目录

实验三　离散时间信号和系统的时域分析

江苏师范大学 • 电气工程及自动化学院 • 李灿

1　实验目的

1. 掌握离散时间信号的 MATLAB 表示方法，绘制离散时间信号图；
2. 掌握离散时间信号的基本运算；
3. 掌握离散系统时域响应的计算；
4. 熟悉离散系统时域响应特征。

2　实验原理

离散时间信号可以由连续时间信号抽样得到，描述的是在某些特定的时间点上的信号值。在 MATLAB 中绘制连续时间信号波形时，需要先给出或者求出离散时间上的点（常用足够小的等间隔取值），然后通过描点连线的方式获得（实验一和实验二中就是这么做的），绘制波形曲线时采用 `plot()` 函数。实际上，如果某个信号本身就是一些离散的值（像数列那样），除了这些点以外都没有定义，那么可以用棉棒图的方式表示这样的信号，即离散时间信号。在 MATLAB 中常采用 `stem()` 函数来实现。和连续时间信号采用 $f(t)$ 来表示不同，离散时间信号采用 $f[k]$ 的记法，其中 $k \in \mathbb{Z}$ 为序列的标号。例如离散时间信号

$$f[k] = \{0,\ 2,\ 0,\ ,1,\ 3,\ 1,\ 0\} \tag{1}$$

$$f[k] = \sin[k], \quad k \in \mathbb{Z} \tag{2}$$

许多常见的连续时间信号都可以通过将 t 换为 k 的方式来获得它们的离散时间版本。需要注意的是，离散时间信号只在一些孤立的点上有定义，因此就不能像连续时间信号中采用极限模型去定义单位冲激信号 $\delta(t)$ 了。离散时间信号中对应的 $\delta[k]$ 称为单位脉冲序列，定义为

$$\delta[k] = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases} \tag{3}$$

由于单位阶跃序列的自然表示为

$$u[k] = \begin{cases} 1, & k \geqslant 0 \\ 0, & k < 0 \end{cases} \tag{4}$$

所以，单位脉冲序列和单位阶跃序列的关系为

$$u[k]=\sum_{n=-\infty}^{k}\delta[n] \tag{5}$$

$$\delta[k]=u[k]-u[k-1] \tag{6}$$

式(5)的右端称为单位脉冲序列的求和，式(6)的右端称为单位阶跃序列的差分。求和与差分的作用对应于连续时间信号中的积分和微分。

离散时间信号的基本运算包括两类：关于自变量变换的翻转、平移和“展缩”，关于信号值运算的加法、乘积、差分与求和。需要特别注意的是，这里的“展缩”和连续时间信号的有所区别，离散信号中表示将原离散序列样本个数减少或者增加。后面将结合例子进行演示。

连续时间 LTI 系统的时域模型用线性常系数微分方程进行描述。类似地，离散时间 LTI 系统的时域模型用线性常系数差分方程进行描述。差分方程也可以表示系统动态，只是这种动态体现在不同时间点之间的迭代关系，这种特性使得差分方程的求解更加简单，能够从上一步推导出下一步的值。连续时间系统通过周期采样，并且计算单个采样周期内的状态转移矩阵，就能获得精确的离散化系统方程（这一点在《计算机控制系统理论》等课程中有均有体现）。当然，有的系统本身就可以建模为离散时间系统，取决于观测的时间是否连续。例如，每月统计额金融数据、每隔单位时间检测的运动物体行程等。

对于 n 阶离散时间 LTI 系统，其模型可以用如下 n 阶差分方程描述：

$$\sum_{i=0}^{n}a_iy[k-i]=\sum_{j=0}^{m}b_jf[k-j] \tag{7}$$

它在结构上和用来描述 n 阶连续时间 LTI 系统的 n 阶微分方程十分类似：

$$\sum_{i=0}^{n}a_iy^{(i)}(t)=\sum_{j=0}^{m}b_jf^{(j)}(t) \tag{8}$$

从式(7)不难得出，只要已知系统的初始状态和输入信号，它的完全响应可以编写为迭代方程

$$y[k]=-\sum_{i=1}^{n}\left(\frac{a_i}{a_0}\right)y[n-i]+\sum_{j=0}^{n}\left(\frac{b_j}{b_0}\right)f[n-j] \tag{9}$$

