案例七：FIR低通滤波

1. 案例目的

•熟悉FIR数字滤波器窗函法和频率抽样法设计的一般原理

•理解窗函数FIR滤波器设计的具体方法和步骤

•理解FIR滤波器的阶数对滤波器性能的影响

2 案例内容

•针对FIR低通滤波器，改变N=31、101、301、1001三种长度，

观察低通滤波器效果。参考程序：FIR\_LOWPASS\_blackman

•根据不同窗函数设计如下滤波器，并与加矩形窗时的通带、阻带

性能比较。

1）：髙通滤波器设计，用hanning(N)窗。

2）：带通滤波器设计，用hamming(N)窗。

3）：带阻滤波器设计，用blackman(N)窗。

4）:带阻滤波器设计，用triang(N)窗。

设计滤波器，用hamming (N)窗。

3 案例知识点

Matlab编程

FIR低通滤波器

4 案例时长

共2学时，具体安排如下：

• 学习FIR低通滤波器（0.5学时）

• 进行相关实验（1学时）

• 查看实验结果（0.5学时）

• 对实验结果进行分析（0.5学时）

5 案例实验环境

**•操作系统：**

1）Windows 10 x64位操作系统

**•软件环境：**

1）MatlabR2019bx64

6 案例分析

本案例主要分为以下3部分：

1）针对FIR低通滤波器，改变N=31、101、301、1001三种长度，

观察低通滤波器效果。

2）用hamming窗设计带通滤波器，用blackman窗设计带阻滤波器；

3）设计滤波器，用hamming (N)窗。

7 案例实验过程

案例的实验过程可分为以下4个步骤：

•提出问题。

•问题的来源与求解思路。

•实验数据以及分析过程。

•得出实验结论。

7.1设置N的长度

针对FIR低通滤波器，改变N=31、101、301、1001三种长度，观察低通滤波器效果。

对滤波器的时域函数h(n)加窗函数截取时，截取的点数N与滤波器的滤波效果是否有关。

可以设置相同的输入信号，使其分别通过采样点数为31、101、301、1001的滤波器，观察滤波器的频域图形与输出信号的频域波形，对比滤波器通过滤波器前后的图形观察输出结果。以此来判断滤波器的滤波效果与N是否有关。

