差分方程的解法分析及 MATLAB 实现（程序）

摘自：张登奇,彭仕玉.差分方程的解法分析及其 MATLAB 实现[J]. 湖南理工学院学报.2014(03)

引言

线性常系数差分方程是描述线性时不变离散时间系统的数学模型,求解差分方程是分析离散时间系统的重要内容.在《信号与系统》课程中介绍的求解方法主要有迭代法、时域经典法、双零法和变换域法[1].

1 迭代法

例 1 已知离散系统的差分方程为 $y(n)-\frac{3}{4}y(n-1)+\frac{1}{8}y(n-2)=x(n)+\frac{1}{3}x(n-1)$,激励信号为 $x(n)=(\frac{3}{4})^n u(n)$,初始状态为 $y(-1)=4,\ \ y(-2)=12$.求系统响应.

根据激励信号和初始状态,手工依次迭代可算出 $y(0)=\frac{5}{2}, y(1)=\frac{59}{24}$.

利用 MATLAB 中的 filter 函数实现迭代过程的 m 程序如下:

```
clc;clear;format compact;
a=[1,-3/4,1/8],b=[1,1/3,0], %输入差分方程系数向量,不足补 0 对齐
n=0:10;xn=(3/4).^n, %输入激励信号
zx=[0,0],zy=[4,12], %输入初始状态
zi=filtic(b,a,zy,zx),%计算等效初始条件
[yn,zf]=filter(b,a,xn,zi),%迭代计算输出和后段等效初始条件
```

2 时域经典法

用时域经典法求解差分方程:先求齐次解；再将激励信号代入方程右端化简得自由项,根据自由项形式求特解；然后根据边界条件求完全解[3].用时域经典法求解例 1 的基本步骤如下.

（1)求齐次解.特征方程为 $\alpha^2-\frac{3}{4}\alpha+\frac{1}{8}=0$,可算出 $\alpha_1=\frac{1}{2}$, $\alpha_2=\frac{1}{4}$.高阶特征根可用 MATLAB 的 roots 函数计算.齐次解为 $y_h(n)=C_1(\frac{1}{2})^n+C_2(\frac{1}{4})^n$, $n\geq 0$.

（2）求方程的特解.将 $x(n)=(\frac{3}{4})^n u(n)$ 代入差分方程右端得自由项为

$$(\frac{3}{4})^n u(n)+\frac{1}{3}\cdot(\frac{3}{4})^{n-1}u(n-1)=\begin{cases}1, & n=0\\ \frac{13}{9}\cdot(\frac{3}{4})^n, & n\geq 1\end{cases}$$

当 $n\geq 1$ 时,特解可设为 $y_p(n)=D(\frac{3}{4})^n$,代入差分方程求得 $D=\frac{13}{2}$.

（3）利用边界条件求完全解.当 n=0 时迭代求出 $y(0)=\frac{5}{2}$,当 $n\geqslant 1$ 时,完全解的形式为 $y(n)=C_1(\frac{1}{2})^n+C_2(\frac{1}{4})^n+\frac{13}{2}\cdot(\frac{3}{4})^n$, 选择求完全解系数的边界条件可参考文[4]选 $y(0), y(-1)$.根据边界条件求得 $C_1=-\frac{17}{3}, C_2=\frac{5}{3}$.注意完全解的表达式只适于特解成立的 n 取值范围,其他点要用 $\delta(n)$ 及其延迟表示,如果其值符合表达式则可合并处理.差分方程的完全解为

$$y(n)=\frac{5}{2}\delta(n)+[-\frac{17}{3}\cdot(\frac{1}{2})^n+\frac{5}{3}\cdot(\frac{1}{4})^n+\frac{13}{2}\cdot(\frac{3}{4})^n]u(n-1)=[-\frac{17}{3}\cdot(\frac{1}{2})^n+\frac{5}{3}\cdot(\frac{1}{4})^n+\frac{13}{2}\cdot(\frac{3}{4})^n]u(n)$$

