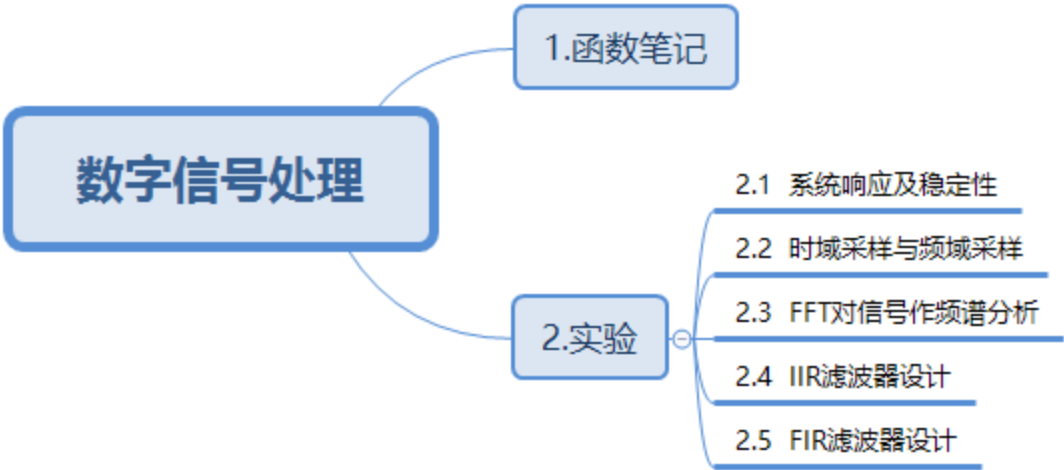




https://blog.csdn.net/weixin_42121843

1.函数笔记

abs	Absolute value and complex magnitude
angle	Phase angle
imag	Imaginary part of complex number
real	Real part of complex number
fft	Fast Fourier transform
ifft	Inverse fast Fourier transform
conv	Convolution and polynomial multiplication
step	Step response plot of dynamic system; step response data
stepinfo	Rise time, settling time, and other step response characteristics
impulse	Impulse response plot of dynamic system; impulse response data
initial	Initial condition response of state-space model
lsim	Simulate time response of dynamic system to arbitrary inputs
lsiminfo	Compute linear response characteristics

tf	Create transfer function model, convert to transfer function model
zpk	Create zero-pole-gain model; convert to zero-pole-gain model
ss	Create state-space model, convert to state-space model
butter	Butterworth filter design
buttord	Butterworth filter order and cutoff frequency
cheby1	Chebyshev Type I filter design
cheb1ord	Chebyshev Type I filter order
cheby2	Chebyshev Type II filter design
cheb2ord	Chebyshev Type II filter order
designfilt	Design digital filters
ellip	Elliptic filter design
ellipord	Minimum order for elliptic filters

fft():快速傅里叶变换

$Y = \text{fft}(X, n, \text{dim})$

计算x的n点DFT，x长度不够n时补0，比n长时截短，dim为对x做傅里叶变换的维数。

ifft():傅里叶反变换

$y = \text{ifft}(X, n)$

计算X的n点DFT反变换

Stem():火柴杆图

$\text{stem}(X, Y)$

显示离散序列，X与Y大小必须相同，一一对应。如Y是一个长度为6的序列，x可以为0:5，若X省略，程序会自动把横坐标变为Y的大小。

若一个连续序列用stem显示，会变成实心的，因为离散序列每个点下都有一条竖线，因此最好用plot显示连续序列。

Abs():绝对值或幅值

$Y = \text{abs}(X)$

求绝对值或复数的幅度，可用于求幅度谱。

Angle():相位

$P = \text{angle}(Z)$

求复数Z的角度和相位，返回值在 $\pm\pi$ 之间。

Conj():求共轭

$ZC = \text{conj}(Z)$

矩形信号：

矩形序列R(N):可以用zeros和ones构造，如： $R20 = [\text{zeros}(1, 40), \text{ones}(1, 20), \text{zeros}(1, 41)]$;

连续矩形信号可用rectpuls()函数， $y = \text{rectpuls}(t, w)$ ，t为时间序列，w为宽度，宽度以t=0为对称

单位脉冲信号：

只需在坐标原点处设一个脉冲即可

Tripuls():三角波信号

$y = \text{tripuls}(T, \text{width}, \text{skew})$

T为时间序列，width为三角波宽度，skew为斜度，范围0-1，默认为0，即对称三角波。

注意三角波以T=0对称，若要完整显示，时间序列要从负到正（如：T=-30:1:30）。

若要移动三角波，对T进行加减（如T+50表示向右移50单位）

Impulse():单位脉冲响应

$Y = \text{Impulse}(\text{sys}, t)$

Sys为系统函数，t为时间序列（Ts:dt:Te），y为单位阶跃响应，t省略则自动生成时间间隔

Step():单位阶跃响应

$Y = \text{step}(\text{sys}, t)$

Sys为系统函数，t为时间序列（Ts:dt:Te），y为单位阶跃响应，t省略则自动生成时间间隔

$[y, t] = \text{step}(\text{sys})$

返回单位阶跃响应y，时间间隔t

Filter():一维滤波

$y = \text{filter}(b, a, x)$

a,b为滤波器的传递函数的分子分母，x为向量或矩阵，y为输出，大小与x相同。

也可以用来求系统响应，比如系统差分方程：，则b和a为：，x即为x(n)，输出y为系统响应

Impz():求离散LTI系统的单位冲激响应

$h = \text{impz}(b, a, k)$

b为x的向量，a为y的向量，k为输出序列的取值范围。h为单位脉冲响应。

tf(): 系统函数

$\text{sys} = \text{tf}(\text{Numerator}, \text{Denominator})$

创建系统函数，numerator为分子向量，denominator为分母向量

后面可加参数，如：'Variable','p'，意思是把p作为变量

Freqz():离散系统频率响应

$[h, w] = \text{freqz}(b, a, n)$

B为分子，a为分母，返回n个频率点的频率响应，放在h向量中，n个频率放在向量w中，这n个频率均匀设置在（0，pi）上，缺省时默认512

Freqs():连续系统频率响应

$h = \text{freqs}(b, a, w)$

B, a为分子分母，w为角频率向量，返回频率响应h

Ellipord():椭圆滤波器参数

`[n,Wp] = ellipord(Wp,Ws,Rp,Rs,'s')`

Rp、Rs分别为通带最大波纹和阻带最小衰减；wp、ws分别为为通带边界频率和阻带边界频率，单位为rad/s。

's'表示模拟滤波器，缺省时该函数适用于数字滤波器，此时wp及ws均为0~1之间的数值，即频率与抽样频率的比值。返回最小阶数n和截止频率Wp。

编写N点循环卷积程序：

```
X_K1=fft(xn1,N);
```

```
X_K2=fft(xn2,N);
```

```
X_K3=X_K1.*X_K2;
```

```
x_n3=ifft(X_K3,N);
```

因fft可以自动补0，故可以利用其将两序列变成长度为N的DFT序列，再对应相乘，相当于循环卷积的效果。此程序可用来比较线性卷积（conv函数）和循环卷积的区别。

其他笔记：

如果希望直接调用**FFT**子程序计算**IFFT**，则可用下面的方法：

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn}$$
$$x^*(n) = \frac{1}{N} \sum_{k=0}^{N-1} X^*(k) W_N^{kn}$$

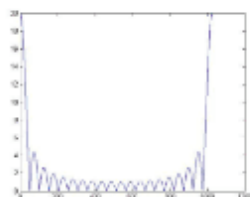
对上式两边同时取共轭，得：

$$x(n) = \frac{1}{N} \left[\sum_{k=0}^{N-1} X^*(k) W_N^{kn} \right]^* = \frac{1}{N} \{ DFT [X^*(k)] \}^*$$

https://blog.csdn.net/weixin_42121843

结果如下图

从513点到1024点是负频率部分吗？连续信号的矩形窗函数的傅立叶变换后半部分可不是这样的啊，频谱是一直衰减的。这是怎么回事呢？



我来答

分享

举报

浏览 4765 次

1个回答

#活动# 【专属】芝麻团队六一狂欢，答题赢万元好礼！



ohyeee

推荐于2016-12-01

对，后一半应当平移到Y轴左边，原因是这样的，本来我们要算的是从 $-\pi$ 到 π 区间，但利用周期性，为了方便计算FFT算的是0到 2π 区间， π 到 2π 区间(图中的513点到1024点)减 2π 对应的就是 $-\pi$ 到0区间

https://blog.csdn.net/weixin_42121843

2.1系统响应及系统稳定性

一、实验目的和步骤

目的：

- (1) 掌握 求系统响应的方法。
- (2) 掌握时域离散系统的时域特性。
- (3) 分析、观察及检验系统的稳定性。

步骤：

(1) 编制程序，包括产生输入信号、单位脉冲响应序列的子程序，用 filter 函数或 conv 函数求解系统输出响应的主程序。程序中要有绘制信号波形的功能。..

(2) 给定一个低通滤波器的差分方程为..

$$y(n) = 0.05x(n) + 0.05x(n-1) + 0.9y(n-1) ..$$

输入信号 $x_1(n) = R_8(n)$..

$$x_2(n) = u(n) ..$$

a) 分别求出系统对 $x_1(n) = R_8(n)$ 和 $x_2(n) = u(n)$ 的响应序列，并画出其波形。..

b) 求出系统的单位冲响应，画出其波形。..

(3) 给定系统的单位脉冲响应为..

$$h_1(n) = R_{10}(n) ..$$

$$h_2(n) = \delta(n) + 2.5\delta(n-1) + 2.5\delta(n-2) + \delta(n-3) ..$$

用线性卷积法分别求系统 $h_1(n)$ 和 $h_2(n)$ 对 $x_1(n) = R_8(n)$ 的输出响应，并画出波形。..

(4) 给定一谐振器的差分方程为..

$$y(n) = 1.8237y(n-1) - 0.9801y(n-2) + b_0x(n) - b_0x(n-2) ..$$

令 $b_0 = 1/100.49$ ，谐振器的谐振频率为 0.4rad 。..

a) 用实验方法检查系统是否稳定。输入信号为 $u(n)$ 时，画出系统输出波形。..

b) 给定输入信号为..

$$x(n) = \sin(0.014n) + \sin(0.4n) ..$$

求出系统的输出响应，并画出其波形。



https://blog.csdn.net/weixin_42121843

I.

二、代码

```
close all
clear all
clc

x1_n=[ones(1,8),zeros(1,90)];
x2_n=ones(1,100);
t=0:1:99;
h_n=impz([0.05 0.05],[1 -0.9],t);%系统单位脉冲响应
y1_n=conv(x1_n,h_n);
y2_n=conv(x2_n,h_n);
figure
subplot(311);stem(h_n);title('系统单位脉冲响应');
subplot(312);stem(y1_n(1:100));title('R8(n)系统响应');
subplot(313);stem(y2_n(1:100));title('u(n)系统响应');

h1_n=[ones(1,10)];
h2_n=[1 2.5 2.5 1];
y3_n=conv(x1_n,h1_n);
y4_n=conv(x1_n,h2_n);
figure
```

