

文章编号: 1001-0920(2008)10-1182-04

## 一种动态数据流的实时趋势分析算法

周 黔, 吴铁军

(浙江大学 工业控制技术国家重点实验室, 杭州 310027)

**摘 要:** 有效的趋势提取可提供监控对象早期预警、评估监控对象状态和决策支持信息. 基于增量递推最小二乘回归参数估计和广义似然比变化点检测算法, 提出一种动态数据流趋势分析算法. 其计算实时性和分析精度均比现有算法有明显的改进. 仿真实验结果验证了该算法的有效性.

**关键词:** 趋势分析; 动态数据挖掘; 增量递推最小二乘; 广义似然比检验

**中图分类号:** TP311      **文献标识码:** A

### Real-time algorithm for trend analysis of dynamic data streams

ZHOU Qian, WU Tie-jun

(State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China.

Correspondent: ZHOU Qian, E-mail: qzhou@ipc.zju.edu.cn)

**Abstract:** Efficient trend extraction methods can provide early warnings, severity assessments of monitored subjects and information for decision support. In this paper, a real-time trend extraction algorithm for dynamic data streams algorithm is proposed by combining an incremental recursive least square algorithm for regression parameter estimation with the generalized likelihood ratio test for change-point detection. Remarkably faster computational speed and higher trend analysis accuracy are achieved by this algorithm compared with the existing algorithms in the same field. The experiment results show the effectiveness of the proposed algorithm.

**Key words:** Trend analysis; Dynamic data mining; Incremental recursive least square; Generalized likelihood ratio test

## 1 引 言

动态数据流是指由大量连续到达、潜在无限长、快速变化的数据组成的有序时间序列. 动态数据流趋势分析的目的在于提取趋势变化信息, 为监控对象提供早期预警、状态评估和决策支持<sup>[1]</sup>. 由于动态数据流自身的特点, 要求数据流分析算法计算效率高, 满足实时性或近似实时性, 并且计算过程资源消耗少<sup>[2,3]</sup>.

在现有常用的趋势提取算法中<sup>[4-9]</sup>, 滑动窗口算法<sup>[4-6]</sup>和外推式在线数据分割算法<sup>[7]</sup>最接近于满足上述动态数据流趋势分析的要求. 但前一种算法需对每一到达的数据流元素使用线性拟合算法建模, 计算量大, 实时性差; 后一种算法只对给定长度的数据序列回归建模, 因此趋势提取误差较大.

本文提出一种基于增量递推最小二乘法的动态数据流趋势分析算法, 采用增量计算快速检测数据流分割点并确定分段, 因而实时性较强. 同时根据新

到数据流元素在线调整趋势参数, 采用广义似然比检测分割点, 因而趋势分析的精度明显提高.

## 2 问题描述

动态数据流实时趋势分析的基础是按某一统计特性指标实时地分割数据流, 使所分割的数据段内数据服从同一统计模型, 而相邻分段服从不同的统计模型. 记不断到达的一维实值数据流为

$$X = \{x(t_1), \dots, x(t_i), \dots, x(t_c), \dots\}, \quad (1)$$

其中  $t_c$  为当前时刻. 数据流分割将  $X$  分割成一系列连续的非空数据段  $\{X_1, \dots, X_j, \dots, X_s, \dots\}$ , 其中第  $j$  数据段  $X_j = \{x(t_{j,1}), \dots, x(t_{j,l}), \dots, x(t_{j,n_j})\}$ , 对应的数据到达时间  $t_{j,l} \in \{t_1, \dots, t_i, \dots, t_c, \dots\}$ ,  $j = 1, 2, \dots, s$ ,  $l = 1, 2, \dots, n_j$  ( $n_j$  为数据段  $X_j$  的长度),  $t_{1,1} = t_1$ . 记  $X_s$  为包含当前数据  $x(t_c)$  的数据段, 称为当前数据段.

设  $X_j$  中的数据可用线性回归模型描述, 即

收稿日期: 2007-06-30; 修回日期: 2007-09-26.

基金项目: 国家 863 计划项目 (2002AA412010).

作者简介: 周黔 (1970—), 男, 重庆人, 博士生, 从事模式识别、数据挖掘的研究; 吴铁军 (1950—), 男, 南京人, 教授, 博士生导师, 从事复杂系统智能控制与决策等研究.

