



第5章 辨识的最小二乘法

5.3 最小二乘递推算法

5.4 数据递推的饱和及解决办法

作者：李鹏波、胡德文

单位：国防科技大学机电工程与自动化学院

Email: gaoxia73@163.com

中国水利水电出版社版权所有



5.3 最小二乘递推算法

5.4 数据递推的饱和及...

2. 渐消记忆法

3. 限定记忆法

4. 振荡记忆法

Home Page

Title Page

◀ ▶

◀ ▶

Page 1 of 16

Go Back

Full Screen

Close

Quit



第5章 辨识的最小二乘法

5.3 最小二乘递推算法

5.4 数据递推的饱和及解决办法

1. 数据饱和现象

2. 渐消记忆法

3. 限定记忆法

4. 振荡记忆法

5.3 最小二乘递推算法

5.4 数据递推的饱和及...

2. 渐消记忆法

3. 限定记忆法

4. 振荡记忆法

Home Page

Title Page

◀▶

◀▶

Page 2 of 16

Go Back

Full Screen

Close

Quit



5.3 最小二乘递推算法

5.1节已经给出了最小二乘一次完成算法，但具体使用时不仅占用内存量大，而且不能用于在线辨识。一次完成算法还有如下的缺陷。

1. 数据量越多，系统参数估计的精度就越高。为了获得满意的辨识结果，矩阵 $\Phi^T \Phi$ 的阶数常常取得相当大。这样，矩阵求逆的计算量很大，存储量也很大。
2. 每增加一次观测量，都必须重新计算 $\Phi, (\Phi^T \Phi)^{-1}$ 。
3. 如果出现 Φ 列相关，即不满秩的情况， $\Phi^T \Phi$ 为病态矩阵，则不能得到最小二乘估计值。

解决这个问题的办法是把它化成递推算法。依观测次序的递推算法就是每获得一次新的观测数据就修正一次参数估计值，随着时间的推移，便能获得满意的辨识结果。递推辨识算法具有无矩阵求逆，以及跟踪时变系统等特点，这样不仅可以减少计算量和储存量，而且能实现在线辨识。

5.3 最小二乘递推算法
5.4 数据递推的饱和及...
2. 渐消记忆法
3. 限定记忆法
4. 振荡记忆法

Home Page

Title Page

◀ ▶

◀ ▶

Page 3 of 16

Go Back

Full Screen

Close

Quit

系统用线性差分方程模型来描述

$$\begin{aligned} y(k) &= \varphi^T(k)\theta + e(k) \\ \varphi(k) &= \begin{bmatrix} -y(k-1) & \cdots & -y(k-n) & u(k) & \cdots & u(k-n) \end{bmatrix}^T \end{aligned} \quad (1)$$

$$\begin{aligned} \begin{bmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(n+N) \end{bmatrix} &= \begin{bmatrix} -y(n) & \cdots & -y(1) & u(n+1) & \cdots & u(1) \\ -y(n+1) & \cdots & -y(2) & u(n+2) & \cdots & u(2) \\ \vdots & & \vdots & & & \vdots \\ -y(n+N-1) & \cdots & -y(N) & u(n+N) & \cdots & u(N) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_0 \\ \vdots \\ b_n \end{bmatrix} \\ &+ \begin{bmatrix} e(n+1) \\ e(n+2) \\ \vdots \\ e(n+N) \end{bmatrix} \end{aligned} \quad (2)$$



5.3 最小二乘递推算法

5.4 数据递推的饱和及...