这种迭代格式用来做仿真分析非常方便，不过有时候要在理论上分析系统特性则需要求出 $y[k]$ 关于 k 的表达式，也就是求解差分方程。

在 MATLAB 中，提供了一维滤波函数 `filter()` 来求零状态响应，`impz()` 函数求单位脉冲响应，`stepz()` 函数求单位阶跃响应。和连续时间 LTI 系统一样，离散时间 LTI 系统的零状态响应 $y_f[k]$ 也可以利用激励信号 $f[k]$ 和单位脉冲响应 $h[k]$ 的卷积和来求得，即

$$y_f[k]=f[k]*h[k] \tag{10}$$

3 程序讲解

3.1 离散时间信号的表示

下面讲到的离散时间信号实际上都是指序列 $f[k]$，根据指定 k 的取值范围，可以确定序列的个数，也就是 MATLAB 中用多少维向量来表示序列。

3.1.1 指数序列

指数序列 a^k，在 MATLAB 中可以用点幂（按元素求幂）函数表示，调用格式为如下任意一种形式：

```
f = a .^ k
f = power(a, k)
```

如此就能获得 f 关于 k 的一个序列。利用 MATLAB 中的 `stem()` 函数将该序列中的每一个点绘制出来，即为指数序列的波形图。

例：用 MATLAB 表示指数序列 $f[k]=2(-0.9)^k$，并绘制其波形，其中 $k=0\sim 50$。

```
Kf = 50;
k = 0 : Kf;
A = 2;
a = -0.9;
fk = A * a .^ k;
stem(k, fk, 'filled', 'linewidth', 2);
axis([0 Kf -2 2])
```

其中

```
fk = A * a .^ k;
```

可以替换为

```
fk = power(a, k);
```

3.1.2 正弦序列

正弦序列 $A\sin(\omega_0 k+\varphi)$ 和 $A\cos(\omega_0 k+\varphi)$，在 MATLAB 中可以用 `sin()` 和 `cos()` 函数表示，调用格式为

```
f = A * sin(omega_0*k + varphi)
f = A * cos(omega_0*k + varphi)
```

例：用 MATLAB 表示正弦信号 $f(t)=\sin\left(\frac{\pi}{12}k\right)$，并绘制其波形，其中 $k=0\sim 50$。

```
Kf = 50;
k = 0 : Kf;
a = -0.9;
fk = sin(pi/12 * k)
stem(k, fk, 'filled', 'linewidth', 2, 'color', 'red')
axis([0 Kf -1.5 1.5])
```

扩展：自行利用 `sinc()` 函数绘制抽样序列。

3.1.3 单位脉冲序列

单位脉冲序列仅在 $k=0$ 处为 1，其他地方均为 0。带延迟的单位脉冲序列则仅在 $k=k_0$ 处为 1，MATLAB 表示的程序举例为

```
Kf = 50;
k = 0 : Kf;
k0 = 5;
fk = [ k - k0 == 0 ];
stem(k, fk, 'filled', 'linewidth', 2)
axis([0 Kf -1 2])
```

这里利用逻辑运算符 `==`，取出 $k=k_0$ 点，返回逻辑真，即 1；其他地方为假，即为 0。

3.1.4 单位阶跃序列

单位阶跃序列的表示方法和单位脉冲序列类似，只需要利用逻辑运算符，为特定点之后的序列赋值为 1 即可，MATLAB 表示的程序举例为

```
Kf = 50;
k = 0 : Kf;
k0 = 5;
fk = [ k - k0 >= 0 ];
stem(k, fk, 'filled', 'linewidth', 2)
axis([0 Kf -1 2])
```