MATLAB没有专用的差分方程求解函数,但可调用maple符号运算工具箱中的rsolve函数实现[5],格式为y=maple('rsolve({equs, inis},y(n))'),其中:equs为差分方程表达式, inis为边界条件,y(n)为差分方程中的输出函数式.rsolve的其他格式可通过mhelp rsolve命令了解.在MATLAB中用时域经典法求解例1中的全响应和单位样值响应的程序如下.

```
clc;clear;format compact;
yn=maple('rsolve({y(n)-3/4*y(n-1)+1/8*y(n-2)=(3/4)^n+1/3*(3/4)^(n-1),y(0)=5/2,y(-1)=4},y(n))'),
hn=maple('rsolve({y(n)-3/4*y(n-1)+1/8*y(n-2)=0,y(0)=1,y(1)=13/12},y(n))'),
```

3 双零法

根据双零响应的定义,按时域经典法的求解步骤可分别求出零输入响应和零状态响应.理解了双零法的求解原理和步骤,实际计算可调用rsolve函数实现.

```
yzi=maple('rsolve({y(n)-3/4*y(n-1)+1/8*y(n-2)=0,y(-1)=4, y(-2)=12},y(n))'),
yzs=maple('rsolve({y(n)-3/4*y(n-1)+1/8*y(n-2)=(3/4)^n+1/3*(3/4)^(n-1),y(0)=1,
y(-1)=0},y(n))'),
```

4 变换域法

设差分方程的一般形式为$\sum_{k=0}^{N} a_k y(n-k)=\sum_{r=0}^{M} b_r x(n-r)$.

对差分方程两边取单边 z 变换,并利用 z 变换的位移公式得

$$\sum_{k=0}^{N} a_k z^{-k}[Y(z)+\sum_{l=-k}^{-1} y(l)z^{-l}]=\sum_{r=0}^{M} b_r z^{-r}[X(z)+\sum_{m=-r}^{-1} x(m)z^{-m}]$$

整理成$A(z)Y(z)+Y_0(z)=B(z)X(z)+X_0(z)$形式有

$$A(z)=a_0+a_1z^{-1}+\cdots+a_Nz^{-N}\,,B(z)=b_0+b_1z^{-1}+\cdots+b_Mz^{-M}\,.$$

$$Y_0(s)=\sum_{k=1}^{N}\sum_{l=-k}^{-1}a_k y(l)z^{-k-l}\,,X_0(s)=\sum_{r=1}^{M}\sum_{m=-r}^{-1}b_r x(m)z^{-r-m}\,.$$

可以看出,由差分方程可直接写出$A(z)$ 和$B(z)$,系统函数$H(z)=B(z)/A(z)$,将系统函数进行逆z变换可得单位样值响应.由差分方程的初始状态可算出$Y_0(z)$,由激励信号的初始状态可算出$X_0(z)$,将激励信号进行z变换可得$X(z)$,求解z域代数方程可得输出信号的象函数

$$Y(z)=\frac{B(z)X(z)+X_0(z)-Y_0(z)}{A(z)},$$

对输出象函数进行逆 z 变换可得输出信号的原函数$y(n)$.利用 z 变换求解差分方程各响应的步骤可归纳如下:

(1) 根据差分方程直接写出$A(z)$ 、$B(z)$ 和$H(z)$, $H(z)$的逆变换即为单位样值响应;

(2) 根据激励信号算出$X(z)$,如激励不是因果序列则还要算出前 M 个初始状态值;

(3) 根据差分方程的初始状态 $y(-1),y(-2),\cdots,y(-N)$ 和激励信号的初始状态 $x(-1),x(-2),\cdots,x(-M)$ 算出$Y_0(z)$ 和$X_0(z)$;

(4)在z域求解代数方程$A(z)Y(z)+Y_0(z)=B(z)X(z)+X_0(z)$得输出象函数$Y(z)$, $Y(z)$ 的逆变换即为全响应;

(5)分析响应象函数的极点来源及在z平面中的位置,确定自由响应与强迫响应,或瞬态响应与稳态

响应;

（6）根据零输入响应和零状态响应的定义,在z域求解双零响应的象函数,对双零响应的象函数进行逆z变换,得零输入响应和零状态响应.

用变换域法求解例 1 的基本过程如下.

