一、实验目的

通过仿真实验掌握利用递推最小二程方法辨识差分方程模型参数的原理和方法。

二、实验内容

给出系统的差分方程为

$$x(k)=a_1x(k-1)+a_2x(k-2)+b_1u(k-1)+b_2u(k-2)$$
$$z(k)=x(k)+v(k)$$

其中 $a_1=1.5,a_2=-0.7,b_1=1,b_2=0.5$；$u(k),x(k)$和$z(k)$ 分别为过程的输入，状态和输出变量；$v(k)$ 为测量白噪声过程，服从正态分布，均值为零，方差为σ_v^2，记作 $v(k)\sim N(0,\sigma_v^2)$。

过程的输入驱动采用 M 序列，输出受到白噪声 $v(k)$ 的污染。根据过程的输入和输出数据 $\{u(k),z(k)\}$，利用递推最小二乘算法辨识系统模型参数 a_1,a_2和b_1,b_2 。

三、实验要求

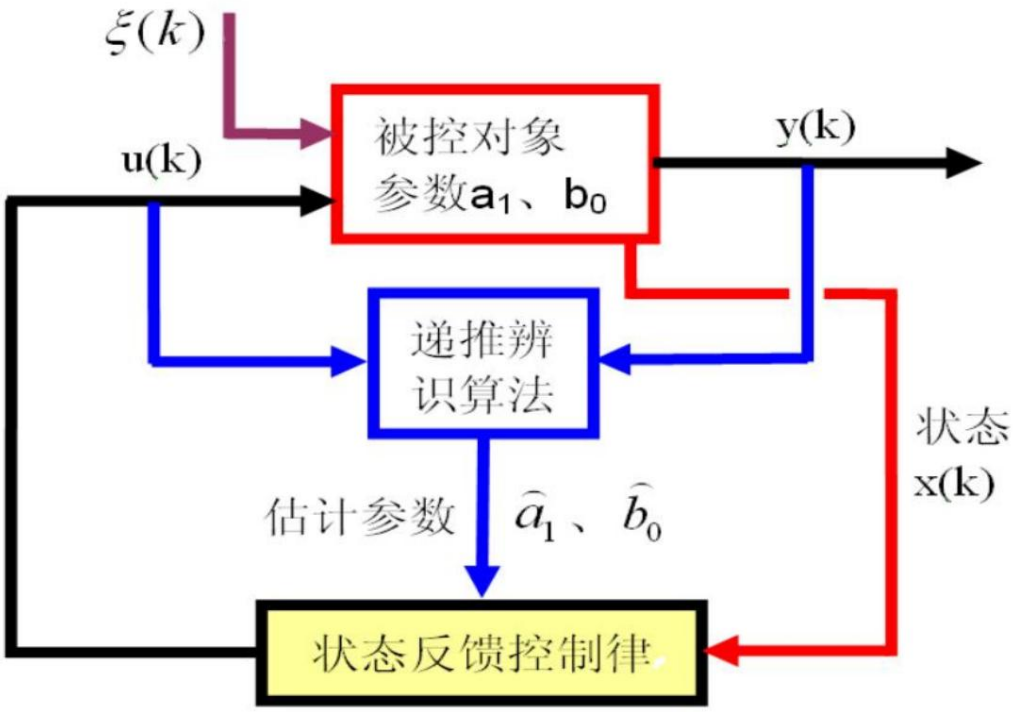
进行方案设计，模拟过程进行仿真，获得输出数据，用M序列作为辨识的输入信号，噪声采用根方差 $\sigma_v=0.01$ 的正态分布白噪声，通过最小二乘辨识模型参数，计算参数估计值与理论值之间的误差，画出相应的仿真曲线，分析噪声及算法对辨识结果的影响。

四、实验原理

基本最小二乘法（LS）有诸多缺点，例如对于一组动态的数据，每次接收到新数据，就要全部重算一遍。这种重复的计算的成本很大，导致实用性不好。所以对于一组离线数据，基本最小二乘法是适用的。但是如果是实时统计分析一系列数据，那么基本最小二乘法就会遇到计算困难。

