

第3章

离散时间系统的响应

3.1 基础理论及相关 MATLAB 函数语法介绍

3.1.1 基础理论

1. 差分方程的时域求解

若离散时间系统差分方程为

$$\sum_{k=0}^N a_k y(n-k) = \sum_{r=0}^M b_r x(n-r) \quad (3-1)$$

先求得 N 个特征根 $C_k, k=1, 2, \dots, N$, 从而得到非重根时的齐次解:

$$y(n) = \sum_{k=1}^N C_k \alpha_k^n \quad (3-2)$$

和 L 次重根时的齐次解:

$$y(n) = \sum_{k=1}^L C_k n^{L-k} \alpha_k^n \quad (3-3)$$

然后求特解。对于自由项为 n^k 的多项式, 其特解为 $D_0 n^k + D_1 n^{k-1} + \dots + D_k$; 对于自由项含有 a^n 且 a 不是齐次根时, 则特解为 $D a^n$; 对于自由项含有 a^n 且 a 是一阶齐次根时, 则特解为 $(D_1 n + D_2) a^n$; 对于自由项含有 a^n 且 a 是 k 阶齐次根时, 则特解为 $(D_1 n^k + D_2 n^{k-1} + \dots + D_{k+1}) a^n$ 。

将特解代入差分方程求出待定系数 D_i , 代入系统的初始状态求出待定系数 C_i , 得到完全解, 完全解 = 齐次解 + 特解。

2. 单位样值响应和阶跃响应的定义

当离散时间系统输入为单位样值信号且系统的初始状态全部为零时的系统响应, 称为该系统的单位样值响应, 用 $h(n)$ 表示。当离散时间系统输入为阶跃信号且系统的初始状态全部为零时的系统响应, 称为该系统的阶跃响应, 用 $g(n)$ 表示。当离散时间系统具有线性时不变属性时满足如下关系:

$$h(n) = g(n) - g(n-1) \quad (3-4)$$

3.1.2 相关的 MATLAB 函数语法介绍

1. 卷积和的求解函数

作用 1: 多项式乘法。

语法介绍:

(1) $w = \text{conv}(u, v)$, 返回向量 u 和 v 的卷积。如果 u 和 v 是多项式系数的向量, 对

其卷积与将这两个多项式相乘等效。向量 u 和 v 可具有不同的长度或数据类型。如果 u 和 v 是离散时间信号,长度分别为 N 和 M ,则计算结果的长度为 $M+N-1$ 。当 u 或 v 的类型为 `single` 时,输出的类型为 `single`。否则, `conv` 会将输入类型转换为 `double`,并返回 `double` 类型。

(2) `w=conv(u,v,shape)`,返回如 `shape` 指定的卷积的分段。例如, `conv(u,v,'same')` 仅返回与 u 等大小的卷积的中心部分,而 `conv(u,v,'valid')` 仅返回计算的没有补零边缘的卷积部分。

例 3.1 展开多项式 $(s^2+2s+2)(s+4)(s+1)$ 。

```
w = conv([1,2,2],conv([1,4],[1,1]))
w =
     1     7    16    18     8
p = poly2str(w,'s')
p =
s^4 + 7 s^3 + 16 s^2 + 18 s + 8
```

作用 2: 卷积运算。

例 3.2 求一个随机向量与一个已知向量的卷积。

```
a = randn(1,8); b = [-2 3 -6];
conv(a,b)
ans =
-1.0753    -2.0548     6.7933   -19.5042    15.5021    -1.6014    -4.9685
 5.8601     3.6294    -2.0557
```

2. 互相关运算的求解函数

语法介绍:

(1) `r=xcorr(x,y)`,返回两个离散时间信号的互相关。互相关测量向量 x 和移位(滞后)副本向量 y 之间的相似性,形式为滞后的函数。如果 x 和 y 的长度不同,函数会在较短向量的末尾添加零,使其长度与另一个向量相同。

(2) `r=xcorr(x)`,返回 x 的自相关信号。如果 x 是矩阵,则 r 也是矩阵,其中包含 x 的所有列组合的自相关和互相关信号。

(3) `r=xcorr(___,maxlag)`,将上述任一语法中的滞后范围限制为 $-\text{maxlag} \sim \text{maxlag}$ 。

(4) `r=xcorr(___,scaleopt)`,为互相关或自相关指定归一化选项。除 `'none'`(默认值)以外的任何选项都要求 x 和 y 具有相同的长度。

