

## 一个多维次成分并行提取算法及其收敛性分析

董海迪<sup>1</sup> 何兵<sup>1</sup> 刘刚<sup>1</sup> 郑建飞<sup>1</sup>

**摘要** 次成分分析是信号处理领域内一项重要的分析工具。目前, 多维次成分并行提取算法数量稀少, 而且现有的算法在应用时还存在很多限制条件。针对上述问题, 在分析研究 OJAm 次子空间跟踪算法的基础上, 采用加权矩阵法提出了一种多维次成分提取算法, 并采用递归最小二乘法对所提算法进行了简化, 最后采用李雅普诺夫函数法确定了所提算法的全局收敛域。相比现有算法, 所提算法对信号的特征值大小没有要求, 也不需要迭代过程中进行模值归一化操作, 同时算法具有较低的计算复杂度。仿真实验表明: 所提算法能够并行提取多维次成分, 而且收敛速度要优于现有同类算法。

**关键词** 多维次成分, 递归最小二乘, OJAm 算法, 李雅普诺夫函数

**引用格式** 董海迪, 何兵, 刘刚, 郑建飞. 一个多维次成分并行提取算法及其收敛性分析. 自动化学报, 2019, 45(2): 427-433

**DOI** 10.16383/j.aas.2018.c170343

### A Parallel Multiple Minor Components Extraction Algorithm and Its Convergence Analysis

DONG Hai-Di<sup>1</sup> HE Bing<sup>1</sup> LIU Gang<sup>1</sup> ZHENG Jian-Fei<sup>1</sup>

**Abstract** Minor component analysis is a powerful tool in the field of signal processing. Up to now, there are few algorithms that can extract multiple minor components in parallel. Some existing algorithms have some limitations for real applications. In order to solve these problems, a novel multiple minor components extraction algorithm is proposed by introducing the weighted matrix method into the OJAm minor subspace algorithm. A recursive least square (RLS) algorithm is also derived based on the proposed algorithm. The global convergence region of the proposed algorithm is obtained by using the Lyapunov function method. Compared with some existing algorithms, the proposed algorithm has no restriction on the values of eigenvalue and does not need normalization operation during iterations. Its computation complexity is less than those of some counterparts. Simulation results show that the proposed algorithm can adaptively extract multiple minor components from input signals in parallel and has faster convergence speed than some same-type algorithms.

**Key words** Multiple minor components (MC), recursive least square (RLS), OJAm algorithm, Lyapunov function

**Citation** Dong Hai-Di, He Bing, Liu Gang, Zheng Jian-Fei. A parallel multiple minor components extraction algorithm and its convergence analysis. *Acta Automatica Sinica*, 2019, 45(2): 427-433

在信号处理领域, 将信号自相关矩阵最小特征值对应的特征向量定义为信号的次成分 (Minor component, MC)。次

成分分析是一种从信号提取次成分的技术。次成分分析已经广泛应用于计算机视觉<sup>[1]</sup>、波达方向估计<sup>[2]</sup>、FIR 滤波器设计<sup>[3]</sup>、曲面拟合<sup>[4]</sup> 等领域。目前, 基于神经网络的次成分分析算法是国际上的一个研究热点。相比传统的特征值分解算法, 神经网络算法具有计算复杂度低、能够在线实施和跟踪非平稳信号等优点<sup>[5]</sup>。

根据提取次成分的维数, 次成分分析算法可以分为: 单维次成分提取、次子空间跟踪和多维次成分提取等三类<sup>[6]</sup>。目前, 学者们已经提出了大量的单维次成分提取算法和次子空间跟踪算法<sup>[2-3, 6]</sup>, 而多维次成分提取算法还非常稀少。传统的多维次成分提取大多是通过串行提取或空间旋转来实现<sup>[6]</sup>。串行提取法是首先利用单维次成分分析算法提取信号的第一个次成分, 然后利用膨胀技术依次提取下一个次成分。串行算法的缺点在于存储器需求量大、提取时间有延迟、误差累计传播等。空间旋转法是首先利用次子空间跟踪算法提取信号的次子空间, 然后进行空间旋转得到多维次成分。空间旋转法的缺点在于计算复杂度的增加<sup>[7]</sup>。

相比串行提取法和空间旋转法, 并行算法能够直接从信号中提取多维次成分, 且不需要中间转换过程, 因此可以避免两类算法的缺点。最早的并行提取算法是由芬兰学者 Oja 提出来的<sup>[8]</sup>, 该算法虽然可以并行提取多维次成分, 但是其要求信号的最小特征值必须小于 1; Mathew 等提出的算法<sup>[9]</sup> 虽然没有特征值大小的限制, 但是该算法只适用于信号具有多个相同的最小特征值; 基于 Douglas 次子空间跟踪算法, Jankovic 等<sup>[10]</sup> 提出了一种新型多维次成分提取算法—MDouglas 算法, 虽然该算法对特征值大小没有要求, 但是需要事先选取参数, 而且该参数选取的结果直接影响着算法的收敛性能; 采用主/次成分转换机制, Tan 等提出了一种并行提取算法<sup>[11]</sup>, 该算法的缺点在于需要事先对最小特征值进行估计; Bartelmaos 等所提出的 MCA-OOJAH 算法<sup>[12]</sup> 虽然对信号特征值没有要求, 但是需要在每次迭代过程中增加模值归一化措施, 以保证算法的收敛性。频繁模值归一化操作极大地增加了 MCA-OOJAH 算法的计算复杂度。综上所述可得, 现有算法存在以下几个方面的问题: 1) 对信号的特征值有要求; 2) 需要事先选取一些参数; 3) 算法计算过程复杂。本文研究目的就是发展能够避免上述缺点的多维次成分并行提取算法。

本文的主要贡献可以归纳为以下三个方面: 1) 提出了一种多维次成分并行提取算法, 该算法对信号的特征值没有要求, 而且不需要模值归一化操作; 2) 利用递归最小二乘 (Recursive least square, RLS) 技术对所提算法进行改进, 导出了一种具有低计算复杂度的算法; 3) 利用李雅普诺夫函数法完成了对所提算法的全局收敛性分析, 确定了所提算法的全局收敛域。

论文的结构安排如下: 第 1 节是对神经网络模型和次成分进行简介; 第 2 节是在研究 OJAm 算法的基础上, 提出两种多维次成分提取算法; 第 3 节是采用李雅普诺夫函数法对所提算法的全局收敛性进行证明; 第 4 节是通过仿真实验对所提算法的性能进行验证; 论文的总结安排在第 5 节。