$x(t) = f(t, j) + j(t), t \in \{t_{j,1}, \dots, t_{j,n_j}\}$ . (2)  
 其中:  $f(t, j) = a_j t + b_j$  为数据段  $X_j$  的线性回归模型,  $j = [a_j, b_j]^T$  为模型参数向量;  $j(t)$  为独立同分布零均值白噪声. 参数  $a_j$  称为  $X_j$  的趋势特征值, 数据段  $X_{j+1}$  的第 1 个数据元素  $x(t_{j+1,1})$  为  $X_j$  的分割点.

数据流实时趋势分析的基本任务是: 在当前已接收数据序列  $X_{s,n} = \{x(t_{s,1}), \dots, x(t_{s,n})\}$  的基础上, 对新到达的数据流元素  $x(t_c)$  完成以下计算: 1) 分割点检测; 2) 趋势特征值计算. 数据流的分割原则上是以前趋势特征值的不同为依据, 而趋势特征值计算又建立在数据流正确分割的基础上, 因此这两个任务具有相互依存性.

滑动窗口 (SW) 算法是一种连续建模分析数据流分割算法<sup>[4-6]</sup>, 即在当前已建立回归模型的数据段基础上, 用新到达的每一数据扩充当前数据段, 并重新建立新的回归模型. 若该模型的拟合均方差大于预先给定的分割点阈值, 则认为新到达的数据为当前数据段的分割点, 该数据段的趋势特征值由已有回归模型的参数给出, 并将后续到达的数据归入新的当前数据段, 启动新的趋势分析过程; 若上述分割点检测判据不成立, 则继续分析下一个到达的数据. 这种趋势分析算法虽能连续给出数据流趋势估计值, 但对每个到达的数据均需进行一次完整的回归建模和数据段分割点判定, 且随着当前数据段内数据的增多, 回归建模计算时间加长, 因此计算量加大.

Sylvie 等提出一次建模、连续分析的外推式在线数据分割 (OSD) 算法<sup>[7]</sup>. 该算法待到当前数据序列达到一定长度时, 才对其建立回归模型; 此后对于新到达的数据, 只将其代入已建立的模型, 分析外推累积误差. 若大于事先给定的阈值, 则认为新到达的数据为当前数据段的分割点, 并从已建立的模型中获得该数据段的趋势特征值; 若上述分割点检测判据不成立, 则继续分析下一个到达的数据.

OSD 算法比 SW 算法节约了大量计算时间, 但对回归建模后新到达的数据, 不能像 SW 算法那样动态地更新回归模型, 因而趋势分析的精度较低.

### 3 增量式数据流实时趋势分析方法

针对现有趋势分析算法的缺陷, 本文提出一种新的数据流实时趋势分析算法, 用以克服 SW 算法计算复杂度高和 OSD 算法分析精度低的弊端.

#### 3.1 递推回归建模

为了实时辨识时变系统, Zhou 等提出一种增量递推最小二乘 (IRLS) 算法<sup>[10]</sup>. 该算法可利用新到达的数据修正原模型参数, 从而得到新的模型参数.

设当前数据序列  $X_{s,n} = \{x(t_{s,1}), \dots, x(t_{s,n})\}$  构造的线性回归模型为  $f(t, s, n)$ , 以向量形式将数据序列表示为  $X_{s,n} = [x(t_{s,1}), \dots, x(t_{s,n})]^T$ , 则有  $X_{s,n} = U_{s,n} p_{s,n} + q_{s,n}$ . 其中

$$U_{s,n} = \begin{bmatrix} t_{s,1} & \dots & t_{s,n} \\ 1 & \dots & 1 \end{bmatrix}^T, \text{rank}(U_{s,n}) = 2 < n;$$

