

天津工程师范学院

电子工程系实验实训中心

《数字信号处理》实验教学大纲

1.课程基本信息

中文课程名称	数字信号处理				
英文课程名	Digital signal processing				
课程编号	课程类型	开设专业	课程总学时	实验学时	备注
090012504	专业基础课	电信、微电、	64	10	
090012402	专业基础课	通信、电科	32	6	

2.课程简介

当今人类正进入信息时代，而数字化是信息时代发展的方向。因此，数字信号处理技术已成为每一个电子信息科学工作者必须掌握的重要知识。数字信号处理课程已成为通信、电信、应教、微电等专业的必修专业课程。它采用计算机和专用处理设备，以数值计算的方法对信号进行分析、变换、滤波、检测、估计与识别等加工处理，以达到提取信息便于使用的目的。

3.实验目的

通过本实验课程的学习，使学生掌握利用 Matlab 软件进行数字信号处理的方法，从而验证理论知识，并能够解决实际问题。

4.实验报告内容

- (1) 学生姓名、学号、实验组号及组内编号等；
- (2) 实验题目：阐述做本实验的目的；
- (3) 目的要求：实验过程中的基础知识、预习报告、特殊器材等；
- (4) 实验设备：仪器名称及主要规格（包括量程、分度值等）、用具名称、软件等。
- (5) 实验原理：简单但要抓住要点，即要写出依据原理的公式名称、公式表达式、公式中各物理量的名称、公式成立的条件。画出电路图（光路图）等。
- (6) 实验内容：（对一些实验，根据要求要有实验步骤）

实验内容主要含实验数据表格、实验程序代码、数据处理、仿真波形等。

(7) 实验结果：程序运行结果、测试结果等；

(8) 实验总结：通过实验学到的知识、实验过程中的难点、存在的问题和实验收获等。

5.实验考试（考核）

(1) 学生实验课成绩由过程考核和实验报告两部分组成，各占 50%，平时成绩和报告成绩采用 5 分制、总成绩采用百分制。

(2) 实验过程：

由老师考核实验过程中学生实验线路连接是否正确；实验程序编写是否正确；仪器使用是否规范；能否按要求独立操作；遇到问题能否判断处理；测量数据是否正确；能否在规定时间内完成实验内容；实验结束时是否能按要求将仪器复位；课上是否能主动探索和研究问题和考勤等。

(3)实验报告：

是否符合实验报告要求的格式，叙述是否清楚；数据处理中的有效数字、单位、作图、思考题回答是否正确等。

6.实验项目设置与内容

序号	实验名称	内容要求	实验学时	每组人数	实验属性	开出要求
1	常见数字信号的 MATLAB 实现	掌握正弦序列 MATLAB 实现；随机信号的 MATLAB 实现；指数信号的 MATLAB 实现；单位阶跃序列。	2	1	验证	必做
2	离散序列的卷积和系统差分方程的 ATLAB 实现	掌握卷积定理计算及验证；卷积定律验证；求系统的单位冲激响应和阶跃响应；一个三阶低通滤波器差分方程描述。	2	1	验证	必做
3	Z 变换和系统频域特性的 MATLAB 实现	通过 Z 变换分析求解系统的冲激响应 $h(n)$ ；通过 z 变换分析离散系统的频率响应和零、极点分布 画出因果系统 $y(n) = 0.9y(n-1) + x(n)$ 的零、极点图和频率响应曲线，并求出脉冲响应 $h(n)$ 。定性画出系统 $H(z) = 1 - z^{-N}$ 的零、极点图和频率响应曲线。	2	1	验证	必做

4	离散傅里叶变换(DFT)的MATLAB实现	离散傅里叶变换及基本性质;	2	1	验证	必做
5	快速傅里叶变换及其应用	快速傅里叶变换; 用FFT分析信号频率成分	2	1	验证	必做
6	IIR滤波器的设计	基本概念; 求滤波器频率响应的函数; ButterWorth(巴特沃斯)等模拟和数字滤波器的设计	2	1	验证	必做
7	FIR滤波器设计	在MATLAB中产生窗函数的方法; 基本窗函数的FIR滤波器的设计——标准频率响应形状。	2	1	验证	必做

7. 教材及实验指导书名称:

(1) 实验教材:

丁玉美, 数字信号处理 (内附实验指导书)。西安: 西安电子科技大学出版

(2) 教学参考书:

①程佩青, 数字信号处理教程。北京: 清华大学出版社, 2014

②刘舒帆, 费诺等, 数字信号处理实验。西安: 西安电子科技大学出版社, 2014

数字信号处理实验讲义

实验一 常见数字信号的 MATLAB 实现

1. 实验目的

- (1) 掌握 MATLAB 语言基础知识。
- (2) 掌握 MATLAB 软件操作基础。
- (3) 利用 MATLAB 工具产生数字信号基本序列。

2. 实验仪器及设备

- (1) 计算机一台。
- (2) MATLAB6.5 以上软件一套。

3. 实验任务及原理

实验任务 1: 正弦序列 MATLAB 实现

离散正弦序列的 MATLAB 表示与连续信号类似, 只不过是用 stem 函数, 而不是用 plot 函数来绘出序列的波形。编写程序段绘出正弦序列 $\sin(\frac{\pi}{12}k)$ 的波形。

```
%正弦信号的离散序列  
k=0: 39;  
m=sin(pi / 12*k);  
stem(k, fk);  
运行结果如图 1-1 所示。
```

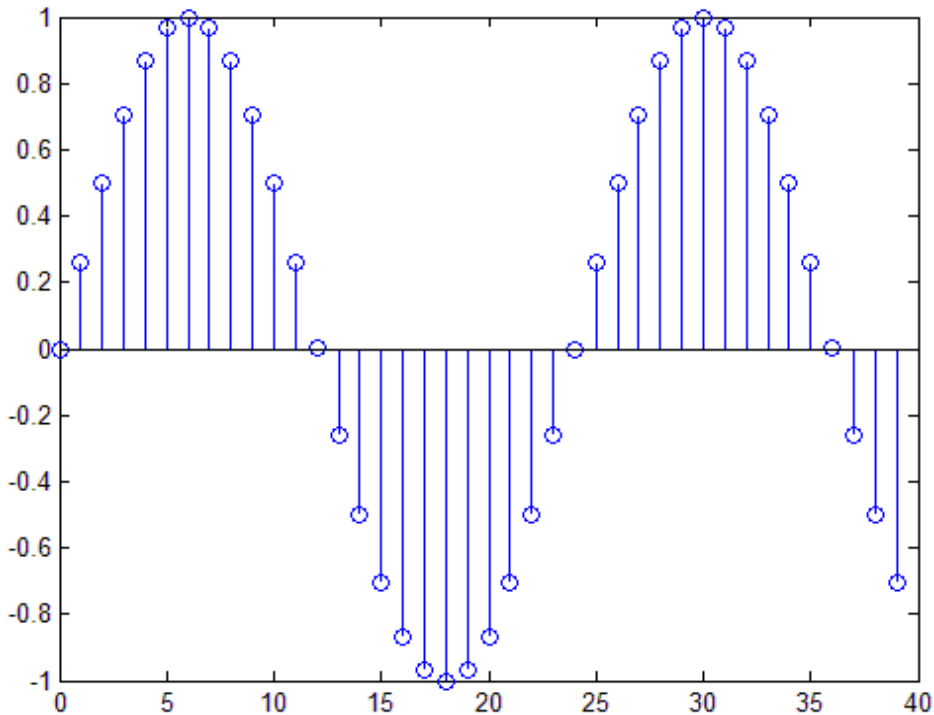


图 1-1 离散正弦信号 $\sin(\frac{\pi}{12}k)$ 的波形

实验任务 2: 随机信号的 MATLAB 实现

在区间(0, 1)中均匀分布的长度为 N 的随机信号, 可通过如下的 MATLAB 命令生成:

```
x=rand(1, N)
```

使用下面的 MATLAB 命令，可生成长度为 N 且具有零均值和单位方差的正态分布的随机信号 x[n]:

```
x=randn(1, N)
```

编程绘出一个随机信号的波形。

```
N=50;
```

```
n=1: N;
```

```
x=rand(1, N);
```

```
stem(n, x);
```

```
title(' 随机序列' );
```

```
xlabel(' 时间序列 n' );
```

```
ylabel(' 振幅' );
```

运行结果如图 1-2 之所示。

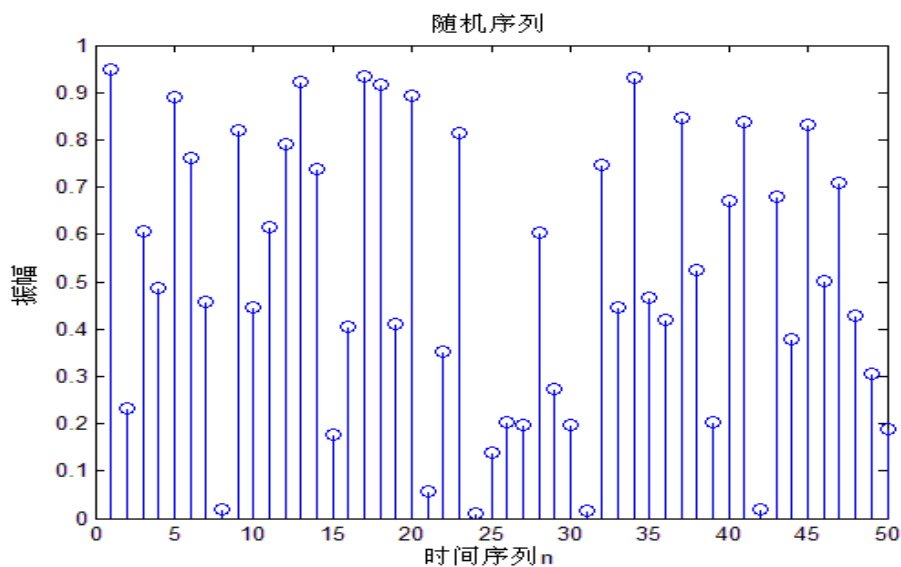


图 1-2 随机序列

实验任务 3: 指数信号的 MATLAB 实现

指数信号也是一种基本的离散时间序列，生成指数信号要用到 MATLAB 运算符 `.`和 `exp`。

①绘出实指数序列 $x(n)=0.2 \bullet 2^n$ 的波形。

```
%生成一个实指数序列
```

```
n=1: 10;
```

```
a=2;
```

```
K=0.2;
```

```
x=K*a.^n;
```

```
stem(n, x);
```

```
xlabel(' 时间序列 n' );
```

```
ylabel(' 振幅' );
```

运行结果如图 1-3 所示。

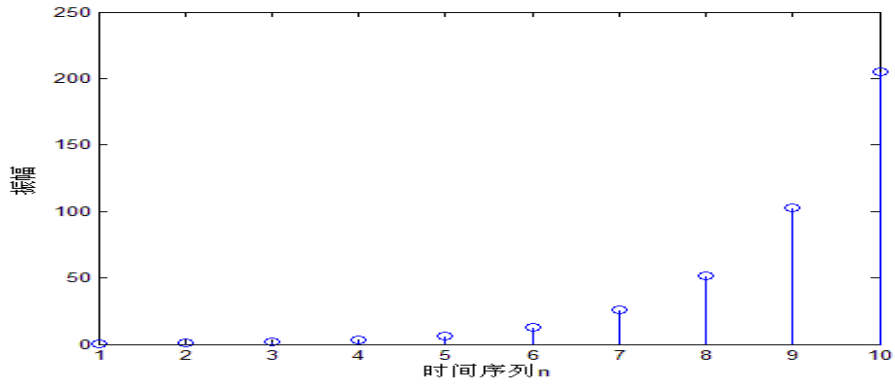


图 1-3 实指数序列

②绘出复指数序列 $x(n)=2e^{(-\frac{1}{6}+\frac{\pi}{6}i)n}$ 的波形。

%生成一个复指数序列

```
c = -1 / 6 + (pi / 6)*i;
```

```
K = 2;
```

```
n = 0: 20;
```

```
x = K * exp(c * n);
```

```
subplot(2, 1, 1);
```

```
stem(n, real(x));
```

```
xlabel('时间序列 n');
```

```
ylabel('振幅');
```

```
title('实部');
```

```
subplot(2, 1, 2);
```

```
stem(n, imag(x));
```

```
xlabel('时间序列 n');
```

```
ylabel('振幅');
```

```
title('虚部');
```

