2.3 Recursive Least Squares Method

■ Concept of online identification

A computer participates in data acquisition, processing and system identification, and constantly modifies the identification results, which is also called online identification.

The identification technology has been developed with the development of digital electronic computer technology since the 1960s. It has two obvious characteristics:

- First, make full use of the new data collected each time to update the identification results continuously;
- Second, all operations of the identification process are required to be completed between the two sampling times. According to this, online identification is also called real-time identification.

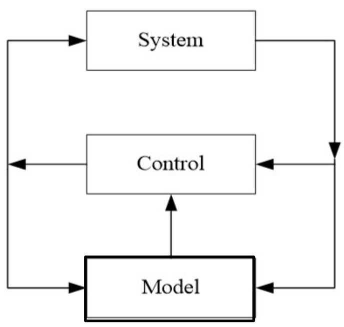
The online identification method is often used in adaptive control and prediction, which can overcome the failure caused by time-varying and outdated data.

1

To reduce the requirement of computer memory and improve real-time performance, online identification algorithm generally appears in the form of a time recursive algorithm

Block diagram of online identification

- On the one hand, the online model provides guidance for the operation of the control system.
- On the other hand, the online model continuously corrects itself through real-time input and output data.

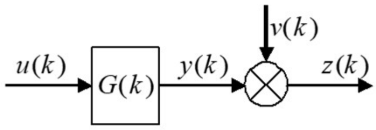


2

♦ **Principle of the recursive least squares method**

Generally, the least squares or weighted least squares method is a one-time algorithm or a batch algorithm.

It needs a large amount of calculation and large storage, and is not suitable for online identification.



The parameters are recursively estimated—the recursive least squares algorithm.

Current estimate $\hat{\theta}(k)$ = Previous estimate $\hat{\theta}(k-1)$+ Correction (2.3.1)

■ Algorithm format of online identification

Generally, identification can be considered to be a mapping from data Z^k to parameters θ

$$\hat{\theta}_k = F(k, Z^k)$$

However, this form cannot be applied to online identification algorithms because

— the size of Z^k is not fixed,

— the calculation time is unpredictable

For online identification, the following algorithm format needs to be met

$$X(k) = H(k, X(k-1), y(k), u(k))$$
$$\hat{\theta}_k = h(X(k))$$

where

— $X(k)$ can be regarded as the current identification state,

— it is obtained from the previous state $X(k-1)$ and the sampled $y(k)$ and $u(k)$,

— the time of each cycle is fixed.

■ Online least squares method

In the previous sections, least squares estimation was introduced

$$\hat{\theta}_k = \arg\min \sum_{i=1}^{k} [y(i) - \phi^T(i)\theta\ ]^2$$

where

$$\hat{\theta}_k = R^{-1}(k)f(k) \tag{2.3.2}$$

$$R(k) = \sum_{i=1}^{k} \phi(i)\phi^T(i)$$

$$f(k) = \sum_{i=1}^{k} \phi(i)y(i)$$

If the expression in (2.3.2) is used to calculate the least squares estimation, it is necessary to know all historical data series before time k

Suppose we know the estimated $\hat{\theta}_{k-1}$ of θ at $k-1$, can it be applied to calculate $\hat{\theta}_k$ of the model with latest data $\mathrm{y}(k)$ and $\mathrm{u}(k)$ at time k and $\hat{\theta}_{k-1}$?

It can be seen from (2.3.2) that $f(k)$ and $f(k-1)$ are related to each other, $R(k)$ and $R(k-1)$ as well.

The relationship between time k and time k-1 can be obtained from historical data

$$R(k) = R(k-1) + \phi(k)\phi^T(k) \tag{2.3.3}$$

$$f(k) = f(k-1) + \phi(k)y(k) \tag{2.3.4}$$

Following (2.3.1), the online least squares parameter identification method can be derived.

It can be concluded from (2.3.2), (2.3.3) and (2.3.4)

$$\begin{aligned}\hat{\theta}_k &= R^{-1}(k)f(k) = R^{-1}(k)[f(\mathrm{k}-1) + \phi(k)y(k)] \\ &= R^{-1}(k)\left[R(\mathrm{k}-1)\hat{\theta}_{\mathrm{k}-1} + \phi(k)y(k)\right] \\ &= R^{-1}(k)\left\{[R(k) - \phi(k)\phi^T(k)]\hat{\theta}_{\mathrm{k}-1} + \phi(k)y(k)\right\} \\ &= \hat{\theta}_{k-1} + R^{-1}(k)\phi(k)\left[y(k) - \phi^T(k)\hat{\theta}_{\mathrm{k}-1}\right]\end{aligned}$$

Therefore, it is obtained that

$$\hat{\theta}_k = \hat{\theta}_{k-1} + R^{-1}(k)\phi(k)\left[y(k) - \phi^T(k)\hat{\theta}_{k-1}\right] \tag{2.3.5}$$

$$R(k) = R(k-1) + \phi(k)\phi^T(k) \tag{2.3.6}$$

It meets the requirements of an online recursive algorithm given by (2.3.1).