### 1 问题描述

考虑如下一个具有多输入多输出的神经网络模型:

$$\mathbf{y}(k) = W^T \mathbf{x}(k), \quad k = 1, 2, \dots \quad (1)$$

其中,  $\mathbf{y}(k) \in \mathbf{R}^{r \times 1}$  是神经网络的输出,  $W \in \mathbf{R}^{n \times r}$  是神经网络的状态矩阵, 输入信号  $\mathbf{x}(k) \in \mathbf{R}^{n \times 1}$  是一个零均值的随

收稿日期 2017-06-21 录用日期 2017-10-11  
Manuscript received June 21, 2017; accepted October 11, 2017  
国家自然科学基金 (61403399) 资助  
Supported by National Natural Science Foundation of China (61403399)  
本文责任编辑 王占山  
Recommended by Associate Editor WANG Zhan-Shan  
1. 火箭军工程大学 西安 710025  
1. Rocket Force University of Engineering, Xi'an 710025

机过程, 这里作为神经网络的输入,  $n$  为输入向量的维数,  $r$  为所需提取次成分成分的维数.

令  $R = E[\mathbf{x}(k)\mathbf{x}^T(k)]$  为输入信号  $\mathbf{x}(k)$  的自相关矩阵,  $\lambda_i$  和  $\mathbf{u}_i$  ( $i = 1, 2, \dots, n$ ) 分别为自相关矩阵  $R$  的特征值和对应的特征向量. 根据矩阵理论可得: 矩阵  $R$  是一个对称非负定矩阵, 且其特征值均是非负的. 为了后续方便, 这里将其矩阵  $R$  的特征值按照从大到小方式排列, 即

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{n-r+1} \geq \dots \geq \lambda_n \geq 0 \quad (2)$$

则矩阵  $R$  的特征值分解可以表示为

$$R = U\Lambda U^T = U_1\Lambda_1U_1^T + U_n\Lambda_nU_n^T \quad (3)$$

其中,  $U = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n]$  是由矩阵  $R$  的特征向量构成的矩阵,  $\Lambda = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_n\}$  是由矩阵  $R$  的特征值构成的对角矩阵,  $U_1 = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{n-r}]$  是由前  $n-r$  个特征向量构成的矩阵,  $\Lambda_1 = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_{n-r}\}$  是由前  $n-r$  个特征值构成的对角矩阵,  $U_n = [\mathbf{u}_{n-r+1}, \mathbf{u}_{n-r+2}, \dots, \mathbf{u}_n]$  和  $\Lambda_n = \text{diag}\{\lambda_{n-r+1}, \lambda_{n-r+2}, \dots, \lambda_n\}$  分别为后  $r$  个特征向量和特征值构成的矩阵.

根据定义可知特征值  $\lambda_{n-r+1}, \lambda_{n-r+2}, \dots, \lambda_n$  所对应的特征向量  $\mathbf{u}_{n-r+1}, \mathbf{u}_{n-r+2}, \dots, \mathbf{u}_n$  称为信号的前  $r$  个次成分, 而由特征向量  $\mathbf{u}_{n-r+1}, \mathbf{u}_{n-r+2}, \dots, \mathbf{u}_n$  张成的子空间  $V_1 = \text{Span}\{\mathbf{u}_{n-r+1}, \mathbf{u}_{n-r+2}, \dots, \mathbf{u}_n\}$  称为信号的次子空间. 子空间跟踪算法的目的是能够估计出子空间  $V_1$  的一组基, 而多维次成分提取算法则是要找到确定的特征向量  $\mathbf{u}_{n-r+1}, \mathbf{u}_{n-r+2}, \dots, \mathbf{u}_n$ .

## 2 多维次成分并行提取算法

### 2.1 多维次成分提取算法的提出

在文献 [13] 中, Feng 等提出了 OJAm 次子空间跟踪算法. 当 OJAm 算法收敛时, 该算法的状态矩阵只能收敛到信号次子空间的一组正交基, 而非多维次成分. 通过对 OJAm 算法进行加权处理, 这里提出如下算法

$$\begin{aligned} W(k) &= W(k-1) + \eta \left[ W(k-1)A \times \right. \\ &\quad \left. (W^T(k-1)W(k-1)A)^{-2} - \hat{R}(k) \times \right. \\ &\quad \left. W(k-1)(W^T(k-1)\hat{R}(k)W(k-1))^{-1} \right] \quad (4) \end{aligned}$$

其中,  $\eta$  是神经网络的学习因子, 满足关系:  $0 < \eta \leq 1$ ;  $A = \text{diag}\{a_1, a_2, \dots, a_r\}$  是一个  $r \times r$  维对角矩阵, 其对角线元素均为正数, 且满足关系:  $a_r > a_{r-1} > \dots > a_1 > 0$ ;  $\hat{R}(k)$  为  $k$  时刻对自相关矩阵  $R$  的估计值, 即

$$\begin{aligned} \hat{R}(k) &= \frac{1}{k} \sum_{i=1}^k \alpha^{k-i} \mathbf{x}(i)\mathbf{x}^T(i) = \\ &\quad \frac{k-1}{k} \alpha \hat{R}(k-1) + \frac{1}{k} \mathbf{x}(k)\mathbf{x}^T(k) \quad (5) \end{aligned}$$

其中,  $\alpha$  被称为遗忘因子. 如果样本来自于一个平稳的随机过程, 则可以令  $\alpha = 1$ , 此时式 (5) 相当于对  $\mathbf{x}(k)\mathbf{x}^T(k)$  取瞬时数学期望. 显然当训练样本非常大时,  $\hat{R}(k)$  将等价于自相关矩阵  $R$ . 另一方面, 如果  $\mathbf{x}(k)$  取自一个非平稳的随机过程, 则遗忘因子  $\alpha$  应该在区间  $(0, 1)$  内取值, 以便对采样样本施加一个宽度为  $1/(1-\alpha)$  的遗忘窗. 该遗忘窗通过对不同时刻的采样样本赋予不同的权值来降低以往数据对于当前结

果的影响. 遗忘因子  $\alpha$  的取值是由采样信号来决定. 通常对于变化缓慢的样本,  $\alpha$  的取值要接近 1 以产生一个较大的遗忘窗; 而对于变化快速的样本, 遗忘窗口宽度应比较小, 此时  $\alpha$  则应该在接近 0 处取值<sup>[14]</sup>. 为了方便后续使用, 这里将式 (4) 所描述的算法记为 WOJAm (Weighted OJAm) 算法.

