系统辨识实验

单容水箱系统建模

实验人员1：廖锦涛JL21060004

实验人员2：张笑一JL21010001

实验时间：2022年5月21日

一、实验题目：单容水箱系统建模

二、实验内容

本实验数据来源于对单容水箱系统输入输出数据的采集，采用某种辨识算法辨识出系统模型，并对结果进行分析。

三、实验目的

(1)学习系统辨识理论知识，掌握最小二乘辨识的方法。

(2)分析比较不同最小二乘辨识方法与效果。

(3)学习未知的实际系统建模过程中，数据的采集和预处理。

(4)学习 MATLAB 中 M 文件的编写，自主编写最小二乘辨识程序。

(5)分析辨识出系统和实际系统的偏差，对实验结果进行验证。

(6)在原系统基础上增加扰动，分析对比不同算法的抗扰能力。

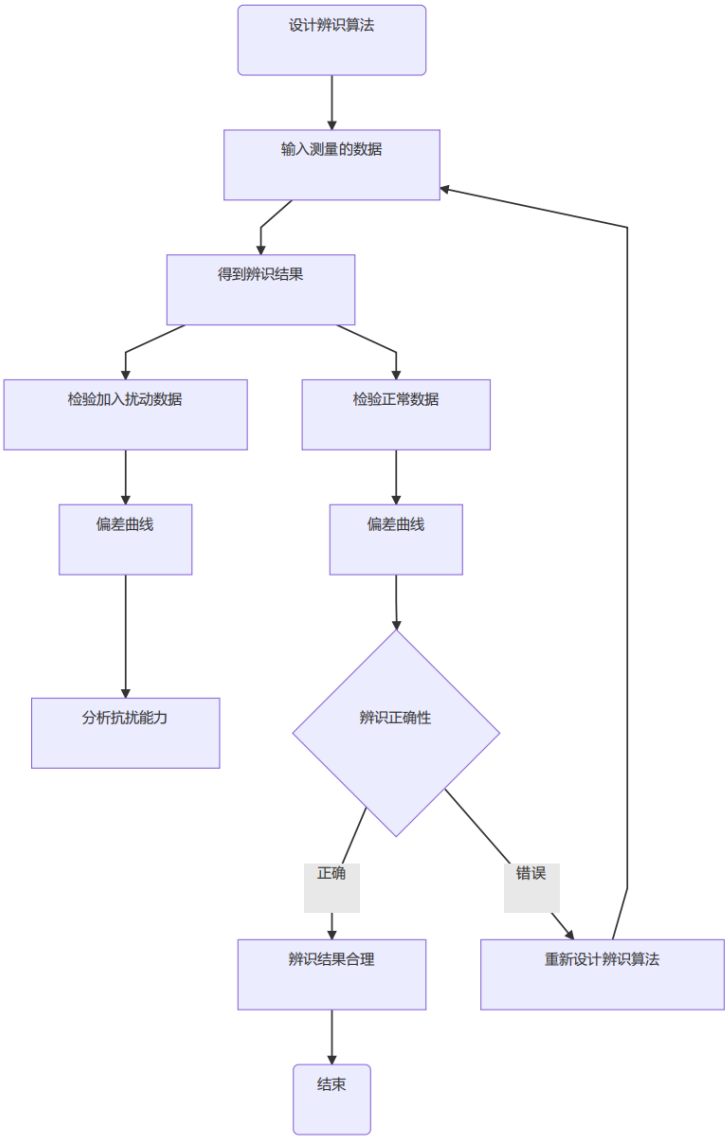
四、实验步骤

（1）采集单容水箱系统的输入输出数据；

（2）用已获得的输入输出数据，通过编好的辨识算法辨识已知系统的阶次，估计其参数，系统的时延通过观察实验数据直接获取，通过绘制辨识结果与实际输出的偏差曲线和误差变化曲线来验证所采用算法的正确性；

（3）给系统加入扰动，使用原系统参数对加入扰动后的数据进行拟合，观察抗扰动效果。

我们小组先从理论的角度,了解掌握系统辨识的方法，掌握最小二乘的原理及算法，再引入递推最小二乘方法，大大降低了计算的复杂程度。鉴于系统在一开始并未稳定，希望降低老数据的重要程度，提高新数据的比重，所以本次实验采用渐消记忆的最小二乘的方法。该方法得到了较好的实验效果。编写渐消记忆最小二乘算法，分析结论，验证结果。同时编写增广最小二乘法，与渐消记忆最小二乘算法的实验结果相比较，分析结论，验证结果。将已经获得的输入输出数据，通过辨识算法，估计未知的水箱系统的参数。为了验证实验结果的正确性，需要绘制辨识系统的输出和实际系统的偏差曲线，并绘制误差变化曲线图，便于观察误差变化。最后对原系统加入扰动，并用原系统的参数拟合扰动曲线，观察抗扰能力。



实验流程

五、实验原理

5.1数据预处理

所有的数据文件第一行是时间，导入时，输入从第二行开始；从数据中观察得到延时，输出数据从减去延时开始导入

5.2最小二乘法的介绍

5.2.1线性系统模型的提出

考虑系统稳态模型，系统稳定运行时，在同一时刻系统输入、输出的关系：

n 个输入（自变量）：$u_i = 1, 2, \ldots\ldots, n$

1 个输出（因变量）：y

N：数据组总数(数据长度,样本数) 每一次观测，得到一组数据，构成一个观测方程。N 次观测，得到 N 组数据，构成N 个观测方程：

$$
\begin{aligned}
y(1) &= a_0 + a_1u_1(1) + a_2u_2(1) + \ldots\ldots + a_nu_n(1) \\
y(2) &= a_0 + a_1u_1(2) + a_2u_2(2) + \ldots\ldots + a_nu_n(2) \\
&\ldots\ldots \\
y(N) &= a_0 + a_1u_1(N) + a_2u_2(N) + \ldots\ldots + a_nu_n(N)
\end{aligned}
$$