$q_{s,n}$  为期望为 0 的随机误差向量. 以最小二乘法估计参数  $p_{s,n}$ , 即使  $q_{s,n}$  满足

$$\min p_{s,n} \|q_{s,n}\|^2 = \|X_{s,n} - U_{s,n} p_{s,n}\|^2. \quad (3)$$

令  $p_{s,n} = U_{s,n}^T U_{s,n}^{-1} X_{s,n}$ ,  $q_{s,n} = U_{s,n}^T X_{s,n}$ , 对  $p_{s,n}$  求导并令其导数为 0, 可求得  $p_{s,n} = p_{s,n}^{-1} q_{s,n}$ . 当数据流元素  $x(t_c)$  到达后, 将其记为  $x(t_{s,n+1})$ , 并将当前数据序列扩充为  $X_{s,n+1}$ , 则有

$$p_{s,n+1} = U_{s,n+1}^T U_{s,n+1}^{-1} X_{s,n+1} = U_{s,n}^T U_{s,n}^{-1} X_{s,n} + U_{n+1}^T U_{n+1}^{-1} X_{n+1} = p_{s,n} + U_{n+1}^T U_{n+1}^{-1} X_{n+1}, \quad (4)$$

其中  $U_{n+1} = [t_{s,n+1} \ 1]$ . 与式 (4) 类似, 对  $q_{s,n}$  有

$$q_{s,n+1} = U_{s,n+1}^T X_{s,n+1} = q_{s,n} + U_{n+1}^T x(t_{s,n+1}). \quad (5)$$

此时回归模型的参数向量满足递推方程

$$p_{s,n+1}^{-1} = p_{s,n+1}^{-1} q_{s,n+1} = p_{s,n}^{-1} + p_{s,n+1}^{-1} U_{n+1}^T (x(t_{s,n+1}) - U_{n+1} p_{s,n}). \quad (6)$$

为避免式 (6) 中求逆运算, 引入矩阵求逆引理<sup>[11]</sup>

$$[A + BCD]^{-1} = A^{-1} - A^{-1} B [DA^{-1} B + C^{-1}]^{-1} DA^{-1}. \quad (7)$$

令  $A = p_{s,n}^{-1}$ ,  $B^T = D = U_{n+1}$ ,  $C = 1$ ,  $g_{s,n} = p_{s,n}^{-1}$ , 可得

$$g_{s,n+1}^{-1} = p_{s,n+1}^{-1} = g_{s,n}^{-1} - (1 + U_{n+1} g_{s,n} U_{n+1}^T)^{-1} g_{s,n} U_{n+1}^T U_{n+1} g_{s,n}. \quad (8)$$

其中初始值  $g_{s,0} = I$ , 为一正数,  $I$  为  $2 \times 2$  单位矩阵. 当数据序列噪声较大时, 取较小值; 反之, 取较大值. 式 (8) 代入式 (6), 可得回归模型参数的递推计算式

$$p_{s,n+1} = p_{s,n} + p_{s,n} g_{s,n+1}^{-1} q_{s,n+1}, \quad (9)$$

其中

$$g_{s,n} = g_{s,n}^{-1} U_{n+1}^T (1 + U_{n+1} g_{s,n} U_{n+1}^T)^{-1} (x(t_{s,n+1}) - U_{n+1} p_{s,n}). \quad (10)$$

注意式中  $(1 + U_{n+1} g_{s,n} U_{n+1}^T)$  为标量, 因此避免了矩阵求逆运算. 式 (10) 的初始值  $g_{s,0}$  一般取为 0 向量.

#### 3.2 广义似然比分割点检测

Hawkins 提出一种基于动态规划的广义似然比检验 (GLRT), 用于对变化点的检测<sup>[12]</sup>. 相对于 SW 算法和 OSD 算法中的分隔点检测方法, GLRT 方法能克服阈值完全根据经验选择, 在某些环境下分割效果较差等问题. 但这种算法计算代价较大, 而且需要预先给定数据序列分段数<sup>[13]</sup>.



本文在 GLRT 方法的基础上,给出一种增量递推数据流分割点检测算法,以满足数据流实时趋势分析的需要. 设当前长度为  $n$  的数据序列为  $X_{s,n}$ , 当新数据流元素  $x(t_c) = x(t_{s,n+1})$  到达后,对于扩充数据序列  $X_{s,n+1}$ ,递推建立回归模型  $X_{s,n+1} = U_{s,n+1} \mu_{s,n+1} + \varepsilon_{s,n+1}$ , 其中残差  $\varepsilon_{s,n+1} = [\varepsilon_{s,n+1}(t_{s,1}), \dots, \varepsilon_{s,n+1}(t_{s,n+1})]^T$  的每个元素均服从正态分布  $N(0, \sigma^2)$ .