7

To avoid the trouble caused by calculating $R^{-1}(k)$, let

$$P(k) = R^{-1}(k)$$

The following is the theorem of a matrix inverse

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B[DA^{-1}B + C^{-1}]^{-1}DA^{-1}$$

For (2.3.6), let $A = R(\mathrm{k}-1)$

$B = D^T = \phi(\mathrm{k})$

$C = 1$

Then

$$P(k) = P(k-1) - \frac{P(k-1)\phi(k)\phi^T(k)P(k-1)}{1 + \phi^T(k)P(k-1)\phi(k)}$$

8

According to the value of $P(k)$, we have

$$R^{-1}(k)\phi(k) = P(k-1)\phi(k) - \frac{P(k-1)\phi(k)\phi^T(k)P(k-1)\phi(k)}{1+\phi^T(k)P(k-1)\phi(k)}$$

$$= \frac{P(k-1)\phi(k)}{1+\phi^T(k)P(k-1)\phi(k)}$$

Let

$$K(k) = \frac{P(k-1)\phi(k)}{1+\phi^T(k)P(k-1)\phi(k)}$$

9

In this way, the formula of the recursive least squares method can be obtained below:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k)\left[y(k) - \phi^T(k)\hat{\theta}(k-1)\right] \quad (2.3.7)$$

$$K(k) = \frac{P(k-1)\phi(k)}{1+\phi^T(k)P(k-1)\phi(k)} \quad (2.3.8)$$

$$P(k) = P(k-1) - \frac{P(k-1)\phi(k)\phi^T(k)P(k-1)}{1+\phi^T(k)P(k-1)\phi(k)} \quad (2.3.9)$$

10

♦ **Initial value problem**

The initial values of P and θ can be selected

$$P(0) = P_0 \qquad \hat{\theta}(0) = \theta_i$$

where P_0 can be taken as a unit matrix, θ_i can be initialized with the preliminary results of offline identification.

For example, $\quad P(0) = \alpha I \qquad \hat{\theta}(0)=\varepsilon$

where α is a sufficiently large positive real number (*e.g.*, 10^4~10^{10}), ε is a zero vector or a sufficiently small positive real vector.

If the identification time is long enough, the initial values have little influence on identification results.

11

Self Assessment Question

Prove the following matrix inversion formula:

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B[DA^{-1}B + C^{-1}]^{-1}DA^{-1}$$

12

♦ **Time-delay systems**

Consider a time-delay system described by the ARX model:

$$A(q^{-1})y(k) = q^{-d}B(q^{-1})u(k) + e(k) \qquad \text{or} \qquad A(q^{-1})y(k) = B(q^{-1})u(k-d) + e(k)$$

where

$$A(q^{-1}) = 1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + b_2 q^{-1} + \cdots + b_{n_b} q^{-n_b}$$

d is a time-delay.

In a more concise form,

$$y(k) = \theta^T \phi(k) = \phi^T(k)\theta$$

where

$$\theta = \begin{bmatrix} a_1 & a_2 & \cdots & a_{n_a} & b_0 & b_1 & \cdots & b_{n_b} \end{bmatrix}^T$$

$$\phi(k) = [-y(k-1) \quad \cdots \quad -y(k-n_a) \quad u(k-d) \quad \cdots \quad u(k-d-n_b)]^T$$

♦ **Algorithm 2.3.1** (Recursive Least Squares Estimation, RLS)

Given n_a, n_b, and d.

Step 1: Set initial value $\hat{\theta}(0)$ and $\mathrm{P}(0)$, enter initial data.

Step 2: Sample the current output $\mathrm{y}(k)$ and input $\mathrm{u}(k)$.

Step 3: Use equations (2.3.7) –(2.3.9) to calculate $\mathrm{K}(k), \hat{\theta}(k)$ and $\mathrm{P}(k)$.

Step 4: $k \rightarrow k+1$, return to Step 2, and continue the loop.

■ Example

Consider the following system:

$$y(k) - 1.5y(k-1) + 0.7y(k-2) = u(k-3) + 0.5u(k-4) + \xi(k)$$

where $\xi(k)$ is a white noise with variance of σ^2.

Assuming input/output data $u(-3), u(-2), u(-1), u(0), y(-1)$, and $y(0)$ are all 0. The inverse M-sequence in section 1.6 serves as an input signal $u(k) = \{-1, -1, 1, -1, \ldots\}$ ($k = 1, 2, \ldots$), with an initial value $P(0) = 10^6 I$, $\hat{\theta}(0) = 0$, and set $\sigma^2 = 0$.

For $\lambda(k) = 1$,

$$\varphi(1) = [-y(0), -y(-1), u(-2), u(-3)]^T = [0,0,0,0]^T$$
$$y(1) = 1.5y(0) - 0.7y(-1) + u(-2) + 0.5u(-3) = 0$$
$$K(1) = \frac{P(0)\varphi(1)}{1 + \varphi^T(1)P(0)\varphi(1)} = 0$$
$$P(1) = [I - K(1)\varphi^T(1)]P(0) = 10^6 I$$
$$\hat{\theta}(1) = \hat{\theta}(0) + K(1)\left[y(1) - \varphi^T(1)\hat{\theta}(0)\right] = 0$$

$$\varphi(2) = [-y(1), -y(0), u(-1), u(-2)]^T = [0,0,0,0]^T$$
$$y(2) = 1.5y(1) - 0.7y(0) + u(-1) + 0.5u(-2) = 0$$
$$K(2) = \frac{P(1)\varphi(2)}{1 + \varphi^T(2)P(1)\varphi(2)} = 0$$
$$P(2) = [I - K(2)\varphi^T(2)(2)]P(1) = 10^6 I$$
$$\hat{\theta}(2) = \hat{\theta}(1) + K(2)\left[y(2) - \varphi^T(2)(2)\hat{\theta}(1)\right] = 0$$
$$\hat{\theta}(3) = 0$$
$$\hat{\theta}(4) = [0, 0, 1.0, 0]^T$$
$$\hat{\theta}(5) = [-1.0, 0, 1.0, 1.0]^T$$
$$\hat{\theta}(6) = [-1.1, -0.1, 1.0, 0.9]^T$$
$$\hat{\theta}(7) = [-1.4999, 0.6999, 1.0, 0.5001]^T$$
$$\hat{\theta}(8) = [-1.5, 0.7, 1.0, 0.5]^T$$

From the above derivation, it can be seen that when $k = 8$, the estimated values of the parameters identified by the RLS algorithm are identical to the true values.

