第 3 章程序及注释

例 3.3　考虑仿真对象

$$z(k)-1.5z(k-1)+0.7z(k-2)=u(k-1)+0.5u(k-2)+v(k) \qquad (3.41)$$

其中，$v(k)$ 是服从正态分布的白噪声 $\mathbf{N}(0,1)$。输入信号采用 4 阶 M 序列，幅度为 1。选择如下形式的辨识模型

$$z(k)+a_1z(k-1)+a_2z(k-2)=b_1u(k-1)+b_2u(k-2)+v(k) \qquad (3.42)$$

设输入信号的取值是从 k =1 到 k =16 的 M 序列，则待辨识参数 $\hat{\theta}_{LS}$ 为 $\hat{\theta}_{LS}=(\boldsymbol{H}_L^{\tau}\boldsymbol{H}_L)^{-1}\boldsymbol{H}_L^{\tau}z_L$。其中，被辨识参数 $\hat{\theta}_{LS}$、观测矩阵 $\boldsymbol{z}_L$、$\boldsymbol{H}_L$ 的表达式为

$$\hat{\boldsymbol{\theta}}_{LS}=\begin{bmatrix}a_1\\a_2\\b_1\\b_2\end{bmatrix},\quad \boldsymbol{z}_L=\begin{bmatrix}z(3)\\z(4)\\\cdots\\z(16)\end{bmatrix},\quad \boldsymbol{H}_L=\begin{bmatrix}-z(2) & -z(1) & u(2) & u(1)\\-z(3) & -z(2) & u(3) & u(2)\\ & \cdots & & \cdots\\-z(15) & -z(14) & u(15) & u(14)\end{bmatrix} \qquad (3.43)$$

程序框图如图 3.2 所示。Matlab6.0 仿真程序如下：

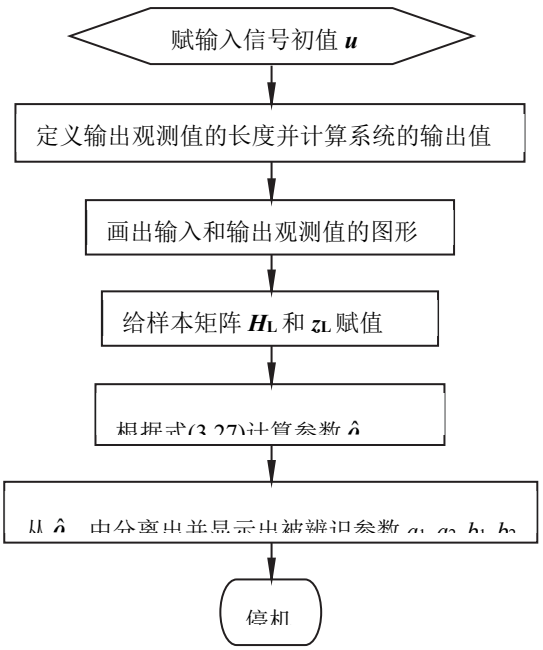

图 3.2　最小二乘一次完成算法程序框图

```
%二阶系统的最小二乘一次完成算法辨识程序，在光盘中的文件名：FLch3LSeg1.m
u=[-1,1,-1,1,1,1,1,1,-1,-1,-1,1,-1,-1,1,1]; %系统辨识的输入信号为一个周期的 M 序列
z=zeros(1,16); %定义输出观测值的长度
for k=3:16
    z(k)=1.5*z(k-1)-0.7*z(k-2)+u(k-1)+0.5*u(k-2); %用理想输出值作为观测值
end
subplot(3,1,1) %画三行一列图形窗口中的第一个图形
stem(u) %画输入信号 u 的径线图形
subplot(3,1,2) %画三行一列图形窗口中的第二个图形
i=1:1:16; %横坐标范围是 1 到 16，步长为 1
plot(i,z) %图形的横坐标是采样时刻 i，纵坐标是输出观测值 z，图形格式为连续曲线
subplot(3,1,3) %画三行一列图形窗口中的第三个图形
stem(z),grid on %画出输出观测值 z 的径线图形，并显示坐标网格
u,z %显示输入信号和输出观测信号
%L=14 %数据长度
HL=[-z(2) -z(1) u(2) u(1);-z(3) -z(2) u(3) u(2);-z(4) -z(3) u(4) u(3);-z(5) -z(4) u(5) u(4);-z(6) -z(5) u(6) u(5);-z(7) -z(6) u(7) u(6);-z(8) -z(7) u(8) u(7);-z(9) -z(8) u(9) u(8);-z(10) -z(9) u(10) u(9);-z(11) -z(10) u(11)
```

u(10);-z(12) -z(11) u(12) u(11);-z(13) -z(12) u(13) u(12);-z(14) -z(13) u(14) u(13);-z(15) -z(14) u(15) u(14)] %给样本矩阵 $\boldsymbol{H}_{\mathrm{L}}$ 赋值