2. 渐消记忆法

3. 限定记忆法

4. 振荡记忆法

Home Page

Title Page



Page 4 of 16

Go Back

Full Screen

Close

Quit

系统用线性差分方程模型来描述

$$\begin{aligned} y(k) &= \varphi^T(k)\theta + e(k) \\ \varphi(k) &= \begin{bmatrix} -y(k-1) & \cdots & -y(k-n) & u(k) & \cdots & u(k-n) \end{bmatrix}^T \end{aligned} \quad (1)$$

$$\begin{aligned} \begin{bmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(n+N) \end{bmatrix} &= \begin{bmatrix} -y(n) & \cdots & -y(1) & u(n+1) & \cdots & u(1) \\ -y(n+1) & \cdots & -y(2) & u(n+2) & \cdots & u(2) \\ \vdots & & \vdots & & & \vdots \\ -y(n+N-1) & \cdots & -y(N) & u(n+N) & \cdots & u(N) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_0 \\ \vdots \\ b_n \end{bmatrix} \\ &+ \begin{bmatrix} e(n+1) \\ e(n+2) \\ \vdots \\ e(n+N) \end{bmatrix} \end{aligned} \quad (2)$$

令

$$Y_N = \begin{bmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(n+N) \end{bmatrix}, \quad \Phi_N = \begin{bmatrix} \varphi^T(1) \\ \varphi^T(2) \\ \vdots \\ \varphi^T(N) \end{bmatrix}, \quad e_N = \begin{bmatrix} e(n+1) \\ e(n+2) \\ \vdots \\ e(n+N) \end{bmatrix}$$



5.3 最小二乘递推算法
5.4 数据递推的饱和及...
2. 渐消记忆法
3. 限定记忆法
4. 振荡记忆法

Home Page

Title Page



Page 4 of 16

Go Back

Full Screen

Close

Quit

则式(2)记为向量方程的形式

$$Y_N = \Phi_N \theta + e_N \quad (3)$$

上式的最小二乘解为

$$\hat{\theta}_N = [\Phi_N^T \Phi_N]^{-1} \Phi_N^T Y_N \quad (4)$$



- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 5 of 16

Go Back

Full Screen

Close

Quit



则式(2)记为向量方程的形式

$$Y_N = \Phi_N \theta + e_N \quad (3)$$

上式的最小二乘解为

$$\hat{\theta}_N = [\Phi_N^T \Phi_N]^{-1} \Phi_N^T Y_N \quad (4)$$

记

$$P_N = [\Phi_N^T \Phi_N]^{-1} \quad (5)$$

由估计误差的方差阵

$$\text{Cov} \hat{\theta}_{LS} = \sigma^2 (\Phi_N^T \Phi_N)^{-1} = \sigma^2 P_N \quad (6)$$

于是, P_N 为**协方差阵**, σ^2 为方程误差方差。

- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

Home Page

Title Page

◀ ▶

◀ ▶

Page 5 of 16

Go Back

Full Screen

Close

Quit



- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

则式(2)记为向量方程的形式

$$Y_N = \Phi_N \theta + e_N \quad (3)$$

上式的最小二乘解为

$$\hat{\theta}_N = [\Phi_N^T \Phi_N]^{-1} \Phi_N^T Y_N \quad (4)$$

记

$$P_N = [\Phi_N^T \Phi_N]^{-1} \quad (5)$$

由估计误差的方差阵

$$\text{Cov} \hat{\theta}_{LS} = \sigma^2 (\Phi_N^T \Phi_N)^{-1} = \sigma^2 P_N \quad (6)$$

于是, P_N 为**协方差阵**, σ^2 为方程误差方差。

如果再增加1组新的观测值 $u(n+N+1), y(n+N+1)$, 记为 u_{N+1}, y_{N+1} , 则又增加1个方程

$$Y_{N+1} = \Phi_{N+1}^T \theta + e_{N+1} \quad (7)$$

其中

$$Y_{N+1} = \begin{bmatrix} Y_N \\ y_{N+1} \end{bmatrix}, \quad \Phi_{N+1} = \begin{bmatrix} \Phi_N \\ \varphi_{N+1}^T \end{bmatrix}$$

Home Page

Title Page



Page 5 of 16

Go Back

Full Screen

Close

Quit



- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

根据式(7)得到新的参数估计值

$$\hat{\theta}_{N+1} = [\Phi_{N+1}^T \Phi_{N+1}]^{-1} \Phi_{N+1}^T Y_{N+1} \quad (8)$$