从式 (4) 可得: 学习因子  $\eta$  控制着算法的学习步长. 一般说来, 学习因子  $\eta$  越大, 算法的学习步长越大, 算法的收敛速度也越快, 但是学习因子过大有时会引起算法的震荡甚至发散. 反之, 学习因子越小, 算法的学习步长越小, 收敛速度也越慢, 因此过小的学习因子会降低算法的性能. 目前, 学习因子的选择主要依靠经验知识, 如何选取最优的学习因子仍是一个难以解决的问题. 从式 (5) 可得: 遗忘因子主要影响着信号的自相关矩阵. 根据文献 [15] 的研究, 遗忘因子对于算法的收敛速度几乎没有影响.

### 2.2 基于递归最小二乘的改进型算法

虽然 WOJAm 算法可以提取信号的多维次成分, 但其计算复杂度为  $n^2r + 5nr^2 + 11r^3/3$ , 高于 OJAm 算法  $n^2r + 2nr^2 + 10r^3/3$  的计算复杂度. 通常, 高昂的计算复杂度会在一定程度上限制算法的使用范围. 由于 RLS 技术能够在不改变算法性能的前提下降低算法的计算复杂度<sup>[16]</sup>, 因此本节将研究基于 RLS 技术的多维次成分提取算法.

假设在第  $i$  ( $1 \leq i \leq k$ ) 次迭代时, 输入数据  $\mathbf{x}(i)$  在神经网络状态矩阵  $W(k-1)$  上的投影可以通过式  $\mathbf{y}(i) \approx W^T(i-1)\mathbf{x}(i)$  来近似 (这一近似已经广泛应用于很多算法的推导过程中, 并不会影响算法的最终性能<sup>[17]</sup>). 通过这一近似, 就可以在任意时刻  $i$  ( $i = 1, 2, \dots, k$ ) 获得神经网络的输出  $\mathbf{y}(i)$  的估计值. 将式 (5) 代入式 (4) 可得:

$$\begin{aligned} W(k) &= W(k-1) + \mu \left[ W(k-1)A \times \right. \\ &\quad \left. (W^T(k-1)W(k-1)A)^{-2} - \right. \\ &\quad \left. \left( \sum_{i=1}^k \alpha^{k-i} \mathbf{x}(i)\mathbf{y}^T(i) \right) \left( \sum_{i=1}^k \alpha^{k-i} \mathbf{y}(i)\mathbf{y}^T(i) \right)^{-1} \right] \quad (6) \end{aligned}$$

为了方便计算, 这里做如下简记:

$$P(k) = \left( \sum_{i=1}^k \alpha^{k-i} \mathbf{y}(i)\mathbf{y}^T(i) \right)^{-1} \quad (7)$$

$$\tilde{W}(k) = \left( \sum_{i=1}^k \alpha^{k-i} \mathbf{x}(i)\mathbf{y}^T(i) \right) P(k) \quad (8)$$

应用矩阵求逆引理<sup>[18]</sup>, 则式 (7) 可以化简为:

$$\begin{aligned} P(k) &= \left( \alpha \sum_{i=1}^{k-1} \alpha^{k-i-1} \mathbf{y}(i)\mathbf{y}^T(i) + \mathbf{y}(k)\mathbf{y}^T(k) \right)^{-1} = \\ &\quad \left( \alpha P(k-1) + \mathbf{y}(k)\mathbf{y}^T(k) \right)^{-1} = \\ &\quad \alpha^{-1} \left( P(k-1) - \mathbf{g}(k)\mathbf{y}^T(k)P(k-1) \right) \quad (9) \end{aligned}$$

其中,  $\mathbf{g}(k) = P(k-1)\mathbf{y}(k)/(\alpha + \mathbf{y}^T(k)P(k-1)\mathbf{y}(k))$ . 从式 (9) 可得:  $\mathbf{g}(k) = P(k-1)\mathbf{y}(k)$ , 进而有:

$$\tilde{W}(k) = \left( \sum_{i=1}^k \alpha^{k-i} \mathbf{x}(i)\mathbf{y}^T(i) \right) P(k) =$$

$$\begin{aligned} & \left( \sum_{i=1}^{k-1} \alpha^{k-i-1} \mathbf{x}(i) \mathbf{y}^T(i) \right) P(k-1) + \mathbf{x}(k) \mathbf{y}_k^T P(k) - \\ & \left( \sum_{i=1}^{k-1} \alpha^{k-i-1} \mathbf{x}(i) \mathbf{y}^T(i) \right) P(k-1) \mathbf{y}(k) \mathbf{g}^T(k) = \\ & \tilde{W}(k-1) + \left( \mathbf{x}(k) - \tilde{W}(k-1) \mathbf{y}(k) \right) \mathbf{g}^T(k) \end{aligned} \quad (10)$$

上式推导过程中用到了如下结论:

$$\begin{aligned} & \mathbf{g}(k) \mathbf{y}^T(k-1) P(k-1) = \\ & \left( \mathbf{g}(k) \mathbf{y}^T(k-1) P(k-1) \right)^T = \\ & P(k-1) \mathbf{y}(k-1) \mathbf{g}^T(k) \end{aligned} \quad (11)$$

综合式 (6)~(10), 就可以利用  $k$  时刻的输入向量  $\mathbf{x}(k)$  和  $k-1$  时刻的  $\tilde{W}(k-1)$  来估计状态矩阵  $\tilde{W}(k)$ , 即完成了利用 RLS 技术对 WOJAm 算法的改进, 因此将该算法记为 RLS-WOJAm 算法, 其计算过程如下.

初始化: 令  $k=0$ , 选择合适的学习因子  $\eta$  和遗忘因子  $\alpha$ , 其他初始化参数设置为:  $P(0) = \delta I_r$  (其中  $\delta$  是一个非常小的数),  $\tilde{W}(0) = 0$ .