实验数据与分析过程

以blackman窗为例分析N的值与滤波器滤波效果的关系。

输入信号的频率为150Hz，噪声的频率为190Hz

抽样频率选取1000Hz。

输入信号未通过滤波器的频域波形

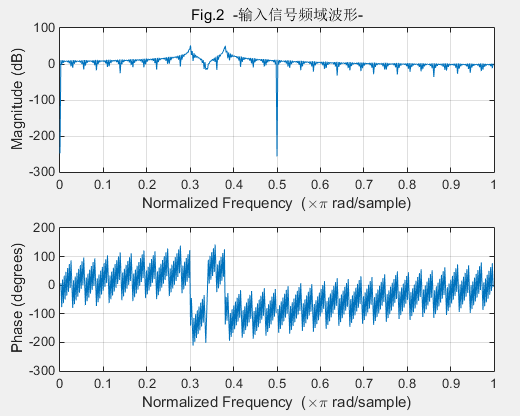


图1 N=31点的滤波器的频域波形

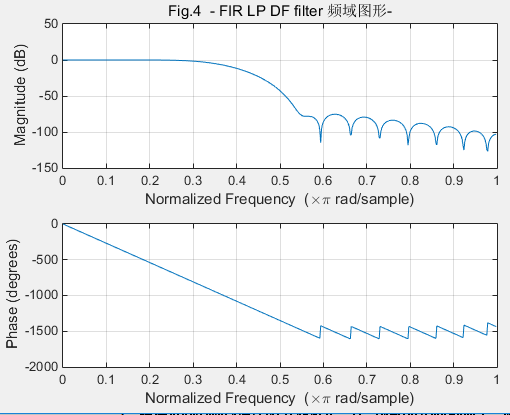


图2 通过N=31点的滤波器后输入信号的频域波形

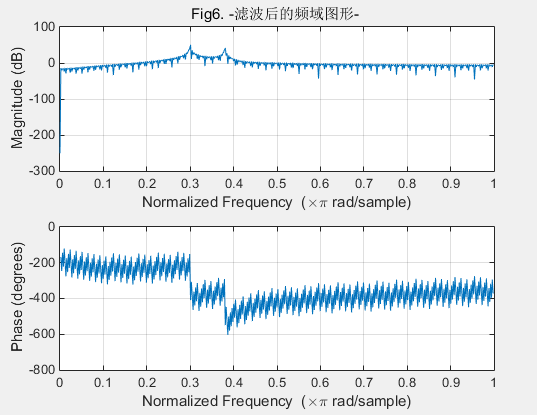


图3 N=101点的滤波器频域波形

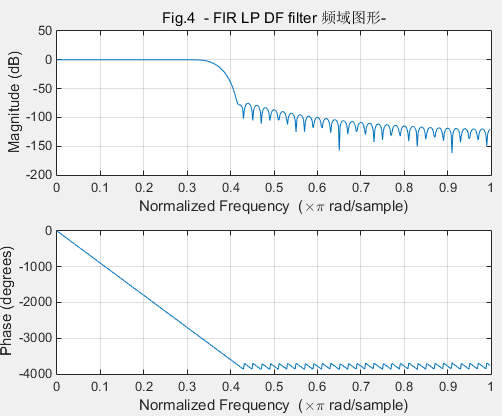


图4 通过N=101点的滤波器后的输入信号的频域波形

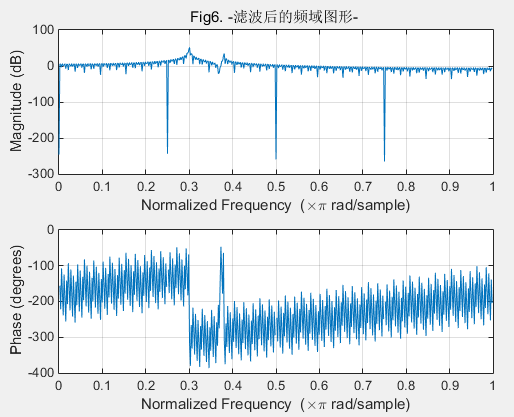


图5 N=301时滤波器频域的波形

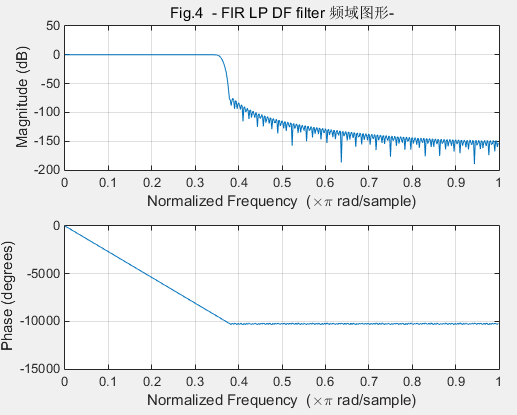


图6 当N=301时的滤波效果

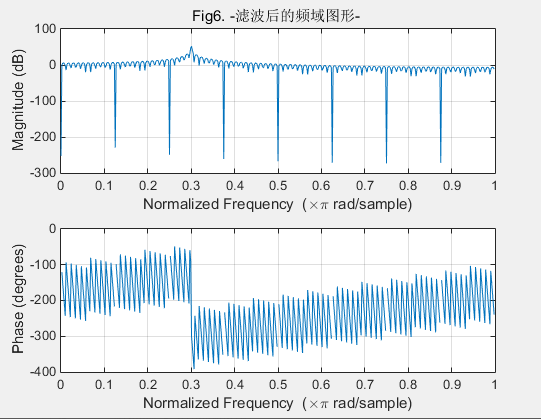


图7 N=1001时滤波器频域的波形

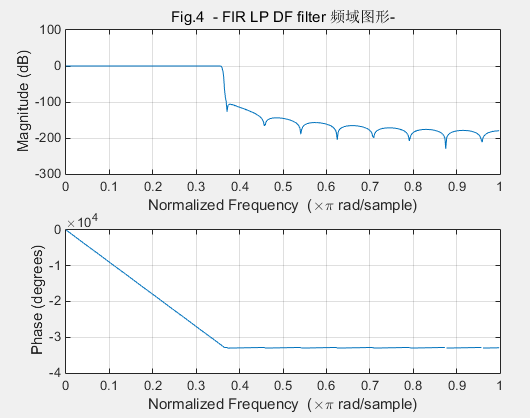


图8 当N=1001时的滤波效果

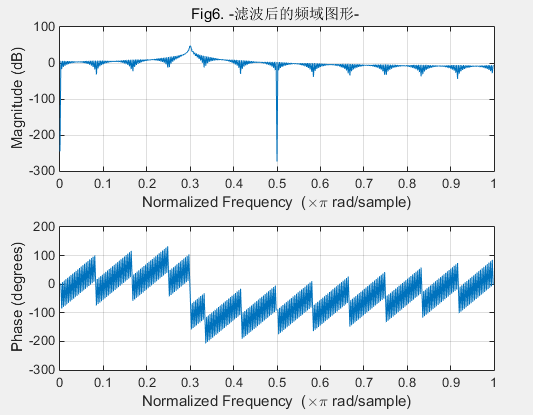


