

# 实验一 三种最小二乘方法估计参数值

## 实验报告

中国地质大学（武汉）

自动化学院

专    业： 自动化

班    级： 231212

姓    名： 廖志豪

日期： 2023 年 12 月 7 日

# 1 实验题目

设单输入-单输出系统的差分方程为：

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

$$V(k) = c_1 v(k) + c_2 v(k-1) + c_3 v(k-2)$$

取真值： $a_1 = 1.6$ ,  $a_2 = 0.7$ ,  $b_1 = 1.0$ ,  $b_2 = 0.4$ ,  $c_1 = 1.1$ ,  $c_2 = 1.4$ ,  $c_3 = 0.3$ ，输入信号采用 4 阶 M 序列，幅值为 1。当  $v(k)$  的均值为 0，方差分别为 0.1 和 0.5 的高斯噪声时，分别用一般最小二乘法、递推最小二乘法和增广递推最小二乘法估计参数  $\theta$ 。并通过对三种方法的辨识结果的分析 and 比较，说明上述三种参数辨识方法的优缺点。

## 2 实验目的

1. 掌握最小二乘法估计参数值的原理；
2. 掌握三种最小二乘方法的实现和估计参数；
3. 掌握三种最小二乘方法的优缺点，能够选择适合的方法估计参数值。

## 3 实验主要原理

### 3.1 最小二乘法简介

高斯的最小二乘基本原理：成对等精度地测得一组数据  $t_i, y_i (i = 1, 2, \dots, m)$ ，试找出一条最佳的拟合曲线，使得这条拟合曲线上的各点的值与测量值差的平方在所有拟合曲线中最小。即对于系统的观测输出  $y(k)$ ：

$$y(k) = x(k) + v(k)$$

其中， $x(k)$  表示系统的真实输出， $v(k)$  表示系统输出的测量误差，则基于最小二乘法，应使得误差平方之和：

$$J = \sum_{k=1}^m w(k) |y(k) - x(k)|^2$$

的值达到最小，其中  $w(k)$  代表权重值。

### 3.2 一般最小二乘法

对于单输入-单输出系统：

$$y(k) = - \sum_{i=1}^n a_i y(k-i) + \sum_{i=0}^n b_i u(k-i) + \xi(k)$$

其中  $\xi(k)$  代表测量噪声。对该系统分别测出  $n + N$  个输入输出值：

$$y(1), y(2), \dots, y(n+N)$$

$$u(1), u(2), \dots, u(n+N)$$

则可以写出  $N$  个方程：

$$\begin{aligned} y(n+1) &= -\sum_{i=1}^n a_i y(n+1-i) + \sum_{i=0}^n b_i u(n+1-i) + \xi(n+1) \\ y(n+2) &= -\sum_{i=1}^n a_i y(n+2-i) + \sum_{i=0}^n b_i u(n+2-i) + \xi(n+2) \\ &\vdots \\ y(n+N) &= -\sum_{i=1}^n a_i y(n+N-i) + \sum_{i=0}^n b_i u(n+N-i) + \xi(n+N) \end{aligned}$$

定义以下变量：

$$y = \begin{bmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(n+N) \end{bmatrix}, \quad \theta = \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_0 \\ \vdots \\ b_n \end{bmatrix}, \quad \xi = \begin{bmatrix} \xi(n+1) \\ \xi(n+2) \\ \vdots \\ \xi(n+N) \end{bmatrix}$$

$$\Phi = \begin{bmatrix} -y(n) & \dots & -y(1) & u(n+1) & \dots & u(1) \\ -y(n+1) & \dots & -y(2) & u(n+2) & \dots & u(2) \\ \vdots & & \vdots & \vdots & \vdots & \vdots \\ -y(n+N-1) & \dots & -y(N) & u(n+N) & \dots & u(N) \end{bmatrix}$$

则可以得到：

$$y = \Phi\theta + \xi$$

定义残差变量：

$$e = y - \hat{y} = y - \Phi\hat{\theta}$$

最小二乘估计要求残差的平方和最小，即按照指标函数：

$$J = e^T e = (y - \Phi\hat{\theta})^T (y - \Phi\hat{\theta})$$

为最小来确定估值  $\hat{\theta}$ 。求  $J$  对  $\hat{\theta}$  的偏导数并令其等于 0 可得  $\theta$  的最小二乘估计：

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T y$$

$J$  为极小值的充分条件是：

$$\frac{\partial^2 J}{\partial \hat{\theta}^2} = 2\Phi^T \Phi > 0$$

即矩阵  $\Phi^T \Phi$  为正定矩阵。

### 3.3 递推最小二乘法

递推最小二乘法采用参数递推估计，其大致思路为：

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

递推最小二乘算法的计算公式如下：

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

$$\begin{aligned} P_{m+1} &= P_m - P_m h^T(m+1)[w^{-1}(m+1) + h(m+1)P_m h^T(m+1)]^{-1}h(m+1)P_m \\ &= P_m - K_{m+1}K_{m+1}^T[w^{-1}(m+1) + h(m+1)P_m h^T(m+1)] \end{aligned}$$

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

其中：

- $\hat{\theta}_m$ ：前一时刻的参数估值
- $z(m+1)$ ：当前时刻的量测值
- $h(m+1)\hat{\theta}_m$ ：在上一量测的基础上对在  $(m+1)$  的预测
- $z(m+1) - h(m+1)\hat{\theta}_m$ ：预测误差，又称为新息
- $K_{m+1}$ ：修正的增益矩阵

此外，关于递归参数估计的停止条件，可以指定迭代的次数，也可以指定相邻两次参数估计值的偏差大小  $\varepsilon$ ：

$$\forall i \max \left| \frac{\hat{\theta}_i(m+1) - \hat{\theta}_i(m)}{\hat{\theta}_i(m)} \right| < \varepsilon$$

### 3.4 增广递推最小二乘法

当噪声均值为 0 时，最小二乘参数估计法为无偏估计；当噪声均值不为 0 时，最小二乘参数估计法为有偏估计。为了解决最小二乘参数估计的有偏性，引入了增广最小二乘参数估计法，考虑噪声的影响，对噪声项参数也进行辨识。其系统模型结构如下图所示：

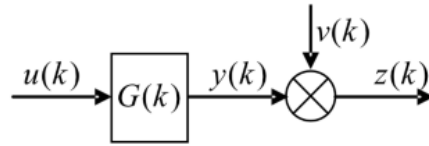


图 1: SISO 系统灰箱结构

其中， $y(k)$ ,  $z(k)$  可表示为：

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

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

其中  $v(k)$  为非白噪声序列。定义以下变量：

$$\begin{aligned} h(k) &= [-z(k-1), -z(k-2), \dots, -z(k-n), u(k-1), u(k-2), \\ &\quad \dots, u(k-n), \hat{v}(k-1), \dots, \hat{v}(k-n)] \end{aligned}$$

$$\theta = [a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n, c_1, \dots, c_n]^T$$

则  $z(k)$  也可以表示为:

$$z(k) = h(k)\theta + v(k)$$

类似于递推最小二乘算法, 下面列出增广递推最小二乘算法的计算公式:

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

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

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

## 4 实验对象和参数

**实验对象** 单输入-单输出系统, 差分方程为:

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

$$V(k) = c_1 v(k) + c_2 v(k-1) + c_3 v(k-2)$$

**实验参数** 取真值:  $a_1 = 1.6$ ,  $a_2 = 0.7$ ,  $b_1 = 1.0$ ,  $b_2 = 0.4$ ,  $c_1 = 1.1$ ,  $c_2 = 1.4$ ,  $c_3 = 0.3$ , 输入信号采用 4 阶 M 序列, 幅值为 1。  $v(k)$  的均值为 0, 方差分别为 0.1 和 0.5 的高斯噪声。

## 5 程序框图

### 5.1 一般最小二乘法程序框图

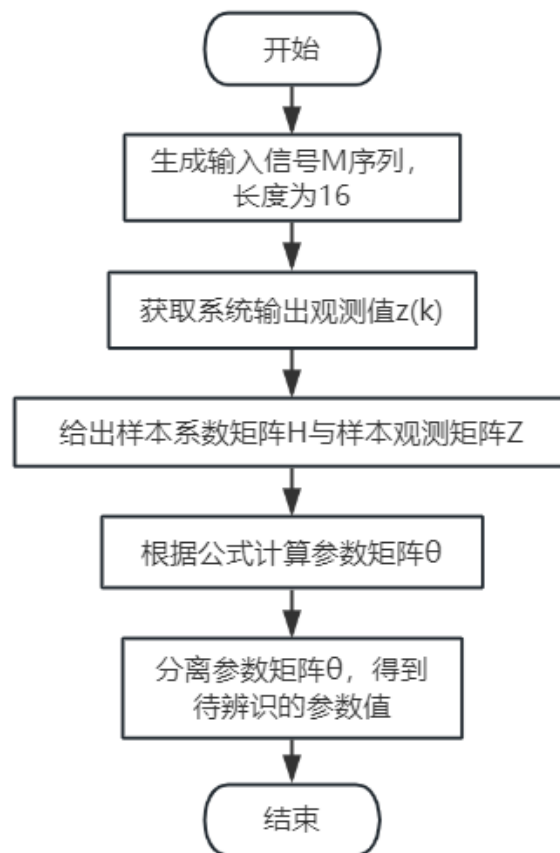


图 2: 一般最小二乘法-程序框图

## 5.2 递推最小二乘法程序框图

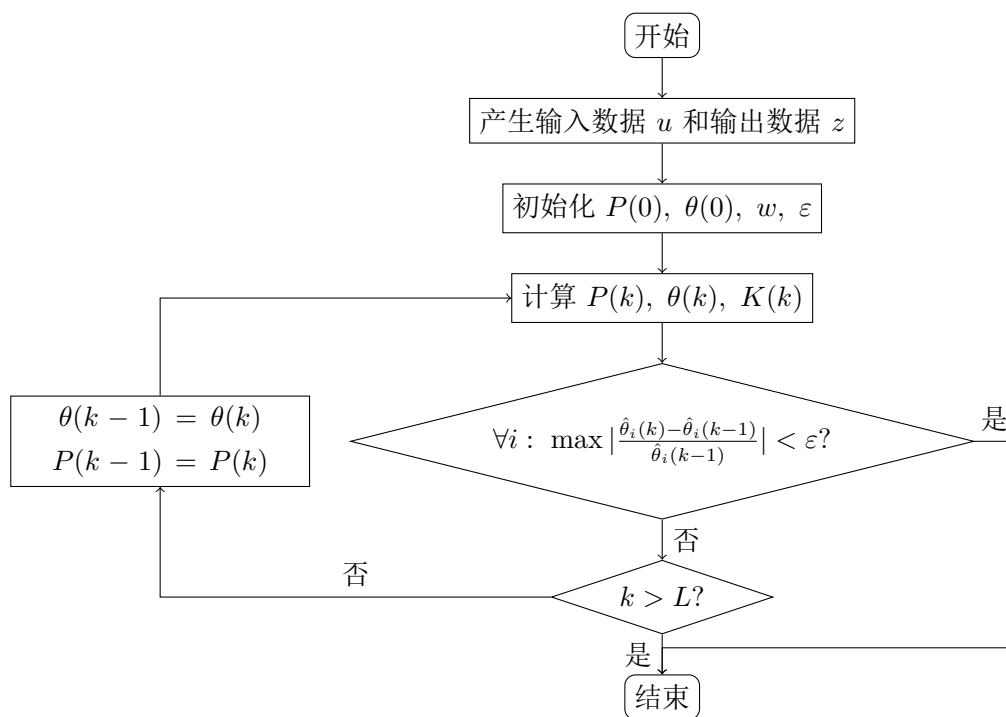


图 3: 递推最小二乘法-程序框图

### 5.3 增广递推最小二乘法程序框图

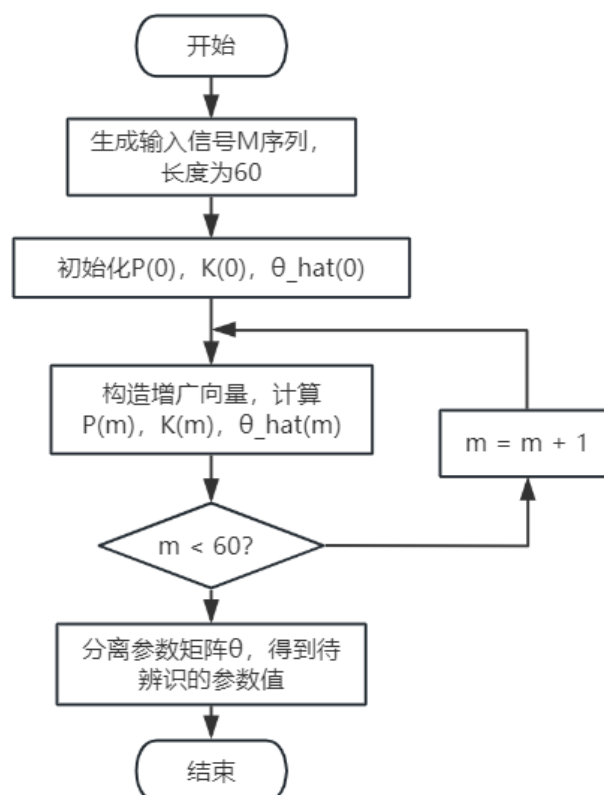


图 4: 增广递推最小二乘法-程序框图

## 6 程序代码

### 6.1 一般最小二乘法

```
1 % 一般最小二乘法
2 clear all
3 close all
4 clc
5
6 % 输入输出序列的长度
7 len = 16;
8 % 均值为 0, 方差为 0.1 的高斯噪声
9 v = normrnd(0, 0.1, 1, len);
10 % 均值为 0, 方差为 0.5 的高斯噪声
11 % v = normrnd(0, 0.5, 1, len);
12
13 %% M序列产生程序 4阶M序列 幅值为 1
14 L = len; % M序列的周期
15 y1 = 1; y2 = 1; y3 = 1; y4 = 0; %四个移位寄存器的输出初始值
```



```

16 for i = 1:L
17     x1 = xor(y3, y4);
18     x2 = y1;
19     x3 = y2;
20     x4 = y3;
21     y(i) = y4;
22     if y(i) > 0.5, u(i) = -1;
23     else u(i) = 1;
24     end
25     y1 = x1; y2 = x2; y3 = x3; y4 = x4;
26 end
27 figure
28 stem(u), grid on
29 title('输入信号M序列')
30
31 %% 最小二乘辨识程序
32 a1 = 1.6;
33 a2 = 0.7;
34 b1 = 1.0;
35 b2 = 0.4;
36 c1 = 1.1;
37 c2 = 1.4;
38 c3 = 0.3;
39
40 z = zeros(1, len); %定义输出观测值的长度
41 for k = 3:16
42     %  $V(k) = c1*v(k) + c2*v(k-1) + c3*v(k-2);$ 
43     V(k) = v(k); %一般最小二乘用白噪声
44     z(k) = -a1*z(k-1) - a2*z(k-2) + b1*u(k-1) + b2*u(k-2) + V(k);
45 end
46 figure(2)
47 plot([1:len], z)
48 title('输出观测值')
49 figure(3)
50 stem(z), grid on
51 title('输出观测值z的经线图形')
52
53 %% 参数辨识
54 %给样本系数矩阵
55 H=[-z(2) -z(1) u(2) u(1);-z(3) -z(2) u(3) u(2);
56     -z(4) -z(3) u(4) u(3);-z(5) -z(4) u(5) u(4);
57     -z(6) -z(5) u(6) u(5);-z(7) -z(6) u(7) u(6);