扩展：请用类似的方法表示一个矩形脉冲序列。

3.1.5 * 白噪声序列

在信号处理中常常用到白噪声序列，它是一类随机信号。虽然不是本课程的重点，但是可以方便地利用 MATLAB 进行表示，可以作为了解内容。MATLAB 中提供的 `rand()` 函数可以产生 $[0,1]$ 区间内均匀分布的白噪声序列，而 `randn()` 函数可以产生均为 0，方差为 1 的高斯白噪声。MATLAB 表示的程序举例为

```
Kf = 20;
k = 0 : Kf - 1;
fk1 = rand(1, Kf);
fk2 = randn(1, Kf);
hold on
stem(k, fk1, 'linewidth', 2)
stem(k, fk2, 'linewidth', 2)
```

注意信号个数 `Kf` 和 `k` 大小的关系。可以多次运行程序，观察现象。

3.2 离散时间信号的基本运算

3.2.1 自变量变换

对于信号的自变量变换，数学表达式上只需将相应 $f[k]$ 改为 $f[k-k_0]$（平移）、$f[-k]$（翻转）、$f[Mk]$（M 倍抽取或者 $1/M$ 倍内插，参考教材 p45）即可。在 MATLAB 中，实现起来要相对复杂一些。

平移可以看作将离散序列的时间序号向量平移，而表示对应时间序号的序列样值不变。由于离散序列通常仅能确定 k 的大小，因此在非函数表达式中，需要指定序列的坐标原点（$k=0$ 对应的序列元素）或者指定序列的起点（序列第一个元素对应的 k 值）。在进行自变量变换时，需要关注变换前后 k 取值范围的变化。

例：已知序列 $f[k]=\{0,\ 2,\ 0,\ 1,\ 3,\ 1,\ 0,\ 1\}_{-3}$（下标表示序列第一个元素对应 $k=-3$），求 $f[k-2]$、$f[k+2]$，并绘制波形图。

```
fk = [0, 2, 0, 1, 3, 1, 0, 1];
k = -3 : length(fk) - 4;
k1 = k - 2;
k2 = k + 2;
subplot(3,1,1)
stem(k, fk, 'linewidth', 1.5)
axis([-5 6 -1 4])
subplot(3,1,2)
stem(k1, fk, 'linewidth', 1.5)
axis([-5 6 -1 4])
subplot(3,1,3)
stem(k2, fk, 'linewidth', 1.5)
axis([-5 6 -1 4])
```

翻转需要将序列以零时刻（$k=0$）为基准点，以纵轴为对称轴翻转。可以利用 MATLAB 提供的 `fliplr(fk)` 函数实现翻转后的序列样值，而翻转后的坐标则可以由`-fliplr(k)` 得到。

例：已知序列 $f[k]=\{0,\ 2,\ 0,\ 1,\ 3,\ 1,\ 0,\ 1\}_{-3}$（下标表示序列第一个元素对应 $k=-3$），求 $f[-k]$，并绘制波形图。

```
fk = [0, 2, 0, 1, 3, 1, 0, 1];
k = -3 : length(fk) - 4;
k1 = -fliplr(k);
fk1 = fliplr(fk);
subplot(2,1,1)
stem(k, fk, 'linewidth', 1.5)
axis([-5 5 -1 4])
subplot(2,1,2)
stem(k1, fk1, 'linewidth', 1.5)
axis([-5 5 -1 4])
```

M **倍抽取**是将原离散序列每隔 $M-1$ 点抽取一点，构成新的序列；而 $L := 1/M$ **倍内插**将原离散序列每两点之间插入 $L-1$ 个零点（内插包括插零、复制、线性插值、高次插值等），构成新的序列。它们分别可以用 MATLAB 中的 `decimate()` 函数和 `interp()` 函数来实现。

