

第一章 三种最小二乘方法估计参数值

1. 实验题目： 三种最小二乘方法估计参数值

2. 实验目的

通过使用三种不同的最小二乘法对系统进行辨识，学习三种不同方法的原理，优缺点以及它的适用范围。

3. 实验主要原理

1. 一般最小二乘法

若考虑被辨识系统或观测信息中含有噪声，被辨识模型可以改写为

$$z(k) = - \sum_{i=1}^n a_i y(k-i) + \sum_{i=1}^n b_i u(k-i) + v(k)$$

式中， $z(k)$ 为系统输出量的第 k 次观测值； $y(k)$ 为系统输出量的第 k 次真值； $u(k)$ 为系统第 k 个输入值； $v(k)$ 是均值为0的随机噪声。

可以将被辨识模型用矩阵形式表示

$$\begin{bmatrix} z(n+1) \\ z(n+2) \\ \vdots \\ z(n+N) \end{bmatrix} = \begin{bmatrix} -z(n) & \cdots & -z(1) & u(n+1) & \cdots & u(1) \\ -z(n+1) & \cdots & -z(2) & u(n+2) & \cdots & u(2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -z(n+N-1) & \cdots & -z(N) & u(n+N) & \cdots & u(N) \end{bmatrix} \times \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_0 \\ \vdots \\ b_n \end{bmatrix} + \begin{bmatrix} v(n+1) \\ v(n+2) \\ \vdots \\ v(n+N) \end{bmatrix}$$

即 $Z_m = H_m \theta + V_m$

最小二乘的思想就是寻找一个 θ 的估计值 $\hat{\theta}$ ，使得各次测量的 $Z_i (i = 1, 2, \dots, m)$ 与由估计 $\hat{\theta}$ 确定的测量估计值 $\hat{Z}_i = H_i \hat{\theta}$ 之差的平方和最小，即

$$J(\hat{\theta}) = (Z_m - H_m \hat{\theta})^T (Z_m - H_m \hat{\theta})$$

要使上式达到最小，根据极值定理，则有

$$\left. \frac{\partial J}{\partial \theta} \right|_{\theta=\hat{\theta}} = -2H_m^T (Z_m - H_m \hat{\theta}) = 0$$

对上式进一步整理，得

$$H_m^T H_m \hat{\theta} = H_m^T Z_m$$

如果 H_m 的行数大于等于列数，即 $m \geq 2n$ ， $H_m^T H_m$ 满秩，即 $\text{rank}(H_m^T H_m) = 2n$ ，

即 $(H_m^T H_m)^{-1}$ 存在。则 θ 的最小二乘估计为

$$\hat{\theta} = (H_m^T H_m)^{-1} H_m^T Z_m$$

2. 递推最小二乘法

参数递推估计是指对被辨识的系统，每取得一次新的测量数据后，就在前一次估计结果的基础上，利用新引入的测量数据对前一次估计的结果进行修正，从而递推地得出新的参数估计值。这样，随着新测量数据的引入一次接一次地进行参数估计，直到估计值达到满意的精确程度为止。最小二乘递推算法的基本思想可以概括为：

当前估计值 $\hat{\theta}(k)$ = 上次估计值 $\hat{\theta}(k-1)$ + 修正项

即新的估计值 $\hat{\theta}(k)$ 是在旧的估计值 $\hat{\theta}(k-1)$ 的基础上，利用新的观测数据对旧的估计值进行修正而成的。

根据加权最小二乘法，利用 m 次测量数据所得到的估值为

$$\hat{\theta}_m = (H_m^T W_m H_m)^{-1} H_m^T W_m Z_m$$

当新获得一对输入、输出数据时， $z(m+1) = h(m+1)\theta + v(m+1)$

利用 $m+1$ 次输入、输出数据，得到的方程为： $Z_{m+1} = H_{m+1}\theta + V_{m+1}$

式中

$$Z_{m+1} = \begin{bmatrix} Z_m \\ z(m+1) \end{bmatrix}, \quad H_{m+1} = \begin{bmatrix} H_m \\ h(m+1) \end{bmatrix}, \quad V_{m+1} = \begin{bmatrix} V_m \\ v(m+1) \end{bmatrix}$$

同理可得

$$\hat{\theta}_{m+1} = (H_{m+1}^T W_{m+1} H_{m+1})^{-1} H_{m+1}^T W_{m+1} Z_{m+1}$$

式中

$$W_{m+1} = \begin{bmatrix} W_m & 0 \\ 0 & w(m+1) \end{bmatrix}$$

设

$$P_m = [H_m^T W_m H_m]^{-1}$$

$$P_{m+1} = [H_{m+1}^T W_{m+1} H_{m+1}]^{-1}$$

则有

$$\hat{\theta}_m = P_m H_m^T W_m Z_m$$

$$\hat{\theta}_{m+1} = P_{m+1} H_{m+1}^T W_{m+1} Z_{m+1}$$

$\hat{\theta}_{m+1}$ 还可以表示为

$$\hat{\theta}_{m+1} = P_{m+1} \begin{bmatrix} H_m^T & h^T(m+1) \end{bmatrix} \begin{bmatrix} W_m & 0 \\ 0 & w(m+1) \end{bmatrix} \begin{bmatrix} Z_m \\ z(m+1) \end{bmatrix}$$

$$= P_{m+1} P_m^{-1} \hat{\theta}_m + P_{m+1} h^T(m+1) w(m+1) z(m+1)$$

又可得

$$P_{m+1} = P_m - P_m h^T(m+1) [w^{-1}(m+1) + h(m+1) P_m h^T(m+1)]^{-1} h(m+1) P_m$$

代入可得

$$\hat{\theta}_{m+1} = \hat{\theta}_m + P_{m+1} h^T(m+1) w(m+1) [z(m+1) - h(m+1) \hat{\theta}_m]$$

令 $K_{m+1} = P_{m+1} h^T(m+1) w(m+1)$ 为增益矩阵, 综上递推算法可以表示为

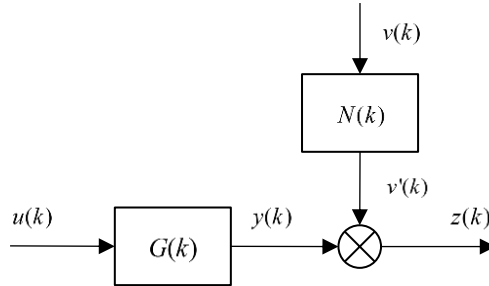
$$\hat{\theta}_{m+1} = \hat{\theta}_m + K_{m+1} [z(m+1) - h(m+1) \hat{\theta}_m]$$

$$P_{m+1} = P_m - P_m h^T(m+1) [w^{-1}(m+1) + h(m+1) P_m h^T(m+1)]^{-1} h(m+1) P_m$$

$$K_{m+1} = P_m h^T(m+1) [w^{-1}(m+1) + h(m+1) P_m h^T(m+1)]^{-1}$$

3. 增广最小二乘法

当噪声均值为 0 时, 最小二乘参数估计算法为无偏估计; 当噪声的均值不为 0 时, 最小二乘参数估计算法为有偏估计。为解决最小二乘参数的有偏性, 将噪声模型的辨识考虑进去, 因此称为增广最小二乘法。该算法可以看成是对一般最小二乘参数辨识算法的简单推广或扩充, 因此又称为扩充最小二乘算法。



考虑一个 SISO 动态系统如上图所示, 被辨识模型为

$$z(k) = -\sum_{i=1}^n a_i y(k-i) + \sum_{i=1}^n b_i u(k-i) + v'(k)$$

$$v'(k) = N(k)v(k)$$

式中, $z(k)$ 为系统输出量的第 k 次观测值, $y(k)$ 为系统输出量的第 k 次真实值; $u(k)$ 为系统第 k 个输入值; $v'(k)$ 是均值非 0 的随机噪声; $v(k)$ 是均值为 0 的随机噪声。