```

```

58     -z(8) -z(7) u(8) u(7); -z(9) -z(8) u(9) u(8);
59     -z(10) -z(9) u(10) u(9); -z(11) -z(10) u(11) u(10);
60     -z(12) -z(11) u(12) u(11); -z(13) -z(12) u(13) u(12);
61     -z(14) -z(13) u(14) u(13); -z(15) -z(14) u(15) u(14)];
62 %给出样本观测矩阵
63 Z = [z(3); z(4); z(5); z(6); z(7); z(8); z(9); z(10); z(11);
64      z(12); z(13); z(14); z(15); z(16)];
65 %计算参数
66 c = inv(H'*H)*H'*Z;
67 %分离参数
68 a1 = c(1), a2 = c(2), b1 = c(3), b2 = c(4)

```

## 6.2 递推最小二乘法

```

1 % 递推最小二乘法
2 clear all
3 close all
4 clc
5 %%产生 $N(0,1)$ 正态分布的随机噪声
6 % randn('seed',100);
7 % v=randn(1,60);
8
9 % 输入输出序列的长度
10 len = 60;
11 % 均值为0, 方差为0.1的高斯噪声
12 v = normrnd(0, 0.1, 1, len);
13 % 均值为0, 方差为0.5的高斯噪声
14 % v = normrnd(0, 0.5, 1, len);
15
16 %% M序列产生程序 4阶M序列 幅值为1
17 L = len - 1; % M序列的周期
18 y1 = 1; y2 = 1; y3 = 1; y4 = 0; %四个移位寄存器的输出初始值
19 for i = 1:L
20     x1 = xor(y3, y4);
21     x2 = y1;
22     x3 = y2;
23     x4 = y3;
24     y(i) = y4;
25     if y(i) > 0.5, u(i) = -1;
26     else u(i) = 1;
27     end
28     y1 = x1; y2 = x2; y3 = x3; y4 = x4;
29 end

```

```

30 figure
31 stem(u), grid on
32 title('输入信号M序列')
33
34 %% 一般递推最小二乘辨识程序
35 a1 = 1.6;
36 a2 = 0.7;
37 b1 = 1.0;
38 b2 = 0.4;
39 c1 = 1.1;
40 c2 = 1.4;
41 c3 = 0.3;
42
43 z = zeros(1, len); %定义输出观测值的长度
44 for k = 3:len
45 %       $V(k) = c1*v(k) + c2*v(k-1) + c3*v(k-2)$ ; % 有色噪声
46     V(k) = v(k)
47     z(k) = -a1*z(k-1) - a2*z(k-2) + b1*u(k-1) + b2*u(k-2) + V(k);
48 end
49 figure(2)
50 plot([1:len], z)
51 title('输出观测值')
52 figure(3)
53 stem(z), grid on
54 title('输出观测值z的经线图形')
55
56 %% 递推最小二乘辨识
57 c0 = [0.001 0.001 0.001 0.001]';
58 p0 = 10^3 * eye(4, 4);
59 E = 0.000000005; %相对误差
60 c = [c0, zeros(4, len - 1)]; %被辨识参数矩阵的初始值及大小
61 e = zeros(4, len); %相对误差的初始值及大小
62 lamt = 1;
63 % 迭代至k=16
64 for k = 3:len
65     h1 = [-z(k-1), -z(k-2), u(k-1), u(k-2)]';
66     k1 = p0 * h1 * inv(h1'*p0*h1 + 1*lamt); %求出K的值
67     new = z(k) - h1'*c0;
68     c1 = c0 + k1*new; %求被辨识参数c
69     p1 = 1/lamt*(eye(4) - k1*h1')*p0;
70     e1 = (c1 - c0)./c0; %求参数当前值与上一次的值的差值
71     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

```

```

72     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
73     c0 = c1;%新获得的参数作为下一次递推的旧参数
74     p0 = p1;
75     if norm(e1) <= E
76         break;%若参数收敛满足要求，终止计算
77     end
78 end
79 %分离参数
80 a1 = c(1, :); a2 = c(2, :); b1 = c(3, :); b2 = c(4, :);
81 ea1 = e(1, :); ea2 = e(2, :); eb1 = e(3, :); eb2 = e(4, :);
82
83 figure(2);
84 i = 1:len;
85 %画出辨识结果
86 plot(i, a1, 'k', i, a2, 'b', i, b1, 'r', i, b2, 'g')
87 legend('a1', 'a2', 'b1', 'b2');
88 title('递推最小二乘参数辨识')
89
90 figure(3);
91 i = 1:len;
92 %画出辨识结果的收敛情况
93 plot(i, ea1, 'k', i, ea2, 'b', i, eb1, 'r', i, eb2, 'g')
94 legend('a1', 'a2', 'b1', 'b2');
95 title('辨识精度')

```

### 6.3 增广最小二乘法

```

1 % 增广最小二乘法
2 clear all
3 close all
4 clc
5 % 输入输出序列的长度
6 len = 60;
7 % 均值为 0，方差为 0.1 的高斯噪声
8 % v = normrnd(0, 0.1, 1, len);
9 % 均值为 0，方差为 0.5 的高斯噪声
10 v = normrnd(0, 0.5, 1, len);
11
12 %% M序列、噪声信号产生及其显示程序
13 L = 60;%四位移位寄存器产生的M序列的周期
14 y1 = 1; y2 = 1; y3 = 1; y4 = 0;
15 for i = 1:L
16     x1 = xor(y3, y4);

```

```

17     x2 = y1;
18     x3 = y2;
19     x4 = y3;
20     y(i) = y4;
21     if y(i) > 0.5, u(i) = -1;
22     else u(i) = 1;
23     end
24     y1 = x1; y2 = x2; y3 = x3; y4 = x4;
25 end
26 %画出M序列输入信号
27 figure(1);
28 stem(u), grid on
29
30 %% 增广递推最小二乘辨识
31 a1 = 1.6;
32 a2 = 0.7;
33 b1 = 1.0;
34 b2 = 0.4;
35 c1 = 1.1;
36 c2 = 1.4;
37 c3 = 0.3;
38 z(2) = 0; z(1) = 0;
39 %直接给出被辨识参数的初始值, 即一个充分小的实向量
40 theat0 = [0.001 0.001 0.001 0.001 0.001 0.001 0.001]';
41 p0 = 10^4 * eye(7, 7); %初始状态 P0
42 theat = [theat0, zeros(7, len - 1)]; %被辨识参数矩阵的初始值及大小
43 for k = 3:len
44     V(k) = c1*v(k) + c2*v(k-1) + c3*v(k-2);
45     z(k) = -a1*z(k-1) - a2*z(k-2) + b1*u(k-1) + b2*u(k-2) + V(k);
46
47     h1 = [-z(k-1), -z(k-2), u(k-1), u(k-2), ...
48           v(k), v(k-1), v(k-2)]';
49     x = h1'*p0*h1 + 1;
50     x1 = inv(x);
51     k1 = p0*h1*x1; %K
52     d1 = z(k) - h1'*theat0;
53     theat1 = theat0 + k1*d1; %辨识参数 c
54     theat0 = theat1; %给下一次用
55     theat(:, k) = theat1; %把辨识参数 c 列向量加入辨识参数矩阵
56     p1 = p0 - k1*k1'*[h1'*p0*h1 + 1]; %find p(k)
57     p0 = p1; %给下次用
58 end %循环结束

```

```

59
60 %分离变量
61 a1v = theat(1, :); a2v = theat(2, :);
62 b1v = theat(3, :); b2v = theat(4, :);
63 c1v = theat(5, :); c2v = theat(6, :); c3v = theat(7, :);
64 i = 1:60;
65
66 % 输出
67 figure(2);
68 plot(i, z)
69
70 figure(3)
71 %参数估计值
72 plot(i, a1v, 'r', i, a2v, 'b', i, b1v, 'k', i, b2v, 'y', ...
73      i, c1v, 'g', i, c2v, 'c', i, c3v, 'm')
74 legend('a1v', 'a2v', 'b1v', 'b2v', 'c1v', 'c2v', 'c3v');
75 title('参数估计值') %标题
76
77 % 参数估计误差
78 a1e = a1 - a1v;
79 a2e = a2 - a2v;
80 b1e = b1 - b1v;
81 b2e = b2 - b2v;
82 c1e = c1 - c1v;
83 c2e = c2 - c2v;
84 c3e = c3 - c3v;
85 figure(4)
86 plot(i, a1e, 'r', i, a2e, 'b', i, b1e, 'k', i, b2e, 'y', i, ...
87      c1e, 'g', i, c2e, 'c', i, c3e, 'm')
88 legend('a1e', 'a2e', 'b1e', 'b2e', 'c1e', 'c2e', 'c3e');
89 title('参数估计误差') %标题

```

## 7 实验结果及分析

采用均值为 0，方差分别为 0.1 和 0.5 的高斯噪声，比较实验结果。

### 7.1 一般最小二乘法

高斯噪声：均值为 0，方差为 0.1

采用四阶 M 序列作为系统输入，幅值为 1，长度为 16:

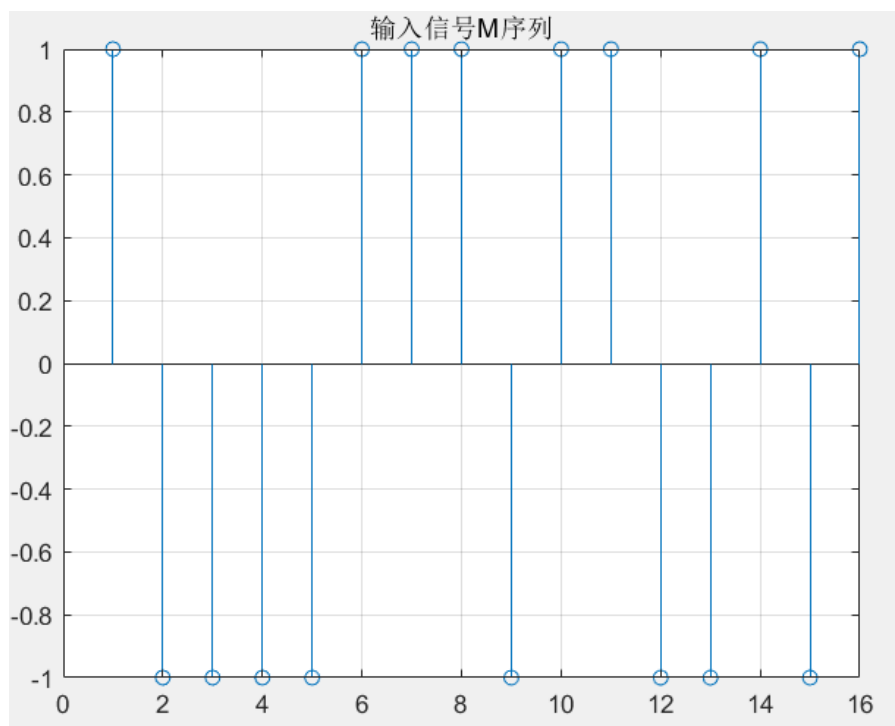


图 5: 一般最小二乘-输入信号 M 序列

系统的输出观测值如下图所示:

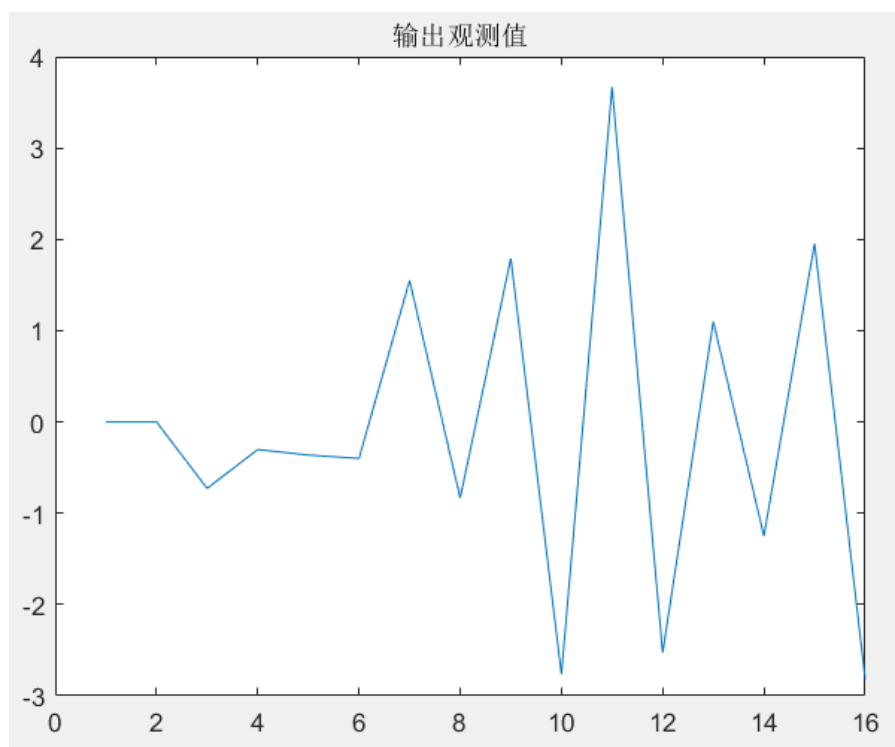


图 6: 一般最小二乘-输出观测值 (噪声方差: 0.1)

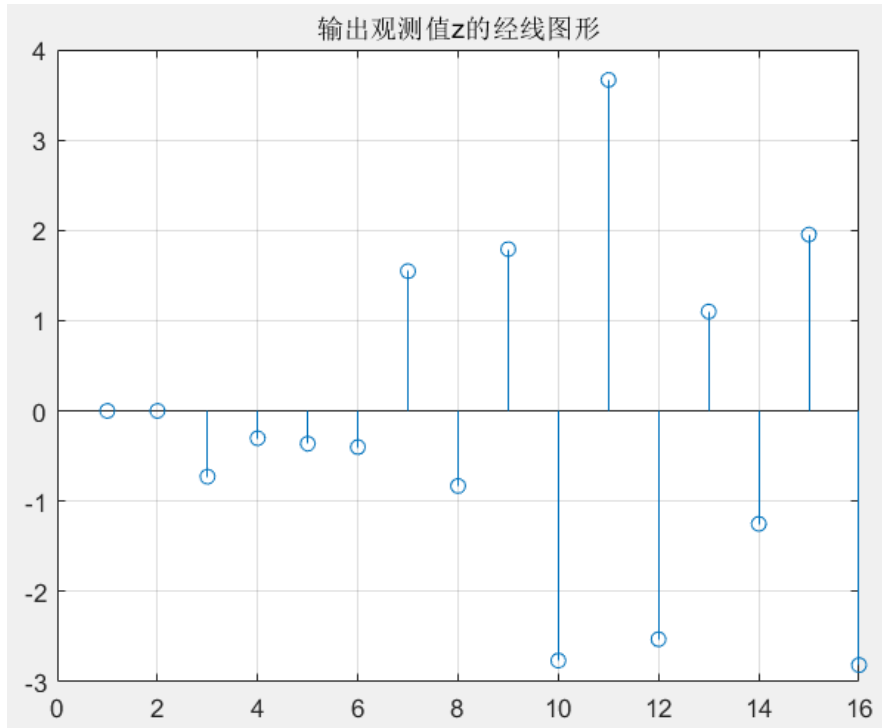


图 7: 一般最小二乘-输出观测值的经线图形 (噪声方差: 0.1)

最终的参数辨识结果如下:

辨识参数	实际值	辨识值	误差百分比 (%)
$a_1$	1.6	1.6346	2.16
$a_2$	0.7	0.7463	6.61
$b_1$	1.0	0.9733	2.67
$b_2$	0.4	0.4069	1.72

表 1: 一般最小二乘法辨识结果 (噪声方差: 0.1)

### 结果分析

由于噪声项产生的随机性, 每次运行程序所得结果均有不同。由实验结果可以看出, 本次参数辨识的误差百分比基本在  $\pm 5\%$  以内, 说明在噪声方差为 0.1 的条件下, 采用一般最小二乘法进行参数辨识的结果是相对可靠的。

### 高斯噪声: 均值为 0, 方差为 0.5

采用四阶 M 序列作为系统输入, 幅值为 1, 长度为 16, 同上。系统的输出观测值如下图所示:



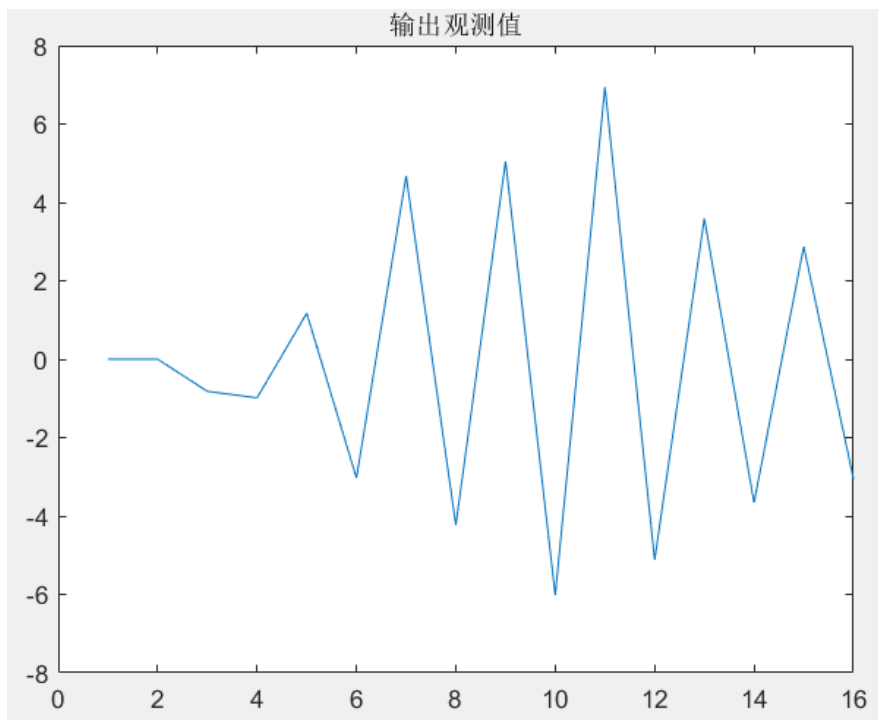


图 8: 一般最小二乘-输出观测值 (噪声方差 0.5)

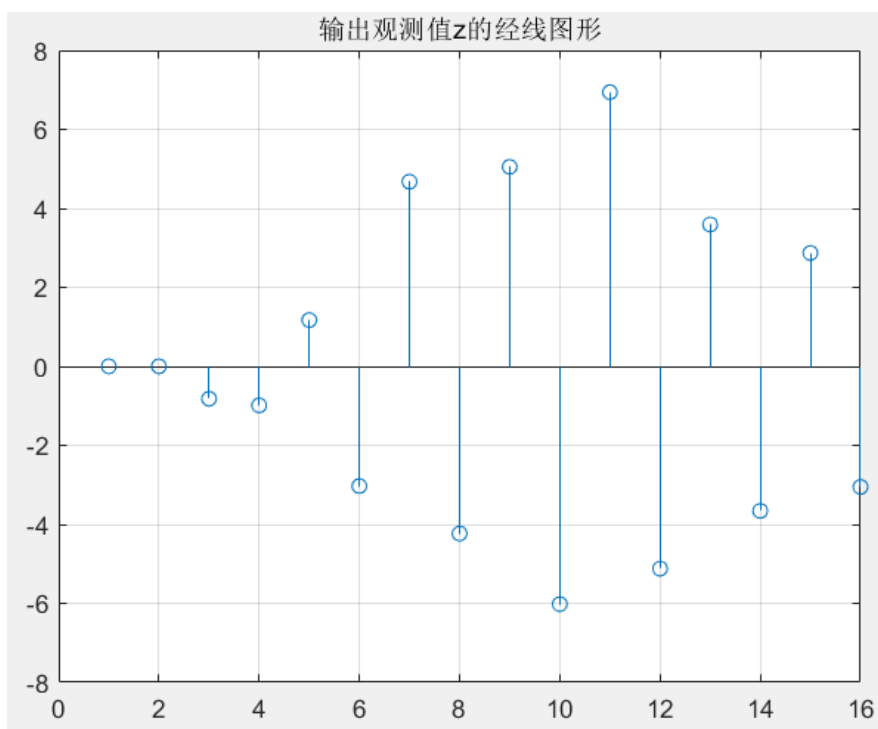


图 9: 一般最小二乘-输出观测值的经线图形 (噪声方差 0.5)

最终的参数辨识结果如下:

辨识参数	实际值	辨识值	误差百分比 (%)
$a_1$	1.6	1.4706	8.09
$a_2$	0.7	0.5299	24.3
$b_1$	1.0	1.0543	5.43
$b_2$	0.4	0.3322	16.95

表 2: 一般最小二乘法辨识结果 (噪声方差: 0.5)

### 结果分析

对比噪声方差为 0.1 的实验结果可以看出, 当噪声方差增大时, 使用一般最小二乘法辨识参数的结果出现了较大的误差。因此在这种条件下, 一般最小二乘参数辨识方法不再适用。

## 7.2 递推最小二乘法

高斯噪声: 均值为 0, 方差为 0.1

采用 M 序列作为系统输入, 幅值为 1, 长度为 60:

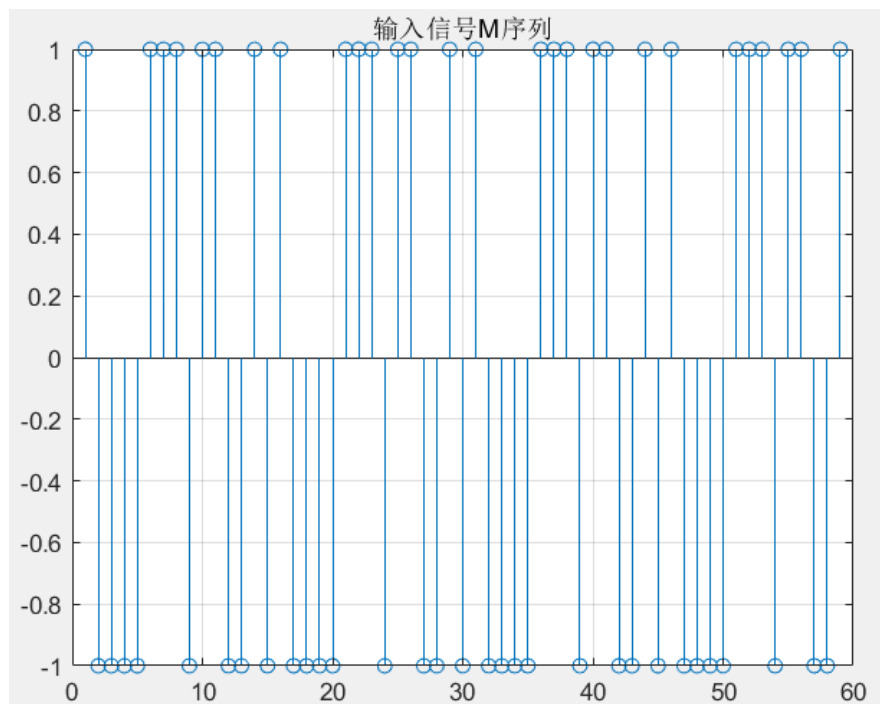


图 10: 递推最小二乘-输入信号 M 序列

迭代过程中参数辨识值及辨识精度如下图所示:

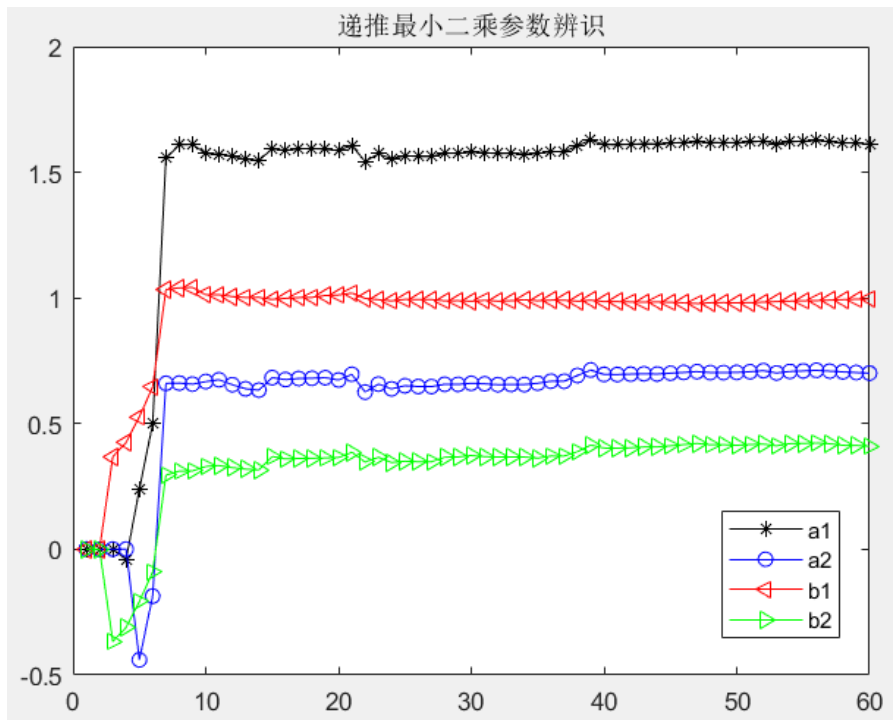


图 11: 递推最小二乘-参数辨识结果 (噪声方差 0.1)