用 GLRT 方法检测由  $\varepsilon_{s,i} (i = 1, 2, \dots, n+1)$  构成的残差序列在时刻  $t_{s,n+1}$  是否发生统计特性的变化,即令广义似然比统计量<sup>[12]</sup>

$$F = (n-1) [F(1, n+1) - F(1, n)] / F(1, n). \quad (11)$$

其中  $F(1, n) = [1 + \log(2 \hat{\sigma}_{s,n}^2)]n/2$  为残差序列  $\{\varepsilon_{s,1}, \dots, \varepsilon_{s,n}\}$  的似然函数,而

$$\hat{\sigma}_{s,n}^2 = \frac{1}{n} \sum_{i=1}^n \varepsilon_{s,i}^2 = \frac{\mu_{s,n}}{n}. \quad (12)$$

在式(11)中,  $F = F(n+1, n)$ , 为显著水平. 给定阈值  $\alpha > 0$ , 当  $F > \alpha$  时,  $x(t_{s,n+1})$  即为分割点. 当希望提取数据流长期趋势特征时,可取较大的  $\alpha$  值; 当希望提取数据流短期趋势特征时,可取较小的  $\alpha$  值. 当给定显著水平  $\alpha = 0.1$  时,  $F(n+1, n)$  的值在 1.9.16 之间,因此  $\alpha$  一般取 1.9 之间的值.

为了高效计算  $F$ , 本文提出一种增量递推算法. 根据式(3), 当新的数据流元素  $x(t_c) = x(t_{s,n+1})$  到达后,由于

$$X_{s,n+1}^T X_{s,n+1} = X_{s,n}^T X_{s,n} + x^2(t_{s,n+1}), \quad (13)$$

则有

$$\begin{aligned} \mu_{s,n+1} &= (X_{s,n+1} - U_{s,n+1} \mu_{s,n+1})^2 = \\ &= (X_{s,n} - U_{s,n} \mu_{s,n})^T (X_{s,n} - U_{s,n} \mu_{s,n}) + \\ &+ (x(t_{s,n+1}) - U_{n+1} \mu_{s,n+1})^T (x(t_{s,n+1}) - U_{n+1} \mu_{s,n+1}), \end{aligned} \quad (14)$$

其中

$$\begin{aligned} (X_{s,n} - U_{s,n} \mu_{s,n})^T (X_{s,n} - U_{s,n} \mu_{s,n}) &= \\ \mu_{s,n} - (X_{s,n} - U_{s,n} \mu_{s,n})^T U_{s,n} \mu_{s,n} - \\ (U_{s,n} \mu_{s,n})^T (X_{s,n} - U_{s,n} \mu_{s,n}) - U_{s,n}^T U_{s,n} \mu_{s,n}. \end{aligned} \quad (15)$$

因为  $q_{s,n} = p_{s,n} \mu_{s,n}$ , 所以

$$\begin{aligned} (X_{1,n} - U_{1,n} \mu_{1,n})^T U_{1,n} \mu_{1,n} &= \\ (q_{1,n} - p_{1,n} \mu_{1,n})^T \mu_{1,n} &= 0. \end{aligned} \quad (16)$$

类似地, 有  $(U_{s,n} \mu_{s,n})^T (X_{s,n} - U_{s,n} \mu_{s,n}) = 0$ . 又因  $p_{s,n} = U_{s,n}^T U_{s,n}$ , 故

$$U_{s,n}^T U_{s,n} \mu_{s,n} = U_{s,n}^T p_{s,n} \mu_{s,n}. \quad (17)$$

因此式(14)可改写成

$$\mu_{s,n+1} = \mu_{s,n} + \mu_{s,n}, \quad (18)$$

其中

$$\begin{aligned} \mu_{s,n} &= - \frac{1}{U_{n+1} \mu_{s,n+1}}^T (x(t_{s,n+1}) - U_{n+1} \mu_{s,n+1}) \\ &= U_{n+1} \mu_{s,n+1}^T (x(t_{s,n+1}) - U_{n+1} \mu_{s,n+1}). \end{aligned} \quad (19)$$

即均方差  $\mu_{s,n+1}$  可根据式(18)增量递推得到, 其初始值  $\mu_{s,0}$  一般取为 0.

