**实验四 利用FFT 实现快速卷积**

**班级 信息173班 学号 201711010228**

**姓名 苏 栋 日期 2020.06.19**

**一、实验要求**

加深理解FFT 在实现数字滤波（或快速卷积）中的重要作用，更好的利用FFT 进行数 字信号处理。掌握循环卷积和线性卷积两者之间的关系。

**二、实验原理**

数字滤波器根据系统的单位脉冲响应h(n) 是有限长还是无限长可分为有限长单位脉冲响应（Finite Impulse Response）系统（简记为FIR系统）和无限长单位脉冲响应（Infinite Impulse Response）系统（简记为IIR系统）。

对于FIR滤波器来说，除了可以通过数字网络来实现外，也可以通过FFT的变换来实现。首先我们知道，一个信号序列x(n) 通过FIR滤波器时，其输出应该是x(n)与h(n)的卷积：



当h(n)是一个有限长序列，即h(n)是FIR滤波器，且时



在数字网络类的FIR滤波器中，普遍使用的横截型结构就是按这个卷积公式构成的。

应用FFT实现数字滤波器实际上就是用FFT来快速计算有限长度序列的线性卷积。

*X*(*k*)

*x*(*n*)

FFT

IFFT

*H*(*k*)

*H*(*k*)*X*(*k*)

*y*(*n*)

这种方法就是先将输入信号x(n) 通过FFT变换为它的频谱采样值X(k)，然后和FIR滤波器的频响采样值H(k)相乘，H(k)可事先存放在存储器中，最后将乘积H(k)X(k)通过快速傅里叶反变换（简称IFFT）还原为时域序列，即得到输出y(n)。

现以FFT求有限长序列间的卷积，和求有限长度序列与较长序列间的卷积为例来讨论FFT的快速卷积方法。

a．序列x(n)和h(n)的长差不多。设x(n)的长为N1，h(n)的长为N2，要求



用FFT完成这一卷积的具体步骤如下：

①为使两有限长序列的线性卷积可用其循环卷积代替而不发生混叠，必须选择循环卷积长度，若采用基2-FFT完成卷积运算，要求（m为整数）。

②用补零方法使x(n)和h(n)变成列长为N的序列。





③用FFT计算x(n)和h(n)的N点离散傅里叶变换

④完成X(k)和H(k)乘积，

⑤用FFT计算Y(k)的离散傅里叶反变换得



b．当x(n)长度很长时，即，通常不允许等x(n)全部采集齐后再进行卷积，否则使输出相对于输入有较长的延时，另外，若太大，h(n)要补上太多的零点，很不经济，且FFT的计算时间也要很长。为此，采用分段卷积的方法，即把x(n)分成长度与h(n)相仿的小段，分别求出每段卷积的结果，然后用相应的方式把它们结合起来，便是总的输出。分段卷积方法主要有两种，即重叠相加法和重叠保留法。具体内容请参考数字信号处理教材中“快速离散傅里叶变换”一章中的线性卷积的FFT算法部分，本实验这部分不作重点要求。

**三、主要实验仪器及材料**

微型计算机、Matlab6.5 教学版、TC 编程环境。

**四、实验内容**

（1）用Matlab 或C语言编制信号产生子程序，产生典型信号供谱分析用；

（2）对给出信号逐个进行谱分析，绘出序列和幅频特性曲线；

（3）设计利用快速傅里叶变换FFT 计算线性卷积的程序；

（4）对结果进行分析；

（5）完成实验报告。

**五、实验结果**

**（1）实验代码：**

>> x=[2,1,-1,2,3];

>> nx= 0:4;

>> K= 128;

>> dw=2\* pi/K;

>> k= floor(( - K/2 +0.5):(K/2- 0.5));

>> X=x\*exp(- j\*dw\* nx' \*k);

>> subplot(221),plot(k \* dw, abs(X)),hold on;

