

文章编号: 1001-0920(2009)01-0086-05

## LTI 状态空间模型的参数估计

甘敏, 彭辉, 王勇

(中南大学信息科学与工程学院, 长沙 410083)

**摘要:** 采用 3 种方法研究了 LTI(Linear time-invariant) 状态空间模型中未知参数的估计问题: 利用 Metropolis-Hastings 算法, 从后验分布中抽取一定容量的样本, 得出其均值和标准差; 采用进化算法来最小化对数似然函数, 得到全局最优解; 采用模拟退火算法来最大化似然函数, 得到全局最优解. 最后, 通过数值实验验证和比较了 3 种估计算法的有效性.

**关键词:** 参数估计; Metropolis-Hastings 算法; 进化算法; 模拟退火

**中图分类号:** TP13 **文献标识码:** A

### Parameter estimation for LTI state space model

GAN Min, PENG Hui, WANG Yong

(College of Information Science and Engineering, Central South University, Changsha 410083, China.

Correspondent: GAN Min, E-mail: ganmin@mail.csu.edu.cn)

**Abstract:** The problem of estimating unknown parameters of linear time-invariant state space model is studied by three approaches: We use Metropolis-Hastings algorithm to draw a sample of some size from the posterior distribution based on Bayesian inference, and then the posterior mean and standard deviation are obtained. A computational model of a population-based algorithm generator is proposed to minimize the log-likelihood function. We apply the simulated annealing algorithm to maximize the likelihood function. Finally, the numerical experiments and comparison results show the effectiveness of the proposed algorithms.

**Key words:** Parameter estimation; Metropolis-Hastings algorithm; Evolutionary algorithm; Simulated annealing

### 1 引言

线性时不变动态系统(也称线性高斯状态空间模型)因其易分析性而广泛应用于控制、经济、生物等领域的建模和预测. 对于从其观测数据估计参数化的模型, 一个最基本也是应用最广泛的方法是最大化观测数据的似然度, 也就是最大似然法. 由于这种方法已被研究了近一个世纪, 它有着很成熟的理论支持<sup>[1]</sup>. 尽管如此, 在实际中应用最大似然法却没有那么简单, 这在很大程度上是因为它是一个非凸最优化问题, 由于没有闭合形式解, 一般采用基于牛顿法的梯度搜索策略. 这种方法的成功与否取决于目标函数对模型中参数的曲率, 这样求得的解多为局部最优解.

最近, 针对以上问题, Gibson 和 Ninness 研究了 LTI(Linear time-invariant) 状态空间系统模型的估计问题, 对于完全参数化状态空间模型(模型中所

有矩阵中的元素都未知且没有限制), 他们提出了一种鲁棒的极大似然估计法, 其思想是利用 EM (Expectation Maximization) 算法估计极大似然度<sup>[2]</sup>.

国内对参数估计也有不少研究<sup>[3,4]</sup>, 但针对 LTI 模型的却很鲜见. 本文研究的也是 LTI 状态空间模型的估计问题, 但模型并非完全参数化, 而是部分参数未知. 这样的情况在各个领域也是比较常见的, 但不能直接应用 Gibson 和 Ninness 所提出的方法.

对于参数估计有两种统计推断的方法: 一是上文提到的基于最大似然原理的经典方法; 二是贝叶斯推断, 它将参数的先验信息与观测数据相结合, 得到后验分布, 然后基于后验分布进行统计推断. 历史上, 在两类统计推断之间有着激烈的争论, 但两种方法都被证明是有用的, 而且都被广泛接受. 本文在这两种统计推断框架下, 采用 3 种方法研究 LTI 状态

收稿日期: 2007-11-15; 修回日期: 2008-06-06.

基金项目: 国家自然科学基金项目(60574058).

作者简介: 甘敏(1982—), 男, 武汉人, 博士生, 从事金融时间序列分析、进化计算的研究; 彭辉(1963—), 男, 长沙人, 教授, 博士生导师, 从事复杂非线性系统的统计建模与辨识、非线性模型参数优化等研究.