### 3.3 算法流程及计算复杂性分析

增量递推回归建模和广义似然比检验方法的引入, 构成了本文的趋势分析算法. 算法的基本流程如下:

1) 从当前数据段  $X_s$  的起始点  $x(t_{s,1})$  开始, 用每一新到达的数据流元素  $x(t_c)$  扩充  $X_s$ .

2) 在回归模型参数  $\mu_{s,n} (n = 1, 2, \dots)$  的基础上, 采用式(8) ~ (10) 的递推方程快速获得新的回归模型参数  $\mu_{s,n+1}$ .

3) 在模型均方差  $\mu_{s,n}$  的基础上, 根据式(18)和(19)计算  $\mu_{s,n+1}$ , 再以式(11)计算  $F$ .

4) 当  $F > \alpha$  时,  $x(t_c)$  为  $X_{s,n}$  的分割点, 令  $X_s = X_{s,n}$ , 并以  $\mu_{s,n}$  中的元素  $\mu_{s,n}$  为  $X_s$  的趋势特征值, 更新当前数据段; 以  $x(t_c)$  为新的当前数据段的起始点, 转 1).

5) 当  $F \leq \alpha$  时,  $x(t_c)$  不是  $X_{s,n}$  的分割点, 更新参数  $\mu_{s,n}$  和  $\mu_{s,n}$ , 转 1).

本文算法对各点计算时间与序列长度  $n$  无关, 算法的计算时间复杂度为  $O(n)$ . SW 算法的计算时间复杂度为  $O(n^2)$ , 这意味着当一个数据段内的序列越来越长时, SW 算法对新到数据流元素的计算时间会越来越长, 最后导致算法崩溃. OSD 算法虽然只用一定长的子序列估计线性回归参数, 但需要外推检测分割点, 因此与本文算法的计算时间复杂度  $O(n)$  相同.

数据流是根据趋势特征值的不同而分割的, 因此趋势特征值估计误差将传递给分割点检测, 而分割位置将进一步影响后续趋势特征值的计算. 为减小这种传递误差对趋势分析精度的影响, 而数据流分段太短对于提取统计特征并无实际意义, 本文限定一个数据流分割长度的下限. 在新的数据流分段开始直至达到最小分割长度前, 算法只递推计算回归模型参数, 而不进行分割点检测. 对于本文采用的递推回归建模算法而言, 模型拟合均方差随递推次数的增加几乎呈线性衰减<sup>[11]</sup>, 因此采用较长的数据段有利于减小趋势特征值估计误差. 根据式(18)和(19), 广义似然比统计量分割点检测指标由趋势特征值决定, 因此减小趋势特征值误差也就减小了传递给分割点检测的误差.

### 4 仿真实验

取某炼油厂催化裂化过程再生催化剂温度传感器 2006 年 5 月 10 日和 11 日共 2880 个数据, 作为

仿真实验数据流. 本文所有仿真都在 Celeron 2.0-GHz CPU,256-Mb RAM 和 Win XP 操作系统计算机上,由 Matlab 6.5 编码完成.

仿真实验分别采用本文算法、SW 算法和 OSD 算法对上述数据流进行趋势分析,比较如下性能指标:1) 计算时间复杂度;2) 数据流分割的合理性;3) 趋势分析的精度.

为了便于比较,对上述 3 种算法作如下限定:1) 采用同一数据流进行分析(为了显示清晰,只取前 800 个数据点);2) 限定同样的数据流最小分割长度(30 个数据点);3) 实验数据流分段数相同.

在满足上述限定下,对各算法的参数尽可能优化设置. 本文算法迭代初始参数  $\alpha = 40$ ,分割点检测 ( $F$  检验) 阈值  $t_1 = 3.4$ ;SW 算法分割点检测(均方误差) 阈值  $t_2 = 24.3$ ;OSD 算法分割点检测(累积误差绝对值) 阈值  $t_{th} = 300$ . 第 1 分段回归建模数据序列长度取 40,其余各分段回归建模数据序列终点检测阈值<sup>[7]</sup>  $t_{h2} = 750$ .