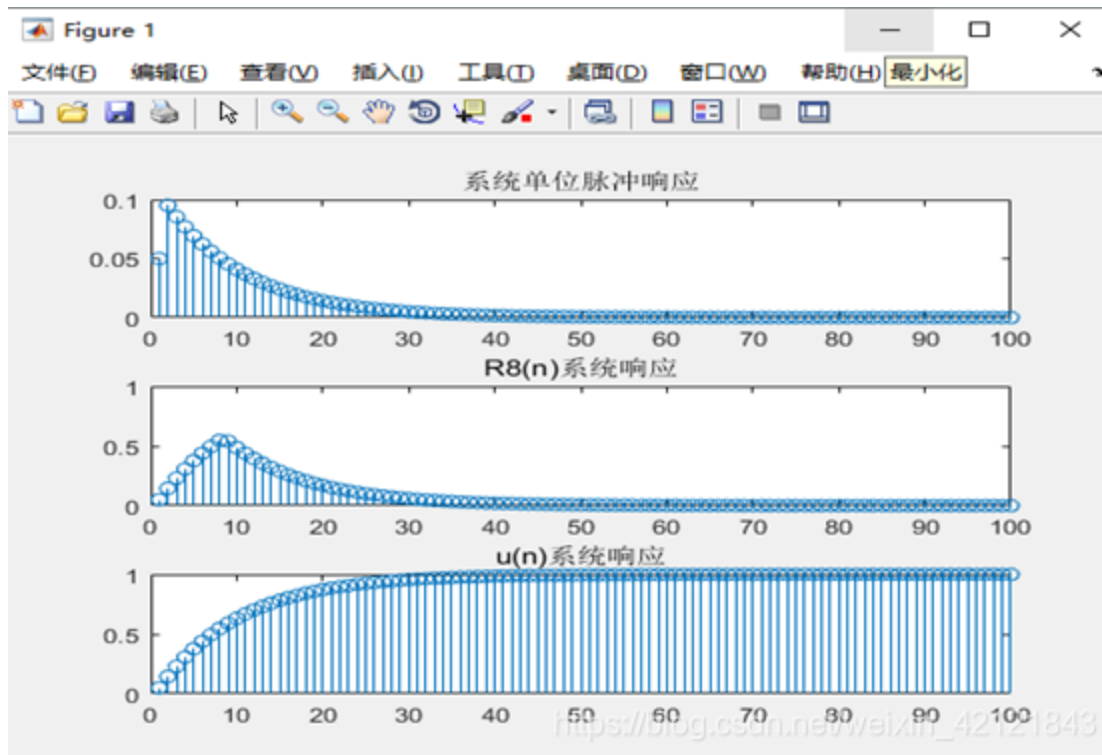
```

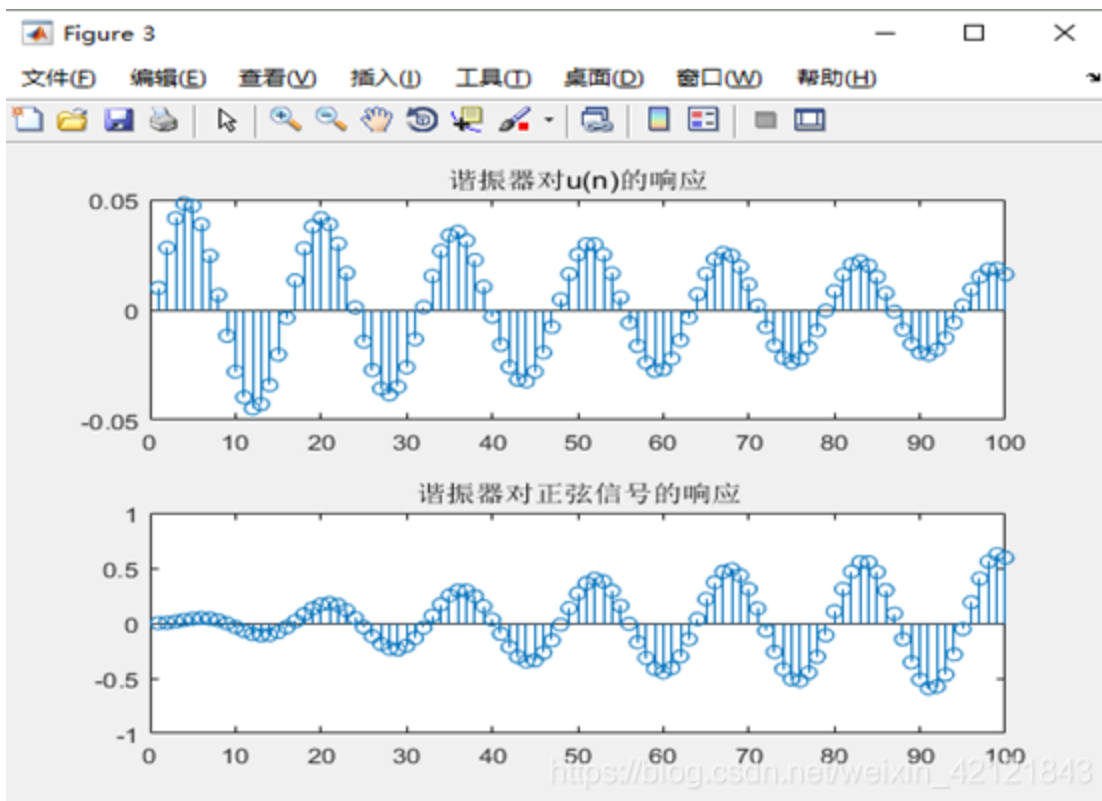
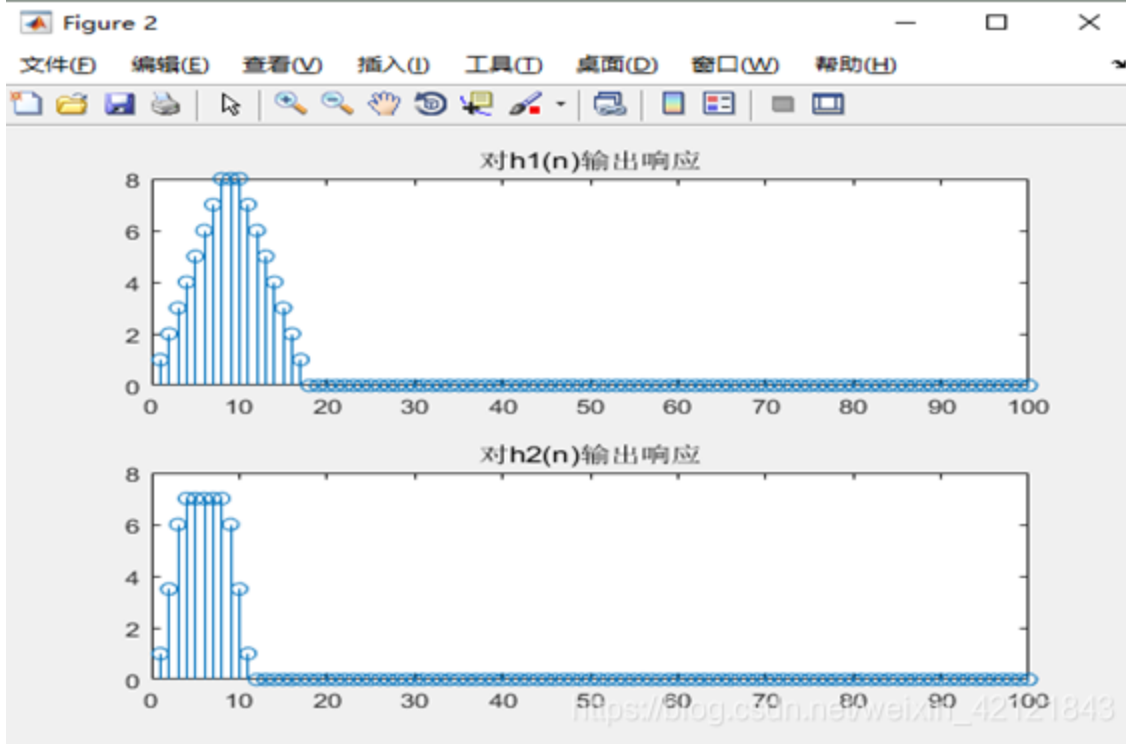
subplot(211);stem(y3_n(1:100));title('对h1(n)输出响应');
subplot(212);stem(y4_n(1:100));title('对h2(n)输出响应');

xsin=sin(0.014*t)+sin(0.4*t); %产生正弦信号
A=[1,-1.8237,0.9801];B=[1/100.49,0,-1/100.49]; %系统差分方程系数向量B和A
y5_n=filter(B,A,x2_n); %谐振器对u(n)的响应
y6_n=filter(B,A,xsin); %谐振器对x(n)的响应
figure
subplot(2,1,1);stem(y5_n);title('谐振器对u(n)的响应');
subplot(2,1,2);stem(y6_n);title('谐振器对正弦信号的响应');

```

三、运行结果





2.2 时域采样与频域采样

一、实验目的和步骤

目的：

验证时域采样和频域采样定理

步骤：

(1) 时域采样理论的验证.

给定模拟信号, $x_a(t) = Ae^{-\alpha t} \sin(\Omega_0 t)u(t)$

式中 $A=444.128$, $\alpha=50\sqrt{2} \pi$, $\Omega_0=50\sqrt{2} \pi \text{ rad/s}$, 它的幅频特性曲线如图 10.2.1.

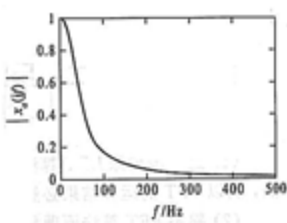


图 10.2.1 $x_a(t)$ 的幅频特性曲线.

现用 DFT(FFT)求该模拟信号的幅频特性, 以验证时域采样理论.

按照 $x_a(t)$ 的幅频特性曲线, 选取三种采样频率, 即 $F_s=1\text{kHz}$, 300Hz , 200Hz . 观

测时间选 $T_p = 50\text{ms}$.

为使用 DFT, 首先用下面公式产生时域离散信号, 对三种采样频率, 采样序列按顺序用 $x_1(n)$, $x_2(n)$, $x_3(n)$ 表示.

$$x(n) = x_a(nT) = Ae^{-\alpha nT} \sin(\Omega_0 nT)u(nT)$$

因为采样频率不同, 得到的 $x_1(n)$, $x_2(n)$, $x_3(n)$ 的长度不同, 长度(点数)

用公式 $N = T_p \times F_s$ 计算. 选 FFT 的变换点数为 $M=64$, 序列长度不够 64 的尾部加零.

$$X(k) = \text{FFT}[x(n)], \quad k=0, 1, 2, 3, \dots, M-1.$$

$$\text{式中 } k \text{ 代表的频率为 } \omega_k = \frac{2\pi}{M} k.$$

要求: 编写实验程序, 计算 $x_1(n)$, $x_2(n)$ 和 $x_3(n)$ 的幅频特性, 并绘图显示, 观察分析频谱混叠失真.

(2) 频域采样理论的验证.

给定信号如下:

$$x(n) = \begin{cases} n+1 & 0 \leq n \leq 13 \\ 27-n & 14 \leq n \leq 26 \\ 0 & \text{其它} \end{cases}$$

编写程序分别对频谱函数 $X(e^{j\omega}) = \text{FT}[x(n)]$ 在区间 $[0, 2\pi]$ 上等间隔采样 32

和 16 点, 得到 $X_{32}(k)$ 和 $X_{16}(k)$:

$$X_{32}(k) = X(e^{j\omega}) \Big|_{\omega = \frac{2\pi}{32}k}, \quad k = 0, 1, 2, \dots, 31.$$

$$X_{16}(k) = X(e^{j\omega}) \Big|_{\omega = \frac{2\pi}{16}k}, \quad k = 0, 1, 2, \dots, 15.$$

再分别对 $X_{32}(k)$ 和 $X_{16}(k)$ 进行 32 点和 16 点 IFFT, 得到 $x_{32}(n)$ 和 $x_{16}(n)$:

$$x_{32}(n) = \text{IFFT}[X_{32}(k)]_{32}, \quad n = 0, 1, 2, \dots, 31.$$

$$x_{16}(n) = \text{IFFT}[X_{16}(k)]_{16}, \quad n = 0, 1, 2, \dots, 15.$$

分别画出 $X(e^{j\omega})$, $X_{32}(k)$ 和 $X_{16}(k)$ 的幅度谱, 并绘图显示 $x(n)$, $x_{32}(n)$ 和 $x_{16}(n)$ 的波形,

进行对比和分析, 验证总结频域采样理论.

提示: 频域采样用以下方法容易实现.

① 直接调用 MATLAB 函数 `fft` 计算 $X_{32}(k) = \text{FFT}[x(n)]_{32}$ 就得到 $X(e^{j\omega})$ 在 $[0, 2\pi]$ 的 32 点频率域采样.

