

第一部分

基础实验



时域离散信号的产生及运算

一、实验目的

- (1) 学习 MATLAB 的使用方法并理解其基本功能；
- (2) 掌握产生常用时域离散信号的方法及信号的显示；
- (3) 掌握基本信号的简单运算方法。

二、实验原理

1. MATLAB 的使用方法及基本功能

打开 MATLAB 软件,界面上有菜单、图标、小窗口,常用的为命令窗口(command window),可以在该窗口中输入命令,执行后可以在工作空间窗口(workspace window)中观察变量发生的变化;更常用的窗口为编辑窗口(edit window),通常将程序代码写在该窗口内,执行后,可以在工作空间窗口中观察变量的取值。

读者可以做以下两个小练习:

- (1) 单击界面内各个子菜单,了解其功能。
- (2) 在命令窗口中输入自己熟悉的命令,按 Enter 键执行后,观察工作空间窗口中变量的取值变化。

2. 常用离散信号的产生

【例 1-1】 离散指数信号通常表示为

$$x(n) = a^n$$

用 MATLAB 中的 stem 函数绘出 0.8^n 序列, n 的取值为 $0 \sim 20$ 。

程序如下:

```
% Lab1_1.m
n = [0:3:20];           % 给出序号序列
x = (0.8).^n;          % 给出值序列
stem(n,x);             % 以序号序列和值序列进行绘图
```

```
xlabel('n');ylabel('x(n)');
```

```
%必要的标记
```

将上述代码输入编辑窗口,保存后单击 run 图标,结果如图 1-1 所示。

【例 1-2】 离散正弦序列表示为

$$x(n) = K \sin(\omega_0 n + \theta_0)$$

试绘制 $x(n) = 2 \sin\left(0.02\pi n + \frac{\pi}{4}\right)$ 序列。

程序如下:

```
% Lab1_2.m
```

```
n = [0:10:100];
```

```
x = 2 * sin(0.02 * pi * n + pi/4);
```

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

```
xlabel('n');ylabel('x(n)');
```

```
% 给出序号序列
```

```
% 给出值序列
```

```
% 绘制离散图
```

```
% 必要标记
```

程序运行结果如图 1-2 所示。

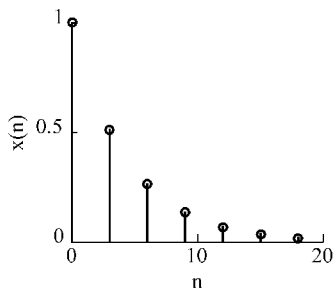


图 1-1 0.8^n 序列

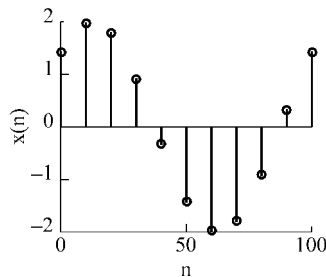


图 1-2 $x(n) = 2 \sin\left(0.02\pi n + \frac{\pi}{4}\right)$ 序列

【例 1-3】 单位阶跃信号的离散形式为

$$u(n) = \begin{cases} 0, & n < 0 \\ 1, & n \geq 0 \end{cases}$$

用序号序列自 $-5 \sim 5$ 表示单位阶跃信号。

程序如下:

```
% Lab1_3.m
```

```
n0 = 0;
```

```
n1 = -5;
```

```
n2 = 5;
```

```
n = [n1:n2];
```

```
x = [(n - n0) == 0];
```

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

```
xlabel('n');ylabel('x(n)');
```

```
% 给出序号序列
```

```
% 给出值序列,应注意只有当 n - n0 == 0 时值为 1, 否则为 0
```

```
% 绘出离散序列
```

程序运行结果如图 1-3 所示。

【例 1-4】 斜坡信号为

$$r(n) = \begin{cases} 0, & n < 0 \\ n, & n \geq 0 \end{cases}$$

用 1s 的采样间隔自 0~10 采样, 绘出离散斜坡信号。

程序如下:

```
% Lab1_4.m
n1 = 0; n2 = 10;           % 设置起始和终止值
n = [n1:1:n2];           % 以 1s 的时间间隔给出时间序列
x = n;                    % 得到斜坡信号
stem(n, x)
xlabel('n'); ylabel('x(n)');
```