将噪声模型 $N(k)$ 的参数也进行辨识, 得

$$z(k) = -\sum_{i=1}^n a_i y(k-i) + \sum_{i=1}^n b_i u(k-i) + \sum_{i=1}^n c_i v(k-i) + v(k)$$

对上式进行递推最小二乘算法的过程就是增广递推最小二乘算法。

4. 实验对象或参数

1. 用一般最小二乘法对以下数学模型进行参数辨识

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

式中，理想的系数值 $a_1 = 1.6$ 、 $a_2 = 0.7$ 、 $b_1 = 1.0$ 和 $b_2 = 0.4$ ； $V(k)$ 是服从 $N(0,0.1)$ 或 $N(0,0.5)$ 的随机噪声；输入 $u(k)$ 采用 4 阶 M 序列，其幅值为 1。

2. 用递推最小二乘法对以下数学模型进行参数辨识

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

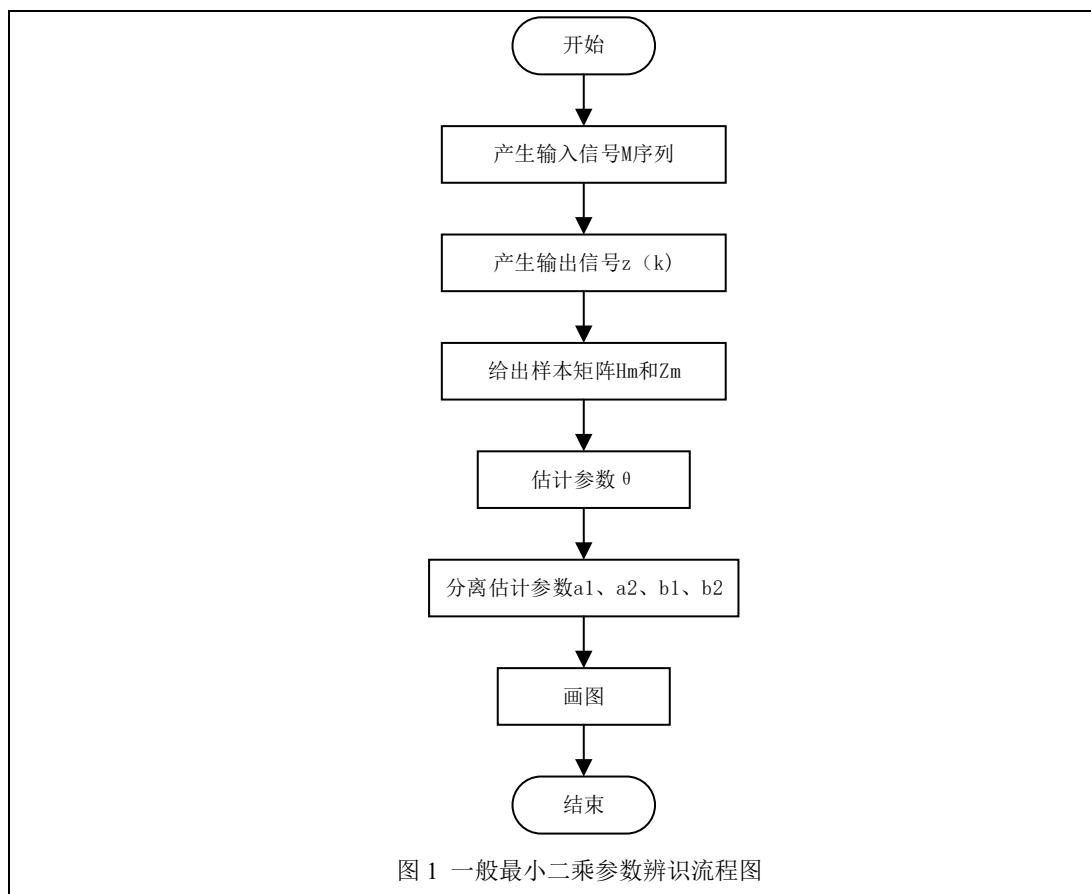
式中，理想的系数值 $a_1 = 1.6$ 、 $a_2 = 0.7$ 、 $b_1 = 1.0$ 和 $b_2 = 0.4$ ； $V(k)$ 是服从 $N(0,0.1)$ 或 $N(0,0.5)$ 的随机噪声；输入 $u(k)$ 采用 4 阶 M 序列，其幅值为 1。权阵 $W_m = I$ 。

3. 用增广递推最小二乘法对以下数学模型进行参数辨识

$$z(k) + a_1 z(k-1) + a_2 z(k-2) = b_1 u(k-1) + b_2 u(k-2) + c_1 v(k) - c_2 v(k-1) + c_3 v(k-2)$$

式中，理想的系数值 $a_1 = 1.6$ 、 $a_2 = 0.7$ 、 $b_1 = 1.0$ 、 $b_2 = 0.4$ 、 $c_1 = 1.2$ 、 $c_2 = -1.0$ 和 $c_3 = 0.2$ ； $\{v(k)\}$ 是服从正态分布的白噪声 $N(0,0.1)$ 或 $N(0,0.5)$ ；输入信号采用 4 阶 M 序列，其幅值为 1。