(5) `[r,lags]=xcorr(___)`,返回用于计算相关性的滞后。

3. 初始条件求解函数

作用：计算等效初始条件的输入向量。

语法介绍： $Z=filtic(b,a,Y,X)$ ，其中 Y 和 X 是初始化条件向量。如果输入 $x(n)$ 是因果信号，则 $X=0$ 。若描述系统的差分方程为

$$a_1 y(n) = b_1 x(n) + \cdots + b_k x(n-k) - a_2 y(n-1) - \cdots - a_m y(n-m)$$

则 $b=[b_1 b_2 \cdots b_k]$ ， $a=[a_1 a_2 \cdots a_m]$ 。

4. 差分方程求解函数

作用：调用 `filter` 函数求解差分方程。

语法介绍： $y=filter(b,a,x)$ ，计算输入向量 $x(n)$ 的零状态响应输出信号 $y(n)$ ， b 和 a 的说明与 `filtic` 函数中的说明相同。

$y=filter(b,a,x,xi)$ ，计算全响应的函数。 xi 是等效初始条件的输入信号，由初始条件确定， xi 的长度必须等于 $\max(\text{length}(a), \text{length}(b))-1$ 。此时需要调用 `filtic` 函数： $xi=filtic(b,a,ys,xs)$ 。

其中 $ys=[y(-1), y(-2), \cdots, y(-n)]$ ， $xs=[x(-1), x(-2), \cdots, x(-n)]$ 是初始条件向量，另外若 $x(n)$ 为因果信号，也就是 $xs=[]$ 时， xs 可默认。

$[y,zf]=filter(___)$ ，返回滤波器延迟的最终条件 zf 。滤波器延迟的最终条件 zf ，是一个长度为 $\max(\text{length}(a), \text{length}(b))-1$ 的列向量。

如果要将 `filter` 函数与来自 FIR 滤波器的 b 系数结合使用，请将参数 a 设置为 1，使用 $y=filter(b,1,x)$ 。

例 3.3 求解差分方程。

若 $y(n)-0.8y(n-1)=x(n)$ ， $x(n)=u(n)$ ，初始条件为 $y(-1)=1$ 。

```
b = [1];
a = [1, -0.8];
x = ones(1,10);
Y = 1;
xic = filtic(b,a,Y);
y = filter(b,a,x,xic);
stem(0:length(y)-1,y,'filled');
title('Example 3.3');
xlabel('n');
ylabel('Amplitude');
grid on
```

运行程序，结果如图 3.1 所示。

例 3.4 求解差分方程。

已知 $y(n)=0.81y(n-2)+x(n)+x(n-1)$ ，初始条件为 $y(-1)=2, y(-2)=2$ ，

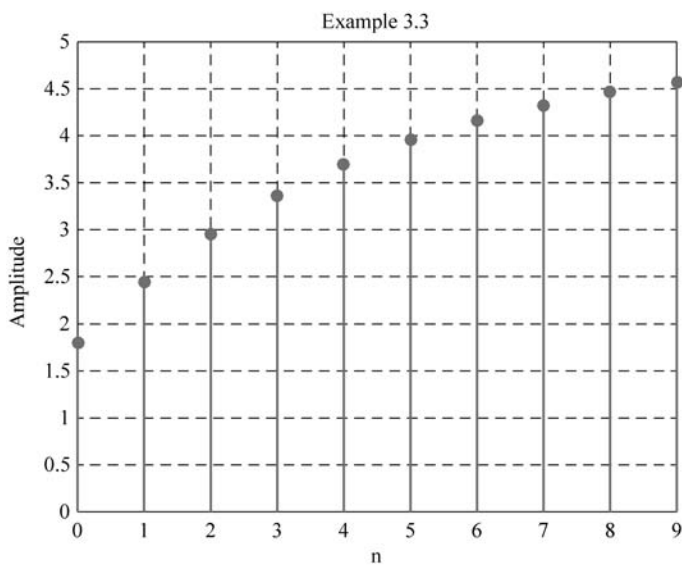


图 3.1 例 3.3 的运行结果

输入 $x(n) = 0.7^n u(n)$ 。

```
b = [1,1]; n = 0:20;
a = [1,0, -0.81];
Y = [2,2]; x = (0.7).^n;
xic = filtic(b,a,Y);
y = filter(b,a,x,xic);
stem(n,y,'filled');
title('Example 3.4');
xlabel('n');
grid on;
ylabel('Amplitude');
```