迭代过程: 令  $k = k + 1$ , 根据以下方程对神经网络状态矩阵进行更新迭代,

$$\begin{aligned} & \mathbf{y}(k) = W^T(k-1) \mathbf{x}(k) \\ & \mathbf{g}(k) = \frac{P(k-1) \mathbf{y}(k)}{\alpha + \mathbf{y}^T(k) P(k-1) \mathbf{y}(k)} \\ & P(k) = \alpha^{-1} (P(k-1) - \mathbf{g}(k) \mathbf{y}^T(k) P(k-1)) \\ & \tilde{W}(k) = \tilde{W}(k-1) + \left( \mathbf{x}(k) - \tilde{W}(k-1) \mathbf{y}(k) \right) \mathbf{g}^T(k) \\ & W(k) = W(k-1) - \eta \tilde{W}(k) + \\ & \eta W(k-1) A (W^T(k-1) W(k-1) A)^{-2} \end{aligned} \quad (12)$$

### 2.3 算法性能评价

本节对所提算法进行分析, 并将所提算法与一些现有算法进行比较:

1) 将 WOJAm 算法与 OJAm 算法进行对比可以发现, 两者相差的只是加权矩阵  $A$ . OJAm 算法只是在对状态矩阵的模值进行了约束, 而 WOJAm 算法则是通过该加权矩阵对状态矩阵不断地实施 Gram-Schmidt 正交化操作, 从而使得状态矩阵能够收敛到所需的次成分. 显然, 这里加权矩阵的作用是与文献 [16] 相类似的.

2) 从文献 [13] 可得, OJAm 算法是一个次子空间跟踪算法, 算法只能收敛到信号次子空间  $V_1 = \text{Span} \{ \mathbf{u}_{n-r+1}, \mathbf{u}_{n-r+2}, \dots, \mathbf{u}_n \}$  的一组基, 不一定是次成分  $\mathbf{u}_{n-r+1}, \mathbf{u}_{n-r+2}, \dots, \mathbf{u}_n$ , 而所提算法收敛结果则是确定的次成分. 根据矩阵理论<sup>[18]</sup> 可得: 子空间  $V_1$  有很多种不同的基, 而次成分只是其中一个特殊的基. 即所提算法不仅适用于多维次成分提取, 而且还可以用于子空间跟踪. 因此相比 OJAm 算法, 所提算法具有更为广泛的应用范围.

3) 这里将所提算法与近期提出的 AMMD 算法<sup>[11]</sup> 和 MCA-OOJAH 算法<sup>[12]</sup> 进行对比. 在应用 AMMD 算法前, 必须要对信号自相关矩阵的最大特征值进行估计, 由于信号是多种多样, 有些情况下是难以估计的, 因此这一约束限制了 AMMD 算法的实际应用. 而所提算法在使用前并没有这一要求, 因此拓宽了使用范围. MCA-OOJAH 算法虽然没有

限制条件, 但是其需要在每一次迭代过程中增加状态矩阵模值归一化操作, 以确保算法的稳定性, 由于大量的归一化操作, 增加了 MCA-OOJAH 算法的计算量, 而所提算法则没有这些要求, 降低了算法的硬件开销.

4) 虽然 WOJAm 算法的计算复杂度为  $n^2 r + 5nr^2 + O(n)$ , 但是由其导出的 RLS-WOJAm 算法的计算复杂度将至  $2nr^2 + 2nr + O(n)$ . 这一计算复杂度不仅要低于原来的 OJAm 次子空间跟踪算法 ( $n^2 r + 2nr^2 + 10r^3/3$ ), 而且还要比 AMMD 算法<sup>[11]</sup>  $n^3 + 2n^2 r + O(n)$  的计算复杂度和 MDouglas 算法<sup>[10]</sup>  $n^2 r + 5nr^2 + O(n)$  的计算复杂度要低很多. 此外, RLS-WOJAm 算法只涉及到矩阵的加、乘和简单求逆操作, 因此实施较为容易.

### 3 算法的全局收敛性分析

文献 [13] 只是对 OJAm 算法进行了收敛性分析, 并未给出算法的收敛域; 本节将通过李雅普诺夫函数法对所提算法的收敛性进行分析, 确定出算法的收敛域. 借鉴现有文献 [19] 分析结论可得: 所提 WOJAm 算法和 RLS-WOJAm 在本质上是相同的, 因此两个算法具有相同的全局收敛域. 假定输入信号  $\mathbf{x}(k)$  是一个零均值的平稳随机过程, 当学习因子  $\eta$  非常小时, 根据随机近似理论<sup>[20]</sup>, 则式 (4) 所描述的高散时间方程可以转化为相对应的一阶常微分方程

$$\frac{dW(t)}{dt} = RW(t) (W^T(t) RW(t))^{-1} - W(t) A (W^T(t) W(t) A)^{-2} \quad (13)$$

其中,  $t = \mu k$ . 通过对式 (13) 的收敛性分析就可以间接获得所提算法的全局收敛性. 事实上, 算法的全局收敛性主要是回答以下两个问题<sup>[21]</sup>:

- 1) 式 (13) 所描述的动态系统能否全局收敛到自相关矩阵的多维次成分?
- 2) 算法对于次成分的吸引域是多少? 或者说确保算法全局收敛的初始条件是什么?

为了回答上述问题, 这里构造如下函数:

$$L(W) = -\frac{1}{2} \text{tr} \left\{ \ln \left[ W^T R_r W \right] + \left( W^T W A \right)^{-1} \right\} \quad (14)$$

其中,  $R_r = U_n \Lambda_n U_n^T$ . 所提算法的收敛性可以通过定理 1 来完成.

**定理 1.**  $L(W)$  是常微分方程 (13) 所对应的李雅普诺夫函数, 通过该函数可以获得神经网络状态矩阵  $W$  在平稳点  $W = U_n A^{-\frac{1}{2}}$  处的吸引域为

$$\Omega = \left\{ W \mid 0 < W^T R_r W < \infty, W^T W A \neq 0 \right\} \quad (15)$$

**证明.** 由于  $L(W)$  是一个连续函数, 则其是连续可微的:

$$\nabla L(W) = - \left[ R_r W (W^T R_r W)^{-1} - W A (W^T W A)^{-2} \right] \quad (16)$$

