

实验二 系统响应及系统稳定性

一、实验目的

- (1) 学会运用 Matlab 求解离散时间系统的零状态响应；
- (2) 学会运用 Matlab 求解离散时间系统的单位取样响应；
- (3) 学会运用 Matlab 求解离散时间系统的卷积和。

二、实验原理及实例分析

1. 离散时间系统的响应

离散时间 LTI 系统可用线性常系数差分方程来描述，即

$$\sum_{i=0}^N a_i y(n-i) = \sum_{j=0}^M b_j x(n-j) \quad (2-1)$$

其中， a_i ($i=0, 1, \dots, N$) 和 b_j ($j=0, 1, \dots, M$) 为实常数。

Matlab 中函数 `filter` 可对 (2-1) 的差分方程，对在指定时间范围内的输入序列所产生的响应进行求解。函数 `filter` 的语句格式为

$$y = \text{filter}(b, a, x)$$

其中， x 为输入的离散序列； y 为输出的离散序列； y 的长度与 x 的长度一样； b 与 a 分别为差分方程右端与左端的系数向量。

【实例 2-1】已知某 LTI 系统的差分方程为

$$3y(n) - 4y(n-1) + 2y(n-2) = x(n) + 2x(n-1)$$

试用 Matlab 命令绘出当激励信号为 $x(n) = (1/2)^n u(n)$ 时，该系统的零状态响应。

解：Matlab 源程序为

```
>>a=[3 -4 2];
```

```
>>b=[1 2];
```

```

>>n=0:30;
>>x=(1/2).^n;
>>y=filter(b,a,x);
>>stem(n,y,'fill'),grid on
>>xlabel('n'),title('系统响应 y(n)')

```

程序运行结果如图 2-1 所示。

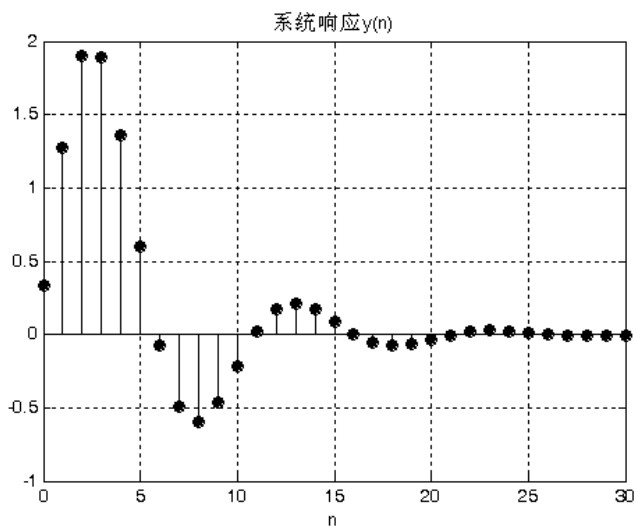


图 2-1 实例 2-1 系统的零状态响应

2. 离散时间系统的单位取样响应

系统的单位取样响应定义为系统在 $\delta(n)$ 激励下系统的零状态响应，用 $h(n)$ 表示。Matlab 求解单位取样响应可利用函数 `filter`，并将激励设为单位抽样序列。例如，求解实例 2-1 中系统的单位取样响应时，MATLAB 源程序为：

```

>>a=[3 -4 2];
>>b=[1 2];
>>n=0:30;
>>x=(n==0);    %产生单位抽样序列
>>h=filter(b,a,x);
>>stem(n,h,'fill'),grid on
>>xlabel('n'),title('系统单位取样响应 h(n)')