运行程序,结果如图 3.2 所示。

5. 单位样值响应和阶跃响应求解函数

impz()函数作用: 求解差分方程所表示的离散时间系统的单位样值响应。

stepz()函数作用: 求解差分方程所表示的离散时间系统的阶跃响应。

语法介绍: impz(b,a)绘制离散时间系统的单位样值响应,stepz(b,a)绘制离散时间系统的阶跃响应。b和a的说明与filtic()函数中的说明相同。

impz(sos)绘制以二阶矩阵参数形式描述的离散时间系统的单位样值响应,stepz(sos)绘制以二阶矩阵参数形式描述的离散时间系统的阶跃响应。

[h,t]= impz(____)返回单位样值响应h和数字滤波器的相应采样时间t,[h,t]= stepz(____)返回阶跃响应h和数字滤波器的相应采样时间t。

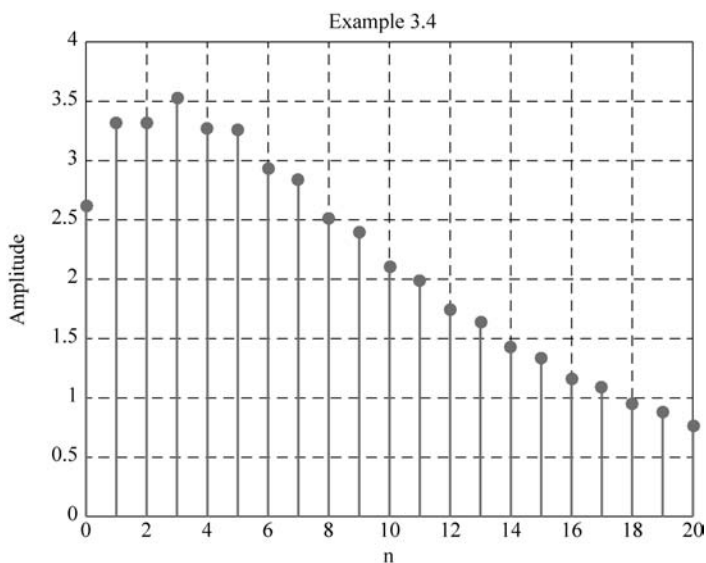


图 3.2 例 3.4 的运行结果

3.1.3 模拟工具 Simulink

Simulink 是 MATLAB 中的一种可视化仿真工具,是一种基于 MATLAB 的框图设计环境,是实现动态系统建模、仿真和分析的一个软件包,被广泛应用于线性系统、非线性系统、数字控制及离散时间信号处理的建模和仿真中。Simulink 提供一个动态系统建模、仿真和综合分析的集成环境。在该环境中,无须书写复杂的程序,而只需要通过简单直观的鼠标操作,就可构造出复杂的仿真系统。Simulink 与 MATLAB 相集成,能够在 Simulink 中将 MATLAB 算法融入模型,还能将仿真结果导出至 MATLAB 做进一步分析。

Simulink 可以用连续采样时间、离散采样时间或两种混合的采样时间进行建模,它也支持多速率系统,即系统中的不同部分具有不同的采样速率。为了创建动态系统模型,Simulink 提供了一个建立模型方块图的图形用户接口,这个创建过程只需单击和拖动鼠标就能完成,它提供了一种更快捷、直接明了的方式,而且用户可以立即看到系统的仿真结果。Simulink 工具界面如图 3.3 所示。