ZL=[z(3);z(4);z(5);z(6);z(7);z(8);z(9);z(10);z(11);z(12);z(13);z(14);z(15); z(16)] % 给样本矩阵 $\boldsymbol{z}_{\mathrm{L}}$ 赋值

%Calculating Parameters

c1=HL'*HL; c2=inv(c1); c3=HL'*ZL; c=c2*c3 %计算并显示 $\hat{\theta}_{LS}$

%Display Parameters

a1=c(1), a2=c(2), b1=c(3),b2=c(4) %从 $\hat{\theta}_{LS}$ 中分离出并显示 a_1 、a_2、b_1、 b_2

%End

程序运行结果：

>>

u =[-1， 1，-1， 1， 1， 1， 1，-1，-1，-1， 1，-1，-1， 1， 1]

z =[0， 0，0.5000，0.2500，0.5250，2.1125， 4.3012，6.4731，6.1988，3.2670，-0.9386， -3.1949，-4.6352，6.2165，-5.5800，-2.5185]

HL =

0	0	1.0000	-1.0000
-0.5000	0	-1.0000	1.0000
-0.2500	-0.5000	1.0000	-1.0000
-0.5250	-0.2500	1.0000	1.0000
-2.1125	-0.5250	1.0000	1.0000
-4.3012	-2.1125	1.0000	1.0000
-6.4731	-4.3012	-1.0000	1.0000
-6.1988	-6.4731	-1.0000	-1.0000
-3.2670	-6.1988	-1.0000	-1.0000
0.9386	-3.2670	1.0000	-1.0000
3.1949	0.9386	-1.0000	1.0000
4.6352	3.1949	-1.0000	-1.0000
6.2165	4.6352	1.0000	-1.0000
5.5800	6.2165	1.0000	1.0000

ZL =[0.5000，0.2500，0.5250，2.1125，4.3012，6.4731，6.1988，3.2670，-0.9386，-3.1949， -4.6352，-6.2165，-5.5800，-2.5185]^T

c =[-1.5000，0.7000，1.0000，0.5000]^T

a1 = -1.5000

a2 = 0.7000

b1 = 1.0000

b2 =0.5000

>>

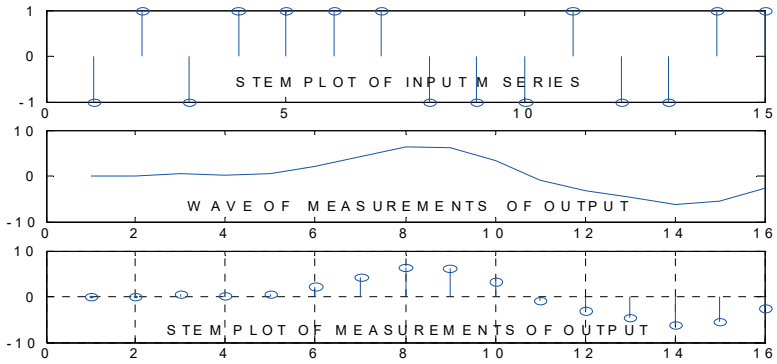

图 3.3　最小二乘一次完成算法仿真实例中输入信号和输出观测值

从仿真结果表 3.1 可以看出，由于所用的输出观测值没有任何噪声成分，所以辨识结果也无任何误差。

表 3.1　最小二乘一次完成算法的辨识结果

参　数	a_1	a_2	b_1	b_2
真　值	-1.5	0.7	1.0	0.5
估计值	-1.5	0.7	1.0	0.5

例 3.4　根据热力学原理，对给定质量的气体，体积 V 与压力 P 之间的关系为 $PV^{\alpha}=\beta$，其中 α 和 β 为待定参数。经实验获得如下一批数据，　V 的单位为立方英寸，P 的单位为帕每平方英寸。

V　　54.3　　61.8　　72.4　　88.7　　118.6　　194.0

P　　61.2　　49.5　　37.6　　28.4　　19.2　　10.1

试用最小二乘一次完成算法确定参数 α 和 β。

首先要写出系统的最小二乘表达式。为此，把体积 V 与压力 P

之间的关系 $PV^{\alpha}=\beta$ 改为对数关系，即， $\log P=-\alpha\log V+\log\beta$ 。此式与式(3.14)， $z(k)=h^{\tau}(k)\theta+e(k)$ ，对比可得： $z(k)=\log P$ ， $h^{\tau}(k)=[-\log V \quad 1]$ ，$\theta=[\alpha \quad \log\beta]^{\tau}$ 。

例 3.4 的 Matlab6.0 程序如下。

%实际压力系统的最小二乘辨识程序，在光盘中的文件名：

FLch3LSeg2.m

clear %工作间清零

V=[54.3,61.8,72.4,88.7,118.6,194.0]'，P=[61.2,49.5,37.6,28.4,19.2,10.1]' %赋初值并显示 V、P