```

程序运行结果如图 2-2 所示。

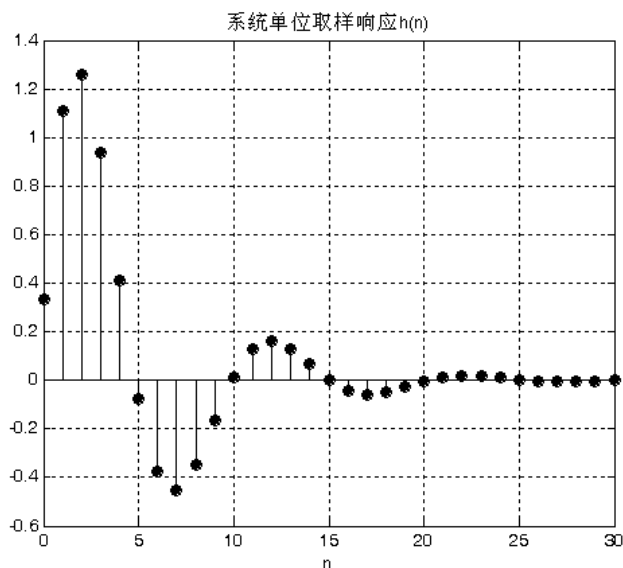


图 2-2 实例 2-1 的系统单位取样响应

Matlab 另一种求单位取样响应的方法是利用控制系统工具箱提供的函数 `impz` 来实现。`impz` 函数的常用语句格式为

$$\text{impz}(b,a,N)$$

其中，参数 N 通常为正整数，代表计算单位取样响应的样值个数。

【实例 2-2】 已知某 LTI 系统的差分方程为

$$3y(n) - 4y(n-1) + 2y(n-2) = x(n) + 2x(n-1)$$

利用 Matlab 的 `impz` 函数绘出该系统的单位取样响应。

解：MATLAB 源程序为

```
>>a=[3 -4 2];
>>b=[1 2];
>>impz(b,a,30);
>>grid on;
>>title('系统单位取样响应 h(n)');
```

程序运行结果如图 2-3 所示，比较图 2-1 和图 2-2，不难发现结果相同。

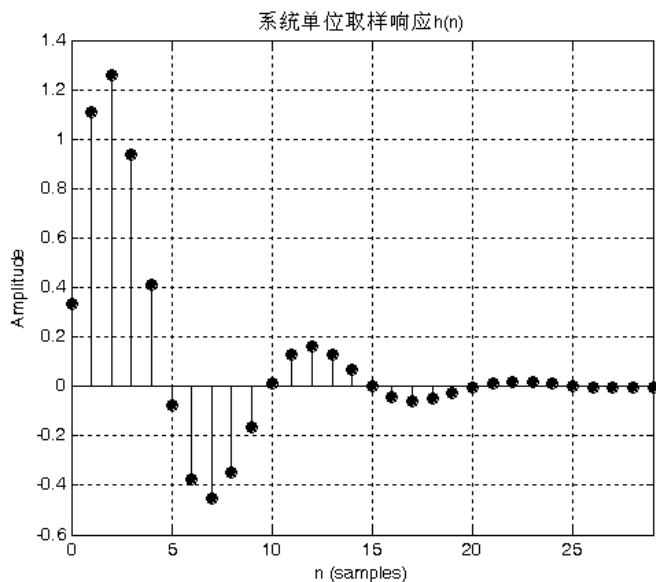


图 2-3 系统单位取样响应

3. 离散时间信号的卷积和运算

由于系统的零状态响应是激励与系统的单位取样响应的卷积，因此卷积运算在离散时间信号处理领域被广泛应用。离散时间信号的卷积定义为

$$y(n) = x(n) * h(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m) \quad (2-2)$$

可见，离散时间信号的卷积运算是求和运算，因而常称为“卷积和”。

Matlab 求离散时间信号卷积和的命令为 `conv`，其语句格式为

$$y = \text{conv}(x, h)$$

其中， x 与 h 表示离散时间信号值的向量； y 为卷积结果。用 Matlab 进行卷积和运算时，无法实现无限的累加，只能计算时限信号的卷积。

例如，利用 Matlab 的 `conv` 命令求两个长为 4 的矩形序列的卷积和，即 $g(n) = [u(n) - u(n-4)] * [u(n) - u(n-4)]$ ，其结果应是长为 7 ($4+4-1=7$) 的三角序列。用向量 `[1 1 1 1]` 表示矩形序列，MATLAB 源程序为

```
>>x1=[1 1 1 1];
```

```
>>x2=[1 1 1 1];
```

```
>>g=conv(x1,x2)
```

```
g=
```

```
1      2      3      4      3      2      1
```

如果要绘出图形来，则利用 stem 命令，即

```
>>n=0:6;
```

```
>>stem(n,g,'fill'),grid on,xlabel('n')
```

程序运行结果如图 2-4 所示。

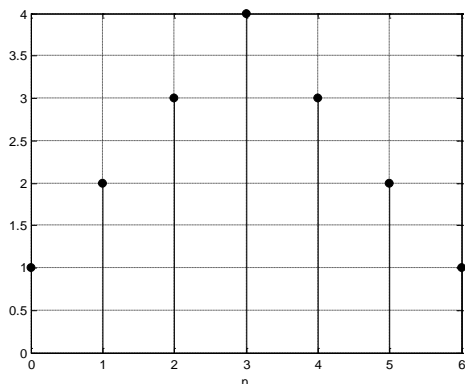


图 2-4 卷积结果图

对于给定函数的卷积和，应计算卷积结果的起始点及其长度。两个时限序列的卷积和长度等于两个序列长度的和减 1。

【实例 2-3】已知某系统的单位取样响应为 $h(n) = 0.8^n [u(n) - u(n-8)]$ ，试用 Matlab 求当激励信号为 $x(n) = u(n) - u(n-4)$ 时，系统的零状态响应。