要在 Simulink 环境中进行信号处理系统建模,需要安装 DSPSystemToolbox 软件。DSPSystemToolbox 为信号处理系统的设计和仿真提供算法和工具。这些功能以 MATLAB 函数、MATLAB System object 和 Simulink 模块的形式提供。该系统工具箱包括专用 FIR 和 IIR 滤波器、FFT、多速率处理的设计方法,以及处理流数据和创建实时原型的 DSP 方法。可以设计自适应和多速率滤波器,使用计算效率高的架构实现滤波器,以及对浮点数字滤波器进行仿真。用于文件和设备的信号输入和输出、信号生成、频谱分析和交互式可视化的工具使用户能够分析系统行为和性能。对于快速原型和嵌入

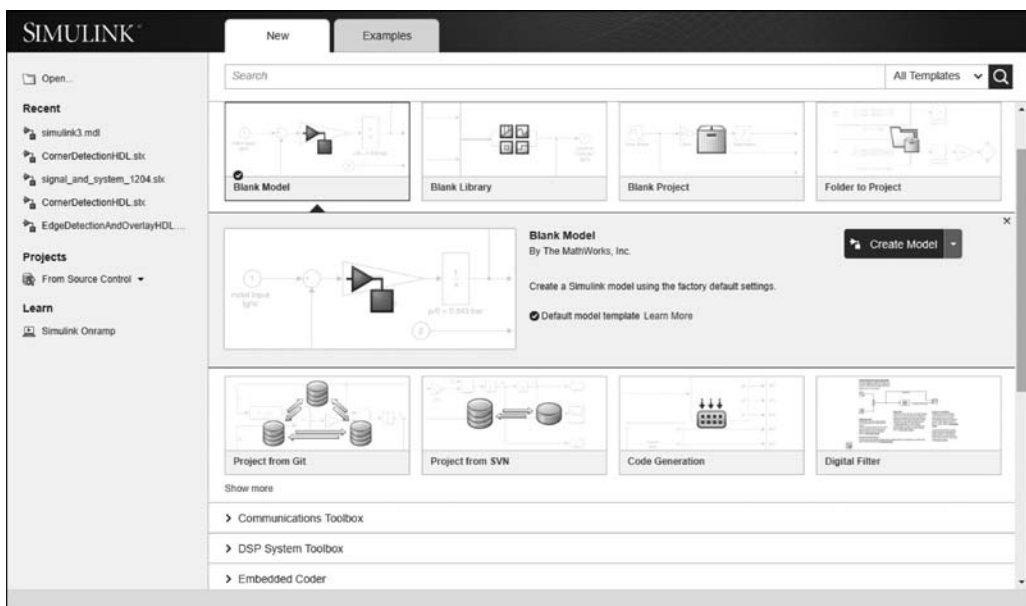


图 3.3 Simulink 工具界面

式系统设计,该系统工具箱支持定点算术和 C 或 HDL 代码生成。

3.2 实验示例

例 3.5 求解差分方程。

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

若 $x(n)=u(n)$, 初始条件为 $x(-1)=1, x(-2)=-1, y(-1)=-1, y(-2)=1$ 。

```
b = [1, -1, 1, -1];
a = [1, 2, 1];
x0 = [1, -1, 0];
y0 = [-1, 1];
xic = filtic(b, a, y0, x0);
N = 20;
n = 0:N-1;
xn = ones(1, N);
yn = filter(b, a, xn, xic);
stem(n, yn);
title('Example 3.5');
xlabel('n');
ylabel('Amplitude');
```