引入测量误差$e（k）$，有

$$y(k)=a_0+a_1u_1(k)+a_2u_2(k)+a_nu_n(k)+e(k)(k=1,2.....N)$$

$e（k）$也称为观测误差、模型误差、模型残差、噪声

$$e(k)=y(k)-a_0-a_1u_1(k)-a_2u_2(k)-.....-a_nu_n(N)$$
$$\theta=(a_0,a_1,.....,a_n)^T$$
$$Y_N=\Phi_N\theta+e_N$$

N 组数据获得的参数估计：

$$\hat{\theta}(N)=(\Phi_N^T\Phi_N)^{-1}\Phi_N^TY_N=\hat{\theta}$$

观测数据矩阵：

$$\Phi_N=\begin{bmatrix}\phi_1^T\\ \cdots\\ \phi_{n-1}^T\\ \phi_n^T\end{bmatrix}=\begin{bmatrix}1 & \cdots & u_{n-1}(1) & u_n(1)\\ \cdots & \cdots & \cdots & \cdots\\ 1 & \cdots & u_{n-1}(N-1) & u_n(N-1)\\ 1 & \cdots & u_n(N) & u_n(N)\end{bmatrix}$$

5.2.2 递推最小二乘法

递推最小二乘的介绍

采用在线数据进行线性稳态模型的最小二乘辨识，由N组老数据建立的观测方程：

$$Y_N=\Phi_N\theta+e_N$$

由 N 组老数据获得的参数估计：

$$\hat{\theta}(N)=(\Phi_N^T\Phi_N)^{-1}\Phi_N^TY_N=\hat{\theta}$$

新获得一组观测数据：

$$y(N+1)$$
$$u_1(N+1),.....,u_n(N+1)$$

我们令$P_N=(\Phi_N^T\Phi_N)^{-1}$

所以$P_{N+1}=(P_N^{-1}+\Phi_{N+1}\Phi_{N+1}^T)^{-1}$

我们最终得到 $n+1$次参数估计，

$$\theta(N+1)=\theta(N)+\frac{P_N\phi_{N+1}\phi_{N+1}^T}{1+\phi_{N+1}^TP_N\phi_{N+1}}[y(N+1)-\phi_{N+1}^T\phi_{N+1}^T\theta(N)]$$

将 $[y(N+1)-\phi_{N+1}^T\phi_{N+1}^T\theta(N)]$称为新息，即采用老估计值 $\theta(N)$时，对新的观测数据的拟合误差,参数θ的新估计值=老估计值+修正值。

我们令

$$K_{N+1}=\frac{P_N\phi_{N+1}\phi_{N+1}^T}{1+\phi_{N+1}^TP_N\phi_{N+1}}$$
$$\theta(N+1)=\theta(N)+K_{N+1}[y(N+1)-\phi_{N+1}^TP_N]$$
$$P_{N+1}=P_N-K_{N+1}\phi_{N+1}^TP_N$$

估计初值 $\hat{\theta}(0)$和P_0的选取

1、对前N组数据采用批量最小二乘

$$\hat{\theta}(N)=(\Phi_N^T\Phi_N)^{-1}\Phi_N^TY_N$$

$$P_N = (\Phi_N^T \Phi_N)$$

以此作为初值 $\hat{\theta}(0)$，P_0，从第N+1步开始递推

2、选取：$\hat{\theta}(0) = (0, 0, \ldots\ldots, 0)^T$,$P_0 = \alpha I$,其中$\alpha > 0$，认为$\alpha$是一个足够大的数，一般取$10^2 - 10^6$

依此作为初值，递推 N 步后，有：

$$P_N^* = [P_0^{-1} + \Phi_N^T \Phi_N]^{-1}$$

注意到有 $\lim\limits_{\alpha\to\infty} P_0^{-1} = \lim\limits_{\alpha\to\infty} \frac{1}{\alpha} I = 0$

即：$\lim\limits_{\alpha\to\infty} P_N^* = [\Phi_N^T \Phi_N]^{-1} = P_N$

当α足够大时，初值$P_0 = \alpha I$对P_N影响很小。

$$\hat{\theta}(N) = P_N^* [\Phi_N^T Y_N + P_0^{-1} \hat{\theta}_0]$$

$$\because \lim_{\alpha\to\infty} P_0^{-1} = 0$$

$$\lim_{\alpha\to\infty} \theta(N) = \lim_{\alpha\to\infty} P^* [\Phi_N^T \Phi_N] = P_N \Phi_N^T Y_N = (\Phi_N^T \Phi_N)^{-1} \Phi_N^T Y_N$$

当α足够大时，初值$P_0 = \alpha I$、$\hat{\theta}_0 = 0$对$\hat{\theta}(N)$影响很小，不必担心具体取值，算法本身是收敛的