解：Matlab 中可通过卷积求解零状态响应，即 $x(n) * h(n)$ 。由题意可知，描述 $h(n)$ 向量的长度至少为 8，描述 $x(n)$ 向量的长度至少为 4，因此为了图形完整美观，我们将 $h(n)$ 向量和 $x(n)$ 向量加上一些附加的零值。MATLAB 源程序为

```
>>nx=-1:5;           %x(n)向量显示范围(添加了附加的零值)
>>nh=-2:10;          %h(n)向量显示范围(添加了附加的零值)
>>x=uDT(nx)-uDT(nx-4); %uDT(nx)产生单位阶跃序列的函数 第 7 页附录中有介绍
>>h=0.8.^nh.*(uDT(nh)-uDT(nh-8));
>>y=conv(x,h);
>>ny1=nx(1)+nh(1);    %卷积结果起始点
>>ny2=nx(end)+nh(end); %卷积结果终点
>>ny=ny1:ny2;
>>subplot(311)
>>stem(nx,x,'fill'),grid on
>>xlabel('n'),title('x(n)')
>>axis([-4 16 0 3])
```

```

>>subplot(312)
>>stem(nh,h,'fill'),grid on
>>xlabel('n'),title('h(n)')
>>axis([-4 16 0 3])
>>subplot(313)
>>stem(ny,y,'fill'),grid on
>>xlabel('n'),title('y(n)=x(n)*h(n)')
>>axis([-4 16 0 3])

```

程序运行结果如图 2-5 所示。

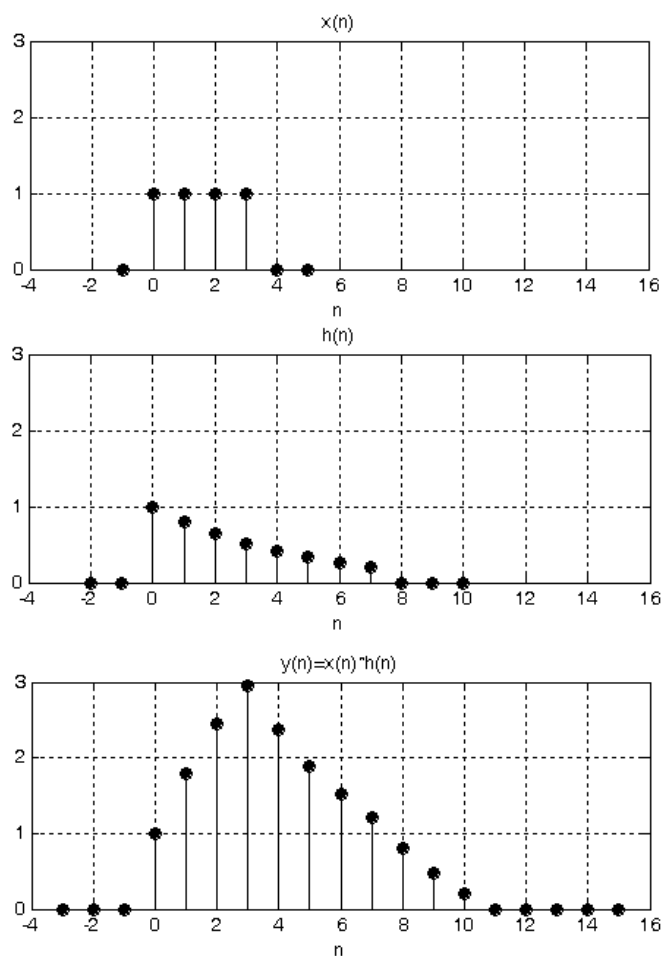


图 2-5 利用卷积和法求解系统的零状态响应

Matlab 中的二维卷积函数 conv2()

(1) $C = \text{conv2}(A, B)$

设矩阵 A 的大小为 $M_a \times N_a$ ，矩阵 B 的大小为卷积 $M_b \times N_b$ ，则卷积后 C 的大小为 $(M_a + M_b - 1) \times (N_a + N_b - 1)$ ；

(2) $C = \text{conv2}(Hcol, Hrow, A)$

矩阵 A 先与 Hcol 向量在列方向进行卷积，然后再与 Hrow 向量在行方向上进行卷积；

(3) $C = \text{conv2}(A, B, 'shape')$

当 $shape = full$ 时，返回全部二维卷积结果，即返回 C 的大小为 $(M_a + M_b - 1) \times (N_a + N_b - 1)$ ；