空间模型的参数估计问题:1) 基于贝叶斯推断,利用 Metropolis-Hastings 算法从后验分布中抽取一定容量的样本,然后得出其均值和标准差;2) 基于最大似然原理,采用进化算法来最小化对数似然函数,这种方法不需要利用梯度的信息,避免了陷入局部最优解;3) 基于最大似然原理,采用模拟退火算法来最大化似然函数,这种算法也可以得到全局最优解.最后,用一个具体的例子验证和比较了 3 种方法的可行性和有效性,并与其他方法作了比较.

## 2 LTI 状态空间模型

LTI 状态空间模型,也称线性高斯状态空间模型,可由以下两个方程来表示:

$$\begin{aligned} X_{k+1} &= AX_k + w_k, \quad w_k \sim N(0, Q), \\ Y_k &= CX_k + v_k, \quad v_k \sim N(0, R). \end{aligned} \quad (1)$$

其中:  $X_k$  是  $m$  维的状态变量;  $Y_k$  是  $n$  维的观测向量;  $A$  是状态转移矩阵 ( $m \times m$  矩阵);  $C$  是观测矩阵 ( $n \times m$  矩阵);  $w_k, v_k$  分别是系统噪声和观测噪声,它们是相互独立的高斯白噪声;协方差分别为  $Q, R$ . 模型(1)中的第 1 个方程是状态转移方程,它表达了系统从前一状态到当前状态的演化过程;第 2 个方程是观测方程,它表达了状态变量到观测值的映射关系.初始状态向量  $X_0$  为均值是  $\mu_0$ ,方差是  $P_0$  的高斯随机向量,即  $X_0 \sim N(\mu_0, P_0)$ .

当  $A, C, Q, R$  矩阵中的所有元素未知且没有限制时, Ghahramani, Hinton<sup>[5]</sup> 和 Gibson, Ninness<sup>[2]</sup> 已提出用 EM 算法来估计模型. 本文所研究的问题是  $A, C, Q, R$  矩阵中的某些参数未知,对于这种情况则无法直接应用他们提出的算法,因而需用不同的方法来估计模型中的参数.

有时系统的输入是可以被观测到的,在这种情况下可以建立一个输入输出模型. 记输入为  $U_k$ , 则模型可表示为

$$\begin{aligned} X_{k+1} &= AX_k + BU_k + w_k, \quad w_k \sim N(0, Q), \\ Y_k &= CX_k + DU_k + v_k, \quad v_k \sim N(0, R), \end{aligned} \quad (2)$$

其中  $B$  和  $D$  是与输入相关的矩阵. 本文给出的算法是针对只有输出的情况,但极容易将其扩展到输入输出的情况.

## 3 基于贝叶斯推断的参数估计

### 3.1 后验分布

令  $\theta$  为模型中未知参数组成的向量,  $Y$  是观测数据. 贝叶斯分析寻求将关于参数的知识与数据相结合来作出推断. 参数的知识是通过参数预先指定一个先验分布来表示的,比如可服从正态分布、平均分布、伽玛分布等. 这样可计算出各参数的概率密度,将它们相乘即可得到  $\theta$  的先验分布,记为  $f(\theta)$ . 对于一个给定的模型,用  $f(Y|\theta)$  表示数据的似然

函数,则由条件概率的定义,有

$$f(\theta|Y) = \frac{f(\theta, Y)}{f(Y)} = \frac{f(\theta)f(Y|\theta)}{f(Y)}. \quad (3)$$

这里边际分布可通过下式得到:

$$f(Y) = \int f(\theta, Y) d\theta = \int f(Y|\theta) f(\theta) d\theta. \quad (4)$$