%logP=-alpha*logV+logbeita=[-logV,1][alpha,log(beita)]'=HL*sita %注释 P、V 之间的关系

for i=1:6; Z(i)=log(P(i)); %循环变量的取值为从 1 到 6，系统的采样输出赋值

End %循环结束

ZL=Z' % z_L 赋值

HL=[-log(V(1)),1;-log(V(2)),1;-log(V(3)),1;-log(V(4)),1;-log(V(5)),1;-log(V(6)),1] %H_L 赋值

%Calculating Parameters

c1=HL'*HL; c2=inv(c1); c3=HL'*ZL; c4=c2*c3 %计算被辨识参数的值

%Separation of Parameters

alpha=c4(1) %α为 c4 的第一个元素

beita=exp(c4(2)) %β为以自然数为底的 c4 的第二个元素的指数

程序运行结果：

V = [54.3000, 61.8000, 72.4000, 88.7000, 118.6000, 194.0000] $^{\tau}$

P = [61.2000, 49.5000, 37.6000, 28.4000, 19.2000, 10.1000]$^{\tau}$

ZL = [4.1141, 3.9020, 3.6270, 3.3464, 2.9549, 2.3125]$^{\tau}$

HL =

$$\begin{pmatrix} -3.9945 & 1.0000 \\ -4.1239 & 1.0000 \\ -4.2822 & 1.0000 \\ -4.4853 & 1.0000 \\ -4.7758 & 1.0000 \\ -5.2679 & 1.0000 \end{pmatrix}$$

c4 =

$$\begin{bmatrix} 1.4042 \\ 9.6786 \end{bmatrix}$$

alpha = 1.4042

beita = 1.5972e+004

>>

仿真结果表明，用最小二乘一次完成算法可以迅速辨识出系统参数，即 α=1.4042， β=1.5972e+004。

例 3.5 考虑图 3.6 所示的仿真对象，图中， $v(k)$ 是服从 $\mathbf{N}(0,1)$ 分布的不相关随机噪声。且

$$G(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} \quad , \quad N(z^{-1}) = \frac{D(z^{-1})}{C(z^{-1})} \quad ,$$

$$\begin{cases} A(z^{-1}) = 1 - 1.5a_1 z^{-1} + 0.7z^{-2} = C(z^{-1}) \\ B(z^{-1}) = 1.0z^{-1} + 0.5z^{-2} \\ D(z^{-1}) = 1 \end{cases}$$

选择图 3.6 所示的辨识模型。仿真对象选择如下的模型结构

$$z(k) + a_1 z(k-1) + a_2 z(k-2) = b_1 u(k-1) + b_2 u(k-2) + v(k)$$

(3.68)

其中， $v(k)$ 是服从正态分布的白噪声 $\mathbf{N}(0,1)$ 。输入信号采用 4 位移位

寄存器产生的 M 序

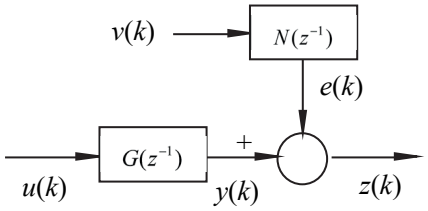

图 3.6 最小二乘递推算法辨识实例结构图

列，幅度为 0.03。按式

$$z(k)-1.5z(k-1)+0.7z(k-2)=u(k-1)+0.5u(k-2)+v(k) \tag{3.69}$$

构造 $\boldsymbol{h}(k)$；加权阵取单位阵 $\boldsymbol{\Lambda}_L=\boldsymbol{I}$；利用式(3.61)计算 $\boldsymbol{K}(k)$、$\hat{\theta}(k)$ 和 $\boldsymbol{P}(k)$，计算各次参数辨识的相对误差，精度满足要求式(3.67)后停机。

最小二乘递推算法辨识的 Malab6.0 程序流程如图 3.7 所示。下面给出具体程序。

%最小二乘递推算法辨识程序， 在光盘中的文件名：

FLch3RLSeg3.m

```
%FLch3RLSeg3
clear %清理工作间变量
L=15; % M 序列的周期
y1=1;y2=1;y3=1;y4=0; %四个移位寄存器的输出初始值
for i=1:L;%开始循环，长度为 L
    x1=xor(y3,y4); %第一个移位寄存器的输入是第三个与第四个移位寄存器的输出的“或”
    x2=y1; %第二个移位寄存器的输入是第一个移位寄存器的输出
```

```
    x3=y2; %第三个移位寄存器的输入是第二个移位寄存器的输出
    x4=y3; %第四个移位寄存器的输入是第三个移位寄存器的输出
    y(i)=y4; %取出第四个移位寄存器的幅值为"0"和"1"的输出信号，即 M 序列
    if y(i)>0.5,u(i)=-0.03; %如果 M 序列的值为"1", 辨识的输入信号取“-0.03”
    else u(i)=0.03; %如果 M 序列的值为"0", 辨识的输入信号取“0.03”
    end %小循环结束
    y1=x1;y2=x2;y3=x3;y4=x4; %为下一次的输入信号做准备
end %大循环结束，产生输入信号 u
figure(1); %第一个图形
stem(u),grid on %显示出输入信号径线图并给图形加上网格
z(2)=0;z(1)=0; %设 z 的前两个初始值为零
for k=3:15; %循环变量从 3 到 15
   z(k)=1.5*z(k-1)-0.7*z(k-2)+u(k-1)+0.5*u(k-2); %输出采样信号
end
```