运行结果如图 1-4 所示。

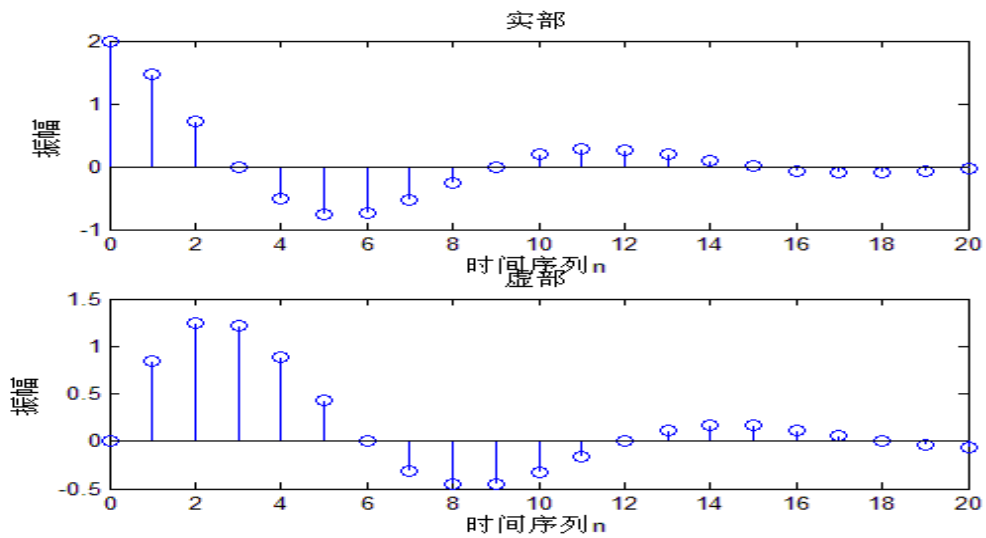


图 1-4 复指数序列

实验任务 4: 单位冲激序列

单位冲激序列 $\delta(n)$ 可以通过 MATLAB 中的零矩阵函数 `zeros` 得到。

绘出单位冲激序列 $\delta(n)$ 的波形。

%得到单位冲激序列

```
k=1: 30;
```

```
delta=[1, zeros(1, length(k)-1)];
```

```
stem(k, delta);
```

```
title(' 单位冲激序列' );
```

运行结果如图 1-5 所示。

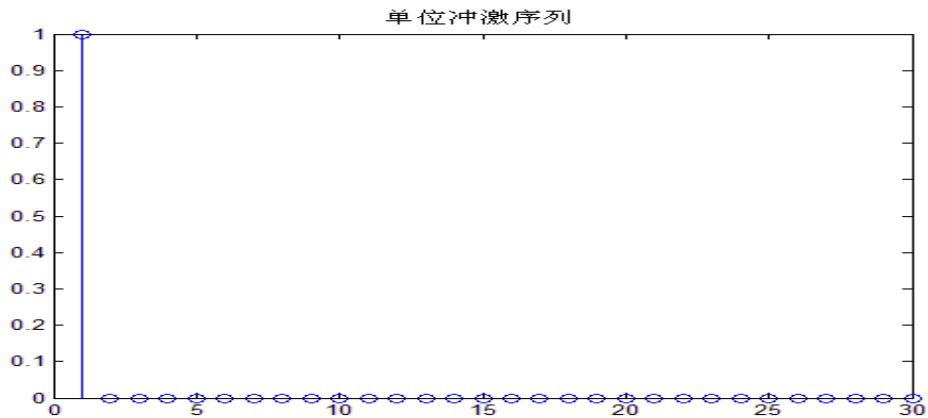


图 1-5 单位冲激序列

实验任务 5: 单位阶跃序列

可以通过 MATLAB 中的单位矩阵函数 `ones`, 来生成一个只含有元素 1 的行向量, 对有限区间的单位阶跃序列 $\mu(n)$ 可以通过下面的程序来实现。绘出单位阶跃序列 $\mu(n)$ 的波形。

%生成单位阶跃序列

```
step1=20;
```

```
step2=20;
```

```
k= -step1: step2;
```

```
uk=[zeros(1, step1), ones(1, step2+1)];
```

```
stem (k, uk);
```

```
title (' 单位阶跃序列');
```

运行结果如图 1-6 所示。

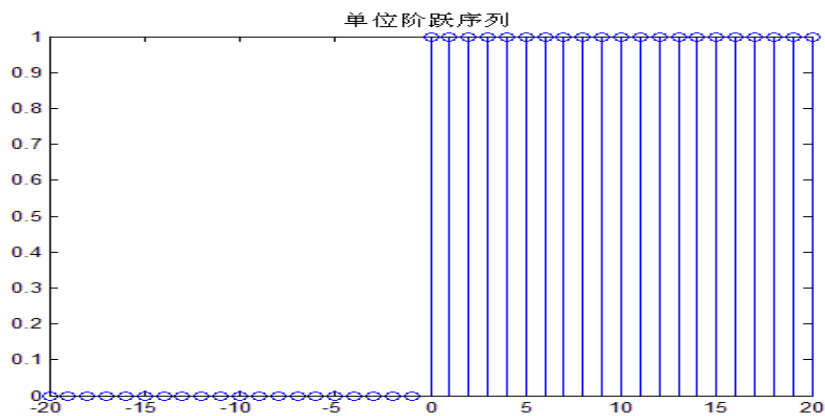


图 1-6 单位阶跃序列

实验二 离散序列的卷积和系统差分方程的 MATLAB 实现

1. 实验目的

- (1) 熟悉 MATLAB 对卷积和系统差分方程的程序设计。
- (2) 掌握卷积方法和原理。
- (3) 掌握离散系统差分方程的 MATLAB 实现。

2. 实验仪器及设备

- (1) 计算机一台。
- (2) MATLAB6.5 以上软件一套。

3. 实验内容

(1) 卷积定理计算及验证

实验任务 1: 编写程序实现卷积的计算

在 MATLAB 中, 提供了卷积函数 `conv`, 即 `y=conv(x,h)`, 调用十分方便。

```
n=1:50; % 定义序列的长度是 50
hb=zeros(1,50); % 注意: MATLAB 中数组下标从 1 开始
hb(1)=1; hb(2)=2.5; hb(3)=2.5; hb(4)=1;
close all;
subplot(3,1,1);stem(hb);title('系统 hb[n]');
m=1:50; % 定义序列的长度
T=0.001; % 定义序列的采样率
A=444.128; % 设置信号有关的参数
a=50*sqrt(2.0)*pi;
w0=50*sqrt(2.0)*pi;
x=A*exp(-a*m*T).*sin(w0*m*T); %pi 是 MATLAB 定义的  $\pi$ , 信号乘可采用 “.*”
subplot(3,1,2);stem(x);title('输入信号 x[n]');
y=conv(x,hb);
subplot(3,1,3);stem(y);title('输出信号 y[n]');
运行结果如图 2-1 所示。
```

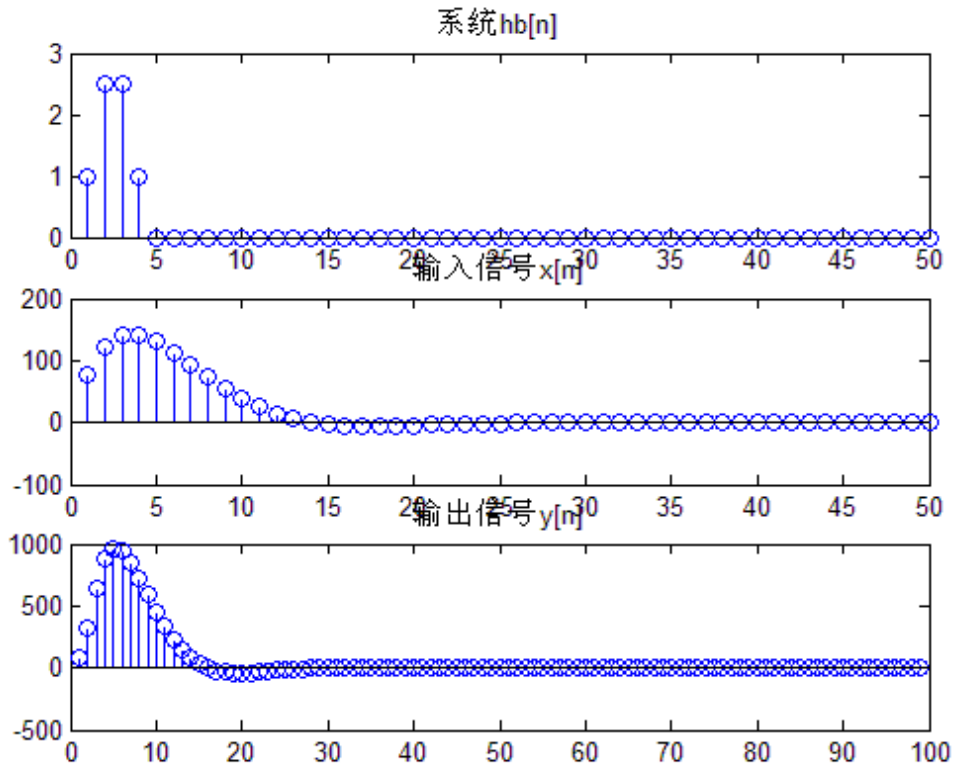



图 2-1 卷积计算

实验任务 2: 卷积定律验证

编写程序验证卷积定理

```

n=1:50; % 定义序列的长度是 50
hb=zeros(1,50); % 注意: MATLAB 中数组下标从 1 开始
hb(1)=1; hb(2)=2.5; hb(3)=2.5; hb(4)=1;
m=1:50; % 定义序列的长度
T=0.001; % 定义序列的采样率
A=444.128; % 设置信号有关的参数
a=50*sqrt(2.0)*pi;
w0=50*sqrt(2.0)*pi;
x=A*exp(-a*m*T).*sin(w0*m*T); % pi 是 MATLAB 定义的 π, 信号乘可采用 “.*”
y=conv(x,hb);
k=-25:25;
X=x*(exp(-j*pi/12.5)).^(n*k);

```

```

magX=abs(X); % 绘制 x(n) 的幅度谱
subplot(3,2,1);stem(magX);title('输入信号的幅度谱');

angX=angle(X); % 绘制 x(n) 的相位谱
subplot(3,2,2);stem(angX);title('输入信号的相位谱');

Hb=hb*(exp(-j*pi/12.5)).^(n*k);
magHb=abs(Hb); % 绘制 hb(n) 的幅度谱
subplot(3,2,3);stem(magHb);title('系统响应的幅度谱');

angHb=angle(Hb); % 绘制 hb(n) 的相位谱
subplot(3,2,4);stem(angHb);title('系统响应的相位谱');

n=1:99; k=1:99;

Y=y*(exp(-j*pi/12.5)).^(n*k);
magY=abs(Y); % 绘制 y(n) 的幅度谱
subplot(3,2,5);stem(magY);title('输出信号的幅度谱');

angY=angle(Y); % 绘制 y(n) 的相位谱
subplot(3,2,6);stem(angY);title('输出信号的相位谱');

```

运行结果如图 2-2 所示。

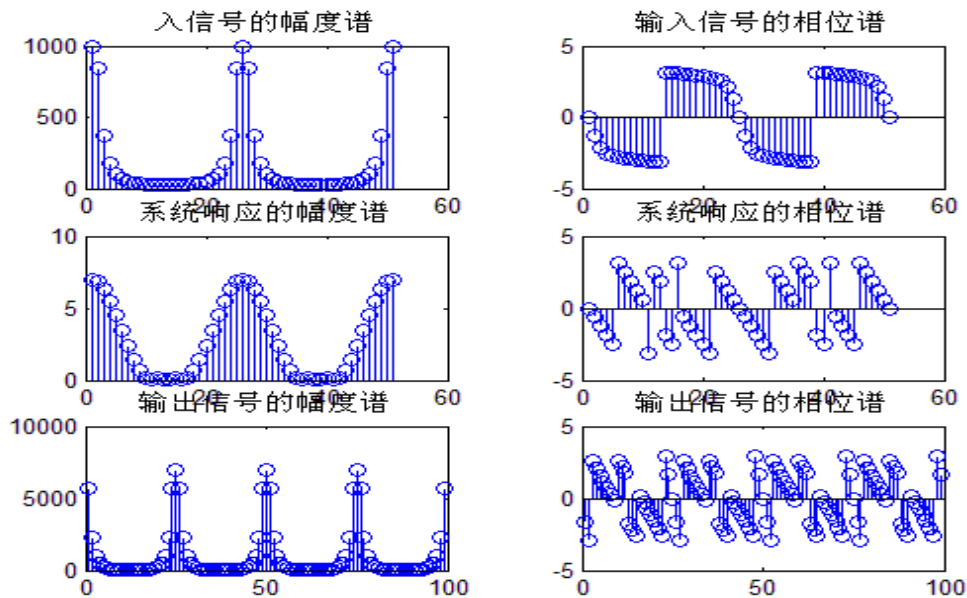


图 2-2 卷积定理输出图

(2) 离散系统差分方程的 MATLAB 实现

对于时域离散系统，可用差分方程描述或研究输入、输出之间的关系。对于线性时不变系统，经常用的是线性常系数差分方程。一个 N 阶线性常系数差分方程用下式表示：

$$\sum_{i=0}^N b_i y(n-i) = \sum_{i=0}^M a_i x(n-i)$$

当 $b_i=0, i=1, 2, \dots, N$ 时， $h[n]$ 是有限长度的，称系统为 FIR 系统，反之称系统为 IIR 系统。在 MATLAB 中，可以用函数 `filter(a, b, x)` 求解差分方程，其中参数 a, b 分别表示系统函数的分子和分母多项式的系数。

实验任务 3： 求系统的单位冲激响应和阶跃响应。

给出系统： $y(n) + 0.75y(n-1) + 0.125y(n-2) = x(n) - x(n-1)$ 的单位冲激响应和阶跃响应。

```
%差分方程求解
N=21;
a=[1,-1];
b=[1,0.75,0.125];
x1=[1 zeros(1,N-1)];
x2=ones(1,N);
k=0:1:N-1;
h=filter(a,b,x1);
y=filter(a,b,x2);
subplot(1,2,1);stem(k,h,'.');xlabel('n');ylabel('单位冲激响应');grid on;
subplot(1,2,2);stem(k,y,'.');xlabel('n');ylabel('单位阶跃响应');grid on;
运行程序将产生如图 2-3 所示序列。
```