图9 滤波后的频域图形

经过不同N值的比较，发现当N增加时，滤波器的性能变好，N越大性能越好。当N=101点时还无法过滤出噪声。当N>=301时开始可以过滤出噪声，当N=1001时滤波器性能最好。

可以证明N的值不同滤波器的滤波效果不同N的值越大，滤波效果越好。

7.2带通滤波器与带阻滤波器的设计

带通与带阻滤波器的设计有多种方法，本次实验是通过求解相应的低通滤波器，经过低通滤波器与全通滤波器的加减来实现带阻和带通滤波器。

blackman窗带阻滤波器的设计先通过设计一个带通滤波器，然后用全通滤波器减去带通滤波器获得带阻滤波器。

带通滤波器的数据与主要代码：

【代码1】

window\_B=blackman(N);

low\_pass\_fc=160;

loww=140;

wc=2\*pi\*low\_pass\_fc/fs;

wc1=2\*pi\*loww/fs;

for i=0:N-1

if i==M/2

h(i+1)=(pi-wc+wc1)/pi; % 当i==M/2时的极限值

else

h(i+1)=(sin((i-M/2)\*pi)/((i-M/2)\*pi)-sin((i-M/2)\*wc)/((i-M/2)\*pi)+sin((i-M/2)\*wc1)/((i-M/2)\*pi)); % 低通滤波器的h(n)表达式

end

end

h=h.\*window\_B';