② 抽取 $X_{32}(k)$ 的偶数点即可得到 $X(e^{j\omega})$ 在 $[0, 2\pi]$ 的 16 点频率域采样 $X_{16}(k)$, 即

$$X_{16}(k) = X_{32}(2k), \quad k = 0, 1, 2, \dots, 15.$$

二、代码

```
close all
clear all
clc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5时域采样定理验证
Tp=64/1000;%观察时间
Fs1=1000;%采样频率，分别为1000,300,200Hz
Fs2=300;
Fs3=200;
T1=1/Fs1;%周期
T2=1/Fs2;
T3=1/Fs3;
M=64;
A=444.128;
a=pi*50*2^0.5;
w=pi*50*2^0.5;
n1=0:Tp*Fs1;
n2=0:Tp*Fs2;
n3=0:Tp*Fs3;
xa_t1=A*exp(-a*n1*T1).*sin(w*n1*T1);
xa_t2=A*exp(-a*n2*T2).*sin(w*n2*T2);
xa_t3=A*exp(-a*n3*T3).*sin(w*n3*T3);
Xa_k1=fft(xa_t1,M);
Xa_k2=fft(xa_t2,M);
Xa_k3=fft(xa_t3,M);
figure
subplot(321);stem(xa_t1);title('采样频率为1000Hz的时域离散信号');
subplot(322);stem(abs(Xa_k1));title('64点DFT');
subplot(323);stem(xa_t2);title('采样频率为300Hz的时域离散信号');
subplot(324);stem(abs(Xa_k2));title('64点DFT');
subplot(325);stem(xa_t3);title('采样频率为200Hz的时域离散信号');
subplot(326);stem(abs(Xa_k3));title('64点DFT');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%频域采样定理的验证

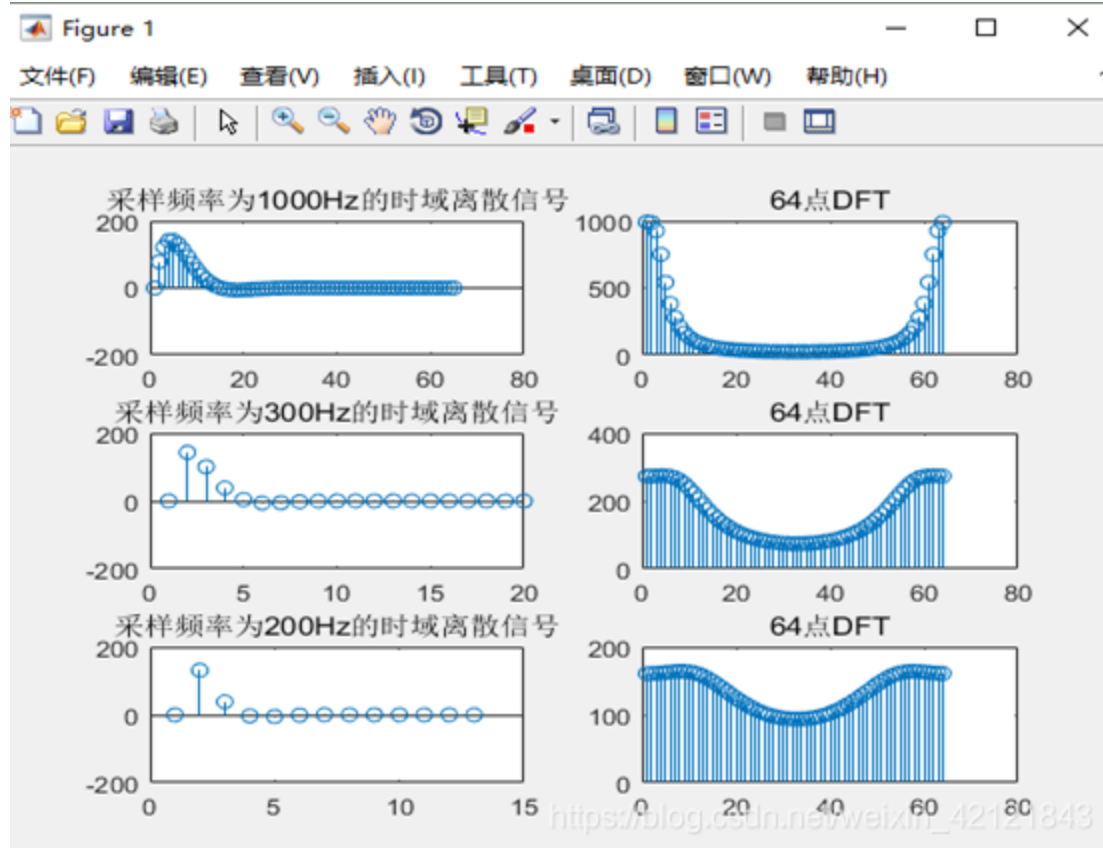
for i=0:13
    x_n(i+1)=i+1;
end
for i=14:26
    x_n(i)=27-i;
end
X_k32=fft(x_n,32);
X_k16=X_k32(1:2:32);
x_n32=ifft(X_k32,32);
x_n16=ifft(X_k16,16);
```

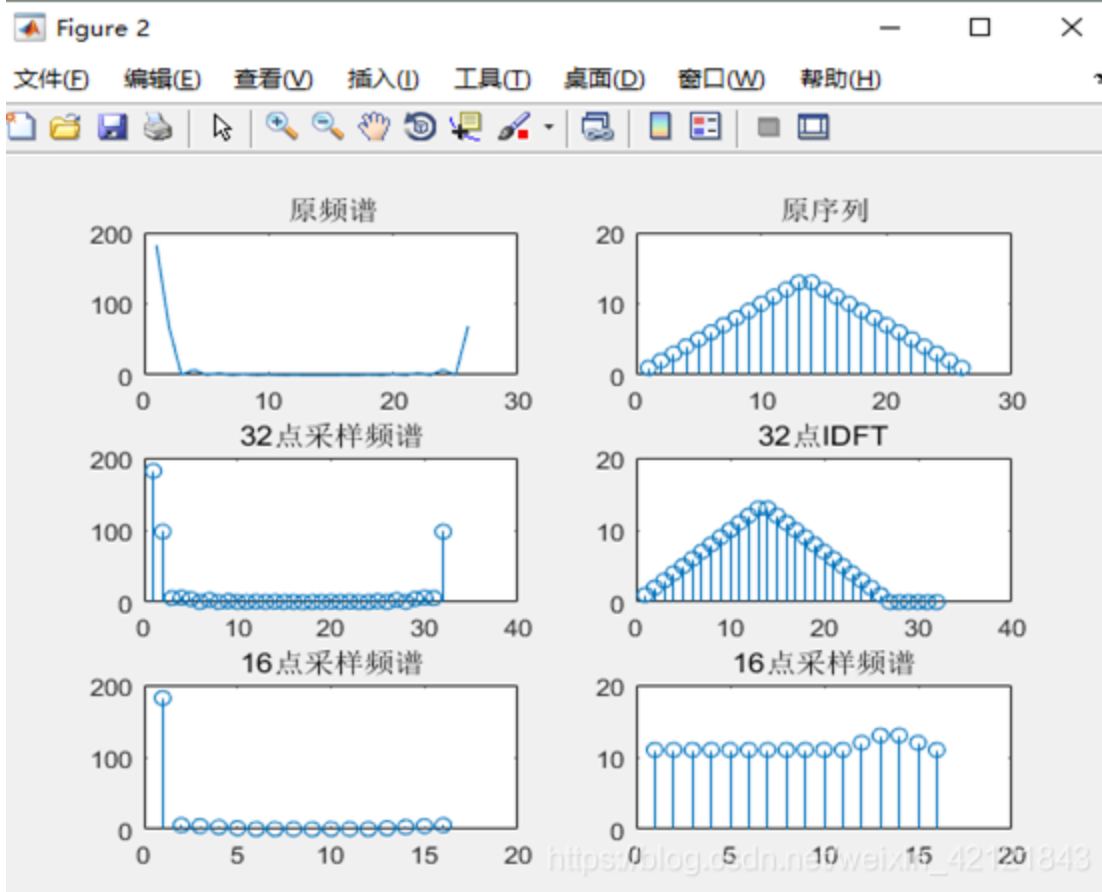
```

figure
subplot(321);plot(abs(fft(x_n)));title('原频谱');
subplot(322);stem(x_n);title('原序列');
subplot(323);stem(abs(X_k32));title('32点采样频谱');
subplot(324);stem(x_n32);title('32点IDFT');
subplot(325);stem(abs(X_k16));title('16点采样频谱');
subplot(326);stem(x_n16);title('32点采样频谱');

```

三、运行结果





2.3 用FFT对信号作频谱分析

一、实验目的和步骤

步骤：

(1) 对以下序列进行谱分析。..

$$x_1(n) = R_4(n)$$

$$x_2(n) = \begin{cases} n+1, & 0 \leq n \leq 3 \\ 8-n, & 4 \leq n \leq 7 \\ 0, & \text{其它}n \end{cases}$$

$$x_3(n) = \begin{cases} 4-n, & 0 \leq n \leq 3 \\ n-3, & 4 \leq n \leq 7 \\ 0, & \text{其它}n \end{cases}$$

选择 FFT 的变换区间 N 为 8 和 16 两种情况进行频谱分析。分别打印其幅频特性曲线。并进行对比、分析和讨论。..

(2) 对以下周期序列进行谱分析。..

$$x_4(n) = \cos \frac{\pi}{4} n..$$

$$x_5(n) = \cos(\pi n/4) + \cos(\pi n/8)..$$

选择 FFT 的变换区间 N 为 8 和 16 两种情况分别对以上序列进行频谱分析。分别打印其幅频特性曲线。并进行对比、分析和讨论。..

(3) 对模拟周期信号进行谱分析。..

$$x_6(t) = \cos 8\pi t + \cos 16\pi t + \cos 20\pi t..$$

选择 采样频率 $F_s = 64Hz$ ，变换区间 $N=16,32,64$ 三种情况进行谱分析。分别打印其幅频特性，并进行分析和讨论。..
https://blog.csdn.net/weixin_42121843

二、代码

```
close all
clear all
clc

x1_n=[ones(1,4)];
for i=1:4
    x2_n(i)=i;
end
for i=5:8
    x2_n(i)=9-i;
end

for i=1:4
    x3_n(i)=5-i;
end
for i=5:8
    x3_n(i)=i-2;
end

X1_k8=fft(x1_n,8);
X1_k16=fft(x1_n,16);
X2_k8=fft(x2_n,8);
X2_k16=fft(x2_n,16);
X3_k8=fft(x3_n,8);
```

```
X3_k16=fft(x3_n,16);
```

```
figure
```

```
subplot(321);stem(abs(X1_k8));title('x1(n)的8点DFT');  
subplot(322);stem(abs(X1_k16));title('x1(n)的16点DFT');  
subplot(323);stem(abs(X2_k8));title('x2(n)的8点DFT');  
subplot(324);stem(abs(X2_k16));title('x2(n)的16点DFT');  
subplot(325);stem(abs(X3_k8));title('x3(n)的8点DFT');  
subplot(326);stem(abs(X3_k16));title('x3(n)的16点DFT');
```

```
n=1:32;
```

```
x4_n=cos(pi/4*n);
```

```
x5_n=cos(pi*n/4)+cos(pi*n/8);
```

```
X4_k8=fft(x4_n,8);
```

```
X4_k16=fft(x4_n,16);
```

```
X5_k8=fft(x5_n,8);
```

```
X5_k16=fft(x5_n,16);
```

```
figure
```

```
subplot(221);stem(abs(X4_k8));title('x4(n)的8点DFT');  
subplot(222);stem(abs(X4_k16));title('x4(n)的16点DFT');  
subplot(223);stem(abs(X5_k8));title('x5(n)的8点DFT');  
subplot(224);stem(abs(X5_k16));title('x5(n)的16点DFT');
```

```
N1=16;
```

```
N2=32;
```

```
N3=64;
```

```
n1=1:1:N1;
```

```
n2=1:1:N2;
```

```
n3=1:1:N3;
```

```
Fs=64;
```

```
t1=n1./Fs;
```

```
t2=n2./Fs;
```

```
t3=n3./Fs;
```

```
x8_t1=cos(8*pi*t1)+cos(16*pi*t1)+cos(20*pi*t1);
```

```
X8_k1=fft(x8_t1,N1);
```

```
x8_t2=cos(8*pi*t2)+cos(16*pi*t2)+cos(20*pi*t2);
```

```
X8_k2=fft(x8_t2,N2);
```

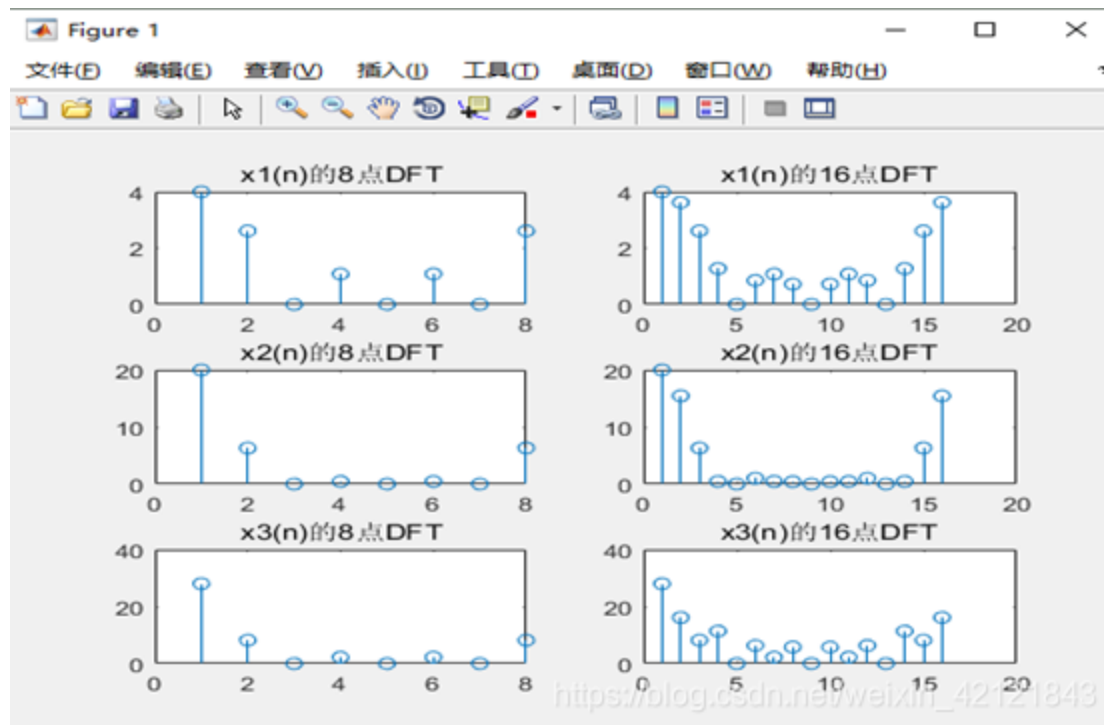
```
x8_t3=cos(8*pi*t3)+cos(16*pi*t3)+cos(20*pi*t3);
```