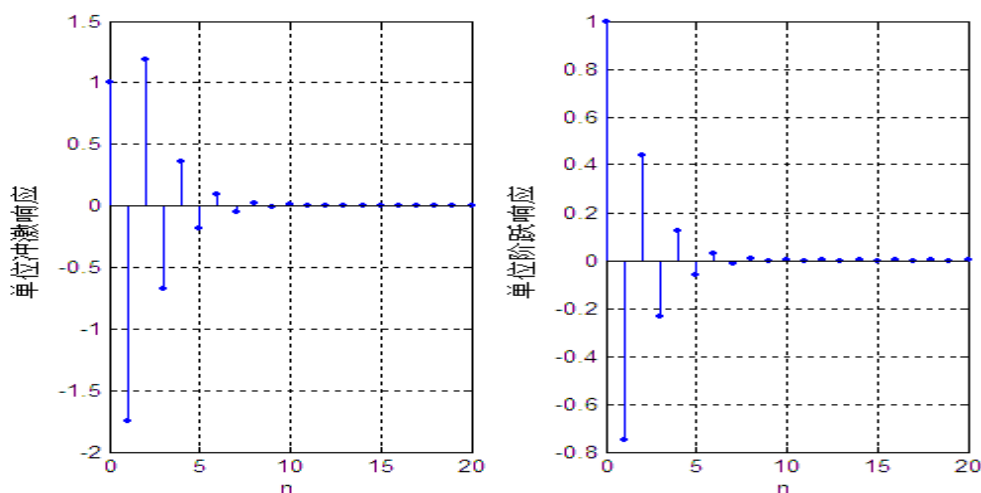


图 2-3 差分方程求解

实验任务 4： 一个三阶低通滤波器由下面差分方程描述：

$$y(n) = 0.0181x(n) + 0.0543x(n-1) + 0.0181x(n-3) + 1.76y(n-1) - 1.1829y(n-2) + 0.2781y(n-3)$$

画出这个滤波器的幅度和相位响应。

%系统频响

```
a=[0.0181,0.0543,0.0543,0.0181];
```

```
b=[1.0000,-1.7600,1.1829,-0.2781];
```

```
m=0:length(a)-1;l=0:length(b)-1;
```

```
N=500;
```

```
k=0:1:N;
```

```
w=pi*k/N;
```

```
num=a*exp(-j*m'*w); %分子
```

```
den=b*exp(-j*l'*w); %分母
```

```
H=num./den; %频响
```

```
magH=abs(H);angH=angle(H);
```

```
subplot(1,2,1);plot(w/pi,magH);xlabel('frequency in units of pi');
```

```
ylabel('|H|');grid on;
```

```
subplot(1,2,2);plot(w/pi,angH);xlabel('frequency in units of pi');
```

```
ylabel('phase in pi radians');grid on;
```

将产生如图 2-4 所示序列

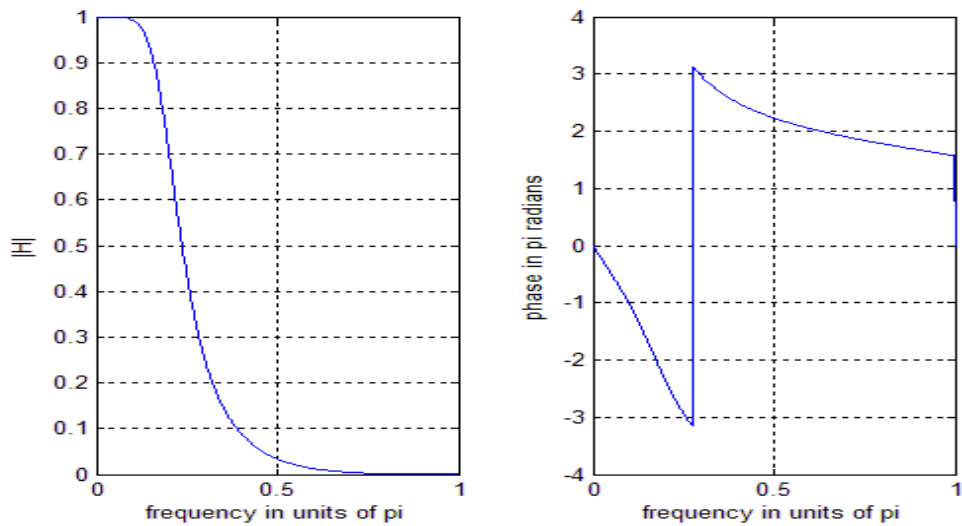


图 2-4 差分方程所表示的系统频响

实验3 Z变换和系统频域特性的 MATLAB 实现

1. 实验目的

- (1) 学习通过 MATLAB 实现 Z 变换来分析离散系统的频率响应。
- (2) 加深对离散系统的零、极点分布概念的理解。

2. 实验仪器及设备

- (1) 计算机一台。
- (2) MATLAB6.5 以上软件一套。

3. 实验任务及原理

(1) 通过 Z 变换分析求解系统的冲激响应 $h(n)$

已知用线性常系数差分方程: $y(n) = \sum_{k=1}^N a_k y(n-k) + \sum_{r=0}^M b_r x(n-r)$ 表示的线性时不变

系统其系统函数为:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{r=0}^M b_r z^{-r}}{1 - \sum_{k=1}^N a_k z^{-k}} \quad (3-1)$$

式(3-1)为两个关于 z^{-1} 的多项式之比, 即 $H(z)$ 为有理分式。

将式(3-1)表小数部分分式的形式

$$H(z) = \sum_{k=1}^N \frac{R_k}{1 - p_k z^{-1}} + \sum_{k=0}^{M-N} C_k z^{-k}$$

则可以根据所熟悉的常见序列的 z 变换形式求得 $H(z)$ 的 z 反变换, 从而得到系统冲激响应函数 $h(n)$ 。

MATLAB 提供了一个内部函数 `residuez()` 来计算有理多项式的留数和直接项,

`residuez()` 函数有几种调用方式: ① $[R, p, C] = \text{residuez}(b, a)$ 。在已知以分子行向量 b 和分母行向量 a 下, 得到列向量 R 含有留数, 列向量 P 是极点位置, 行向量 C 包含直接项; ② $[b, a] = \text{residuez}(R, p, C)$, 将部分分式展开式转换到分子行向量 b 和分母行向量 a 。

MATLAB 还提供了一个内部函数 `impz(b, a, N)`, 在已知分子行向量 b 和分母行向量 a 下, 计算 N 点的单位采样 $h(n)$ 。

实验任务 1: 求系统的单位采样响应

求系统

$$H(z) = \frac{0.001836 + 0.007344z^{-1} + 0.0011016z^{-2} + 0.007374z^{-3} + 0.001836z^{-4}}{1 - 3.0544z^{-1} + 3.8291z^{-2} - 2.2925z^{-3} + 0.55075z^{-4}}$$

的单位采样响应 $h(n)$ 。

解: 由上述基本原理和 MATLAB 提供的函数, 编写程序可得到所求系统的冲激响应 $h(n)$ 。

% $h(n)$ 求解

$b = [0.001836, 0.007344, 0.011016, 0.007374, 0.001836];$

```

a=[1,-3.0544,3.8291,-2.2925,0.55075];
[h,n]=impz(b,a,40);
stem(n,h,'.');ylabel('h[n]');grid

```

运行程序将产生如图 3-1 所示图形。

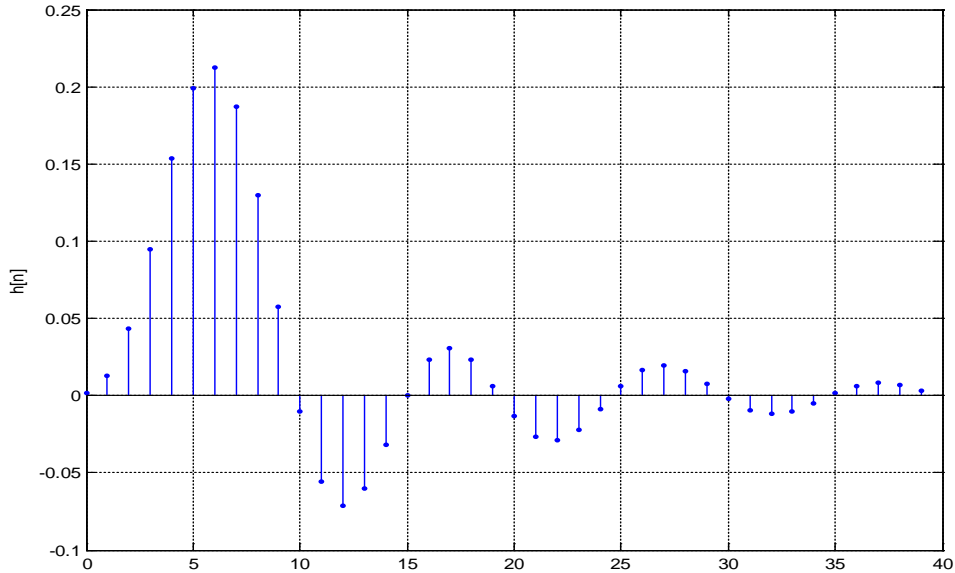


图 3-1 系统冲激响应

实验任务 2: 求 $X(z) = \frac{z}{3z^2 - 4z + 1}$ ($1 < |z| < \infty$) 的 z 反变换。

$$\text{解: } X(z) = \frac{z}{3z^2 - 4z + 1} = \frac{z^{-1}}{3 - 4z^{-1} + z^{-2}}$$

由上述基本原理和 MATLAB 提供的函数，编写程序来求 z 反变换。

% z 反变换求解

```
b=[0,1,0];a=[3,-4,1];
```

```
[R,p,C]=residuez(b,a)
```

```
[b,a]=residuez(R,p,C)
```

运行结果如下：

R =

0.5000

-0.5000

p =

1.0000

0.3333

C =

0

b =

-0.0000

0.3333

0

a =

1.0000 -1.3333 0.3333
>>

因此得到因式分解后的 $X(z) = \frac{z^{-1}}{3-4z^{-1}+z^{-2}} = \frac{1/2}{1-z^{-1}} - \frac{1/2}{1-1/3z^{-1}}$ ，所以 z 变换的结果为 $x(n) = 1/2u(n) - 1/2(\frac{1}{3})^n u(n)$ 。

果为 $x(n) = 1/2u(n) - 1/2(\frac{1}{3})^n u(n)$ 。

(2)通过 z 变换分析离散系统的频率响应和零、极点分布

将式(3-1)表示的 $H(z)$ 的分子、分母进行因式分解。可采用根的形式表示多项式，即

$$H(z) = \frac{A \prod_{r=1}^M (1 - c_r z^{-1})}{\prod_{k=1}^N (1 - d_k z^{-1})}$$

其中 c_r 为分子多项式的根，称为系统函数的零点， d_k 为分母多项式的根，称为系统函数的极点， A 为比例常数。这样系统函数就表示了零极点的位置刻画了系统很重要的特性，可以通过系统函数零极点位置估算系统函数的频率响应，进而判断系统的滤波特性，这是一种非常实用的方法，也称为频率响应的几何确定法。

根据频率响应的定义，将 $z = e^{j\omega}$ 代入上式，系统的频率响应为

$$H(e^{j\omega}) = H(z)|_{z=e^{j\omega}}$$

MATLAB 提供了一个内部函数 `zplane(b,a)`，在已知系统函数的分子行向量 d 和分母行向量 a 下，画出极点和零点。

MATLAB 还提供了一个内部函数 `freqz()` 来求系统的频率响应，`freqz()` 函数有几种调用方式：① `[H, w]=freqz(h, a, N)`，在已知以分子行向量 b 和分母行向量 a 下得到 N 点的频率向量 w 和 N 点的系统复频率响应向量 H ，频率响应在单位圆的上半圆的 N 个等分点上求值；② `[H, w]=freqz(b,a,N,'whole')`，在已知以分子行向量 b 和分母行向量 a 下得到 N 点的频率向量 w 和 N 点的系统复频率响应向量 H ，频率响应用环绕整个单位圆的 N 个等分点上求值；③ `H=frcqz(b,a,w)`，在已知以分子行向量 b 和分母行向量 a 下得到在频率向量 w 上的频率响应向量 H 。

实验任务 3: 画出因果系统 $y(n) = 0.9y(n-1) + x(n)$ 的零、极点图和频率响应曲线，并求出脉冲响应 $h(n)$ 。

解：由上述基本原理和 MATLAB 提供的函数，编写程序可得到所求系统的零、极点图和频率响应，

```
%y(n)=0.9y(n-1)+x(n)求解
b=[1,0];a=[1,-0.9];
subplot(2,2,1);zplane(b,a);
[H,w]=freqz(b,a,100);
magH=abs(H);phaH=angle(H);
subplot(2,2,3);plot(w/pi,magH);grid
xlabel('frequency in pi units');ylabel('magnitude');title('magnitude response');
subplot(2,2,4);plot(w/pi,phaH/pi)
```

```

xlabel('frequency in pi units');ylabel('phase');title('phase response');
[h,n]=impz(b,a,40);
subplot(2,2,2);stem(n,h,');ylabel('h[n]');grid