递推最小二乘法可以在获取一个新数据时，可以直接使用该数据和已经计算过的结果进行某种运算，达到“修正”旧结果的目的。

五、实验框图



六、实验程序代码

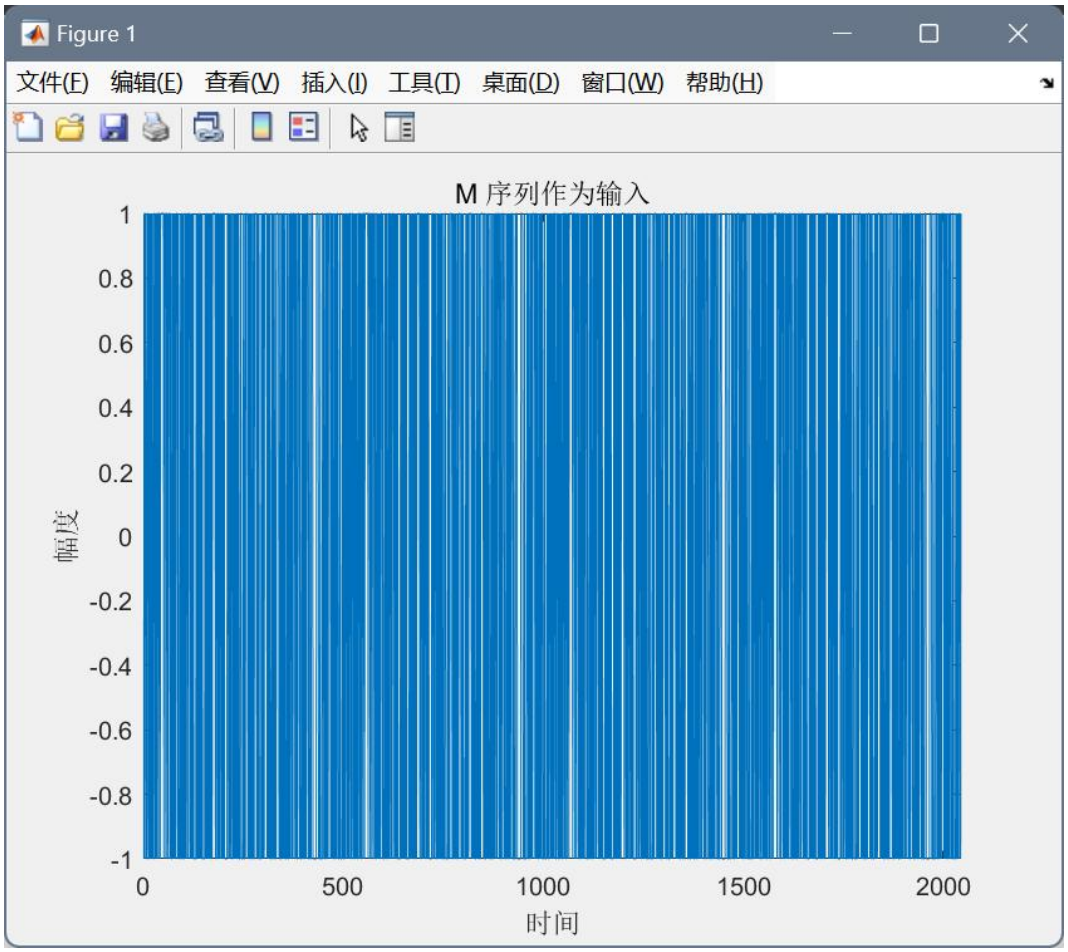
主程序

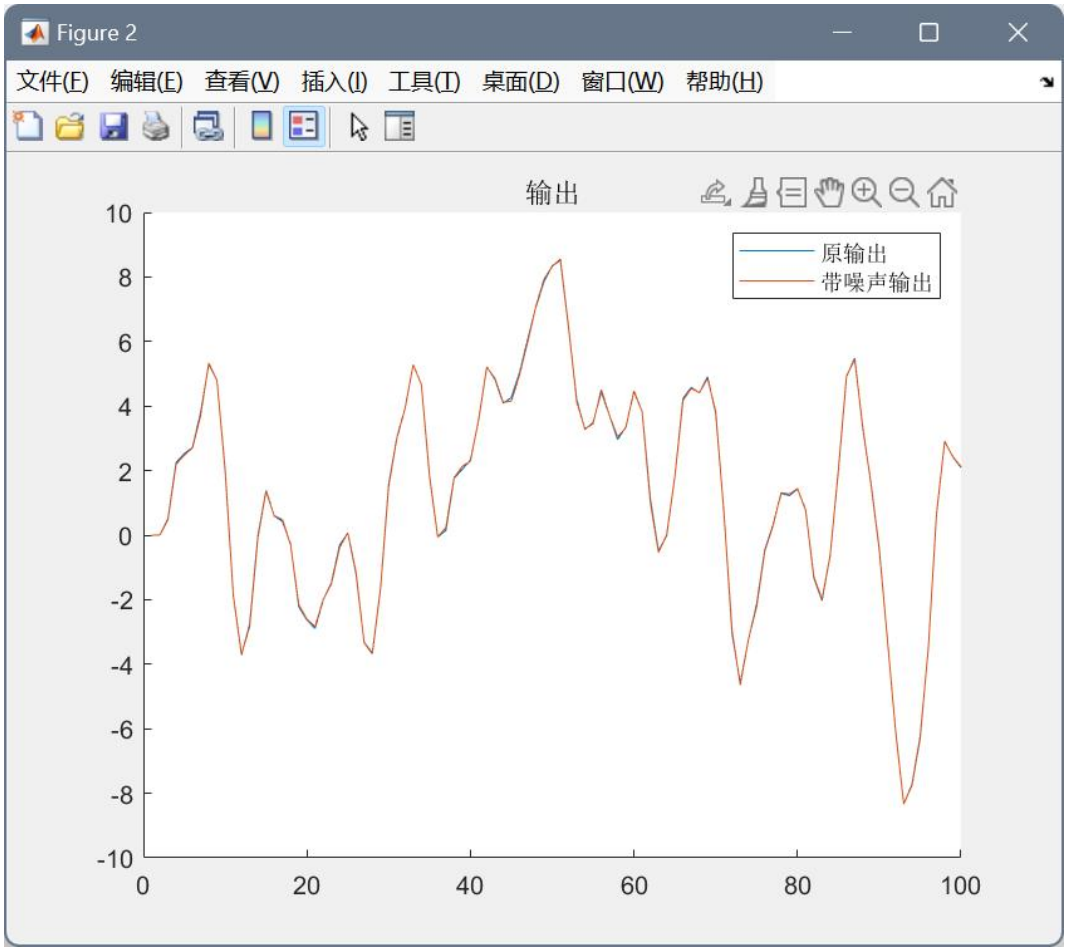
```
clear; clc; close all;

% 生成 M 序列 function u = generate_m_sequence(M_init, r, A)
    Np = 2 ^ length(M_init) - 1; % M 序列周期
    Num = (r + 1) * Np; % M 序列长度
    u = zeros(1, Num);
    M = M_init;

    for i = 1:Num
        u(i) = M(length(M_init)); % 取出最后一位作为当前输出
        m = xor(M(length(M_init)), M(5)); % 异或生成反馈位
        M = [m M(1:length(M_init) - 1)]; % 移位操作
    end

    u = A * (1 - 2 * u); % 值域为 (-A, A) 的 M 序列 end

register_length = 9;
M_init = [1, 1, 0, 0, 0, 1, 0, 0, 1];
u = generate_m_sequence(M_init, 3, 1);

figure;
plot(u);
```