♦ **Simulation results**

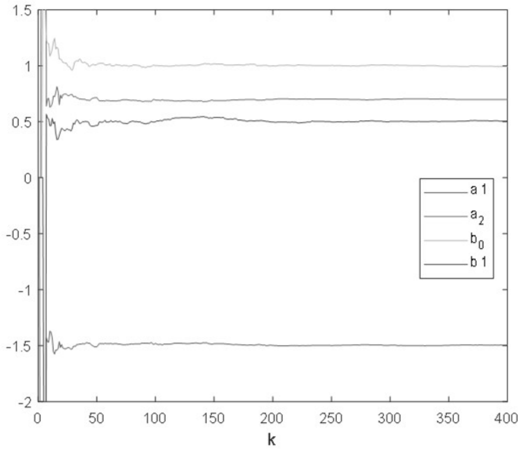
Take the initial value $P(0) = 10^6 I$, $\hat{\theta}(0) = 0$, and set $\sigma^2 = 0.1$.

Select a white noise with a variance of 1 as the input signal $\mathrm{u}(k)$.

The simulation results of parameter estimation using RLS algorithm are shown on the right.

When $k = 400$, the estimation of the parameters is

$\hat{a}_1$=-1.4946
$\hat{a}_2$ = 0.6982
$\hat{b}_0$ = 0.9924
$\hat{b}_1$ = 0.5058



17

♦ **Simulation codes**

example231.m

Self Assessment Question

Identify an unknown continuous system using the RLS method, which has the following output for a ramp input:

$$y(t) = t - 0.2 + \frac{1}{5\sqrt{0.75}} e^{-2.5t} \sin\left(5\sqrt{0.75}t + 2\arctan\left(\sqrt{3}\right)\right) + \xi(t)$$

where $\xi(t)$ is a white noise with variance of 0.2. Take the sample period t_s =0.01s.

saq231.m

18

■ Forgetting factor recursive least squares method

The recursive least squares algorithm (2.3.7-2.3.9) is more suitable for the system with constant unknown parameters.

In the general adaptive control problem, it is meaningful to consider a system with time-varying parameters.

The time-varying parameters can be divided into two cases:

① Parameter mutation but not frequent;

② The parameters change slowly.

The recursive least squares algorithm can be applied to the above two cases by simple expansion:

① For the parameter mutation problem, it can be solved by resetting the matrix P.

At this time, the matrix P in the recursive least squares method will be periodically reset to αI, where α is a sufficiently large number.

② For the problem of slowly time-varying parameters, the recursive least squares method has its limitations: with the growth of data, the so-called the "data saturation“ phenomenon, that is, with the increase of k, P(k) and K(k) become smaller and smaller.

The correction ability of θ(k) is getting weaker and weaker, which makes the newly collected input and output data have little effect to the parameter estimation values

The update function of P(k) is not significant.

So, when the system parameters change, the RLS algorithm will not be able to track this change, which makes the real-time parameter estimation fail.

To overcome this phenomenon, the following recursive least squares method with forgetting factor can be used.

In the previous sections, weighted least squares estimation was introduced

$$\hat{\theta}_k = \arg\min \sum_{i=1}^{k} \beta(k,i)[y(i) - \phi^T(i)\theta\ ]^2$$

where

$$\hat{\theta}_k = \bar{R}^{-1}(k)f(k) \qquad (2.3.10)$$

$$\bar{R}(k) = \sum_{i=1}^{k} \beta(k,i)\phi(i)\phi^T(i)$$

$$f(k) = \sum_{i=1}^{k} \beta(k,i)\phi(i)y(i)$$

Define the following weight formula

$$\beta(k,i) = \lambda^{k-i}, \qquad 0 \leq i \leq k$$

where $0 < \lambda \leq 1$ is the forgetting factor

Example: $\lambda = 0.9,\ \beta(k,k) = 1, \beta(k,k-1) = 0.9, \beta(k,k-2) = 0.81, \ldots, \beta(k,1) = 0.9^{k-1}$

It can be seen from the above formula that the closer to the current time value t the time, the greater the weight of the data.

The relationship between time k and time k-1 can be obtained from historical data

$$\bar{R}(k) = \lambda\bar{R}(k-1) + \phi(k)\phi^T(k) \qquad (2.3.11)$$

$$f(k) = \lambda f(k-1) + \phi(k)y(k) \qquad (2.3.12)$$

It can be concluded from (2.3.10), (2.3.11) and (2.3.12)

$$\begin{aligned}\hat{\theta}_k &= \bar{R}^{-1}(k)f(k) = \bar{R}^{-1}(k)[\lambda f(\mathrm{k}-1) + \phi(k)y(k)] \\ &= \bar{R}^{-1}(k)\left[\lambda \bar{R}(\mathrm{k}-1)\hat{\theta}_{\mathrm{k}-1} + \phi(k)y(k)\right] \\ &= \bar{R}^{-1}(k)\left\{[\bar{R}(k) - \phi(k)\phi^T(k)]\hat{\theta}_{\mathrm{k}-1} + \phi(k)y(k)\right\} \\ &= \hat{\theta}_{k-1} + \bar{R}^{-1}(k)\phi(k)\left[y(k) - \phi^T(k)\hat{\theta}_{\mathrm{k}-1}\right]\end{aligned}$$

Therefore, it is obtained that

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \bar{R}^{-1}(k)\phi(k)\left[y(k) - \phi^T(k)\hat{\theta}_{k-1}\right] \qquad (2.3.13)$$

$$\bar{R}(k) = \lambda\bar{R}(k-1) + \phi(k)\phi^T(k) \qquad (2.3.14)$$

It meets the requirements of an online recursive algorithm required by (2.3.1).