例：已知某序列表达式为 $f[k] = \sin\left(\frac{\pi}{12}k\right)$，求 $f[2k]$ 和 $f[k/2]$，并绘制波形图。

```
Kf = 50;
k = 0 : Kf-1;
fk = sin(pi/12 * k);
% 2倍抽取
k1 = 1 : 2 : Kf;
fk1 = decimate(fk, 2);
% 2倍内插
k2 = 0.5 : 0.5 : Kf;
fk2 = interp(fk, 2);
subplot(3, 1, 1)
stem(k, fk, 'linewidth', 1, 'color', 'blue')
legend('$f[k]$', 'interpreter', 'latex', 'fontsize', 12);
subplot(3, 1, 2)
stem(k1, fk1, 'linewidth', 1, 'color', 'red')
legend('$f[2k]$', 'interpreter', 'latex', 'fontsize', 12);
subplot(3, 1, 3)
stem(k2, fk2, 'color', 'black')
legend('$f[k/2]$', 'interpreter', 'latex', 'fontsize', 12);
xlabel('Time (k)')
```

注意：上面的程序是在不放缩时间尺度时的结果，如果考虑时间尺度变换，即如果 k 到 $k+1$ 的间隔时间不变时，程序修改如下

```
Kf = 50;
k = 0 : Kf-1;
fk = sin(pi/12 * k);
% 2倍抽取
k1 = 1 : Kf/2;
fk1 = decimate(fk, 2);
% 2倍内插
k2 = 1 : 2*Kf;
fk2 = interp(fk, 2);
subplot(3, 1, 1)
stem(k, fk, 'linewidth', 1, 'color', 'blue')
legend('$f[k]$', 'interpreter', 'latex', 'fontsize', 12);
axis([0 2*Kf -1 1]);
subplot(3, 1, 2)
```

```
stem(k1, fk1, 'linewidth', 1, 'color', 'red')
legend('$f[2k]$', 'interpreter', 'latex', 'fontsize', 12);
axis([0 2*Kf -1 1]);
subplot(3, 1, 3)
stem(k2, fk2, 'color', 'black')
legend('$f[k/2]$', 'interpreter', 'latex', 'fontsize', 12);
xlabel('Time (k)')
```

3.2.2 加法与乘积

离散序列的加法和乘积与连续时间信号的相同，都是对向量进行加法和按元素的乘法，当两个序列的长度不一样时，做加法和乘积需要先对长度小的进行 0 元素扩充。

例：已知信号 $f_1[k] = \{1,\ 0,\ 1,\ 0,\ 1\}_{-3}$，$f_2[k] = \{2,\ 1,\ 2,\ 1,\ 2,\ 2\}_{-1}$，分别计算 $g_1[k] = f_1[k] + f_2[k]$ 与 $g_2[k] = f_1[k] \cdot f_3[k]$，并绘制其波形图。

观察两个序列发现第 1 个序列的起点是 $k = -3$，共有 5 个元素，则 k 的取值范围为 $-3 \sim 1$；第 2 个序列的起点是 $k = -1$，共有 6 个元素，则 k 的取值范围为 $-1 \sim 4$。两个序列做加法和乘积运算时，需要利用 0 元素将第 1 个序列往右扩充，将第 2 个序列往左扩充，是的两个序列中 k 的取值范围均变为 $-3 \sim 4$。程序如下

```
figure('position', [0 0 1000 600])
% 实现原信号
k1 = -3 : 1;
f1 = [1, 0, 1, 0, 1];
subplot(4, 1, 1)
stem(k1, f1, '-k', 'linewidth', 2)
legend('$f_1[k]$', 'interpreter', 'latex', 'fontsize', 12);
axis([-3 4 0 3])
k2 = -1 : 4;
f2 = [2 1 2 1 2 2];
subplot(4, 1, 2)
stem(k2, f2, '--k', 'linewidth', 2)
legend('$f_2[k]$', 'interpreter', 'latex', 'fontsize', 12);
axis([-3 4 0 3])
% 实现加法和乘积
k = -3 : 4;
f1 = [f1, zeros(1, length(k)-length(k1))];
f2 = [zeros(1, length(k)-length(k2)), f2];
g1 = f1 + f2;
g2 = f1 .* f2;
subplot(4, 1, 3)
stem(k, g1, '-r', 'linewidth', 2)
```

```
legend('$g_1[k]=f_1[k]+f_2[k]$', 'interpreter', 'latex', 'fontsize', 12);
axis([-3 4 0 3])
subplot(4, 1, 4)
stem(k, g2, '-r', 'linewidth', 2)
legend('$g_2[k]=f_1[k]\cdot f_2[k]$', 'interpreter', 'latex',...
                                      'fontsize', 12);
axis([-3 4 0 3])
xlabel('$k$', 'interpreter', 'latex')
```