```

运行程序将产生如图 3-2 所示图形。

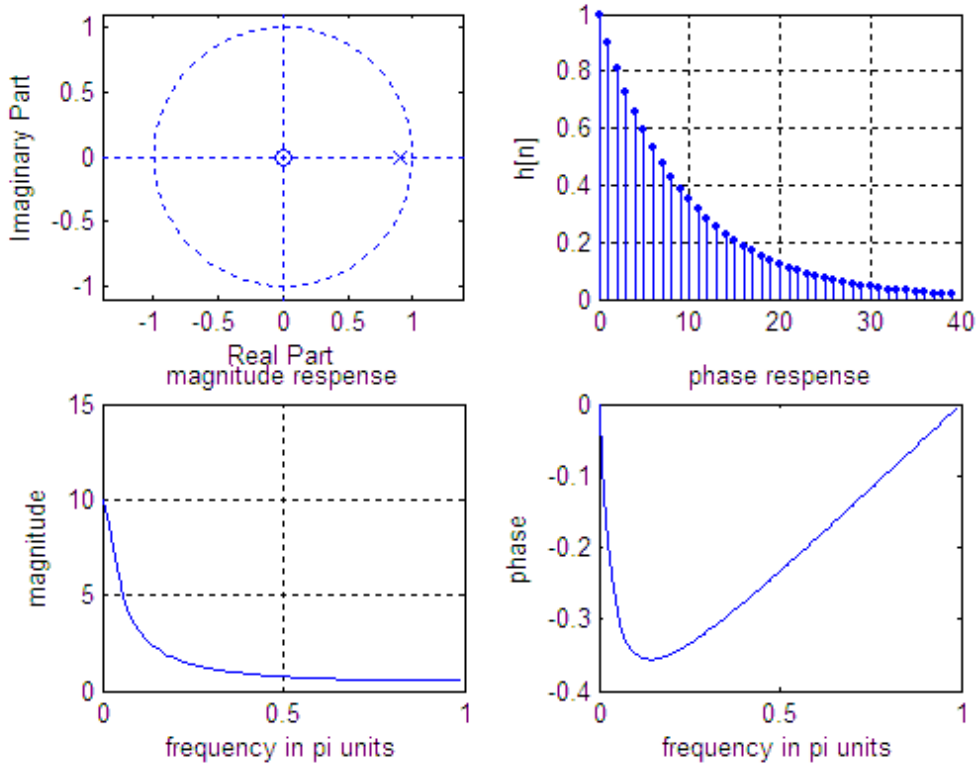


图 3-2 系统零、极点及频率响应曲线

实验任务 4: 定性画出系统 $H(z) = 1 - z^{-N}$ 的零、极点图和频率响应曲线。

解：由上述基本原理和 MATLAB 提供的函数，编写程序可得到所求系统的零、极点图和频率响应。

```

% H(z)=1-z-n
N=8;
b=[1,0,0,0,0,0,0,-1];a=[1,0,0,0,0,0,0,0];
subplot(1,3,1);
zplane(b,a);
title('极点-零点图');
%text(0.01,-0.1,'0');
%text(0.01,-0.1,'0');
[H,w]=freqz(b,a,100);
magH=abs(H);
phaH=angle(H);
subplot(1,3,2);plot(w/pi,magH);grid
xlabel('frequency in pi units');ylabel('magnitude');
title('magnitude response');

```



```

subplot(1,3,3);plot(w/pi,phaH/pi)
xlabel('frequency in pi units');ylabel('phase');
title('phase response');
% [h,n]=impz(b,a,40);
% subplot(2,2,2);stem(n,h,'. ');ylabel('h[n]');grid
运行程序将产生如图 3-3 所示的图形。

```

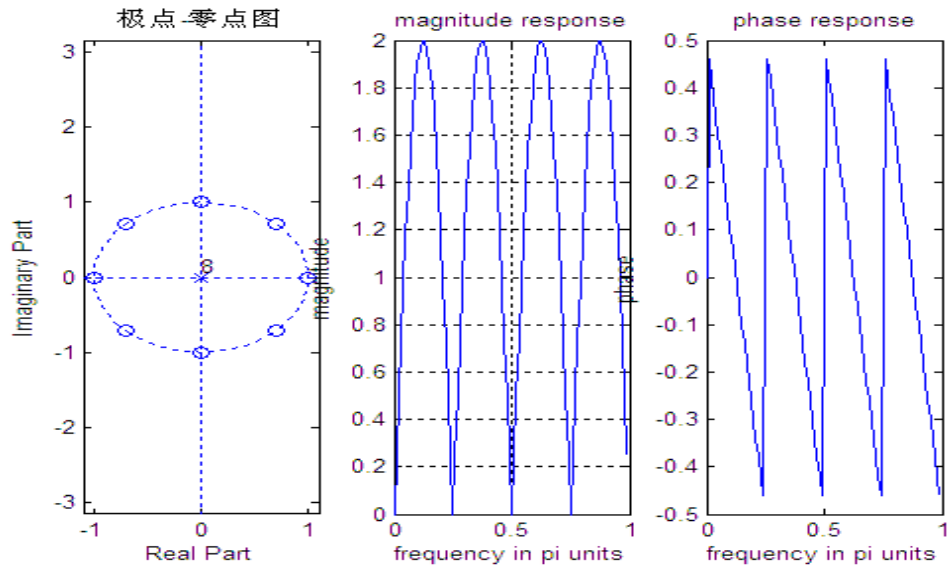


图 3-3 零，极点及频率响应图

实验四 离散傅里叶变换(DFT)的 MATLAB 实现

1. 实验目的

- (1) 熟悉 MATLAB 编程技术。
- (2) 掌握 MATLAB 实现离散傅里叶变换(DFT)的方法。

2. 实验仪器及设备

- (1) 计算机一台。
- (2) MATLAB6.5 以上软件一套。

3. 实验任务及原理

实验任务 1: 离散傅里叶变换及基本性质

(1) 离散傅里叶变换

傅里叶变换是建立在以时间为自变量的“信号”和以频率为自变量的“频率函数”之间的某种变换关系。离散傅里叶变换 (DFT) 就是傅里叶变换在时间域和频率域都离散时的情形。设 $x(n)$ 为长度为 N 的有限长序列, 则该序列的傅里叶变换及其逆变换分别为

$$X(k) = DFT(x(n)) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad k=0, 1, \dots, N-1,$$

$$x(n) = IDFT(X(k)) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)W_N^{-nk}, \quad k=0, 1, \dots, N-1,$$

其中 $W_N = e^{-j\frac{2\pi}{N}}$ 通过定义, 不难写出其 MATLAB 程序。

例: 已知序列 $x(n) = (1, 2, 4, 5, 6, 7, 8, 10)$, 求 $x(n)$ 的傅里叶变换 DFT 及其傅里叶逆变换 IDFT, 并画出原信号 $x(n)$ 与其傅里叶逆变换 $IDFT[X(k)]$ 的图形进行比较。

MATLAB 程序如下:

```
x=[1, 2, 4, 5, 6, 7, 8, 10];           %建立信号 x(n)
N=length(x);                           %信号长度
n=0: N-1; k=0: N-1;
X=x*exp(-j*2*pi / N).^ (n'*k);         %离散傅里叶变换
x1=X*exp(j*2*pi / N).^ (n'*k)/N;       %离散傅里叶逆变换
figure(1),stem(n, x);                   %显示原信号
title(' x(n) ');
figure(2), stem(n, abs(x1));             %显示逆变换结果
title(' IDFT| X(k) | ');
```

程序执行结果如图 4-1 所示。

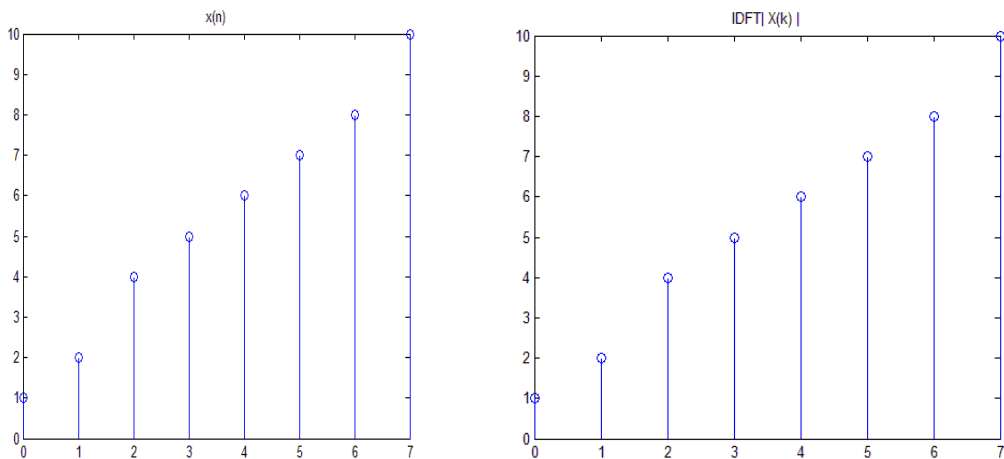


图 4-1 离散傅里叶变换

(2) 离散傅里叶变换的基本性质

DFT 有很多重要性质，在信号处理中有着广泛的应用。最常见的性质有：线性特性、序列的圆周移位性质、循环对称性质、循环折叠性质。下面将给出这些基本性质的 Matlab 程序。

① 线性特性

若两个有限长序列 $x_1(n)$ 和 $x_2(n)$ 的线性组合为

$$x_3(n) = a x_1(n) + b x_2(n),$$

则有

$$\text{DFT}(x_3(n)) = a \text{DFT}(x_1(n)) + b \text{DFT}(x_2(n)), \text{ 式中 } a, b \text{ 为任意常数。}$$

实验任务 2: 已知 $x_1(n) = (0, 1, 3, 4, 5)$, $x_2(n) = (1, 2, 1, 2, 1, 5)$ 。

(1) 求 $y(n) = 2x_1(n) + 3x_2(n)$ ，再由 $y(n)$ 的 N 点 DFT 获得 $Y(k)$ 。

(2) 由 $x_1(n)$, $x_2(n)$ 求 $X_1(k)$, $X_2(k)$ ，再求 $Y(k) = X_1(k) + X_2(k)$ 。

用图形分别表示以上结果，将两种方法求得的 $Y(k)$ 进行比较。

MATLAB 程序如下：

```

xn1 = [0, 1, 3, 4, 5]; % 建立 xn1 序列
xn2 = [1, 2, 1, 2, 1, 5]; % 建立 xn2 序列
N = max(length(xn1), length(xn2)); % 确定和序列的长度
if length(xn1) > length(xn2) % 对长度短的序列补 0，保证两个序列可加
    xn2 = [xn2, zeros(1, length(xn1) - length(xn2))];
else
    xn1 = [xn1, zeros(1, length(xn2) - length(xn1))];
yn = 2*xn1 + 3*xn2; % 计算 yn
n = 0: N-1; k = n;
Yk1 = yn * exp(-j*2*pi / N .* (n .* k)); % 求 yn 的傅里叶变换
    
```

```

Xk1=xn1*exp (-j*2*pi / N).^(n*k);           %求 xn1 的傅里叶变换
Xk2=xn2*exp (-j*2*pi / N).^(n*k);           %求 xn2 的傅里叶变换
Yk2=Xk1+Xk2;                                  %计算 Yk
subplot(3, 2, 1), stem(n, xn1)
title(' x1(n) ');
subplot(3, 2, 2), stem(n, Xk1);
title('X1(k)');
subplot(3, 2, 3), stem(n, xn2);
title('x2(n)');
subplot(3, 2, 4), stem(n, Xk1);
title('X2(k)');
subplot(3, 2, 5), stem(n, Yk1);
title(' DFT(2*x1(n)+3*x2(n)) ');
subplot(3, 2, 6), stem(n, Yk2);
title(' 2*X1(k)+3*X2(k) ');

```

程序执行结果如图 4-2 所示。

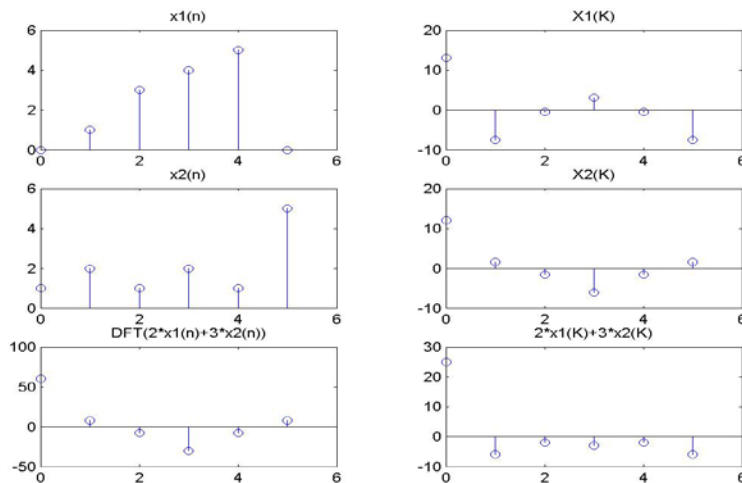


图 4-2 离散傅里叶变换的线性特性

②序列的圆周移位性质

有限长序列 $x(n)$ 的圆周移位是：以它的长度 N 为周期，将其延拓成周期序列 $x^*(n)$ ，并将周期序列进行移位，然后取主值区间上的序列值。设序列 $z(n)$ 右移 m 位的过程如下：

- I、将 $x(n)$ 以 N 为周期进行周期延拓，得到序列 $x^*(n)=x(\langle n \rangle_N)$ 。
- II、将 $x^*(n)$ 右移 m 位，得到 $x^*(n-m)$ 。
- III、取 $x^*(n-m)$ 的主值序列，得到 $x(n)$ 的循环移位序列 $y(n)$ 。

