时域与频域分析实战代码总结

1.有限长序列处理与MATLAB实现

```
x = [1,1,1,1];
%信号序列
A = 200;
%DTFT近似点数
w1 = 0:2/A:2;
w = pi*w1.';
xw = 1+exp(-j*w)+exp(-j*w*2)+exp(-j*w*3)+exp(-j*w*4);
%DTFT频谱分析
stem(w1.',abs(xw))
%DTFT频谱绘制
N = 32
y1 = fft(x,N);
n = 0:N-1;
subplot(3,1,1);
%32点DFT频谱分析
stem(n,abs(y1),'ok');
title('N=32');
N = 64;
y2 = fft(x,N);
n=0:N-1;
subplot(3,1,2);
%64点DFT频谱分析
stem(n,abs(y2),'ok');
title('N=64');
N = 128;
y3 = fft(x,N);
n = 0:N-1;
```

```
subplot(3,1,3);
%128点DFT频谱分析
stem(n,abs(y3),'ok');
title('N=128');
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 hi = 0.00001;
 t = -0.005:hi:0.005;
 %取值台阶
 xn = exp(-500*abs(t));
 %信号序列
 Ww = 2*pi*2000;
 N = 1000;
 n = 0:2:N;
 W = n*Ww/N;
 %傅立叶变换
 Xn = xn*exp(-j*t'*W)*hi;
 Xa = real(Xn);
 W = [-fliplr(W),W(2:501)];
 Xn = [fliplr(Xn),Xn(2:501)];
 subplot(1,2,1);
 plot(t*1000,xn,'o');
 xlabel('t(s)');
 ylabel('xn');
 title('源信号');
 subplot(1,2,2);
 plot(W/(2*pi*1000),Xn*1000,'o');
 xlabel('f(KHz)');
 ylabel('jw');
 title('对应序列的傅立叶变换');
```

2.N 点离散傅里叶变换与 MATLAB 实现

```
N = 32;
N1 = 16;
n = 0:N-1;
k = 0:N1-1;
x1n = exp(j*pi*n/8);
%x1 信号
y1 = fft(x1n,N);
%[x1]N 点 DFT
y2 = fft(x1n,N1);
%[x1]N1 点 DFT
x2n = cos(pi*n/16);
%x2 信号
y3 = fft(x2n,N);
%[x2]N 点 DFT
y4 = fft(x2n,N1);
%[x2]N1 点 DFT
subplot(2,2,1);
stem(n,abs(y1),'o');
ylabel('|y1|')
title('16 点的 DFT:x1')
subplot(2,2,2);
stem(n,abs(y3),'o');
ylabel('|y2|')
title('16 点的 DFT:x2')
subplot(2,2,3);
stem(k,abs(y2),'>');
ylabel('|y1|')
title('8 点的 DFT:x1')
```

```
subplot(2,2,4);
stem(k,abs(y4),'*');
ylabel('|y2|')
title('8 点的 DFT:x2')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
%N = 4;
%n = 0:N-1;
%k = n;
%xn = [0.25 0.5 0.75 1];
%Wn = exp(-j*pi*2/N);
%%xn 信号
%nk = n'*k;
%Wnk = Wn.^nk;
%xk = xn*Wnk
%Wnk0 = Wn.^(-nk);
%xn0 =xn*Wnk0/N
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xn=[0.25 0.5 0.75 1]
N=length(xn);
n=0:(N-1);
k=0:(N-1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xk=xn*exp(-j*2*pi/N).^(n'*k)
xn0=(xk*exp(j*2*pi/N).^(n'*k))/N
subplot(2,2,1); stem(n,xn);title('x(n)');
subplot(2,2,2); stem(n,xk); title('IDFT[X(k)]');
subplot(2,2,3); stem(k,abs(xk)); title('|X(k)|');
subplot(2,2,4); stem(k,angle(xk)); title('arg|X(k)|');
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
N = 4;
n = 0:N-1;
m = 2;
xn = [4 2 1 1/2];
nm = mod((n-m),N);
xm = xn(nm+1);
