实验一、离散时间信号与系统

一、实验目的

1. 掌握常用序列的matlab实现方法;

2. 掌握序列运算的matlab实现方法;

3. 掌握序列的卷积和运算的matlab实现方法;

二、 实验要求

1. 利用 matlab 程序，生成几种常用的序列，如矩形序列，单位脉冲序列等,绘制图形，观察序列特征;

2. 利用 matlab 程序，实现序列的常见运算,如加法，乘法等;

3. 利用matlab程序,实现卷积和运算;

三、 实验步骤

1.序列的实现,输入程序,生成下列序列,并观察序列特征;

(1)单位抽样序列

```
function [x,n] = impseq(n0,n1,n2)
 %产生 x(n) = delta(n - n0);n1 <= n0 <= n2
 % ----------------------------------
 %[x,n] = impseq(n0,n1,n2)
 %
if((n0 < n1)|(n0 > n2)|(n1 > n2))
     error('参数必须满足 n1 <= n0 <= n2')
end
n = [n1:n2];
 %x = [zeros(1,(n0 - n1)),1,zeros(1,(n2 - n0))];
x = [(n - n0) == 0];
```

(2)单位阶跃序列

```
function [x,n] = stepseq(n0,n1,n2)
% 产生 x(n) = u(n - n0);n1 < = n0 < = n2
% ---------------------------------
% [x,n] = stepseq(n0,n1,n2)
if((n0 < n1)|(n0 > n2)|(n1 > n2))
    error('参数必须满足 n1 < = n0 < = n2')
end
n = [n1:n2];
% x = [zeros(1,(n0 - n1)),ones(1,(n2 - n0 + 1))];
x = [(n - n0)> = 0];
```

(3) 矩形序列，可以利用两个单位阶跃序列来生成矩形序列

```
function [x,n]=RN(np1,ns,nf)
N=np1;
n=ns:nf;
np=0;
x=stepseq(np,ns,nf)-stepseq(np1,ns,nf);
stem(n,x);title('矩形序列RN');
```

(4) 实指数序列

实现$x(n) = (0.8)^n, 0 \le n \le 10$

```
n=0:10;
x=(0.8).^n;
stem(n,x);
title('实指数序列');
```

(5) 复指数序列

$$x(n)=\begin{cases}e^{(\alpha+j\omega)n}, & n\geqslant 0\\ 0, & n<0\end{cases}$$

当 $\omega=0$ 时，它就是实指数序列，但是是以 e 为底的实指数序列。

如果 $\alpha=0$，则是纯虚指数序列，$e^{j\omega}=\cos(\omega n)+j\sin(\omega n)$，这一序列的实部为余弦序列，虚部为正弦序列。

实现复指数序列$\mathbf{x(n)}=e^{(0.4+0.6j)n}$，其中$-1\leq \mathbf{n}\leq 10$

```
clc;clear all
n0 = - 1;n2 = 10;
n = n0:n2;
x = exp((0.4 + 0.6j) * n);                                    %复指数序列
figure(1)
subplot(211)
stem(n,real(x),'.');
axis([ - 4,10,min(real(x)) - 1,1.2 * max(real(x))])
title('复指数序列')
ylabel('实部');grid;
subplot(212)
stem(n,imag(x),'.');
axis([ - 4,10,min(imag(x)) - 1,1.2 * max(imag(x))])
ylabel('虚部');xlabel('n');grid;
```

(6)周期序列

设 $x(n)$，$0\leqslant n\leqslant N-1$，MATLAB 可利用 **xtide(n)** 把它拓展成 K 个周期的周期序列，有两种办法。

① 简单复制。

```
xtide[x,x,…,x];nxtide = 0:k * N - 1
```

即原 $x(n)$ 长度为 $N=\text{length}(x)$，复制 K 次得到 xtide 长度为 $K*N$，但 K 太大时容易出错。

② 用求余数的方法(模运算)。

函数 **n1=mode(n,N)完成模运算** $n1=(n \bmod N)=((n))_N$

即完成 $n1=n+KN, 0\leqslant n1\leqslant N-1$, K 整数,即余数 n1 在 0 到 N−1 之间。

把这一运算用到位置向量上,就可实现有限长序列的周期延拓。

设 x 的起始位置为 0,长度为 N 可用以下语句:

```
nxtide = 0:K * N - 1;
xtide = x(mod(nxtide,N) + 1);
```

第二条语句中,由于 mod(nxtide,N)的值永远在 0 到 N−1 之间,而 MATLAB 的规则是 x 的下标为 nx=[1:N],故上面的后一条语句(即第二条语句)中必须把 mod 函数的结果加 1 才对。