方程(3)中的分布  $f(\theta|Y)$  称为  $\theta$  的后验分布,一般可利用贝叶斯准则得到,即

$$f(\theta|Y) = \frac{f(\theta)f(Y|\theta)}{\int f(\theta)f(Y|\theta) d\theta}. \quad (5)$$

### 3.2 卡尔曼滤波和似然函数

由上述可知,如果要基于方程(5)作统计推断,需要计算似然函数  $f(Y|\theta)$ , 这里将用卡尔曼滤波解决此问题. 为表达清晰、方便起见,记  $Y^N = \{Y_1, \dots, Y_N\}$  为模型(1)的观测数据,  $\hat{\theta}_k$  为其估计残差,  $\Sigma_k$  为估计残差的协方差矩阵. 则  $\hat{\theta}_k$  的条件概率密度为

$$\begin{aligned} f(\hat{\theta}_k | Y_{k-1}, \dots, Y_1, \theta) &= \\ (2\pi)^{-\frac{m}{2}} / |\Sigma_k|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \hat{\theta}_k^T \Sigma_k^{-1} \hat{\theta}_k\right), \end{aligned} \quad (6)$$

其中  $|\cdot|$  表示行列式. 于是可以推出似然函数为

$$\begin{aligned} f(Y^N | \theta) &= \\ \prod_{k=1}^N f(Y_k | Y_{k-1}, \dots, Y_1, \theta) &= \\ \prod_{k=1}^N f(\hat{\theta}_k | Y_{k-1}, \dots, Y_1, \theta) &= \\ (2\pi)^{-\frac{mN}{2}} / |\Sigma_k|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \sum_{k=1}^N \hat{\theta}_k^T \Sigma_k^{-1} \hat{\theta}_k\right). \end{aligned} \quad (7)$$

式(7)中的  $\hat{\theta}_k$  和  $\Sigma_k$  可以通过卡尔曼滤波<sup>[6]</sup>得到. 卡尔曼滤波算法如下:

用  $X_{k|k-1}, S_k^-$  表示  $X_k$  的先验估计条件均值和条件方差;  $X_{k|k}, S_k$  表示  $X_k$  的后验估计条件均值和条件方差;  $K_k$  表示卡尔曼增益. 则:

预测为

$$\begin{cases} X_{k|k-1} = E\{X_k | Y^{k-1}\} = AX_{k-1|k-1}, \\ \hat{\theta}_k = Y_k - CX_{k|k-1}, \\ S_k^- = E\{(X_k - X_{k|k-1})(X_k - X_{k|k-1})^T\} = \\ AS_{k-1}A^T + Q; \end{cases} \quad (8)$$

滤波为

$$\begin{cases} K_k = E\{\hat{\theta}_k \hat{\theta}_k^T\} = CS_k^- C^T + R, \\ K_k = S_k^- C^T \Sigma_k^{-1}, \\ X_{k|k} = E\{X_k | Y^k\} = X_{k|k-1} + K_k \hat{\theta}_k, \\ S_k = E\{(X_k - X_{k|k})(X_k - X_{k|k})^T\} = \\ [I - K_k C] S_k^-. \end{cases} \quad (9)$$

### 3.3 Metropolis-Hastings 算法

假定希望从分布  $f(\theta|Y)$  中抽取一个随机样本,而直接抽取要么太浪费时间,要么不可行,但总

存在一个近似分布,可以很容易得到随机抽取. Metropolis-Hastings 算法就是从近似分布中产生一系列的随机抽取,其分布函数收敛到  $f(\cdot / Y)$ . 该算法按如下步骤进行:

- 1) 初始化  $i_0$ .
- 2) For  $t = 1$  to  $M$

从均匀分布  $U(0,1)$  中取一随机数  $u$ .