To avoid the trouble caused by calculating $\bar{R}^{-1}(k)$, let

$$P(k) = \bar{R}^{-1}(k)$$

The following is the theorem of a matrix inverse

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B[DA^{-1}B + C^{-1}]^{-1}DA^{-1}$$

For (2.3.14), let $A = \lambda(\bar{R}(\mathrm{k}-1)$

$$B = D^T = \phi(\mathrm{k})$$

$$C = 1$$

Then

$$P(k) = \frac{1}{\lambda}\left[P(k-1) - \frac{P(k-1)\phi(k)\phi^T(k)P(k-1)}{\lambda + \phi^T(k)P(k-1)\phi(k)}\right]$$

According to the value of $P(k)$, we have

$$\bar{R}^{-1}(k)\phi(k) = \frac{1}{\lambda}P(k-1)\phi(k) - \frac{1}{\lambda}\frac{P(k-1)\phi(k)\phi^T(k)P(k-1)\phi(k)}{\lambda + \phi^T(k)P(k-1)\phi(k)}$$

$$= \frac{P(k-1)\phi(k)}{\lambda + \phi^T(k)P(k-1)\phi(k)}$$

Let

$$K(k) = \frac{P(k-1)\phi(k)}{\lambda + \phi^T(k)P(k-1)\phi(k)}$$

25

In this way, the formula of the recursive least squares method can be obtained

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k)\left[y(k) - \phi^T(k)\hat{\theta}(k-1)\right] \qquad (2.3.15)$$

$$K(k) = \frac{P(k-1)\phi(k)}{\lambda + \phi^T(k)P(k-1)\phi(k)} \qquad (2.3.16)$$

$$P(k) = \frac{1}{\lambda}\left[P(k-1) - \frac{P(k-1)\phi(k)\phi^T(k)P(k-1)}{\lambda + \phi^T(k)P(k-1)\phi(k)}\right] \qquad (2.3.17)$$

26

♦ **Algorithm 2.3.2** (forgetting factor recursive least squares estimation, FFRLS)

Given n_a, n_b and d.

Step 1: Set initial value $\hat{\theta}(0)$ and $\mathrm{P}(0)$ and forgetting factor λ, enter initial data.

Step 2: Sample the current output $\mathrm{y}(k)$ and input $\mathrm{u}(k)$.

Step 3: Use equations (2.3.15)-(2.3.17) to calculate $\mathrm{K}(k), \hat{\theta}(k)$ and $\mathrm{P}(k)$.

Step 4: k → k+1, return to Step 2, and continue the loop.

Note: Forgetting factor λ must be selected to be a positive number between 0 and 1.

- Generally not less than 0.9. If the system is linear, $0.95 \leq \lambda \leq 1$.
- When $\lambda(k) = 1$, FFRLS algorithm degenerates to general RLS algorithm.

27

■ Example

Consider the system:

$$y(k) + a_1 y(k-1) + a_2 y(k-2) = b_0 u(k-3) + b_1 u(k-4) + \xi(k)$$

where $\xi(k)$ is a white noise with a variance of 0.1,

The time-varying parameters of $\theta(k) = [a_1, a_2, b_0, b_1]$ are

$$\theta(k) = [-1.5, 0.7, 1, 0.5]^T, \qquad k \leq 500$$

$$\theta(k) = [-1, 0.4, 1.5, 0.2]^T, \qquad k > 500$$

Take the initial values P(0)=$10^6 I$ and $\hat{\theta}$(0)=0, select white noise with a variance of 1 as the input signal $u(k)$.

28

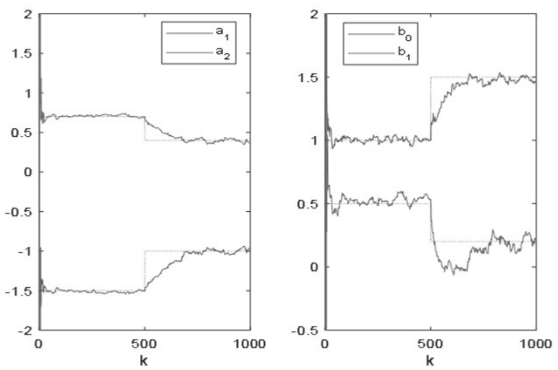
♦ **Simulation results**

Take forgetting factor λ= 0.98, using FFRLS algorithm for parameter estimation, the simulation results are shown below.

When k=500, the parameter estimation is $\hat{\theta}(500) = [-1.5083, 0.6994, 1.0049, 0.5327]^T$;

When k=1000, the parameter estimation is $\hat{\theta}(1000)) = [-0.9593, 0.3640, 1.4684, 0.2363]^T$

It can be seen that even for systems with abrupt parameter changes, the FFRLS algorithm can effectively perform parameter estimation.