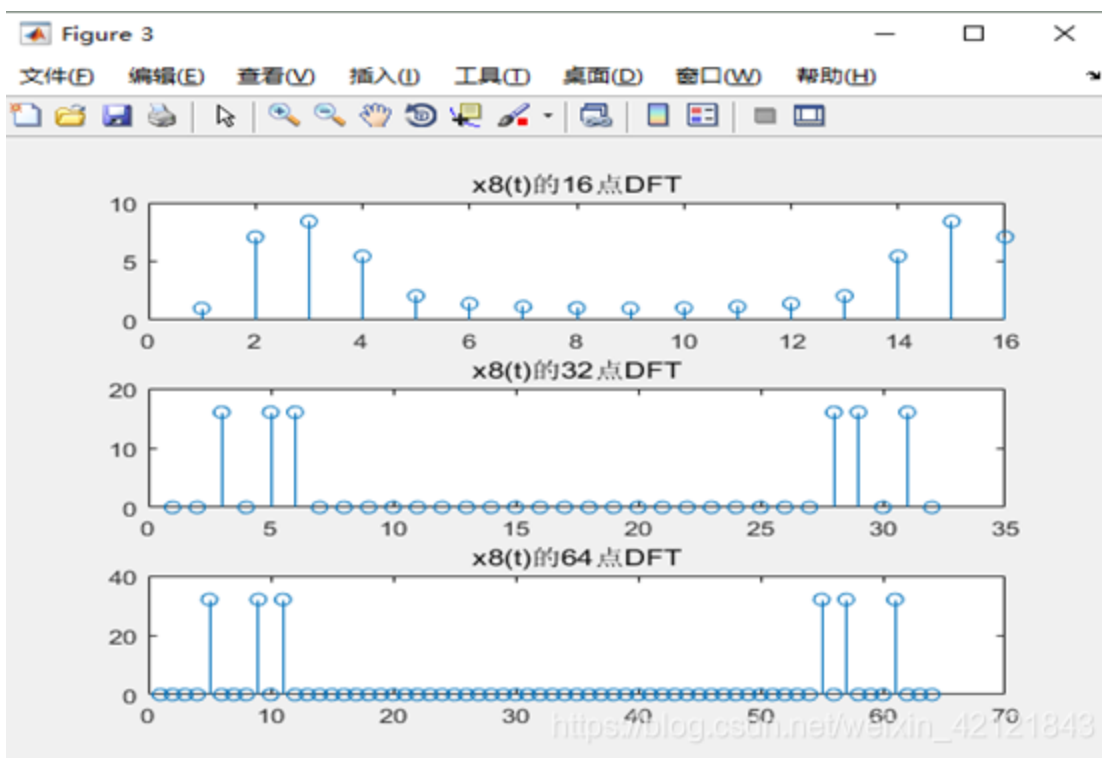
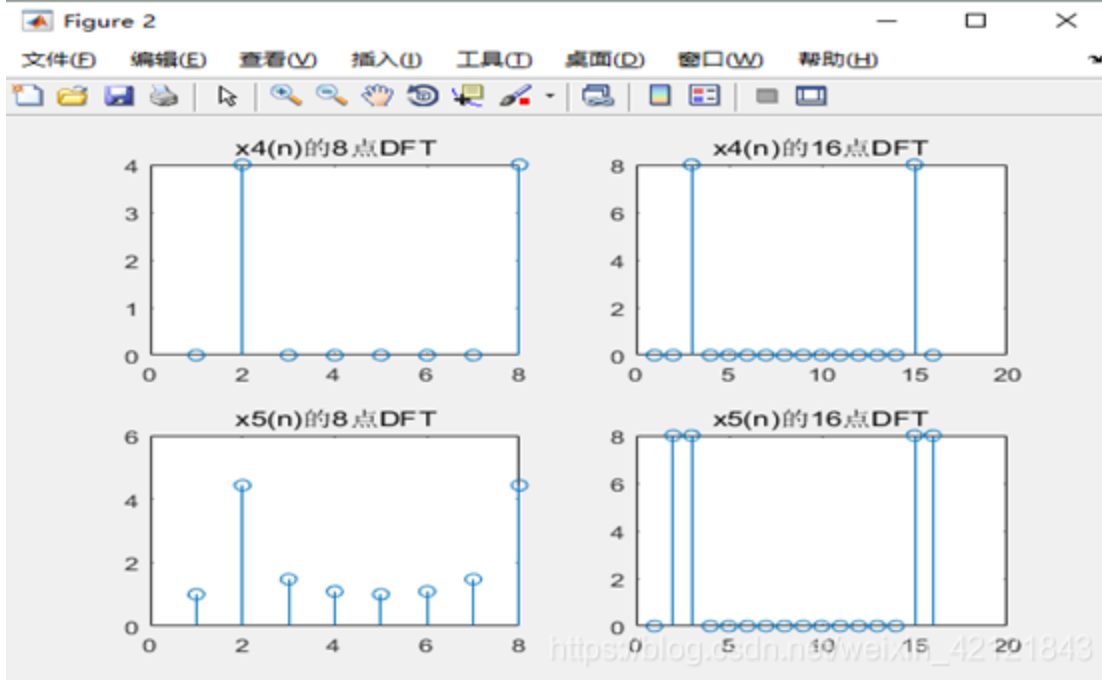
```
X8_k3=fft(x8_t3,N3);
```

```
figure
```

```
subplot(311);stem(abs(X8_k1));title('x8(t)的16点DFT');  
subplot(312);stem(abs(X8_k2));title('x8(t)的32点DFT');  
subplot(313);stem(abs(X8_k3));title('x8(t)的64点DFT');
```

三、运行结果





2.4 IIR数字滤波器设计

一、实验目的和步骤

目的：

- (1) 熟悉用双线性变换法设计IIR数字滤波器的原理与方法；
- (2) 学会调用MATLAB信号处理工具箱中滤波器设计函数（或滤波器设计分析工具fdatool）设计各种IIR数字滤波器，学会根据滤波需求确定滤波器指标参数。
- (3) 掌握IIR数字滤波器的MATLAB实现方法。
- (3) 通过观察滤波器输入输出信号的时域波形及其频谱，建立数字滤波的概念。

步骤:

(1) 调用信号产生函数 `mstg` 产生由三路抑制载波调幅信号相加构成的复合信号 `st`，该函数还会自动绘图显示 `st` 的时域波形和幅频特性曲线，如图 10.4.1 所示。由图可见，三路信号时域混叠无法在时域分离，但频域是分离的，所以可以通过滤波的方法在频域分离，这就是本实验的目的。

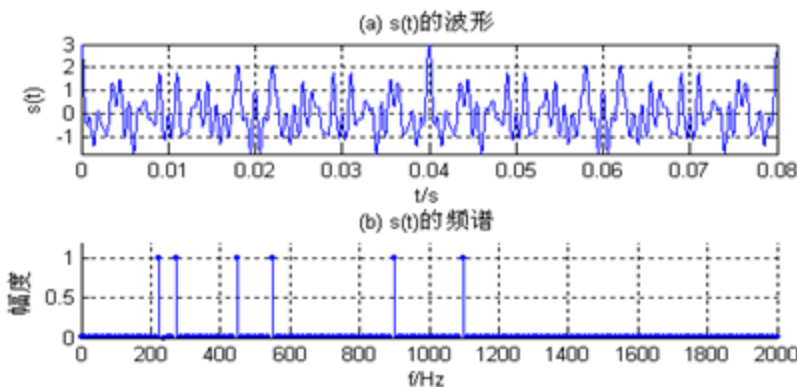


图 10.4.1 三路调幅信号 `st` 的时域波形和幅频特性曲线

(2) 要求将 `st` 中三路调幅信号分离，通过观察 `st` 的幅频特性曲线，分别确定可以分离 `st` 中三路抑制载波单频调幅信号的三个滤波器（低通滤波器、带通滤波器、高通滤波器）的通带截止频率和阻带截止频率。要求滤波器的通带最大衰减为 0.1dB，阻带最小衰减为 60dB。

提示：抑制载波单频调幅信号的数学表示式为：

$$s(t) = \cos(2\pi f_c t) \cos(2\pi f_0 t) = \frac{1}{2} [\cos(2\pi(f_c - f_0)t) + \cos(2\pi(f_c + f_0)t)]$$

其中， $\cos(2\pi f_c t)$ 称为载波， f_c 为载波频率， $\cos(2\pi f_0 t)$ 称为单频调制信号， f_0 为调制正弦信号频率，且满足 $f_c > f_0$ 。由上式可见，所谓抑制载波单频调幅信号，就是 2 个正弦

信号相乘，它有 2 个频率成分：和频 $f_c + f_0$ 和差频 $f_c - f_0$ ，这 2 个频率成分关于载波频率 f_c 对称。所以，1 路抑制载波单频调幅信号的频谱图是关于载波频率 f_c 对称的 2 根谱线，其中没有载频成分，故取名为抑制载波单频调幅信号。容易看出，图 10.4.1 中三路调幅信号的载波频率分别为 250Hz、500Hz、1000Hz。如果调制信号 $m(t)$ 具有带限连续频谱，无直流成分，则 $s(t) = m(t) \cos(2\pi f_c t)$ 就是一般的抑制载波调幅信号。其频谱图是关于载波频率 f_c 对称的 2 个边带（上下边带），在专业通信原理中称为双边带抑制载波（DSB-SC）调幅信号，简称双边带（DSB）信号。如果调制信号 $m(t)$ 有直流成分，则 $s(t) = m(t) \cos(2\pi f_c t)$ 就是一般的双边带调幅信号。其频谱图是关于载波频率 f_c 对称的 2 个边带（上下边带），并包含载频成分。

(3) 编程调用 MATLAB 滤波器设计函数 `ellipord` 和 `ellip` 分别设计这三个椭圆滤波器，并绘图显示其幅频响应特性曲线。

(4) 调用滤波器实现函数 `filter`，用三个滤波器分别对信号产生函数 `mstg` 产生的信号 `st` 进行滤波，分离出 `st` 中的三路不同载波频率的调幅信号 $y_1(n)$ 、 $y_2(n)$ 和 $y_3(n)$ ，并绘图显示 $y_1(n)$ 、 $y_2(n)$ 和 $y_3(n)$ 的时域波形，观察分离效果。