从已知取样密度中抽取一个候选样本  $i^*$ , 用  $q(i^* / i_{t-1})$  表示已知取样密度, 即  $i^* \sim q(i^* / i_{t-1})$ .

$$\text{If } u < \frac{f(i^* / Y) q(i_{t-1} / i^*)}{f(i_{t-1} / Y) q(i^* / i_{t-1})},$$

$$i_t = i^*,$$

else

$$i_t = i_{t-1}.$$

在一些正规性条件下, 序列  $\{i_t\}$  依分布收敛到  $f(\cdot / Y)^{[7]}$ . 常见的取样密度的具体例子有: 1) 独立链, 即  $q(i_t / i_{t-1}) = q(i_t)$ ,  $q(i_t)$  不依赖于  $i_{t-1}$ ; 2) 随机游走链, 即  $q(i_t / i_{t-1}) = q(i_t - i_{t-1})$ .

### 3.4 基于贝叶斯推断的参数估计算法

总结以上内容, 基于贝叶斯推断的参数估计的具体算法如下:

Step1: 令  $t = 0$ , 初始化  $i_0$ , 计算  $f(i_0)$ , 用式(7) 计算  $f(Y^N / i_0)$ , 令  $t = t + 1$ .

Step2: 从取样分布  $q(i_t / i_{t-1})$  中抽取一个  $i^*$ .

Step3: 计算  $f(i^*)$ , 用式(7) 计算  $f(Y^N / i^*)$ .

Step4: 从均匀分布  $U(0,1)$  中抽取一个随机数  $u$ , 如果

$$u < \frac{f(Y^N / i^*) (i_{t-1}) q(i_{t-1} / i^*)}{f(Y^N / i_{t-1}) (i^*) q(i^* / i_{t-1})},$$

则  $i_t = i^*$ ; 否则  $i_t = i_{t-1}$ . 如果  $t < M$ , 则令  $t = t + 1$ , 转 Step2, 否则, 终止程序.

应当注意, 此算法执行的成功与否, 选取好取样密度以获取合适的接受率是极其重要的. 接受率等于 Metropolis-Hastings 算法中序列  $\{i_t\}$  中样本值改变的次数除以样本容量. 如果接受率很低, 则序列没有遍历到后验分布的尾部; 如果接受率很高, 则序列没有在高概率区间停留足够长的时间以达到稳定. 对于接受率应取多大值没有统一的规定, Geweke 和 Tanizaki<sup>[8]</sup> 指出, 接受率在 30% ~ 50% 时效果可能达到最优.

### 4 基于最大似然法的参数估计

对于最大似然法, 传统的梯度搜索策略得到的大多为局部最优解. 为避免陷入局部最优解, 本文采用先设定参数的范围, 然后用进化算法来最小化对数似然函数以得到全局最优解. 若不锁定参数的范

围, 也用最大似然法来估计参数, 则要遍历整个参数空间. 对此, 提出采用模拟退火算法来产生一个马尔可夫链以寻求全局最优解.

#### 4.1 对数似然函数

对式(7) 取对数, 不失一般性, 这里取  $(-\log)$  似然函数以将问题转换为搜索极小值问题, 得到

$$(-1) \log f(Y^N / \Phi) =$$

$$-\log \prod_{k=1}^N f(Y_k / Y_{k-1}, \dots, Y_1, \Phi) =$$

$$-\sum_{k=1}^N \log f(\Phi_k / Y_{k-1}, \dots, Y_1, \Phi) =$$

$$(0.5) \sum_{k=1}^N \{ \log | \Sigma_k / \Phi_k^T \Phi_k^{-1} \Phi_k | + \frac{nN}{2} \log 2 \} .$$

(10)

#### 4.2 进化算法

这里的目的是最小化式(10), 以得到模型中的未知参数. 这是一个非凸优化问题, 而且一般有许多个峰度. 对于此问题, 传统的方法是基于梯度的搜索策略, 所得到的大多是局部最优解. 为得到全局最优解, 本文提出了基于进化计算的搜索策略.