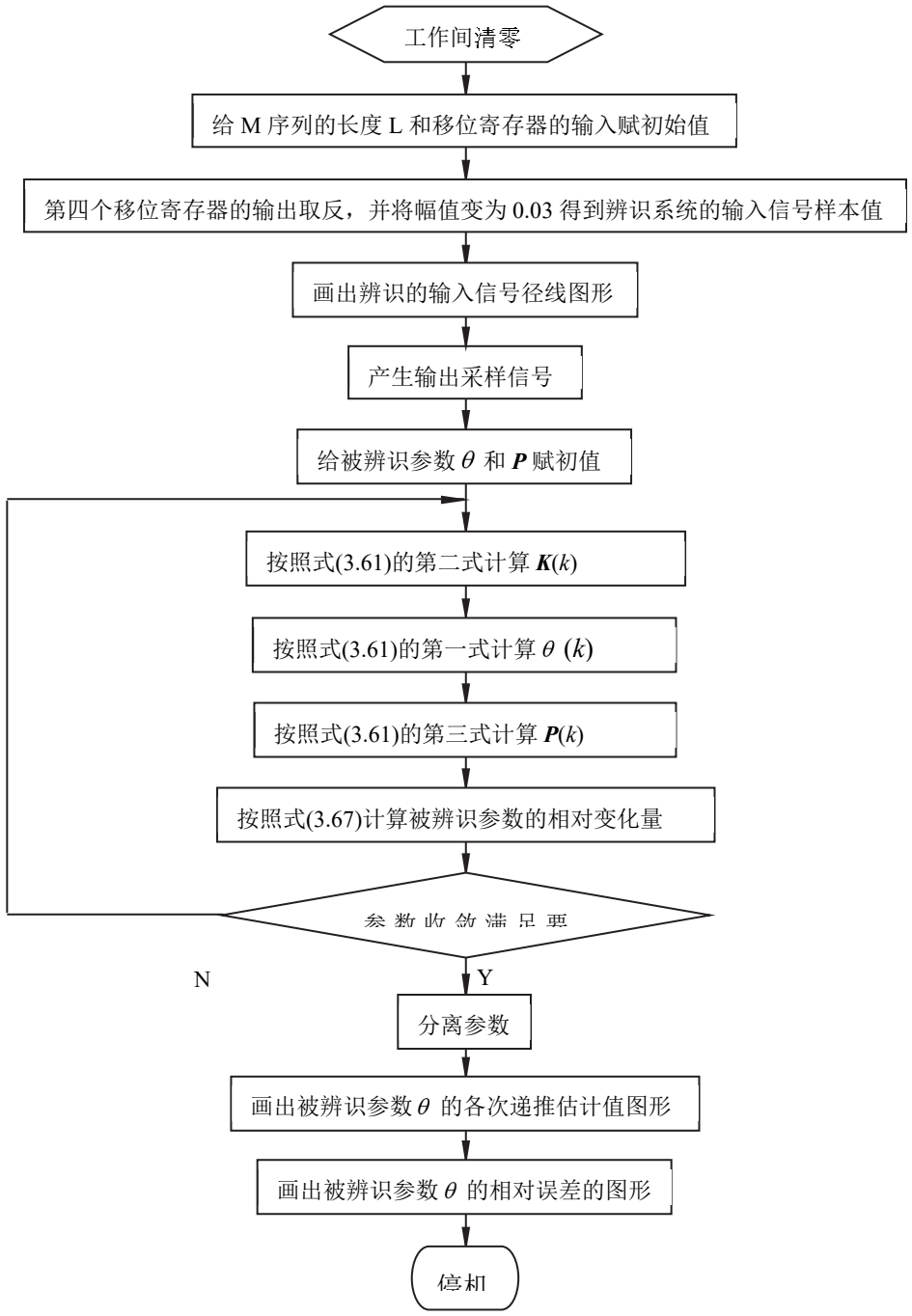

图 3.7　最小二乘递推算法辨识的 Malab6.0 程序流程图

%RLS 递推最小二乘辨识

c0=[0.001 0.001 0.001 0.001]'; %直接给出被辨识参数的初始值,即一个充分小的实向量

```
p0=10^6*eye(4,4); %直接给出初始状态 P0，即一个充分大的实数单位矩阵
E=0.000000005; %取相对误差 E=0.000000005
c=[c0,zeros(4,14)]; %被辨识参数矩阵的初始值及大小
e=zeros(4,15); %相对误差的初始值及大小
for k=3:15; %开始求 K
    h1=[-z(k-1),-z(k-2),u(k-1),u(k-2)]'; x=h1'*p0*h1+1; x1=inv(x); %开始求 K(k)
    k1=p0*h1*x1;%求出 K 的值
    d1=z(k)-h1'*c0; c1=c0+k1*d1; %求被辨识参数 c
    e1=c1-c0; %求参数当前值与上一次的值的差值
    e2=e1./c0; %求参数的相对变化
    e(:,k)=e2; %把当前相对变化的列向量加入误差矩阵的最后一列
    c0=c1; %新获得的参数作为下一次递推的旧参数
    c(:,k)=c1; %把辨识参数 c  列向量加入辨识参数矩阵的最后一列
    p1=p0-k1*k1'*[h1'*p0*h1+1]; %求出  p(k)的值
    p0=p1; %给下次用
    if e2<=E break; %如果参数收敛情况满足要求，终止计算
    end %小循环结束
end %大循环结束
c，e %显示被辨识参数及其误差(收敛)情况
%分离参数
```

```