根据差分方程直接写出$A(z)=1-\frac{3}{4}z^{-1}+\frac{1}{8}z^{-2}$,$B(z)=1+\frac{1}{3}z^{-1}$.系统函数的极点为$\frac{1}{2},\frac{1}{4}$.

对激励信号进行 z 变换得$X(z)=z/(z-\frac{3}{4})$.激励象函数的极点为 3/4.

根据差分方程的初始状态算出$Y_0(z)=-\frac{3}{2}+\frac{1}{2}z^{-1}$.根据激励信号的初始状态算出$X_0(z)=0$.

对 z 域代数方程求解,得全响应的象函数$Y(z)=(\frac{5}{2}z^3-\frac{31}{24}z^2+\frac{3}{8}z)/(z^3-\frac{3}{2}z^2+\frac{11}{16}z-\frac{3}{32})$.

进行逆 z 变换得全响应为$y(n)=[-\frac{17}{3}\cdot(\frac{1}{2})^n+\frac{5}{3}\cdot(\frac{1}{4})^n+\frac{13}{2}\cdot(\frac{3}{4})^n]u(n)$

其中,与系统函数的极点对应的是自由响应;与激励象函数的极点对应的是强迫响应. $Y(z)$ 的极点都在 z 平面的单位圆内故都是瞬态响应.零输入响应和零状态响应可按定义参照求解.

上述求解过程可借助 MATLAB 的符号运算编程实现.实现变换域法求解差分方程的 m 程序如下:

```
clc;clear;format compact;
syms z n %定义符号对象
% 输入差分方程、初始状态和激励信号%
a=[1,-3/4,1/8],b=[1,1/3], %输入差分方程系数向量
y0=[4,12],x0=[0], %输入初始状态,长度分别比 a、b 短 1,长度为 0 时用[]
xn=(3/4)^n, %输入激励信号,自动单边处理,u(n)可用 1^n 表示
% 下面是变换域法求解差分方程的通用程序，极点为有理数时有解析式输出 %
N=length(a)-1;M=length(b)-1;%计算长度
Az=poly2sym(a,'z')/z^N;Bz=poly2sym(b,'z')/z^M;%计算 A(z)和 B(z)
Hz=Bz/Az;disp('系统函数 H(z):'),sys=filt(b,a),%计算并显示系统函数
hn=iztrans(Hz);disp('单位样值响应 h(n)='),pretty(hn),%计算并显示单位样值响应
Hzp=roots(a);disp('系统极点:');Hzp,%计算并显示系统极点
Xz=ztrans(xn);disp('激励象函数 X(z)='),pretty(Xz),%激励信号的单边 z 变换
Y0z=0;%初始化 Y0(z),求 Y0(z)注意系数标号与变量下标的关系
for k=1:N;
    for l=-k:-1;Y0z = Y0z+a(k+1)*y0(-l)*z^(-k-l);end
end
disp('初始 Y0(z)'),Y0z,%系统初始状态的 z 变换
X0z=0;%初始化 X0(z),求 X0(z)注意系数标号与变量下标的关系
for r=1:M;
    for m=-r:-1;X0z = X0z+b(r+1)*x0(-m)*z^(-r-m);end
end
disp('初始 X0(z)'),X0z,%激励信号起始状态的 z 变换
Yz=(Bz*Xz+X0z-Y0z)/Az;disp('全响应的 z 变换 Y(z)'),pretty(simple(Yz)),
yn=iztrans(Yz);disp('全响应 y(n)='),pretty(yn),% 计算并显示全响应
Yziz=-Y0z/Az;disp('零输入象函数 Yzi(z)='),pretty(Yziz),%零激励响应的 z 变换
yzin=iztrans(Yziz);disp('零输入响应 yzi(n)='),pretty(yzin),% 计算并显示零输入响应
Yzsz=(Bz*Xz+X0z)/Az;disp('零状态象函数 Yzs(z)='),pretty(Yzsz),%零状态响应的 z 变换
```

yzsn=iztrans(Yzsz);disp('零状态响应 yzs(n)='),pretty(yzsn),% 计算并显示零状态响应

该程序的运行过程与手算过程对应,显示在命令窗的运行结果与手算结果相同.