输出结果

输入信号频域波形

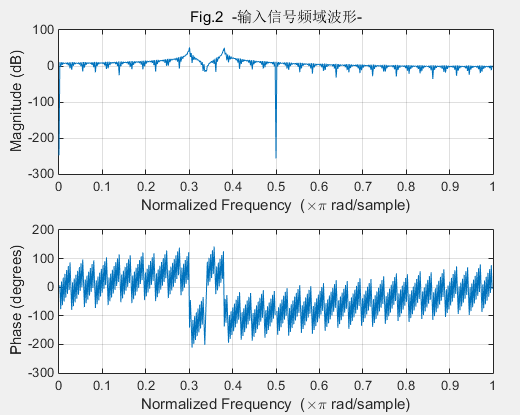


图10 输入信号频域波形

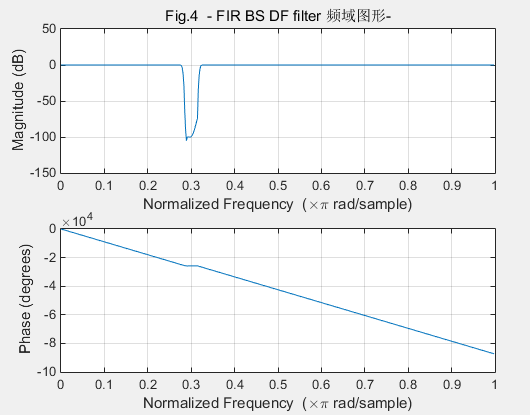


图11 输出信号波形

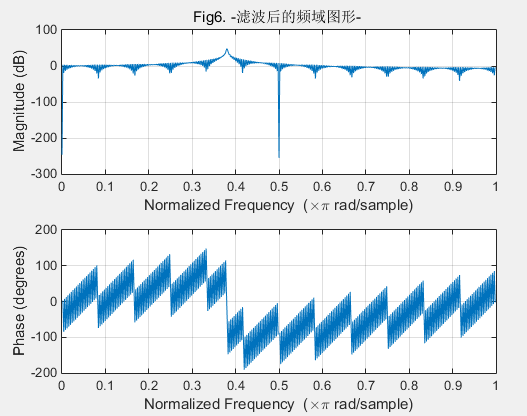


图12 滤波后的频域波形

（2）hamming窗带通滤波器的设计

带通滤波器的设计可以通过两个低通滤波器相减获得

【代码2】

window\_B=hamming(N);

M=N-1;

wc=2\*pi\*low\_pass\_fc/fs;

wc1=2\*pi\*loww/fs;

for i=0:N-1

if i==M/2

h(i+1)=(wc-wc1)/pi; % 当i==M/2时的极限值

else

h(i+1)=(sin((i-M/2)\*wc)/((i-M/2)\*pi)-sin((i-M/2)\*wc1)/((i-M/2)\*pi)); % 低通滤波器的h(n)表达式

end

end

输入信号原频域图形：

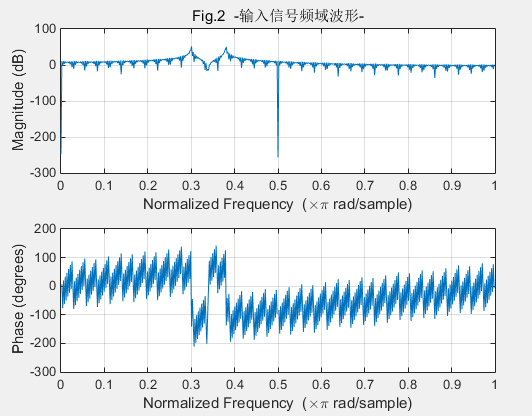


图13 滤波器频域图形

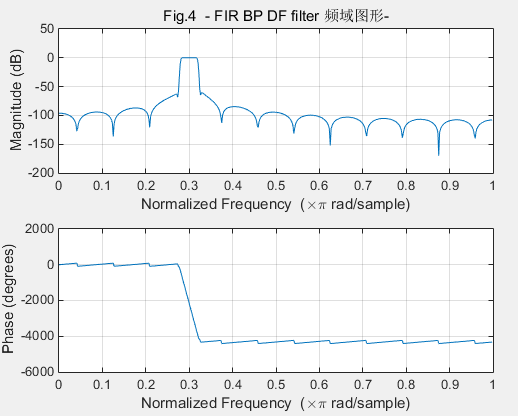


图14 滤波后信号频域波形

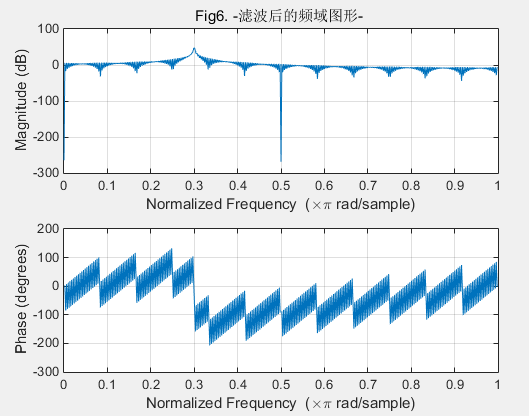


图15 滤波后的频域图形

7.3滤波器设计方案

带通和带阻滤波器的设计可以转化为相应的全通滤波器与带通滤波器的组合来实现，在实际中，低通滤波器的设计已经很成熟，所以设计滤波器可以转化为设计带通滤波器。

7.4设计多通带多阻带滤波器

实际中需要的滤波器往往是多通带滤波器已达到较好的滤波效果。多通带的滤波器可以通过设计多阻带滤波器，然后用全通滤波器减去相应的带阻滤波器即可获得。

7.5实验数据与分析过程

【代码3】

f1=120;