5.2.3渐消记忆最小二乘法

随着数据的增长，$P(k)$越来越小，最后趋于零，最小二乘辨识算法就会逐渐失去修正能力，这种现象称作数据饱和。因为$P(k)$矩阵是用来累积信息量的，随着时间的增长，旧数据的信息在矩阵$P(k)$中不断累积，到了一定程度新数据的信息就无法在增添进去，以致算法会逐渐失去修正能力。对时变系统来说，由于旧数据信息的累积，同样的原因最小二乘辨识算法也会逐渐失去跟踪能力。不论是数据饱和问题，还是时变跟踪能力问题，都是因为数据信息量的不断累积，得不到衰减，从新数据中获取的信息量下降，影响算法的修正能力和跟踪能力。 遗忘因子法就是为克服数据饱和现象和解决时变跟踪问题而提出的一种辨识方法，其基本思想是对旧数据加遗忘因子，降低旧数据信息在矩阵 $P(k)$中的占比， 增加新数据信息的含量。 对N组观测数据，有目标函数：

$$min \quad J_N(\theta) = \sum_{k=1}^{N} \lambda^{N-k} e^2(k) = e^2(N) + \lambda e^2(N-1) + \ldots\ldots + \lambda^{N-1} e^2(1)$$

其中，λ为遗忘因子，$0 < \lambda \leqslant 1$,在新的目标函数中，由于引入小于 1 的遗忘因子，对新数据的拟合误差最小被重视，而对老数据的拟合误差逐渐被淡忘。

最终得到渐消记忆最小二乘辨识算法：

$$\hat{\theta}_{N+1} = \hat{\theta}_N + K_{N+1}[y(N+1) - \phi_{N+1}^T \hat{\theta}_N]$$

$$K_{N+1} = \frac{P_N \phi_{N+1}}{\lambda + \phi_{N+1}^T P_N \phi_{N+1}}$$

$$P_{N+1} = \frac{1}{\lambda}[P_N - K_{N+1} \phi_{N+1}^T P_N]$$

遗忘因子$0 < \lambda \leqslant 1$，一般取$0.95 \leqslant \lambda \leqslant 1$，当λ=1,为递推最小二乘法。遗忘因子越小，越重视新数据，对新的拟合误差加权越重，算法能较好地跟踪系统参数的变化；遗忘因子越小，参数估计收观测噪声、干扰的影响越大，因而参数估计值的波动越大；

5.2.4增广最小二乘法

增广最小二乘法：当噪声均值为0时，最小二乘法参数估计算法为无偏估计，为了解决最小二乘法参数估计的有偏性，将噪声模型的辨识同时考虑进去，就是增广最小二乘法。增广最小二乘法相当于参数向量和数据向量维数扩大了的最小二乘法，它能在有色噪声(可用平均滑动模型来表示)情况下给出参数的一致估计量，同时可以把噪声模型也辨识出来。

如果误差是可测量的，则可以将误差扩充到输入向量中去，称扩充后的向量为增广向量，记

$$\phi_{t-1}=[-y_{t-1},\ldots\ldots,-y_{t-p},\epsilon_{t-1},\ldots\ldots,\epsilon_{t-q}]^T$$
$$\theta=[a_1,\ldots\ldots,a_p,b_1,\ldots\ldots,b_q]^T$$

则有$y_t=\phi_{t-1}^T\theta+\epsilon_t$ 。可用递推最小二乘法求解系数 θ。但是实际上，每一步的误差ϵ都是不精确测量的， ϕ_t 也不是精确的，因而只能用 ϕ的估计量来替代：

$$\hat{\phi}_{t-1}=[-y_{t-1},\ldots\ldots,-y_{t-p},\hat{\epsilon}_{t-1},\ldots\ldots,\hat{\epsilon}_{t-q}]^T$$

因而求出估计量 $\hat{\theta}_t$。很多时候$\hat{\theta}_t$都能收敛到θ，所以该方法是有价值的！

用t时刻的拟合误差e_t作为未知的不可测噪声ξ_t的估计值：$e_t=y_t-\phi_t^T\hat{\theta}_{t-1}$（预测形式）

参数变量：$\theta=(a_1,a_2,\ldots\ldots,a_{n_a},b_0,b_1,\ldots\ldots,b_{n_b},c_1,c_2,\ldots\ldots,c_{n_c})^T$

t时刻的估计参数向量：$\hat{\theta}=(\hat{a}_1,\hat{a}_2,\ldots\ldots,\hat{a}_{n_a},\hat{b}_0,\hat{b}_1,\ldots\ldots,\hat{b}_{n_b},\hat{c}_1,\hat{c}_2,\ldots\ldots,\hat{c}_{n_c})^T$

t时刻的增广观测向量：$\phi_t=(-y_{t-1},\ldots\ldots,-y_{t-n_a},u_{t-k},\ldots\ldots,u_{t-k-n_b},e_{t-1},\ldots\ldots,e_{t-n_c})^T$

t时刻不可测噪声的估计值e_t:

$$e_t=\begin{cases}0, & x<0\\ y_t-\phi_t^T\hat{\theta}_t, & x\geq 0\end{cases}$$
$$K_t=\frac{P_{t-1}\phi_t}{\lambda+\phi_t^TP_{t-1}\phi_t}$$
$$\hat{\theta}_t=\hat{\theta}_{t-1}+K_t(y_t-\phi_T^T\hat{\theta}_{t-1})$$
$$P_t=\frac{1}{\lambda}(I-K_t\phi_t^T)P_{t-1}$$

5.3．算法的设计

5.3.1渐消记忆最小二乘法

在设计渐消记忆最小二乘法时，我们采用 CAR模型，选择二阶系统。

$$CAR\quad Mode:y(k)=-\sum_{i=1}^{n}a_iy(k-1)+\sum_{i=1}^{n}b_iu(k-i)+w(k)$$
$$Y_N=\Phi\theta+e_N$$