运行程序,结果如图 3.4 所示。

例 3.6 用 Simulink 工具仿真绘制如下离散系统的响应图,时间为 $0 \sim 100\text{ms}$ 。输

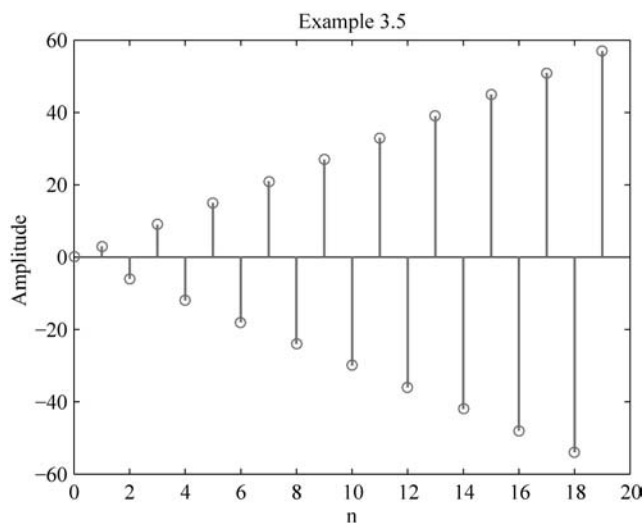


图 3.4 例 3.5 的运行结果

入信号 $x(t) = \sin(200\pi t)$, 采样频率为 1000Hz, 系统的差分方程为

$$y(n] - y(n-1] - y(n-2] = x(n]$$

启动 Simulink 工具, 创建一个空白的工程。在 View 菜单栏中打开 Library Browser (见图 3.5)。每个模块都可以单独设置属性, 在模块拖入模型窗口后在模块上双击即可。在模块上右击, 选择 Format Flip Block 可以翻转模块的方向。

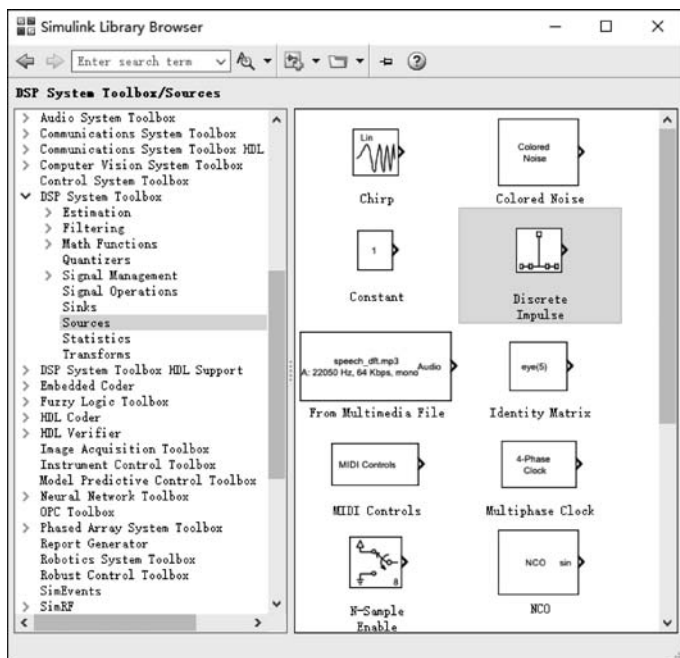


图 3.5 Library Browser 界面

常见的单元在如下位置可以找到。

样值信号：DSP System Toolbox→Sources→Discrete Impulse

阶跃信号：Simulink→Sources→Step

随机信号：Simulink→Sources→Random Number

其他信号：DSP System Toolbox→Sources→Signal from Workspace

延迟单元：Simulink→Discrete→Delay

加法单元：Simulink→Math Operations→Add

信号增益：Simulink→Math Operations→Gain

系统函数：Simulink→Discrete→Discrete Transfer Fcn

示波器：Simulink→Sinks→Scope

停止仿真：Simulink→Sinks→Stop Simulation

保存仿真数据到文件：Simulink→Sinks→To File

保存仿真数据到 MATLAB：Simulink→Sinks→To Workspace

搭建和差分方程对应的系统模型如图 3.6 所示，仿真模型图可以通过 Edit→Copy Current View to Clipboard 方式保存为矢量图。也可以用 saveas() 函数保存为指定格式的图片(如 eps、tif、bmp、pdf、png 等格式)。

```
h = get_param(gcs, 'handle');
saveas(h, 'filename.ext', format);
```

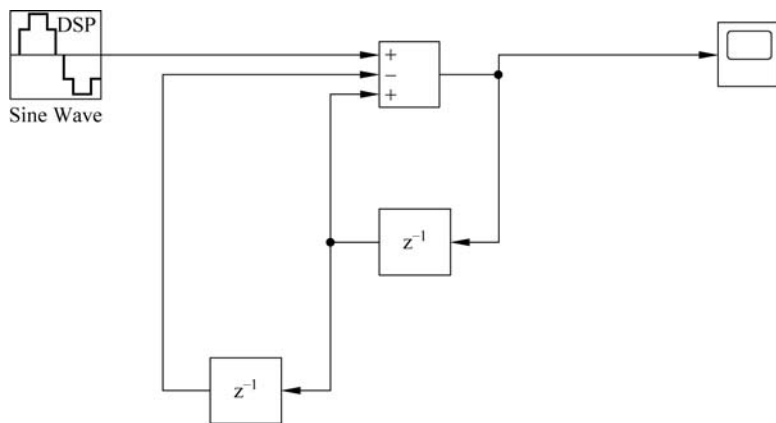


图 3.6 差分方程对应的系统框图(分立元件)