a1=c(1,:);  a2=c(2,:);  b1=c(3,:);  b2=c(4,:);  ea1=e(1,:);  ea2=e(2,:);
eb1=e(3,:); eb2=e(4,:);
figure(2); %第二个图形
i=1:15; %横坐标从 1 到 15
plot(i,a1,'r',i,a2,':',i,b1,'g',i,b2,':')  %画出 a1，a2，b1，b2 的各次辨识结果
title('Parameter Identification with Recursive Least Squares Method') %图形标题
figure(3); %第三个图形
i=1:15; %横坐标从 1 到 15
plot(i,ea1,'r',i,ea2,'g',i,eb1,'b',i,eb2,'r:') %画出 a1，a2，b1，b2 的各次辨识结果的收敛情况
title('Identification Precision') %图形标题
```

程序运行结果：

```
>>
c =
   0.0010    0    0.0010   -0.4984   -1.2328   -1.4951   -1.4962   -1.4991   -1.4998   -1.4999
   0.0010    0    0.0010    0.0010   -0.2350    0.6913    0.6941    0.6990    0.6998    0.6999
   0.0010    0    0.2509    1.2497    1.0665    1.0017    1.0020    1.0002    0.9999    0.9998
   0.0010    0   -0.2489    0.7500    0.5668    0.5020    0.5016    0.5008    0.5002    0.5002
```