♦ **Simulation codes**

example232.m

Self Assessment Question

The conditions are exactly the same as the previous example. But, the estimation model form is assumed to be

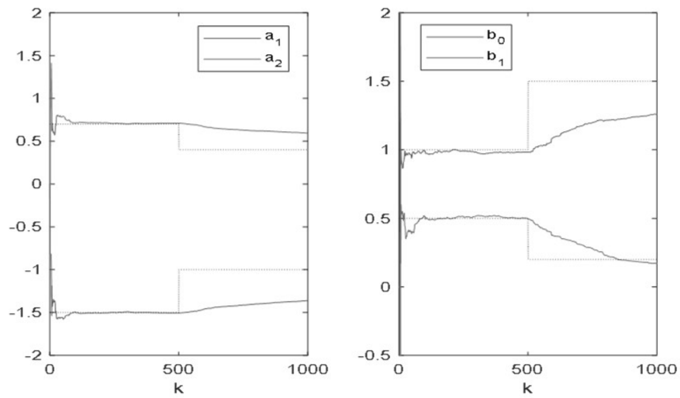
$$y(k) + \hat{a}_1 y(k-1) + \hat{a}_2 y(k-2) + \hat{a}_3 y(k-3) = \hat{b}_0 u(k-2) + \hat{b}_1 u(k-3) + \hat{b}_2 u(k-4)$$

Determine the parameters of the above model at k=500 and k=1000, respectively.

saq232.m

♦ **Ordinary Recursive Least Squares Method**

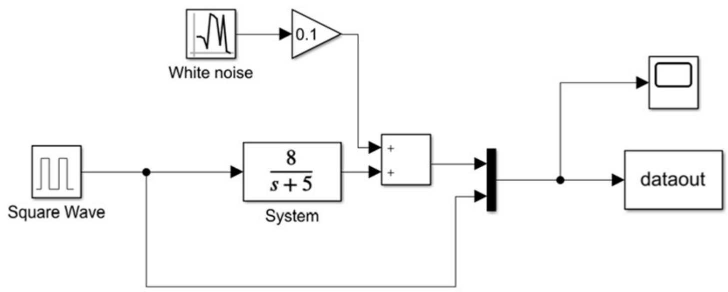
When taking the forgetting factor λ=1, FFRLS will degenerate into a common RLS algorithm, and the simulation results are



It can be seen that RLS cannot effectively track parameter changes for time-varying systems.

■ Example

Use the recursive identification method to identify the model in Simulink below.



The input and output data are stored in the array 'dataout'.

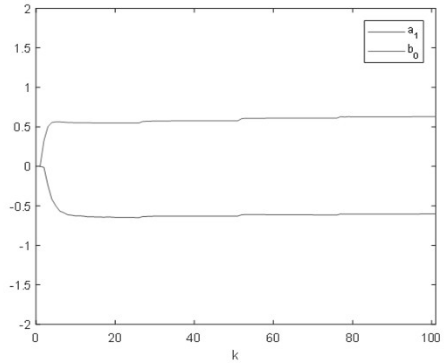
♦ **Simulation codes**

Use the recursive identification method to identify the model in Simulink.

example233s.slx

example233.m

The parameters of the discrete model (the forgetting factor λ=0.99)



The parameters of the continuous model

$$G(s) = \frac{8.006}{s + 5.051}$$

33

Self Assessment Question

Using Matlab/Simulink software and recursive least squares parameter estimation program, identify the discrete transfer function of the following second-order continuous system at the sampling period of 0.1 seconds (simulation time is 10 seconds).

$$G(s) = \frac{3s + 2}{s^2 + 5s + 4}$$

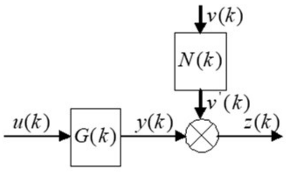
saq233s.slx

saq233.m

34

2.4 Generalised Least Squares Method

■ **Method** (RELS)



Suppose the system adopts CARMA model:

$$A(q^{-1})y(k) = B(q^{-1})u(k-d) + C(q^{-1})\xi(k) \qquad (2.4.1)$$

Compared with the three least squares methods introduced earlier, where $C(q^{-1}) \neq 1$, the noise $e(k)$ is a coloured noise, namely

$$e(k) = C(q^{-1})\xi(k) = \xi(k) + c_1\xi(k-1) + c_2\xi(k-2) + \cdots + c_{n_c}\xi(k-n_c)$$

The following will introduce the recursive extended least squares method used to estimate the parameters of the CARMA model.

The CARMA model is changed to be in the form of least squares, namely

$$y(k) = \varphi^T(k)\theta + \xi(k) \qquad (2.4.2)$$

where

$$\varphi(k) = [-y(k-1), \cdots, -y(k-n_a), u(k-d), \cdots, u(k-d-n_b), \xi(k-1), \cdots, \xi(k-n_c)]^T \in \mathcal{R}^{(n_a+n_b+1+n_c)\times 1}$$

$$\theta = \begin{bmatrix} a_1 & \cdots & a_{n_a} & b_0 & \cdots & b_{n_b} & c_1 & \cdots & c_{n_c} \end{bmatrix}^T \in \mathcal{R}^{(n_a+n_b+1+n_c)\times 1}$$

d is the time delay.

Because $\xi(k)$ in $\varphi(k)$ is immeasurable, only its estimated $\hat{\xi}(k)$ can be used, *i.e.*,

$$\hat{\xi}(k) = y(k) - \hat{y}(k) = y(k) - \hat{\varphi}^T(k)\hat{\theta} \tag{2.4.3}$$

where

$$\hat{\varphi}(k) = \left[-y(k-1), \cdots, -y(k-n_a), u(k-d), \cdots, u(k-d-n_b), \hat{\xi}(k-1), \cdots, \hat{\xi}(k-n_c)\right]^T \in \mathcal{R}^{(n_a+n_b+1+n_c)\times 1}$$

$$\hat{\theta} = \begin{bmatrix}\hat{a}_1 & \cdots & \hat{a}_{n_a} & \hat{b}_0 & \cdots & \hat{b}_{n_b} & \hat{c}_1 & \cdots & \hat{c}_{n_c}\end{bmatrix}^T \in \mathcal{R}^{(n_a+n_b+1+n_c)\times 1}$$

and $\hat{\theta}$ can be $\hat{\theta}(k)$ or $\hat{\theta}(k-1)$.