实现: $\mathbf{x(n)=[\underline{1},2,3,4]}$,求将它延拓 5 个周期所得到的序列

```
clc;clear all
x = [1,2,3,4];N = length(x);k = 5;
nx = 0:N - 1;
ny = 0:(k * N - 1);
y = x(mod(ny,N) + 1);
figure(1)
subplot(211),stem(nx,x,'.');

axis([ - 1,N + 1,0,5]);grid;
subplot(212),stem(ny,y,'.');
axis([ - 1,k * N,0,5]);grid;
```

2. 序列的运算

(1) 序列的和

(1) 序列之和。将两序列 x1、x2 位置序号相同的样值相加,必须使 x1、x2 的位置向量的起点、终点相同,因而序列的长度相同,故有可能要补“零值”。

首先,使 x1、x2 有同样的位置向量,因而要注意 MATLAB 的下标运算,将 x1 的位置向量 n1 及 x2 的位置向量 n2 都变成相同的位置向量 n,显然,min(n)应等于 min(n1)和 min(n2)中的小者。max(n)应等于 max(n1)和 max(n2)中的大者。随后,将 x1 及 x2 放到扩展后的位置向量 n 上,由于 n 比 n1、n2 都可能长,故应将 x1、x2 长度加长,加长后分别用 y1,y2 表示,即将 x1、x2 放到位置向量 n 的对应位置上,将多余之处的元素置零。MATLAB 程序如下。

```
function[y,n] = seqadd(x1,n1,x2,n2)
% 序列之和
n = min(min(n1),min(n2)):max(max(n1),max(n2));          % 位置向量 n
y1 = zeros(1,length(n));y2 = y1                          % y 向量初始化为零
y1(find((n>=min(n1))&(n<=max(n1)) == 1)) = x1;           % 把 x1 延拓到 n 上形成 y1
y2(find((n>=min(n2))&(n<=max(n2)) == 1)) = x2;           % 把 x2 延拓到 n 上形成 y2
y = y1 + y2;
```

其中 find 函数用来寻找非零元素对应的下标区间。

（2）序列的乘积

```
function[y,n] = seqmult(x1,n1,x2,n2)
% 实现 y(n) = x1(n) * x2(n)
% --------------------
% [y,n] = seqmult(x1,n1,x2,n2)
%      y = 在 n 区间上的乘积序列,n 包含 n1 和 n2
%      x1 = 在 n1 上的第一序列
%      x2 = 在 n2 上的第二序列(n2 可与 n1 不等)
%
n = min(min(n1),min(n2)):max(max(n1),max(n2));          % y(n)的长度
y1 = zeros(1,length(n));y2 = y1;                         %
y1(find((n>=min(n1))&(n<=max(n1)) == 1)) = x1;           % 具有 y 的长度的 x1
y2(find((n>=min(n2))&(n<=max(n2)) == 1)) = x2;           % 具有 y 的长度的 x2
y = y1. * y2;                                            % 序列相乘
```

已知两序列为 $x_1(n) = [1,3,5,7,6,4,2,1]$，起始位置 ns1=-3, $x_2(n) = [4,0,2,1,-1,3]$, 起始位置 ns2=1, 求它们的和 ya 以及乘积 ym

```
%已知两序列为 x1,x2 有
%x1 = [1,3,5,7,6,4,2,1],起始位置 ns1 = -3
%x2 = [4,0,2,1,-1,3],起始位置 ns2 = 1
%求他们的和 ya 以及乘积 ym
clc;clear all
x1 = [1,3,5,7,6,4,2,1];ns1 = -3                          %给定 x1 及它的起始点位置 ns1
x2 = [4,0,2,1,-1,3];ns2 = 1;                             %给定 x2 及它的起始点位置 ns2
nf1 = ns1 + length(x1) - 1;                              %求出 x1,x2 的终点位置 nf1,nf2
nf2 = ns2 + length(x2) - 1;                              %故有 nx1 = ns1:nf1,nx2 = ns2:nf2
n1 = ns1:nf1
n2 = ns2:nf2
n = min(ns1,ns2):max(nf1,nf2)                            %y 的位置向量
y1 = zeros(1,length(n));y2 = y1                          %y1,y2 序列的初始化
y1(find((n>=ns1)&(n<=nf1)==1)) = x1;                     %给 y1 赋值 x1
y2(find((n>=ns2)&(n<=nf2)==1)) = x2;                     %给 y2 赋值 x2
ya = y1 + y2;ym = y1.*y2;                                %序列相加后相乘
subplot(221);stem(n1,x1,'.');ylabel('x1(n)');grid;
subplot(223);stem(n2,x2,'.');xlabel('n');ylabel('x2(n)');grid;
subplot(222);stem(n,ya,'.');ylabel('y1(n) + y2(n)');grid;
subplot(224);stem(n,ym,'.');xlabel('n');ylabel('y1(n)*y2(n)');grid;
```