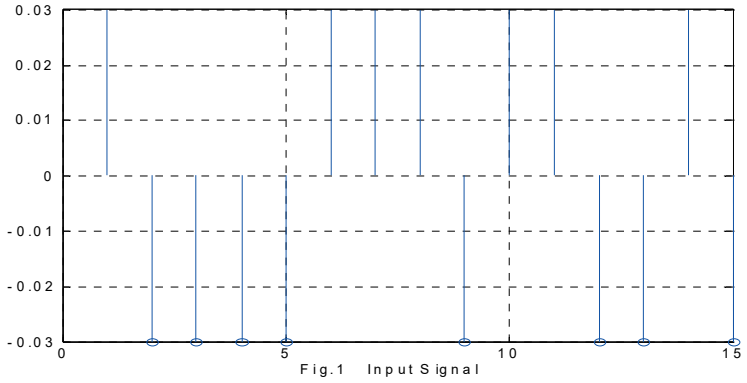

Fig.1 Input Signal

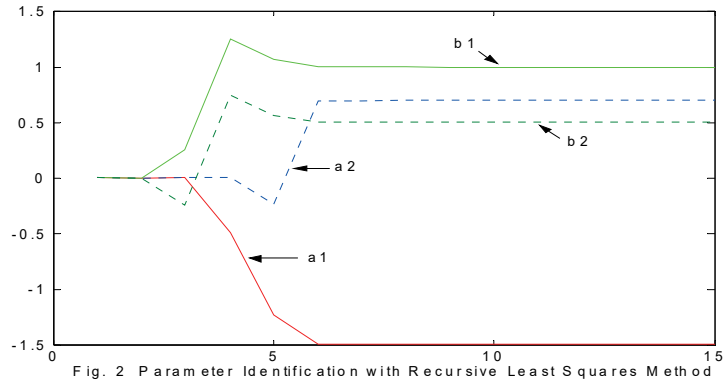

Fig. 2 Parameter Identification with Recursive Least Squares Method

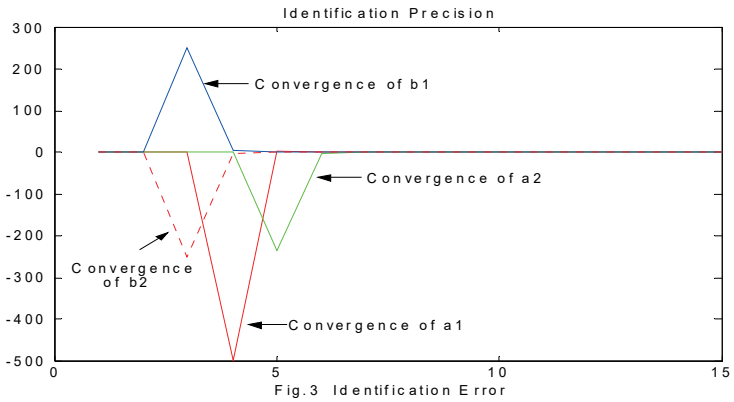

Fig.3 Identification Error

图 3.8　最小二乘递推算法的参数辨识仿真

$$\begin{pmatrix} -1.5000 & -1.5000 & -1.5000 & -1.4999 & -1.4999 \\ 0.7000 & 0.7000 & 0.7000 & 0.7000 & -0.7000 \\ 0.9999 & 0.9999 & 0.9999 & 0.9999 & 0.9999 \\ 0.5000 & 0.5000 & 0.5000 & 0.5000 & 0.5000 \end{pmatrix}$$

e =

$$\begin{pmatrix} 0 & 0 & 0 & -499.4200 & 1.4734 & 0.2128 & 0.0007 & 0.0020 & 0.0004 & 0.0000 \\ 0 & 0 & 0 & 0 & -235.9916 & -3.9416 & 0.0042 & 0.0070 & 0.0012 & 0.0001 \\ 0 & 0 & 249.8612 & 3.9816 & -0.1466 & -0.0607 & 0.0003 & -0.0018 & -0.0003 & -0.0001 \end{pmatrix}$$

0 0 -249.8612 -4.0136 -0.2443 -0.1143 -0.0007 -0.0016 -0.0012 -0.0001

$$\begin{bmatrix} 0.0001 & 0.0000 & -0.0000 & -0.0000 & 0.0000 \\ 0.0001 & -0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0001 & 0.0000 & 0.0000 & 0.0000 & -0.0000 \\ -0.0004 & 0.0000 & -0.0000 & 0.0000 & -0.0000 \end{bmatrix}$$

表 3.2　最小二乘递推算法的辨识结果

参　数	a_1	a_2	b_1	b_2
真　值	-1.5	0.7	1.0	0.5
估计值	-1.4999	0.7	0.9999	0.5000

仿真结果表明，大约递推到第十步时，参数辨识的结果基本达到稳定状态，即 a_1=-1.4999, a_2= 0.7000, b_1=0.9999, b_2=0.5000。此时，参数的相对变化量 E≤0.000000005。从整个辨识过程来看，精度的要求直接影响辨识的速度。虽然最终的精度可以达到很小，但开始阶段的相对误差会很大，从图 3.8(3)图形中可看出，参数的最大相对误差会达到三为数

例 3.6　考虑图 3.10 所示的仿真对象，图中，$v(k)$ 是服从 $\mathbf{N}(0,1)$ 分布的不相关随机噪声。

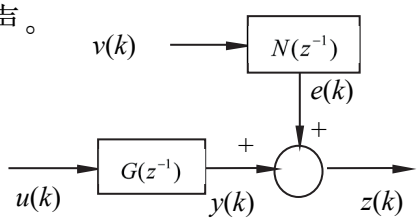

图 3.10 增广最小二乘法辨识实例结构

且

$$G(z^{-1})=\frac{B(z^{-1})}{A(z^{-1})}\quad,\quad N(z^{-1})=\frac{D(z^{-1})}{C(z^{-1})}\quad,$$

$$\begin{cases} A(z^{-1})=1-1.5a_1z^{-1}+0.7z^{-2}=C(z^{-1}) \\ B(z^{-1})=1.0z^{-1}+0.5z^{-2} \\ D(z^{-1})=1-z^{-1}+0.2z^{-2} \end{cases}$$