>> xlabel('\omega');ylabel('幅度响应'),title('5点序列的DTFT和FFT');

>> Xd= fft([2,1, - 1,2,3]);

>> plot([0:4] \* 2\* pi/5,abs(Xd),'. '),grid;

>> Xd1 = fftshift(Xd);

>> subplot(222),plot(k \* dw, abs(X)), hold on;

>> xlabel( '\omega' );ylabel('幅度响应');

>> title('FFT移位后');

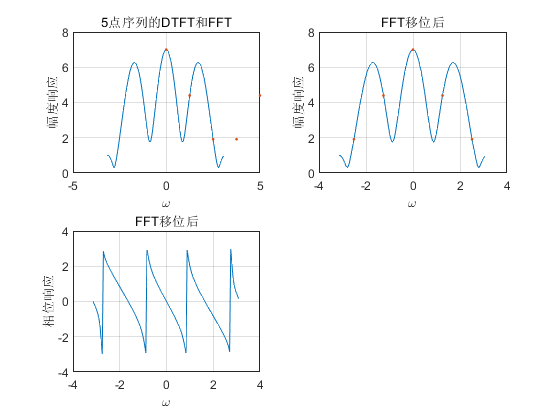
>> plot([ - 2:2] \* 2 \* pi/5, abs(Xd1),'. '),grid;

>> subplot(223),plot(k \* dw, angle(X)), hold on;

>> xlabel('\omega' );ylabel( '相位响应');

>> title('FFT移位后');grid;

**实验结果：**

****

**（2）实验代码：**

>> x=[2,1,3,2,1,5,1];

>> h=[1,2,-1,-3];

>> N= length(x) + length(h)- 1;

>> n= 0:N-1;

>> x= [x, zeros(1,N- length(x))];

>> h= [h, zeros(1,N- length(h))];

>> X= fft(x);H= fft(h);

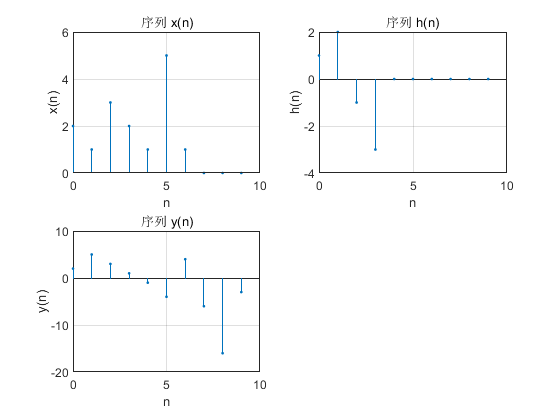
>> Y=X .\*H;y= ifft(Y);

>> subplot(221),stem(n,x,'.');xlabel('n');ylabel( 'x(n)');title('序列 x(n)');grid;

>> subplot(222),stem(n,h,'.');xlabel('n');ylabel( 'h(n)');title('序列 h(n)');grid;

>> subplot(223),stem(n,y,'.');xlabel('n');ylabel( 'y(n)');title('序列 y(n)');grid;

**实验结果：**



**（3）实验代码：**

>> x= [2,1,3,2,1,5,1];

>> N= 2\* length(x)- 1;

>> n=-length(x) + 1:length(x)- 1;

>> r= xcorr(x,x);

>> X= fft(x, N);

>> R= (abs(X)).^2;r1 = ifft(R);

>> subplot(221),stem(x,'.');axis([1,N,-inf,inf]);title('序列x(n)');grid;