37

Replace $\varphi(k)$ by $\hat{\varphi}(k)$ and use the similar RLS method, the recursive extended least squares parameter estimation can be derived, namely

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k)\left(y(k) - \hat{\varphi}^T(k)\hat{\theta}(k-1)\right) \tag{2.4.4}$$

$$K(k) = \frac{P(k-1)\hat{\varphi}(k)}{1 + \hat{\varphi}^T(k)P(k-1)\hat{\varphi}(k)} \tag{2.4.5}$$

$$P(k) = (I - K(k)\hat{\varphi}^T(k))P(k-1) \tag{2.4.6}$$

■ **Algorithm** 2.4.1 (recursive generalised least squares estimation, RGLS)

Given n_a, n_b, n_c and d

Step 1: Set initial value $\hat{\theta}(0)$ and $P(0)$, enter the initial data.

Step 2: Sample the current output $y(k)$ and input $u(k)$.

Step 3: Construct $\hat{\varphi}(k)$ and calculate $K(k)$, $\hat{\theta}(k)$ and $P(k)$ using (2.4.4)-(2.4.6).

Step 4: Use formula (2.4.3) to calculate $\hat{\xi}(k)$.

Step 5: $k \to k+1$, return to Step 2 and continue the loop.

38

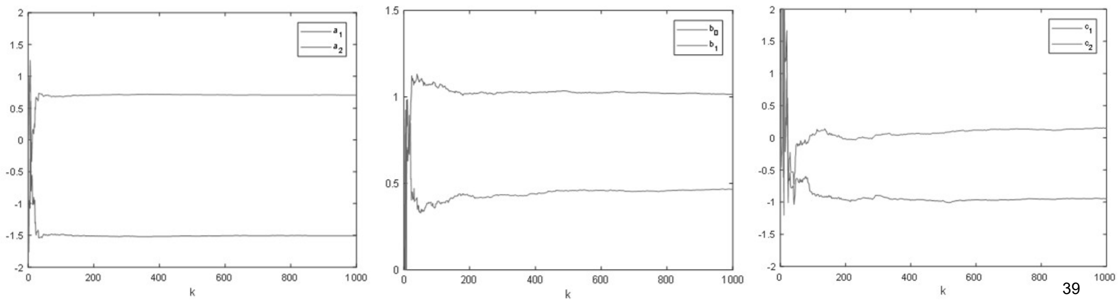
■ Simulated Example

Consider the system:

$$y(k) - 1.5y(k-1) + 0.7y(k-2) = u(k-3) + 0.5u(k-4) + \xi(k) - \xi(k-1) + 0.2\xi(k-2)$$

where, $\xi(k)$ is a white noise with a variance of 0.1.

Take the initial value $P(0) = 10^6 I, \hat{\theta}(0) = 0$.
Select white noise with variance of 1 as the input signal $u(k)$
Use RELS algorithm to perform parameter estimation.



When k=1000, the estimated parameters are

$$\hat{a}_1 = -1.5051, \quad \hat{a}_2 = 0.7045$$
$$\hat{b}_0 = 1.0153, \quad \hat{b}_1 = 0.4654$$
$$\hat{c}_1 = -0.9456, \quad \hat{c}_2 = 0.1508$$

Compared with the estimated parameters, it can be seen that the convergence of parameters $\hat{c}_1$ and $\hat{c}_2$ is relatively slow because the white noise estimation in formula (2.3.24) is not accurate.

To improve the accuracy of parameter estimation, the number of simulation steps can be appropriately increased.

♦ **Simulation codes**

example241.m

Self Assessment Question

Following the previous example, estimate the parameters of the system below:

$$y(k) + 1.8y(k-1) + 0.8y(k-2) = \xi(k) - 0.8\xi(k-1) - 0.2\xi(k-2)$$

where $\xi(k)$ is a white noise with a variance of 0.1.

saq241.m
saq241a.m

41

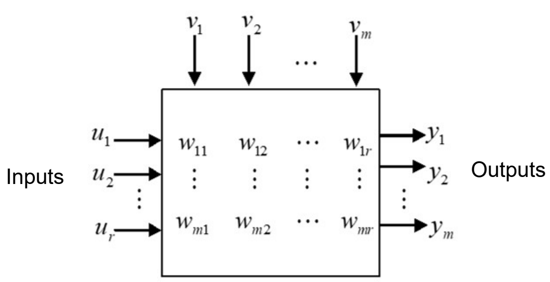
2.5 Least Squares Identification of Multivariable Systems

■ **Multi-input multi-output (MIMO) systems**

$$Y(k) + A_1Y(k-1) + \cdots + A_nY(k-n) = B_0U(k) + B_1U(k-1) + \cdots B_nU(k-n) + V(k)$$

where

$Y(k) \in \mathcal{R}^m$: the output vector
$U(k) \in \mathcal{R}^r$: the input vector
$V(k) \in \mathcal{R}^m$: the noise vector
$A_1, A_2, \ldots, A_n \in \mathcal{R}^{m\times m}$:
the system matrices to be estimated
$B_0, B_1, \ldots, B_n \in \mathcal{R}^{m\times r}$:
the system matrices to be estimated