3.2.3 差分与求和

差分与求和对应连续时间信号中的微分与积分，不过差分与求和的计算要相对简单一些，在 MATLAB 的调用格式为

```
d_f = diff(fk)
s_f = sum( f(k1:k2) )
```

调用格式中的 `k1:k2` 表示求和的范围。

例：已知离散信号的归一化能量定义为

$$E = \lim_{N\to\infty} \sum_{k=-N}^{N} |f[k]|^2 \tag{11}$$

则计算指数信号 $f[k] = (-0.6)^k u[k]$ 能量的程序为

```
k = 0 : 10;
fk = (-0.6) .^ k;
E_fk = sum( abs(fk) .^ 2 )
```

3.3 离散时间系统的零状态响应

零状态响应指的是初始状态为零，系统对输入信号的响应。MATLAB 中，提供了 `filter()` 函数求离散时间系统的零状态响应。调用格式为

```
yf = filter(b, a, f);
```

其中，`b` 和 `a` 分别为差分方程右端和左端的系数，`f` 为输入信号。

例：已知离散时间系统的方程和输入信号为

$$y[k] - 0.8y[k-1] + 0.01y[k-2] = 2f[k]$$
$$f[k] = \cos(\pi k)\, u[k]$$

则求该系统零输入响应的程序为：

```
b = 2;
a = [1, -0.8, 0.01];
```

```
k = 0 : 60;
fk = cos( pi*k );
yf = filter(b, a, fk);
stem(k, yf, 'color', 'black', 'linewidth', 2)
```

3.4 单位脉冲响应与单位阶跃响应

在 MATLAB 中，直接提供了可以求线性 LTI 离散时间系统单位脉冲响应和单位阶跃响应的功能函数，分别为 `impz()` 和 `stepz()`，其调用格式为

```
h = impz(b, a, k)
g = stepz(b, a, k)
```

其中第 3 个参数既可以是表示时间的向量，也可以表示时间区间的上界。

例：已知某系统的差分方程：

$$y[k] + 0.3y[k-1] + 0.2y[k-2] = f[k]$$

求该系统的单位脉冲响应 $h[k]$ 和单位阶跃响应 $g[k]$，并绘制两条响应曲线图形的程序为

```
k = 0 : 50;
b = 1;
a = [1, 0.3, 0.2];
h = impz(b, a, k);
g = stepz(b, a, k);
subplot(2, 1, 1)
stem(k, h, 'color', 'red', 'linewidth', 1.5)
subplot(2, 1, 2)
stem(k, g, 'color', 'black', 'linewidth', 1.5)
```

或者

```
k = 50;
b = 1;
a = [1, 0.3, 0.2];
[h, t] = impz(b, a, k);
[g, t] = stepz(b, a, k);
subplot(2, 1, 1)
stem(t, h, 'color', 'red', 'linewidth', 1.5)
subplot(2, 1, 2)
stem(t, g, 'color', 'black', 'linewidth', 1.5)
```