因此，就有 $y(n)=x(\langle n-m \rangle_N)G_N(n)$ ， $G_N(n)=\begin{cases} 1 & 0 \leq n \leq N-1 \\ 0 & \text{其他} \end{cases}$ ，有了圆周移位的概念后，

我们有下面两个重要的定理。

时域移位定理 若 $DFT(x(n))=X(k)$ ， $y(n)=x(\langle n-m \rangle_N)G_N(n)$ ，则

$$DFT(y(n))=W^{mk} X(k).$$

频域移位定理 若 $DFT(x(n))=X(k)$ ， $Y(k)=X(\langle k-m \rangle_N)G_N(k)$ ，则

$$\text{DFT}(Y(k))=x(n)W^{-nm}$$

实验任务 3: 已知有限长序列 $x(n)=(2, 3, 4, 5, 1, 7, 9)$, 设 $y(n)$ 为 $x(n)$ 右移 2 位得到的新序列。

(1) 根据圆周移位的定义, 求出 $y(n)$, 再由 $y(n)$ 的 N 点 DFT 获得 $Y_1(k)$ 。

(2) 先求出 $x(n)$ 的 N 点 DFT $X(k)$, 再由时域移位定理获得 $y(n)$ 的 N 点 DFT $Y_2(k)$ 。

用图形分别表示以上结果, 将 $Y_1(k)$ 和 $Y_2(k)$ 进行比较。

MATLAB 程序如下:

```

xn=[2, 3, 4, 5, 1, 7, 9];    %建立 x(n)序列
Nx=length(xn);
nx=1: Nx-1;
nxl=-Nx: 2*Nx-1;
xl=xn(mod(nxl, Nx)+1);      %建立周期延拓序列
nyl=nxl+2;
yl=xl;
yn=[];
for i=1: length(nyl)        %建立 Y(n)序列
    if(nyl(i)>=0)&(nyl(i)<Nx)
        yn=[yn  yl(i)];
    end
n=0: Nx-1; k=n;
    Yk1=yn*exp(-j*2*pi / Nx).^(n*k),    %求序列 Y(n)的傅里叶变换
    Xk=xn*exp(-j*2*pi / Nx).^(n*k);    %求序列 x(n)的傅里叶变换
    Yk2=Xk*exp(-j*2*pi / Nx).^(n*k);    %根据时域移位定理得到序
列 Y(n)的傅里叶变换
subplot(3, 2, 1), stem(n, xn);
title(' X(n) ');
subplot(3, 2, 3), stem(n, yn);
title(' y(n) ');
subplot(3, 2, 4), stem(k, Xk);
title(' X(k) ');
subplot(3, 2, 5), stem(k, Yk1),
title(' Y1(k) ');
subplot(3, 2, 6), stem(k, Yk2);
title(' Y2(k) ');

```

程序执行结果如图 4-3 所示。

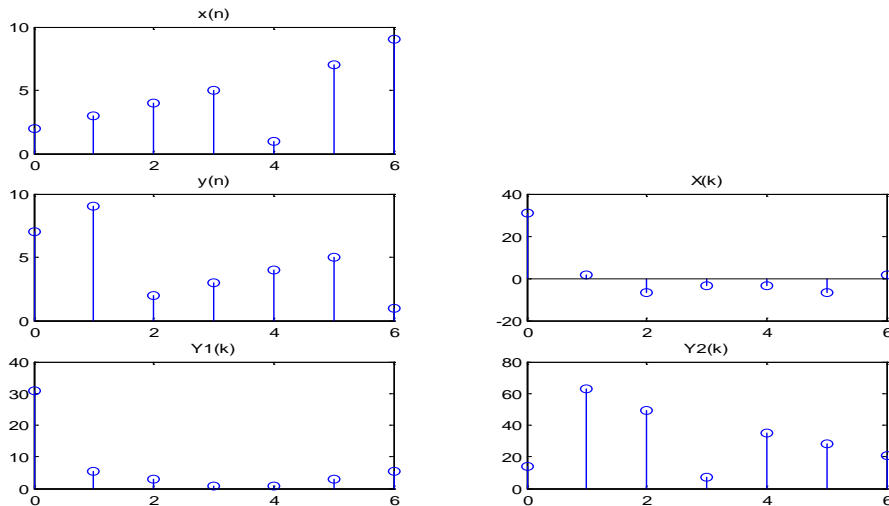


图 4-3 离散傅里叶变换的频域移位特性

③循环对称性质

设实序列 $x(n)$ 可以分解成循环偶序列 $x_1(n)$ 和循环奇序列 $x_2(n)$, 即

$$x(n) = x_1(n) + x_2(n), \quad 0 \leq n \leq N-1,$$

其中

$$x_1(n) = \frac{1}{2}(x(n) + x(-n)), \quad x_2(n) = \frac{1}{2}(x(n) - x(-n))$$

设 $\text{DFT}(x(n)) = X(k) = \text{Re}(x(k)) + j \cdot \text{Im}(x(k))$, 则有

$$\text{DFT}(x_1(n)) = \text{Re}(x(k)),$$

$$\text{DFT}(x_2(n)) = j \cdot \text{Im}(x(k)),$$

实验任务 4: 已知有限长序列 $x(n) = (1, 1, 2, 2, 3, 3, 4, 4)$ 。

(1) 把序列 $x(n)$ 分解成循环偶序列 $x_1(n)$ 和循环奇序列 $x_2(n)$ 。

(2) 利用序列 $x(n)$ 验证离散傅里叶变换的循环对称性。

MATLAB 程序如下(省略作图部分):

```
x=[1, 1, 2, 2, 3, 3, 4, 4]; %建立 x(n)序列
N=length(x);
n=0: N-1; k=n;
xr=x(mod(-n, N)+1); %建立 x(-n)序列
x1=(x+xr) / 2; %建立 x(n)的循环偶序列
x2=(x-xr) / 2; %建立 x(n)的循环奇序列
X=x*exp(-j*2*pi/N).^(n*k); %求 x(n)的傅里叶变换
X1=x1*exp(-j*2*pi/N).^(n*k); %求 x1(n)的傅里叶变换
```

```
X2=x2*exp(-j*2*pi/N).^(n*k); %求 x2(n)的傅里叶变换
```

程序执行结果如图 4-4 所示。

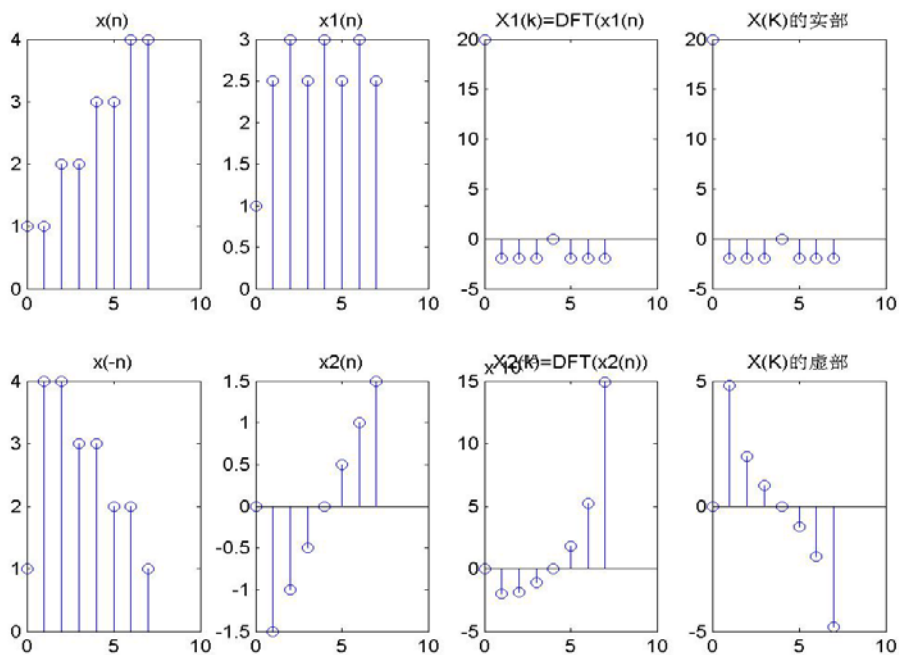


图 4-4 离散傅里叶变换的循环对称性

④循环折叠性质

有限长序列 $x(n)$ 的循环折叠序列 $y(n)$ 定义为

$$y(n) = x((-n)_N) = \begin{cases} x(0) & n=0 \\ x(N-m) & 1 \leq n \leq N-1 \end{cases}$$

循环折叠后的序列 $y(n)$ 的 DFT 由下式给出:

$$\begin{aligned} Y(k) &= \text{DFT}(x((-n)_N)) = X((-k)_N) = X^*(k)_N \\ &= \begin{cases} X(0) & k=0 \\ X(N-k) & 1 \leq k \leq N-1 \end{cases} \end{aligned}$$

也就是说, 对在时域上循环折叠的函数, 其对应的 DFT 在频域上也作循环折叠, 并取 $X(k)$ 的共轭。

实验任务 5: 已知有限长序列 $x(n)=(1, 3, 2, 4, 5, 6, 7)$, 列出循环程度分别为 $N=7$, $N=12$ 时 $x(n)$ 和 $x(-n)$ 的图形。

MATLAB 程序如下:

```

x1=[1, 3, 2, 4, 5, 6, 7];           %建立 N=7 时序列 x(n)
N1=length(x1); n1=1: N1;
y1=x1(mod(-n1, N1)+1);           %建立 N=7 时序列 x(-n)
x2=[x1 zeros(1, 12-7)];          %建立 N=12 时序列 x(n)
N2=length(x2); n2=1: N2;
y2=x2(mod(-n2,N2)+1);           %建立 N=12 时序列 x(-n)
subplot(2, 2, 1), stem(n1, x1);
title(' x(n), N=7' );
subplot(2, 2, 2), stem(n2, x2);
title(' X(n), N=12' );
axis([0 12 0 8])
subplot(2, 2, 3), stem(n1, y1);

```

```

title(' x(n), N=7' );
subplot(2, 2, 4), stem(n2, y2),
title(' x(-n), N=12' );
axis([0 12 0 8])

```

程序执行结果如图 4-5 所示。

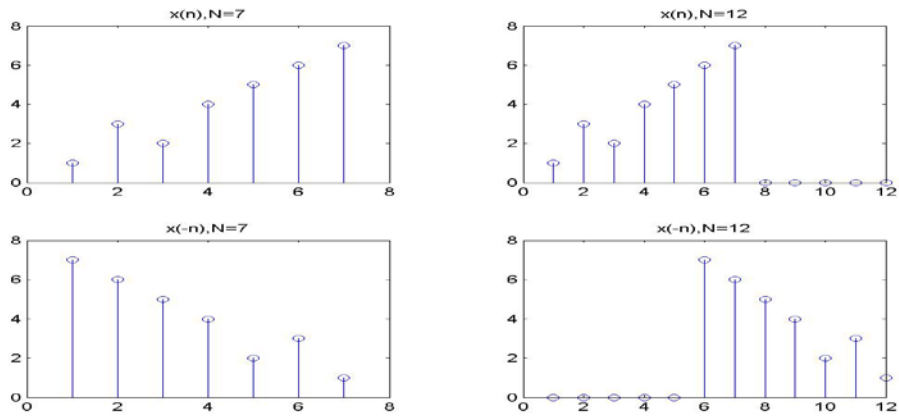


图 4-5 离散傅里叶变换的循环折叠性

实验五 快速傅里叶变换及其应用

1. 实验目的

- (1) 熟悉 MATLAB 编程技术。
- (2) 掌握 MATLAB 实现快速傅里叶变换(FFT)的方法。

2. 实验仪器及设备

- (1) 计算机一台
- (2) MATLAB6.5 以上软件一套

3. 实验任务及原理

(1) 快速傅里叶变换

DFT 是信号分析与处理中的一种重要变换。但直接计算 DFT 的运算量与变换的长度 N 的平方成正比, 当 N 较大时, 计算量太大。在快速傅里叶变换(简称 FFT)出现以前, 直接用 DFT 算法进行谱分析和信号的实时处理是不实际的。FFT 使得 DFT 的运算效率大大提高, 为数字信号处理技术应用于各种信号的实时处理创造了条件, 推动了数字信号处理技术的发展。

1) 基-2FFT 算法

长度为 N 的有限长序列 $x(n)$ 的 DFT 为

$$X(k) = \sum_{n=0}^{N-1} W_N^{kn} \quad k=0,1, \dots, N-1 \quad (5-1)$$

利用 $W_N^{kn} = W_N^{k(n+N)} = W_N^{(k+N)n}$ 和 $W_N^{kn+\frac{N}{2}} = -W_N^{kn}$ 时, 将 N 点 DFT 转化为 $N/2$ 点的 DFT。

当 $n = 2^r$ 时, 将 $x(n)$ 按奇偶分成 $N/2$ 的序列:

$$x_1(r) = x(2r) \quad 0 \leq r \leq \frac{N}{2} - 1$$

$$x_2(r) = x(2r+1) \quad 0 \leq r \leq \frac{N}{2} - 1$$

$$\begin{aligned} \text{则} \quad X(k) &= \sum_{n=0}^{N-1} x(n)W_N^{kn} = \sum_{r=0}^{\frac{N}{2}-1} x(2r)W_N^{k2r} + \sum_{r=0}^{\frac{N}{2}-1} x(2r+1)W_N^{k(2r+1)} \\ &= X_1(k) + W_N^k X_2(k) \quad k=0,1, \dots, N-1 \end{aligned}$$