程序运行结果如图 1-4 所示。

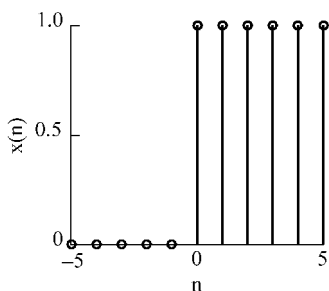


图 1-3 阶跃序列

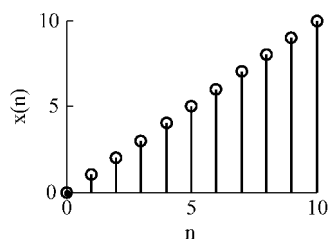


图 1-4 斜坡序列

【例 1-5】 符号转换函数表示为

$$\text{sgn}(t) = \begin{cases} 1, & t < 0 \\ -1, & t > 0 \end{cases}$$

用序号序列自 -5~5 表示单位阶跃信号。

程序如下:

```
% Lab1_5.m
n0 = 0;
n1 = -5;
n2 = 5;
mn = [n1:n2];
for n = n1:n2
    if (n - n0) < 0
        x(n+6) = -1;
    else
        x(n+6) = 1;
    end
end
x(6) = 0;
stem(mn, x);
xlabel('n'); ylabel('x(n)');
```

程序运行结果如图 1-5 所示。

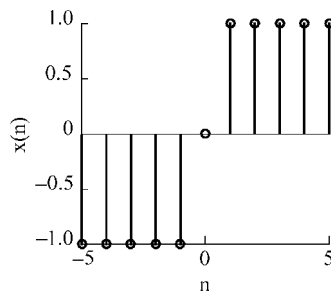


图 1-5 符号序列

【例 1-6】 离散脉冲信号可用下式表示：

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$

用序号序列自 $-20 \sim 20$ 表示单位脉冲信号。

程序如下：

```
% Lab1_6.m
n0 = 0;
n1 = -20;
n2 = 20;
n = [n1:n2];           % 序号序列
x = [(n - n0) == 0];   % 值序列, 应注意只有当 n - n0 = 0 时值才为 1, 否则为 0
stem(n, x)            % 绘出离散序列
xlabel('n'); ylabel('x(n)');
```

程序运行结果如图 1-6 所示。

【例 1-7】 sinc 信号可表示为

$$\text{sinc}(n) = \frac{\sin n}{n}$$

用 0.1s 的采样间隔自 $-10 \sim 10\text{s}$ 采样, 绘出离散 sinc 信号。

程序如下：

```
% Lab1_7.m
n = -10:0.1:10;       % 给出自变量序列值
x = sin(pi * n + eps) ./ (pi * n + eps); % eps 是 MATLAB 系统的精度, 这里防止被零除
plot(n, x);           % 绘图
xlabel('n'); ylabel('x(n)');
```

程序运行结果如图 1-7 所示。

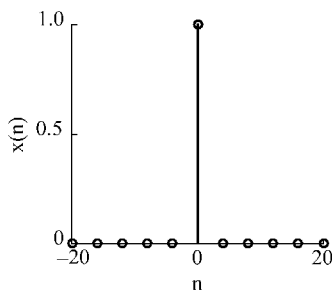


图 1-6 脉冲序列

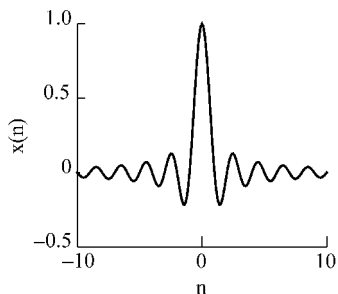


图 1-7 sinc 序列

【例 1-8】 复指数信号 (complex exponential signal) 可表示为

$$f(n) = e^{(\sigma + j\omega)n} = e^{\sigma n} (\cos \omega n + j \sin \omega n)$$

用 $0 \sim 40$ 间隔为 5 的序号序列, 绘出离散 $f(n) = e^{(-0.2 + j0.4)n}$ 的实部信号、虚部信号、振幅信号、相位信号信息。

程序如下：

```

% Lab1_8.m
clf                                     % 清除绘图板
n = [0:5:40];                          % 给出序号序列
alpha = -0.2 + 0.4 * j;                % 给出指数序列
x = exp(alpha * n);                    % 给出复指数信号
Real = real(x);                        % 取复指数信号的实部
Imag = imag(x);                        % 取复指数信号的虚部
Mag = abs(x);                           % 取复指数信号的振幅
Pha = (180/pi) * angle(x);            % 取复指数信号的相位,转化为度
subplot(2,2,1), stem(n, Real);         % 绘制复指数信号的实部
ylabel('Real');
subplot(2,2,2), stem(n, Imag);        % 绘制复指数信号的虚部
ylabel('Imag');
subplot(2,2,3), stem(n, Mag);         % 绘制复指数信号的振幅
xlabel('n'); ylabel('Mag');
subplot(2,2,4), stem(n, Pha);        % 绘制复指数信号的相位
xlabel('n'); ylabel('Pha');