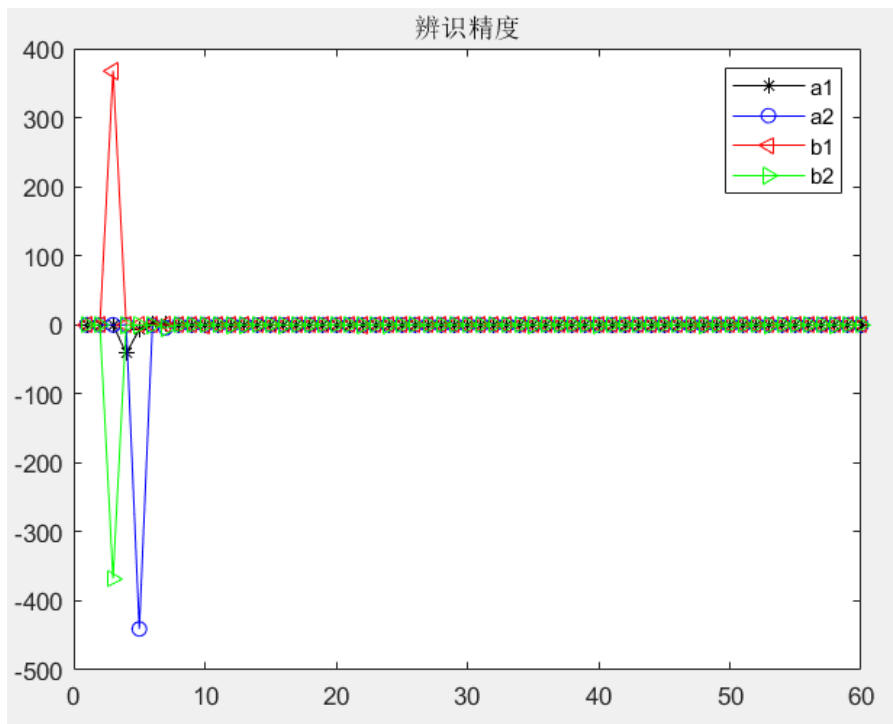


图 12: 递推最小二乘-辨识精度 (噪声方差 0.1)

最终的参数辨识结果如下：

辨识参数	实际值	辨识值	误差百分比 (%)
$a_1$	1.6	1.6163	1.02
$a_2$	0.7	0.7129	0.0184
$b_1$	1.0	1.0003	0.03
$b_2$	0.4	0.4245	6.12

表 3: 递推最小二乘法辨识结果（噪声方差：0.1）

高斯噪声：均值为 0，方差为 0.5

采用 M 序列作为系统输入，幅值为 1，长度为 60，同上。迭代过程中参数辨识值及辨识精度如下图所示：

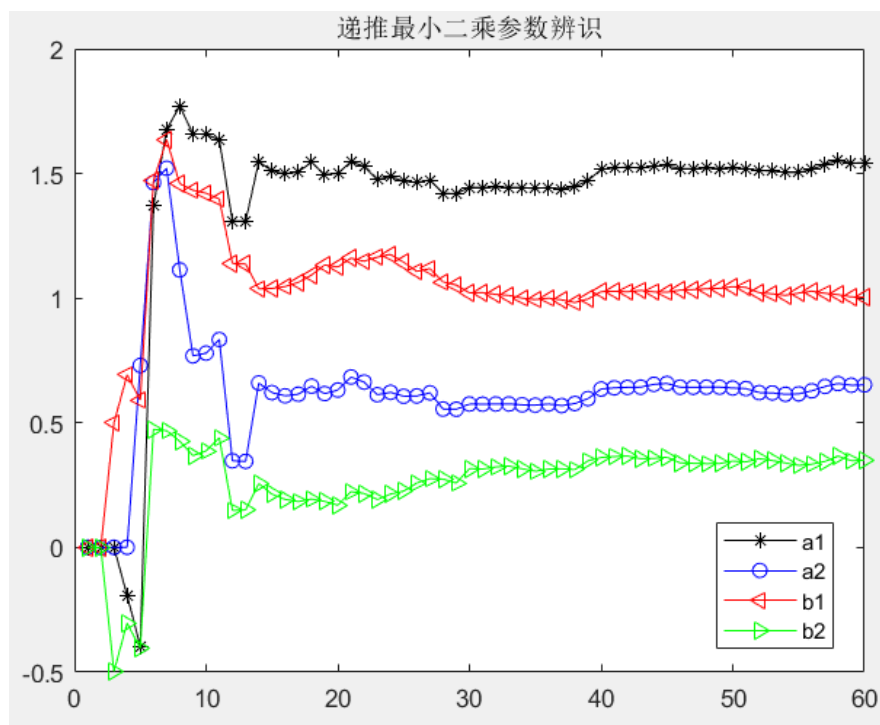


图 13: 递推最小二乘-参数辨识结果（噪声方差 0.5）

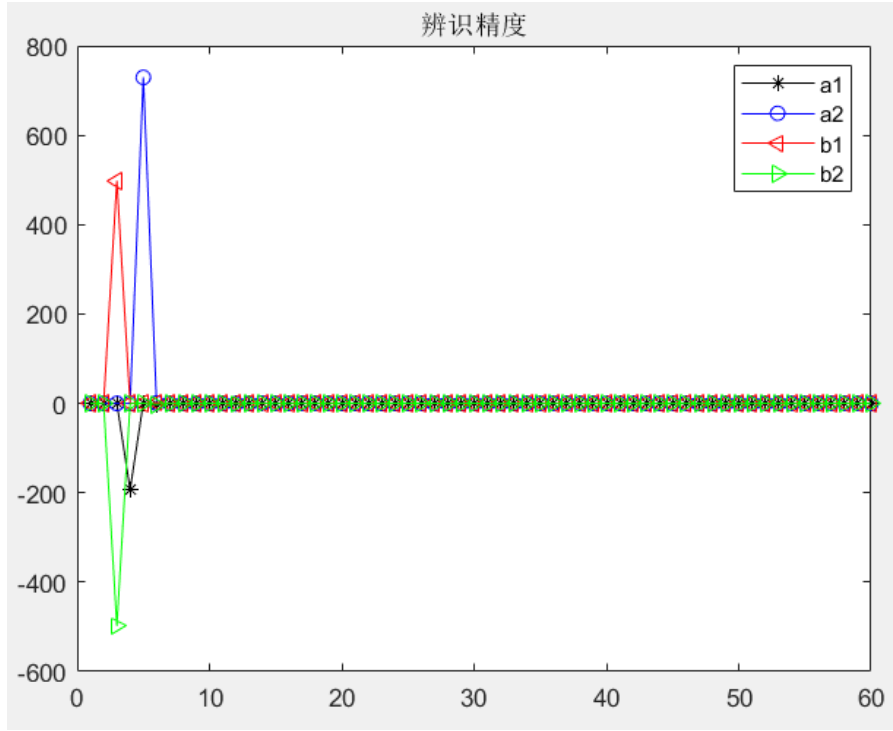


图 14: 递推最小二乘-辨识精度（噪声方差 0.5）

最终的参数辨识结果如下：

辨识参数	实际值	辨识值	误差百分比 (%)
$a_1$	1.6	1.3634	14.79
$a_2$	0.7	0.4394	0.3723
$b_1$	1.0	0.9314	6.86
$b_2$	0.4	0.2035	49.13

表 4: 递推最小二乘法辨识结果（噪声方差：0.5）

### 结果分析

与一般最小二乘参数辨识方法的实验结果类似，当噪声方差较小（0.1）时，递推最小二乘参数辨识方法的参数估计误差大致在  $\pm 5\%$  以内；而当噪声方差较大（0.5）时，最终的参数估计值出现了很大的误差。实验结果说明在噪声干扰较大的情况下，使用递推最小二乘方法进行参数辨识就不再适用了。

另外需要指出的是，一般最小二乘和递推最小二乘参数辨识方法均无法对噪声项参数进行辨识，这是两种方法在适用性上存在的另一局限性。

### 7.3 增广递推最小二乘法

高斯噪声：均值为 0，方差为 0.1

采用 M 序列作为系统输入，幅值为 1，长度为 60：

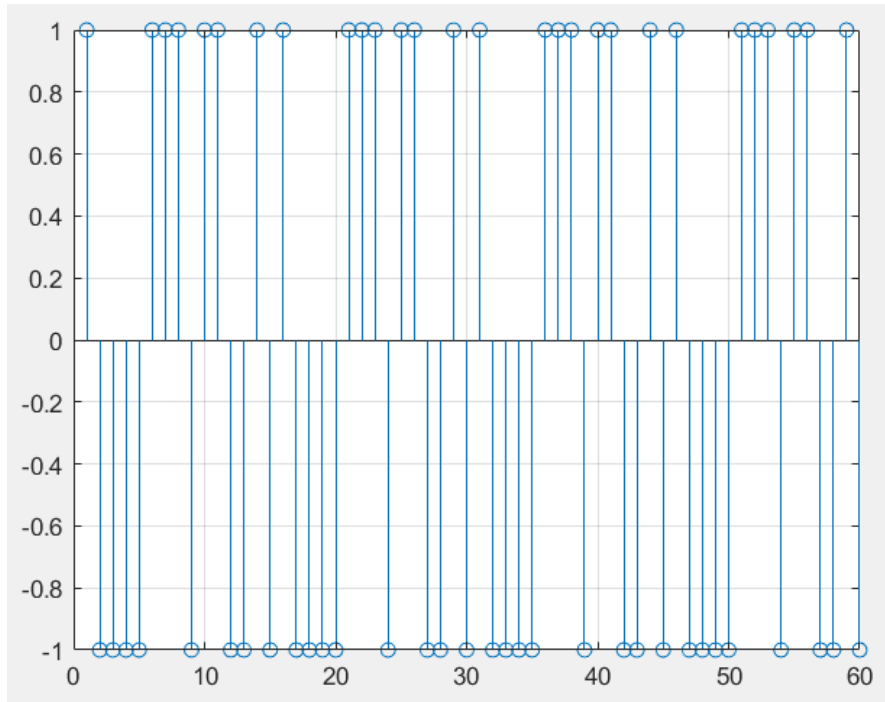


图 15: 增广递推最小二乘-输入信号 M 序列

迭代过程中的参数估计值及估计误差如下图所示:

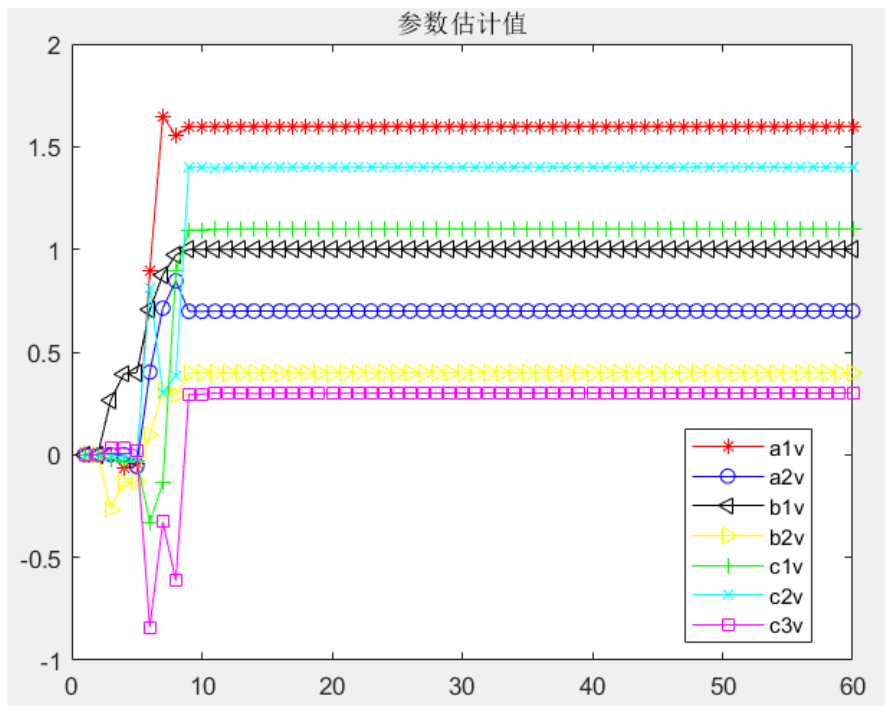


图 16: 增广递推最小二乘-参数估计值 (噪声方差 0.1)

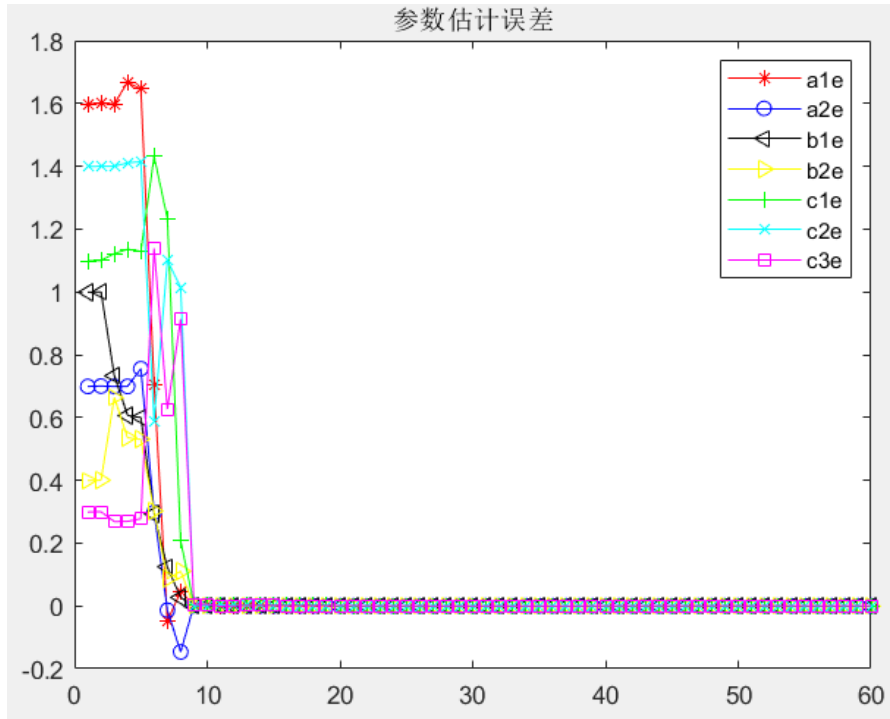


图 17: 增广递推最小二乘-参数估计误差 (噪声方差 0.1)