对于二阶系统而言，渐消记忆的递推最小二乘：参数向量 $\theta=[a_1,a_2,b_1,b_2]$,观测向量：
$\Phi(k)=[-y(k-1),-y(k-2),u(k-1),u(k-2)]^T$

本次实验中加入遗忘因子λ,得到新的矩阵：

$$\theta(k)=\theta(k-1)+K(k)[y(k)-\phi^T(k)\theta(k-1)]$$
$$K(k)=P(k-1)\phi(k)[\phi^T(k)P(k-1)\phi(k)+\lambda]^{-1}$$
$$P(k)=[I-K(k)\phi^T]P(k-1)/\lambda$$

如果是$\lambda=1$，变成递推最小二乘。

5.3.2增广最小二乘法

设计增广最小二乘算法，选择受控自回归滑动平均模型$CARMA$模型，选择二阶系统。

$$CARMA\quad Mode:y(k)=-\sum_{i=1}^{n}a_iy(k-1)+\sum_{i=1}^{n}b_iu(k-i)+\sum_{i=1}^{n}c_i\xi(k-i)+w(k)$$

对于二阶系统而言，增广最小二乘：参数向量：$\theta=[a_1,a_2,b_1,b_2,c_1,c_2]$

观测向量：$\Phi(k)=[-y(k-1),-y(k-2),u(k-1),u(k-2),\xi(k-1),\xi(k-2)]^T$

$RELS$辨识算法：

$$\hat{\theta}(k)=\hat{\theta}(k-1)+K(k)[z(k)-h^T(k)\hat{\theta}(k-1)]$$
$$K(k)=P(k-1)h(k)[h^T(k)P(k-1)h(k)+1]^{-1}$$
$$P(k)=[I-K(k)h^T]P(k-1)$$

初始值取为：$P_0=aI$，a取为10^2-10^6，$\hat{\theta}_0=\theta_0=(0\quad 0\quad \cdots\quad 0)^T$

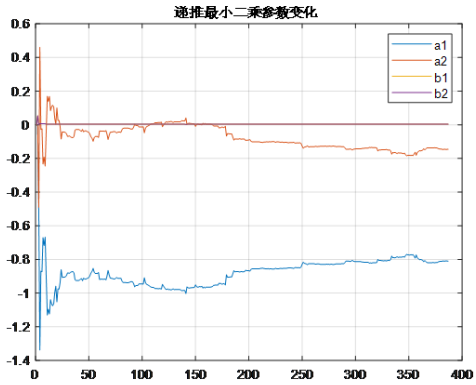
六、实验结果和分析

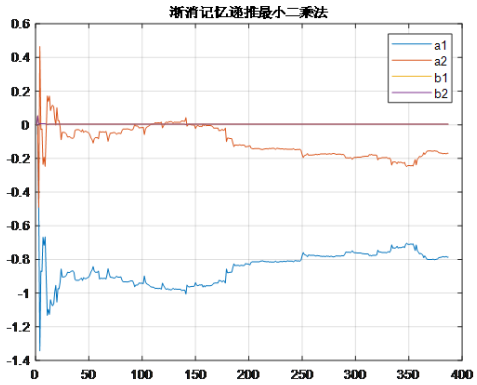
6.1辨识结果

6.1.1渐消记忆递推最小二乘

选取辨识参数的最后五组数据的平均值作为系统模型参数

参数	a_1	a_2	b_1	b_2
$\lambda=1$	-0.8101	-0.1458	0.0032	0.0032
$\lambda=0.996$	-0.7865	-0.1688	0.0035	0.0035