二、代码

```
close all
clear all
clc
```

```
%%%%%%%%%%滤波器设计
```

```
st=mstg; %调用信号产生函数mstg产生由三路抑制载波调幅信号相加构成的复合信号s
```

```

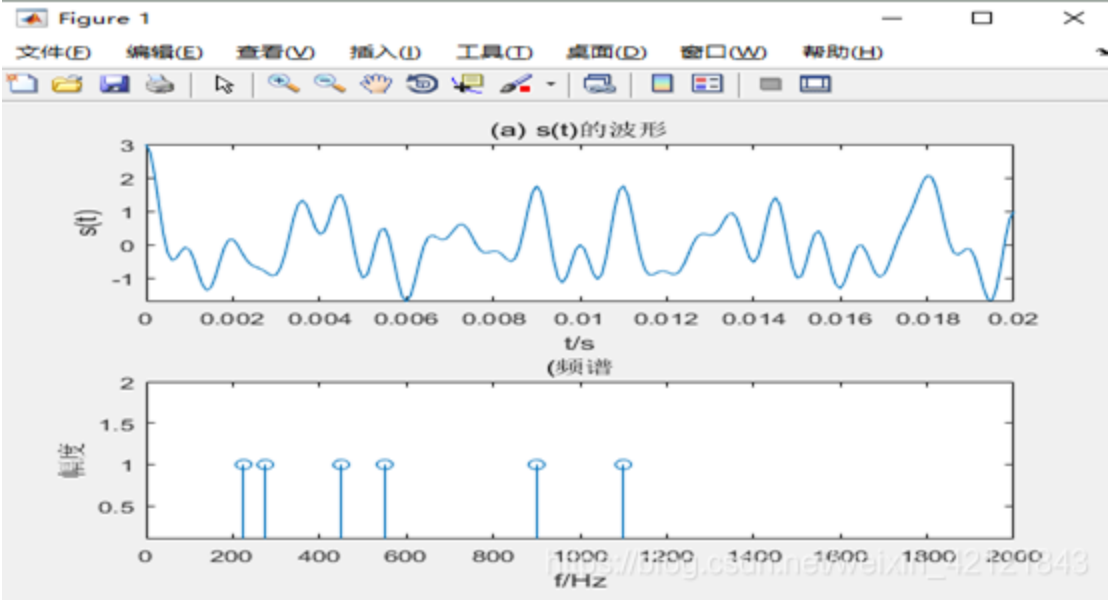
Fs=10000; T=1/Fs; %采样频率
fp=280; fs=450;wp=2*fp/Fs;ws=2*fs/Fs; %低通 (关于pi归一化)
    name='y_1(t)';LHL_filter(wp,ws,name,st,T,'low');
fp_L=440; fp_R=560;fs_L=275; fs_R=900;%带通
    wp=[2*fp_L/Fs,2*fp_R/Fs];ws=[2*fs_L/Fs,2*fs_R/Fs];
    name='y_2(t)'; LHL_filter(wp,ws,name,st,T,'bandpass');
fp=890; fs=600;wp=2*fp/Fs;ws=2*fs/Fs; %高通
    name='y_3(t)';LHL_filter(wp,ws,name,st,T,'high');

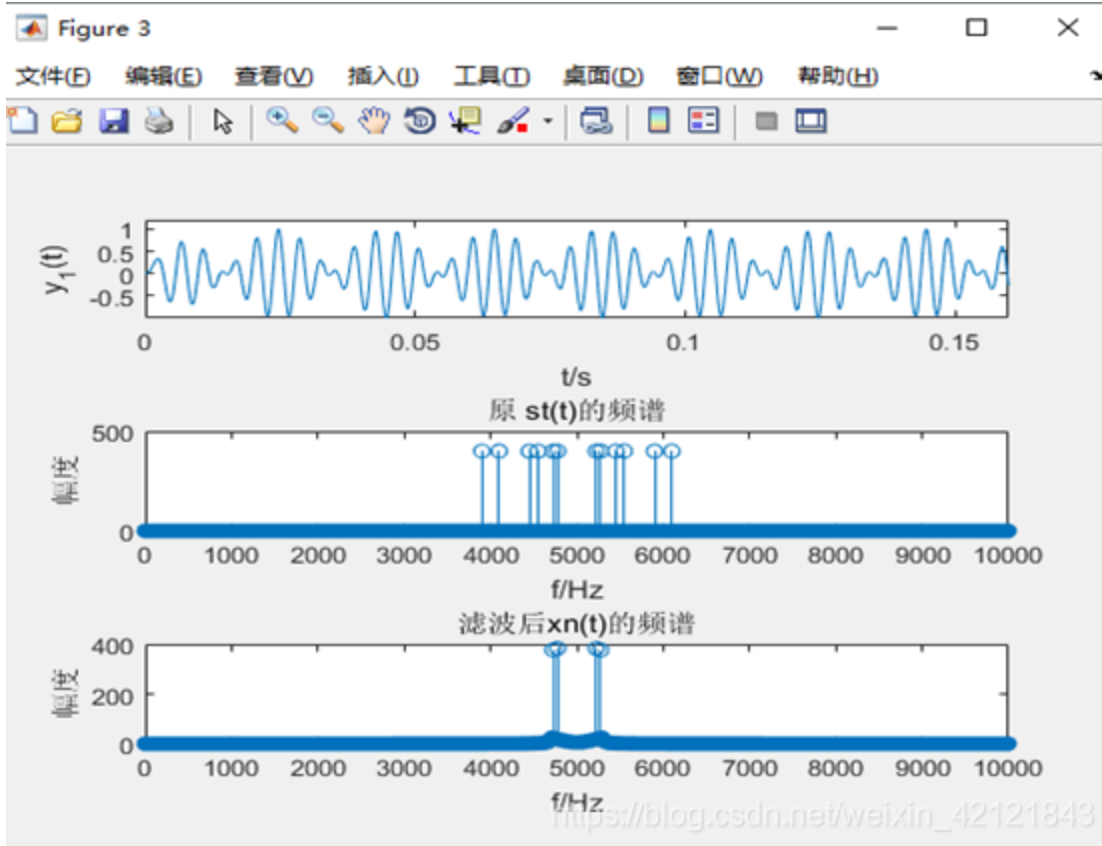
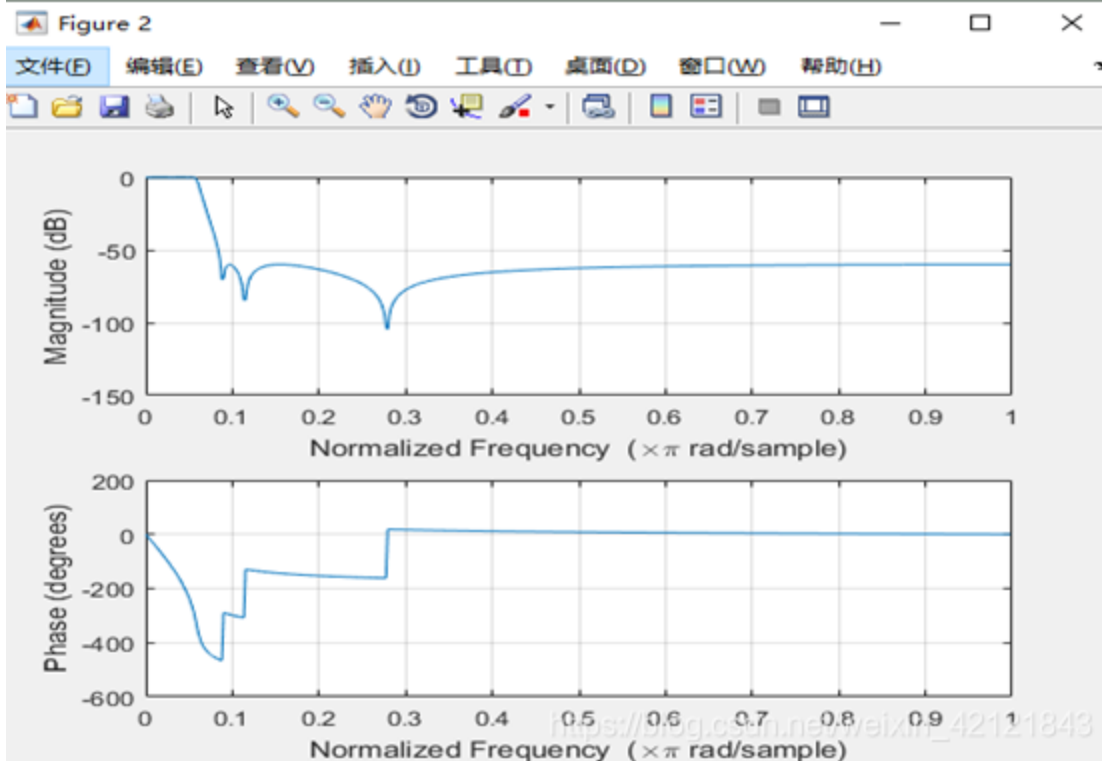
function LHL_filter(wp,ws,name,st,T,flag)
rp=0.1;rs=60; %DF指标 (低通滤波器的通、阻带边界频)
[N,wp]=ellipord(wp,ws,rp,rs); %调用ellipord计算椭圆DF阶数N和通带截止频率wp?
[B,A]=ellip(N,rp,rs,wp,flag); %调用ellip计算椭圆带通DF系统函数系数向量B和A
yt=filter(B,A,st); %滤波器软件实现
    figure; freqz(B,A);%B,A分别为系统函数分子, 分母多项式系数向量?
    figure;subplot(st, yt, T, name);
end
function tplot(st,xn,T,name)
%时域序列连续曲线绘图函数
%?xn:信号数据序列, %?T为采样间隔
N=length(xn);n=0:N-1; t=n*T; Tp=N*T; f=n/Tp;
subplot(311);plot(t,xn);xlabel('t/s');ylabel(name);
axis([0,t(end),min(xn),1.2*max(xn)]);
subplot(312);stem(f,abs(fftshift(fft(st,N))));
title('原 st(t)的频谱');xlabel('f/Hz');ylabel('幅度');
subplot(313);stem(f,abs(fftshift(fft(xn,N))));
title('滤波后xn(t)的频谱');xlabel('f/Hz');ylabel('幅度');
end
function st=mstg
%产生信号序列向量st,并显示st的时域波形和频谱
%st=mstg 返回三路调幅信号相加形成的混合信号,长度N=1600
N=1600;Fs=10000;T=1/Fs;Tp=N*T; % N为信号st的长度。采样频率Fs=10kHz, Tp为采样时间
t=0:T:(N-1)*T;k=0:N-1;f=k/Tp;
fc1=Fs/10;fm1=fc1/10;%第1路调幅信号的载波频率fc1=1000Hz, 调制信号频率fm1=100Hz
fc2=Fs/20;fm2=fc2/10;%第2路调幅信号的载波频率fc2=500Hz, 调制信号频率fm2=50Hz
fc3=Fs/40;fm3=fc3/10;%第3路调幅信号的载波频率fc3=250Hz, 调制信号频率fm3=25Hz
xt1=cos(2*pi*fm1*t).*cos(2*pi*fc1*t); %产生第1路调幅信号
xt2=cos(2*pi*fm2*t).*cos(2*pi*fc2*t); %产生第2路调幅信号
xt3=cos(2*pi*fm3*t).*cos(2*pi*fc3*t); %产生第3路调幅信号
st=xt1+xt2+xt3; %三路调幅信号相加
fxt=fft(st,N); %计算信号st的频谱
subplot(211);plot(t,st);xlabel('t/s');ylabel('s(t)');
axis([0,Tp/8,min(st),max(st)]);title('(a) s(t)的波形');
subplot(212);stem(f,abs(fxt)./max(abs(fxt)));
axis([0,Fs/5,0.1,2]);
title('(b) 频谱');xlabel('f/Hz');ylabel('幅度');

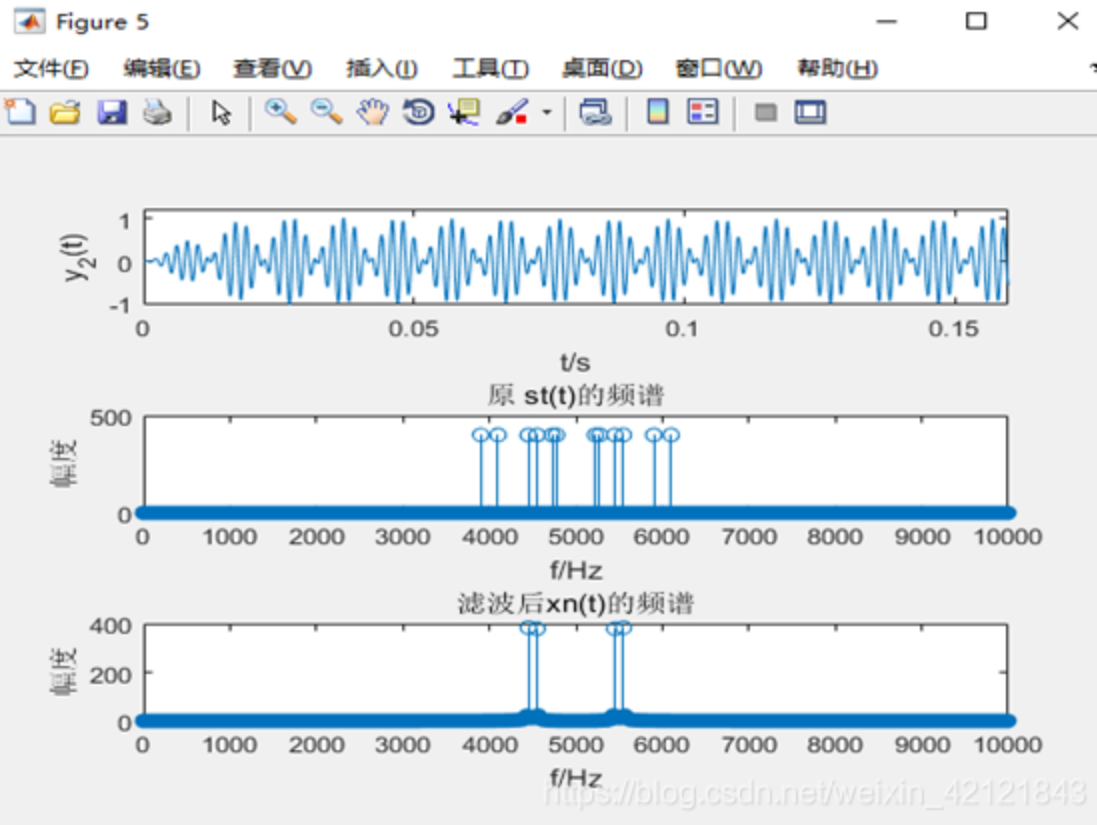
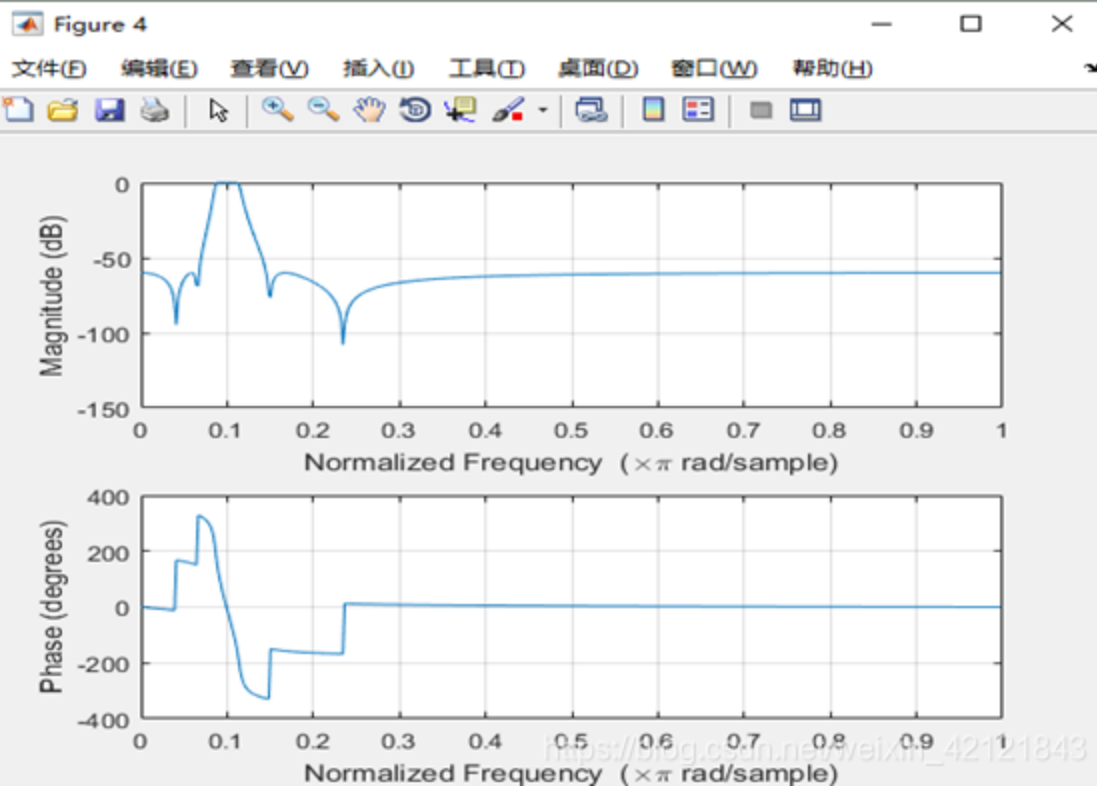
```