3 种算法对实验数据流各点的计算时间如图 1 所示. 本文算法对各数据点的计算时间变化很小,基本不随各点在分段中位置不同而变化;SW 算法在分段内各点的计算时间近似呈线性增长;对于 OSD 算法,当达到各分段回归建模数据序列长度时,算法进行回归建模,因此在该点的计算时间与分段回归建模数据序列长度成正比,其他点只需以回归模型计算累积误差,计算时间基本不变. 由此,SW 算法当某一段长度很长或数据到达速率很高时,OSD 算法当某一段回归建模数据序列长度很长或数据到达速率很高时,都可能出现计算时间大于动态数据流到达时间间隔的情况,从而导致算法崩溃.

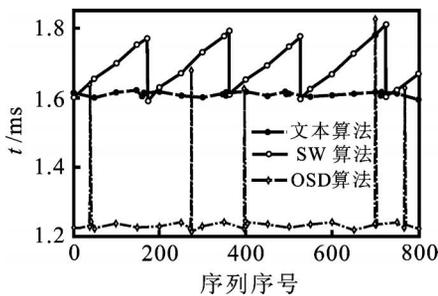


图 1 3 种算法对数据流各点的计算时间

本文算法、SW 算法和 OSD 算法进行数据流分段和线性回归模型建模情况如图 2 所示. 3 种算法都将序列分为 5 段. 以提取的趋势特征值拟合原序列的均方差为标准,3 种算法的趋势分析精度对比如表 1 所示,表中数值表示拟合均方差. 本文算法、SW 算法和 OSD 算法的拟合均方差分别为  $1.559 \times 10^4$ ,  $1.793 \times 10^4$  和  $3.164 \times 10^4$ . 本文算法的拟合均方差

最小,这说明本文算法确定的分割点较合理,趋势分析精度最高,而 OSD 算法精度最低.

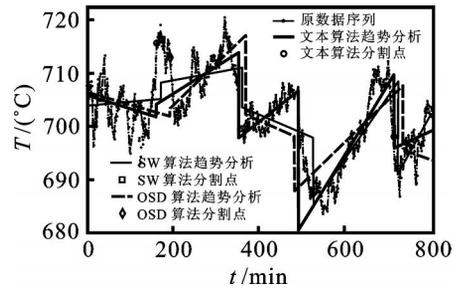


图 2 3 种算法的数据流趋势分析

表 1 3 种算法趋势分析精度比较

分段	SW 算法	本文算法	OSD 算法
1	4170.9	2261.5	7738.2
2	4505.0	5342.7	6528.8
3	3964.5	485.8	2258.1
4	4715.1	6298.7	13243
5	573.5	1198.2	1868.2
合计	17929	15586.9	31636.3

### 5 结 论

本文将增量递归最小二乘回归参数估计算法与广义似然比变化点检测算法相结合,提出一种数据流实时趋势提取算法. 该算法针对不断到达的数据流元素,采用增量机制确定数据序列回归模型参数和分割点,实时提取数据流趋势特征. 算法不但计算速度较快,而且精度较高.

### 参考文献(References)

- [1] Melek W W, Lu Z, Kapps A, et al. Comparison of trend detection algorithms in the analysis of physiological time-series data [J]. IEEE Trans on Biomed Engineering, 2005, 52(4): 639-651.
- [2] Beringer J, Hullermeier E. Online clustering of parallel data streams [J]. Data and Knowledge Engineering, 2005, 58(2): 180-204.
- [3] 王永利, 徐宏柄, 董逸生, 等. 基于低阶近似的多维数据流相关分析[J]. 电子学报, 2006, 43(10): 293-300. (Wang Y L, Xu H B, Dong Y S, et al. A correlation analysis algorithm based on low rank approximation for multiple dimension data streams [J]. Acta Electronica Sinica, 2006, 43(10): 293-300.)
- [4] Koski A, Juhola M, Meriste M. Syntactic recognition of ECG signals by attributed finite automata [J]. Pattern Recognition, 1995, 28(12): 1927-1940.
- [5] Shatkay H, Zdonik S. Approximate queries and representations for large data sequences [C]. Proc of 12th IEEE Int Conf on Data Engineering. Washington: IEEE Computer Society, 1996: 546-553.