5. 程序框图



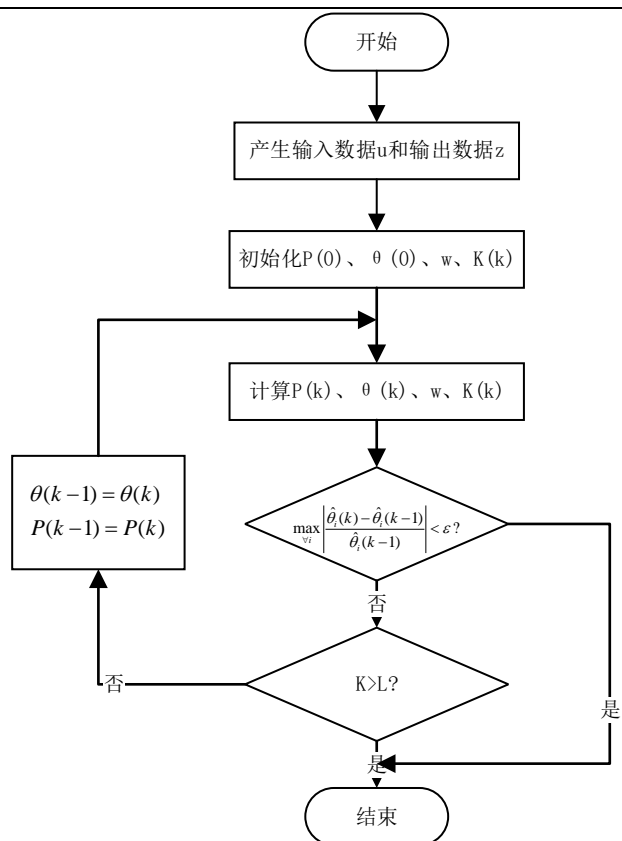


图 2 递推最小二乘参数辨识流程图

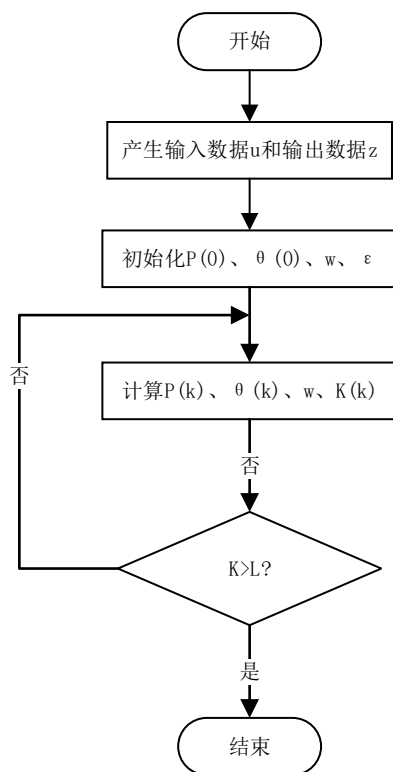


图 3 增广递推最小二乘参数辨识流程图

6. 程序代码

```
1.一般最小二乘法
clear all
close all
clc
randn('seed',100)
v=randn(1,16);
L=15;
y1=1;y2=1;y3=1;y4=0;
for i=1:L
    x1=xor(y3,y4);
    x2=y1;
    x3=y2;
    x4=y3;
    y(i)=y4;
    if y(i)>0.5,u(i)=-5;
    else u(i)=5;
    end
    y1=x1;y2=x2;y3=x3;y4=x4;
end
figure(1);
stem(u),grid on
title('输入信号 M 序列');
%最小辨识程序
z=zeros(1,16);
for k=3:16
    z(k)=-1.5*z(k-1)-0.7*z(k-2)+u(k-1)+0.5*u(k-2)+1*v(k);
end
figure(2);
plot([1:16],z);
title('输出观测值')
figure(3);
stem(z),grid on
title('输出观测值 z 的经线图形');
H=[-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)];
Z=[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)]
```

```

c=inv(H'*H)*H'*Z;
a1=c(1),a2=c(2),b1=c(3),b2=c(4)

2.递推最小二乘法
clear all
close all
clc
randn('seed',100)
v=randn(1,60);
L=60;
y1=1;y2=1;y3=1;y4=0;
for i=1:L
x1=xor(y3,y4);
x2=y1;
x3=y2;
x4=y3;
y(i)=y4;
if y(i)>0.5,u(i)=-5;
else u(i)=5;
end
y1=x1;y2=x2;y3=x3;y4=x4;
end
figure(1);
stem(u),grid on
z(2)=0;z(1)=0;
for k=3:60
z(k)=-1.5*z(k-1)-0.7*z(k-2)+u(k-1)+0.5*u(k-2)+0.5*v(k);
end
c0=[0.001 0.001 0.001 0.001]';
p0=10^3*eye(4,4);
E=0.000000005;%相对误差
c=[c0,zeros(4,59)];%被辨识参数矩阵的初始值及大小
e=zeros(4,60);%相对误差的初始值及大小
lamt=1;
for k=3:60
h1=[-z(k-1),-z(k-2),u(k-1),u(k-2)]';
k1=p0*h1*inv(h1'*p0*h1+1*lamt);%求出 K 的值
new=z(k)-h1'*c0;
c1=c0+k1*new;%求辨识参数 C
p1=1/lamt*(eye(4)-k1*h1')*p0;
e1=(c1-c0)./c0;%求参数当前值与上一次的值的差值
e(:,k)=e1;%把当前相对变化的列向量加入误差矩阵的最后一列
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

```

```

c0=c1;%新获得的参数作为下一次递推的旧参数
p0=p1;
if norm(e1)<=E
break;
end
end
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);
i=1:60;
plot(i,a1,'k',i,a2,'b',i,b1,'r',i,b2,'g')
legend('a1','a2','b1','b2');
title('递推最小二乘参数辨识')
figure(3);
i=1:60;
plot(i,ea1,'k',i,ea2,'b',i,eb1,'r',i,eb2,'g')
legend('a1','a2','b1','b2');
title('辨识精度');

```

3. 增广最小二乘法

```

clear all
close all
clc
L=60;
y1=1;y2=1;y3=1;y4=0;
for i=1:L
    x1=xor(y3,y4);
    x2=y1;
    x3=y2;
    x4=y3;
    y(i)=y4;
    if y(i)>0.5,u(i)=-1;
    else u(i)=1;
    end
    y1=x1;
    y2=x2;
    y3=x3;
    y4=x4;
end
figure(1);
stem(u),grid on
randn('seed',100)
v=randn(1,60);

```



```

z(2)=0;z(1)=0;
theat0=[0.001 0.001 0.001 0.001 0.001 0.001 0.001]';
p0=10^4*eye(7,7);
theat=[theat0,zeros(7,59)];
for k=3:60
    z(k)=-1.5*z(k-1)-0.7*z(k-2)+u(k-1)+0.5*u(k-2)+1.2*v(k)-v(k-1)+0.2*v(k-1)
    h1=[-z(k-1),-z(k-2),u(k-1),u(k-2),v(k),v(k-1),v(k-2)]';
    x=h1'*p0*h1+1;
    x1=inv(x);
    k1=p0*h1*x1;
    d1=z(k)-h1'*theat0;
    theat1=theat0+k1*d1;
    theat0=theat1;
    theat(:,k)=theat1;
    p1=p0-k1*k1'*[h1'*p0*h1+1];
    p0=p1;
end
a1=theat(1,:);a2=theat(2,:);b1=theat(3,:);b2=theat(4,:);
c1=theat(5,:);c2=theat(6,:);c3=theat(7,:);
i=1:60;
figure(2);
plot(i,z);
figure(3);
plot(i,a1,'r',i,a2,'b',i,b1,'k',i,b2,'y',i,c1,'g',i,c2,'c',i,c3,'m')
title('增广递推最小二乘辨识算法');

```

7. 实验结果及分析

1.一般最小二乘法:

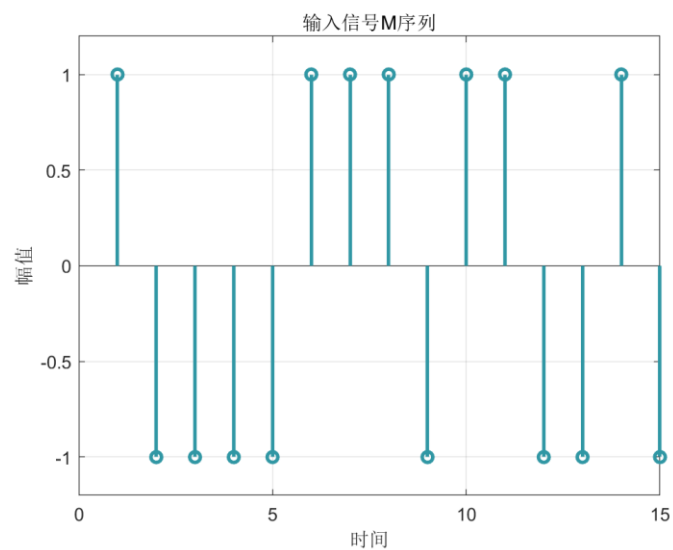


图4 输入 M 序列

表 1 一般最小二乘估计的辨识参数

参数	a1	a2	b1	b2
真实值	1.6	0.7	1.0	0.40
N(0,0.1)白噪声估计值	1.5067	0.6335	0.9363	0.3053
N(0,0.5)白噪声估计值	1.3400	0.4980	0.8425	0.1613

采用幅值为 1 的 M 序列作为系统输入，其时间序列图如图 4 所示，运行代码后可以得出结果如表 1 所示，图 5 图 6 为系统在激励下的输出信息。尝试方差为 0.8，1 的正态分布容易发现，随着白噪声方差的增加，系统辨识的精度逐步降低。针对方差较小的数据，噪声的幅值分布越加集中，由于设定的噪声为符合正态分布的白噪声，即信号在 0 的范围内波动，方差越小，其噪声幅值整体而言更加接近于 0，即噪声的干扰作用相比于方差大的白噪声越小，则系统在此基础上进行辨识，噪声对辨识精度的影响也将越小，符合真实的测试结果。