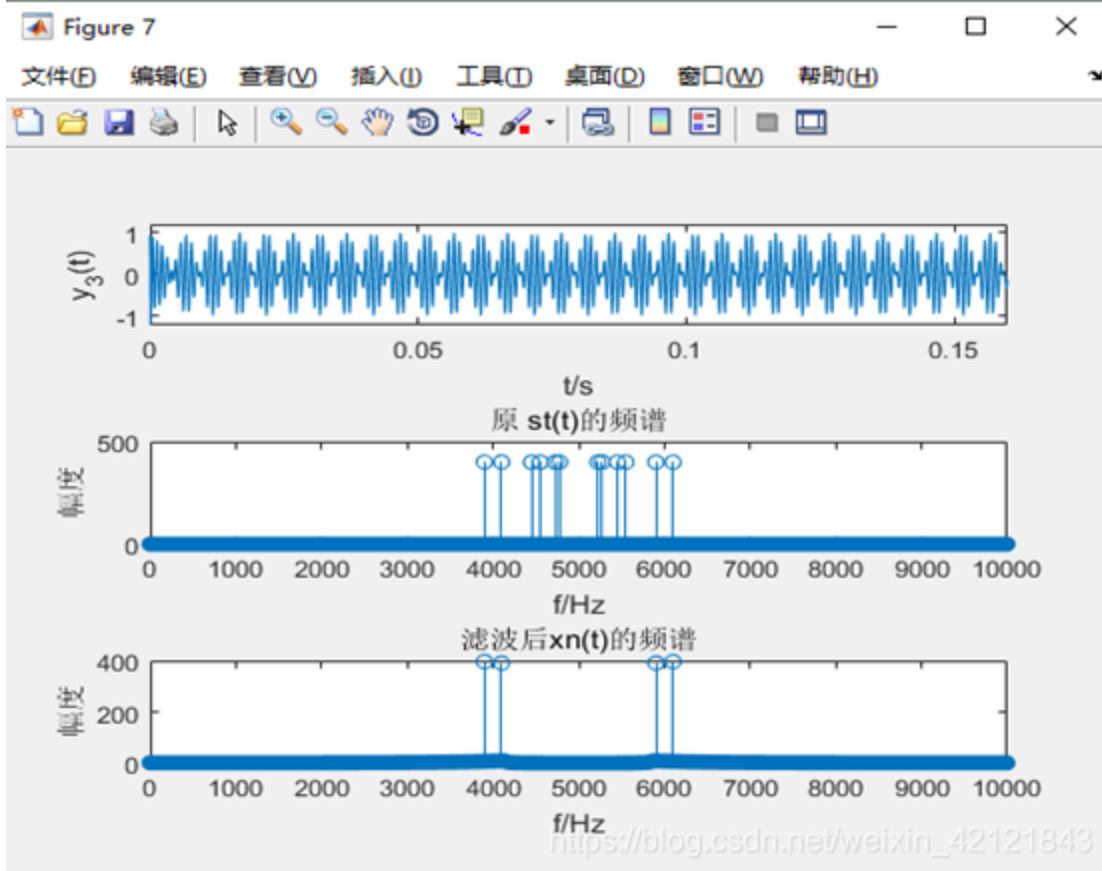
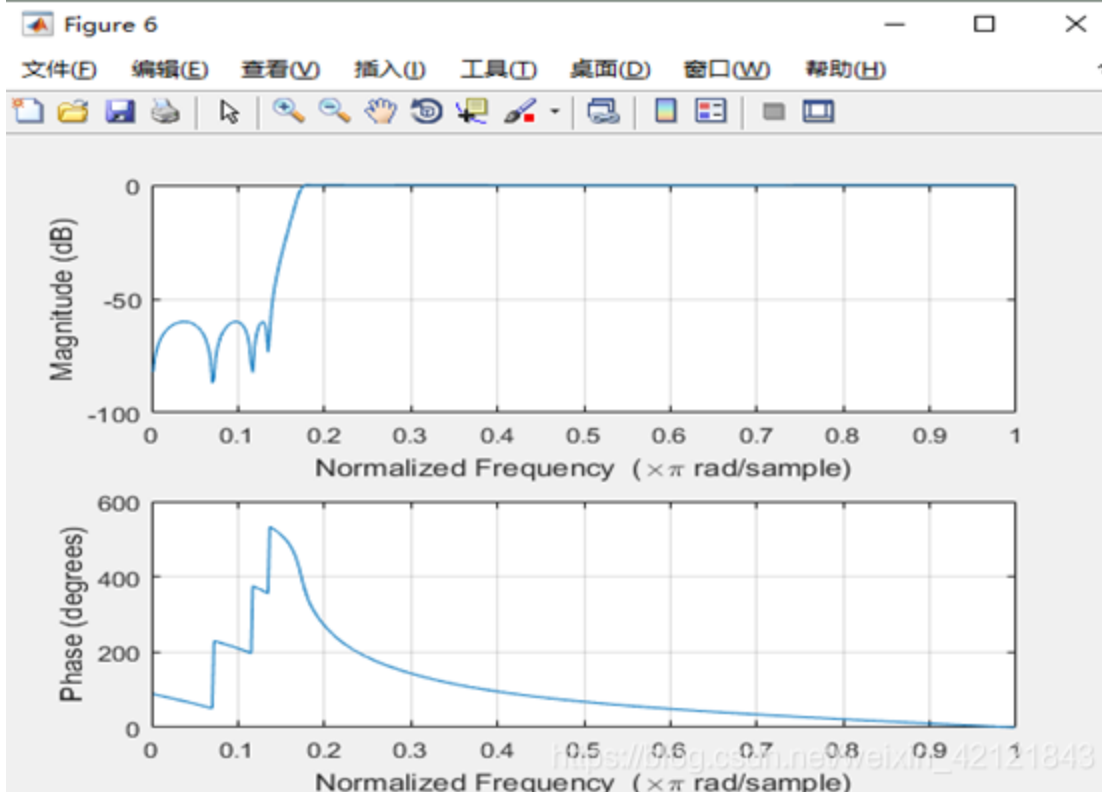
end

三、运行结果









2.5 FIR数字滤波器设计

一、实验目的和步骤

目的：

- (1) 掌握用窗函数法设计FIR数字滤波器的原理和方法。

- (2) 掌握用等波纹最佳逼近法设计FIR数字滤波器的原理和方法。
- (3) 掌握FIR滤波器的快速卷积实现原理。
- (4) 学会调用MATLAB函数设计与实现FIR滤波器。

步骤：

- (1) 认真复习第七章中用窗函数法和等波纹最佳逼近法设计 FIR 数字滤波器的原理；
- (2) 调用信号产生函数 `xtg` 产生具有加性噪声的信号 `xt`，并自动显示 `xt` 及其频谱，如图 10.5.1 所示；

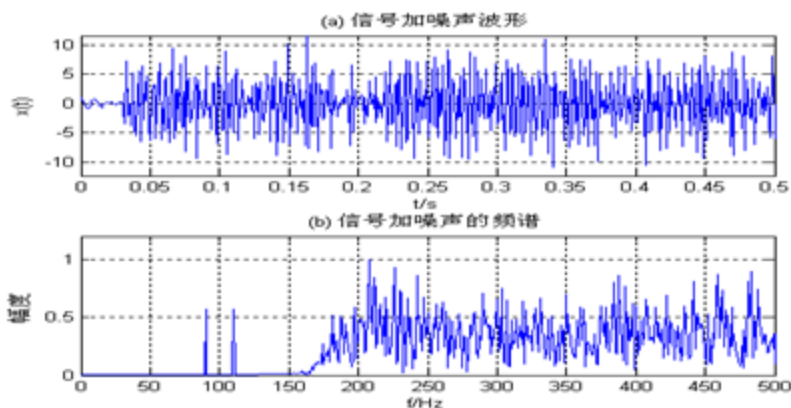


图 10.5.1 具有加性噪声的信号 $x(t)$ 及其频谱如图。

- (3) 请设计低通滤波器，从高频噪声中提取 `xt` 中的单频调幅信号，要求信号幅频失真小于 0.1dB，将噪声频谱衰减 60dB。先观察 `xt` 的频谱，确定滤波器指标参数。

- (4) 根据滤波器指标选择合适的窗函数，计算窗函数的长度 N ，调用 MATLAB 函数 `fir1` 设计一个 FIR 低通滤波器。并编写程序，调用 MATLAB 快速卷积函数 `fftfilt` 实现对 `xt` 的滤波。绘图显示滤波器的频响特性曲线、滤波器输出信号的幅频特性图和时域波形图。

- (4) 重复(3)，滤波器指标不变，但改用等波纹最佳逼近法，调用 MATLAB 函数 `ramazord` 和 `ramaz` 设计 FIR 数字滤波器。并比较两种设计方法设计的滤波器阶数。

提示：①MATLAB 函数 `filter` 和 `fft` 的功能及其调用格式请查阅本书。

②采样频率 $F_s=1000\text{Hz}$ ，采样周期 $T=1/F_s$ 。

③根据图 10.6.1(b)和实验要求，可选择滤波器指标参数：通带截止频率 $f_p=120\text{Hz}$ ，阻带截止频率 $f_s=150\text{Hz}$ ，换算成数字频率，通带截止频率 $\omega_p=2\pi f_p T=0.24\pi$ ，通带最大衰减为 0.1dB ，阻带截止频率 $\omega_s=2\pi f_s T=0.3\pi$ ，阻带最小衰减为 60dB 。