最终的参数辨识结果如下:

辨识参数	实际值	辨识值	误差百分比 (%)
$a_1$	1.6	1.59996	0.0025
$a_2$	0.7	0.699968	0.0046
$b_1$	1.0	0.999994	0.0006
$b_2$	0.4	0.399967	0.0083
$c_1$	1.1	1.0998	0.0182
$c_2$	1.4	1.39968	0.0229
$c_3$	0.3	0.299943	0.019

表 5: 增广递推最小二乘法辨识结果 (噪声方差: 0.1)

**高斯噪声: 均值为 0, 方差为 0.5**

采用 M 序列作为系统输入, 幅值为 1, 长度为 60, 同上。迭代过程中的参数估计值及估计误差如下图所示:

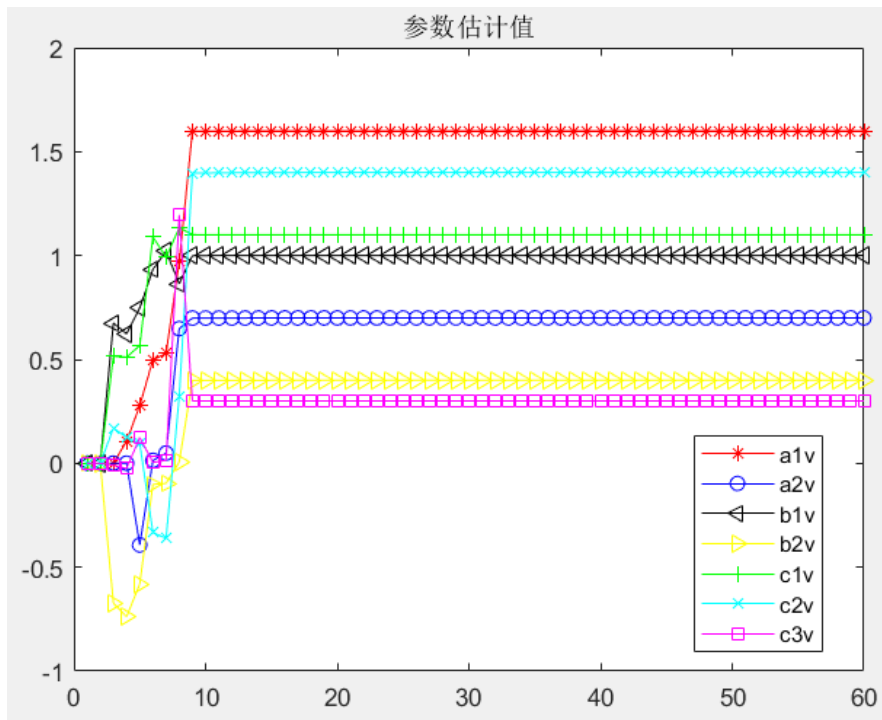


图 18: 增广递推最小二乘-参数估计值 (噪声方差 0.5)

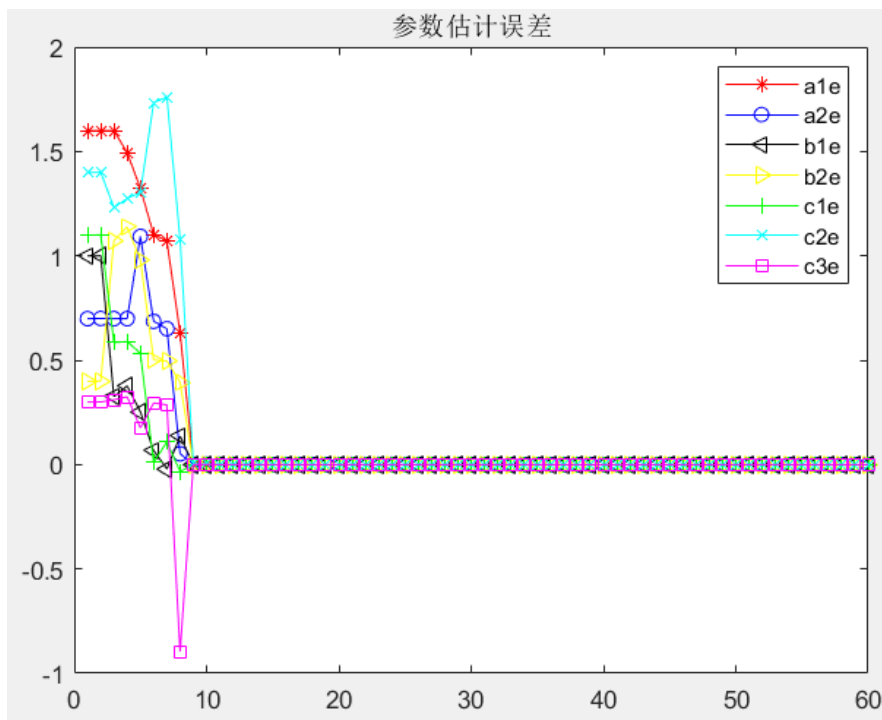


图 19: 增广递推最小二乘-参数估计误差 (噪声方差 0.5)

最终的参数辨识结果如下：



辨识参数	实际值	辨识值	误差百分比 (%)
$a_1$	1.6	1.59997	0.0019
$a_2$	0.7	0.69997	0.0043
$b_1$	1.0	0.999996	0.0004
$b_2$	0.4	0.399968	0.0080
$c_1$	1.1	1.09999	0.0009
$c_2$	1.4	1.39996	0.0029
$c_3$	0.3	0.299979	0.0070

表 6: 增广递推最小二乘法辨识结果 (噪声方差: 0.5)

### 结果分析

对比前述两种最小二乘参数辨识方法, 基于增广递推的参数辨识方法的辨识效果明显更好。通过参数估计值的变化情况可以看出, 增广递推的收敛速度更快, 参数估计值的波动更小; 而且在相同的迭代次数下, 增广递推得到的参数辨识结果的精度显著高于前述两种辨识方法; 此外, 在噪声干扰较大的条件下, 增广递推参数辨识方法也有非常好的表现; 最后值得指出的一点是, 增广递推还可以对噪声项参数进行辨识, 这使得增广递推参数辨识方法拥有更广的适用范围。

## 8 结论

根据以上实验结果可以得出, 在弱噪声干扰情况下, 使用一般最小二乘、递推最小二乘和增广递推最小二乘的参数辨识方法均能取得不错的效果, 具有较高的可靠性; 而在强噪声干扰情况下, 一般最小二乘和递推最小二乘参数辨识方法则会出现较大的偏差, 不再适用, 而增广最小二乘参数辨识方法不会受到明显的影响, 参数辨识精度始终能保持在较高的水平。此外, 相比一般最小二乘和递推最小二乘参数辨识方法, 增广递推最小二乘参数辨识方法还可以针对噪声项参数进行辨识, 且辨识精度很高。综上所述, 在参数辨识的可靠性和辨识精度方面, 增广递推最小二乘辨识方法是最优的, 其他两种参数辨识方法效果相对较差。

# 实验二      极大似然法估计参数值

## 实验报告

中国地质大学（武汉）

自动化学院

专      业：           自动化          

班      级：           231212          

姓      名：           廖志豪          

日期：   2023   年   12   月   7   日

# 1 实验题目

设系统动态模型为：

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

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  $\varepsilon(k)$  是均值为零，方差为 0.01 的高斯白噪声， $u(k)$  是由 4 级移位寄存器产生的幅度为 1 的 M 序列，利用极大似然参数估计递推算法对系统参数进行辨识。

## 2 实验目的

1. 掌握极大似然法估计参数值的原理；
2. 掌握极大似然法的 Matlab 实现和估计参数的编程；
3. 分析极大似然法的优缺点、理解极大似然法辨识的物理意义。

## 3 实验主要原理

### 3.1 极大似然参数估计原理

设离散随机过程  $Y_k$  与未知参数  $\theta$  有关，假定已知概率分布密度函数  $f(Y_k|\theta)$ 。如果我们得到  $n$  个独立的观测值：  $Y_1, Y_2, \dots, Y_n$ ，则可得概率密度分布分别为：  $f(Y_1|\theta)$ ,  $f(Y_2|\theta)$ ,  $\dots$ ,  $f(Y_n|\theta)$ 。要求根据这些观测值来估计未知参数  $\theta$ ，估计的准则是使得观测值  $Y_k$  的出现概率为最大。为此，定义一个似然函数：

$$L(Y_1, Y_2, \dots, Y_N|\theta) = f(Y_1|\theta)f(Y_2|\theta)\dots f(Y_N|\theta)$$

上式的右边是  $n$  个概率密度函数的连乘，似然函数  $L$  是  $\theta$  的函数。如果  $L$  达到最大值，则  $Y_k$  出现的概率为最大。因此极大似然法的实质就是求出使得  $L$  达到极大值的  $\theta$  的估值。对上式取对数并求偏导即可得到  $\theta$  的极大似然估计：

$$\hat{\theta}_{ML} = \frac{1}{N} \sum_{i=1}^N y_i$$

极大似然法辨识的物理意义：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  $\theta$ （真值）条件下的概率密度函数，即：

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

### 3.2 Newton-Raphson 方法原理

**Newton-Raphson 方法的思想**

根据第  $L$  次迭代得到的参数估计  $\hat{\theta}(L)$ ,  $\hat{\theta}(L+1)$ ：

$$J_{L+1}(\hat{\theta}(L+1)) \leq J_L(\hat{\theta}(L))$$

其中：

$$J_L(\theta) = \frac{1}{2} \sum_{k=(L-1)N+n+1}^{NL+n} \varepsilon^2(k)$$

## Newton-Raphson 法应用于极大似然参数估计

Newton-Raphson 法应用于极大似然估计求解的步骤:

### 1. 确定初始值 $\hat{\theta}_0$

先采集一批输入输出数据  $\{y(k)\}, \{u(k)\}$ , 用最小二乘法获得  $\hat{a}_1, \dots, \hat{a}_n, \hat{b}_1, \dots, \hat{b}_n$ , 对于  $\hat{\theta}(L)$  中的  $\hat{d}_1, \dots, \hat{d}_n$ , 可先任意假定初值。

### 2. 设定初始 $\varepsilon(1), \varepsilon(2), \dots, \varepsilon(n)$ 和 $\frac{\partial \varepsilon(1)}{\partial \theta}|_{\theta=\hat{\theta}(L)}, \frac{\partial \varepsilon(2)}{\partial \theta}|_{\theta=\hat{\theta}(L)}, \dots, \frac{\partial \varepsilon(n)}{\partial \theta}|_{\theta=\hat{\theta}(L)}$ , 为方便起见, 通常均取零。

### 3. 计算 $\varepsilon(k)$

采集一批长度为  $N$  的数据  $\{y(k)\}, \{u(k)\}$ , 利用  $\hat{\theta}(L)$  和  $\varepsilon(k)$ , 根据下式计算新的  $\varepsilon(k) (k = NL + n + 1, NL + n + 2, \dots, NL + n + N)$ :

$$\varepsilon(k) = y(k) + \sum_{i=1}^n \hat{a}_i y(k-i) - \sum_{i=1}^n \hat{b}_i u(k-i) - \sum_{i=1}^n \hat{d}_i \varepsilon(k-i)$$

### 4. 计算 $\frac{\partial \varepsilon(k)}{\partial \theta}|_{\theta=\hat{\theta}(L)} (k = NL + n + 1, NL + n + 2, \dots, NL + n + N)$

根据下式计算  $\frac{\partial \varepsilon(k)}{\partial \theta}|_{\theta=\hat{\theta}(L)}$ :

$$\frac{\partial \varepsilon(k)}{\partial a_j} = y(k-j) - \sum_{i=1}^n d_i \frac{\partial \varepsilon(k-i)}{\partial a_j}|_{\theta=\hat{\theta}(L)}$$

$$\frac{\partial \varepsilon(k)}{\partial b_j} = -u(k-j) - \sum_{i=1}^n d_i \frac{\partial \varepsilon(k-i)}{\partial b_j}|_{\theta=\hat{\theta}(L)}$$

$$\frac{\partial \varepsilon(k)}{\partial d_j} = -\varepsilon(k-j) - \sum_{i=1}^n d_i \frac{\partial \varepsilon(k-i)}{\partial d_j}|_{\theta=\hat{\theta}(L)}$$

### 5. 计算梯度阵 $\frac{\partial J_{L+1}}{\partial \theta}|_{\theta=\hat{\theta}(L)}$ 和 Hessian 阵 $\frac{\partial^2 J_{L+1}}{\partial (\theta)^2}|_{\theta=\hat{\theta}(L)}$

利用所计算的  $\varepsilon(k)$  和  $\frac{\partial \varepsilon(k)}{\partial \theta}|_{\theta=\hat{\theta}(L)}$ :

$$\frac{\partial J_{L+1}}{\partial \theta}|_{\theta=\hat{\theta}(L)} = \sum_{k=NL+n+1}^{(L+1)N+n} \varepsilon(k) \cdot \frac{\partial \varepsilon(k)}{\partial \theta}|_{\theta=\hat{\theta}(L)}$$

$$\frac{\partial^2 J_{L+1}}{\partial (\theta)^2}|_{\theta=\hat{\theta}(L)} \approx \sum_{k=NL+n+1}^{(L+1)N+n} \frac{\partial \varepsilon(k)}{\partial \theta} \left[ \frac{\partial \varepsilon(k)}{\partial \theta} \right]^T|_{\theta=\hat{\theta}(L)}$$

