
```

```
-1.7680 + 1.2673i
```

```
-1.7680 - 1.2673i
```

```
0.4176 + 1.1130i
```

```
0.4176 - 1.1130i
```

```
-0.2991 + 0.0000i
```

```
k =
```

```
[]
```

10.16 解方程组 $\begin{bmatrix} 2 & 9 & 0 \\ 3 & 4 & 11 \\ 2 & 2 & 6 \end{bmatrix} x = \begin{bmatrix} 13 \\ 6 \\ 6 \end{bmatrix}$ 。

```
>> b=[13;6;6];
```

```
>> a=[2 9 0;3 4 11;2 2 6];
```

```
>> a\b
```

```
ans =
```

```
7.4000
```

```
-0.2000
```

```
-1.4000
```

10.17 矩阵 $a = \begin{bmatrix} 4 & 2 & -6 \\ 7 & 5 & 4 \\ 3 & 4 & 9 \end{bmatrix}$ ，计算 a 的行列式和逆矩阵。

```
>> a=[4 2 -6;7 5 4;3 4 9];
```

```
>> a^-1
```

```
ans =
```

```
-0.4531    0.6562   -0.5937
```

```
0.7969   -0.8437    0.9062
```

```
-0.2031    0.1562   -0.0937
```

```
>> inv(a)
```

```
ans =
```

```
-0.4531    0.6562   -0.5937
```

```
0.7969   -0.8437    0.9062
```

```
-0.2031    0.1562   -0.0937
```

```
>> det(a)
```

```
ans =
```

```
-64
```

10.18 $y = \sin(x)$ ， x 从 0 到 2π ， $\Delta x = 0.02\pi$ ，求 y 的最大值、最小值、均值和标准差。

```
>> x=0:0.02*pi:2*pi;
```

```
>> y=sin(x);
>> ymax=max(y)
```

```
ymax =
```

```
1
```

```
>> ymin=min(y)
```

```
ymin =
```

```
-1
```

```
>> ymean=mean(y)
```

```
ymean =
```

```
2.2995e-17
```

```
>> ystd=std(y)
```

```
ystd =
```

```
0.7071
```

10.19 $x=[1 \ 2 \ 3 \ 4 \ 5]$, $y=[2 \ 4 \ 6 \ 8 \ 10]$, 计算 x 的协方差、 y 的协方差、 x 与 y 的互协方差。