式中

$$X_1(k) = \sum_{r=0}^{N/2-1} x_1(r)W_{N/2}^{kr} = \text{DFT}[x_1(r)]$$

$$X_2(k) = \sum_{r=0}^{N/2-1} x_2(r)W_{N/2}^{kr} = \text{DFT}[x_2(r)]$$

由于 $X_1(k)$ 和 $X_2(k)$ 都是 $N/2$ 点的 DFT, 且 $W_N^{kn+\frac{N}{2}} = -W_N^{kn}$, 所以 $X(k)$ 也可以表示为

$$X(k) = X_1(k) + W_N^k X_2(k) \quad k=0,1, \dots, N/2-1$$

$$X\left(k + \frac{N}{2}\right) = X_1(k) - W_N^k X_2(k) \quad k=0,1, \dots, N/2-1$$

这样就将 N 点 DFT 分解为两个 N / 2 点的 DFT 的运算。N / 2 点的 DFT 还可以再分解为 N / 4 点的 DFT,共可以分解 γ 级,最后达到 N / 2 个 2 点 DFT 运算。这种方法由于每一步都是按每级输入序列在时间上的次序是属于偶数还是奇数来分解为两个更短的子序列,所以称为“按时间抽取法”。例如 N=8 时,其信号流图见图 5-1 所示。

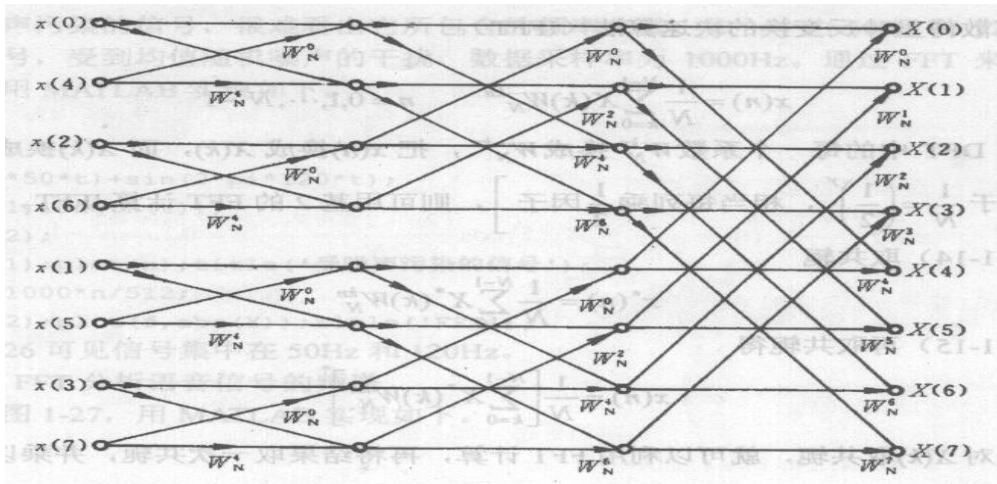


图 5-1 N=8 时按时间抽取的流程

2) FFT 算法的运算量

直接计算 DFT 共需

$$\text{复乘法数 } m = N^2$$

$$\text{复加法数 } a = N(N-1)$$

计算 FFT 共需

$$\text{复乘法数 } a = \frac{N}{2} \gamma = \frac{N}{2} \log_2 N$$

$$\text{复加法数 } a = N\gamma = N \log_2 N$$

由于计算机上乘法所需时间比加法所需时间多得多,故直接计算 DFT 与计算 FFT 的运算量之比为

$$\frac{N^2}{\frac{N}{2} \gamma} = \frac{2N}{\log_2 N}$$

具体数据见表 5-1,运算量比较如图 5-2 所示。

表 5-1 DFT 与 FFT 算法的比较

N	N^2	$\frac{N}{2} \log_2 N$	$N^2 / (\frac{N}{2} \log_2 N)$
2	4	1	4.0
4	16	4	4.0
8	64	12	5.4
16	256	32	8.0
32	1024	80	12.8
64	4096	192	21.4
128	16384	448	36.6
256	65536	1024	64.0
512	262144	2304	113.8
1024	1048576	5120	204.8
2048	4194304	11264	372.4

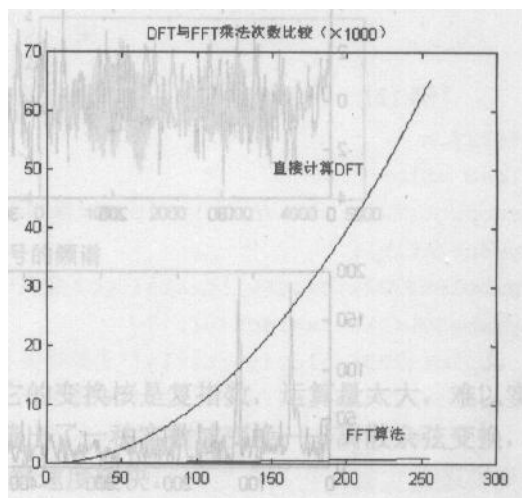


图 5-2 DFT 与 FFT 运算量比较

3) 离散傅里叶反变换的快速算法 (IFFT)

$$\text{由式} \quad x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn} \quad k=0,1,\dots,N-1 \quad (5-2)$$

只要把 DFT 中的每一个系数 W_N^{kn} 换成 W_N^{-kn} ，把 $x(n)$ 换成 $X(k)$ ，而 $X(k)$ 换成 $x(n)$ 每列

乘 $\frac{1}{2}$ 因子 [由于 $\frac{1}{N} = \left(\frac{1}{2}\right)^{\gamma}$ ，相当每列乘 $\frac{1}{2}$ 因子]，则可用基 2 的 FFT 计算 IFFT。

对式(5-2)取共轭

$$x^*(n) = \frac{1}{N} \sum_{k=0}^{N-1} X^*(k) W_N^{kn} \quad (5-3)$$

对式(5-3)再取共轭

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

只要先对 $X(k)$ 取共轭，就可以利用 FFT 计算，再将结果取一次共轭，并乘以 $\frac{1}{N}$ 。

4) FFT 算法的 MATLAB 实现

MATLAB 提供 fft 函数来计算 $x(n)$ 的 DFT。

格式： $y=\text{fft}(x)$ ，计算信号 x 的快速离散傅里叶变换 y 。当 x 为矩阵(多通道信号)时，计算 x 中每一列信号的离散傅里叶变换。当 x 的长度为 2 的幂时，用基-2 算法，否则采用较慢的分裂基算法。

$Y = \text{fft}(x,n)$ ，计算 n 点 FFT，当 x 的长度大于 n 时，截断 x ，否则补零。IFFT 可由 ifft 函数来计算。

(2) 快速傅里叶变换应用实例

实验任务 1: 用 FFT 分析信号频率成分

一被噪声污染的信号，很难看出它所包含的频率分量，如一个由 50Hz 和 120Hz 正弦信号构成的信号，受到均值随机噪声的干扰，数据采样率为 1000Hz。通过 FFT 来分析其信号频率成分。MATLAB 实现如下：

```
t=0:0.001:0.6;
x=sin(2*pi*50*t)+sin(2*pi*120*t);
y=x+randn(1,length(t));
Y=fft(y,512);
subplot(211);plot(x);title('受噪声污染的信号');
n=0:511;f=1000*n/512;
subplot(212);plot(f,abs(Y));title('FFT');
```

从图 5-3 可见信号集中在 50Hz 和 120Hz。

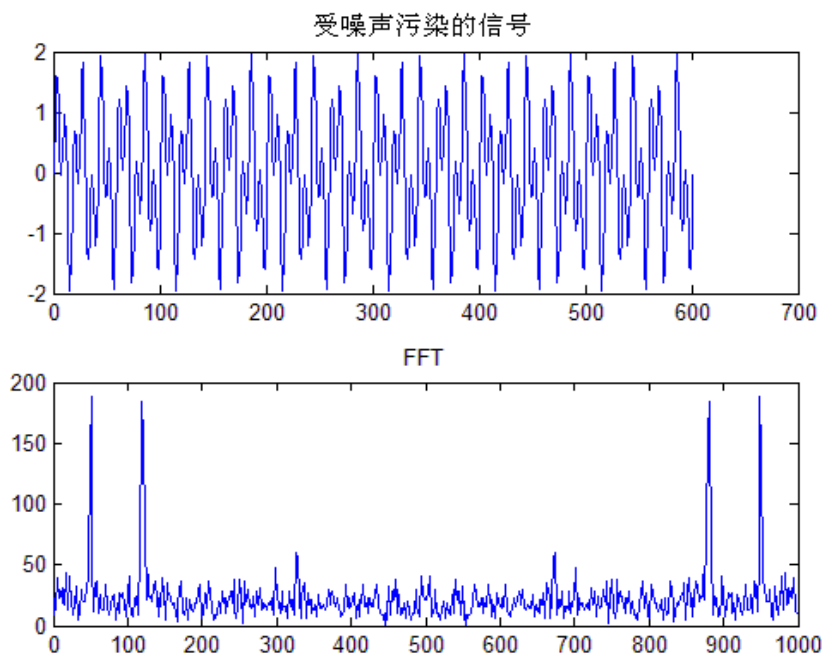


图 5-3 用 FFT 分析受噪声污染的信号

实验任务 2：用 FFT 分析语音信号的频谱

MATLAB 程序如下：

```
load mtlb;
subplot(221); plot(mtlb); title('原始语音信号');
y=fft(mtlb);
subplot(222); plot(abs(y)); title('FFT 变换');
y(abs(y)<1)=0; x=ifft(y);
subplot(223); plot(abs(y)); title('去掉复制小于 1 的 FFT 变换值');
subplot(224); plot(real(x)); title('重构语音信号');
运行结果见图 5-4 所示。
```

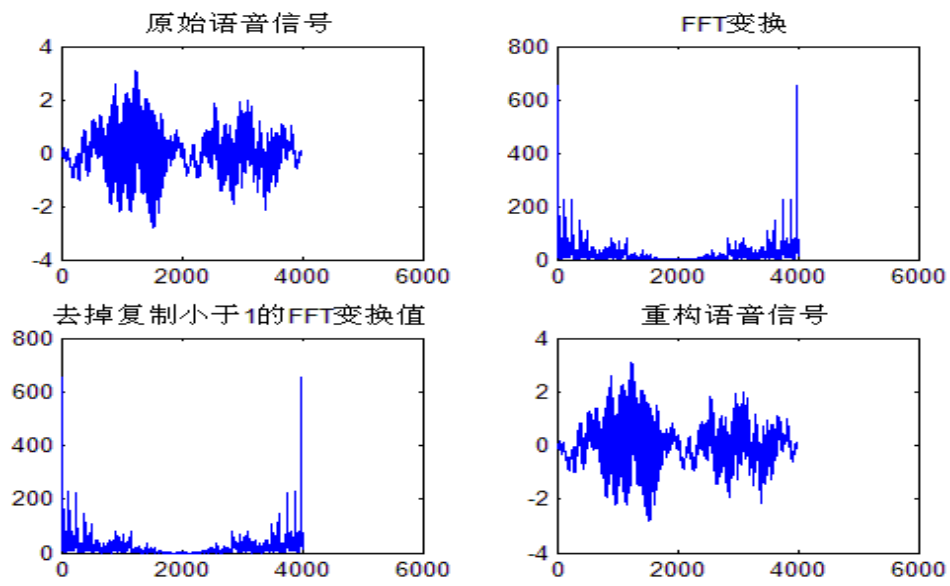


图 5-4 语音信号的频谱

实验六 IIR 滤波器的设计

1. 实验目的

- (1) 熟悉 MATLAB 编程技术。
- (2) 熟悉用双线性变换法设计 IIR 数字滤波器。

2. 实验仪器及设备

- (1) 计算机一台。
- (2) MATLAB6.5 以上软件一套。

3. 实验任务及原理

- (1) 基本概念

①**数字滤波器**是指输入、输出均为数字信号，通过一定运算关系，改变输入信号所含频率成分的相对比例或者滤除某些频率成分的器件。数字滤波器与模拟滤波器概念相同，只是信号的形式和实现滤波的方法不同。一般数字滤波器从现实的网络结构或者从单位脉冲响应分类，可以分成无限脉冲响应(IIR)滤波器和有限脉冲响应(FIR)滤波器。

②**IIR 滤波器设计的主要方法**是先设计低通模拟滤波器，进行频率变换，将其转换为相应的高通、带通模拟滤波器，再将模拟滤波器转换为相应的数字滤波器。

③**N 阶模拟滤波器系统函数的一般形式为**

$$H(s) = \frac{b_M s^M + b_{M-1} s^{M-1} + \dots + b_1 s + b_0}{a_N s^N + a_{N-1} s^{N-1} + \dots + a_1 s + a_0} = \frac{B(s)}{A(s)}$$

④**N 阶数字模拟滤波器系统函数的一般形式为**

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_{M-1} z^{-(M-1)} + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{N-1} z^{-(N-1)} + a_N z^{-N}} = \frac{B(z)}{A(z)}$$

(2) 求滤波器频率响应的函数

①**freqs 函数**：求模拟滤波器的频率响应

freqs(b,a,w)计算由向量 w (rad/s) 指定的频率点上（或 W 直接为采样点数）频率响应，其中 b 和 a 分别为模拟滤波器系统函数 H(s)的分子与分母。freqs 函数将自动绘出幅频和相频曲线。

实验任务 1：设系统传输函数为 $H(s) = \frac{0.2s^2 + 0.3s + 1}{s^2 + 0.4s + 1}$ 的模拟滤波器，绘出其幅频和相频