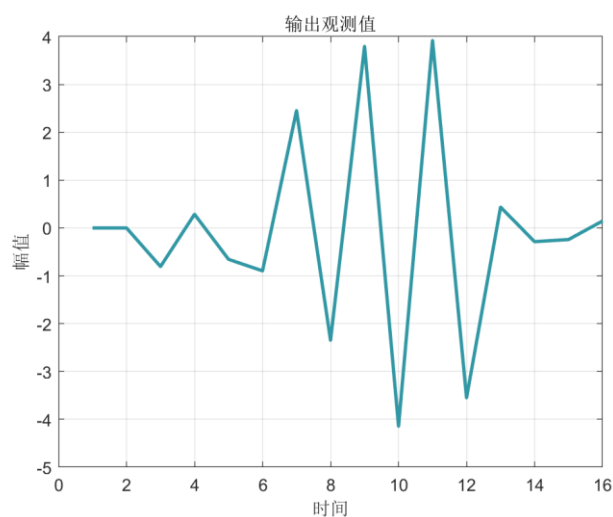


图 5 输出观测值

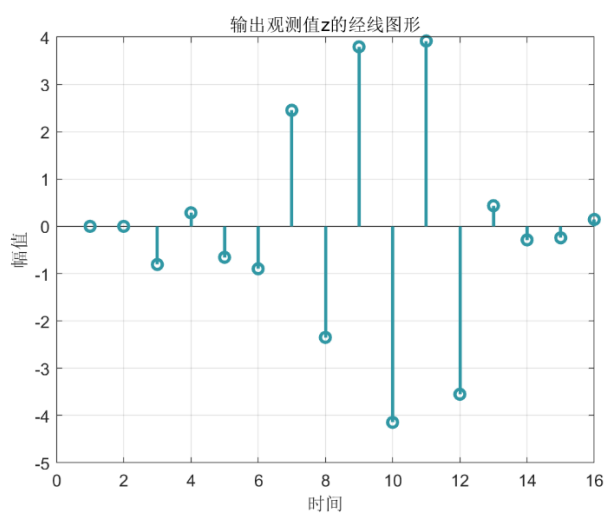


图 6 输出观测值 z 的经线图

2.递推最小二乘:

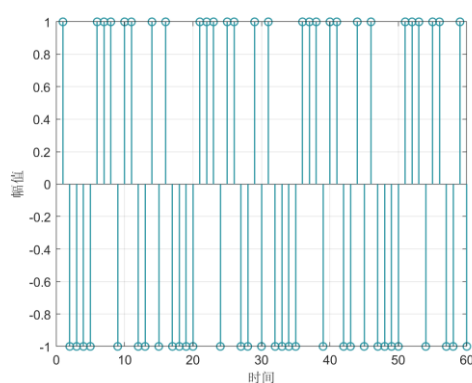


图 7 M 序列

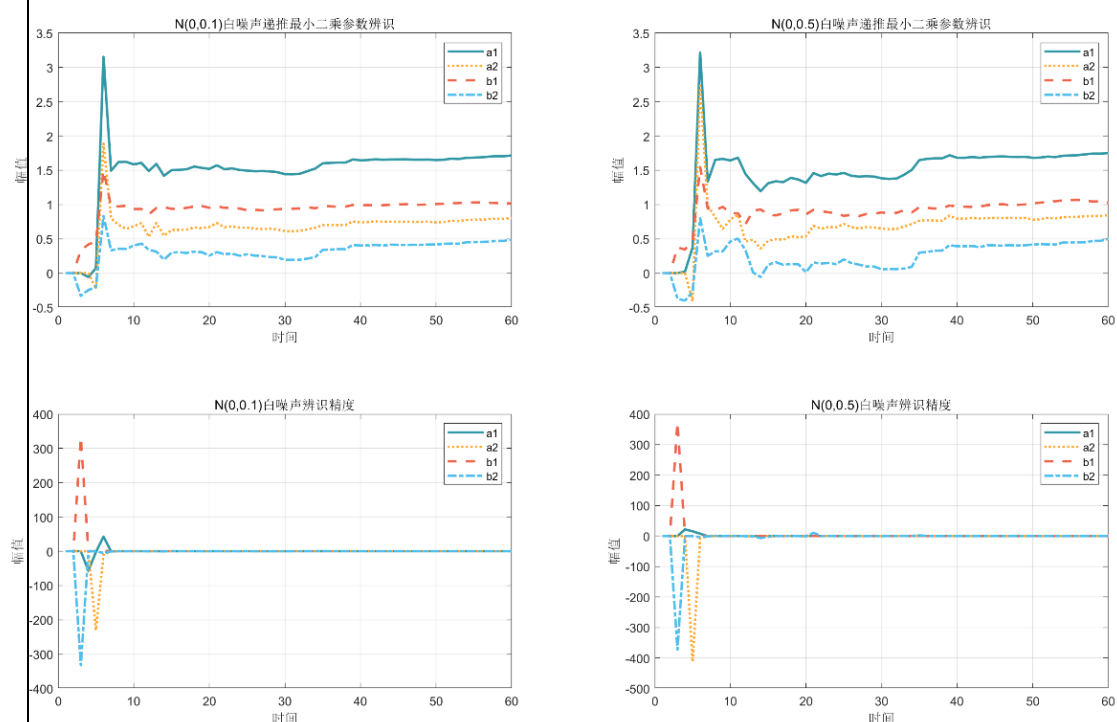


图 8 递推最小二乘结果图

图 7 展示了用于递推最小二乘的幅值为 1 的 M 序列输出，图 8 针对白噪声方差不同的情况给出了分别的参数递推过程以及精度变化过程。通过图 8 可以发现，递推算法在递推初期不能够快速收敛，而是在剧烈波动后逐渐稳定，通过两种方差大小的对比再回看一般最小二乘的辨识结果，递推最小二乘相比于一般最小二乘，减弱了白噪声方差不同带来的影响，辨识最终精度差别不大且收敛速度大致相同。同时，通过表 2 给出的结果可以看出，随着迭代次数的增加，精度逐步增加，但是计算时间也随之增加。

表 2 递推最小二乘迭代次数比较表

参数	a1	a2	b1	b2
真值	1.6	0.7	1.0	0.40
估计值 (60 次迭代)	1.7508	0.8488	1.0275	0.4999
估计值 (1000 次迭代)	1.6051	0.7106	1.015	0.4262

3.增广递推最小二乘:

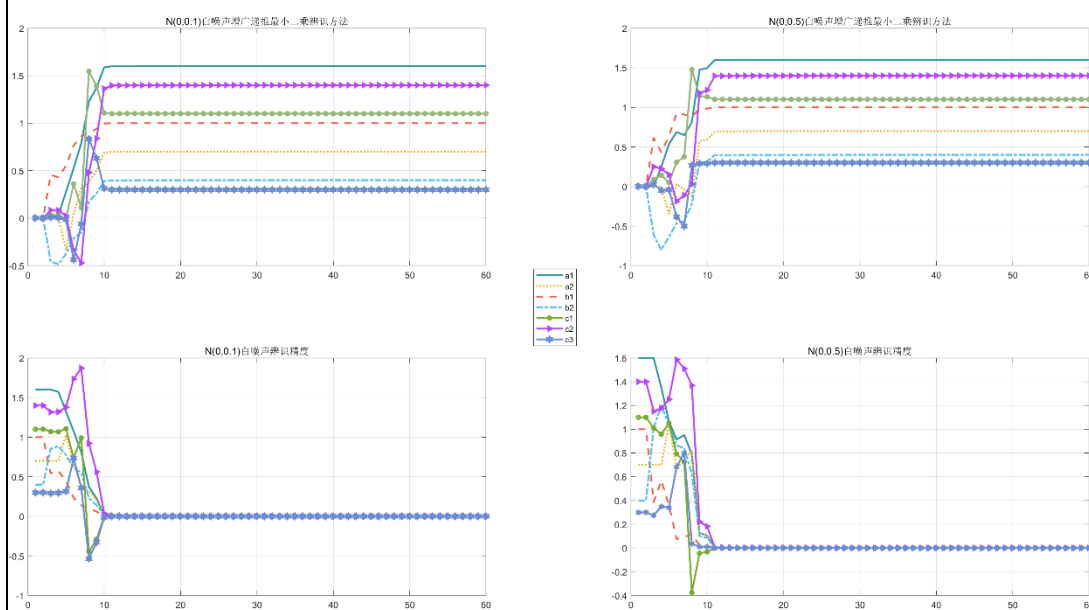


图9 增广递推最小二乘算法

图9针对白噪声方差不同的情况给出了分别的参数递推过程以及精度变化过程。通过图9可以发现，递推算法在递推初期不能够快速收敛，而是在剧烈波动后逐渐稳定，通过两种方差大小的对比再回看一般最小二乘以及递推最小二乘的辨识结果，增广递推最小二乘相比于一般最小二乘，减弱了白噪声方差不同带来的影响，辨识最终精度差别不大且收敛速度大致相同。