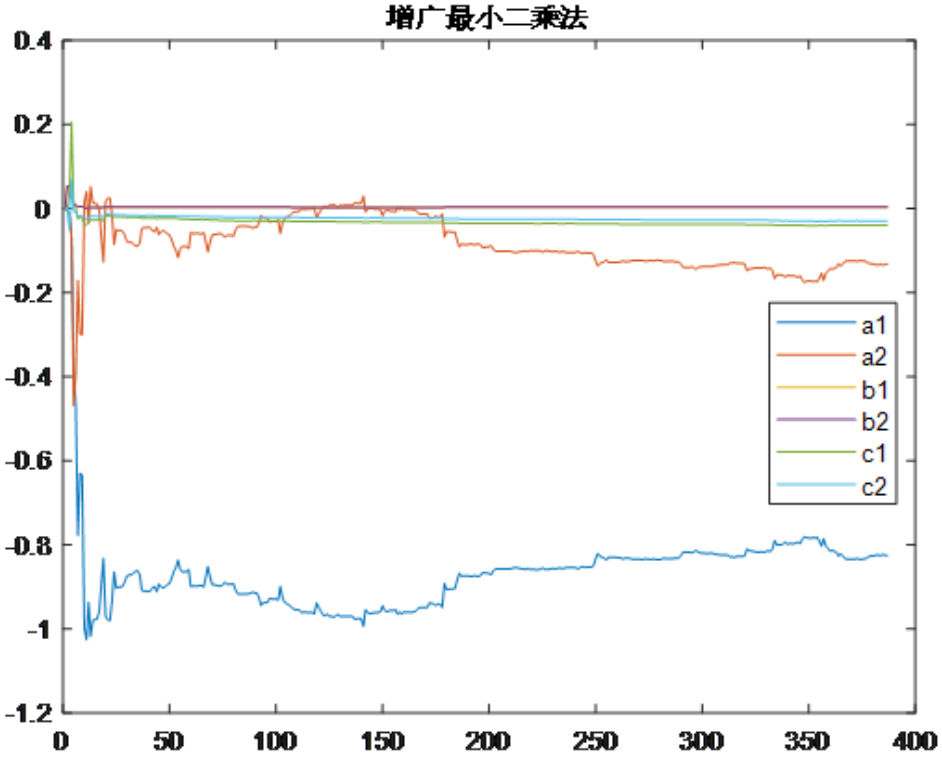


在对程序设计时， 对于矩阵P的选取必须是正定，且较大的值，在多次测试后，发现10^6是效果最好的，这也与理论经验相符合。在θ参数的初始化中，我们先选择了都为3的初始值，发现效果并不是很好，没有足够多的数据量时，迭代次数不够，会导致参数出现严重偏差，最后还是选择了0作为初始值。因为本实验中，延时是通过对数据观察得到的，但是每组实验数据的时延存在一定的误差，这可能是在开始实验阶段人为控制导致的，所以是调试过程中，本实验效果最好的时延在20s附近，当然也有数据的时延比较小，比如后续的验证中，有一组测试数据的时延为8s左右。时延的错误选取会导致训练时加入不必要的干扰数据，影响数据的递推效果，并且在数据拟合阶段会导致辨识数据与真实数据存在严重偏差，所以时延对实验的效果影响很大。 最后对比递推最小二乘和渐消记忆最小二乘，辨识出来的参数差距不大，我们选择能够对新数据起更大作用的渐消记忆算法作为后续的验证参数。其误差为$e^2=0.0025878040$ 方差 :0.0000066868

6.1.2增广最小二乘

选取辨识参数的最后五组数据的平均值作为系统模型参数

a_1	a_2	b_2	b_2	c_1	c_2
-0.7911	-0.1656	0.0034	0.0034	-0.0052	-0.0060



在设计增广最小二乘时，与递推最小算法类似，只是在递推算法的模型基础上增加了白噪声模型，修改了迭代公式，将噪声也加入参数计算中。在后续的验证中，选取实验中误差较小的一组数据参数作为后续的验证参数。增广算法误差：$e^2 = 0.0065536480$ 方差：0.0000171114

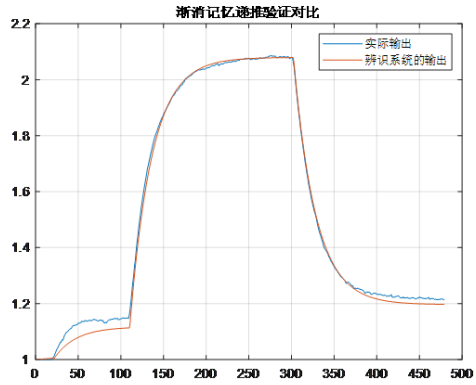
6.2结果验证

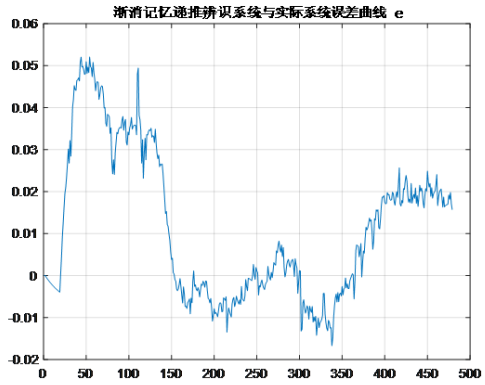
6.2.1渐消记忆最小二乘

选取参数为：$\theta = [-0.7865, -0.1688, 0.0035, 0.0035]^T$

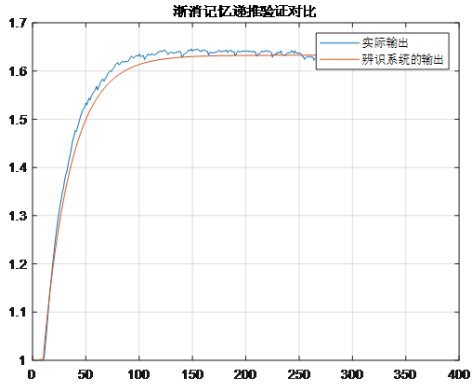
系统模型为：$y(k) = -0.7865y(k-1) - 0.1688y(k-2) + 0.0035u(k-1) + 0.0035u(k-2)$

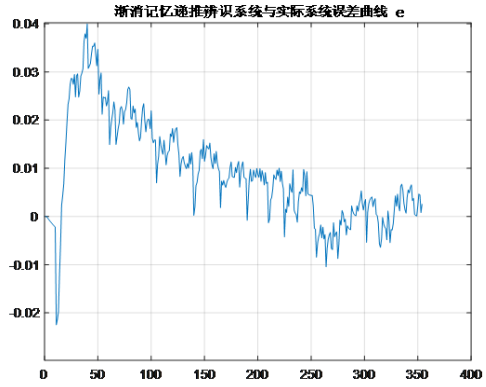
通过该模型对原系统进行辨识，得到以下对比图





误差$e^2 = 0.2136594985$ 方差0.0004460532





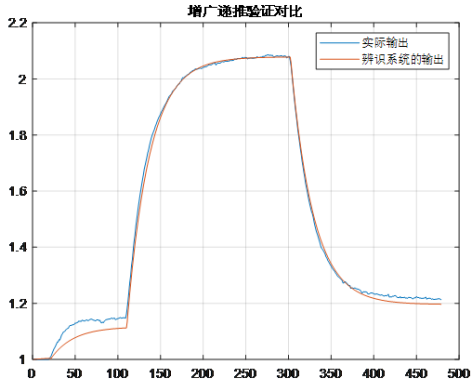
$$\text{误差}e^2 = 0.0710323991 \quad \text{方差}0.0002006565$$