设定运行时间范围后,选择合适的显示方式,包括曲线颜色、曲线类型、坐标轴颜色、绘图背景颜色等(见图 3.7),即可得到系统的运行结果(见图 3.8)。

还可以直接计算出系统函数,采用如图 3.9 所示的系统模型进行仿真计算。设定运行时间范围后,同样得到如图 3.8 所示的系统运行结果。

为了验证仿真结果,编写如下 MATLAB 代码计算并绘制响应图,结果如图 3.10 所示。


```

a = [1 -1 1]; b = [1]; n = 0:100; fs = 1000;
x = sin(200 * pi * n/fs); y = filter(b,a,x);
stem(y); xlabel('n'); ylabel('value');
title('Example 3.6'); axis([0,102, -2.5,2.5]);

```

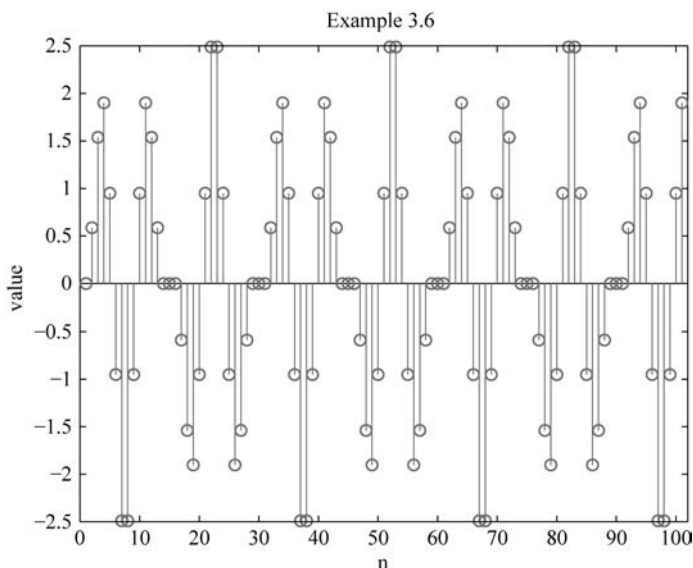
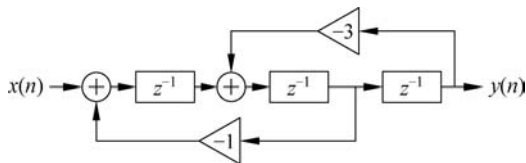


图 3.10 MATLAB 代码计算出来的系统响应图

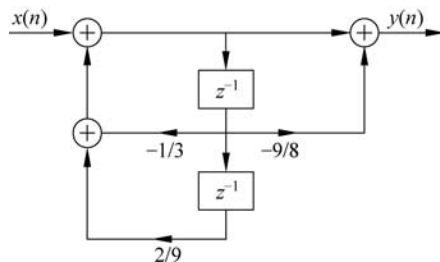
3.3 练习题

3.1 对题图 3.1 中采用框图表示的离散时间系统,分别采用数值仿真和 Simulink 图形仿真,计算并绘制其单位样值响应和阶跃响应前 30 个数值。

3.2 写出题图 3.2 中信号流图表示的离散时间系统的差分方程,使用 `impz()` 函数计算系统的单位样值响应 $h(n)$,然后绘制 $h(n)$ 从 $n=0$ 到 $n=50$ 的结果;使用 `stepz()` 函数计算系统的单位阶跃响应 $g(n)$,然后绘制 $g(n)$ 从 $n=0$ 到 $n=50$ 的结果;使用 `filter()` 函数计算并绘制 $g(n)$ 从 $n=0$ 到 $n=50$ 的结果;使用 Simulink 仿真工具绘制 $g(n)$ 从 $n=0$ 到 $n=50$ 的结果。



题图 3.1



题图 3.2