f2=125;

f3=130;

f4=135;

f5=140;

f6=145;

f7=150;

fl1=110;

fh1=124;

fl2=126;

fh2=134;

fl3=136;

fh3=144;

fl4=146;

fh4=155;

window\_B=hamming(N);

M=N-1;

wl1=2\*pi\*fl1/fs;

wh1=2\*pi\*fh1/fs;

wl2=2\*pi\*fl2/fs;

wh2=2\*pi\*fh2/fs;

wl3=2\*pi\*fl3/fs;

wh3=2\*pi\*fh3/fs;

wl4=2\*pi\*fl4/fs;

wh4=2\*pi\*fh4/fs;

for i=0:N-1

if i==M/2

h(i+1)=(pi-(wh1-wl1+wh2-wl2+wh3-wl3+wh4-wl4))/pi; % 当i==M/2时的极限值

else

h(i+1)=(sin((i-M/2)\*pi)/((i-M/2)\*pi)-(sin((i-M/2)\*wh1)/((i-M/2)\*pi)-sin((i-M/2)\*wl1)/((i-M/2)\*pi)+sin((i-M/2)\*wh2)/((i-M/2)\*pi)-sin((i-M/2)\*wl2)/((i-M/2)\*pi)+sin((i-M/2)\*wh3)/((i-M/2)\*pi)-sin((i-M/2)\*wl3)/((i-M/2)\*pi)+sin((i-M/2)\*wh4)/((i-M/2)\*pi)-sin((i-M/2)\*wl4)/((i-M/2)\*pi))); % 低通滤波器的h(n)表达式

end

end

h=h.\*window\_B';