subplot(1,2,1);
stem(xn,'o');
xlabel('n');
ylabel('|xn|');
title('信息序列');
subplot(1,2,2);
stem(xm,'o');
xlabel('n');
ylabel('|xm|');
title('位移序列');
```

3. 时域圆周积分与 MATLAB 实现

```
n = 0:30;
x1 = [0.25 1.5 0.5 1.5 0.75];
x2 = 0.9.^n;
N = 30;
x1 = [x1 zeros(1, N-length(x1))];
x2 = [x2 zeros(1, N-length(x2))];
for n = 1:N
    for m = 1:N
        y(n,m) = x1(m)*x2(mod( (n-m), N)+1 );
    end
end
y = sum(y');
```

```
subplot(131);
stem( x1 );
title('x1');
subplot(132);
stem( x2 );
title('x2');
subplot(133);
stem( y );
title('y');
```

4.基于 FFT 的噪声信号分析与 MATLAB 实现

```
Fs = 5000;
% 采样值
T = 1/Fs;
% 采样间隔
L = 5000;
% 信号时长
t = (0:L-1)*T;
% 构造信号，其中包含幅值为 0.9 的 10 Hz 正弦信号和幅值为 1.1 的 300 Hz 正弦信号。
S = 0.9*sin(2*pi*10*t) + 1.1*sin(2*pi*300*t);
% 将均值为零、方差为 4 的白噪声叠加 S 信号。
Y = S + 2*randn(size(t));
% 在时域中绘制噪声信号。通过查看信号 Y(t) 很难确定频率分量。
Subplot(121);
stem(5000*t(1:100),Y(1:100))
title('白噪声混合输出')
xlabel('t (ms)')
ylabel('Y(t)')
Y1 = fft(Y);
```

```
% 计算双边频谱 P2。然后基于 P2 和偶数信号长度 L 计算单边频谱 P1。
P2 = abs(Y1/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);
% 定义频域 f 并绘制单边幅值频谱 P1。与预期相符，由于增加了噪声，
% 幅值并不精确等于 S。
f = Fs*(0:(L/2))/L;
subplot(122);
stem(f,P1)
title('信号幅频 Amplitude Spectrum of X(t)')
xlabel('f (Hz)')
ylabel('|P1(f)|')
```

5.高斯脉冲信号分析与 MATLAB 实现

```
Fs = 80;
% 采样频率
t = -0.5:1/Fs:0.5;
% 时间间隔
L = length(t);
% 信号时长
X = 1/(2*sqrt(3*pi*0.01))*(exp(-t.^2/(3*0.01)));
% 在时域中绘制脉冲
subplot(121);
stem(t,X)
title('信号时域波形')
xlabel('(t)')
ylabel('X(t)')
% fft 将信号转换为频域形式，从原始信号长度确定是下一个 2 次幂的新输入长度。
% 这将用尾随零填充信号 X 以改善 fft 的性能。
```

```
n = 2^nextpow2(L);
% 将高斯脉冲转换为频域。
Y = fft(X,n);
% 定义频域并绘制唯一频率。
f = Fs*(0:(n/2))/n;
P = abs(Y/n);
subplot(122);
stem(f,P(1:n/2+1))
title('信号频域形式')
xlabel('(f)')
ylabel('|P(f)|')
```

6.三角函数 FFT 分析与 MATLAB 实现

```
Fs = 3000;
% 采样频率
T = 1/Fs;
% 采样间隔
L = 2000;
% 信号时长
t = (0:L-1)*T;
% 创建一个矩阵，其中每一行代表一个频率经过缩放的余弦波
% 结果 X 为 3×L 矩阵，第一行为 50，第二行为 150，第三行为 300
x1 = sin(2*pi*150*t);
x2 = sin(2*pi*175*t);
x3 = sin(2*pi*325*t);
X = [x1; x2; x3];
% 比较频率分布
for i = 1:3
    subplot(3,2,i)
    stem(t(1:100),X(i,1:100))
```

```
    title(['Row ',num2str(i),'  时域波形'])
end
n = 2^nextpow2(L);
% 指定 dim 参数沿 X 的行（即对每个信号）使用 fft
dim = 2;
% 计算信号的傅里叶变换
Y = fft(X,n,dim);
% 计算每个信号的双边谱和单边谱
P2 = abs(Y/L);
P1 = P2(:,1:n/2+1);
P1(:,2:end-1) = 2*P1(:,2:end-1);
for i=1:3
    subplot(3,2,i+3)
    plot(0:(Fs/n):(Fs/2-Fs/n),P1(i,1:n/2))