```

程序运行结果如图 1-8 所示。

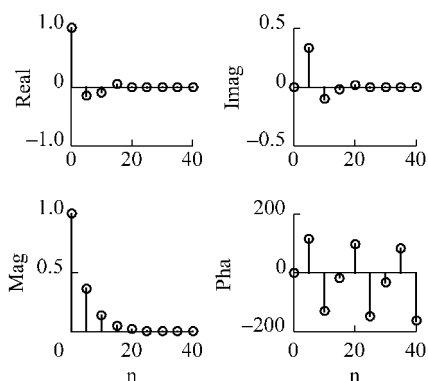


图 1-8 复指数序列

【例 1-9】 随机序列指的是离散随机信号,常用来模拟随机噪声。MATLAB 函数库中提供了 rand 和 randn 函数,可以生成随机信号,其中,

rand(1,n)产生 1 行 n 列在 $[0,1]$ 上均匀分布的随机序列函数。

randn(1,n)产生 1 行 n 列均值为 0、方差为 1 的高斯随机序列,即白噪声序列。

试绘出序号序列自 1~100 间隔为 10 的随机信号。

程序如下:

```

% Lab1_9.m
n = 1:10:100;                          % 序号序列
N = length(n);
x1 = rand(1,N);                          % rand 产生的随机值序列
subplot(2,1,1), stem(n, x1); hold on; plot(n, zeros(1,N)); % 绘制随机序列
ylabel('x1(n)');
x2 = randn(1,N);                         % randn 产生的随机值序列

```

```
subplot(2,1,2),stem(n,x2);hold on;plot(n,zeros(1,N)); % 绘制横轴
xlabel('n');ylabel('x2(n)');
```

程序运行结果如图 1-9 所示。

3. 离散信号的基本运算

1) 信号时移

给定 $x(n)$, $y(n) = x(n-k)$ 称为 $x(n)$ 的移位序列。当 $k > 0$ 时,称为延序列,原序列右移;当 $k < 0$ 时,称为超前序列,原序列左移。

【例 1-10】 将序列 $x = \sin(2\pi n/15)$ 移位为 $x = \sin(2\pi(n+4)/15)$ 和 $x = \sin(2\pi(n-4)/15)$ 。

程序如下:

```
% Lab1_10.m
n = -10:10;
k1 = 4;
k2 = -4;
x = sin(2 * pi * n/15);
x1 = sin(2 * pi * (n - k1)/15);
x2 = sin(2 * pi * (n - k2)/15);
subplot(311),stem(n,x);ylabel('x(n)');
subplot(312),stem(n,x1);ylabel('x1(n)');
subplot(313),stem(n,x2);xlabel('n');ylabel('x2(n)');
```

程序运行结果如图 1-10 所示。

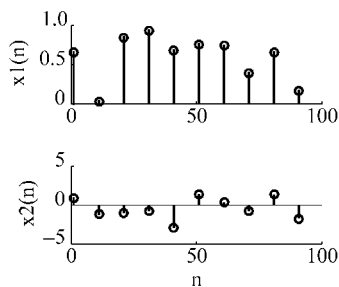


图 1-9 随机序列

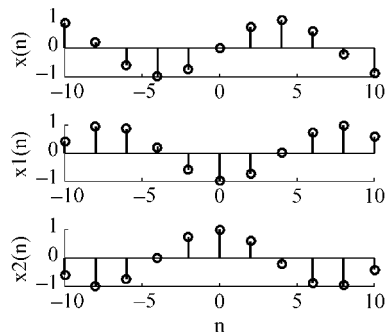


图 1-10 随机序列

2) 信号折叠

给定 $x(n)$, $y(n) = x(-n)$ 称为 $x(n)$ 的折叠序列。

【例 1-11】 将信号 $x = \exp(-0.6n)$ 进行折叠。

程序如下:

```
% Lab1_11.m
n = -10:5;
x = exp(-0.6 * n);
x1 = fliplr(x);
n1 = -fliplr(n);
```

% x 是原始序列
% x1 是折叠后的序列

```
subplot(211),stem(n,x);ylabel('x(n)');
subplot(212),stem(n1,x1);xlabel('n');ylabel('x1(n)');
```