8. 结论

- 1、离线辨识精度较高，在线辨识则精度较低，但可实时进行参数估计。
- 2、最小二乘估计是的无偏的、一致的、有效的、渐近正态的。
- 2、一般最小二乘的估计精度较低，如果对不同测量值的置信度有所了解，则可采用有加权的办法分别对待各测量值，构成加权最小二乘法提高辨识精度。
- 3、最小二乘法不适合实时处理，递推最小二乘法的算法简单，能减少计算量，减少数据在计算机中占用的内存，并实时辨识出系统动态特性。
- 4、噪声方差会对辨识结果精度造成影响，且噪声方差越大时起始阶段估计值波动较大。如果噪声为白噪声时最小二乘法估计精度较高，如果为有色噪声时则可对噪声进行建模并一起辨识，构成增广最小二乘辨识。
- 5、递推最小二乘法只适合与噪信比较小时的白噪声影响下的系统辨识，而不适合对于有色噪声影响下的系统辨识或者噪信比较大时的白噪声影响下的系统辨识。

第二章 两种极大似然方法估计参数值

1. 实验题目：_____两种极大似然方法估计参数值_____

2. 实验目的

通过使用两种不同的极大似然方法对系统进行辨识，学习两种不同方法的原理，优缺点以及它的适用范围。

3. 实验主要原理

设 y 为一随机变量，在未知参数 θ 条件下， y 的概率密度分布函数 $p(Y_N|\theta)$ 的分布类型已知。为了得到 θ 的估计值，对随机变量 y 进行 N 次观测，得到一随机观测序列 $\{y(k)\}$ ，其中 $k=1,2,\dots,N$ 。如果把这 N 个观测值记作 $Y_N=[y(1),y(2),\dots,y(N)]^T$ ，则 Y_N 的联合概率密度为 $p(Y_N|\theta)$ ，那么参数 θ 的极大似然估计就是使观测值 $Y_N=[y(1),y(2),\dots,y(N)]^T$ 出现概率为最大的参数估计值 $\hat{\theta}_{ML}$ ， $\hat{\theta}_{ML}$ 称为 θ 的极大似然估计。即

$$p(Y_N|\theta)_{\hat{\theta}_{ML}} = \max$$

因此，极大似然参数估计的意义在于：对一组确定的随机观测值 Y_N ，设法找到极大似然估计值 $\hat{\theta}_{ML}$ ，使随机变量 y 在 $\hat{\theta}_{ML}$ 条件下的概率密度函数最大可能地逼近随机变量 y 在 θ_0 (θ 的真值) 条件下的概率密度函数，即

$$p(y|\theta)_{\hat{\theta}_{ML}} \xrightarrow{\max} p(y|\theta_0)$$

对一组确定的观测数据 Y_N ， $p(Y_N|\theta)$ 仅仅是未知参数 θ 的函数，已不再是概率密度函数的概念，此时的 $p(Y_N|\theta)$ 称作 θ 的似然函数，记为 $L(Y_N|\theta)$ ，使 $L(Y_N|\theta)$ 达到极大值的 θ 即为极大似然估计值 $\hat{\theta}_{ML}$ ，一般通过求解下列方程获得 $L(Y_N|\theta)$ 的驻点，从而解得 $\hat{\theta}_{ML}$

$$\left[\frac{\partial L(Y_N|\theta)}{\partial \theta} \right]_{\hat{\theta}_{ML}}^T = 0$$

上述方程称为似然方程。由于对数函数是单调递增函数， $L(Y_N|\theta)$ 和 $\ln L(Y_N|\theta)$ 具

有相同的极值点。在实际应用中，为便于求取 $\hat{\theta}_{ML}$ ，将连乘计算转变为连加，对似然函数取对数得到 $\ln L(Y_N|\theta)$ ，称为对数似然函数，求解对数似然方程

$$\left[\frac{\partial \ln L(\theta)}{\partial \theta} \right]_{\hat{\theta}_{ML}}^T = 0$$

从而获得参数 θ 的极大似然估计 $\hat{\theta}_{ML}$ 。

4. 实验对象或参数

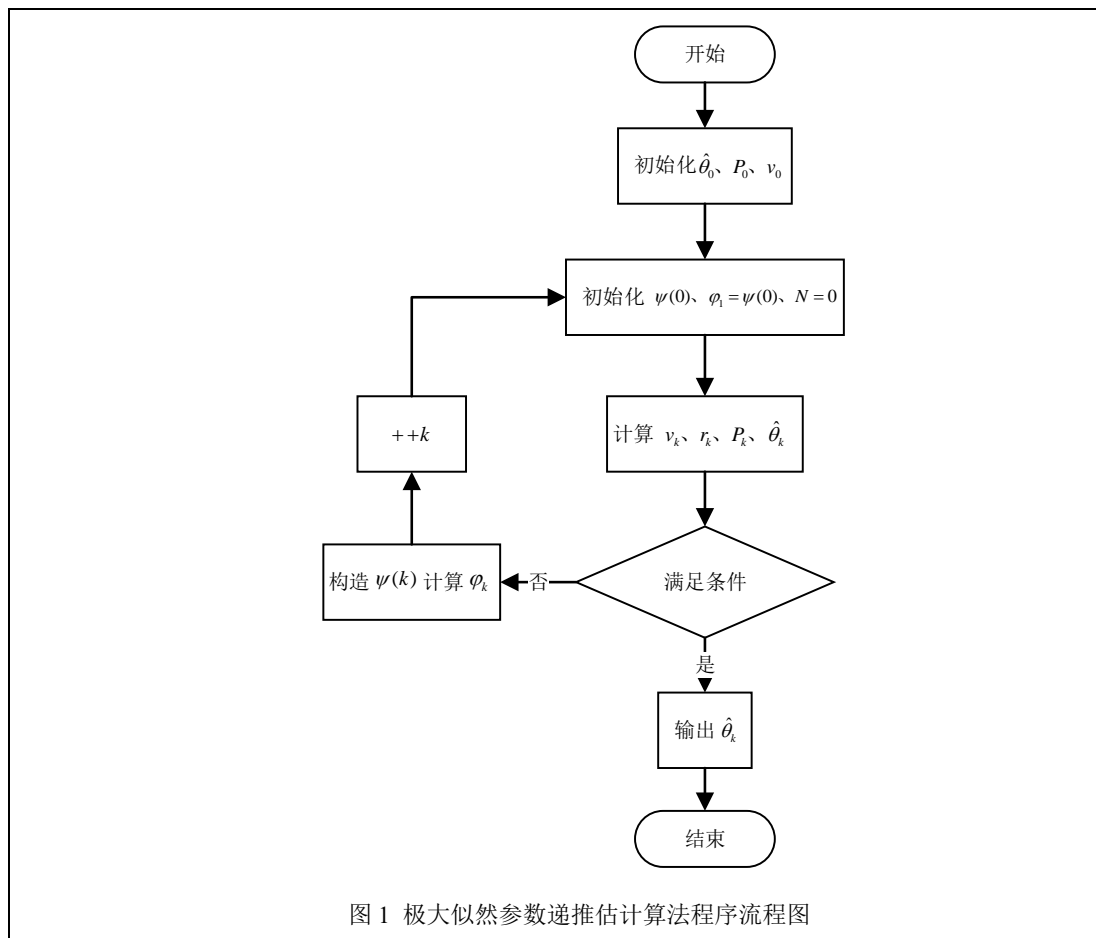
动态系统模型为

$$y(k) + a_1 y(k-1) + a_2 y(k-2) = b_1 u(k-1) + b_2 u(k-2) + v(k) - d_1 v(k-1) + d_2 v(k-2)$$