    title(['Row ',num2str(i),'  频域波形'])
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N1=1024;
N2=256;
fs=1;
f1=.1;
f2=.2;
f3=.3;
a1=5;
a2=3;
a3=2;
w=2*pi/fs;
x1=a1*cos(w*f1*(0:N1-1))+a2*sin(w*f2*(0:N1-1))+a3*sin(w*f3*(0:N1-1))+randn(1,N1);
```

```
% FFT 频谱；
subplot(2,2,1);
plot(x1(1:N1/4));
title('原始信号 x1');
f=-0.5:1/N1:0.5-1/N1;
X=fft(x1);
subplot(2,2,2);
plot(f,fftshift(abs(X)));
title('频域信号 x1');
x2=a1*cos(w*f1*(0:N2-1))+a2*sin(w*f2*(0:N2-1))+a3*sin(w*f3*(0:N2-
1))+randn(1,N2);
subplot(2,2,3);
stem(x1(1:N2/4));
title('原始信号 x2');
f=-0.5:1/N2:0.5-1/N2;
X=fft(x2);
subplot(2,2,4);
stem(f,fftshift(abs(X)));
title('频域信号 x2');
```

7. FFT 快速卷积与 MATLAB 实现

```
xn = [0 0.25 0.5 0.75 1 0.5 0.25 0];
%序列 xn[]
xk = fft (xn);
yk = xk.*xk;
yn = ifft (yk);
%FFT 循环卷积
yn1 = conv (xn, xn)
%conv 线性卷积
subplot (311);
```

```
stem (xn);
title('xn');
subplot (312);
stem (yn);
title('圆周卷积');
subplot (313);
stem (yn1);
title('线性卷积');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Fs = 50;
N = 128;
%采样频率
n = 0:N-1;
%采样点数
xn = 2*cos ( 3*pi*n/Fs ) - cos ( 6*pi*n/Fs ) - cos ( 12*pi*n/Fs ) -
cos ( 24*pi*n/Fs );
xk = fft (xn , N);
f = ( 0: N-1 ) * Fs/N;
xkAM = abs (xk);
xkAN = angle (xk);
subplot (121);
stem (f,xkAM);
xlabel('f');
ylabel('幅度');
title('输出幅度响应');
axis( [0 20 0 140 ] );
subplot (122);
stem (f,xkAN);
xlabel('f');
```

```
ylabel('相位');
title('输出相位响应');
axis( [0 20 -4 4 ] );
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N1=32;
N2=64;
%FFT 的变换长度 N1 和 N2
n=0:N1-1;
k1=0:N1-1;
k2=0:N2-1;
t=2*pi*(0:2047)/2048;
Xw=(1-exp(-j*6*t))./( 1-exp(-j*0.8*t));
%对 x(n)的连续采样
xn=[(n>=0)&(n<8)];
%x(n)
X1k=fft(xn,N1);
%计算 N1=32 点的 X1k
X2k=fft(xn,N2);
%计算 N2=64 点的 X2k
subplot(4,2,1);
plot(t/pi,abs(Xw));
xlabel('t');
ylabel('X1');
subplot(4,2,2);
plot(t,angle(Xw));
axis([0,2,-pi,pi]);
line([0,2],[0,0]);
xlabel('t');
ylabel('X2');
```

```
subplot(4,2,3);
stem(k1,abs(X1k),'o');
axis([0,N1,0,8]);
xlabel('k(t=2pik/N1)');
ylabel('|X1k|幅度');
hold on
plot(N1/2*t/pi,abs(Xw))
%图形上叠加连续频谱的幅度曲线
subplot(4,2,4);
stem(k1,angle(X1k));
axis([0,N1,-pi,pi]);
line([0,N1],[0,0]);
xlabel('k(w=2pik/N1)') ;
ylabel('[X1k]相位');
hold on
plot(N1/2*t/pi,angle(Xw))
%图形上叠加连续频谱的相位曲线
subplot(4,2,5);
stem(k2,abs(X2k),'*');
axis([0,N2,0,8]);
xlabel('k(t=2pik/N2)');
ylabel('|X2k|幅度');
hold on
plot(N2/2*t/pi,abs(Xw))