式中

$$\begin{aligned} P_{N+1} &= \left[\begin{pmatrix} \Phi_N^T & \varphi_{N+1} \end{pmatrix} \begin{pmatrix} \Phi_N \\ \varphi_{N+1}^T \end{pmatrix} \right]^{-1} \\ &= \{\Phi_N^T \Phi_N + \varphi_{N+1} \varphi_{N+1}^T\}^{-1} \\ &= \{P_N^{-1} (I + P_N \varphi_{N+1} \varphi_{N+1}^T)\}^{-1} \\ &= \{I + P_N \varphi_{N+1} \varphi_{N+1}^T\}^{-1} P_N \end{aligned} \quad (9)$$

应用矩阵求逆引理，可得 P_{N+1} 与 P_N 的递推关系式。下面先介绍矩阵求逆引理。

[Home Page](#)

[Title Page](#)

[◀◀](#) [▶▶](#)

[◀](#) [▶](#)

Page 6 of 16

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



矩阵求逆引理 设 A 为 $n \times n$ 矩阵, B 为 $n \times m$ 矩阵, C 为 $m \times n$ 矩阵, 并且 A , $A + BC$ 和 $I + CA^{-1}B$ 都是非奇异矩阵, 则有恒等式

$$[A + BC]^{-1} = A^{-1} - A^{-1}B[I + CA^{-1}B]^{-1}CA^{-1} \quad (10)$$

- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

Home Page

Title Page

◀ ▶

◀ ▶

Page 7 of 16

Go Back

Full Screen

Close

Quit



矩阵求逆引理 设 A 为 $n \times n$ 矩阵, B 为 $n \times m$ 矩阵, C 为 $m \times n$ 矩阵, 并且 A , $A + BC$ 和 $I + CA^{-1}B$ 都是非奇异矩阵, 则有恒等式

$$[A + BC]^{-1} = A^{-1} - A^{-1}B[I + CA^{-1}B]^{-1}CA^{-1} \quad (10)$$

令 $A_{M \times M}$, $B_{M \times 1} = P_N \varphi_{N+1}$, $C_{1 \times M} = \varphi_{N+1}^T$ 。则由矩阵求逆引理得到

$$\begin{aligned} P_{N+1} &= P_N - P_N \varphi_{N+1} [1 + \varphi_{N+1}^T P_N \varphi_{N+1}]^{-1} \varphi_{N+1}^T P_N \\ &= P_N - \frac{P_N \varphi_{N+1} \varphi_{N+1}^T P_N}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} \end{aligned} \quad (11)$$

从上面的推导可看出, 最小二乘估计批处理算法中, 需要求 $(2n + 1) \times (2n + 1)$ 矩阵 $P_N^{-1} + \varphi_{N+1} \varphi_{N+1}^T$ 的逆矩阵, 计算较为复杂。应用矩阵求逆引理之后, 把矩阵求逆转变为求标量 $1 + \varphi_{N+1}^T P_N \varphi_{N+1}$ 的倒数, 大幅度地减少了计算量。同时, 又得到了 P_{N+1} 与 P_N 之间的递推关系式。

- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

[Home Page](#)[Title Page](#)[Page 7 of 16](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)



由式(8)和式(11)得到

$$\begin{aligned}\hat{\theta}_{N+1} &= P_{N+1} \begin{bmatrix} \Phi_N^T & \varphi_{N+1} \end{bmatrix} \begin{bmatrix} Y_N \\ y_{N+1} \end{bmatrix} \\ &= P_{N+1} [\Phi_N^T Y_N + \varphi_{N+1} y_{N+1}] \\ &= P_N \Phi_N^T Y_N - \frac{P_N \varphi_{N+1} \varphi_{N+1}^T P_N \Phi_N^T Y_N}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} \\ &\quad + P_N \varphi_{N+1} y_{N+1} - \frac{P_N \varphi_{N+1} \varphi_{N+1}^T P_N \varphi_{N+1} y_{N+1}}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} \\ &= \hat{\theta}_N + \frac{P_N \varphi_{N+1}}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} [y_{N+1} - \varphi_{N+1}^T \hat{\theta}_N]\end{aligned}\tag{12}$$