通过链式法则, 可以获得  $L(W)$  对于时间  $t$  的导数

$$\begin{aligned} \frac{dL(W(t))}{dt} &= \text{tr} \left[ \nabla L(W) \cdot \frac{dW^T}{dt} \right] = \\ & - \text{tr} \left\{ \left[ R_r W (W^T R_r W)^{-1} - W A (W^T W A)^{-2} \right]^T \cdot \right. \end{aligned}$$

$$\left[ RW(W^T RW)^{-1} - WA(W^T WA)^{-2} \right] \quad (17)$$

为了方便起见, 上式中省略了时间变量  $t$ . 根据李雅普诺夫第二定律, 需要证明如下结论: 在域  $\Omega^* = \Omega - \{W|W = U_n A^{-\frac{1}{2}}\}$  内  $(dL(W)/dt) < 0$ , 而在域  $\{W|W = U_n A^{-\frac{1}{2}}\}$  内  $(dL(W)/dt) = 0$ . 为了完成上述证明, 这里将  $R = U_1 \Lambda_1 U_1^T + U_n \Lambda_n U_n^T$  代入式 (17), 并进行适当化简可得:

$$\frac{dL(W)}{dt} = -\text{tr} \left\{ \begin{bmatrix} -Z_1 A (Z_2^T Z_2 A)^{-2} \\ \Lambda_2 Z_2 (Z_2^T \Lambda_2 Z_2)^{-1} - Z_2 A (Z_2^T Z_2 A)^{-2} \end{bmatrix}^T \right. \\ \left. \begin{bmatrix} \Lambda_1 Z_1 (Z_1^T \Lambda_1 Z_1 + Z_2^T \Lambda_2 Z_2)^{-1} - Z_1 A (Z_2^T Z_2 A)^{-2} \\ \Lambda_2 Z_2 (Z_1^T \Lambda_1 Z_1 + Z_2^T \Lambda_2 Z_2)^{-1} - Z_2 A (Z_2^T Z_2 A)^{-2} \end{bmatrix} \right\} \quad (18)$$

其中,  $Z_1 = U_1^T W$  和  $Z_2 = U_n^T W$ . 由于  $W^T R_r W \neq 0$ , 则矩阵  $Z_2$  是可逆的. 因此, 必定存在一矩阵  $B \in \mathbf{R}^{(n-r) \times r}$  使得  $Z_1 = B Z_2$  成立. 将式 (18) 中的  $Z_1$  用  $Z_2$  来表示, 则有:

$$\frac{dL(W)}{dt} = \text{tr} \left\{ 2C + C Z_2^T (I + B^T B) Z_2 C^T \right\} - \text{tr} \left\{ Z_2^{-1} (I + B^T \Lambda_1 B \Lambda_2^{-1})^{-1} Z_2^{-T} \right\} \quad (19)$$

其中,  $C = (Z_2^T Z_2)^{-1} A^{-1} (Z_2^T Z_2)^{-1}$ . 当  $W$  接近平稳点  $W = U_n A^{-\frac{1}{2}}$  处时, 矩阵  $C$  可以近似为一个对角矩阵, 即  $C \approx C' = \text{diag}\{c_1, c_2, \dots, c_r\}$ . 此时, 式 (19) 就可以化简为:

$$\frac{dL(W)}{dt} \approx \text{tr} \left\{ 2C' + C' Z_2^T (I + B^T B) Z_2 C' \right\} - \text{tr} \left\{ Z_2^{-1} (I + B^T \Lambda_1 B \Lambda_2^{-1})^{-1} Z_2^{-T} \right\} \quad (20)$$

接下来, 考虑矩阵  $Z_1 \neq 0$  的情况. 根据矩阵理论有:

$$\text{tr} \left\{ Z_2^T (I + B^T B) Z_2 \right\} < \text{tr} \left\{ Z_2 (I + B^T \Lambda_1 B \Lambda_2^{-1}) Z_2 \right\} \quad (21)$$

令  $D = Z_2^T (I + B^T B) Z_2$ , 则其特征值分解可以表示为:

$$D = \Phi \Psi \Phi^T \quad (22)$$

其中,  $\Psi = \text{diag}\{\mu_1, \mu_2, \dots, \mu_r\}$  是一对角矩阵,  $\mu_i$  ( $i = 1, 2, \dots, r$ ) 是矩阵  $D$  的特征值,  $\Phi$  是由特征向量组成的矩阵. 将式 (22) 代入式 (21) 可得:

$$\frac{dL(W)}{dt} < -\text{tr} \left\{ -2C' + C' D C' + D^{-1} \right\} = 2\text{tr} (C') - \text{tr} (C' \Psi C') - \text{tr} [(\Psi)^{-1}] = \sum_{i=1}^r \left( 2c_i - c_i^2 \mu_i - \frac{1}{\mu_i} \right) \leq 0 \quad (23)$$

接下来考虑  $Z_1 = 0$  的情况, 此时式 (19) 可简化为:

$$\frac{dL(Z)}{dt} = -\text{tr} (E^T E) \leq 0 \quad (24)$$

其中,  $E = \Lambda_2 Z_2 (Z_2^T \Lambda_2 Z_2)^{-1} - Z_2 A (Z_2^T Z_2 A)^{-2}$ .

从式 (24) 可得: 当且仅当  $E = 0$  时, 有  $dL(Z)/dt = 0$  成立. 此时  $Z_2 = A^{-\frac{1}{2}}$ , 即  $W = U_n A^{-\frac{1}{2}}$ . 从式 (23) 和式 (24) 可得: 对于任意的  $W \in \Omega^*$  均有  $d(L(Z))/dt < 0$  成立. 因此, 在域  $\Omega$  内平稳点  $W = U_n A^{-\frac{1}{2}}$  是渐近稳定的, 即只要初始化状态矩阵  $W$  在域  $\Omega$  以内, 则算法 (13) 全局渐近收敛到自相关矩阵  $R$  的前  $r$  个次成分.  $\square$

## 4 仿真实验

本节将通过两个数值仿真实验来验证所提算法的性能. 第一个实验是考察所提算法提取多维次成分的能力, 第二个实验将所提算法与一些现有算法进行对比. 在仿真实验过程中, 为了衡量算法的性能, 这里采取两个评价函数, 第一个是方向余弦 (Direction cosine, DC):

$$DC_i(k) = \frac{|W_i^T(k) \mathbf{v}_i|}{\|W_i(k)\| \|\mathbf{v}_i\|} \quad (25)$$

其中,  $W_i(k)$  表示在  $k$  时刻神经网络状态矩阵的第  $i$  列, 而  $\mathbf{v}_i$  则是代表着自相关矩阵  $R$  的第  $i$  个次成分. 显然, 如果  $W_i(k)$  能够收敛到次成分  $\mathbf{v}_i$  的方向, 则方向余弦的值应该收敛到 1.

方向余弦只能评价状态矩阵的收敛方向, 而状态矩阵模值也是一个非常重要的评价函数:

$$\text{Norm}_i(k) = |W_i^T(k) W_i(k)|, \quad i = 1, 2, \dots, r \quad (26)$$

当神经网络状态矩阵收敛到次成分的方向 (此时方向余弦等于 1) 时, 如果状态矩阵模值能够收敛到一个常值, 则算法是收敛的.