subplot(4,2,6);
stem(k2,angle(X2k),'>');
axis([0,N2/4,-pi,pi]);
line([0,N2],[0,0]);
xlabel('t') ;
```

```
 ylabel('[X2k]相位');
 hold on
 plot(N2/2*t/pi,angle(Xw))
 subplot(4,2,7);
 stem(k2,abs(X2k),'*');
 axis([0,N2/4,0,8]);
 xlabel('k(t=2pik/N2/4)');
 ylabel('|X2k|幅度');
 subplot(4,2,8);
 stem(k1,abs(X1k),'o');
 axis([0,N1/4,0,8]);
 xlabel('k(t=2pik/N1/4)');
 ylabel('|X1k|幅度');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N1=10;
n1=0:N1-1;
x1=sin(0.2*pi*n1)+sin(0.6*pi*n1);
Xk1=fft(x1,N1);
%N1 个点的 FFT
k1=0:N1-1;
t1=2*pi/10*k1;
subplot(4,2,1);
stem(n1,x1,'o');
axis([0,N1,-2.5,2.5]);
title('x(n),n=N1')
subplot(4,2,2);
stem(t1/pi,abs(Xk1),'*');
axis([0,1,0,10]);
title('DFT[x(n)]')
```

```
N2=100;
n2=0:N2-1;
x2=[x1(1:1:N1) zeros(1,N2-N1)];
Xk2=fft(x2,N2)
k2=0:N2-1;
t2=2*pi/100*k2;
subplot(4,2,3);
stem(n2,x2,'*');
axis([0,100,-2.5,2.5]);
title('信号 N2-N1=0')
subplot(4,2,4);
plot(t2/pi,abs(Xk2));
axis([0,1,0,10]);
title('DFT[x(n)]')
N3=100;
n3=0:N3-1;
x3=sin(0.2*pi*n3)+sin(0.6*pi*n3);
Xk3=fft(x3,N3)
k3=0:N3-1;
t3=2*pi/100*k3;
subplot(4,2,5);
stem(n3,x3,'>');
axis([0,100,-2.5,2.5]);
title('信号 x(n)拓展=100')
subplot(4,2,6);
plot(t3/pi,abs(Xk3),'.');
axis([0,1,0,60]);
title('DFT[x(n)]')
N4=40;
```

```
n4=0:N4-1;
x4=[x1(1:1:N1) zeros(1,N4-N1)];
Xk4=fft(x2,N4)
k4=0:N4-1;
t4=2*pi/N4*k4;
subplot(4,2,7);
stem(n4,x4,'*');
axis([0,100,-2.5,2.5]);
title('信号N4-N1=0')
subplot(4,2,8);
plot(t4/pi,abs(Xk4));
axis([0,1,0,10]);
title('DFT[x(n)]')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Fs=40;
%采样频率值Fs
N=200;
%采样点点数
n=0:N-1;
t=n/Fs;
%采样点间隔
f=3;
%设定正弦信号频率
y=cos ( 2*pi*f*t )+cos ( 3*pi*f*t );
%产生三角函数波形
subplot(121);
plot(t,y);
%三角函数时域波形
xlabel('t');
```

```
ylabel('y');
title('三角函数y时域波形');

y1=fft(y,N);
%fft运算变换频谱
A=abs(y1);
%幅值
f=(0:length(y1)-1)'*Fs/length(y1);
%进行对应的频率转换
subplot(122);
plot(f,A);
%频谱分析
xlabel('f');
ylabel('y1幅值');
title('对应N点y1幅频谱图');
grid;
figure(2)
Squ=abs(y1);
%均方根谱Squ
subplot(121);
plot(f,Squ);
xlabel('f(Hz)');
ylabel('均方根谱');
title('白噪声均方根谱');
grid;
Pow=Squ.^2;
%功率谱Pow
subplot(122);
plot(f,Pow);
```

```
xlabel('f(Hz)');
ylabel('功率谱');
title('白噪声功率谱');
grid;
figure(3)
X_ifft=ifft(y1);
y2=real(X_ifft);
t2=[0:length(X_ifft)-1]/Fs;
plot(t2,y2);
xlabel('t');
ylabel('y2');
title('基于 IFFT 得到三角函数波形');
grid;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Fs=100;