式中 $\frac{P_N \varphi_{N+1}}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}}$ 为增益矩阵，记为 G_{N+1} ，而 $y_{N+1} - \varphi_{N+1}^T \hat{\theta}_N$ 为预报误差。

- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

Home Page

Title Page

◀ ▶

◀ ▶

Page 8 of 16

Go Back

Full Screen

Close

Quit



由式(8)和式(11)得到

$$\begin{aligned}\hat{\theta}_{N+1} &= P_{N+1} \begin{bmatrix} \Phi_N^T & \varphi_{N+1} \end{bmatrix} \begin{bmatrix} Y_N \\ y_{N+1} \end{bmatrix} \\ &= P_{N+1} [\Phi_N^T Y_N + \varphi_{N+1} y_{N+1}] \\ &= P_N \Phi_N^T Y_N - \frac{P_N \varphi_{N+1} \varphi_{N+1}^T P_N \Phi_N^T Y_N}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} \\ &\quad + P_N \varphi_{N+1} y_{N+1} - \frac{P_N \varphi_{N+1} \varphi_{N+1}^T P_N \varphi_{N+1} y_{N+1}}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} \\ &= \hat{\theta}_N + \frac{P_N \varphi_{N+1}}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} [y_{N+1} - \varphi_{N+1}^T \hat{\theta}_N]\end{aligned}\quad (12)$$

式中 $\frac{P_N \varphi_{N+1}}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}}$ 为增益矩阵，记为 G_{N+1} ，而 $y_{N+1} - \varphi_{N+1}^T \hat{\theta}_N$ 为预报误差。

综合上述推导过程，得到最小二乘估计递推算法(RLS)如下

$$\hat{\theta}_{N+1} = \hat{\theta}_N + G_{N+1} [y_{N+1} - \varphi_{N+1}^T \hat{\theta}_N] \quad (13)$$

$$G_{N+1} = \frac{P_N \varphi_{N+1}}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} \quad (14)$$

$$P_{N+1} = P_N - G_{N+1} \varphi_{N+1}^T P_N \quad (15)$$

- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

Home Page

Title Page

◀ ▶

◀ ▶

Page 8 of 16

Go Back

Full Screen

Close

Quit

递推过程如下:

得到 $\hat{\theta}_N, P_N, u_{N+1}, y_{N+1}$

计算 $\varphi_{N+1} \longrightarrow \hat{\theta}_{N+1} \longrightarrow P_{N+1} \longrightarrow$ 下一步递推



5.3 最小二乘递推算法

5.4 数据递推的饱和及...

2. 渐消记忆法

3. 限定记忆法

4. 振荡记忆法

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 9 of 16

Go Back

Full Screen

Close

Quit



- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

递推过程如下:

得到 $\hat{\theta}_N, P_N, u_{N+1}, y_{N+1}$

计算 $\varphi_{N+1} \longrightarrow \hat{\theta}_{N+1} \longrightarrow P_{N+1} \longrightarrow$ 下一步递推

对于初值 $\hat{\theta}_0, P_0$ 的选取:

方法1: m组数据, 用LS成批算法, 得 $\hat{\theta}_m, P_m$, 再从 $m+1$ 开始递推。

$$\hat{\theta}_m = [\Phi_m^T \Phi_m]^{-1} \Phi_m^T Y_m \quad (16)$$

$$P_m = [\Phi_m^T \Phi_m]^{-1} \quad (17)$$

方法2: $\hat{\theta}_0 = 0$, 任取 $P_0 = \sigma^2 I, \sigma$ 特别大。

Home Page

Title Page

◀ ▶

◀ ▶

Page 9 of 16

Go Back

Full Screen

Close

Quit



- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

递推过程如下:

得到 $\hat{\theta}_N, P_N, u_{N+1}, y_{N+1}$

计算 $\varphi_{N+1} \longrightarrow \hat{\theta}_{N+1} \longrightarrow P_{N+1} \longrightarrow$ 下一步递推