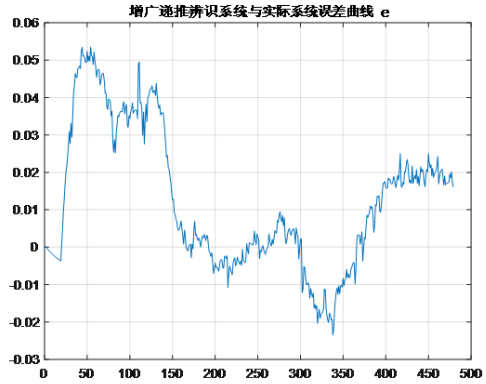
对比两组检验，在实验过程中改变输入而影响输出产生阶跃变化的数据拟合的效果没有未改变输入的效果好，但是两组检验的误差都不大，第一组数据最终误差在0.02上下波动，第二组数据的误差最终趋于0，可以说明辨识出来的模型参数是合理的。

6.2.2增广递推最小二乘

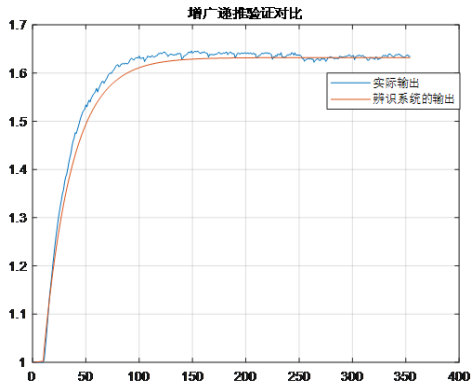
选取训练参数$\theta = [-0.7911, -0.1656, 0.0034, 0.0034, 0.0052, 0.0060]^T$

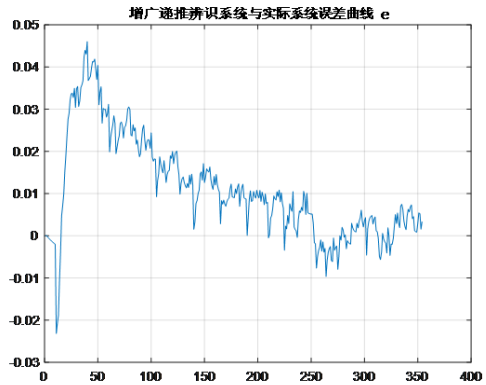
系统模型为：
$y(k) = -0.7911y(k-1) - 0.1656y(k-2) + 0.0034u(k-1) + 0.0034u(k-2) + 0.0052\xi(k-1) + 0.0060\xi(k-2)$





$$\text{误差}e^2 = 0.1757348177 \quad \text{方差}0.0003668785$$





$$\text{误差}e^2 = 0.0324479973 \quad \text{方差}0.0000916610$$

利用增广递推算法辨识的参数进行验证效果与渐消记忆相差不大，趋势都大致相同，也同样较好的拟合了原系统模型，误差也在一定的容许范围内，但是通过计算误差可以发现微小的差距，增广递推算法所得到的误差更小一点，并且多次实验得到的结果也是如此，可以说明增广递推算法的泛化能力相对更好。

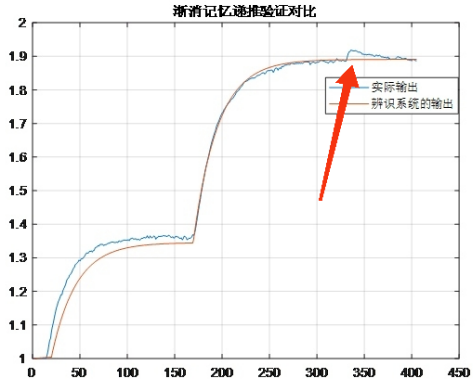
6.3验证加入扰动的数据

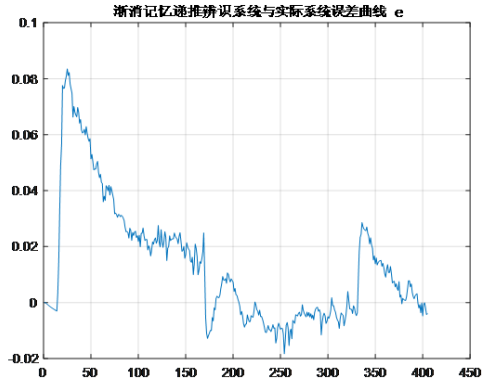
在本实验中，我们选择在第二次改变输入后（即第二个阶跃处）轻微旋转阀门，再迅速还原，得到如图红色箭头处凸起的扰动，因为本实验所采用的两个算法渐消记忆递推和增广递推都是属于递推算法，参数是实时更新的，所以有很强的抗扰能力，从以下的验证中也可以发现这一特点。

与上述未加干扰的正常数据检验采取相同的参数。

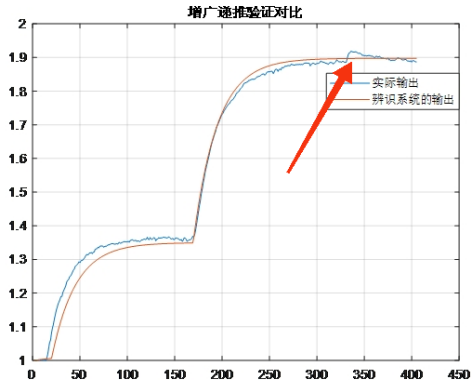
渐消记忆模型：$y(k)=-0.7865y(k-1)-0.1688y(k-2)+0.0035u(k-1)+0.0035u(k-2)$