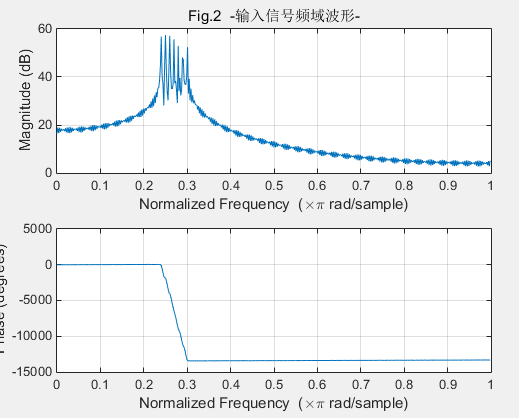


图16 输入信号频域图形

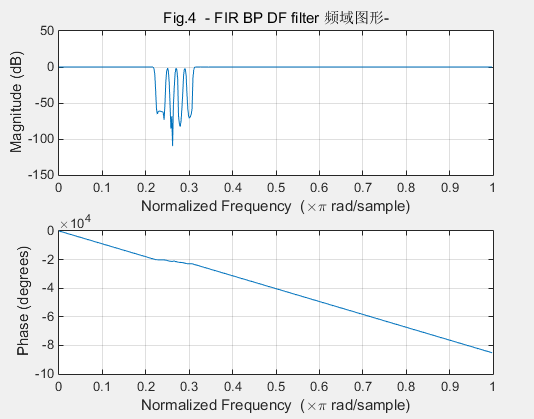


图17滤波器频域图形，通带带宽为2Hz

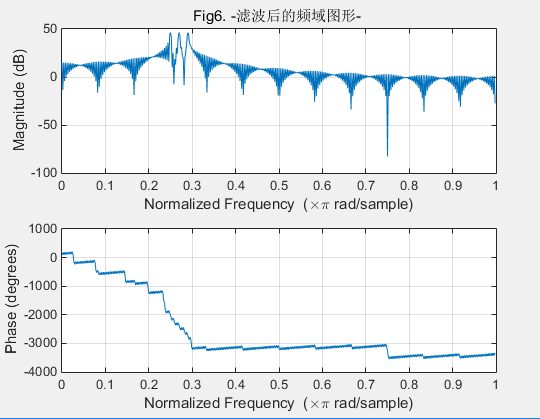


图18 滤波后的频域图形

7.6案例实验结果

多通带的滤波器设计可以通过，先设计多阻带滤波器，然后用全通滤波器减去多阻带滤波器进而获得多通带滤波器，多阻带滤波器可以通过低通滤波器设计，所以滤波器的设计都可以用低通滤波器和全通滤波器来实现。

8 案例实验代码

【案例代码】

%====FIR\_LOWPASS\_blackman===============% 第一步：初始化参数

clear all;

close all;

pi=4\*atan(1);

f1=150; % f1=50 is signal

f2=190; % f2 is noise.

fs=1000; % sampling frequency

t=1/fs; % sampling period

low\_pass\_fc=180; % cutoff frequency is 200Hz

length\_x=1500; % input signal length

%----------第二步：产生（信号+噪声），并用函数freqz显示其DTFT变换

input\_x=zeros(1,length\_x);

for i=1:1:length\_x

input\_x(i)=sin(2\*pi\*f1\*i\*t)+sin(2\*pi\*f2\*i\*t);

end

figure(1); % figure 1 input\_x discrete time figure

plot(input\_x); %show x[] in time domain

title('Fig.1 -输入信号时域波形-');

grid on;

figure(2); % figure 2

freqz(input\_x);

title('Fig.2 -输入信号频域波形-'); %show x[] in frequence domain

% 第三步：设计滤波器的h(n)，不同的滤波器h(n)不同，加不同的窗函数产生不同的滤波器频谱

N=101;

window\_B=blackman(N);

M=N-1;

wc=2\*pi\*low\_pass\_fc/fs;

for i=0:N-1

if i==M/2

h(i+1)=wc/pi; % 当i==M/2时的极限值

else

h(i+1)=sin((i-M/2)\*wc)/((i-M/2)\*pi); % 低通滤波器的h(n)表达式

end

end

h=h.\*window\_B';

figure(3);

plot(h);

title('Fig.3 - FIR LP DF 时域波形-');

figure(4);

freqz(h); %显示低通滤波器的频率响应函数

title('Fig.4 - FIR LP DF filter 频域图形-'); %show x[] in frequence domain

%%%%%%%%%%%%%% 第四步 计算时域卷积求y(n)输出，并显示其时、频图形

for i=1:(length\_x-N) %inatialze y(1) to y(N) as zero.

y(i)=0;

end

for i=1:N

h\_inverse(i)=h(N+1-i);

end

for j=1:(length\_x-N)

for i=1:N

y(j)=y(j)+h\_inverse(i)\*input\_x(i+j-1);

end

end

figure(5);

plot(y);

title('Fig5. -滤波后的时域图形-');

figure(6);

freqz(y);

title('Fig6. -滤波后的频域图形- ');

%END %\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* IIR filter recursive computation to figure out y()

【案例代码】

%====FIR\_LOWPASS\_blackman===============% 第一步：初始化参数

clear all;

close all;

f1=150; % f1=50 is signal

f2=190; % f2 is noise.

fs=1000; % sampling frequency

t=1/fs; % sampling period

low\_pass\_fc=160; % cutoff frequency is 200Hz

loww=140;

length\_x=1500; % input signal length

%----------第二步：产生（信号+噪声），并用函数freqz显示其DTFT变换

input\_x=zeros(1,length\_x);

for i=1:1:length\_x

input\_x(i)=sin(2\*pi\*f1\*i\*t)+sin(2\*pi\*f2\*i\*t);

end

figure(1); % figure 1 input\_x discrete time figure

plot(input\_x); %show x[] in time domain

title('Fig.1 -输入信号时域波形-');

grid on;

figure(2); % figure 2

freqz(input\_x);

title('Fig.2 -输入信号频域波形-'); %show x[] in frequence domain

% 第三步：设计滤波器的h(n)，不同的滤波器h(n)不同，加不同的窗函数产生不同的滤波器频谱

N=1001;

window\_B=blackman(N);