④实验程序框图如图 10.5.2 所示，供读者参考。

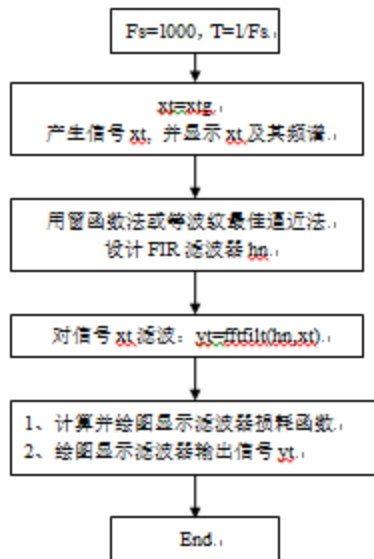


图 10.5.2 实验程序框图

二、代码

```
clc
close all
clear all

xt=xtg;
Fs=1000;T=1/Fs;
fp=120;fs=150;
wp=2*fp*T;ws=2*fs*T;
%%%%%%%%%%%%低通滤波器
rp=0.1;rs=60; %DF指标
[N,wp]=ellipord(wp,ws,rp,rs); %调用ellipord计算椭圆DF阶数N和通带截止频率wp
[B,A]=ellip(N,rp,rs,wp,'low'); %调用ellip计算椭圆带通DF系统函数系数向量B和A
low_filter=freqz(B,A);
yt1=filter(B,A,xt);
figure
subplot(311);plot(abs(low_filter));title('椭圆低通滤波器幅度谱');xlabel('f/Hz');axis([0,600,0,1]);
subplot(312);plot(angle(low_filter));title('椭圆低通滤波器相位谱');xlabel('f/Hz');axis([0,600,-1.2,1.2]);
subplot(313);plot(20*log(abs(low_filter)));title('椭圆低通滤波器损耗函数');xlabel('f/Hz');axis([0,600,-10,0]);
figure
subplot(311);plot(yt1);title('椭圆低通滤波后时域波形');axis([0,600,-1.2,1.2]);
subplot(312);plot(abs(fft(yt1)));title('幅频特性');
```

```
subplot(313);plot(angle(fft(yt1)));title('相频特性');

%%%%%%窗函数法，逼近的函数为理想低通滤波器，使用凯塞窗
Bt=ws-wp;%过渡带宽度
alph=0.112*(rs-8.7);%凯塞窗参数a
M=ceil((rs-8)/2.285/Bt)%凯塞窗阶数
wc=(ws+wp)/2;%已对pi归一化
hn=fir1(M,wc,aiser(M+1,alph));
[HK,w]=freqz(hn,1);
yt2=fftfilt(hn,xt,1000);
figure
subplot(411);plot(hn);title('凯塞窗h(n)波形');xlabel('n');
subplot(412);plot(w/pi,abs(HK));title('幅度谱');xlabel('w/\pi');
subplot(413);plot(w/pi,angle(HK));title('相位谱');xlabel('w/\pi');
subplot(414);plot(w/pi,20*log(abs(HK)));title('损耗函数');xlabel('w/\pi');
```

```
figure
subplot(311);plot(yt2);title('凯塞窗滤波后时域波形');axis([0,600,-1.2,1.2]);
subplot(312);plot(abs(fft(yt2)));title('幅频特性');
subplot(313);plot(angle(fft(yt2)));title('相频特性');
```

```
%%%%%%等波纹法
f=[fp,fs];m=[1,0];
dat1=(10^(rp/20)-1)/(10^(rp/20)+1);dat2=10^(-rs/20);
rip=[dat1,dat2];
[M1,fo,mo,w]=remezord(f,m,rip,Fs);
M1=M1+1
hn1=remez(M1,fo,mo,w);
[HK1,w]=freqz(hn1,1);
yt3=fftfilt(hn1,xt,1000);
figure
subplot(411);plot(hn);title('等波纹法滤波器h(n)波形');xlabel('n');
subplot(412);plot(w/pi,abs(HK1));title('幅度谱');xlabel('w/\pi');
subplot(413);plot(w/pi,angle(HK1));title('相位谱');xlabel('w/\pi');
subplot(414);plot(w/pi,20*log(abs(HK1)));title('损耗函数');xlabel('w/\pi');
```

```
figure
subplot(311);plot(yt3);title('等波纹法滤波后时域波形');axis([0,600,-1.2,1.2]);
subplot(312);plot(abs(fft(yt3)));title('幅频特性');
subplot(313);plot(angle(fft(yt3)));title('相频特性');
```

```
function xt=xtg
N=1000;Fs=1000;T=1/Fs;Tp=N*T;
t=0:T:(N-1)*T;
fc=Fs/10;
f0=fc/10;
```

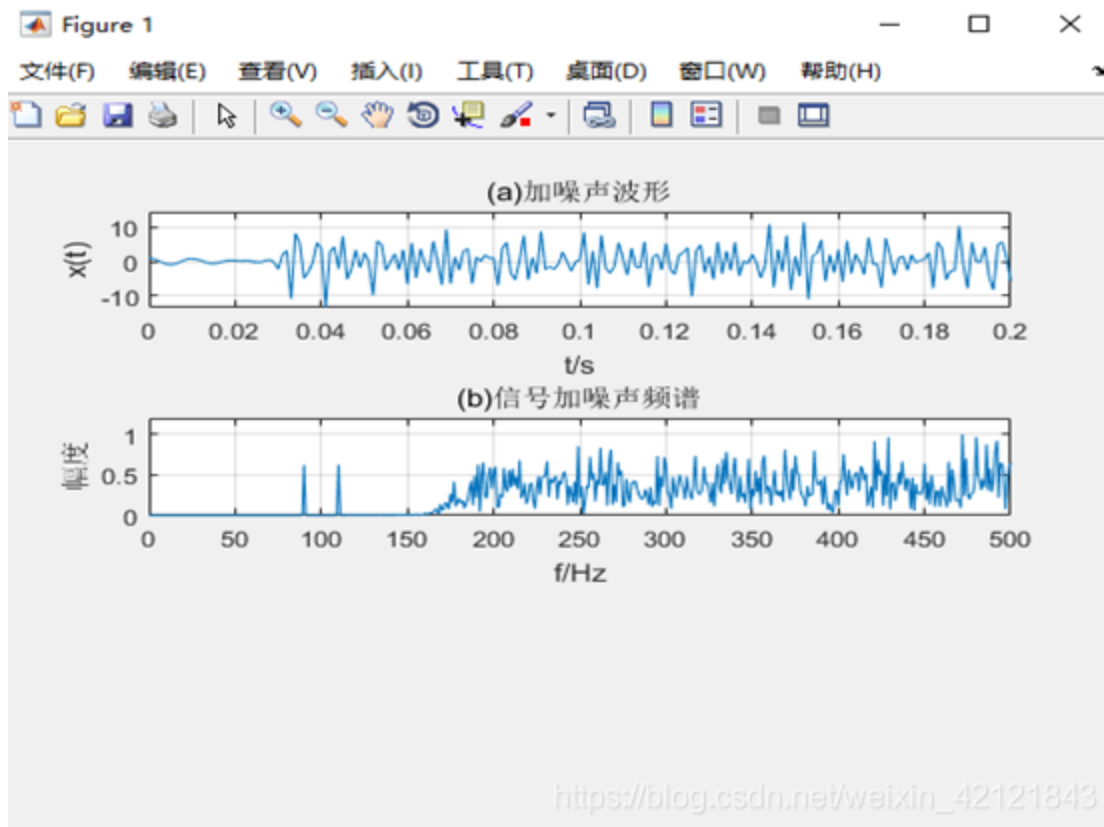
```

mt=cos(2*pi*f0*t);
ct=cos(2*pi*fc*t);
xt=mt.*ct;
nt=2*rand(1,N)-1;
fp=150;fs=200;Rp=0.1;As=70;
fb=[fp,fs];m=[0,1];
dev=[10^(-As/20),(10^(Rp/20)-1)/(10^(Rp/20)+1)];
[n,fo,mo,W]=remezord(fb,m,dev,Fs);
hn=remez(n,fo,mo,W);
yt=filter(hn,1,10*nt);

xt=xt+yt;
fst=fft(xt,N);k=0:N-1;f=k/Tp;
subplot(311);plot(t,xt);grid;xlabel('t/s');ylabel('x(t)');
axis([0,Tp/5,min(xt),max(xt)]);title('(a)加噪声波形');
subplot(312);plot(f,abs(fst)/max(abs(fst)));grid;title('(b)信号加噪声频谱');
axis([0,Fs/2,0,1.2]);xlabel('f/Hz');ylabel('幅度');
end

```

三、运行结果



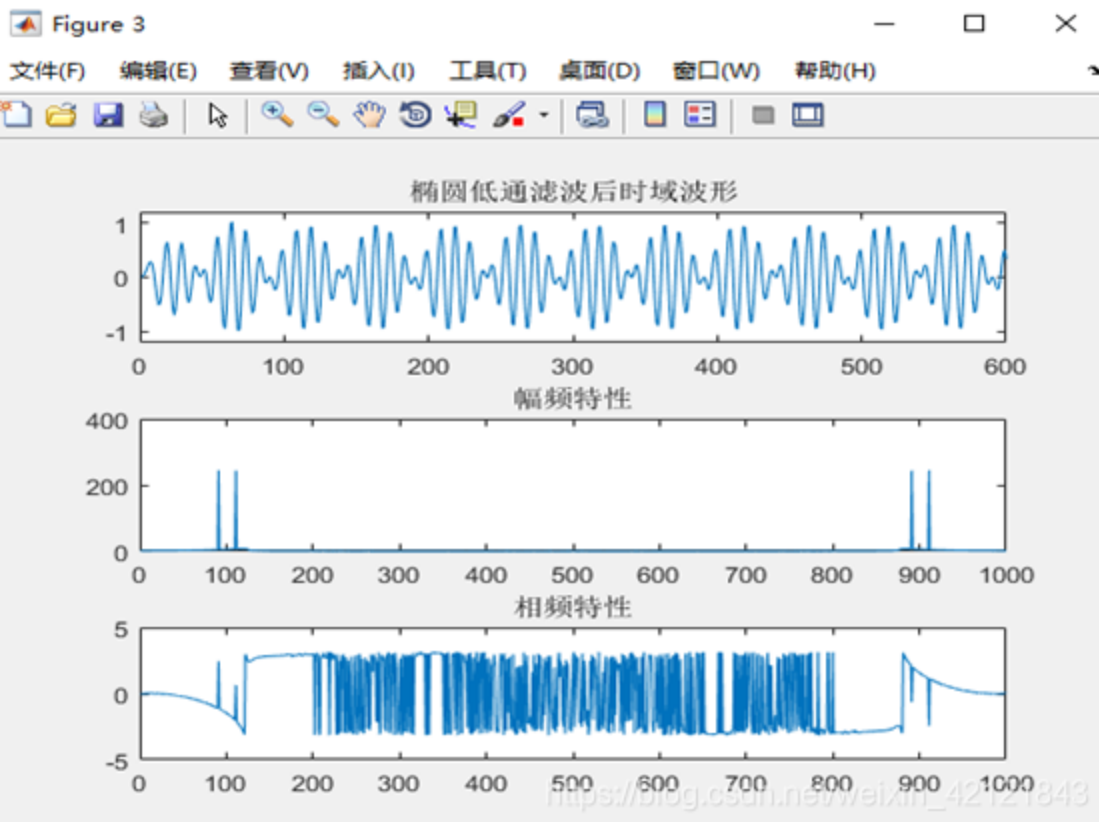
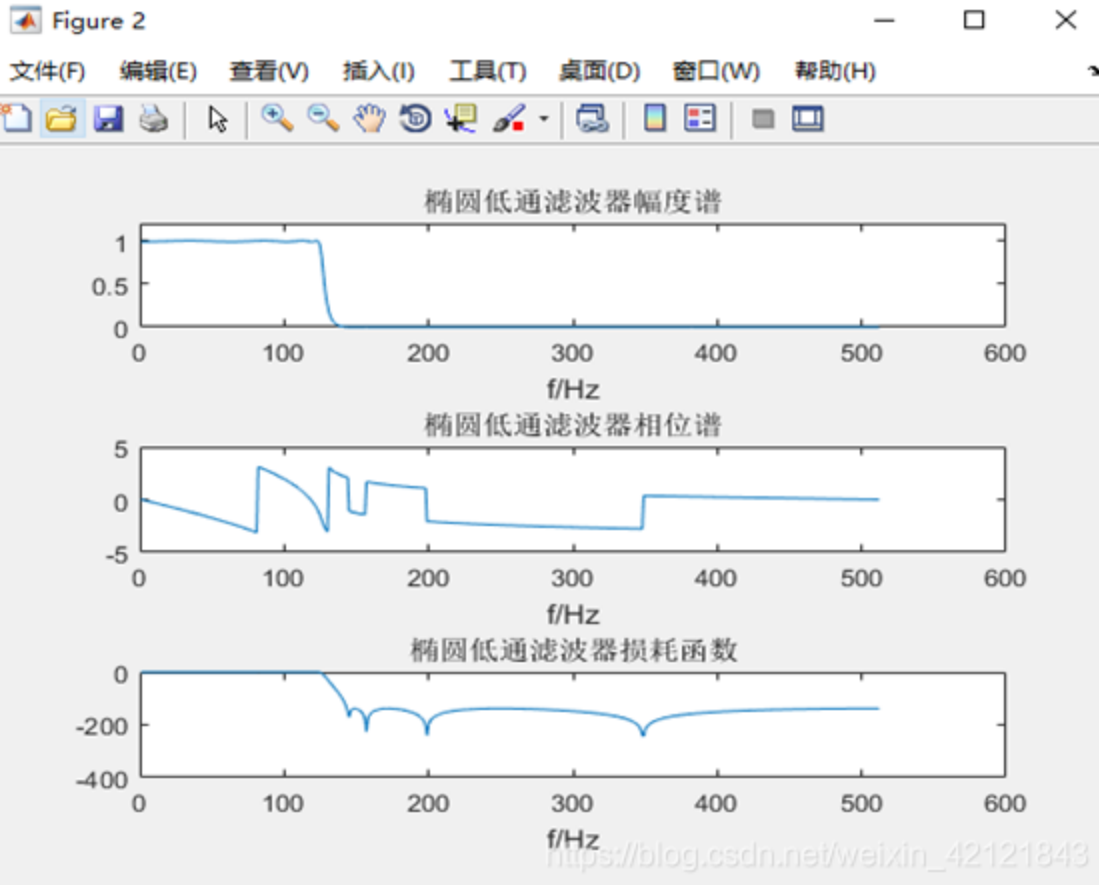