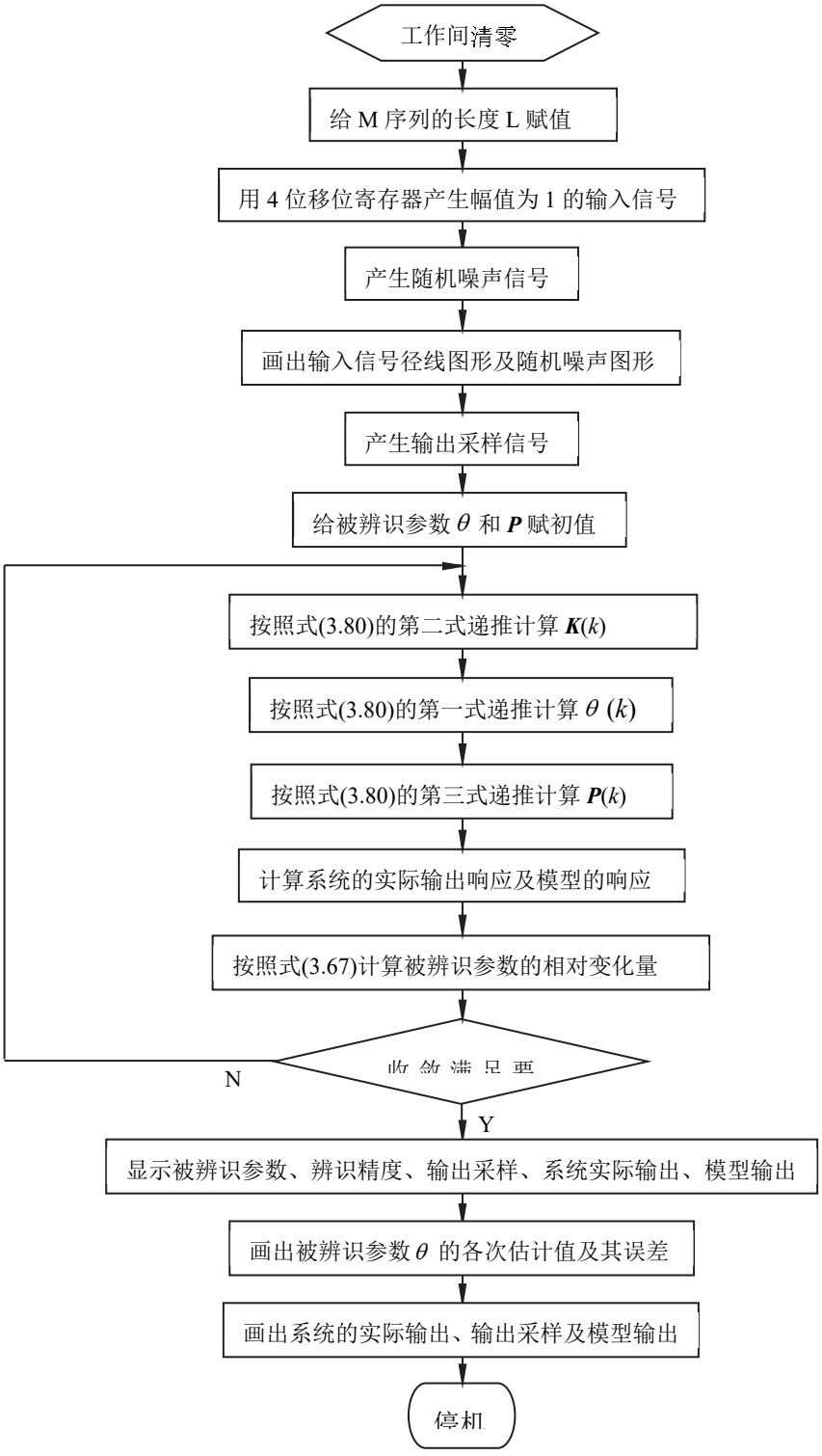

图 3.11　增广最小二乘递推算法辨识的 Malab6.0 程序流程图

模型结构选用如下形式：

$$z(k)+a_1z(k-1)+a_2(k-2)=b_1u(k-1)+b_2u(k-2)+v(k)+d_1v(k-1)+d_2(v(k-2)$$

增广最小二乘辨识程序流程如图 3.11 所示。Matlab6.0 程序如下。

```
%增广最小二乘辨识程序，在光盘中的文件名：FLch3ELSeg4.m
clear
L=60; %4 位移位寄存器产生的 M 序列的周期
y1=1;y2=1;y3=1;y4=0; %4 个移位寄存器的输出初始值
for i=1:L;
    x1=xor(y3,y4); %第一个移位寄存器的输入信号
    x2=y1; %第二个移位寄存器的输入信号
    x3=y2; %第三个移位寄存器的输入信号
    x4=y3; %第四个移位寄存器的输入信号
    y(i)=y4; %第四个移位寄存器的输出信号，M 序列, 幅值"0"和"1",
    if y(i)>0.5,u(i)=-1; %M 序列的值为"1"时,辨识的输入信号取“-1”
    else u(i)=1; %M 序列的值为"0"时,辨识的输入信号取“1”
    end
    y1=x1;y2=x2;y3=x3;y4=x4; %为下一次的输入信号准备
end
figure(1); %画第一个图形
subplot(2,1,1); %画第一个图形的第一个子图
stem(u),grid on %画出 M 序列输入信号
v=randn(1,60); %产生一组 60 个正态分布的随机噪声
subplot(2,1,2); %画第一个图形的第二个子图
```

```
plot(v),grid on; %画出随机噪声信号
u ， v %显示输入信号和噪声信号
z=zeros(7,60);zs=zeros(7,60);zm=zeros(7,60);zmd=zeros(7,60); % 输 出采样矩阵、不考虑噪声时系统输出矩阵、不考虑噪声时模型输出矩阵、模型输出矩阵的大小
z(2)=0; z(1)=0; zs(2)=0; zs(1)=0; zm(2)=0; zm(1)=0; zmd(2)=0; zmd(1)=0; %输出采样、不考虑噪声时系统输出、不考虑噪声时模型输出、模型输出的初值
c0=[0.001 0.001 0.001 0.001 0.001 0.001 0.001]'; %直接给出被辨识参数的初始值,即一个充分小的实向量
p0=10^6*eye(7,7); %直接给出初始状态 P0，即一个充分大的实数单位矩阵
E=0.00000000005; %相对误差 E=0.000000005
c=[c0,zeros(7,14)]; %被辨识参数矩阵的初始值及大小
e=zeros(7,15); %相对误差的初始值及大小
for k=3:60; %开始求 K
    z(k)=1.5*z(k-1)-0.7*z(k-2)+u(k-1)+0.5*u(k-2)+v(k)-v(k-
    1)+0.2*v(k-2); %系统在 M 序列输入下的输出采样信号
    h1=[-z(k-1),-z(k-2),u(k-1),u(k-2),v(k),v(k-1),v(k-2)]'; %为求 K(k)作准备
    x=h1'*p0*h1+1; x1=inv(x); k1=p0*h1*x1; %K
    d1=z(k)-h1'*c0; c1=c0+k1*d1; %辨识参数 c
```

```
    zs(k)=-1.5*z(k-1)+0.7*z(k-2)+u(k-1)+0.5*u(k-2); %系统在 M 序列的输入下的输出响应