M=N-1;

wc=2\*pi\*low\_pass\_fc/fs;

wc1=2\*pi\*loww/fs;

for i=0:N-1

if i==M/2

h(i+1)=(pi-wc+wc1)/pi; % 当i==M/2时的极限值

else

h(i+1)=(sin((i-M/2)\*pi)/((i-M/2)\*pi)-sin((i-M/2)\*wc)/((i-M/2)\*pi)+sin((i-M/2)\*wc1)/((i-M/2)\*pi)); % 低通滤波器的h(n)表达式

end

end

h=h.\*window\_B';

figure(3);

plot(h);

title('Fig.3 - FIR BS DF 时域波形-');

figure(4);

freqz(h); %显示低通滤波器的频率响应函数

title('Fig.4 - FIR BS DF filter 频域图形-'); %show x[] in frequence domain

%%%%%%%%%%%%%% 第四步 计算时域卷积求y(n)输出，并显示其时、频图形

for i=1:(length\_x-N) %inatialze y(1) to y(N) as zero.

y(i)=0;

end

for i=1:N

h\_inverse(i)=h(N+1-i);

end

for j=1:(length\_x-N)

for i=1:N

y(j)=y(j)+h\_inverse(i)\*input\_x(i+j-1);

end

end

figure(5);

plot(y);

title('Fig5. -滤波后的时域图形-');

figure(6);

freqz(y);

title('Fig6. -滤波后的频域图形- ');

%END %\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* IIR filter recursive computation to figure out y()

【案例代码】

%====FIR\_LOWPASS\_blackman===============% 第一步：初始化参数

clear all;

close all;

f1=150; % f1=50 is signal

f2=190; % f2 is noise.

fs=1000; % sampling frequency

t=1/fs; % sampling period

low\_pass\_fc=160; % cutoff frequency is 200Hz

loww=140;

length\_x=1500; % input signal length

%----------第二步：产生（信号+噪声），并用函数freqz显示其DTFT变换

input\_x=zeros(1,length\_x);

for i=1:1:length\_x

input\_x(i)=sin(2\*pi\*f1\*i\*t)+sin(2\*pi\*f2\*i\*t);

end

figure(1); % figure 1 input\_x discrete time figure

plot(input\_x); %show x[] in time domain

title('Fig.1 -输入信号时域波形-');

grid on;

figure(2); % figure 2

freqz(input\_x);

title('Fig.2 -输入信号频域波形-'); %show x[] in frequence domain

% 第三步：设计滤波器的h(n)，不同的滤波器h(n)不同，加不同的窗函数产生不同的滤波器频谱

N=1001;

window\_B=hamming(N);

M=N-1;

wc=2\*pi\*low\_pass\_fc/fs;

wc1=2\*pi\*loww/fs;

for i=0:N-1

if i==M/2

h(i+1)=(wc-wc1)/pi; % 当i==M/2时的极限值

else

h(i+1)=(sin((i-M/2)\*wc)/((i-M/2)\*pi)-sin((i-M/2)\*wc1)/((i-M/2)\*pi)); % 低通滤波器的h(n)表达式

end

end

h=h.\*window\_B';

figure(3);

plot(h);

title('Fig.3 - FIR BP DF 时域波形-');

figure(4);

freqz(h); %显示低通滤波器的频率响应函数

title('Fig.4 - FIR BP DF filter 频域图形-'); %show x[] in frequence domain

%%%%%%%%%%%%%% 第四步 计算时域卷积求y(n)输出，并显示其时、频图形

for i=1:(length\_x-N) %inatialze y(1) to y(N) as zero.