### 4.1 多维次成分提取实验

本实验考察所提算法提取多维次成分的能力. 实验中, 输入信号由式  $\mathbf{x}(k) = T \mathbf{z}(k)$  产生, 其中,  $T = \text{randn}(10, 10)/10$  是一个随机产生的  $10 \times 10$  维矩阵, 而  $\mathbf{z}(k) \in \mathbf{R}^{10 \times 1}$  是一个均值为零方差为  $\sigma^2 = 1$  的高斯白噪声序列. 这里分别采用 WOJAm 算法和 RLS-WOJAm 算法对信号的前 3 个次成分进行提取 (即  $r = 3$ ). 两种算法的初始化参数设置如下: 根据加权矩阵  $a_r > a_{r-1} > \dots > a_1 > 0$  的要求, 这里取加权矩阵  $A = \text{diag}\{3, 2, 1\}$ ; 根据  $0 < \eta \leq 1$  要求, 同时也要保证算法的收敛性能, 这里取学习因子  $\eta = 0.1$ ; 由于输入信号是一个近近平稳的随机过程, 所以这里取遗忘因子  $\alpha = 0.998$ ; 此外 RLS-WOJAm 算法中初始化逆矩阵  $P(0) = 0.001 \times I_3$ , 其中  $I_3$  是一个  $3 \times 3$  的单位矩阵. 两个算法的采取相同的初始化状态矩阵, 且该矩阵是随机产生的. 仿真结果如图 1~ 图 4 所示, 该结果是 100 次独立实验的平均值.

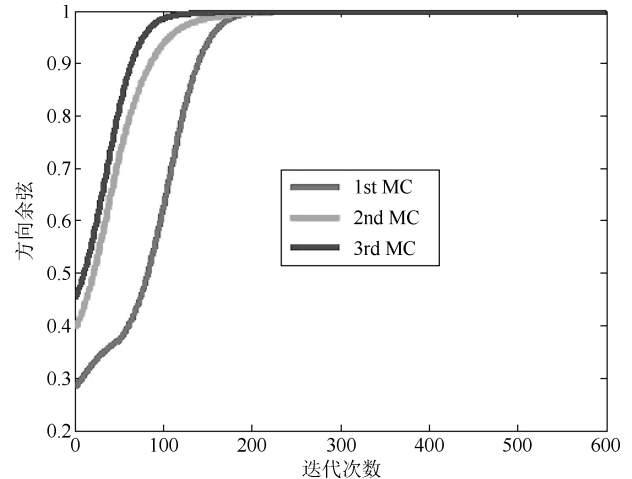


图 1 WOJAm 算法的 DC 曲线  
Fig. 1 DC curves of WOJAm algorithm

图 1 和图 3 分别是两种算法状态矩阵列向量与所求次成分之间的方向余弦曲线; 图 2 和图 4 分别是状态矩阵各列向量的模值, 黑线则是整个状态矩阵的  $F$ -范数. 从图 1 和图 3 中可以看出, 经历若干次迭代后, WOJAm 算法和 RLS-WOJAm 算法的方向余弦均收敛到 1, 即两种算法均最终收敛到了信号次成分的方向. 从图 2 和图 3 中可得: 两种算法在方向余弦收敛到 1 的同时, 状态矩阵模值也已经收到一个常值. 综合两图可以得出结论: 本文所提两种算法均能够自适应地提取信号的多维次成分, 而且具有很好的收敛特性.

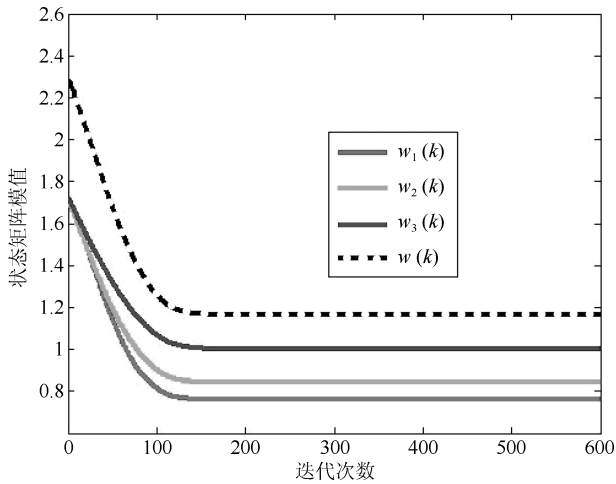


图 2 WOJAm 算法的 Norm 曲线  
Fig. 2 Norm curves of WOJAm algorithm

此外对比上述 4 图可以得出结论: 对于方向余弦曲线和状态矩阵模值曲线, WOJAm 算法和 RLS-WOJAm 算法的收敛情况近似相同, 只是 WOJAm 算法的收敛速度略快于 RLS-WOJAm 算法. 由于 RLS-WOJAm 算法是由 WOJAm 算法推导而来, 两者本质上是相同的, 所以上述现象很容易解释. 从第 2.3 节可得, RLS-WOJAm 算法的计算复杂度远小于 WOJAm 算法, 因此 RLS-WOJAm 算法更符合实际使用需要. 在后续实验中, 将重点对 RLS-WOJAm 算法的性能进行考核.

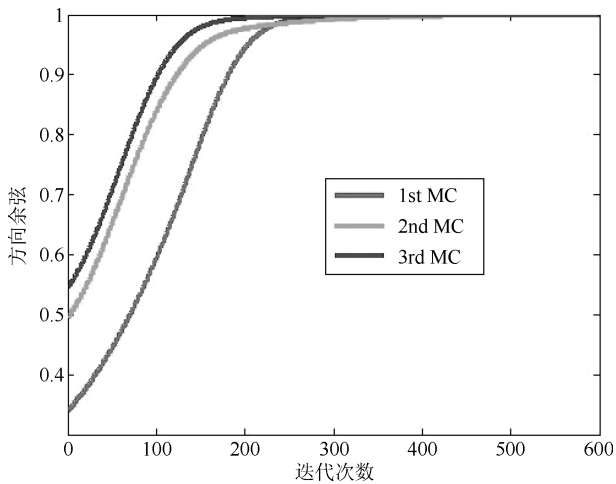


图 3 RLS-WOJAm 算法的 DC 曲线  
Fig. 3 DC curves of RLS-WOJAm algorithm

#### 4.2 算法性能对比实验

为了突出所提算法的性能, 本实验将所提 RLS-WOJAm 算法与 MCA-OOJAH 算法<sup>[12]</sup> 和 AMMD 算法<sup>[11]</sup> 进行比较. 实验中输入信号采用如下—阶滑动回归模型来产生:

$$x(k) = 0.75x(k - 1) + e(k) \quad (27)$$

该模型由一个均值为零方差为 1 的高斯白噪声  $e(k)$  作为模型驱动输入. 取该模型的 10 个不连续的输出构成神经网络的输入向量. 然后利用上述三种算法对该输入信号的前 3 个次成分进行提取, 即  $r = 3$ . 试验中三个算法采用相同的初始化状态矩阵, 学习因子也都设置为 0.1, 仿真结果如图 5~图 8 所示.

