**ARMAX模型的一般形式可以表示为：**

**A(z)y(t)=B(z)u(t)+D(z)e(t)**

**其中：**

***y*(*t*) 是时间点*t* 的系统输出。**

***u*(*t*) 是时间点*t* 的系统外部输入。**

***v*(*t*) 是时间点*t* 的噪声项，通常假设为白噪声。**

**A(z), B(z), D(z) 分别是输出y(t)，输入u(t)，和噪声项v(t) 的多项式系数。**

**其中：**

**选取仿真步数N=500。**

**白噪声幅值0.01。**

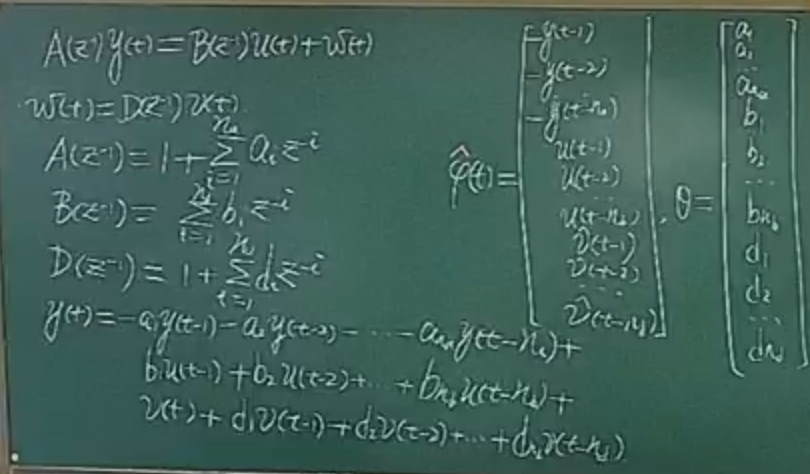
**输入选取幅值为1的M序列。**

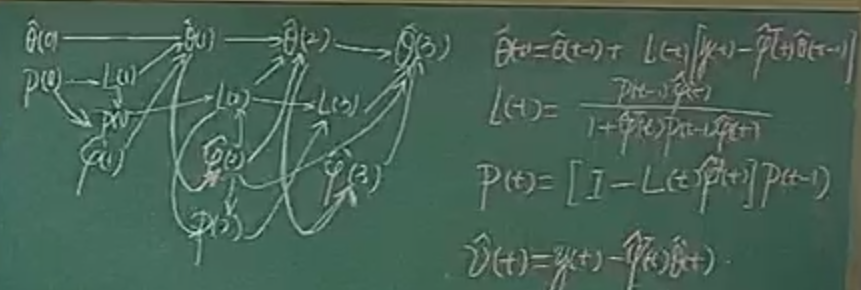
**A(z)=1-0.5z-1+0.2z-2。**

**B(z)=0.5+0.2 z-1。**

**D(z)=1+0.2 z-1-0.1z-2。**

**系统辨识算法如图**





**仿真代码如下：**

|  |
| --- |
| clc  clear  close all  % 数据长度  N = 500;  % 定义系统 ARMAX 模型的阶数  na = 2; % AR 部分的阶数  nb = 2; % X 部分的阶数  nd = 2; % MA 部分的阶数  % 定义模型系数  A = [-0.5, 0.2]; % AR部分  B = [0.5, 0.2]; % X输入部分  D = [0.2, -0.1]; % MA部分  % 定义输入信号  u = idinput(N);  % 系统初始化  y = zeros(N, 1);  y\_h = zeros(na, 1);  u\_h = zeros(nb, 1);  v\_h = zeros(nd, 1);  phi = zeros(na + nb + nd, N);  v = 0.01 \* randn(N, 1); % 标准正态分布白噪声  v\_hat = zeros(N, 1);  % RLS算法初始化  lambda = 1; % 遗忘因子  theta\_hat = zeros(na + nb + nd, N); % 参数估计初始化  P = 100 \* eye(na + nb + nd); % 协方差矩阵初始化  for i = 1:N  % 更新系统状态  if i > na  y\_h = [y(i-1); y\_h(1:end-1)];  end  if i > nb  u\_h = [u(i-1); u\_h(1:end-1)];  end    if i > 2  % 计算当前时刻的系统输出  y(i) = - A(2)\*y(i-2) - A(1)\*y(i-1) + B(1)\*u(i-1) + B(2)\*u(i-2) + v(i)+ D(1)\*v(i-1) + D(2)\*v(i-2);  v\_h = [v\_hat(i-1); v\_h(1:end-1)];  phi(:, i) = [-y\_h; u\_h; v\_h];  L = P \* phi(:, i) / (lambda + phi(:, i)' \* P \* phi(:, i)); % 增益  theta\_hat(:, i) = theta\_hat(:, i-1) + L \* (y(i) - phi(:, i)' \* theta\_hat(:, i-1)); % 参数更新  P = (P - L \* phi(:, i)' \* P) / lambda; % 协方差矩阵更新  v\_hat(i) = y(i) - phi(:,i)' \* theta\_hat(:,i);    end    end  % 绘制输出  plot(1:N, theta\_hat(1, :), 'b-', 'LineWidth', 1.5);  hold on;  plot(1:N, theta\_hat(2, :), 'b--', 'LineWidth', 1.5);  hold on;  plot(1:N, theta\_hat(3, :), 'r-', 'LineWidth', 1.5);  hold on;  plot(1:N, theta\_hat(4, :), 'r--', 'LineWidth', 1.5);  hold on;  plot(1:N, theta\_hat(5, :), 'g-', 'LineWidth', 1.5);  hold on;  plot(1:N, theta\_hat(6, :), 'g--', 'LineWidth', 1.5);  xlabel('时间');  ylabel('参数值');  title('ARMAX系统辨识参数');  legend('a1', 'a2', 'b1', 'b2', 'd1', 'd2');  grid on; |

**仿真输出如下：**