对于初值 $\hat{\theta}_0, P_0$ 的选取:

方法1: m组数据, 用LS成批算法, 得 $\hat{\theta}_m, P_m$, 再从 $m+1$ 开始递推。

$$\hat{\theta}_m = [\Phi_m^T \Phi_m]^{-1} \Phi_m^T Y_m \quad (16)$$

$$P_m = [\Phi_m^T \Phi_m]^{-1} \quad (17)$$

方法2: $\hat{\theta}_0 = 0$, 任取 $P_0 = \sigma^2 I$, σ 特别大。

另外, 可用下式作为递推算法的**停机准则**

$$\max_i \left| \frac{\hat{\theta}_{N+1}(i) - \hat{\theta}_N(i)}{\hat{\theta}_{N+1}(i)} \right| < \varepsilon \quad (18)$$

式中 $\hat{\theta}_{N+1}(i)$ 为参数向量 θ 的第 i 个元素在 $N+1$ 次递推计算结果。 ε 为给定的表示精度要求的某一正数。

Home Page

Title Page

◀ ▶

◀ ▶

Page 9 of 16

Go Back

Full Screen

Close

Quit

5.4 数据递推的饱和及解决办法

1. 数据饱和现象

所谓数据饱和现象，就是随着时间的推移，采集的数据越来越多，新数据提供的信息被旧数据所淹没。如果辨识算法对新、旧数据给予相同的信度，那么随着从新数据中获得的信息量相对下降，算法就会慢慢失去修正能力。这时参数估计值可能还偏离真值较远就无法更新了；对时变过程来说，它又将导致参数估计值不能跟踪时变参数的变化。



- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

Home Page

Title Page

◀▶

◀▶

Page 10 of 16

Go Back

Full Screen

Close

Quit

5.4 数据递推的饱和及解决办法

1. 数据饱和现象

所谓数据饱和现象，就是随着时间的推移，采集的数据越来越多，新数据提供的信息被旧数据所淹没。如果辨识算法对新、旧数据给予相同的信度，那么随着从新数据中获得的信息量相对下降，算法就会慢慢失去修正能力。这时参数估计值可能还偏离真值较远就无法更新了；对时变过程来说，它又将导致参数估计值不能跟踪时变参数的变化。

这是因为 P_0 正定，而式(9)中， $\varphi_{N+1}\varphi_{N+1}^T$ 是非负定的，所以 $P_i, i = 1, \dots, n$ 都是正定的。又因为

$$P_N - P_{N+1} = P_N \frac{\varphi_{N+1}\varphi_{N+1}^T}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} P_N \geq 0 \quad (19)$$

所以

$$P_N \geq P_{N+1} \quad (20)$$

由此可知， P_N 是递减的正定阵。随着递推次数的增加，这会导致 $P_N \xrightarrow{N \rightarrow \infty} 0$ 。所以增益阵 G_{N+1} 也将随着 N 的增加而逐渐趋于零向量，从而使RLS算法失去修正能力。



- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

[Home Page](#)[Title Page](#)[◀◀](#) [▶▶](#)[◀](#) [▶](#)[Page 10 of 16](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

5.4 数据递推的饱和及解决办法

1. 数据饱和现象

所谓数据饱和现象，就是随着时间的推移，采集的数据越来越多，新数据提供的信息被旧数据所淹没。如果辨识算法对新、旧数据给予相同的信度，那么随着从新数据中获得的信息量相对下降，算法就会慢慢失去修正能力。这时参数估计值可能还偏离真值较远就无法更新了；对时变过程来说，它又将导致参数估计值不能跟踪时变参数的变化。

这是因为 P_0 正定，而式(9)中， $\varphi_{N+1}\varphi_{N+1}^T$ 是非负定的，所以 $P_i, i = 1, \dots, n$ 都是正定的。又因为

$$P_N - P_{N+1} = P_N \frac{\varphi_{N+1}\varphi_{N+1}^T}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} P_N \geq 0 \quad (19)$$