在实数编码遗传算法中, 除了杂交算子外, 群体进化模型对优化性能起着重要的作用. 最近, Deb<sup>[9]</sup> 提出了一种基于群的算法发生器. 他将搜索分为 4 个独立的策略: 1) 选择策略; 2) 产生后代策略; 3) 替换策略; 4) 更新策略. 受 Deb 算法发生器的启发, 文献[10] 设计了一种具体的计算模型来求解约束优化问题. 在此基础上, 本文提出一种类似的计算模型——实数编码的进化计算模型, 以求解无约束优化问题. 具体步骤如下:

Step1: 初始化种群  $P$ , 令遗传代数为  $gen = 0$ .

Step2: 从第  $gen$  代群体中随机抽取  $\mu$  个父体 (集合  $Q$ ),  $P = P - Q$ .

Step3: 用单形杂交算子从  $\mu$  个父体中产生  $\mu$  个后代个体.

Step4: 从  $\mu$  个后代个体中找出适应值最好的个体  $Q$ .

Step5: 设  $Q$  中有  $m$  个个体的适应值比  $P$  的差.

1) 如果  $m = 0$ , 则没有替换;

2) 如果  $m = 1$ , 则用  $Q$  中相应的个体;

3) 如果  $m > 1$ , 则用  $Q$  中最差的个体.

Step6:  $P = P - Q$ ,  $gen = gen + 1$ . 如果  $gen < M$ , 则转 Step2; 否则, 终止程序.

此模型具有协调算法的探索和开发的能力. 在早期搜索中, 从群中随机选取一些个体, 它们的方差比较大, 因此, 从它们中产生的个体可以探索更广的

区域. 到了后期, 搜索群体方差变小, 这就保证了搜索集中在临近最优点区域, 然后收敛于最优点. 此外, 从后代群体中选取最好的个体并动态地替换父代中最差的个体, 这可保证算法在早期对全局信息和在晚期对局部信息的开发能力.

### 4.3 模拟退火算法

这里的目的是最大化式(7), 即  $f(Y^N / \cdot)$ . 对于这样一个似然度, 可以用 Metropolis-Hastings 算法生成一个固定分布  $f(Y^N / \cdot)$  的马尔可夫链来估计全局最优解, 即

$$\hat{\theta} = \arg \max_{i=1, \dots, n} f(Y^N / i). \quad (11)$$

但是这种方法非常低效, 因为随机取样很难落到最优区域附近. 一种高效的策略是采用模拟退火. 这种技术可生成一个非同质的马尔可夫链, 即在第  $i$  次迭代不再是固定分布等于  $f(Y^N / \cdot)$ , 而是  $f_i(Y^N / \cdot) = f^{1/T_i}(Y^N / \cdot)$ , 其中  $T_i$  是递减的冷却策略, 且  $\lim_i T_i = 0$ . 在弱正规性条件假定下,  $f(Y^N / \cdot)$  是一个集中在  $f(Y^N / \cdot)$  的全局极大值区域的概率密度. 模拟退火算法的具体步骤如下:

Step1: 令  $t = 0$ , 初始化  $i_t, T_0 = 1$ , 用式(7) 计算  $f(Y^N / i_t)$ , 令  $t = t + 1$ .

Step2: 从取样分布  $q(i_{t+1} / i_t)$  中抽取一个  $i_{t+1}^*$ .

Step3: 用式(7) 计算  $f(Y^N / i_{t+1}^*)$ .

Step4: 从均匀分布  $U(0, 1)$  中抽取一个随机数  $r_t$ , 如果

$$r_t \leq \frac{f(Y^N / i_{t+1}^*)^{1/T_{t+1}} q(i_t / i_{t+1}^*)}{f(Y^N / i_t)^{1/T_t} q(i_{t+1}^* / i_t)},$$

则  $i_t = i_{t+1}^*$ ; 否则  $i_t = i_{t-1}$ . 如果  $t < M$ , 则令  $t = t + 1$ , 根据选择的冷却策略设置  $T_{t+1}$ , 转 Step2; 否则, 终止程序.