```
x=[1 2 3 4 5]
>> y=[2 4 6 8 10];
>> cx=cov(x)
```

```
cx =
```

```
2.5000
```

```
>> cy=cov(y)
```

```
cy =
```

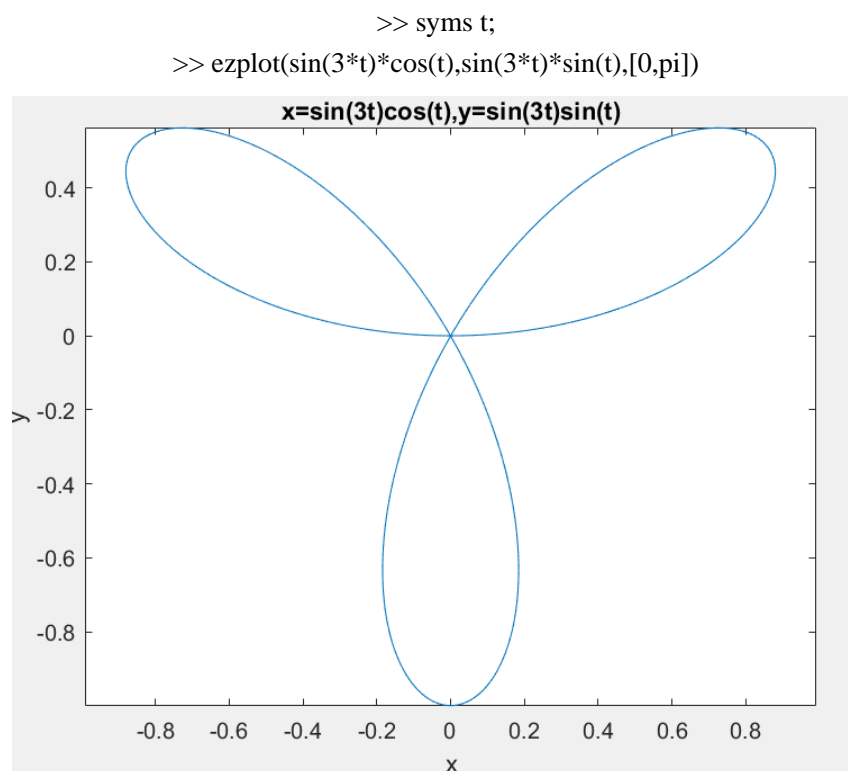
```
10
```

```
>> cxy=cov(x,y)
```

```
cxy =
```

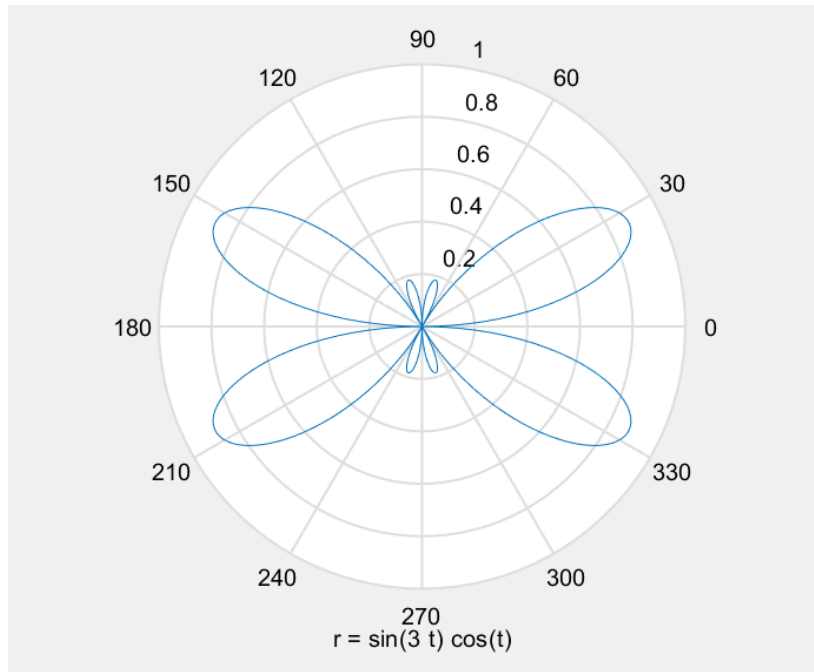
2.5000 5.0000
5.0000 10.0000

10.20 符号函数绘图法绘制函数 $x=\sin(3t)\cos(t)$, $y=\sin(3t)\sin(t)$ 的图形,
 t 的变化范围为 $[0, 2\pi]$ 。



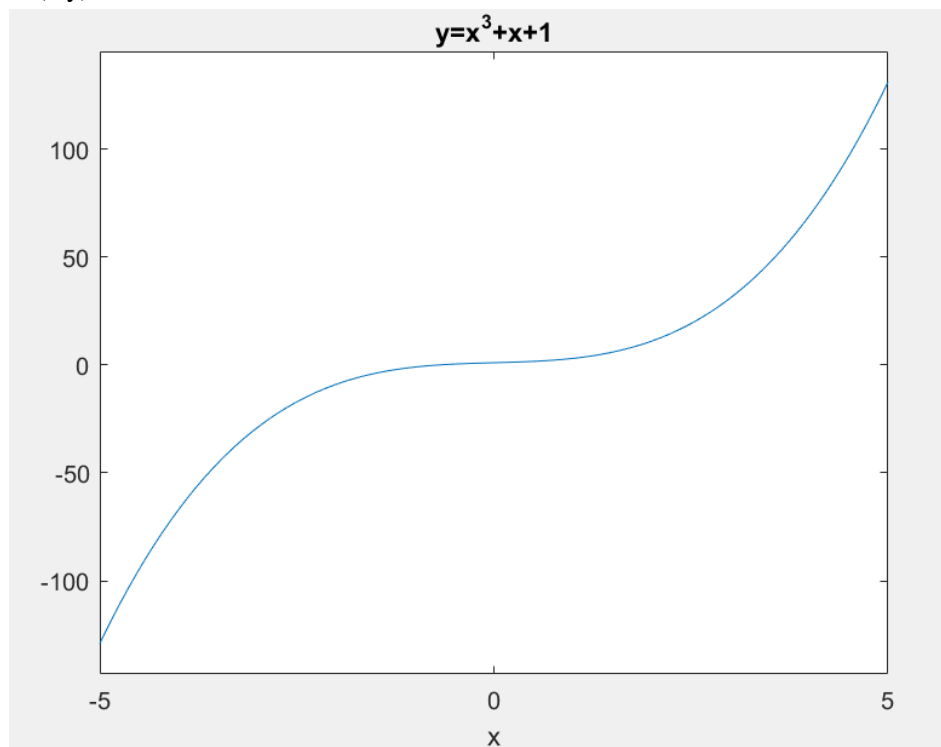
10.21 绘制极坐标下 $\sin(3t)*\cos(t)$ 的图形。