42

$$Y(k)=\begin{bmatrix} y_1(k) \\ y_2(k) \\ \vdots \\ y_m(k) \end{bmatrix} \qquad U(k)=\begin{bmatrix} u_1(k) \\ u_2(k) \\ \vdots \\ u_r(k) \end{bmatrix} \qquad V(k)=\begin{bmatrix} v_1(k) \\ v_2(k) \\ \vdots \\ v_m(k) \end{bmatrix}$$

$$A_i=\begin{bmatrix} a_{11}^i & a_{12}^i & \cdots & a_{1m}^i \\ a_{21}^i & a_{22}^i & \cdots & a_{2m}^i \\ \vdots & \vdots & & \vdots \\ a_{m1}^i & a_{m2}^i & \cdots & a_{mm}^i \end{bmatrix} \qquad B_i=\begin{bmatrix} b_{11}^i & b_{12}^i & \cdots & b_{1r}^i \\ b_{21}^i & b_{22}^i & \cdots & b_{2r}^i \\ \vdots & \vdots & & \vdots \\ b_{m1}^i & b_{m2}^i & \cdots & b_{mr}^i \end{bmatrix}$$

$i=1,\cdots,n$ $\qquad$ $i=0,1,\cdots,n$

- **Algorithm and Design of Least Squares Identification for Multivariable System**

$$Y(k)+A_1Y(k-1)+\cdots+A_nY(k-n)=B_0U(k)+B_1U(k-1)+\cdots B_nU(k-n)+V(k)$$

$$\Downarrow$$

$$A(q^{-1})Y(k)=B(q^{-1})\,U(k)+V(k)$$

$$A(q^{-1})=I+A_1q^{-1}+\cdots+A_nq^{-n}=I+\sum_{i=1}^{n}A_iq^{-i}$$

$$B(q^{-1})=B_0+B_1q^{-1}+\cdots+B_nq^{-n}=\sum_{i=0}^{n}B_iq^{-i}$$

The number of parameters to be identified: $nm^2+(n+1)mr$

$$A(q^{-1})Y(k) = B(q^{-1})U(k) + V(k)$$

$$A_i Y(k-i) = \begin{bmatrix} a_{11}^i & a_{12}^i & \cdots & a_{1m}^i \\ a_{21}^i & a_{22}^i & \cdots & a_{2m}^i \\ \vdots & \vdots & & \vdots \\ a_{m1}^i & a_{m2}^i & \cdots & a_{mm}^i \end{bmatrix} \begin{bmatrix} y_1(k-i) \\ y_2(k-i) \\ \vdots \\ y_m(k-i) \end{bmatrix} \qquad i = 1, 2, \dots, n$$

$$B_i U(k-i) = \begin{bmatrix} b_{11}^i & b_{12}^i & \cdots & b_{1r}^i \\ b_{21}^i & b_{22}^i & \cdots & b_{2r}^i \\ \vdots & \vdots & & \vdots \\ b_{m1}^i & b_{m2}^i & \cdots & b_{mr}^i \end{bmatrix} \begin{bmatrix} u_1(k-i) \\ u_2(k-i) \\ \vdots \\ u_r(k-i) \end{bmatrix} \qquad i = 0, 1, \dots, n$$

45

♦ **Subsystems of a MIMO system**

Taking the j-th row of the following MIMO system:

$$A(z^{-1})Y(k) = B(z^{-1})U(k) + V(k)$$

leads to the j-th subsystem:

$$y_j(k) + a_{j1}^1 y_1(k-1) + \cdots + a_{jm}^1 y_m(k-1) + a_{j1}^2 y_1(k-2) + \cdots +$$

$$a_{jm}^2 y_m(k-2) + \cdots + a_{j1}^n y_1(k-n) + \cdots + a_{jm}^n y_m(k-n) =$$

$$= b_{j1}^0\, u_1(k) + b_{j2}^0 u_2(k) + \cdots + b_{jr}^0 u_r(k) + b_{j1}^1 u_1(k-1) +$$

$$b_{j2}^1 u_2(k-1) + \cdots + b_{jr}^1 u_r(k-1) + \cdots + b_{j1}^n u_1(k-n) +$$

$$b_{j2}^n u_2(k-n) + \cdots + b_{jr}^n u_r(k-n) + v_j(k)$$

46

$$
\begin{aligned}
y_j(k) = &-a_{j1}^1 y_1(k-1) - \cdots - a_{jm}^1 y_m(k-1) - a_{j1}^2 y_1(k-2) - \cdots \\
&-a_{jm}^2 y_m(k-2) - \cdots - a_{j1}^n y_1(k-n) - \cdots - a_{jm}^n y_m(k-n) + \\
&b_{j1}^0 u_1(k) + b_{j2}^0 u_2(k) + \cdots + b_{jr}^0 u_r(k) + b_{j1}^1 u_1(k-1) + \\
&b_{j2}^1 u_2(k-1) + \cdots + b_{jr}^1 u_r(k-1) + \cdots + b_{j1}^n u_1(k-n) + \\
&b_{j2}^n u_2(k-n) + \cdots + b_{jr}^n u_r(k-n) + v_j(k)
\end{aligned}
$$

For $k = 1 \sim N$, there are N equations.

The j-th subsystem can be expressed as

$$\boxed{Y_j = H\theta_j + V_j}$$

where

$$Y_j = \begin{bmatrix} y_j(1) \\ y_j(2) \\ \vdots \\ y_j(N) \end{bmatrix} \quad V_j = \begin{bmatrix} v_j(1) \\ v_j(2) \\ \vdots \\ v_j(N) \end{bmatrix} \quad Y(k-i) = \begin{bmatrix} y_1(k-i) \\ y_2(k-i) \\ \vdots \\ y_m(k-i) \end{bmatrix} \quad U(k-i) = \begin{bmatrix} u_1(k-i) \\ u_2(k-i) \\ \vdots \\ u_r(k-i) \end{bmatrix}$$

$$i = 1,2,\cdots,n \qquad i = 0,1,\cdots,n$$

$$\theta_j = \begin{bmatrix} a_{j1}^1 & \cdots & a_{jm}^1 & \cdots & a_{j1}^n & \cdots & a_{jm}^n & b_{j1}^0 & \cdots & b_{jr}^0 & \cdots & b_{j1}^n & \cdots & b_{jr}^n \end{bmatrix}^T$$