### 5 数值实验及分析

为验证本文所提出算法的有效性, 现举例说明. 假设在模型(1) 中

$$A = \begin{bmatrix} a & 0.15 \\ 0.45 & b \end{bmatrix}, \quad C = \begin{bmatrix} 1 & c \\ d & 1 \end{bmatrix},$$

$$Q = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{3} \end{bmatrix}, \quad R = \begin{bmatrix} 0.04^2 & 0 \\ 0 & 0.04^2 \end{bmatrix}.$$

其中  $a, b, c, d, 1, 2$  是未知常量;  $X_0 \sim N(10, 0.1^2)$ . 则模型中的参数向量  $\theta = [a, b, c, d, 1, 2]$ , 其真实值为  $[0.85, 0.55, 0.35, 0.75, 0.5, 0.5]$ . 在此真实值和初始状态下随机产生 500 组观测数据, 并将其作为实际的观测值.

对于贝叶斯推断法, 参数的先验分布由表 1 给出, 将参数全部设定为均匀分布是为了便于同其他

两种方法进行比较. 每一次实验都抽取 10000 个样本值, 选取随机游走链的取样分布并使接受率在 40% 左右. 对于进化算法, 所有未知参数的范围都是  $[0, 1]$ . 对于模拟退火算法, 初始化参数设为  $[0, 0, 0, 0, 0, 0]$ .

表 1 参数 的先验分布

参数	分 布	超参数
$a$	Uniform	0, 1
$b$	Uniform	0, 1
$c$	Uniform	0, 1
$d$	Uniform	0, 1
1	Uniform	0, 1
2	Uniform	0, 1

文献[11] 中, Peng 等提出了在最大似然法框架下基于 Nelder-Mead 方法的参数估计方法. 这种方法的缺点是结果的质量依赖于初始参数的选取. 用各种方法估计未知参数的实验结果由表 2 给出. 表 2 中的结果是每种算法做 10 次实验的均值和标准差(括号内为标准差). 因为 Nelder-Mead 是确定性搜索方法, 所以其标准差均为 0.

表 2 各种方法的估计结果

参 数	真实值	贝叶斯推断	进化算法	模拟退火	Nelder-Mead
$a$	0.85	0.825 (0.010)	0.832 (0.35E-3)	0.831 (0.0022)	0.801 (0)
$b$	0.55	0.600 (0.021)	0.588 (0.88E-3)	0.591 (0.0055)	0.653 (0)
$c$	0.35	0.366 (0.015)	0.355 (0.45E-3)	0.357 (0.0033)	0.398 (0)
$d$	0.75	0.698 (0.011)	0.703 (0.88E-3)	0.701 (0.0039)	0.663 (0)
1	0.50	0.494 (0.004)	0.492 (0.17E-3)	0.491 (0.0013)	0.478 (0)
2	0.50	0.491 (0.008)	0.485 (0.37E-3)	0.486 (0.0014)	0.507 (0)

由表 2 可以看到, 贝叶斯推断、进化算法和模拟退火 3 种方法都得到了较好的结果(只有参数  $b$  和  $d$  的误差为 0.05, 其他参数都在 0.02 左右), 且明显好于 Nelder-Mead 方法得到的结果. 比较本文所论述的 3 种方法估计结果的均值可以看到, 进化算法和模拟退火的结果非常相似, 且稍好于贝叶斯推断. 再观察 3 种方法较小的标准差, 说明了算法具有较强的鲁棒性. 其中, 进化算法得到的结果的标准差要比另外两种方法小得多, 而且进化算法不需要选取取样分布函数. 因此综合来看, 进化算法的性能最高.

表 3 给出了各种方法运行的平均计算时间. 可

以看到, Nelder-Mead 的时间最短, 进化算法次之, 而贝叶斯推断和模拟退火所用的时间最长.