### 6. 计算新的估值 $\hat{\theta}(L+1)$

$$\hat{\theta}(L+1) = \hat{\theta}(L) - \left[ \frac{\partial^2 J_{L+1}}{\partial (\theta)^2} \right]^{-1}|_{\theta=\hat{\theta}(L)} \cdot \frac{\partial J_{L+1}}{\partial \theta}|_{\theta=\hat{\theta}(L)}$$

### 7. 取最后 $n$ 个 $\varepsilon(k)$ 和 $\frac{\partial \varepsilon(k)}{\partial \theta}|_{\theta=\hat{\theta}(L)}$ 值, 作为下一次迭代的初值, $L = L + 1$ , 转到 3. 继续循环, 直至满足停止条件。设经过 $r$ 次迭代计算后得到 $\hat{\theta}(r)$ , 则停止迭代标准为:

$$\|\hat{\theta}(r+1) - \hat{\theta}(r)\| \leq \Delta$$

$$\left| \frac{\sigma_{r+1}^2 - \sigma_r^2}{\sigma_r^2} \right| \leq \nu, \quad r = M$$

## 说明

1. 该数值算法即使当系统噪声水平较高时也能获得良好的估计
2. 需要进行  $N$  次观测，才能进行一次递推

## 4 实验对象或参数

实验对象 系统动态模型为：

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

实验参数 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  $\varepsilon(k)$  是均值为零，方差为 0.01 的高斯白噪声， $u(k)$  是由 4 级移位寄存器产生的幅度为 1 的 M 序列。

## 5 程序框图

### 5.1 递推的极大似然参数估计

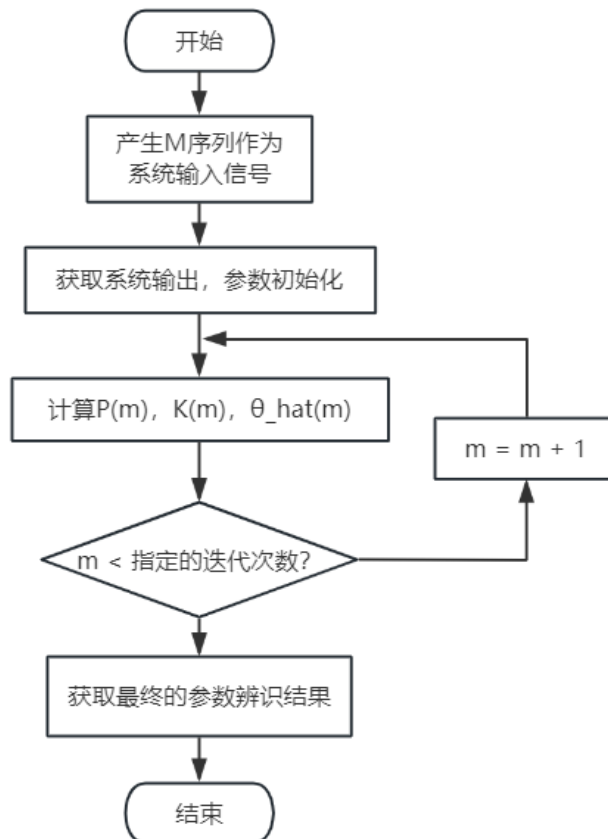


图 20: 递推极大似然法-程序框图

## 5.2 Newton-Raphson 方法

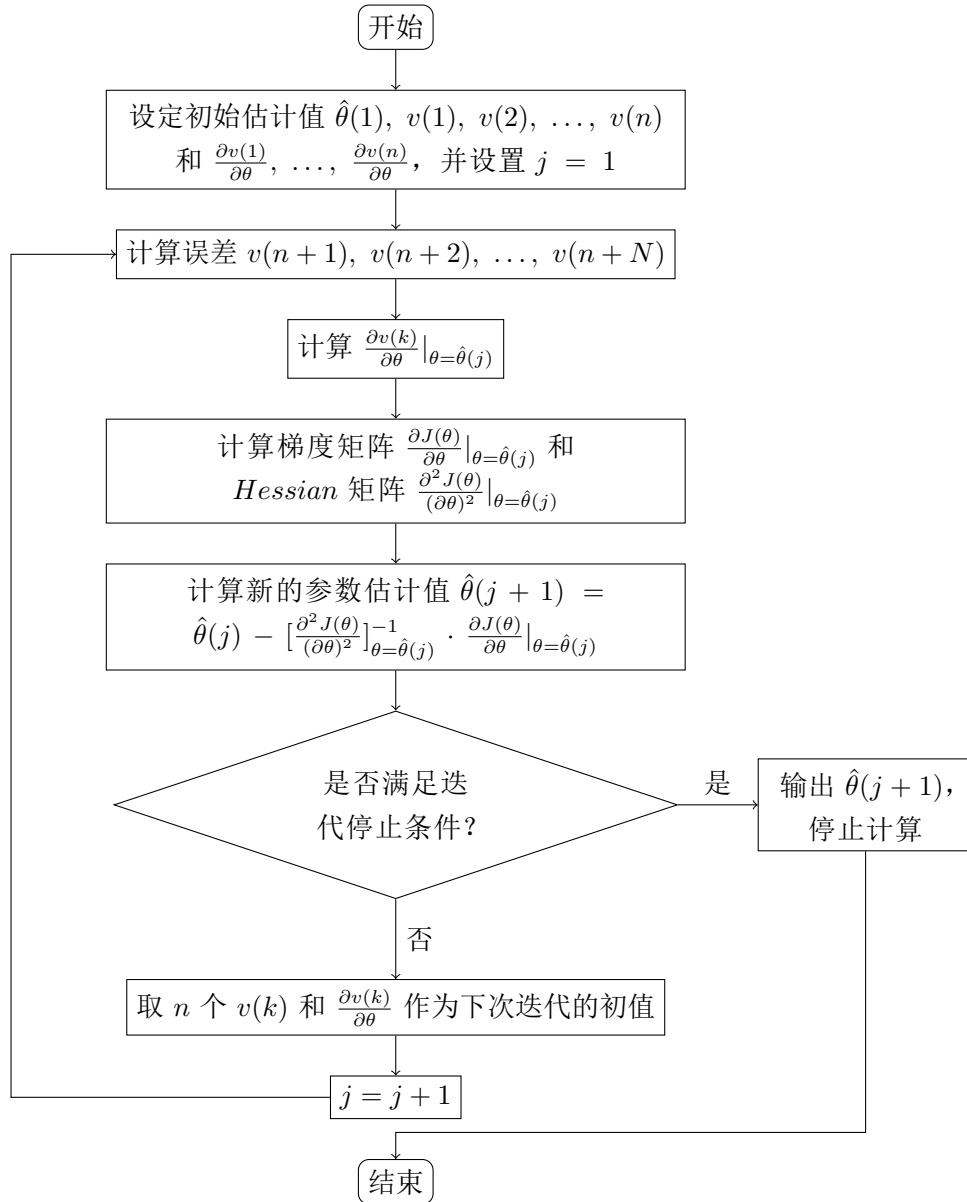


图 21: Newton-Raphson 方法用于极大似然参数估计-程序框图

## 6 程序代码

### 6.1 递推的极大似然参数估计

```

1 %% ----- 极大似然参数估计递推算法 ----- %
2 clc
3 clear
4 %产生仿真数据
5 n = 2;
6 total=2000; %迭代 2000 次
7 % total = 500; %迭代 500 次, 分析对比不同迭代次数对辨识结果的影响

```

```

8 sigma = 0.1; %噪声变量的均方根，即方差为 0.01
9
10 %% M 序列作为输入
11 z1 = 1; z2 = 1; z3 = 1; z4 = 0;
12 for i = 1:total
13     x1 = xor(z3, z4);
14     x2 = z1;
15     x3 = z2;
16     x4 = z3;
17     z(i) = z4;
18     if z(i) > 0.5
19         u(i) = -1; %限定幅度为 1
20     else
21         u(i) = 1;
22     end
23     z1 = x1; z2 = x2; z3 = x3; z4 = x4;
24 end
25 figure(1);
26 stem(u), grid on
27 title('输入信号 M 序列')
28
29 %% 系统输出
30 y(1) = 0; y(2) = 0; %系统输出初始化
31 % 产生均值为 0，方差为 0.01 的高斯白噪声
32 v = sigma*randn(total, 1);
33 y(1) = 1; y(2) = 0.01;
34 for k = 3:total
35     % 系统动态模型
36     y(k) = 0.5*y(k-1) + 0.2*y(k-2) + 1*u(k-1) + 1.5*u(k-2)
37     + v(k) - 0.8*v(k-1) + 0.3*v(k-2);
38 end
39 %初始化
40 theta0 = 0.001*ones(6, 1); %参数
41 e1(1) = -0.5 - theta0(1); %误差初始化
42 e2(1) = -0.2 - theta0(2);
43 e3(1) = 1.0 - theta0(3);
44 e4(1) = 1.5 - theta0(4);
45 e5(1) = -0.8 - theta0(5);
46 e6(1) = 0.3 - theta0(6);
47 a_hat(1) = theta0(1); %参数分离
48 a_hat(2) = theta0(2);
49 b_hat(1) = theta0(3);

```

```

50 b_hat(2) = theta0(4);
51 c_hat(1) = theta0(5);
52 c_hat(2) = theta0(6);
53 P0 = eye(6, 6); %矩阵 P 初始化
54 for i = 1:n
55     yf(i) = 0.1; uf(i) = 0.1; vf(i) = 0.1;
56     fai0(i, 1) = -yf(i);
57     fai0(n+i, 1) = uf(i);
58     fai0(2*n + i, 1) = vf(i);
59 end
60 e(1) = 1.0; e(2) = 1.0;
61
62 %% 极大似然参数估计递推算法
63 for i = n+1 : total
64     pusai=[-y(i-1);-y(i-2);u(i-1);u(i-2);e(i-1);e(i-2)];
65     C=zeros(n* 3,n* 3);
66     Q=zeros(3* n,1);
67     Q(1)=-y(i-1);Q(n+1)=u(i-1);Q(2* n+1)=e(i-1);
68     for j=1:n
69         C(1,j)=-c_hat(j);
70         C(n+1,n+j)=-c_hat(j);
71         C(2* n+1,2* n+j)=-c_hat(j);
72         if j>1
73             C(j,j-1)=1.0;
74             C(n+j,n+j-1)=1.0;
75             C(2* n+j,2* n+j-1)=1.0;
76         end
77     end
78     fai=C* fai0+Q;
79     K=P0* fai* inv(fai'* P0* fai+1);
80     P=[eye(6,6)-K* fai']* P0;
81     e(i)=y(i)-pusai'* theta0;
82     theta=theta0+K* e(i);
83     P0=P;
84     theta0=theta;
85     fai0=fai;
86     a_hat(1) = theta(1); %参数更新
87     a_hat(2) = theta(2);
88     b_hat(1) = theta(3);
89     b_hat(2) = theta(4);
90     c_hat(1) = theta(5);
91     c_hat(2) = theta(6);

```



```

92     e1(i) = -0.5 - a_hat(1); %误差更新
93     e2(i) = -0.2 - a_hat(2);
94     e3(i) = 1.0 - b_hat(1);
95     e4(i) = 1.5 - b_hat(2);
96     e5(i) = -0.8 - c_hat(1);
97     e6(i) = 0.3 - c_hat(2);
98 end
99
100 %% 参数估计误差
101 figure(2)
102 plot(e1); hold on
103 plot(e2); hold on
104 plot(e3); hold on
105 plot(e4); hold on
106 plot(e5); hold on
107 plot(e6); hold on
108 legend('e1','e2','e3','e4','e5','e6');
109 title('Parameter Estimation Error');
110 xlabel('times');
111 ylabel('error'); %error——误差
112 hold off
113
114 % 系统输出估计误差
115 figure(3)
116 plot(e);
117 title('Output Error ');
118 xlabel('times');
119 ylabel('error'); %error——误差
120
121 % 由于噪声随机性，导致参数 c1 c2 的估计误差较大
122 theta %输出辨识结果，即各参数辨识值

```

## 6.2 Newton-Raphson 方法

```

1 clc
2 clear
3 close all;
4 n = 2; % 注意这个 n
5 total = 10;
6 sigma = 0.1; %噪声变量的均方根 方差即为 0.01
7 N = 150;
8
9 %M 序列作为输入

```

```

10 z1 = 1; z2 = 1; z3 = 1; z4 = 0;
11 for i = 1 : total+2
12     x1 = xor(z3, z4);
13     x2 = z1;
14     x3 = z2;
15     x4 = z3;
16     z(i) = z4;
17     if z(i) > 0.5
18         u(i) = -1;
19     else
20         u(i) = 1;
21     end
22     z1 = x1; z2 = x2; z3 = x3; z4 = x4;
23 end
24 figure(1);
25 stem(u), grid on
26
27 %% 系统输出
28 y(1) = 0; y(2) = 0;
29 epsilon = sigma*randn(N+3, 1); %噪声
30 y(1) = 1; y(2) = 0.01;
31 for k = 3 : total+2
32     y(k) = 0.5*y(k-1)+0.2*y(k-2)+1.0*u(k-1)+1.5*u(k-2)
33     +epsilon(k)-0.8*epsilon(k-1)+0.3*epsilon(k-2);
34 end
35
36 %% 最小二乘初始化
37 phi = zeros(total, 2*n + 1);
38 for k = 1:total
39     phi(k, :) = [-y(1, (k-1)+n), -y(1, k), ...
40         u(1, k+n), u(1, k-1+n), u(1, k)];
41 end
42 yhat = y(1, n+1:n+total)';
43 theta = inv(phi'*phi)*phi'*yhat;
44 % 设置初始值
45 a1 = theta(1);
46 a2 = theta(2);
47 b1 = theta(3);
48 b2 = theta(4);
49 d1 = 0.1;
50 d2 = 0.1;
51 theta0 = [a1, a2, b1, b2, d1, d2]';