(下转第 1191 页)

- 37th IEEE Conf on Decision and Control. Tampa, 1998, 1: 84-89.
- [3] Ning Lv, Xiaoyang Yu, Junfeng Wu. A fault diagnosis model through  $GK$  fuzzy clustering[C]. 2004 IEEE Int Conf on Systems, Man and Cybernetics. The Hague, 2004, 6: 5114-5118.
- [4] Laukonen E G, Passino K M, Krishnaswami V, et al. Fault detection and isolation for an experimental internal combustion engine via fuzzy identification [J]. IEEE Trans on Control System Technology, 1995, 3(3): 347-355.
- [5] Jaehong Park, Rizzoni G. An eigenstructure assignment algorithm for the design of fault detection filters [J]. IEEE Trans on Automatic Control, 1994, 39(7): 1521-1524.
- [6] Diego Del Gobbo, Marcello Napolitano, Parviz Famouri, et al. Experimental application of extended Kalman filtering for sensor validation[J]. IEEE Trans on Control Systems Technology, 2001, 9(2): 376-380.
- [7] Kabore P, Wang H. Design of fault diagnosis filters and fault-tolerant control for a class of nonlinear systems [J]. IEEE Trans on Automatic Control, 2001, 46(11): 1805-1810.
- [8] Gianluca Antonelli, Fabrizio Caccavale, Carlo Sansone, et al. Fault diagnosis for AUVs using support vector machines [C]. Proc of the 2004 IEEE Int Conf on Robotics & Automation. New Orleans, 2004: 4486-4491.
- [9] 张平, 陈宗基. 用于检测操纵面损伤的故障检测滤波器 [J]. 航空学报, 1999, 20(4): 371-373.  
(Zhang P, Chen Z J. Fault detection filter for the control surface failures of aircraft [J]. Acta Aeronautica et Astronautica Sinica, 1999, 20(4): 371-373.)
- [10] Jiang B, Staroswiechi M, Cocquempot V. Fault estimation in nonlinear uncertain systems using robust/sliding-mode observers [J]. IEE Proc of Control Theory Application, 2004, 151(1): 29-37.
- [11] Yi Xiong, Mehrdad Saif. Robust and nonlinear fault diagnosis using sliding mode observers [C]. Proc of 40th IEEE Conf on Decision and Control. Orlando, 2001: 567-572.
- [12] Marcel Staroswiechi, Bin Jiang. Fault identification for a class of linear systems based on adaptive observer [C]. Proc of 40th IEEE Conf on Decision and Control. Orlando, 2001: 2283-2288.
- [13] Hong Wang, Zhen Huang, Steve Daley. On the use of adaptive updating rules for actuator and sensor fault diagnosis[J]. Automatica, 1997, 33(2): 217-225.
- [14] Cesare Fantuzzi, Silvio Simani, Sergio Beghelli. Robust fault diagnosis of dynamic processes using parametric identification with eigenstructure assignment approach[C]. Proc of 40th IEEE Conf on Decision and Control. Orlando, 2001: 155-160.

### (上接第 1185 页)

- [6] Keogh E, Chu S, Hart D, et al. Segmenting time series: A survey and novel approach[C]. Proc of IEEE Int Conf on Data Mining. Los Jose: IEEE Computer Society, 2001: 289-296.
- [7] Sylvie C, Carlos G B, Catherine C, et al. Trends extraction and analysis for complex system monitoring and decision support [J]. Engineering Applications of Artificial Intelligence, 2005, 18(1): 21-36.
- [8] Bakshi B R, Stephanopoulos G. Representation of process trends — : Multi-scale extraction of trends from process data [J]. Computers and Chemical Engineering, 1994, 18(4): 267-302.
- [9] Vedam H, Venkatasubramanian, V, Bhalodia M. A B-spline based method for data compression, process monitoring and diagnosis [J]. Computers and Chemical Engineering, 1998, 22(S1): 827-830.
- [10] Zhou Q, Cluett W. Recursive identification of time-varying systems via incremental estimation [J]. Automatica, 1996, 32(10): 1427-1431.
- [11] Simon H. Adaptive filter theory [M]. 4th ed. Upper Saddle River: Prentice Hall, 2002.
- [12] Hawkins D M. Fitting multiple change-point models to data [J]. Computational Statistic & Data Analysis, 2001, 37(3): 323-341.
- [13] Wang Z J, Willett P. Joint segmentation and classification of time series using class-specific features [J]. IEEE Trans on Systems, Man and Cybernetics — Part B: Cybernetic, 2004, 34(2): 1056-1067.