所以

$$P_N \geq P_{N+1} \quad (20)$$

由此可知， P_N 是递减的正定阵。随着递推次数的增加，这会导致 $P_N \xrightarrow{N \rightarrow \infty} 0$ 。所以增益阵 G_{N+1} 也将随着 N 的增加而逐渐趋于零向量，从而使RLS算法失去修正能力。

另外，由于递推在有穷字长的计算机上实现时，每步都存在舍入误差。因此数据饱和后，由于这些原因致使新的采样值不仅对参数估计不起改进作用，反而还可能使所计算的 P_N 失去正定性，甚至失去对称性，造成参数的估计量与真实参数之间的偏差越来越大。



- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

[Home Page](#)[Title Page](#)[◀◀](#) [▶▶](#)[◀](#) [▶](#)[Page 10 of 16](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)



- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

2. 渐消记忆法

渐消记忆法又称为遗忘因子法，这种方法的思想是对旧数据加上遗忘因子，按指数加权来使得旧数据的作用衰减。

最小二乘估计值为

$$\hat{\theta}_N = [\Phi_N^T \Phi_N]^{-1} \Phi_N^T Y_N \quad (21)$$

指标函数

$$J_N = \sum_{k=1}^N e^2(k) = e_N^T e_N \quad (22)$$

新增观测 y_{N+1} 之后的数据阵

$$\Phi_{N+1} = \begin{bmatrix} \Phi_N \\ \varphi_{N+1}^T \end{bmatrix}, \quad Y_{N+1} = \begin{bmatrix} Y_N \\ y_{N+1} \end{bmatrix}$$

[Home Page](#)[Title Page](#)[◀◀](#) [▶▶](#)[◀](#) [▶](#)[Page 11 of 16](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)



- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

2. 渐消记忆法

渐消记忆法又称为遗忘因子法，这种方法的思想是对旧数据加上遗忘因子，按指数加权来使得旧数据的作用衰减。

最小二乘估计值为

$$\hat{\theta}_N = [\Phi_N^T \Phi_N]^{-1} \Phi_N^T Y_N \quad (21)$$

指标函数

$$J_N = \sum_{k=1}^N e^2(k) = e_N^T e_N \quad (22)$$

新增观测 y_{N+1} 之后的数据阵

$$\Phi_{N+1} = \begin{bmatrix} \Phi_N \\ \varphi_{N+1}^T \end{bmatrix}, \quad Y_{N+1} = \begin{bmatrix} Y_N \\ y_{N+1} \end{bmatrix}$$

则加衰减因子 $\lambda(0 < \lambda \leq 1)$ 后的数据阵为

$$\Phi_{N+1} = \begin{bmatrix} \lambda \Phi_N \\ \varphi_{N+1}^T \end{bmatrix}, \quad Y_{N+1} = \begin{bmatrix} \lambda Y_N \\ y_{N+1} \end{bmatrix}$$

[Home Page](#)[Title Page](#)[◀◀](#) [▶▶](#)[◀](#) [▶](#)[Page 11 of 16](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)



- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

得到如下递推公式

$$P_{N+1} = [\Phi_{N+1}^T \Phi_{N+1}]^{-1} = [\lambda^2 P_N^{-1} + \varphi_{N+1} \varphi_{N+1}^T]^{-1} \quad (23)$$

$$\begin{aligned} \hat{\theta}_{N+1} &= P_{N+1} \cdot [\lambda^2 P_N^{-1} \hat{\theta}_N - \varphi_{N+1} y_{N+1}] \\ &= \hat{\theta}_N + G_{N+1} [y_{N+1} - \varphi_{N+1}^T \hat{\theta}_N] \end{aligned} \quad (24)$$

$$G_{N+1} = \frac{P_N \varphi_{N+1}}{\lambda^2 + \varphi_{N+1}^T P_N \varphi_{N+1}} \quad (25)$$

$$P_{N+1} = \frac{1}{\lambda^2} \left[I - \frac{P_N \varphi_{N+1} \varphi_{N+1}^T}{\lambda^2 + \varphi_{N+1}^T P_N \varphi_{N+1}} \right] P_N \quad (26)$$

$$J_{N+1} = e_{N+1}^T e_{N+1} = \lambda^2 J_N + e^2(N+1) \quad (27)$$

[Home Page](#)[Title Page](#)[◀◀](#) [▶▶](#)[◀](#) [▶](#)[Page 12 of 16](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)



- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

令 $\rho = \lambda^2$, $0 < \rho \leq 1$, ρ 称为遗忘因子。综合以上分析, 渐消记忆法的递推算法(RFF)可归纳为

$$\hat{\theta}_{N+1} = \hat{\theta}_N + G_{N+1}[y_{N+1} - \varphi_{N+1}^T \hat{\theta}_N] \quad (28)$$

$$G_{N+1} = \frac{P_N \varphi_{N+1}}{\rho + \varphi_{N+1}^T P_N \varphi_{N+1}} \quad (29)$$

$$P_{N+1} = \frac{1}{\rho} [I - G_{N+1} \varphi_{N+1}^T] P_N \quad (30)$$

RFF算法的结构和计算流程和RLS算法基本是一致的。初始状态 $\hat{\theta}_0$ 和 P_0 也可采用上一节的方法。但是遗忘因子必须选择接近于1的正数, 通常不小于0.9。如果系统是线性的, 应选择 $0.95 \leq \rho \leq 1$ 。当 $\rho = 1$ 时, 称为无限增长记忆。 ρ 越小, 则遗忘速度越快。

[Home Page](#)[Title Page](#)

Page 13 of 16

[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)



3. 限定记忆法

限定记忆法每次估值只依据最新的 N 个数据，在此以前的数据则全部剔除。如考虑一个固定长度的矩形窗，每一时刻一个新数据点增加进来，一个旧数据点剔除出去，这样保持了每次都只取最新的 N 个数据进行计算。递推算法分为两个过程。

- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

Home Page

Title Page

◀▶

◀▶

Page 14 of 16

Go Back

Full Screen

Close

Quit



- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

3. 限定记忆法

限定记忆法每次估值只依据最新的 N 个数据，在此以前的数据则全部剔除。如考虑一个固定长度的矩形窗，每一时刻一个新数据点增加进来，一个旧数据点剔除出去，这样保持了每次都只取最新的 N 个数据进行计算。递推算法分为两个过程。

(1) 先进一个观测数据 y_{i+N} ，即在 $i + N$ 时刻，进一个数据 y_{i+N} 。递推公式为

$$\hat{\theta}_{i+N,i} = \hat{\theta}_{i+N-1,i} + G_{i+N,i} [y_{i+N} - \varphi_{i+N}^T \hat{\theta}_{i+N-1,i}] \quad (31)$$

$$G_{i+N,i} = P_{i+N-1,i} \frac{\varphi_{i+N}}{1 + \varphi_{i+N}^T P_{i+N-1,i} \varphi_{i+N}} \quad (32)$$

$$P_{i+N,i} = \left[I - P_{i+N-1,i} \frac{\varphi_{i+N}^T \varphi_{i+N}}{1 + \varphi_{i+N}^T P_{i+N-1,i} \varphi_{i+N}} \right] P_{i+N-1,i} \quad (33)$$

式中， $\hat{\theta}_{i+N,i}$ 为 i 到 $i + N$ 时距中 $N + 1$ 个数据计算得到的参数估值。

[Home Page](#)[Title Page](#)[Page 14 of 16](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)



- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

(2) 再出一个数据 y_i ，即剔除一个旧的观测数据。递推公式为

$$\hat{\theta}_{i+N,i+1} = \hat{\theta}_{i+N,i} - G_{i+N,i+1}[y_i - \varphi_i^T \hat{\theta}_{i+N,i}] \quad (34)$$

$$G_{i+N,i+1} = P_{i+N,i} \frac{\varphi_i}{1 - \varphi_i^T P_{i+N,i} \varphi_i} \quad (35)$$

$$P_{i+N,i+1} = \left[I + P_{i+N,i} \frac{\varphi_i^T \varphi_i}{1 - \varphi_i^T P_{i+N,i} \varphi_i} \right] P_{i+N,i} \quad (36)$$