表3 各种方法的平均运行时间

方法	贝叶斯推断	进化算法	模拟退火	Nelder-Mead
平均时间	581	455	585	172

由于 LTI 状态空间模型是一个随机动态系统, 上面得到的结果和真实值之间还是稍有差距. 如果从观测值中能得到更多的信息, 即观测误差更小, 则可得更精确的结果. 比如, 观测噪声的协方差矩阵变为  $R = [0.01^2 \quad 0; 0 \quad 0.01^2]$ , 其他参数不变, 则估计的结果如表4所示.

表4 在观测噪声协方差更小情况下的估计结果

参数	真实值	贝叶斯推断	进化算法	模拟退火
$a$	0.85	0.843 (0.0052)	0.846 (0.35E-4)	0.846 (0.0011)
$b$	0.55	0.543 (0.0052)	0.539 (0.98E-4)	0.539 (0.0028)
$c$	0.35	0.342 (0.0093)	0.339 (0.63E-4)	0.339 (0.0018)
$d$	0.75	0.742 (0.0087)	0.746 (0.64E-4)	0.746 (0.0021)
$1$	0.50	0.484 (0.0030)	0.482 (0.49E-4)	0.481 (0.0010)
$2$	0.50	0.506 (0.0057)	0.503 (0.50E-4)	0.503 (0.0018)

## 6 结 论

本文用3种方法研究了LTI状态空间模型的参数估计问题. 实验结果表明, 3种方法都能有效地估计出模型中的未知参数, 而且进化算法得到的结果要优于贝叶斯推断和模拟退火. 另外, 进化算法不需选取取样分布, 因而在实际应用中更为方便.

## 参考文献(References)

- [1] Ljung L. 系统辨识——使用者的理论[M]. 第2版. 北京: 清华大学出版社, 2002.  
(Ljung L. System Identification — Theory for the user [M]. 2nd ed. Beijing: Tsinghua University Publication, 2002.)

- [2] Gibson S, Ninness B. Robust maximum likelihood parameter estimation of multivariable dynamic systems [J]. Automatica, 2005, 41(10): 1667-1682.
- [3] 王凌, 李彬彬, 郑大钟, 等. 模型降阶和参数估计的一种快速遗传算法[J]. 控制与决策, 2005, 20(4): 426-433.  
(Wang L, Li B B, Zheng D Z, et al. A fast GA for model reduction and parameter estimation [J]. Control and Decision, 2005, 20(4): 426-433.)
- [4] 童勤业, 陈琢, 李光. 系统参数估计的混沌方法[J]. 控制与决策, 2003, 18(4): 427-431.  
(Tong Q Y, Chen Z, Li G. Parameter estimation of systems based on chaotic theory [J]. Control and Decision, 2003, 18(4): 427-431.)
- [5] Ghahramani Z, Hinton G E. Parameter estimation for linear dynamical systems [R]. Toronto: University of Toronto, 1996.
- [6] Haykin S. Kalman filtering and neural networks [C]. Adaptive and Learning Systems for Signal Processing, Communications and Control. New York: Wiley, 2001.
- [7] Gelman A E, Carlin J B, Stern H S, et al. Bayesian data analysis [M]. London: CRC Press, 1995.
- [8] Geweke J, Tanizaki H. Note on sampling distribution for the metropolis-hastings algorithm [J]. Communications in Statistics, 2003, 32(4): 775-789.
- [9] Deb K. A population-based algorithm generator for real-parameter optimization [J]. Soft Computing — A Fusion of Foundations, Methodologies and Applications, 2005, 9(4): 236-253.
- [10] Cai Z, Wang Y. A multiobjective optimization-based evolutionary algorithm for constrained optimization [J]. IEEE Trans on Evolutionary Computation, 2006, 10(6): 658-675.
- [11] Peng H, Ozaki T, Haggan-Ozaki V. Modeling and asset allocation for financial markets based on discrete time microstructure model [J]. The European Physical Journal B, 2003, 31(2): 285-293.