```
>> syms t;  
>> ezpolar(sin(3*t)*cos(t))
```



10.22 绘制曲线 $y = x^3 + x + 1$ ， x 的取值范围为 $[-5, 5]$ 。

```
>> x=-5:0.2:5;
>> y=x.^3+x+1;
>> plot(x,y)
```

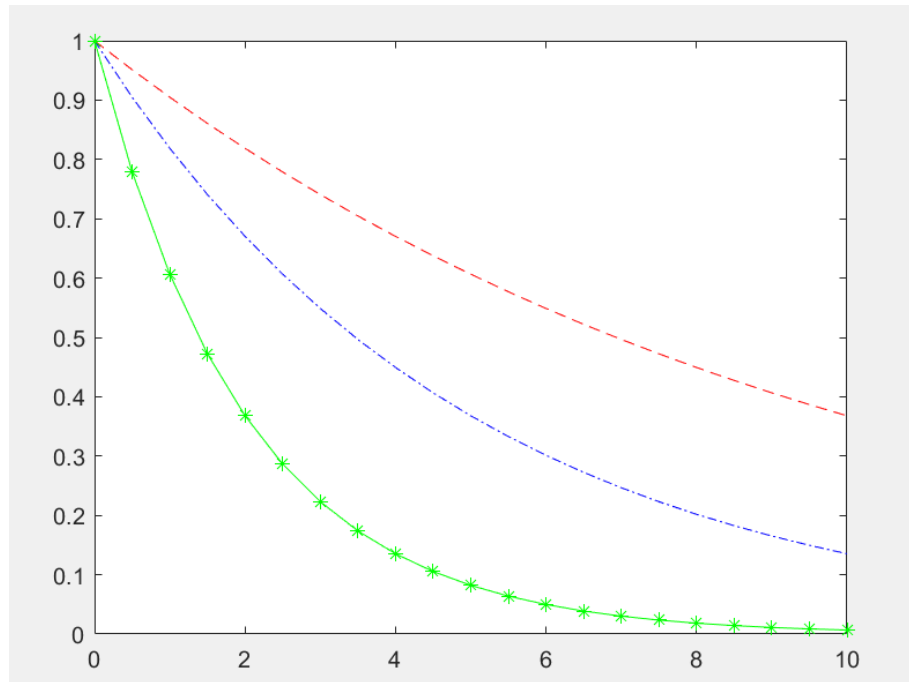


10.23 有一组测量数据满足 $y = e^{-at}$ ， t 的变化范围为 $0 \sim 10$ ，用不同的线型和标记

点画出 $a=0.1$ 、 $a=0.2$ 和 $a=0.5$ 三种情况下的曲线。

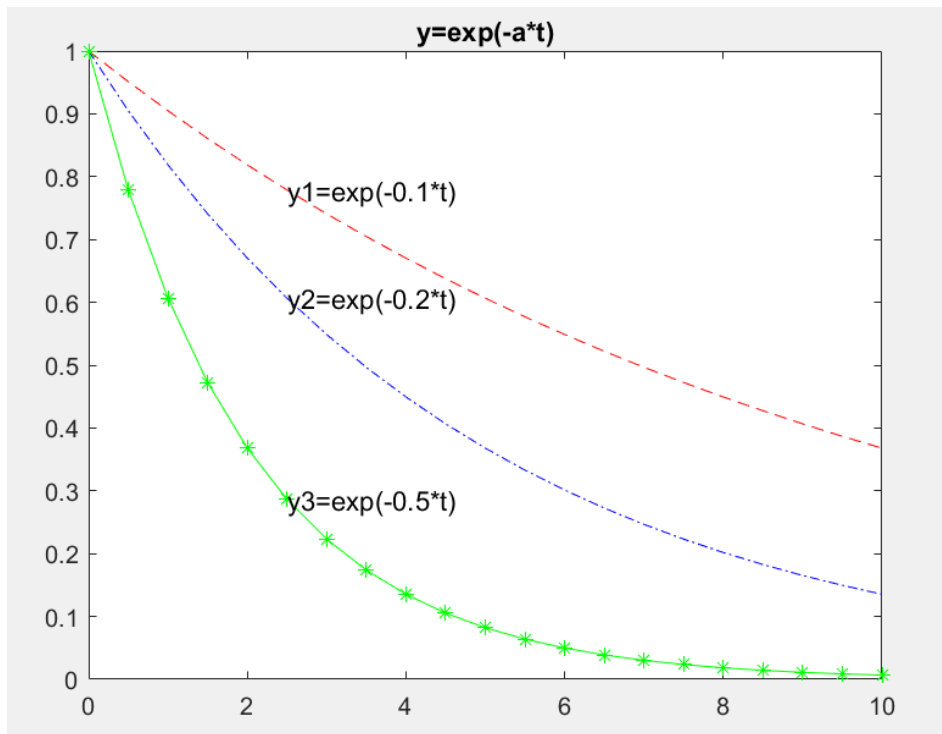
```
>> t=0:0.5:10;
```