式中 $\hat{\theta}_{i+N,i+1}$ 为 $i+1$ 到 $i+N$ 时距中 N 个数据对 θ 的估值。

[Home Page](#)[Title Page](#)[◀◀](#) [▶▶](#)[◀](#) [▶](#)[Page 15 of 16](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)



4. 振荡记忆法

振荡记忆法采用整段剔除 N 组数据的方法，即当数据长度已达 $2N$ 时，可剔除开始的 N 个数据。这样使数据在 N 到 $2N - 1$ 个之间变化。

- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

Home Page

Title Page



Page 16 of 16

Go Back

Full Screen

Close

Quit



4. 振荡记忆法

振荡记忆法采用整段剔除 N 组数据的方法，即当数据长度已达 $2N$ 时，可剔除开始的 N 个数据。这样使数据在 N 到 $2N - 1$ 个之间变化。

(1) 数据由 $N + 1$ 至 $2N + 1$ ，每增加1个数据，用上节递推算法中式(31)至式(33)进行递推计算。

- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

Home Page

Title Page

◀ ▶

◀ ▶

Page 16 of 16

Go Back

Full Screen

Close

Quit



4. 振荡记忆法

振荡记忆法采用整段剔除 N 组数据的方法，即当数据长度已达 $2N$ 时，可剔除开始的 N 个数据。这样使数据在 N 到 $2N - 1$ 个之间变化。

(1) 数据由 $N + 1$ 至 $2N + 1$ ，每增加1个数据，用上节递推算法中式(31)至式(33)进行递推计算。

(2) 数据增加至 $2N$ ，去掉前整段 N 个数据。

$$\hat{\theta}_{i+2N,i+N+1} = P_{i+2N,i+N+1} [P_{i+2N,i+1}^{-1} \hat{\theta}_{i+2N,i+1} - P_{i+N,i+1}^{-1} \hat{\theta}_{i+N,i+1}] \quad (37)$$

$$P_{i+2N,i+N+1}^{-1} = P_{i+2N,i+1}^{-1} - P_{i+N,i+1}^{-1} \quad (38)$$

- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

Home Page

Title Page



Page 16 of 16

Go Back

Full Screen

Close

Quit



- 5.3 最小二乘递推算法
- 5.4 数据递推的饱和及...
- 2. 渐消记忆法
- 3. 限定记忆法
- 4. 振荡记忆法

Home Page

Title Page



Page 16 of 16

Go Back

Full Screen

Close

Quit

4. 振荡记忆法

振荡记忆法采用整段剔除 N 组数据的方法，即当数据长度已达 $2N$ 时，可剔除开始的 N 个数据。这样使数据在 N 到 $2N - 1$ 个之间变化。

(1) 数据由 $N + 1$ 至 $2N + 1$ ，每增加1个数据，用上节递推算法中式(31)至式(33)进行递推计算。

(2) 数据增加至 $2N$ ，去掉前整段 N 个数据。

$$\hat{\theta}_{i+2N,i+N+1} = P_{i+2N,i+N+1} [P_{i+2N,i+1}^{-1} \hat{\theta}_{i+2N,i+1} - P_{i+N,i+1}^{-1} \hat{\theta}_{i+N,i+1}] \quad (37)$$

$$P_{i+2N,i+N+1}^{-1} = P_{i+2N,i+1}^{-1} - P_{i+N,i+1}^{-1} \quad (38)$$

估计值 $\hat{\theta}_{i+2N,i+N+1}$ 和协方差 $P_{i+2N,i+N+1}$ 需存储，而 $\hat{\theta}_{i+N,i+1}$ 与 $P_{i+N,i+1}$ 是上周期算出的，事先需保存。