Figure 4

文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)

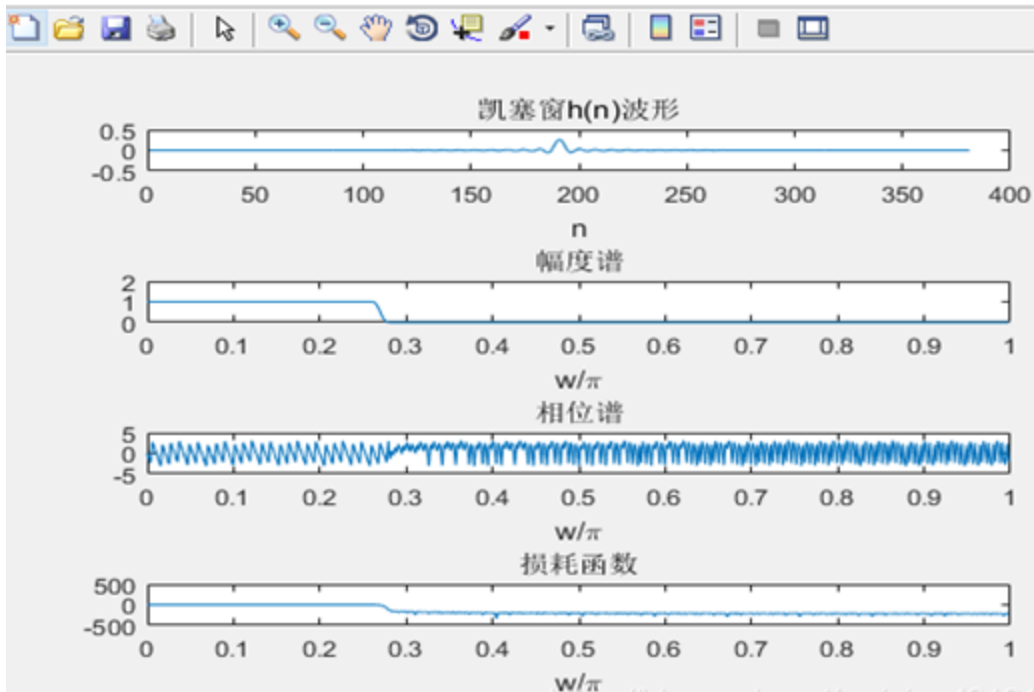
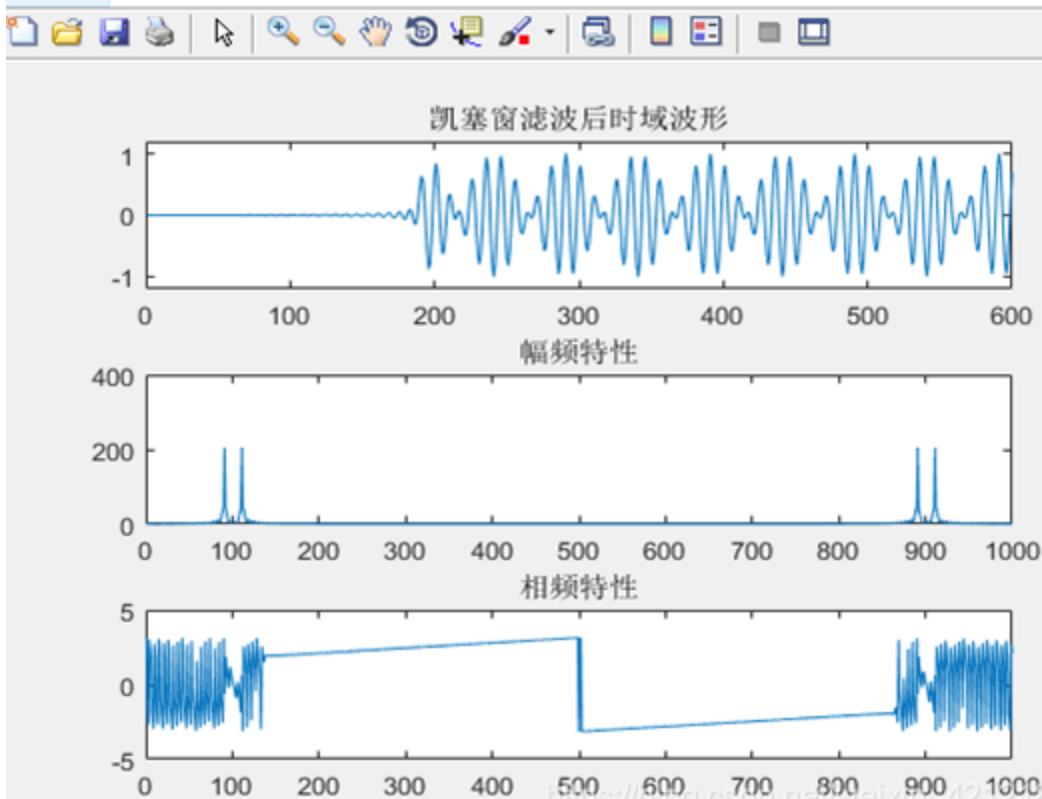
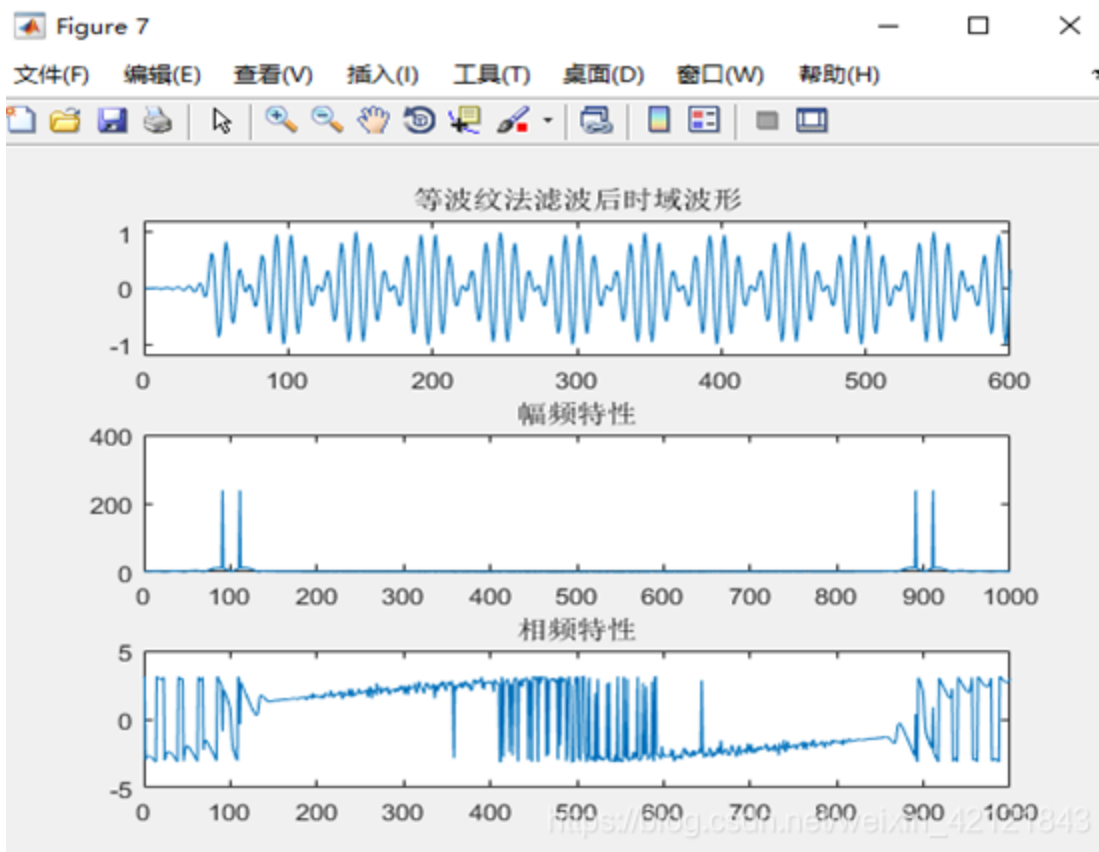
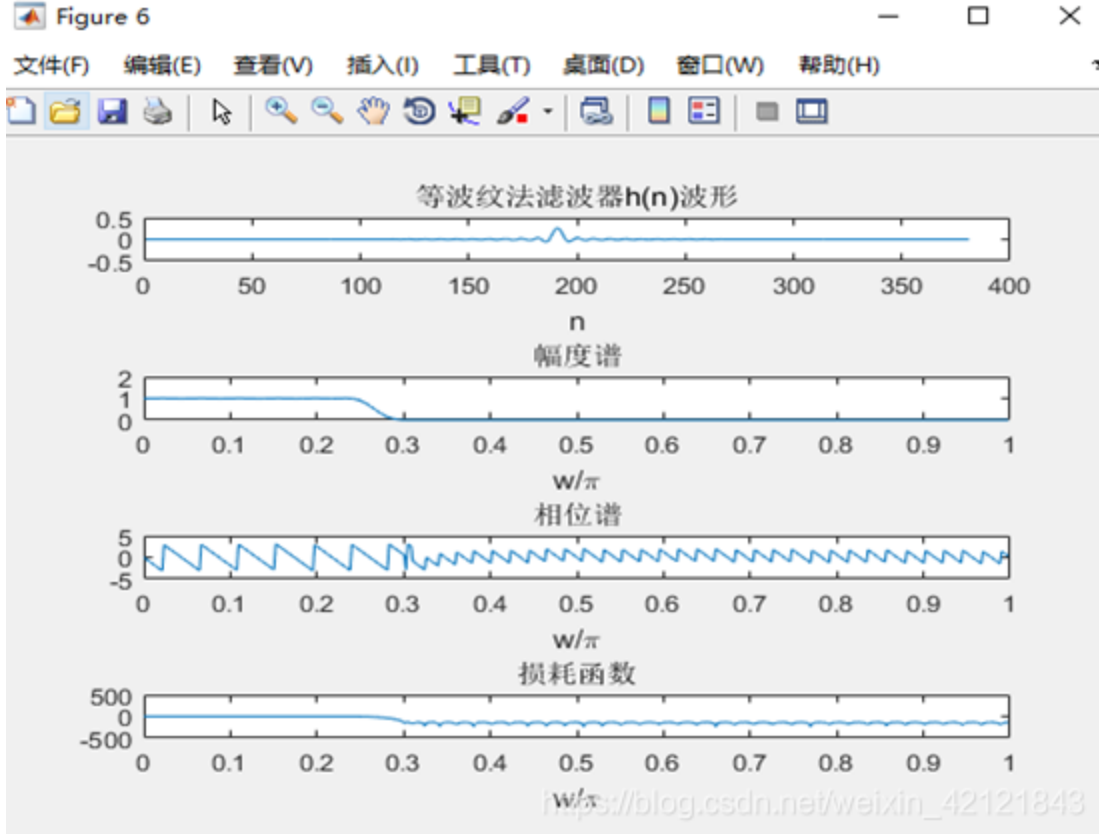
https://blog.csdn.net/weixin_42121843

Figure 5

文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)

https://blog.csdn.net/weixin_42121843



总结

通过matlab编程，更深刻掌握了数字信号处理的知识，very interesting