其中模型参数 $a_1 = -0.5$ 、 $a_2 = -0.2$ 、 $b_1 = 1.0$ 、 $b_2 = 1.5$ 、 $d_1 = -0.8$ 、 $d_2 = 0.3$ ；

噪声 $v(k)$ 是均值为 0，方差为 0.01 的高斯白噪声，输入信号为 4 级移位寄存器产生的 M 序列，利用极大似然估计算法对该动态系统模型参数进行辨识。

5. 程序框图



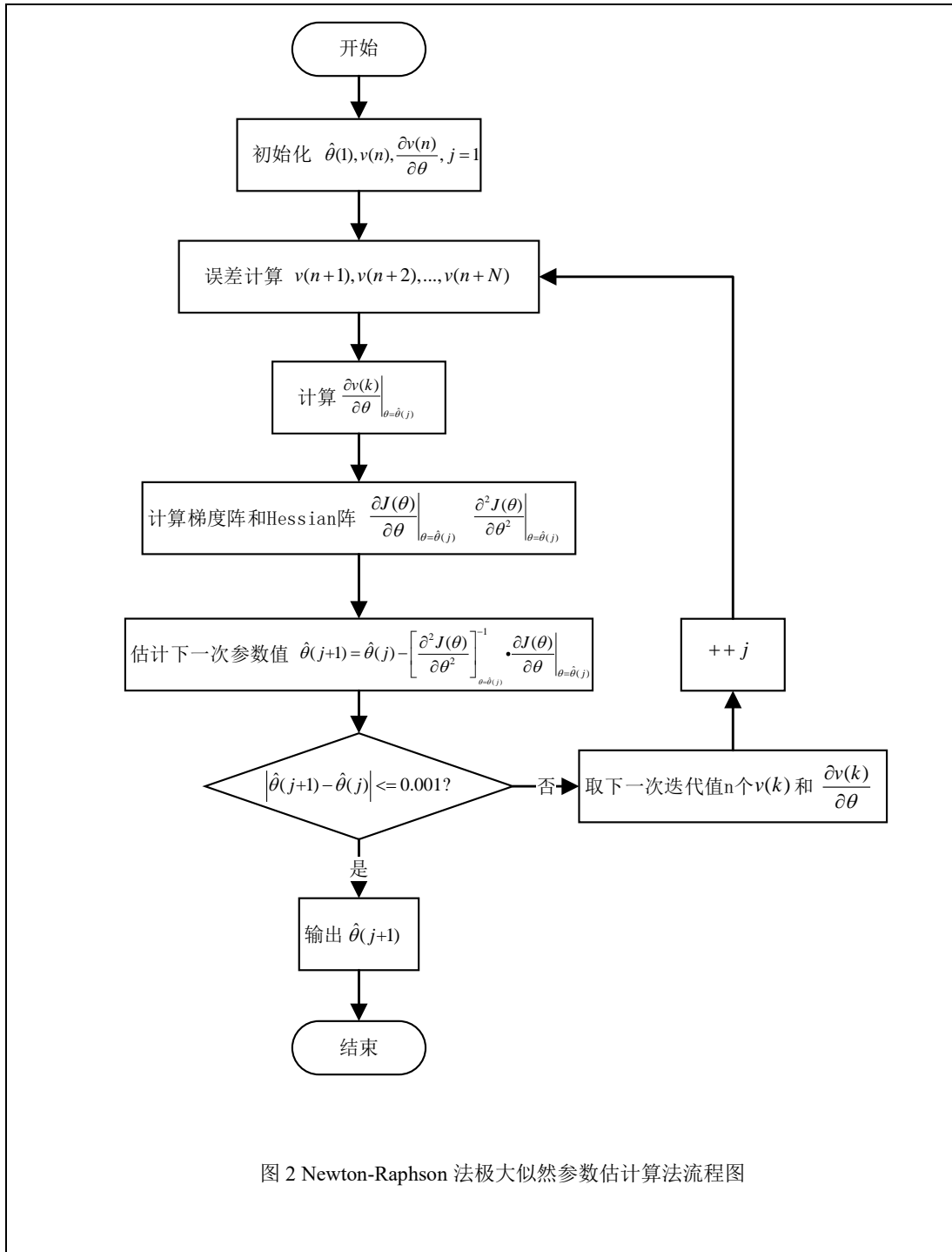


图 2 Newton-Raphson 法极大似然参数估计算法流程图

6. 程序代码

```
1、递推极大似然估计

clc;clear

n=2;

total=1000;

sigma=0.1;

z1=1;z2=1;z3=1;z4=0;

for i=1:total

    x1=xor(z3,z4);

    x2=z1;

    x3=z2;

    x4=z3;

    z(i)=z4;

    if z(i)>0.5

        u(i)=-1;

    else u(i)=1;

    end

    z1=x1;z2=x2;z3=x3;z4=x4;

end

figure(1);

stem(u),grid on

y(1)=0;y(2)=0;

v=sigma*randn(total,1);

y(1)=1;y(2)=0.01;

for k=3:total

    y(k)=0.5*y(k-1)+0.2*y(k-2)+u(k-1)+1.5*u(k-2)+v(k)-0.8*v(k-1)+0.3*v(k-2);

end

theta0=0.001*ones(6,1);

e1(1)=-0.5-theta0(1); e2(1)=-0.2-theta0(2);
```



```

e3(1)=1.0-theta0(3);e4(1)=1.5-theta0(4);

e5(1)=-0.8-theta0(5);e6(1)=0.3-theta0(6);

a_hat(1)=theta0(1);a_hat(2)=theta0(2);

b_hat(1)=theta0(3);b_hat(2)=theta0(4);

c_hat(1)=theta0(5);c_hat(2)=theta0(6);

P0=eye(6,6);

for i=1:n

    yf(i)=0.1;uf(i)=0.1;vf(i)=0.1;

    fai0(i,1)=-yf(i);

    fai0(n+i,1)=uf(i);

    fai0(2*n+i,1)=vf(i);

end

e(1)=1.0;

e(2)=1.0;

for i=n+1:total

    pusai=[-y(i-1);-y(i-2);u(i-1);u(i-2);e(i-1);e(i-2)];

    C=zeros(n*3,n*3);

    Q=zeros(3*n,1);

    Q(1)=-y(i-1);

    Q(n+1)=u(i-1);

    Q(2*n+1)=e(i-1);

    for j=1:n

        C(1,j)=-c_hat(j);

        C(n+1,n+j)=c_hat(j);

        C(2*n+1,2*n+j)=-c_hat(j);

        if j>1

            C(j,j-1)=1.0;

            C(n+j,n+j-1)=1.0;

            C(2*n+j,2*n+j-1)=1.0;