图 5~图 7 分别是所提次成分的方向余弦曲线, 图 8 是在迭代过程中三种算法的状态矩阵模值曲线. 从图 8 中可以看出, 由于 MCA-OOJAH 算法在迭代过程中存在模值归一化操作, 所以其模值始终为一常值, 而 RLS-WOJAm 算法和 AMMD 算法虽然没有归一化操作, 但状态矩阵模值仍能收敛到一个固定值. 从图 5~图 7 中可以发现: 在所有次成分的提取过程中, RLS-WOJAm 算法的收敛速度均快于其他两种算法.

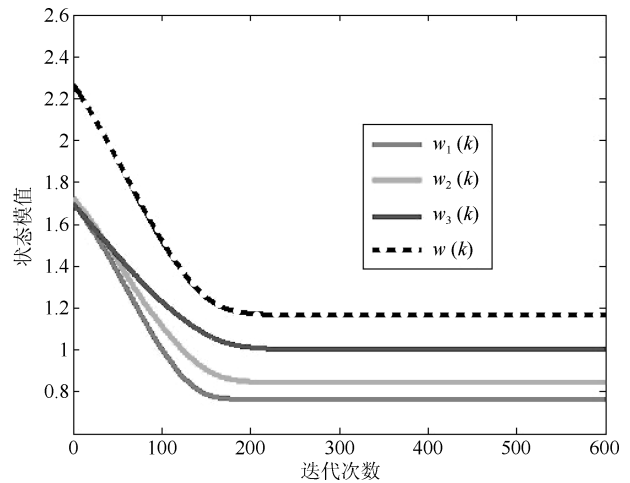


图 4 RLS-WOJAm 算法的 Norm 曲线  
Fig. 4 Norm curves of RLS-WOJAm algorithm

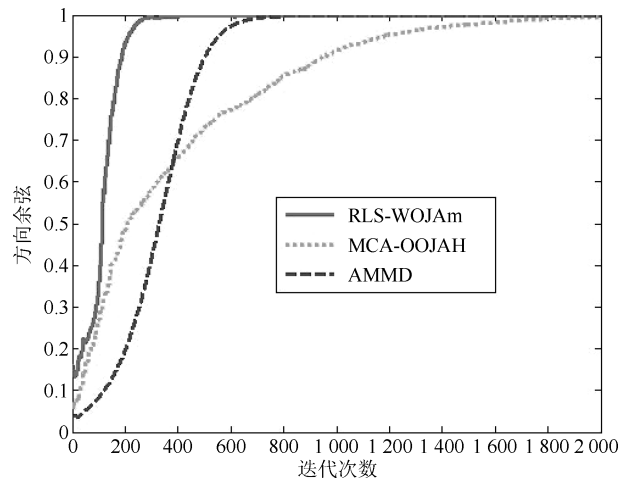


图 5 第一个次成分的 DC 曲线  
Fig. 5 DC curves of the 1st MC

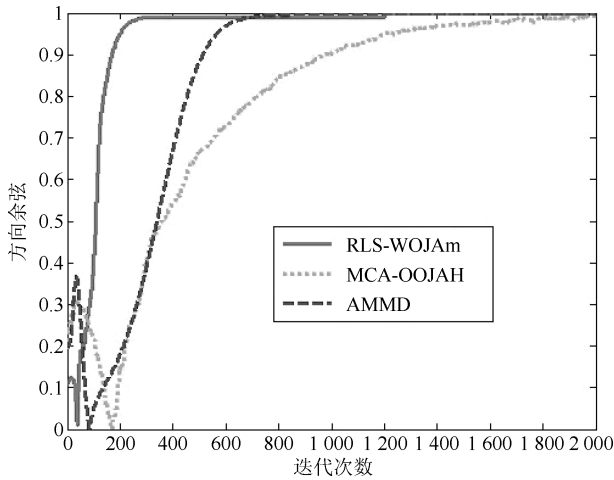


图6 第二个次成分的 DC 曲线  
Fig.6 DC curves of the 2nd MC

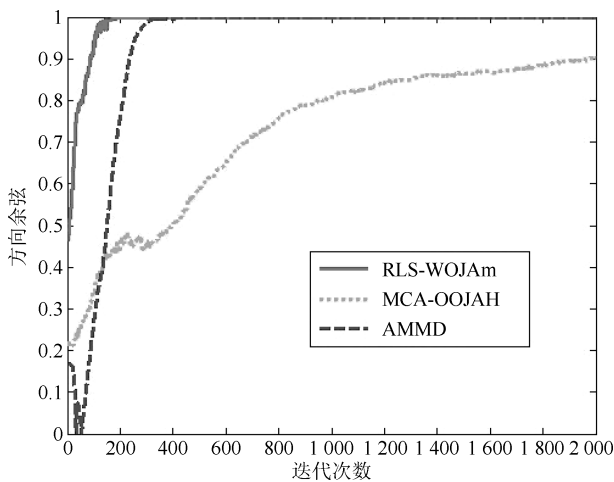


图7 第三个次成分的 DC 曲线  
Fig.7 DC curves of the 3rd MC

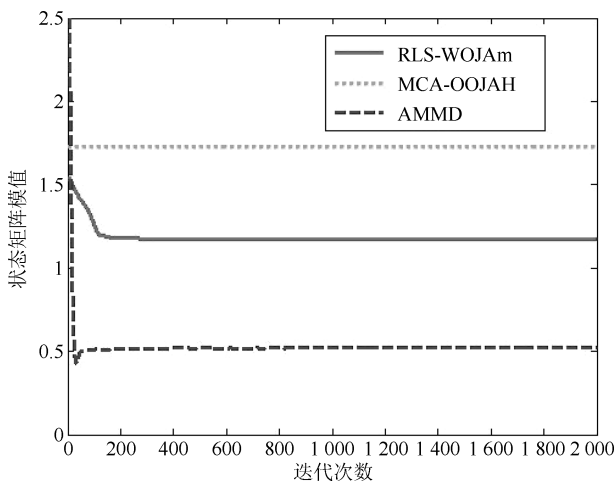


图8 三种算法的 Norm 曲线  
Fig.8 Norm curves of the three algorithms

## 5 结论

本文对多维次成分并行提取算法进行了研究. 首先, 通过对 OJAm 次子空间跟踪算法进行加权改造, 提出了一种多维

次成分并行提取算法; 然后, 为了降低该的计算复杂度的, 采用递归最小二乘技术导出了一种新型算法—RLS-WOJAm 算法; 相比现有算法, 所提算法对信号自相关矩阵的特征值大小没有要求, 也没有模值归一化措施; 最后, 通过李雅普诺夫函数法确定了所提算法的全局收敛域. 仿真实验表明: 相比现有一些同类型算法, 所提算法具有较快的收敛速度.