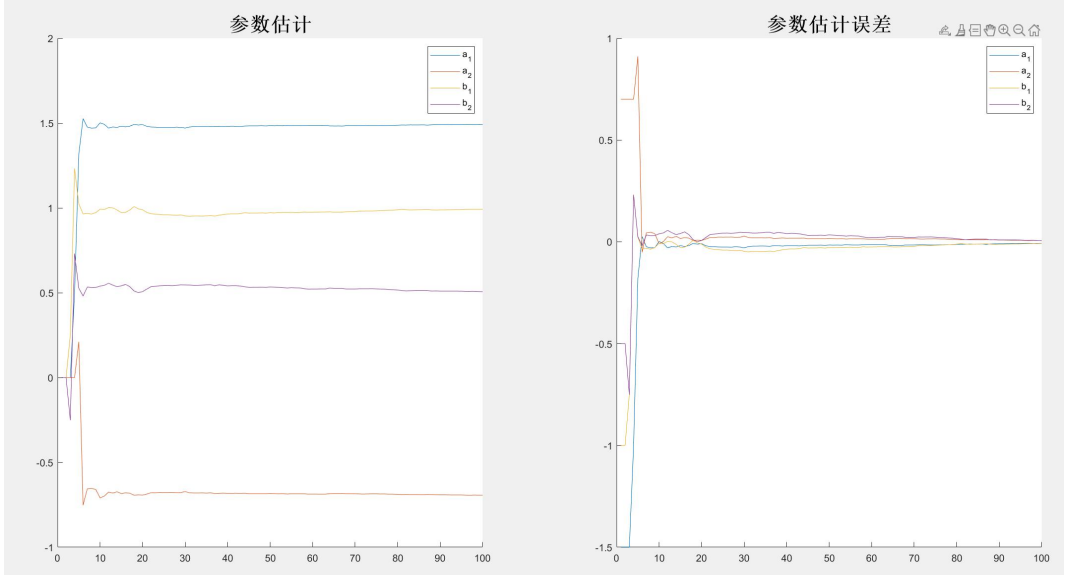
```
title("M 序列作为输入");
xlim([0, length(u)]);
xlabel('时间');
ylabel('幅度');

sigma = 0.05; % 高斯白噪声方差
Nmax = 100; % 计算次数
x = zeros(1, Nmax);
y = zeros(1, Nmax);
z = randn(1, Nmax) * sigma;
theta = zeros(4, Nmax); % 初值设为 0
P = 100 * eye(4);
for i = 3:Nmax
    % 系统差分方程
    x(i) = 1.5 * x(i - 1) -0.7 * x(i - 2) + 1 * u(i - 1) + 0.5 * u(i - 2);
    y(i) = x(i) + z(i);
    % 计算 phi
    phi = [y(i - 1) y(i - 2) u(i - 1) u(i - 2)]';
    % 计算 theta, K 和 P
    K = P * phi / (1 + phi' * P * phi);
    theta(:, i) = theta(:, i - 1) + K * (y(i) - phi' * theta(:, i - 1));
    P = P - P * phi / (1 + phi' * P * phi) * phi' * P;
end

figure;
hold on;
title("输出");
plot(x);
plot(y);
legend("原输出", "带噪声输出");
hold off;
figure;

subplot(1, 2, 1);
hold on;
title("参数估计", "FontSize", 20, "FontWeight", "bold");
for i = 1:4
    plot(1:length(theta(i, :)), theta(i, :));
end

legend("a_1", "a_2", "b_1", "b_2");
hold off;
param = [1.5 -0.7 1 0.5];
subplot(1, 2, 2);
```

```
hold on;
title("参数估计误差", "FontSize", 20, "FontWeight", "bold");
for i = 1:4
    error = theta(i, :) - param(i);
    plot(1:length(error), error);
end

legend("a_1", "a_2", "b_1", "b_2");
hold off;
```