```

```

        end

    end

    fai=C*fai0+Q;

    K=P0*fai*inv(fai'*P0*fai+1);

    P=[eye(6,6)-K*fai']*P0;

    e(i)=y(i)-pusai'*theta0;

    theta=theta0+K*e(i);

    P0=P;

    theta0=theta;

    fai0=fai;

    a_hat(1)=theta0(1);a_hat(2)=theta0(2);

    b_hat(1)=theta0(3);b_hat(2)=theta0(4);

    c_hat(1)=theta0(5);c_hat(2)=theta0(6);

    e1(i)=-0.5-a_hat(1);e2(i)=-0.2-a_hat(2);

    e3(i)=1.0-b_hat(1);e4(i)=1.5-b_hat(2);

    e5(i)=-0.8-c_hat(1);e6(i)=0.3-c_hat(2);

end

figure(2)

plot(e1);

hold on

plot(e2);

hold on

plot(e3);

hold on

plot(e4);

hold on

plot(e5);

hold on

plot(e6);

```

```

title('Parameter Estimation Error');

xlabel('times');

ylabel('error');

hold off

figure(3)

plot(e);

title('Output Error');

xlabel('times');

ylabel('error');

```

2、Newton-Raphson 法极大似然参数估计

```

clc;clear

rng('default');

a1=-0.5;a2=-0.2;b1=1;b2=1.5;d1=-0.8;d2=0.3; %参数

Sigma=1;%均方差

N=500; %一次取样数据

%由一般最小二乘辨识产生初始参数

Length=202; %最后产生 200 组观测值

Reg1=1;Reg2=1;Reg3=1;Reg4=0; %寄存器初值

for i=1:Length

    s1=xor(Reg3,Reg4);

    s2=Reg1;

    s3=Reg2;

    s4=Reg3;

    Output(i)=Reg4;

    if Output(i)>0.5

        U(i)=-3;

    else

        U(i)=3;
    end
end

```

```

end

Reg1=s1;Reg2=s2;Reg3=s3;Reg4=s4;

end

e=Sigma*randn(Length,1); %高斯噪声

Y=zeros(1,Length);

for k=3:Length

    Y(k)=-a1*Y(k-1)-a2*Y(k-2)+b1*U(k-1)+b2*U(k-2)+e(k)+d1*e(k-1)+d2*e(k-2);

end

for i=3:Length

    H(i-2,:)=[-Y(i-1),-Y(i-2),U(i-1),U(i-2)]; %样本系数矩阵

    Z(i-2,1)=Y(i);%输出观测矩阵

end

theta_hat=(H'*H)\(H'*Z) %最小二乘产生估计值

d1=0.1;d2=0.1; %d1,d2 初始值

theta(:,1)=[theta_hat',d1,d2]';%参数估计初值

v(1)=0;v(2)=0; %初始化 v(k)

%初始化参数各阶导

d_a1(1)=0;d_a2(1)=0;d_b1(1)=0;d_b2(1)=0;d_d1(1)=0;d_d2(1)=0;

d_a1(2)=0;d_a2(2)=0;d_b1(2)=0;d_b2(2)=0;d_d1(2)=0;d_d2(2)=0;

j=1;Diff=ones(6,1);

while(max(Diff)>0.001) %当变化不超过 0.001 时结束辨识

    %采集输入输出

    for i=1:N+3

        s1=xor(Reg3,Reg4);

        s2=Reg1;

        s3=Reg2;

        s4=Reg3;

        Output(i)=Reg4;

        if Output(i)>0.5

```

```

        U(i)=-1;

    else

        U(i)=1;

    end

    Reg1=s1;Reg2=s2;Reg3=s3;Reg4=s4; %更新寄存器

end

e=Sigma*randn(N+3,1); %高斯噪声

Y=zeros(N+3,1);

for k=3:N+3

    %Y(k)=-a1*Y(k-1)-a2*Y(k-2)+b1*U(k-1)+b2*U(k-2)+e(k)-d1*e(k-1)+d2*e(k-
2);

    Y(k)=-a1*Y(k-1)-a2*Y(k-2)+b1*U(k-1)+b2*U(k-2)+e(k)-d1*e(k-1)+d2*e(k-
2);

end

Grad=0; %梯度矩阵清零

Hess=0; %Hessian 矩阵清零

for k=3:N+3

    v(k)=Y(k)+theta(:,j) '*[Y(k-1),Y(k-2),-U(k-1),-U(k-2),-v(k-1),-v(k-
2)]';

    d_a1(k)=Y(k-1)-d1*d_a1(k-1)-d2*d_a1(k-2);

    d_a2(k)=Y(k-2)-d1*d_a2(k-1)-d2*d_a2(k-2);

    d_b1(k)=-U(k-1)-d1*d_b1(k-1)-d2*d_b1(k-2);

    d_b2(k)=-U(k-2)-d1*d_b2(k-1)-d2*d_b2(k-2);

    d_d1(k)=-v(k-1)-d1*d_d1(k-1)-d2*d_d1(k-2);

    d_d2(k)=-v(k-2)-d1*d_d2(k-1)-d2*d_d2(k-2);

    d_theta=[d_a1(k),d_a2(k),d_b1(k),d_b2(k),d_d1(k),d_d2(k)]';

    Grad=Grad+v(k)*d_theta;

```

```

        Hess=Hess+d_theta'*d_theta;

    end

    j=j+1;

    theta(:,j)=theta(:,j-1)-(Hess)\Grad;

    Diff=abs((Hess)\Grad);%记录辨识差值

    v(1)=v(N+1);v(2)=v(N+2);

    d_a1(1)=d_a1(N+1);d_a2(1)=d_a2(N+1);

    d_b1(1)=d_b1(N+1);d_b2(1)=d_b2(N+1);

    d_d1(1)=d_d1(N+1);d_d2(1)=d_d2(N+1);

    d_a1(2)=d_a1(N+2);d_a2(2)=d_a2(N+2);

    d_b1(2)=d_b1(N+2);d_b2(2)=d_b2(N+2);

    d_d1(2)=d_d1(N+2);d_d2(2)=d_d2(N+2);

end

theta(:,end)%输出估计参数

figure(1)

plot(1:1000,theta(:,1:1000));

legend('a1','a2','b1','b2','d1','d2');

```

7. 实验结果及分析

1. 极大似然递推参数估计

根据系统差分方程的描述与要求，可生成噪声 $v(k)$ 是均值为零、方差为 0.01 的高斯白噪声序列，在本实验中，选取了长度 1000 作为数据点进行运算。

经过递推演算，最终输出辨识结果与实际结果的误差，具体可见图 3 所示，有图可知，其中系统参数 a_1 , a_2 , b_1 , b_2 ，最终辨识结果较为理想，误差极小，而 d_1 , d_2 （图中图例为 c_1 , c_2 ）最终存在稳态误差。

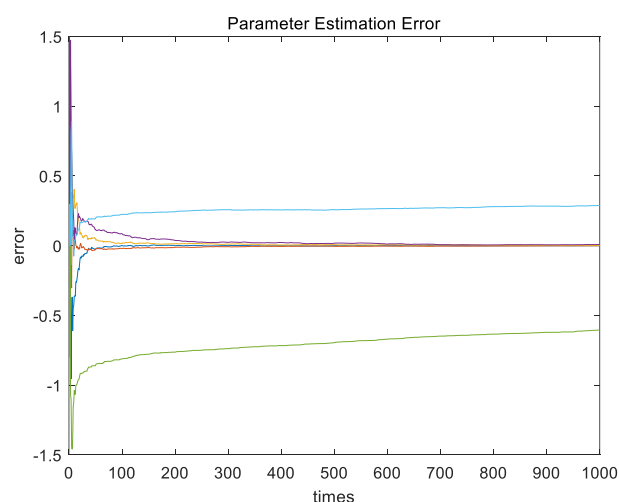


图 3 极大似然递推参数估计误差曲线

如图 4 给出了输出实际值与根据估计参数的模型的输出理论值进行了差值结果，从图中不难发现，当递推到 100 次之后数据误差进入振荡稳定环节，整体趋势明显，系统模型估计较为吻合。