## References

- 1 Thameri M, Abed-Meraim K, Belouchrani A. Low complexity adaptive algorithms for Principal and Minor Component Analysis. *Digital Signal Processing*, 2013, **23**(1): 19–29
- 2 Arjomandi-Lari M, Karimi M. Generalized YAST algorithm for signal subspace tracking. *Signal Processing*, 2015, **117**: 82–95
- 3 Jou Y D, Sun C M, Chen F K. Eigenfilter design of FIR digital filters using minor component analysis. In: 9th International Conference on Information, Communications and Signal Processing. Tainan, China: IEEE, 2013. 1–5
- 4 Pattanadach N, Yutthagowith P. Fast curve fitting algorithm for parameter evaluation in lightning impulse test technique. *IEEE Transactions on Dielectrics and Electrical Insulation*, 2015, **22**(5): 2931–2936
- 5 Cirrincione G, Cirrincione M. *Neural-Based Orthogonal Data Fitting: The EXIN Neural Networks*. New York, USA: John Wiley and Sons, 2011
- 6 Kong X Y, Hu C H, Duan Z S. Neural networks for principal component analysis. *Principal Component Analysis Networks and Algorithms*. Singapore: Springer, 2017. 47–73
- 7 Gao Ying-Bin, Kong Xiang-Yu, Hu Chang-Hua, Hou Li-An. Modified Möller algorithm for multiple minor components extraction. *Control and Decision*, 2017, **32**(3): 493–497 (高迎彬, 孔祥玉, 胡昌华, 侯立安. 并行提取多个次成分的改进型 Möller 算法. *控制与决策*, 2017, **32**(3): 493–497)
- 8 Oja E. Principal components, minor components, and linear neural networks. *Neural Networks*, 1992, **5**(6): 927–935
- 9 Mathew G, Reddy V U. Orthogonal eigensubspace estimation using neural networks. *IEEE Transactions on Signal Processing*, 1994, **42**(7): 1803–1811
- 10 Jankovic M V, Reljin B. A new minor component analysis method based on Douglas-Kung-Amari minor subspace analysis method. *IEEE Signal Processing Letters*, 2005, **12**(12): 859–862
- 11 Tan K K, Lv J C, Zhang Y, Huang S N. Adaptive multiple minor directions extraction in parallel using a PCA neural network. *Theoretical Computer Science*, 2010, **411**: 4200–4215
- 12 Bartelmaos S, Abed-Meraim K. Fast adaptive algorithms for minor component analysis using Householder transformation. *Digital Signal Processing*, 2011, **21**(6): 667–678
- 13 Feng D Z, Zheng W X, Jia Y. Neural network learning algorithms for tracking minor subspace in high-dimensional data stream. *IEEE Transactions on Neural Networks*, 2005, **16**(3): 513–521

- 14 Nguyen T D, Takahashi N, Yamada I. An adaptive extraction of generalized eigensubspace by using exact nested orthogonal complement structure. *Multidimensional Systems and Signal Processing*, 2013, **24**(3): 457–483
- 15 Yang B. Projection approximation subspace tracking. *IEEE Transactions on Signal Processing*, 1995, **43**(1): 95–107
- 16 Gao Y B, Kong X Y, Hu C H, Li H Z, Hou L A. A generalized information criterion for generalized minor component extraction. *IEEE Transactions on Signal Processing*, 2017, **65**(4): 947–959
- 17 Nguyen T D, Yamada I. Adaptive normalized quasi-newton algorithms for extraction of generalized Eigen-Pairs and their convergence analysis. *IEEE Transactions on Signal Processing*, 2013, **61**(6): 1404–1418
- 18 Lewis D W. *Matrix Theory*. Singapore: World Scientific, 2012
- 19 Miao Y, Hua Y. Fast subspace tracking and neural network learning by a novel information criterion. *IEEE Transactions on Signal Processing*, 1998, **46**(7): 1967–1979
- 20 Gao Y B, Kong X Y, Hu C H, Zhang H H, Hou L A. Convergence analysis of Möller algorithm for estimating minor component. *Neural Processing Letters*, 2014, **42**(2): 355–368
- 21 Kong X Y, Hu C H, Ma H G, Han C Z. A unified self-stabilizing neural network algorithm for principal and minor components extraction. *IEEE Transactions on Neural Networks and Learning Systems*, 2012, **23**(2): 185–198
- 董海迪 火箭军工程大学空间工程系博士生. 主要研究方向为自适应信号处理. E-mail: donghaidi123@163.com  
(**DONG Hai-DI** Ph. D. candidate in the Department of Space Engineering, Rocket Force University of Engineering. His main research interest is adaptive signal processing.)
- 何 兵 火箭军工程大学空间工程系副教授. 主要研究方向为系统特征提取, 自适应信号处理. 本文通信作者. E-mail: hb830513@126.com  
(**HE Bing** Associate professor in the Department of Space Engineering, Rocket Force University of Engineering. His research interest covers system feature extracting and adaptive signal processing. Corresponding author of this paper.)
- 刘 刚 火箭军工程大学空间工程系教授. 主要研究方向为系统特征提取, 自适应信号处理.  
E-mail: liugangepgc@163.com  
(**LIU Gang** Professor in the Department of Space Engineering, Rocket Force University of Engineering. His research interest covers system feature extracting and adaptive signal processing.)
- 郑建飞 火箭军工程大学控制工程系讲师. 主要研究方向为预测与健康管理, 可靠性和预测维护.  
E-mail: zjf302@126.com  
(**ZHENG Jian-Fei** Lecturer in the Department of Automation Engineering, Rocket Force University of Engineering. His research interest covers prognostics and health management, reliability, and predictive maintenance.)