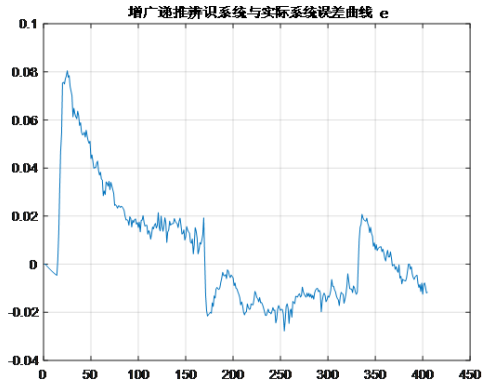
增广模型：
$y(k)=-0.7911y(k-1)-0.1656y(k-2)+0.0034u(k-1)+0.0034u(k-2)+0.0052\xi(k-1)+0.0060\xi(k-2)$





误差$e^2=0.2704154898$ 方差0.0006676926





误差$e^2=0.2329327028$ 方差0.0005751425

对比于未加扰动的数据，从误差中可以看出，加了扰动后效果明显更差了，这是合理的，因为在扰动处无法正常拟合，必然会导致误差增大，但根据验证图片以及计算的误差也能发现对于加了扰动的数据仍然有较好的辨识效果，并且最终误差也是趋于0的，只是在扰动处有较大的误差变化，可以说明递推算法的抗扰能力强。再对比两种算法，两种算法的拟合情况相差不大，误差变化也相似，但渐消记忆递推曲线拟合的更完美，但是起始阶段误差更大，最终的误差也相对更大。为了有更明显的对比，再次使用第二组数据进行验证（这里不再增加验证图，数据名为：rd_2.mat）计算得到增广递推：误差:0.4226276311 方差:0.0008860118；渐消记忆递推：误差0.4878451336 方差 0.0010227361，结果表明，增广递推的误差仍然更小，可以说明增广递推最小二乘算法相对于渐消记忆递推最小二乘算法有更好的抗扰动能力，即鲁棒性更强。

七、实验总结

感谢秦老师这次实验的倾情指导，我们小组收获良多。通过这次实验我们对系统辨识有了更深入的了解。系统辨识是研究如何利用系统实验或运行的、含有噪声的输入输出数据来建立被研究对象数学模型的一种理论和方法。系统辨识是建模的一种方法，不同的学科领域，对应着不同的数学模型。从某种意义上来说，不同学科的发展过程就是建立他的数学模型的过程。辨识问题可以归结为用一个模型来表示客观系统(或将要构造的系统)本质特征的一种演算，并用这个模型把对客观系统的理解表示成有用的形式。辨识有三个要素：数据，模型和准则。辨识就是按照一个准则在一组模型类中选择一个与数据拟合的较好的模型。总而言之，辨识的实质就是从

一组模型类中选择一个模型，按照某种准则，使之能最好地拟合所关心的实际过程的静态或动态特性。在这次实验中，秦老师在实验前详细地给我们进行了实验的具体步骤，在我们实验过程中也一直为我们答疑解惑。我们的实验过程存在些许缺陷。其一，我们在采集数据过程中非常随意，简单粗暴认为数据随意更具有代表性，采集了很多奇怪的数据，并且没有注意到采样周期，采取的数据采样周期没有控制为相同值，并且在增加扰动时，数据未达到稳定就随意的加入干扰，导致数据区分度很低。所以在刚开始设计算法阶段，训练能够获得误差较低的参数，但是进行验证时，总不能与原系统较好的拟合，误差特别大，很长一段时间没有得到解决。在与同学讨论也没有发现算法的问题，最终选择采用他们的数据，才发现原来我们采集的数据存在问题，所以与老师重新约了个时间，重新采集了数据，这时才能完成基本的实验要求。其二，我们对各种最小二乘法的算法不是很熟悉，导致实验进度较缓慢，对算法原理没有很透彻的理解，在调试中花费了大量时间才得到较好的效果。其三，我们在第一次做实验时，没有认识到最小二乘算法进行系统辨识的基本原理，我们最初都在连续系统方面进行探索，做实验一头雾水，直到第二次采集数据时才真正了解了算法进行辨识的是离散的差分方程。所以在进行一次实验之前，我们应该更清楚的去学习实验所采用的算法基本原理，才能更好的帮助完成实验。我们小组在学习过程中，互相给予鼓励，团结协作，最终完成实验操作和代码完善。通过这次实验，我们小组对系统辨识和模型校验都有了更深的了解。再次真诚感谢秦老师的教导！