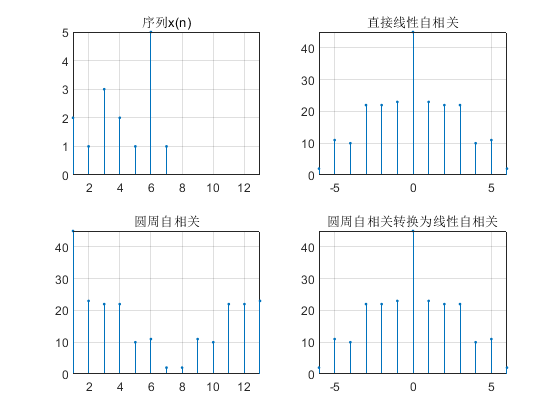
>> subplot(222),stem(n,r,'.');axis([min(n),max(n), - inf, inf]);title('直接线性自相关');grid;

>> subplot(223),stem(r1,'.');axis([1,N,-inf,inf]);title('圆周自相关');grid;

>> subplot(224),stem(n,fftshift(r1),'.');axis([min(n), max(n), - inf, inf]);

>> title('圆周自相关转换为线性自相关');grid;

**实验结果：**

**（4）实验代码：**

>> n=0:31;x1=[1 2 3 4 5 5 4 3 2 1];

>> x2=[1 2 4 8 16 32 -1 -2 -4 -8 -16 -32];

>> y1 = fft(x1,32);y2= fft(x2,32);yy=y1 .\* y2;y= ifft(yy,32);

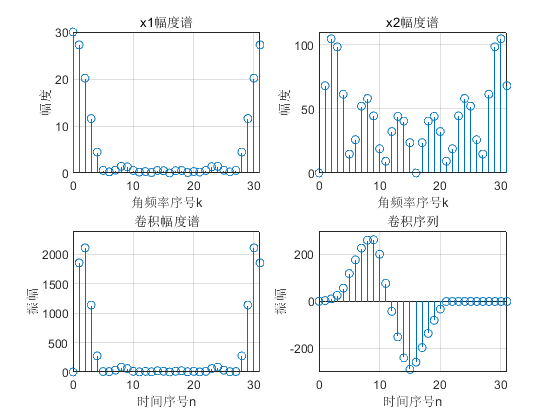
>> subplot(2,2,1) ;stem(n, abs(y1));title('x1幅度谱');grid;axis([0,31,0,30]);xlabel('角频率序号k');ylabel('幅度');

>> subplot(2, 2, 2);stem(n, abs(y2));title('x2幅度谱');grid;axis([0, 31,0,110]);xlabel('角频率序号k');ylabel('幅度');

>> subplot(2,2, 3); stem(n, abs(yy));title('卷积幅度谱'); grid; axis([0,31,0,2400]);xlabel('时间序号n');ylabel('振幅');

>> subplot(2,2, 4); stem(n, y);title('卷积序列'); grid; axis([0,31,-300,300]);xlabel('时间序号n');ylabel('振幅');

**实验结果：**

****

**（5）实验代码：**

**代码一：**

function y= circonvt(x1, x2,N)

if length(x1)> N

error('N必须>=x1的长度')

end

if length(x2)> N

error('N必须>=x2的长度')

end

x1=[x1 zeros(1, N-length(x1))];

x2= [x2 zeros(1,N- length(x2))];

m=[0:1:N- 1];

x2=x2(mod(-m,N) + 1);

H= zeros(N,N);

for n= 1:1:N

H(n,:) = cirshift(x2,n- 1,N);

end

y=x1\*H'

**代码二：**

function y=cirshift(x,m,N)

if length(x)>N

error('N必须 >= x的长度');

end

x=[x,zeros(1,N-length(x))];

n=0:N-1;

n=mod(n-m,N);

y=x(n+1);

**实验结果：**

>> x1=[2 4 3 1];x2=[2 1 3];

>> y=circonvt(x1,x2,4);

y =

14 13 16 17

1. **实验总结**

通过此次实验，我学会了FFT 在实现数字滤波中的作用；懂得了循环卷积和线性卷积两者之间的关系，明白了FFT子程序的应用，加强了MATLAB中代码编程画图；发现了在编程中，MATLAB中存在大量常用函数，可直接使用；在今后学习中应该加强代码实现常用函数能力。