```
>> y1=exp(-0.1*t);
>> y2=exp(-0.2*t);
>> y3=exp(-0.5*t);
>> plot(t,y1,'r--',t,y2,'b-.',t,y3,'g-*')
```



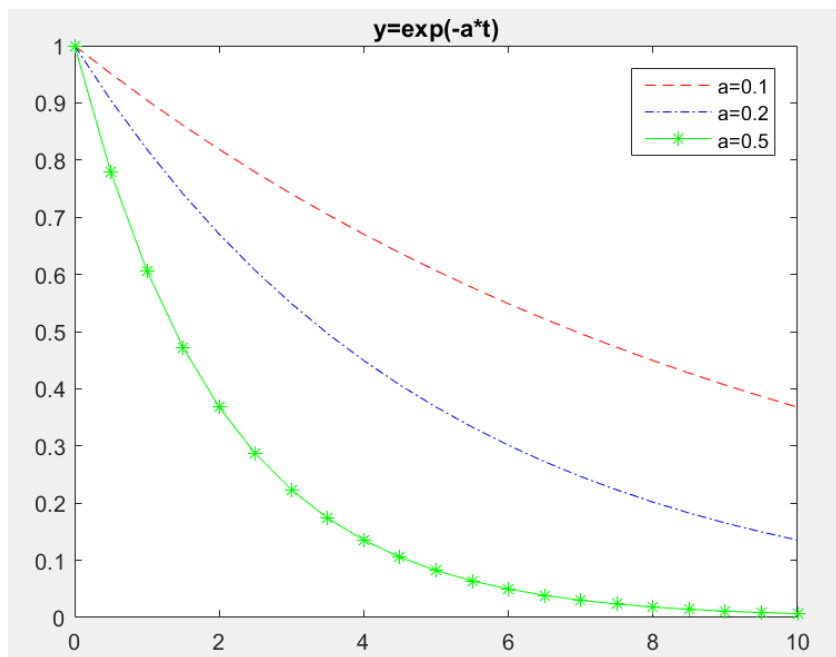
10. 24 在 23 题结果图中添加标题 $y = e^{-at}$ ，并用箭头线标识出各曲线 a 的取值。

```
t=0:0.5:10;
y1=exp(-0.1*t);
y2=exp(-0.2*t);
y3=exp(-0.5*t);
plot(t,y1,'r--',t,y2,'b-.',t,y3,'g-*');
title('y=exp(-a*t)');
text(t(6),y1(6),'y1=exp(-0.1*t)','FontSize',11)
text(t(6),y2(6),'y2=exp(-0.2*t)','FontSize',11)
text(t(6),y3(6),'y3=exp(-0.5*t)','FontSize',11)
```



10. 25 在 23 题结果图中添加标题 $y = e^{-at}$ 和图例框。