$$H = \begin{bmatrix} -Y^T(0) & \cdots & -Y^T(1-n) & U^T(1) & \cdots & U^T(1-n) \\ -Y^T(1) & \cdots & -Y^T(2-n) & U^T(2) & \cdots & U^T(2-n) \\ \vdots & & \vdots & \vdots & & \vdots \\ -Y^T(N-1) & \cdots & -Y^T(N-n) & U^T(N) & \cdots & U^T(N-n) \end{bmatrix}$$

$$Y_j = H\theta_j + V_j \quad \Longrightarrow \quad \hat{\theta}_j = (H^T H)^{-1} H^T Y_j$$

$$\theta_j = \begin{bmatrix} a_{j1}^1 & \cdots & a_{jm}^1 & \cdots & a_{j1}^n & \cdots & a_{jm}^n & b_{j1}^0 & \cdots & b_{jr}^0 & \cdots & b_{j1}^n & \cdots & b_{jr}^n \end{bmatrix}^T$$

$$\boxed{j = 1,2,\cdots,m} \Downarrow \boxed{\hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_m}$$

$$\begin{bmatrix} a_{11}^i & a_{12}^i & \cdots & a_{1m}^i \\ a_{21}^i & a_{22}^i & \cdots & a_{2m}^i \\ \vdots & \vdots & & \vdots \\ a_{m1}^i & a_{m2}^i & \cdots & a_{mm}^i \end{bmatrix} \quad i = 1,2,\cdots,n$$

$$\begin{bmatrix} b_{11}^i & b_{12}^i & \cdots & b_{1r}^i \\ b_{21}^i & b_{22}^i & \cdots & b_{2r}^i \\ \vdots & \vdots & & \vdots \\ b_{m1}^i & b_{m2}^i & \cdots & b_{mr}^i \end{bmatrix} \quad i = 0,1,\cdots,n$$

49

The recursive least squares identification algorithm for multivariable systems

$$\hat{\theta}_j(k+1) = \hat{\theta}_j(k) + K(k+1)[y_j(k+1) - h^T(k+1)\hat{\theta}_j(k)]$$

$$K(k+1) = P(k)h(k+1)[1 + h^T(k+1)P(k)h(k+1)]^{-1}$$

$$P(k+1) = P(k) - P(k)h(k+1)[1 + h^T(k+1)P(k)h(k+1)]^{-1}h(k+1)P(k)$$

where

$$h(k) = [-Y^T(k-1), \cdots, -Y^T(k-n), U^T(k), \cdots, U^T(k-n)]^T$$

50

Example: Using the least squares identification method of a multivariable system to identify the parameters of the following MIMO system

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} + A_1 \begin{bmatrix} y_1(k-1) \\ y_2(k-1) \end{bmatrix} + A_2 \begin{bmatrix} y_1(k-2) \\ y_2(k-2) \end{bmatrix} = \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} + B_1 \begin{bmatrix} u_1(k-1) \\ u_2(k-1) \end{bmatrix} + B_2 \begin{bmatrix} u_1(k-2) \\ u_2(k-2) \end{bmatrix} + \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix}$$

where, $v_1(k)$ and $v_2(k)$ are random noises with the same distribution and have $N(0{,}0.05)$, the input signal adopts random signal.

The ideal coefficients of the model are

$$A_1 = \begin{bmatrix} 0.5 & -0.2 \\ -0.3 & 0.6 \end{bmatrix} \qquad B_0 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 1.2 & -0.6 \\ 0.1 & -0.6 \end{bmatrix} \qquad B_1 = \begin{bmatrix} 0.5 & -0.4 \\ 0.2 & -0.3 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 0.4 & -0.3 \\ -0.2 & 0.1 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} 0.5 & -0.2 \\ -0.3 & 0.6 \end{bmatrix} \Longrightarrow \hat{A}_1 = \begin{bmatrix} 0.4448 & -0.0596 \\ -0.1486 & 0.6594 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 1.2 & -0.6 \\ 0.1 & -0.6 \end{bmatrix} \Longrightarrow \hat{A}_2 = \begin{bmatrix} 1.2389 & -0.4445 \\ 0.1154 & -0.4829 \end{bmatrix}$$

$$B_0 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} \Longrightarrow \hat{B}_0 = \begin{bmatrix} 1.0678 & 0.0057 \\ -0.0030 & 1.0714 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0.5 & -0.4 \\ 0.2 & -0.3 \end{bmatrix} \Longrightarrow \hat{B}_1 = \begin{bmatrix} 1.0678 & 0.0057 \\ -0.0030 & 1.0714 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 0.4 & -0.3 \\ -0.2 & 0.1 \end{bmatrix} \Longrightarrow \hat{B}_2 = \begin{bmatrix} 0.4296 & -0.4122 \\ -0.3223 & 0.1938 \end{bmatrix}$$

♦ **Simulation codes**

example251.m

Exercise 2.2

Let the difference equation of the single-input-single-output system be

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

Take

$$a_1 = 1.6,\ \ a_2 = 0.7,\ b_1 = 1.0, b_2 = 0.4, c_1 = 0.9, c_2 = 1.2,\ c_3 = 0.3$$

and the input signal adopts a 4th-order M-sequence with an amplitude of 1.

When the mean and variance of Gaussian noise $v(k)$ are 0 and 0.5, the parameters are estimated by least squares method, recursive least squares method and forgetting-factor recursive least squares method ($\lambda = 0.95$), respectively.

Through the analysis and comparison of the identification results of the three methods, explain the advantages and disadvantages of the above three parameter identification methods.