程序运行结果如图 1-11 所示。

3) 信号尺度改变

给定 $x(t)$, $y(t)=x(at)$ 称为 $x(t)$ 的不同尺度序列。

【例 1-12】 试将信号 $y=\sin t$ 的周期分别扩展两倍和压缩一半。

程序如下：

```
% Lab1_12.m
n = -63:63;
dt = 0.1; t = n * dt;
y = sin(t);
y1 = sin(2 * t);
y2 = sin(0.5 * t);
subplot(3,1,1),plot(n,y1);
subplot(3,1,2),plot(n,y);
subplot(3,1,3),plot(n,y2);
```

% y 是原始信号
% y1 是周期压缩一半的信号
% y2 是周期扩展 2 倍的信号

程序运行结果如图 1-12 所示。

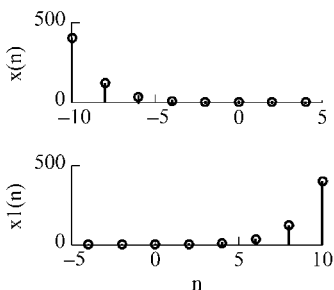


图 1-11 信号折叠

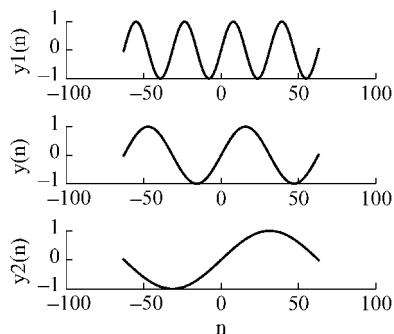


图 1-12 信号压缩、扩展

4) 信号相加、相乘

给定两个序列 $x_1(n)$ 、 $x_2(n)$, 定义 $y(n)=x_1(n)+x_2(n)$ 为和序列, 定义 $y(n)=x_1(n) * x_2(n)$ 为积序列。

【例 1-13】 构造两个序列, 进行相加和相乘运算。

程序如下：

```
% Lab1_13.m
x1 = [2,1,2,3,4,3,2,1,3]; nx1 = -2;
x2 = [2,2,0,0,0,-2,-2]; nx2 = 2;
nf1 = nx1 + length(x1) - 1;
nf2 = nx2 + length(x2) - 1;
ny = min(nx1, nx2) : max(nf1, nf2);
xx1 = zeros(1, length(ny)); xx2 = xx1;
xx1(find((ny == nx1) & (ny == nf1) == 1)) = x1;
```

```

xx2(find((ny == nx2)&(ny == nf2) == 1)) = x2;
ya = xx1 + xx2;
yp = xx1 .* xx2;
subplot(411), stem(ny, xx1); ylabel('xx1(n)');
subplot(412), stem(ny, xx2); ylabel('xx2(n)');
subplot(413), stem(ny, ya); ylabel('ya(n)');
subplot(414), stem(ny, yp); xlabel('ny'); ylabel('yp(n)');

```

程序运行结果如图 1-13 所示。

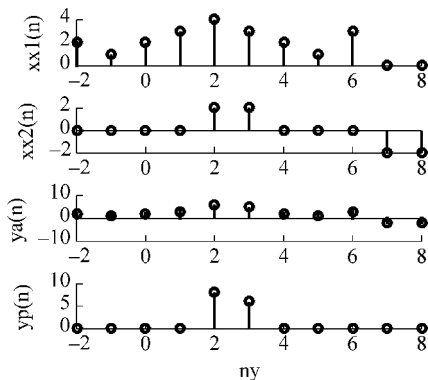


图 1-13 信号相加、相乘

【例 1-14】 画出序列 $x(n) = 3\delta(n+3) - \delta(n-4)$, $-5 \leq n \leq 5$ 。

程序如下：

```

% Lab1_14.m
n = [-5:5];
x = 3 * imp(-3, -5, 5) - imp(4, -5, 5);
stem(n, x);
xlabel('n');
ylabel('x(n)');

function [x, n] = imp(n0, n1, n2)
% generates x(n) = delta(n - n0); n1 = n = n2
n = [n1:n2]; x = [(n - n0) == 0];
end

```

程序运行结果如图 1-14 所示。

【例 1-15】 画出序列 $x(n) = u(n-5) - u(n-15)$, $0 \leq n \leq 20$ 。

程序如下：

```

% Lab1_15.m
n0 = 10; n1 = 0; n2 = 30; n = n1:n2;
x = step(n0, n1, n2) - step(20, n1, n2);
stem(n, x);
xlabel('n'); ylabel('x(n)');

```