(3)移位 $y(n)=x(n-m)$

```
function[y,ny] = seqshift(x,nx,m)
%实现 y(n) = x(n-m)
%------------------
%[y,n] = seqshift(x,nx,m)
%
ny = nx + m;y = x;
```

(4)翻褶 $y(n)=x(-n)$

$$y(n)=x(-n)$$

在 MATLAB 中用 **fliplr** 函数来实现,采用两次调用 fliplr 的办法:第 1 次 y=fliplr(x)把行向量 x 中元素的排列次序左右翻转;第 2 次 ny=-fliplr(nx)将位置向量左右翻转,且改变正负号。这相当于对 n=0 点左右翻转后,再将 n>0 的位置改写为 n<0。

以此两次翻转可写成翻褶函数 seqfold:m,即

```
function[y,ny] = seqfold[x,nx]
%序列翻褶
y = fliplr(x);ny = -fliplr(nx);
```

3.卷积和运算

(1) 序列的卷积。

$$y(n)=\sum_{m=-\infty}^{\infty}x(m)h(n-m)=\sum_{m=-\infty}^{\infty}h(m)x(n-m)$$

MATLAB 提供了内部函数 conv 计算两个有限长序列的卷积，应注意两点：第一，它只对有限长序列作卷积。第二，这一卷积假定都是从 $n=0$ 开始，调用方法为 y=conv(x,h)，不需要输入序列的位置信息，从而也无法给出输出序列的位置信息。因此需要将函数 conv 加以扩展。

设已知两个有限长序列 x 和 h，其位置向量已知，即有{x(n)；nx=nx1：nx2}及{h(n)；nh=nh1：nh2}。要求 $x(n)$ 与 $h(n)$ 的卷积 $y(n)$，以及其位置向量 ny。我们已经知道 $y(n)$ 的位置向量的起点终点应满足

```
ny1 = nx1 + nh1,   ny2 = nx2 + nh2
```

而 $y(n)$ 的长度应满足 length(y)=length(x)+length(h)-1，此时**可利用 MATLAB 中的 conv 函数，编出有位置矢量为 ny 输出为 y(n)的 convwthn.m 程序为**

```
% 本函数利用 MATLAB 中的 conv 函数，编出有位置矢量 ny 的输出 y(n)的 convwthn.m 程序
function[y,ny] = convwthn[x,nx,h,nh]
y = conv(x,h);
ny1 = nx(1) + nh(1);
ny2 = nx(end) + nh(end);                    % end 表示最后一个下标
ny = [ny1:ny2];
```

已知$\mathbf{x(n)=[1,\underline{2},3,-1,-1],h(n)=[2,2,1,\underline{-1},4,-2]}$，求

$$\mathbf{y(n)=x(n)*h(n)}$$

```
clc;clear all
x = [1,2,3, - 1, - 2];nx = - 1:3
h = [2,2,1, - 1,4, - 2];nh = - 3:2
[y,ny] = convwthn(x,nx,h,nh)
stem(ny,y,'.');xlabel('n');ylabel('y(n)');grid;
```

四、思考练习

1. 编程产生复指数序列$\mathbf{x(n)=0.5^n e^{j0.8n}}$，其中$\mathbf{-2\le n\le 10}$，要求绘制出序列的实部和虚部；

2. 已知 序列$\mathbf{x(n)=[\underline{1},2,3,4,5,6,7,8,9]}$，要求：

（1）编程生成并绘制该序列；

（2）生成并绘制将该序列延拓 3 个周期后得到的序列；

（3）生成并绘制$\mathbf{y(n)=x(n-3)}$序列；

（4）生成并绘制$\mathbf{y(n)=x(n+2)}$序列；

（5）生成并绘制$\mathbf{y(n) = x(-n)}$序列；

3．已知 序列$\mathbf{x(n) = [\underline{1}, 2, 3, 4, 5], h(n) = [-5, 4, \underline{7}, 9]}$, 要求：

（1）编程生成并绘制上述两序列；

（2）生成并绘制$\mathbf{y_1(n) = x(n) + h(n)}$序列；

（3）生成并绘制$\mathbf{y_2(n) = x(n)h(n)}$序列；

（4）生成并绘制$\mathbf{y_3(n) = x(n) * h(n)}$序列；

五、实验报告要求

1． 完成实验步骤中所有函数的输入和验证程序，记录实验结果；

2． 完成所有的思考练习，在实验报告中记录所有的编写的程序以及运行结果；

3． 用不少于 100 字总结此实验；