七、实验结果及分析

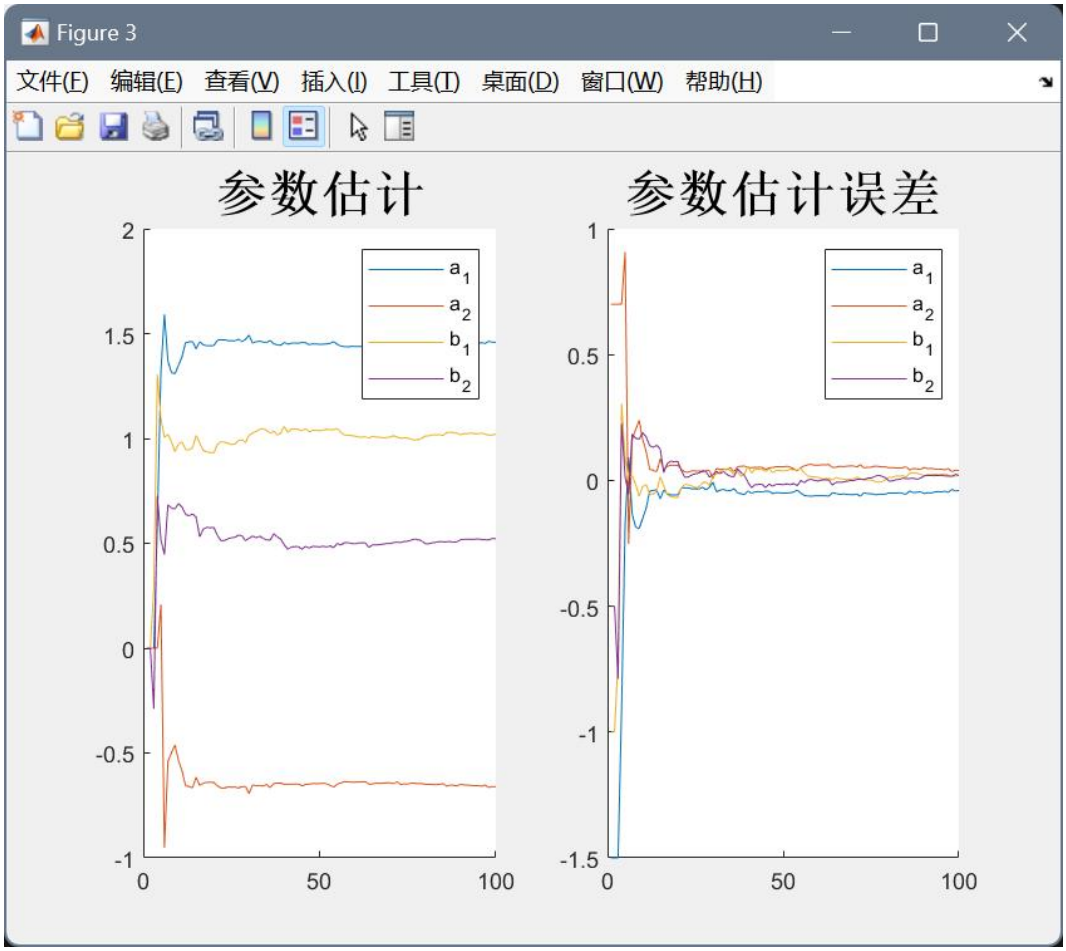






由图像可知，以零为初始值的待辨识的参数 a1，a2，b1，b2 在递推最小二乘法中持续更新数值，随着数据的逐渐增多，参数经历多次在线计算后趋于稳定，与真实值之间的误差也逐渐趋于零，说明递推最小二乘法能够有效地辨识系统的参数。

如果将高斯白噪声的反差增大到原来的两倍，结果如下：



此时估计的参数的不稳定性增加，最终相比较真实值的稳态误差增大，说明系统的噪声会影响递推最小二乘法的参数辨识效果。

八、实验结论

通过绘制参数估计值随时间的变化曲线，观察到系统中待辨识的参数 a1，a2，b1，b2 在经历多次在线计算后趋于稳定，这一现象表明递推最小二乘法算法能够通过迭代计算有效地辨识系统的动态特性，为进一步研究和应用提供可靠的基础。

在实验过程中引入了白噪声，观察到噪声对参数估计的影响，参数的估计会受到噪声的扰动，尤其在最初几次计算时，参数估计值的波动较大。随着数据量的增加，迭代次数的增加，待辨识参数的估计逐渐趋于稳定。这表明系统在处理不确定性时的适应能力得到了显著增强。

递推最小二乘算法具备在线更新待辨识参数的能力，使其在需要实时调整控制策略的应用场景（如自适应控制系统）中，成为一种非常有效的参数辨识方法。递推最小二乘法在系统参数辨识中表现出较好的鲁棒性，能够适应存在噪声干扰

或参数实时变化的复杂系统环境。这一特点使得该方法在实际应用中具有广泛的适用性和重要性，为动态系统的建模和控制提供了强有力的支持。