y(i)=0;

end

for i=1:N

h\_inverse(i)=h(N+1-i);

end

for j=1:(length\_x-N)

for i=1:N

y(j)=y(j)+h\_inverse(i)\*input\_x(i+j-1);

end

end

figure(5);

plot(y);

title('Fig5. -滤波后的时域图形-');

figure(6);

freqz(y);

title('Fig6. -滤波后的频域图形- ');

%END %\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* IIR filter recursive computation to figure out y()

【案例代码】

%====FIR\_LOWPASS\_blackman===============% 第一步：初始化参数

clear all;

close all;

f1=120;

f2=125;

f3=130;

f4=135;

f5=140;

f6=145;

f7=150;

fl1=110;

fh1=124;

fl2=126;

fh2=134;

fl3=136;

fh3=144;

fl4=146;

fh4=155;

fs=1000; % sampling frequency

t=1/fs; % sampling period

length\_x=1500; % input signal length

%----------第二步：产生（信号+噪声），并用函数freqz显示其DTFT变换

input\_x=zeros(1,length\_x);

for i=1:1:length\_x

input\_x(i)=sin(2\*pi\*f1\*i\*t)+sin(2\*pi\*f2\*i\*t)+sin(2\*pi\*f3\*i\*t)+sin(2\*pi\*f4\*i\*t)+sin(2\*pi\*f5\*i\*t)+sin(2\*pi\*f6\*i\*t)+sin(2\*pi\*f7\*i\*t);

end

figure(1); % figure 1 input\_x discrete time figure

plot(input\_x); %show x[] in time domain

title('Fig.1 -输入信号时域波形-');

grid on;

figure(2); % figure 2

freqz(input\_x);

title('Fig.2 -输入信号频域波形-'); %show x[] in frequence domain

% 第三步：设计滤波器的h(n)，不同的滤波器h(n)不同，加不同的窗函数产生不同的滤波器频谱

N=1001;

window\_B=hamming(N);

M=N-1;

wl1=2\*pi\*fl1/fs;

wh1=2\*pi\*fh1/fs;

wl2=2\*pi\*fl2/fs;

wh2=2\*pi\*fh2/fs;

wl3=2\*pi\*fl3/fs;

wh3=2\*pi\*fh3/fs;

wl4=2\*pi\*fl4/fs;

wh4=2\*pi\*fh4/fs;

for i=0:N-1

if i==M/2

h(i+1)=(pi-(wh1-wl1+wh2-wl2+wh3-wl3+wh4-wl4))/pi; % 当i==M/2时的极限值

else

h(i+1)=(sin((i-M/2)\*pi)/((i-M/2)\*pi)-(sin((i-M/2)\*wh1)/((i-M/2)\*pi)-sin((i-M/2)\*wl1)/((i-M/2)\*pi)+sin((i-M/2)\*wh2)/((i-M/2)\*pi)-sin((i-M/2)\*wl2)/((i-M/2)\*pi)+sin((i-M/2)\*wh3)/((i-M/2)\*pi)-sin((i-M/2)\*wl3)/((i-M/2)\*pi)+sin((i-M/2)\*wh4)/((i-M/2)\*pi)-sin((i-M/2)\*wl4)/((i-M/2)\*pi))); % 低通滤波器的h(n)表达式

end

end

h=h.\*window\_B';

figure(3);

plot(h);

title('Fig.3 - FIR BP DF 时域波形-');

figure(4);

freqz(h); %显示低通滤波器的频率响应函数

title('Fig.4 - FIR BP DF filter 频域图形-'); %show x[] in frequence domain

%%%%%%%%%%%%%% 第四步 计算时域卷积求y(n)输出，并显示其时、频图形

for i=1:(length\_x-N) %inatialze y(1) to y(N) as zero.

y(i)=0;

end

for i=1:N

h\_inverse(i)=h(N+1-i);

end

for j=1:(length\_x-N)

for i=1:N

y(j)=y(j)+h\_inverse(i)\*input\_x(i+j-1);

end

end

figure(5);

plot(y);

title('Fig5. -滤波后的时域图形-');

figure(6);

freqz(y);

title('Fig6. -滤波后的频域图形- ');

%END %\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* IIR filter recursive computation to figure out y()