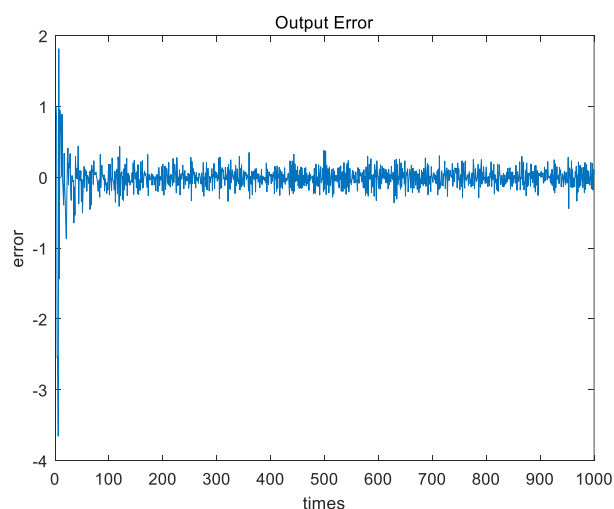


图 4 极大似然递推估计系统输出估计误差

取最后一次递推结果，可获得如下数据： $a_1 = -0.4981$ ， $a_2 = -0.1979$ ， $b_1 = 1.0015$ ， $b_2 = 1.5037$ ， $d_1 = -0.0219$ ， $d_2 = 0.0944$ 具体对比可以详见表 1。

表 1 极大似然递推估计结果表

参数	a_1	a_2	b_1	b_2	d_1	d_2
真值	-0.5	-0.2	1	1.5	-0.8	0.3
估计值	-0.4981	-0.1979	1.0015	1.5037	-0.0219	0.0944
相对误差	0.38%	1.05%	0.15%	0.25%	97.26%	68.53%

参考此表，可知使用递推极大似然法进行参数估计，针对系统本身的参数估计效果十分理想，但的针对误差的参数估计存在十分严重的相对误差。造成此现象的原因主要考虑是噪声本身的方差过于小，其对系统本身的影响 也削弱了，使得在辨识过程中，由于系统本身参数并不准确，从而使得微小作用的误差参数都呈现较大的变动。

2. Newton-Raphson 法极大似然参数估计

图 5 给出了 Newton-Raphson 法的参数估计结果，参数能够在较短的时间内达到收敛状态，其中 d_1 随时间变化较为缓慢，这一现象还可以从图 6 进行观察，针对白噪声参数误差始终存在无法实现最终的收敛或者收敛速度较慢。表 2 给出了最后一次迭代的输出结果与实际结果的对比情况。

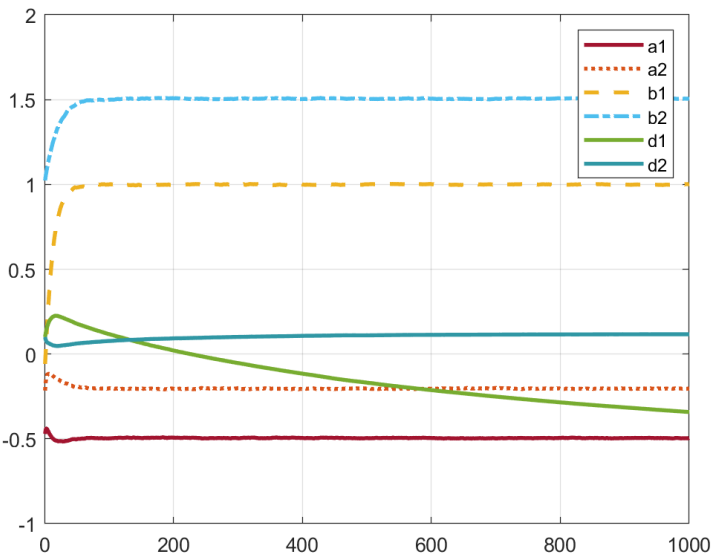


图 5 Newton-Raphson 法极大似然参数估计结果

表 2 Newton-Raphson 法极大似然递推估计结果表

参数	a_1	a_2	b_1	b_2	d_1	d_2
真值	-0.5	-0.2	1	1.5	-0.8	0.3
估计值	-0.5015	-0.1959	1.0007	1.5003	-0.8010	0.2750
相对误差	0.30%	2.05%	0.07%	0.02%	0.13%	8.33%

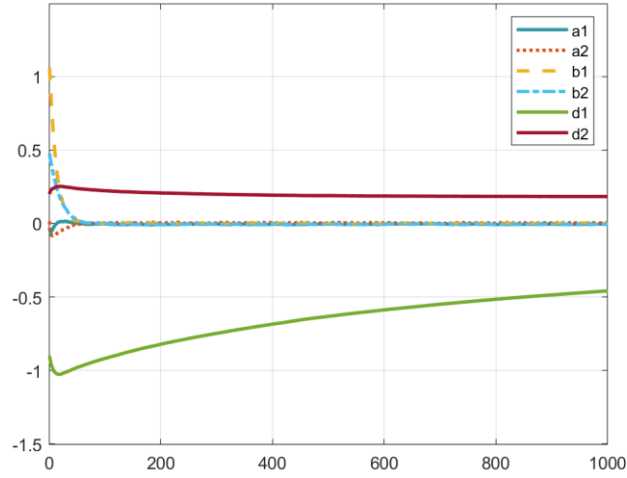


图 6 Newton-Raphson 法极大似然参数估计误差

8. 结论

1、对比最小二乘法，极大似然也能完成对系统参数的估计工作。同时均使用了递归算法使得两种方案都能进行在线辨识以及运算速度较快。但其根本原理有所区别。最小二乘法主要利用随机误差的累计和最小来进行参数的估计，而极大似然则从统计学角度出发，利用统计学特性来完成参数的估计。更进一步的原理则是根据一组确定的随机序列 Y_N ，设法找到参数估计值 $\hat{\theta}_{ML}$ ，它使得随机变量 y 在 $\hat{\theta}_{ML}$ 条件下的概率密度函数最大可能地逼近随机变量 y 在 θ （真值）条件下的概率密度函数，即：

$$p(y|\hat{\theta}_{ML}) \rightarrow p(y|\theta)$$

从辨识结果的角度出发，其具有良好的辨识结果，对系统本身的参数辨识也具有好的特性，当假设数据满足正态分布函数的特征性，在这种情况下，极大似然法和最小二乘法估计相同。极大似然法通过选择参数，让已知数据在某种意义上最有可能出现，这个某种意义上指的就是最大似然估计，而似然函数指的就是数据的概率分布函数。和最小二乘法不同的是，最大似然法需要提前知道这个数据的分布函数，这在实践中是很难的，所以这也使得工程中更多对采取最小二乘法进行系统参数的估计。

极大似然递推参数估计法每迭代一次即可估计一次参数，其计算量较小，可实时进行估计，但由于噪声的干扰，辨识结果误差并不会收敛于 0，会降低结果的精确度，并且与噪声的方差大小有关，尤其是对于 d_1 、 d_2 来说。

2、在本实验中主要完成了基于 Newton-Raphson 法进行参数辨识，其辨识结果与实际结果十分接近，并且迭代次数在较低的情况下即可将结果进行收敛，其中初始值若使用一般最小二乘法输出的结果后能够更快收敛于稳定，但若使用一个未知的数据，最终也能完成收敛。Newton-Raphson 的应用缺点主要是源自于其每次迭代都需要 N 组数据的运算，在运算量上大于其他算法，但是其估计结果也是较为优秀的，在针对实时性不高但对辨识精度有需求的系统中可以尝试应用。