```

```

52 epsilon(1) = 0; epsilon(2) = 0;
53
54 epsida1(1) = 0; epsida2(1) = 0;
55 epsida1(2) = 0; epsida2(2) = 0;
56 epsidb1(1) = 0; epsidb2(1) = 0;
57 epsidb1(2) = 0; epsidb2(2) = 0;
58 epsidd1(1) = 0; epsidd2(1) = 0;
59 epsidd1(2) = 0; epsidd2(2) = 0;
60
61 %%
62 j=1;
63 theta1 = zeros(6,1);
64 epsidtheda = zeros(6,1);
65 while j <= 10000 %100000 迭代次数
66     % 参数估计值
67     a1value(j)=a1;
68     a2value(j)=a2;
69     b1value(j)=b1;
70     b2value(j)=b2;
71     d1value(j)=d1;
72     d2value(j)=d2;
73     % 参数估计误差
74     a1e(j)=-0.5-a1;
75     a2e(j)=-0.2-a2;
76     b1e(j)=1.0-b1;
77     b2e(j)=1.5-b2;
78     d1e(j)=-0.8-d1;
79     d2e(j)=0.3-d2;
80
81     % M序列作为输入
82     for i = 1:N+2
83         x1 = xor(z3, z4);
84         x2 = z1;
85         x3 = z2;
86         x4 = z3;
87         z(i) = z4;
88         if z(i) > 0.5
89             u(i) = -1;
90         else
91             u(i) = 1;
92         end
93         z1 = x1; z2 = x2; z3 = x3; z4 = x4;

```

```

94     end
95
96     % 获取系统输出
97     %       $y(1) = 0; y(2) = 0;$ 
98     epsilon = sigma*randn(N+3, 1); %噪声
99     y(1) = 1; y(2) = 0.01;
100    for k = 3:N+2
101        y(k) = 0.5*y(k-1)+0.2*y(k-2)+1.0*u(k-1)+1.5*u(k-2)
102        +epsilon(k)-0.8*epsilon(k-1)+0.3*epsilon(k-2);
103    end
104    gradmat = 0;
105    hessianmat = 0;
106    a1 = theta0(1); a2 = theta0(2);
107    b1 = theta0(3); b2 = theta0(4);
108    d1 = theta0(5); d2 = theta0(6);
109
110    for k=3:N+2
111        epsilon(k)=y(k)+a1*y(k-1)+a2*y(k-2)-b1*u(k-1)
112        -b2*u(k-2)-d1*epsilon(k-1)-d2*epsilon(k-2);
113        epsida1(k)=y(k-1)-d1*epsida1(k-1)-d2*epsida1(k-2);
114        epsida2(k)=y(k-2)-d1*epsida2(k-1)-d2*epsida2(k-2);
115        epsidb1(k)=-u(k-1)-d1*epsidb1(k-1)-d2*epsidb1(k-2);
116        epsidb2(k)=-u(k-2)-d1*epsidb2(k-1)-d2*epsidb2(k-2);
117        epsidd1(k)=-epsilon(k-1)-d1*epsidd1(k-1)-d2*epsidd1(k-2);
118        epsidd2(k)=-epsilon(k-2)-d1*epsidd2(k-1)-d2*epsidd2(k-2);
119        epsidtheda=[epsida1(k),epsida2(k),epsidb1(k), ...
120        epsidb2(k),epsidd1(k),epsidd2(k)]';
121        gradmat=gradmat+epsilon(k)*epsidtheda;
122        hessianmat=hessianmat+epsidtheda'*epsidtheda;
123    end
124
125    theta1=theta0;
126    theta0=theta0-inv(hessianmat)*gradmat;
127    a1=theta0(1);a2=theta0(2);
128    b1=theta0(3);b2=theta0(4);
129    d1=theta0(5);d2=theta0(6);
130
131    epsilon(1)=epsilon(N+1);epsilon(2)=epsilon(N+2);
132    epsida1(1)=epsida1(N+1);epsida2(1)=epsida2(N+1);
133    epsidb1(1)=epsidb1(N+1);epsidb2(1)=epsidb2(N+1);
134    epsidd1(1)=epsidd1(N+1);epsidd2(1)=epsidd2(N+1);
135    epsida1(2)=epsida1(N+2);epsida2(2)=epsida2(N+2);

```

```

136     epsidb1(2)=epsidb1(N+2);epsidb2(2)=epsidb2(N+2);
137     epsidd1(2)=epsidd1(N+2);epsidd2(2)=epsidd2(N+2);
138     j = j+1;
139 end %递推次数达到最大值即停止
140
141 %% 参数估计
142 figure(2)
143 plot(a1value);
144 hold on
145 plot(a2value);
146 hold on
147 plot(b1value);
148 hold on
149 plot(b2value);
150 hold on
151 plot(d1value);
152 hold on
153 plot(d2value);
154 legend('a1','a2','b1','b2','d1','d2');
155 title('参数估计值');
156
157 % 参数估计误差
158 figure(3)
159 plot(a1e);
160 hold on
161 plot(a2e);
162 hold on
163 plot(b1e);
164 hold on
165 plot(b2e);
166 hold on
167 plot(d1e);
168 hold on
169 plot(d2e);
170 legend('a1e','a2e','b1e','b2e','d1e','d2e');
171 title('参数估计误差');

```

## 7 实验结果及分析

### 7.1 递推的极大似然参数估计

分别选取迭代次数为 500 次和 2000 次，分析对比不同迭代次数对辨识结果的影响。

迭代次数：500 次

采用 M 序列作为系统输入，如下图所示：

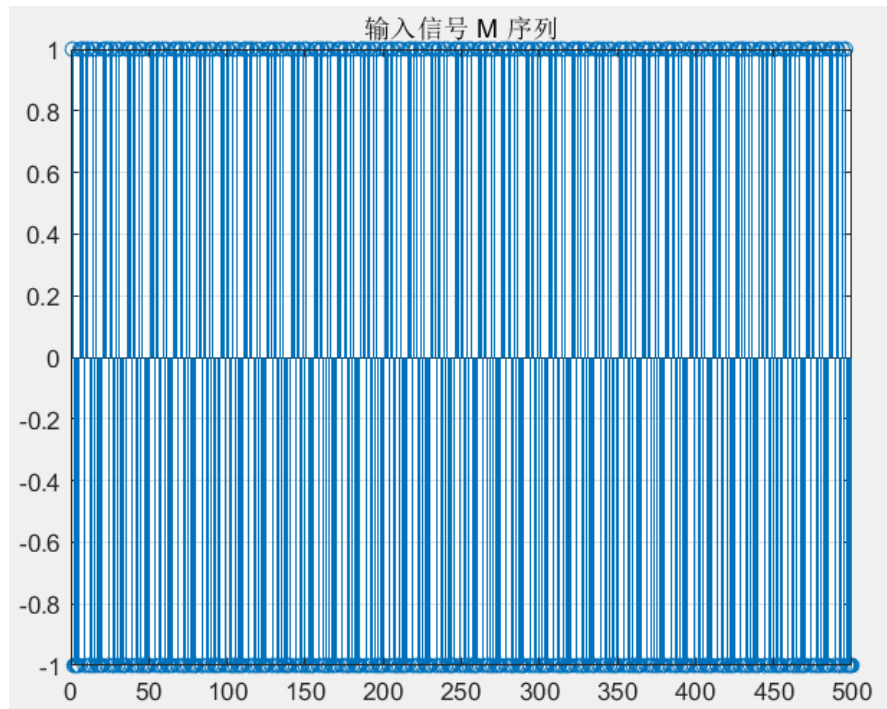


图 22: 输入信号 M 序列（迭代次数 500）

迭代过程中的参数估计误差如下图所示：

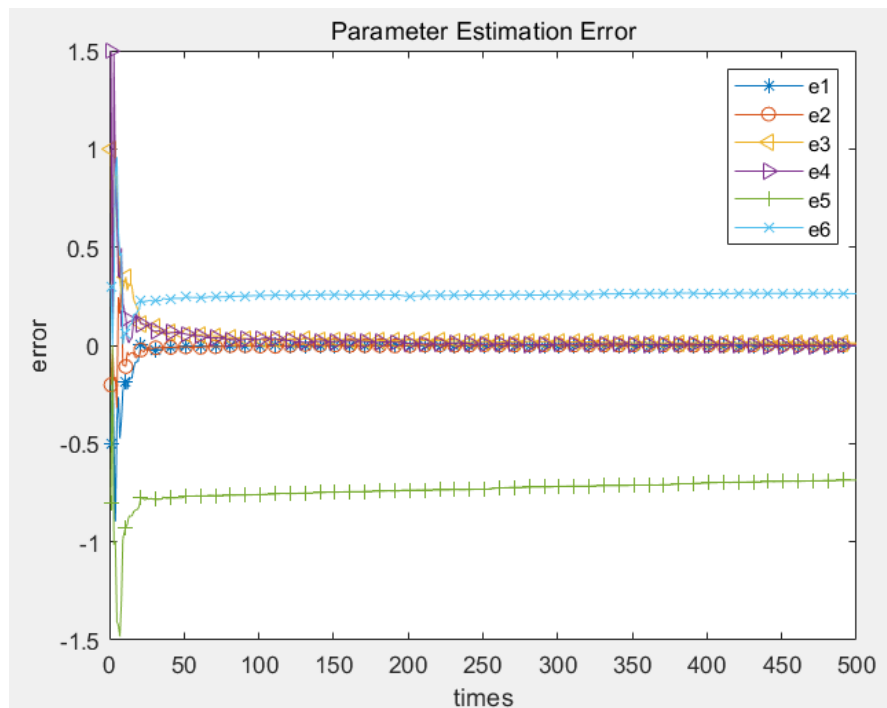


图 23: 参数估计误差（迭代次数 500）

系统的输出误差如下图所示：

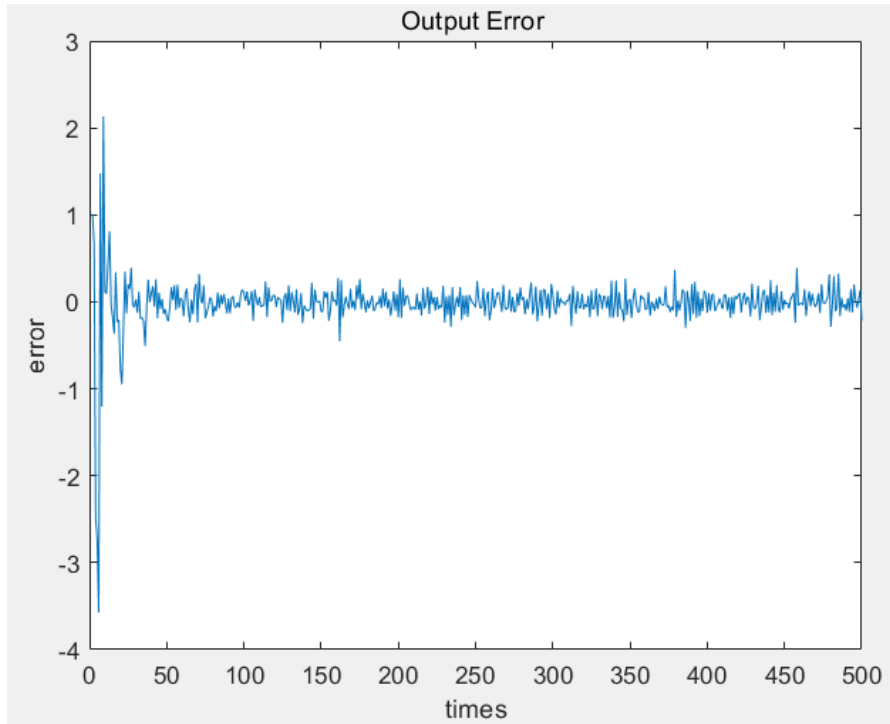


图 24: 系统输出误差（迭代次数 500）

最终的参数估计如下：

辨识参数	实际值	辨识值	误差百分比 (%)
$a_1$	-0.5	-0.5015	0.3
$a_2$	-0.2	-0.2005	0.25
$b_1$	1.0	1.0048	0.48
$b_2$	1.5	1.4868	0.88
$d_1$	-0.8	-0.0517	93.54
$d_2$	0.3	-0.0014	99.53

表 7: 递推极大似然辨识结果（迭代次数 500）

迭代次数：2000 次

采用 M 序列作为系统输入：

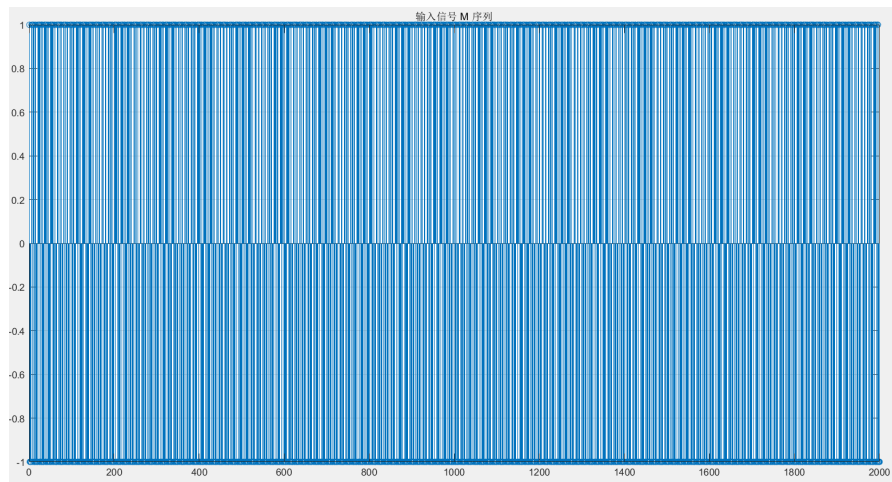


图 25: 输入信号 M 序列 (迭代次数 2000)