谱。

```
a=[1,0.4,1];
```

```
b=[0.2,0.3,1];
```

```
w=logspace(-1,1); %产生从 10-1到 101之间地 50 个等间距点,即 50 个频率点
```

freqs(b,a,w);

运行得幅频特性和相频特性如图 6-1 所示:

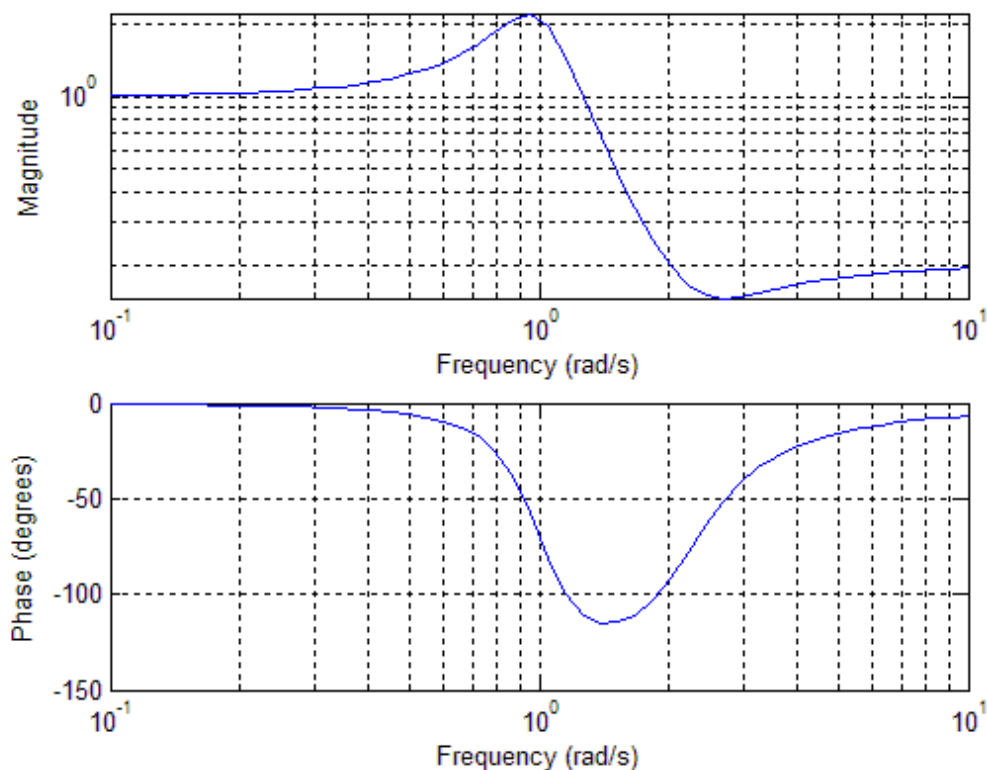


图 6-1 模拟滤波器的幅频特性和相频特性

②freqz 函数: 求数字滤波器的频率响应

freqz(b,a,w)计算由向量 w 指定的频率点上模拟滤波器 $H(z)$ 的频率响应, 其中 b 和 a 分别为数字滤波器系统函数 $H(z)$ 的分子与分母。freqz 函数自动将频率点均匀设置在频率范围 $[0, \Pi]$ 上, 并将自动绘出幅频和相频曲线。

实验任务 2: 系统传输函数为 $H(z) = \frac{0.2 + 0.3z + z^{-2}}{1 + 0.4z + z^{-2}}$ 的数字滤波器, 绘出其幅频和相频谱。

a=[1,0.4,1];

b=[0.2,0.3,1];

freqz(b,a,128);

运行得幅频特性和相频特性如图 6-2 所示:

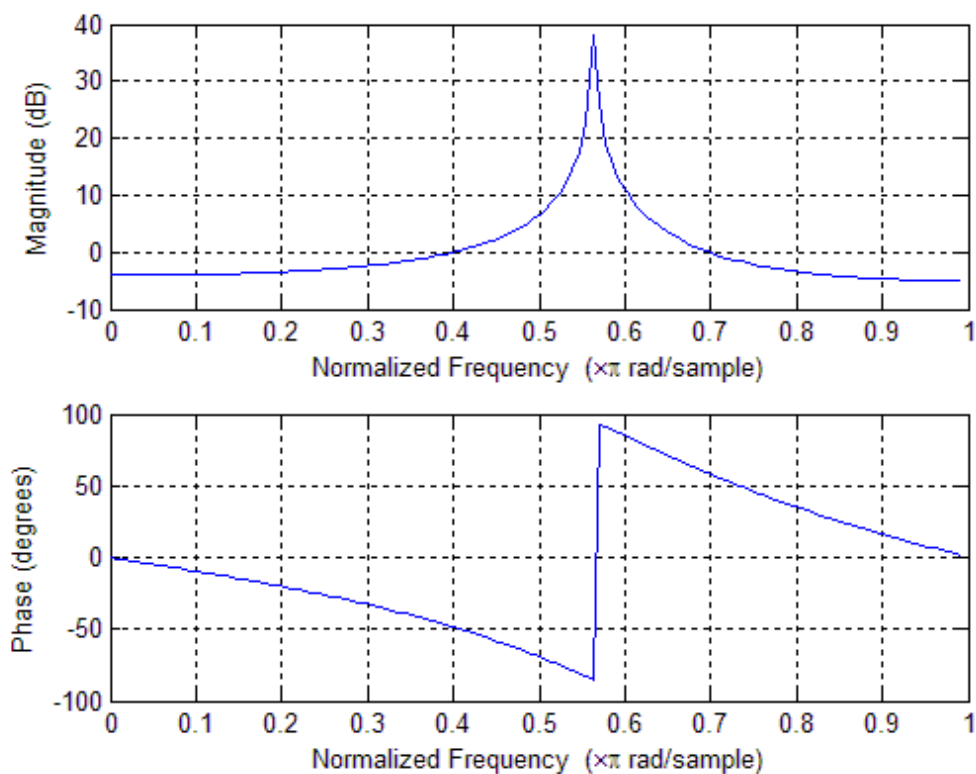


图 6-2 数字滤波器的幅频特性和相频特性

(3) ButterWorth(巴特沃斯)模拟和数字滤波器的设计

①直接设计

I. **buttord** 函数: ButterWorth 滤波器阶数的选择 (即 ButterWorth 滤波器的最小阶数的计算)

ButterWorth 数字滤波器: 调用格式: $[n,Wn]=buttord(Wp,Ws,Rp,Rs)$

ButterWorth 模拟滤波器: 调用格式: $[n,Wn]=buttord(Wp,Ws,Rp,Rs,'s')$,其中's'为变元

在给定滤波器性能的情况下 (通带临界频率 Wp 、阻带临界频率 Ws 、通带由最大衰减 Rp 和阻带内最小衰减 Rs), 计算 ButterWorth 滤波器的阶数 n 和截止频率 Wn 。

II. **butter** 函数: ButterWorth 滤波器设计

ButterWorth 数字滤波器: 调用格式: $[b,a]=butter(n,Wn)$

ButterWorth 模拟滤波器: 调用格式: $[b,a]=butter(n,Wn,'s')$,其中's'为变元

根据滤波器的阶数 n 和截止频率 Wn 计算 ButterWorth 滤波器分子和分母系数 (b 为分子系数的矢量形式, a 为分母系数的矢量形式)。

实验任务 3: 采用直接设计方法设计一个 ButterWorth 数字滤波器满足以下参数: 采样频率为 1Hz, 通带临界频率 $Wp=0.2\text{Hz}$ 、通带内衰减小于 1dB ($Rp=1$);阻带临界频率 $Ws=0.3\text{Hz}$ 、阻带内衰减大于 25dB($Rs=25$)。


```
[n,Wn]=buttord(0.2,0.3,1,25);
```

```
[b,a]=butter(n,Wn);
```

```
freqz(b,a,128);
```

运行得幅频特性和相频特性如图 6-3 所示：

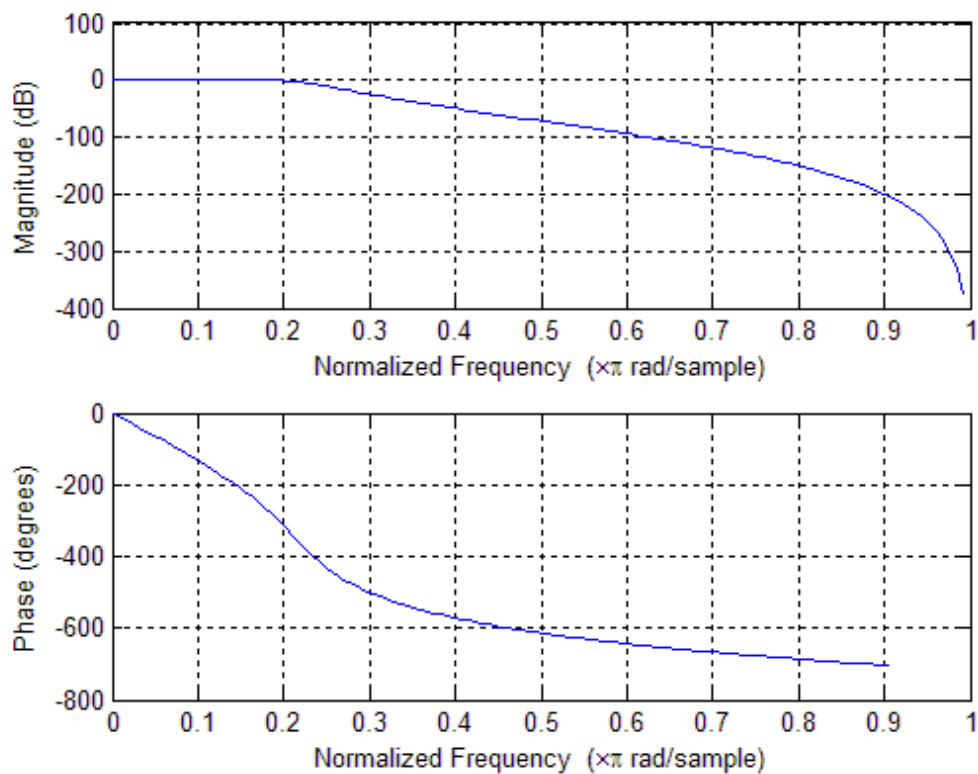


图 6-3 直接设计的 ButterWorth 数字滤波器幅频特性和相频特性

②脉冲响应不变法设计 ButterWorth(巴特沃斯)数字滤波器

设计思路：先设计模拟滤波器，再模拟滤波器转换成数字滤波器。

调用格式： $[bz,az]=impinvar(b,a,Fs)$ ，在给定模拟滤波器参数 b 、 a 和采样频率 F_s 的前提下，把模拟滤波器的参数变为近似等价的数字滤波器参数，从而设计数字滤波器。

实验任务 4：采用脉冲响应不变法设计上述 ButterWorth(巴特沃斯)数字滤波器

```
[n,Wn]=buttord(0.2,0.3,1,25,'s');
```

```
[b,a]=butter(n,Wn,'s');
```

```
[bz,az]=impinvar(b,a,1);
```

```
freqz(bz,az,128);
```

运行得到幅频特性和相频特性如图 6-4 所示：

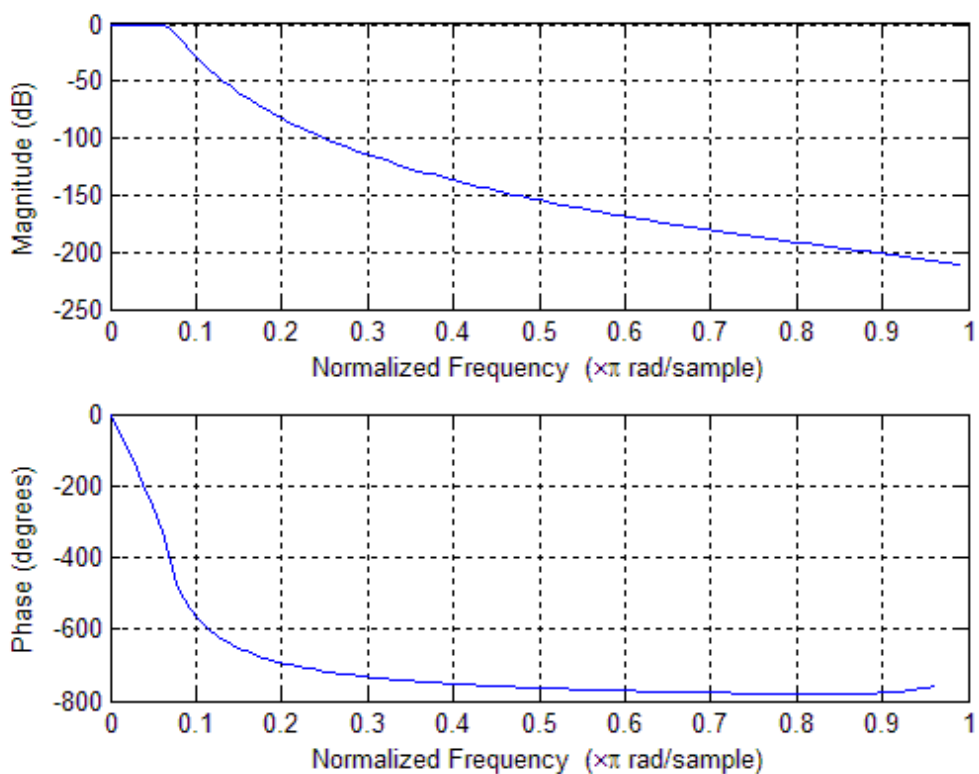


图 6-4 脉冲响应不变法设计的 ButterWorth 数字滤波器幅频特性和相频特性

(3) 双线性不变法设计 ButterWorth(巴特沃斯)数字滤波器

设计思路：先设计模拟滤波器，再模拟滤波器转换成数字滤波器。

调用格式： $[bz,az]=bilinear(b,a,Fs)$,在给定模拟滤波器参数 b,a 和采样频率 Fs 的前提下，把模拟滤波器的参数变为近似等价的数字滤波器参数，从而设计数字滤波器。