%采样频率值 Fs
t=-10:0.2:10;
x=rectpuls(t,5);
y=x(1:99);

subplot(231);
plot(t(1:99),y);
%门函数时域波形
xlabel('t');
ylabel('y');
title('门函数 y 时域波形');
grid;

y1=fft(y);
```

```
%fft
A=abs(y1);
%幅值
f=(0:length(y1)-1)'*Fs/length(y1);
%频率计算
subplot(232);
plot(f,A);
%做频分析
xlabel('f');
ylabel('y1 幅值');
title('对应 y1 幅频谱图');
grid;

Squ=abs(y1);
subplot(233);
plot(f,Squ);
xlabel('f(Hz)');
ylabel('均方根谱');
title('门函数均方根谱');
grid;

Pow=Squ.^2;
%y1 功率频谱函数
subplot(234);
plot(f,Pow);
xlabel('f');
ylabel('y1 功率频谱函数');
title('y1 功率谱');
grid;
```

```
Pow=Squ.^4;
subplot(235);
plot(f,Pow);
xlabel('f');
ylabel('y1 功率频谱函数');
title('y1 功率谱');
grid;

X_ifft=ifft(y1);
y2=real(X_ifft);
t2=[0:length(X_ifft)-1]/Fs;
subplot(236);
plot(t2,y2);
xlabel('t');
ylabel('y2');
title('基于 IFFT 得到的门函数时域波形');
grid;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%噪声信号处理
Fs=10;
%采样频率 Fs
t=-10:0.02:10;
y=zeros(1,800);
y(500)=200;
subplot(231);
plot(t(1:800),y);
%白噪声时域波形
xlabel('t');
```

```
ylabel('y');
title('白噪声时域波形');
grid;

y1=fft(y);
%进行 fft 变换,FFT 变换频谱 y1
A=abs(y1);
%y1 幅值
f=(0:length(y)-1)'*Fs/length(y1);
%频率计算
subplot(232);
plot(f,A);
%做频谱图
xlabel('f');
ylabel('y1 幅值');
title('对应 y1 幅频谱分布');
grid;

Squ=abs(y1);
%y1 均方根谱
subplot(233);
plot(f,Squ);
xlabel('f(Hz)');
ylabel('abs(y1)均方根谱');
title('y1 均方根谱');
grid;

Pow=Squ.^2;
%y1 功率谱
```

```
subplot(234);
plot(f,Pow);
xlabel('f(Hz)');
ylabel('y1 功率谱');
title('y1 功率谱');
grid;

X_ifft=ifft(y1);
y2=real(X_ifft);
t2=[0:length(X_ifft)-1]/Fs;
subplot(235);
plot(t2,y2);
xlabel('t');
ylabel('y2 信号时域波形');
title('基于 IFFT 计算得到');
grid;

X_ifft=ifft(y1);
%IFFT 计算 y1 时域波形
y3=real(X_ifft);
t3=[0:length(X_ifft)-1]/Fs;
subplot(236);
stem(t3,y3);
xlabel('t');
ylabel('y3 信号时域波形');
title('基于 IFFT 计算得到');
grid;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M=20;
```

```
N=50;
%基于分段计算的快速卷积
t1_x=1:0.4:M;
t1_h=1:0.2:N;
x1=cos(0.75*t1_x);
h1=0.45.^t1_h;
T=pow2(nextpow2(M+N-1));
Xn_fft=fft(x1,T);
Hn_fft=fft(h1,T);
Yn_fft=Xn_fft.*Hn_fft;
yn=ifft(Yn_fft,T);
t1_y=1:T
subplot(2,2,1);
stem(t1_x,x1,'o');
title('x');
subplot(2,2,2);
stem(t1_h,h1,'*');
title('h');
subplot(2,2,3);
stem(t1_y,real(yn),'>');
title('y');
T2=pow2(nextpow2(M+N+N-1));
Yn2_fft=Xn_fft.*Hn_fft.*Hn_fft;
yn_2=ifft(Yn2_fft,T2);
t1_y2=1:T2;
subplot(2,2,4);
stem(t1_y2,real(yn_2),'>');
title('y');
```