    zm(k)=[-z(k-1),-z(k-2),u(k-1),u(k-2)]*[c1(1);c1(2);c1(3);c1(4)]; %模型在 M 序列的输入下的输出响应
    zmd(k)=h1'*c1; %模型在 M 序列的输入下的输出响应
    e1=c1-c0; e2=e1./c0; %求参数误差的相对变化
    e(:,k)=e2;
    c0=c1; %给下一次用
    c(:,k)=c1; %把递推出的辨识参数 c 的列向量加入辨识参数矩阵
    p1=p0-k1*k1'*[h1'*p0*h1+1]; %find p(k)
    p0=p1; %给下次用
    if e2<=E break; %若收敛情况满足要求，终止计算
    end %判断结束
end %循环结束
c, e, %显示被辨识参数及参数收敛情况
z, zs, zm %显示输出采样值、系统实际输出值、模型输出值
%分离变量
a1=c(1,:); a2=c(2,:); b1=c(3,:); b2=c(4,:); %分离出 a1、 a2、 b1、 b2
d1=c(5,:); d2=c(6,:); d3=c(7,:); %分离出 d1、 d2、 d3
ea1=e(1,:); ea2=e(2,:); eb1=e(3,:); eb2=e(4,:); %分离出 a1、 a2、 b1、 b2 的收敛情况
ed1=e(5,:); ed2=e(6,:); ed3=e(7,:); %分离出 d1 、d2 、d3 的收敛情况
```

figure(2); %画第二个图形

i=1:60;plot(i,a1,'r',i,a2,'r:',i,b1,'b',i,b2,'b:',i,d1,'g',i,d2,'g:',i,d3,'g+') %画出各个被辨识参数

title('Parameter Identification with Recursive Least Squares Method') %标题

figure(3); i=1:60; %画出第三个图形

plot(i,ea1,'r',i,ea2,'r:',i,eb1,'b',i,eb2,'b:',i,ed1,'g',i,ed2,'g:',i,ed2,'r+') %画出各个参数收敛情况

title('Identification Precision') %标题

figure(4); subplot(4,1,1); %画出第四个图形，第一个子图

i=1:60;plot(i,zs(i),'r') %画出被辨识系统在没有噪声情况下的实际输出响应

subplot(4,1,2); i=1:60;plot(i,z(i),'g') %第二个子图，画出被辨识系统的采样输出响应

subplot(4,1,3); i=1:60;plot(i,zm(i),'b') %第三个子图，画出模型含有噪声的输出响应

subplot(4,1,4); i=1:60;plot(i,zs(i),'b') %第四个子图，画出模型去除噪声后的输出响应

程序执行结果：

```
>>
u =
[1,-1,-1,-1,-1,1,1,1,-1,1,1,-1,-1,1,-1,1,-1,-1,1,-1,-1,1,1,-1,1,1,-1,-1,1,-1,1,-1,-1,-1,-1,1,1,1,-1,1,1,-1,-1,1,-1,1,-1,-1,-1,-1,1,1,1,-1,1,1,-1,-1,1,-1]
v =
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：https://d.book118.com/408072045001006124