实验任务 5：采用双线性不变法设计上述 ButterWorth(巴特沃斯)数字滤波器

```
[n,Wn]=buttord(0.2,0.3,1,25,'s');
```

```
[b,a]=butter(n,Wn,'s');
```

```
[bz,az]=bilinear(b,a,1);
```

```
freqz(bz,az,128);
```

运行得幅频特性和相频特性如图 6-5 所示：

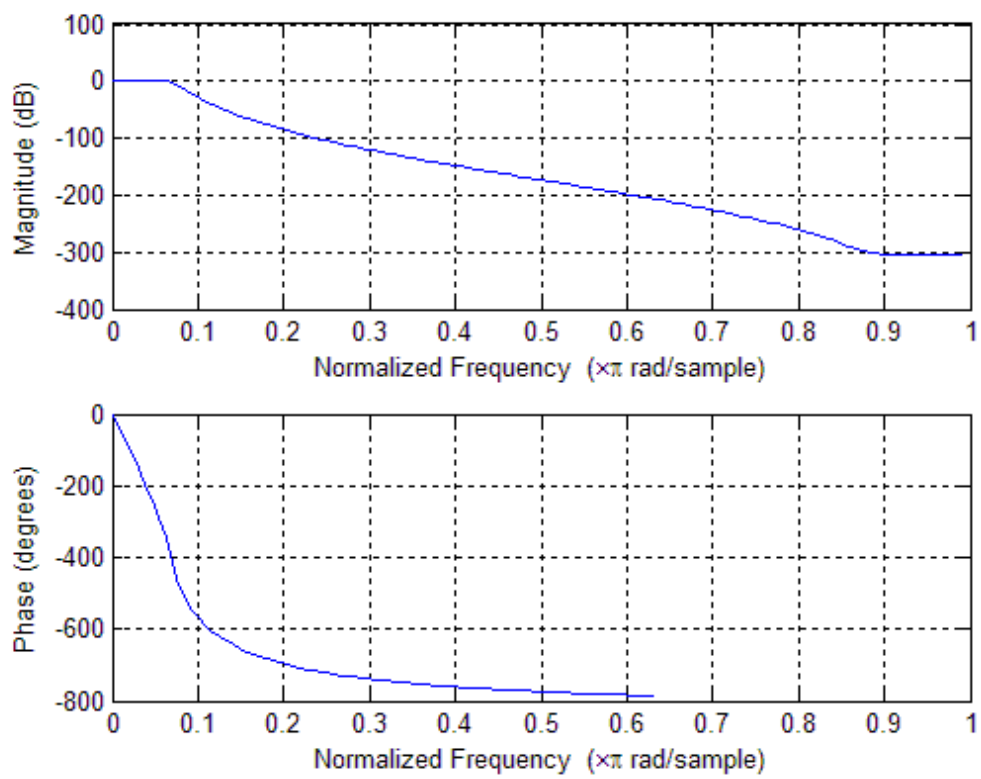


图 6-5 双线性不变法设计的 ButterWorth(巴特沃斯)数字滤波器幅频特性和相频特性

实验七 FIR 滤波器设计

1. 实验目的

- (1) 熟悉 MATLAB 编程技术。
- (2) 熟悉用窗函数法设计 FIR 数字滤波器。

2. 实验仪器及设备

- (1) 计算机一台。
- (2) MATLAB6.5 以上软件一套。

3. 实验任务及原理

(1) 在 MATLAB 中产生窗函数的方法

1) 矩形窗 (Rectangle Window)

调用格式: $w=\text{boxcar}(n)$, 根据长度 n 产生一个矩形窗 w 。

2) 三角窗 (Triangular Window)

调用格式: $w=\text{triang}(n)$, 根据长度 n 产生一个三角窗 w 。

3) 汉宁窗 (Hanning Window)

调用格式: $w=\text{hanning}(n)$, 根据长度 n 产生一个汉宁窗 w 。

4) 海明窗 (Hamming Window)

调用格式: $w=\text{hamming}(n)$, 根据长度 n 产生一个海明窗 w 。

5) 布拉克曼窗 (Blackman Window)

调用格式: $w=\text{blackman}(n)$, 根据长度 n 产生一个布拉克曼窗 w 。

6) 恺撒窗 (Kaiser Window)

调用格式: $w=\text{Kaiser}(n, \beta)$, 根据长度 n 和影响窗函数旁瓣的 β 参数产生一个恺撒窗 w 。

(2) 基本窗函数的 FIR 滤波器的设计——标准频率响应形状

利用 MATLAB 提供的函数 `fir1` 来实现。

调用格式: `fir1(n,Wn,'ftype',window)`, n 为阶数、 W_n 是截止频率 (如果输入是形如 $[W_1 \ W_2]$ 的矢量时, 本函数将设计带通滤波器, 其通带为 $W_1 < \omega < W_2$)、`ftype` 是滤波器的类型 (低通—省略、高通—`ftype=high`、带阻—`ftype=stop`)、`window` 是窗函数, 默认为海明窗。

实验任务 1: 利用矩形窗设计一个长度为 8、截止频率为 0.4Hz 的线性相位 FIR 滤波器。

```
window=boxcar(8);
```

```
b=fir1(7,0.4>window);
```

```
freqz(b,1);
```

运行得幅频相频特性如图 7-1 所示:

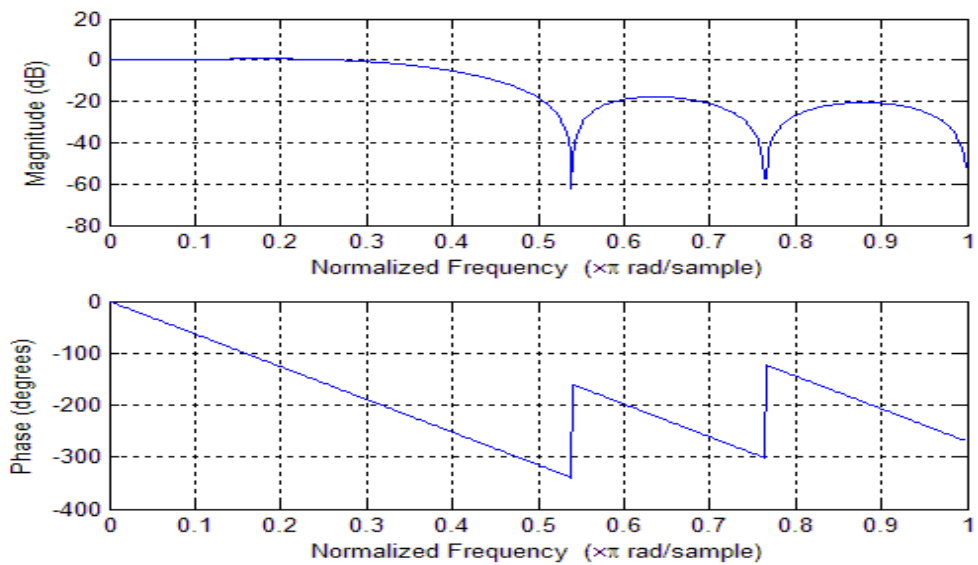


图 7-1 利用矩形窗的 FIR 滤波器幅频相频特性

实验任务 2: 利用布拉克曼窗设计一个长度为 8、截止频率为 0.4Hz 的线性相位 FIR 滤波器。

```
window=blackman(8);
```

```
b=fir1(7,0.4>window);
```

```
freqz(b,1);
```

运行得幅频相频特性如图 7-2 所示:

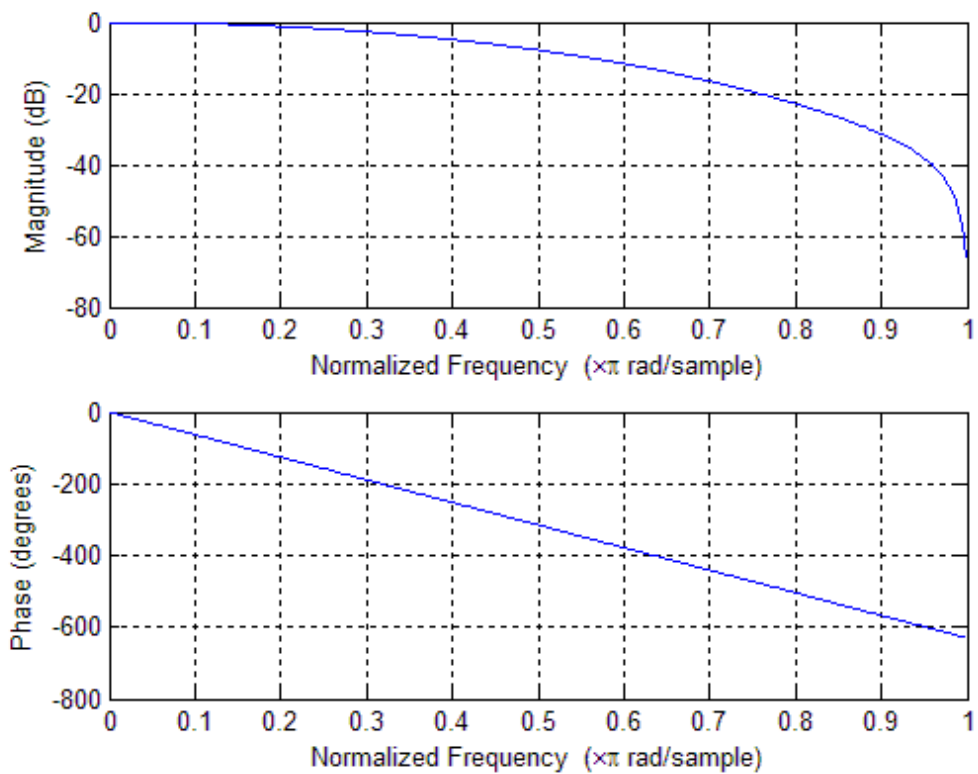


图 7-2 利用布拉克曼窗设计的 FIR 滤波器幅频相频特性

实验任务 3: 设计线性相位 FIR 带通滤波器,其长度 $N=16$, 上下边带截止频率分别为 $\omega_1=0.3$

π , $\omega_2=0.5\pi$

```
window=blackman(16);
```

```
b=fir1(15,[0.3 0.5],window);
```

```
freqz(b,1);
```

运行得幅频相频特性如图 7-3 所示:

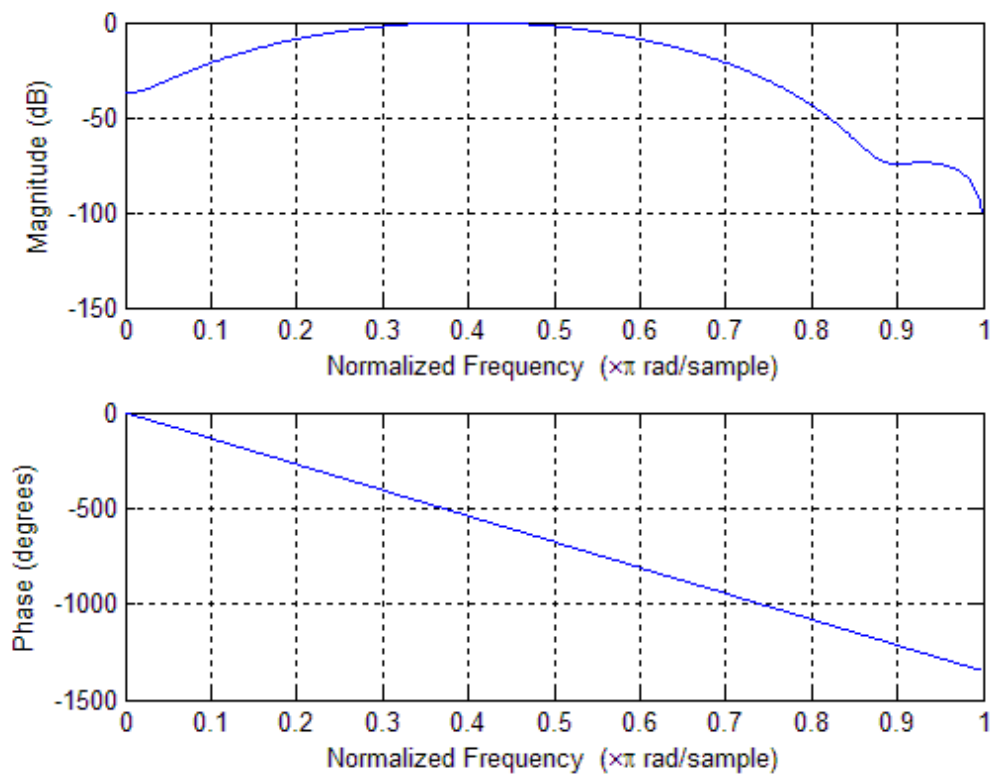


图 7-3 线性相位 FIR 带通滤波器的幅频相频特性