3.5 卷积和

在连续时间系统分析中，已经用过离散序列的卷积和去定义卷积积分，因而关于这一点应该比较熟悉了，调用格式为

```
y = conv(f, h)
```

下面仅以例子进行说明。

例：已知序列 $f[k]=\{1,\ 2,\ 3,\ 4\}_0$，$h[k]=\{1,\ 1,\ 1,\ 1,\ 1,\ 1\}_0$，计算卷积和 $y[k]=f[k]*h[k]$，并绘制图形。

```
f = [1, 2, 3, 4];
h = [1, 1, 1, 1, 1, 1];
y = conv(f, h);
k = 0 : length(y)-1;
stem(k, y, 'color', 'red', 'linewidth', 1.5)
figure
subplot(2, 1, 1)
k1 = 0 : length(f)-1;
stem(k1, f, 'color', 'blue', 'linewidth', 1.5)
subplot(2, 1, 2)
k2 = 0 : length(h)-1;
stem(k2, h, 'color', 'blue', 'linewidth', 1.5)
```

3.6 仿真工具箱

利用 MATLAB 的 simulink 工具箱，搭建简单的离散系统模型，观察不同激励信号的响应曲线。

4 实验内容

1. 利用 MATLAB 绘制以下离散序列的图形：
   (1) $f[k]=2\delta[k-5]$
   (2) $f[k]=u[k]$
   (3) $f[k]=u[k+3]-2u[k-5]$
   (4) $f[k]=ku[k]$
   (5) $f[k]=2(0.9)^k\sin(0.5\pi k)$
2. 已知离散序列表达式为 $f[k]=(0.9)^k\sin(0.5\pi k)(u[k]-u[k-20])$，请计算 $f[k-3]$，$f[-k]$，$f[4k]$ 和 $f[k/4]$，并绘制相应的图形。
3. 已知系统的差分方程为

$$y[k]+1.2y[k-1]-0.3y[k-2]=f[k]+2f[k-1]$$
$$f[k]=2\sin(\pi n/6)u[k]$$

   (1) 利用 `filter()` 函数求该系统的零状态响应，绘制响应曲线；
   (2) 利用 `filter()` 函数求该系统的单位脉冲响应，绘制响应曲线。

4. 已知系统的差分方程为

$$y[k]+0.7y[k-1]-0.45y[k-2]-0.6y[k-3]=0.8f[k]-0.44f[k-1]+0.36f[k-2]+0.02f[k-3]$$

   (1) 利用 `impz()` 函数求该系统的单位脉冲响应，绘制响应曲线；
   (2) 利用 `stepz()` 函数求该系统的单位阶跃响应，绘制响应曲线；
   (3) 根据响应曲线，分析该系统是否稳定？

5. 已知某离散时间系统的单位脉冲响应为 $h[k]=u[k]-u[k-5]$，请计算输入信号为 $f[k]=(0.5)^k(u[k]-u[k-5])$ 时的零状态响应 $y[k]$，并绘制 $f[k]$、$h[k]$ 和 $y[k]$ 的图形。
6. * 利用 Simulink 工具箱搭建一个简单的离散时间 LTI 系统，设置激励信号，观察系统响应。

5 实验报告

1. 简述实验目的和实验原理。
2. 完成实验内容部分的第 1–5 题，要求独自完成程序编写和绘图，写入报告。
3. 选做实验内容部分第 6 题，完成的内容也写入报告。
4. 总结实验过程，记录心得体会。

参考文献

[1] 陈后金, 胡健, 薛健. 信号与系统 (第三版). 北京: 清华大学出版社, 2017.

[2] 龚晶, 许凤慧, 卢娟, 等. 信号与系统实验. 北京: 机械工业出版社, 2022.

[3] 徐亚宁, 唐璐丹, 王旬, 等. 信号与系统分析实验指导书 (MATLAB 版). 西安: 西安电子科技大学出版社, 2012.