当 $shape = same$ 时，返回与 A 同样大小的卷积中心部分；

当 $shape = valid$ 时，不考虑边界补零，即只要有边界补出的零参与运算的都舍去，返回 C 的大小为 $(M_a + M_b - 1) \times (N_a + N_b - 1)$ ；

conv2 函数常用于对图像进行滤波处理。

三、实验内容

(1) 试用 Matlab 命令求解以下离散时间系统的单位取样响应，并判断系统的稳定性。

$$1) \quad 3y(n) + 4y(n-1) + y(n-2) = x(n) + x(n-1)$$

$$2) \quad \frac{5}{2}y(n) + 6y(n-1) + 10y(n-2) = x(n)$$

(2) 已知某系统的单位取样响应为 $h(n) = \left(\frac{7}{8}\right)^n [u(n) - u(n-10)]$ ，试用 MATLAB 求当激励信号为 $x(n) = u(n) - u(n-5)$ 时，系统的零状态响应。

(3) 用数字图像处理中的 Sobel 算子，调用 conv2 函数对 lena 图像进行滤波。设两个 3×3 的 Sobe 矩阵算子 $G_x = [-1 \ 0 \ 1; -2 \ 0 \ 2; -1 \ 0 \ 1]$ ， $G_y = [1 \ 2 \ 1; 0 \ 0 \ 0; -1 \ -2 \ -1]$ 。

1) 读取 lena 图像数据，保存到 B 矩阵中；

2) 分如下三种情况对 B 进行卷积滤波

① 采用 G_x 与 B 进行卷积；

② 采用 G_y 与的 B 进行卷积；

③ 先采用 G_x 与 B 进行卷积，然后再采用 G_y 与 B 进行卷积；

④ 在一个 figure 中，分四个子图显示出：原始图像，经过 G_x 卷积滤波后图像、经过 G_y 卷积滤波后图像，先后采用 G_x 和 G_y 与的 B 进行卷积滤波后

的图像，观察滤波后的图像的效果。

四、思考题

(1) Matlab 的工具箱函数 `conv`，能用于计算两个有限长序列之间的卷积，但 `conv` 函数假定这两个序列都从 $n=0$ 开始。试编写 M 文件计算 $x(n)=[3,11,7,0,-1,4,2]$, $-3 \leq n \leq 3$ 和 $h(n)=[2,3,0,-5,2,1]$, $-1 \leq n \leq 4$ 之间的卷积，并绘制 $y(n)$ 的波形图。

(2) 在数字图像处理边缘检测技术中，除了 Sobel 算子外，还有哪些边缘检测算子？通过查找资料，选择某种边缘检测算子，在 Matlab 平台上编程实现对 `lena` 或其他图像进行边缘检测，显示出原图和边缘检测后的图像。

附录：

1. 单位阶跃序列

单位阶跃序列 $u(n)$ 定义为

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

在 MATLAB 中，冲激序列可以通过编写 `uDT.m` 文件来实现，即

```
function y=uDT(n)
y=n>=0;           %当参数为非负时输出 1
```

调用该函数时 n 也同样必须为整数或整数向量。

【实例 4】 利用 MATLAB 的 `uDT` 函数绘出单位阶跃序列的波形图。

解：MATLAB 源程序为

```
>>n=-3:5;
>>x=uDT(n);
>>stem(n,x,'fill'),xlabel('n'),grid on
>>title('单位阶跃序列')
>>axis([-3 5 -0.1 1.1])
```

程序运行结果如图 2-6 所示。

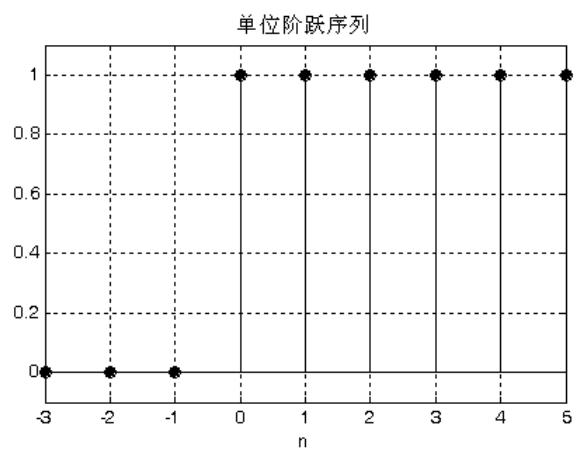


图 6 单位阶跃序列