```
t=0:0.5:10;
y1=exp(-0.1*t);
y2=exp(-0.2*t);
y3=exp(-0.5*t);
plot(t,y1,'r--',t,y2,'b-.',t,y3,'g-*');
title('y=exp(-a*t)');
legend('a=0.1','a=0.2','a=0.5')
```



11. MATLAB 实验心得

通过这次课程设计，我对 MATLAB 有了一个新的认识。其实在上大学之前我就听说过 MATLAB 的强大。大一时本想自学一段时间，无奈当时基础不好，学起来比较吃力，只好放弃。当时当这次课程设计需要学习 MATLAB 时，有了大学两年的积淀，自学起来就相对容易了。

通过这次 MATLAB 的学习，我对 MATLAB 有了一个基础的认识，MATLAB 是一个实用性很强，操作相对容易，比较完善的工具软件，使用方便，通过操作可以很快看到结果，能够清晰的感觉到成功与失败。matlab 是一个功能强大的数学软件，在工程运算上非常使用且功能强大，并且图形功能非常强大，科学计算与图形功能紧密结合是 MATLAB 的主要特点之一。

自我感觉学习 MATLAB 与其说是学习一门软件，更不如说是学习一门语言。用一种数理的语言描述现象，揭示表象下的规律。此外，我认为 matlab 中的作图功能很强大，不仅简单的函数现象可以明确画出，而且一些点状物，甚至立体图也可以画出。大一上微积分的时候，老师曾经多次在课件中加入用 matlab 画出的图来。不论是一维二维三维等等，都能很好的画出来。只要能编写出函数式，在短短的几秒之内，他就会呈现在你眼前。另外就是图形的直观性，这是由阴影的制作的。而且可以根据需要，坐标图上加标题，坐标轴标记，文本注释级栅格等，也可以指定图线形式，比如是虚线。颜色也可以自己来定。可以在同一张图上画，也可以单个显示。

在学习过程中，我明白了通信仿真的现实意义，系统模型是对实际系统的一种抽象，对系统本质的一种描述。知道了仿真技术的必要性，利用系统建模和软件仿真技术，几乎可以对所有的设计细节进行分层次的建模和评估，迅速构建一个系统模型，使学习更加高效。

参考文献

- [1]刘浩，韩晶，MATLAB 完全自学一本通, 北京:电子工业出版社, 2016
- [2]卢光跃，黄庆东，包志强, 数字信号处理及应用 ，北京:人民邮电出版社, 2012
- [3]樊昌信，曹丽娜，通信原理, 北京:国防工业出版社, 2012

西安邮电大学通信与信息工程学院

信息对抗专业课程设计 I 过程考核表

学生姓名	郝希烜	班级/学号	信息对抗 1602/03166063
承担任务实验室（单位）	信息安全实验室	所在部门	通信与信息工程学院
实施时间	2018 年 9 月 3 日 — 2018 年 9 月 14 日		
具体内容	第一周	熟悉掌握 MATLAB 语言的基本语法和功能，熟悉操作界面，可以完成对 MATLAB 软件的基本操作，学会用 MATLAB 仿真基本的图形。	
	第二周	掌握实验任务的基本仿真流程，了解实验任务，并且学会自己分解细化实验任务，同时，学会查阅一些外语参考资料，了解用 MATLAB 仿真建模一些通信过程。	
指导教师（师傅）姓名		职务或职称	
指导教师（师傅） 对学生的评价	学习态度	<input type="checkbox"/> 认真 <input type="checkbox"/> 一般 <input type="checkbox"/> 不认真	
	学习纪律	<input type="checkbox"/> 全勤 <input type="checkbox"/> 偶尔缺勤 <input type="checkbox"/> 经常缺勤	
	实践能力	<input type="checkbox"/> 很强 <input type="checkbox"/> 一般 <input type="checkbox"/> 较差	
指导教师（师傅）对学生 专业知识或科研实践能力等 情况的意见	<div style="text-align: right;"> 指导教师（师傅）签字 年 月 日 </div>		

西安邮电大学通信与信息工程学院

信息对抗专业课程设计 I 成绩鉴定表

学生姓名	郝希烜	专业班级/学号	信息对抗 1602/03166063
进行时间	2018 年 9 月 3 日 — 2018 年 9 月 14 日		
考核内容	平时成绩 (30 分)	课堂出勤情况、课堂纪律及学习态度 (25 分) 遵守实验室的规章制度情况 (5 分)	
	实践能力 (40 分)	分析、解决问题能力 (10 分) 文件检索或软硬件设计能力 (10 分) 项目完成情况 (20 分)	
	实习报告 (30 分)	语言及文字表达力 (10 分) 实验报告结构 (10 分) 书写格式规范 (10 分)	
指导教师		评定成绩	
成绩对应等级			