八、附录

附录1

```
%MLS.m
%渐消记忆递推最小二乘 JL21060004 廖锦涛
clc,clear;
close all;
load train.mat
datalength=data(1,1);%数据总长
delay=20;%时延
mu=0.996;%遗忘因子
length=datalength-delay;%减去时延后的数据总长
u=data(2:length+1,2);%输入
y=data(2+delay:datalength+1,3);%输出

%递推算法求解参数
%参数初始化
theta = zeros(4,1);
Theta = zeros(4,1);
J=0;
P = 10^6*eye(4);%P一般选取10^6
for k = 3:length-1
    h = [-y(k-1) -y(k-2) u(k-1) u(k-2)]';%将二阶系统每一次的输入、输出导入进行迭代
    K = P*h/(mu + h'*P*h);       %更新K
    P = (P - K*h'*P)/mu;         %更新P
    J=J+(y(k) - h'*theta)^2/(mu + h'*P*h);  %计算准则函数
    theta = theta + K*(y(k) - h'*theta);    %更新辨识参数a1,a2,b1,b2
    Theta = [Theta,theta];                  %存放辨识参数
end
%%计算误差与绘制参数曲线
L=length-2;%真值数据长度
y1=y(3:length);%真实值
y2=zeros(L,1);%估计值
a=size(y2);
for i=1:2
 y2=y2-Theta(i,a(1))*y(3-i:length-i);%先计算输出的辨识树池
end
for i=1:2
 y2=y2+Theta(i+2,a(1))*u(3-i:length-i);%计算输入的辨识输出
end
error=y1-y2;%残差向量
J=error'*error;
J2=J/L;%残差方差
if(mu==1)
```

```
    fprintf("递推最小二乘法 RLS\n");
else fprintf("渐消记忆递推最小二乘法 MLS\n");
end
fprintf("误差: J = %.10f\n",J);
fprintf("方差: J2 = %.10f\n",J2);

sizeiden=size(Theta);
k=1:sizeiden(2);
plot(k,Theta');grid;
if(mu==1)
    title("加扰动递推最小二乘参数变化 ");
else title('加扰动渐消记忆递推最小二乘法')
end
legend('a1','a2','b1','b2');
```

附录2

```
%ELS.m
%增广最小二乘递推算法 JL21010001 张笑一
clear;clc;
close all
%% 导入数据
load train.mat
datalength=data(1,1);%数据总长
u=data(2:datalength,2);%输入数据
v=randn(1,datalength);%构造随机扰动
delay=20;%时延
length=datalength-delay;%减去时延后总长
y=data(2+delay:datalength,3);

%% 递推求解
P=10^6*eye(6); %选取P的初始值为100-10^6之间，一般选10^6
Theta=zeros(6,length-2);      %参数的估计值，存放中间过程估值
% K=[10;10;10;10;10;10];      %初始化K矩阵
for i=3:length-1
    h=[-y(i-1);-y(i-2);u(i-1);u(i-2);v(i-1);v(i-2)];%将二阶系统每一次的输入、输出、扰动导入进
行迭代
    K=P*h*inv(h'*P*h+1);     %更新K值
    P=(eye(6)-K*h')*P;       %更新P值
    Theta(:,i-1)=Theta(:,i-2)+K*(y(i)-h'*Theta(: ,i-2));%计算参数向量
[a1,a2,b1,b2,c1,c2]，全部放在Theta矩阵中
    v(i)=y(i)-h'*Theta(:,i-1);  %更新扰动
end

%% 计算误差和绘制参数辨识结果
L=length-2;%真值数据长度
y1=y(2:length-1);%真实值
y2=zeros(L,1);%估计值
a=size(y2);
for i=1:2 %1-2
 y2=y2-Theta(i,a(1))*y(3-i:length-i);%先计算输出的前两阶
end
for i=1:2
 y2=y2+Theta(i+2,a(1))*u(3-i:length-i);%计算输入的前两阶
end
error=y1-y2;%残差向量
J=error'*error;
J2=J/L;%残差方差
fprintf("增广最小二乘法\n");
fprintf("误差: J=%.10f\n",J);
```

```
fprintf("方差: J2=%.10f\n",J2);

%绘制辨识参数曲线
sizeiden=size(Theta);
i=1:sizeiden(2);
figure (1)
plot(i, Theta(1, :), i,Theta(2,:) , i,Theta(3,:), i, Theta(4,:), i, Theta(5,:),
i,Theta(6,:))
title('增广最小二乘法')
legend('a1','a2','b1','b2','c1','c2');
```

附录3

```
%test.m
%效果验证
close all
clearvars -except Theta   %只保留辨识出来的参数进行检验

%% 取参数的最后五个数据的平均值作为检验参数
sizeiden=size(Theta);
ls=Theta(1:4,sizeiden(2)-4:sizeiden(2));
ls=mean(ls,2);

%% 读取数据与处理
load rd_2.mat
datalength=data(1,1); %数据总长
delay=18;  %观察所得延时时间 test1时延为20，test2时延为8

length=datalength-delay-2;  %真实输出数据的长度
u=data(2:datalength+1,2);%输入
y=data(2:datalength+1,3); %输出
u0=u(3:length+2);%u(t)
u1=u(2:length+1);%u(t-1)
u2=u(1:length);%u(t-2)

%% 对原始数据进行拟合，得到辨识后的输出
y2(1:delay+2,1)=y(1:delay+2);
for k=3:datalength
    i=k-delay-2;  %保证从延时之后的时间开始拟合
    if(i>0)
        y2(k,1)=[-y2(k-1,1) -y2(k-2,1) u1(i) u2(i)]*ls;
    else          %延时之前的输入设定为初始输入
        y2(k,1)=[-y2(k-1,1) -y2(k-2,1) 6.5 6.5]*ls;
    end
end
%% 原辨识系统的结果对比
t=1:datalength;
figure(1);
plot(t,y);hold on;
plot(t,y2);hold off;
grid;
title("渐消记忆递推验证对比");
legend('实际输出','辨识系统的输出');

figure(2)
yerror=y-y2;    %计算辨识曲线与原曲线的误差
plot(t,yerror);grid; J=yerror'*yerror;%检验误差J=e^2
Jvc=J/datalength;
title("渐消记忆递推辨识系统与实际系统误差曲线 e");
fprintf("检验误差: J = %.10f\n",J);
fprintf("检验方差: Jvc = %.10f\n",Jvc);
```