迭代过程中的参数估计误差:

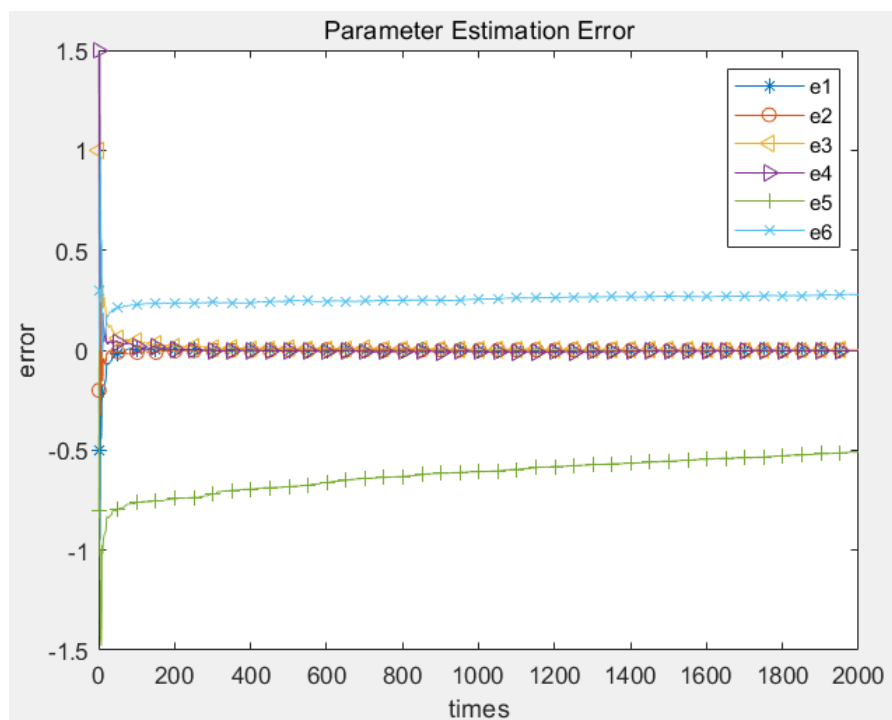


图 26: 参数估计误差 (迭代次数 2000)

系统的输出误差:



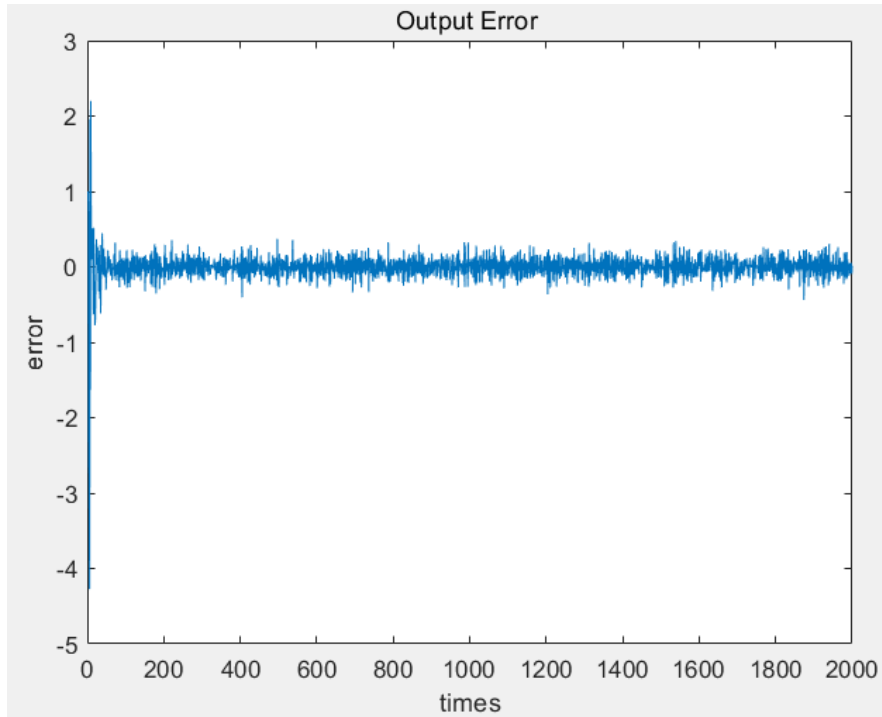


图 27: 系统输出误差 (迭代次数 2000)

最终的参数估计如下:

辨识参数	实际值	辨识值	误差百分比 (%)
$a_1$	-0.5	-0.4951	0.98
$a_2$	-0.2	-0.2052	2.6
$b_1$	1.0	0.9959	0.41
$b_2$	1.5	1.5076	0.51
$d_1$	-0.8	-0.2471	69.11
$d_2$	0.3	0.0043	98.57

表 8: 递推极大似然辨识结果 (迭代次数 2000)

## 结果分析

1. 根据迭代过程中参数估计误差的变化情况可以看出, 完成 500 次迭代以后参数  $a_1$ ,  $a_2$ ,  $b_1$ ,  $b_2$  均已收敛至误差接近于零的状态, 且随着迭代次数的继续增加, 参数估计误差的收敛效果更加明显;
2. 参数  $d_1$ ,  $d_2$  的估计误差在经过 2000 迭代以后仍然保持在较高水平, 这是因为由于噪声随机性 (高斯白噪声), 使得递推的极大似然参数估计对噪声项的参数估计效果很差。

## 7.2 Newton-Raphson 方法

采用 M 序列作为系统输入, 长度为 12, 如下图所示:

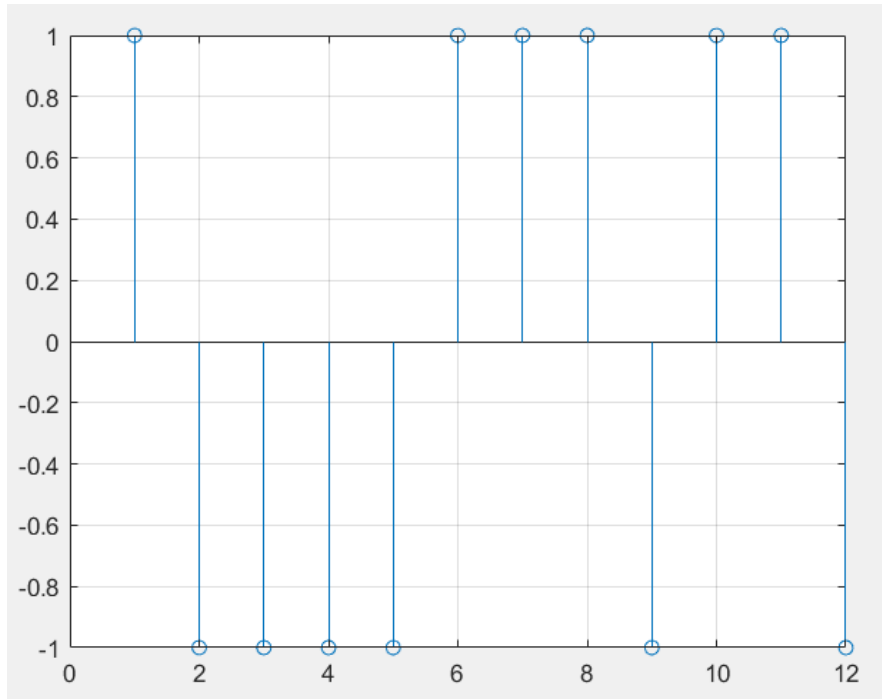


图 28: Newton-Raphson 方法 M 序列输入

迭代过程中的参数估计值如下图所示:

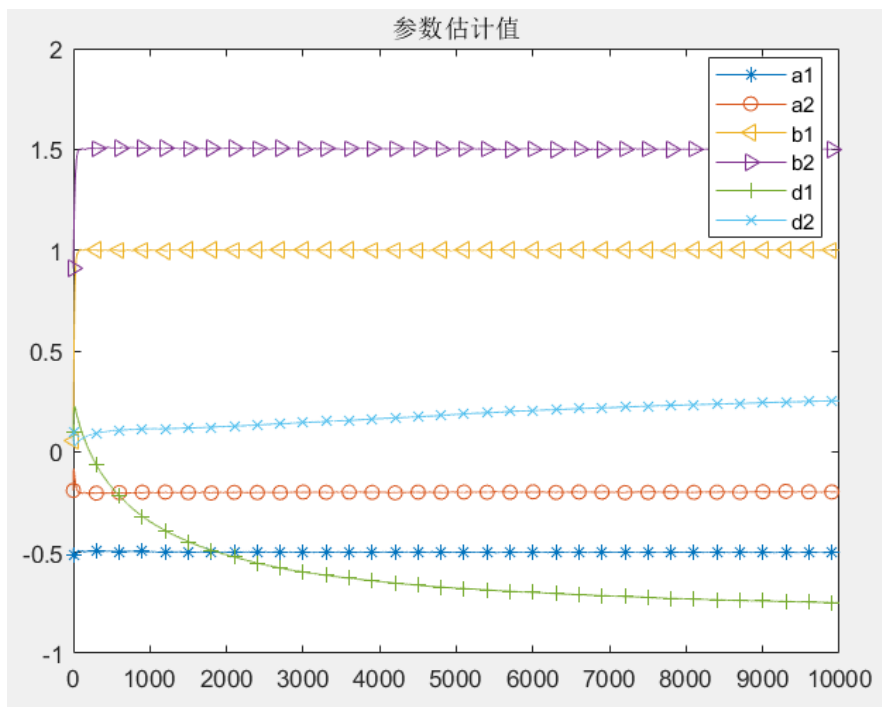


图 29: Newton-Raphson 方法参数估计值

参数估计误差如下图所示:

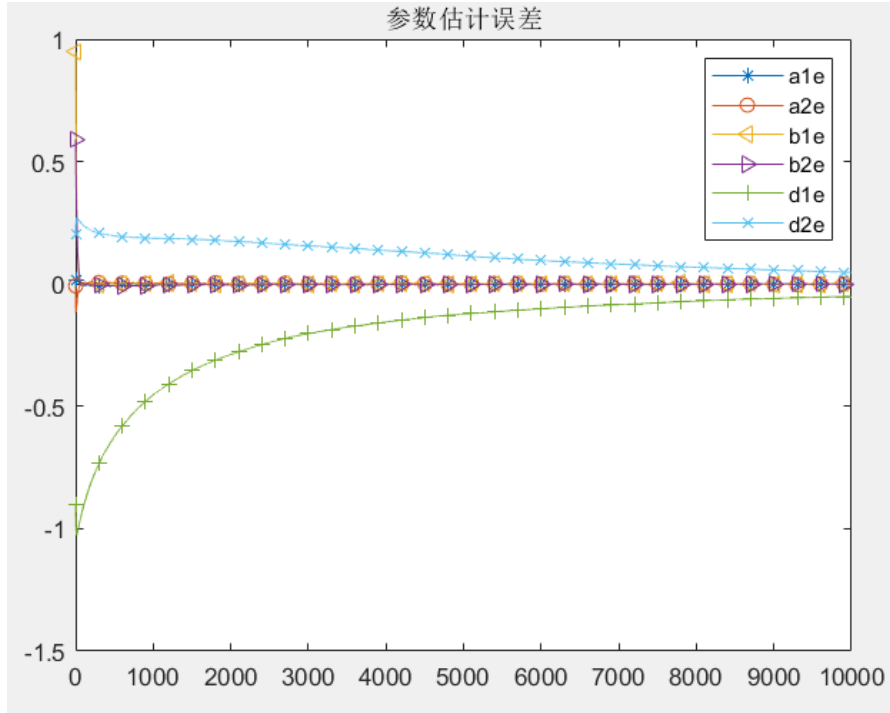


图 30: Newton-Raphson 方法参数估计误差

最终的参数辨识结果如下：

辨识参数	实际值	辨识值	误差百分比 (%)
$a_1$	-0.5	-0.498588	0.2824
$a_2$	-0.2	-0.200589	0.2945
$b_1$	1.0	1.00164	0.164
$b_2$	1.5	1.49964	0.0240
$d_1$	-0.8	-0.750652	6.1685
$d_2$	0.3	0.252325	15.8917

表 9: Newton-Raphson 方法辨识结果

### 结果分析

相比递推的极大似然参数估计方法，Newton-Raphson 方法对于噪声项的参数估计效果明显更好，收敛速度更快。由实验结果可以看到，经过 10000 次迭代以后，噪声项参数估计误差基本已收敛至零值。此外，根据本次实验结果可以看出，Newton-Raphson 方法针对除噪声项参数之外的参数辨识效果较好，若对于噪声项参数的辨识精度有较高要求，则需要适当增大迭代次数。

## 8 结论

递推的极大似然参数估计方法对于除噪声项参数之外的其他参数的辨识效果较好，收敛速度快，而且精度高，可靠性好；但是对噪声项参数的辨识效果很差，这是由于噪声产生过程的随机性以及基于递推的极大似然参数估计的自身局限造成的。

相对来说，Newton-Raphson 方法不仅在系统参数辨识上有着很高的精度，而且对于噪声项参数的辨识也有着较好的效果；其缺点在于，如果对噪声项参数的辨识精度有比较高的要求，则需要经过更多的迭代次数才能得到满意的结果，且完成一次迭代过程需要更多的数据作为支撑，运算量较大，对于数据获取的要求较高。