

文章编号:1000-8055(2010)05-1177-07

基于 K 均值聚类 and 约简最小二乘支持 向量回归机的推力估计器设计

赵永平¹, 孙健国², 王前宇², 陈霆昊²

- (1. 南京理工大学 机械工程学院 智能弹药技术国家重点实验室, 南京 210094;
2. 南京航空航天大学 能源与动力学院, 南京 210016)

摘 要: 提出了一种基 K 均值聚类和约简最小二乘支持向量回归机的推力估计器设计方法. 首先用 K 均值聚类法将全包线范围内的数据进行聚类, 然后在每一个类当中, 用迭代约简最小二乘支持向量回归机设计一个子推力估计器. 在用迭代约简最小二乘支持向量回归机设计子推力估计器的过程中, 为了使计算数值更稳定, 用 Cholesky 分解代替原来的迭代方法. 最后仿真实验表明, 此推力估计器能满足直接推力控制的需要, 并和别的方案比较起来, 该方案存在一定的优势.

关 键 词: 支持向量机; 最小二乘; K 均值聚类; 直接推力控制

中图分类号: V233.7

文献标识码: A

Thrust estimator design based on K means clustering and reduced least squares support vector regression

ZHAO Yong-ping¹, SUN Jian-guo², WANG Qian-yu², CHEN Ting-hao²

- (1. Intelligent Ammunition Technology Laboratory of National Defence,
School of Mechanical Engineering, Nanjing University of Science and Technology,
Nanjing 210094, China;

2. College of Energy and Power Engineering,
Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: In this paper, a design scheme for thrust estimator was proposed based on K -means clustering and reduced least squares support vector regression. Firstly, the K -means clustering algorithm was utilized to cluster the data in the full flight envelope, and the, a sub-estimator was designed in each cluster using the recursive reduced least squares support vector regression (RRLSSVR). In the process of designing the sub-estimator with RRLSSVR, the Cholesky factorization was utilized to replace the original iteration method for achieving higher numerical stability. Finally, the simulation experiments show that the thrust estimator can satisfy the requirement of direct thrust control for aeroengines, and as compared with other scheme for thrust estimator, the proposed scheme in this paper is also superior.

Key words: support vector machine; least squares; K -means clustering;
direct thrust control

众所周知, 航空发动机主要是为了给飞机提 供推力, 控制航空发动机的目的主要是为了控制

收稿日期: 2009-04-02; 修订日期: 2009-11-14

基金项目: 国家自然科学基金(50576033)

作者简介: 赵永平(1982 -), 男, 河南南阳人, 博士生, 主要研究航空发动机控制与故障诊断. E-mail: y. p. zhao @nuaa. edu. cn

推力.然而,在飞行中推力是不可测量的,传统的控制方法是通过控制与推力关系比较密切的可测参数如转速或者压比达到控制推力的目的,其结构如图 1 所示.但是在发动机全状态及全飞行包线范围内,发动机工作条件变化很大,发动机转速或压比的大小并不能直接而准确地表示推力的大小,若考虑到发动机的制造、安装公差及使用期内的性能蜕化等因素,那么转速或压比与推力之间

更不是简单的对应关系.鉴于上述情况,在发动机及其控制系统设计中,不得不保留足够的安全裕度,以补偿上述的诸多不确定因素,其后果是发动机性能潜力没有得到充分的发挥^[1].这是一种保守的设计方法.为了充分挖掘航空发动机的潜在性能以满足日益增长的飞机对推力的需求,就产生了航空发动机的直接推力控制^[2],其结构如图 2 所示.

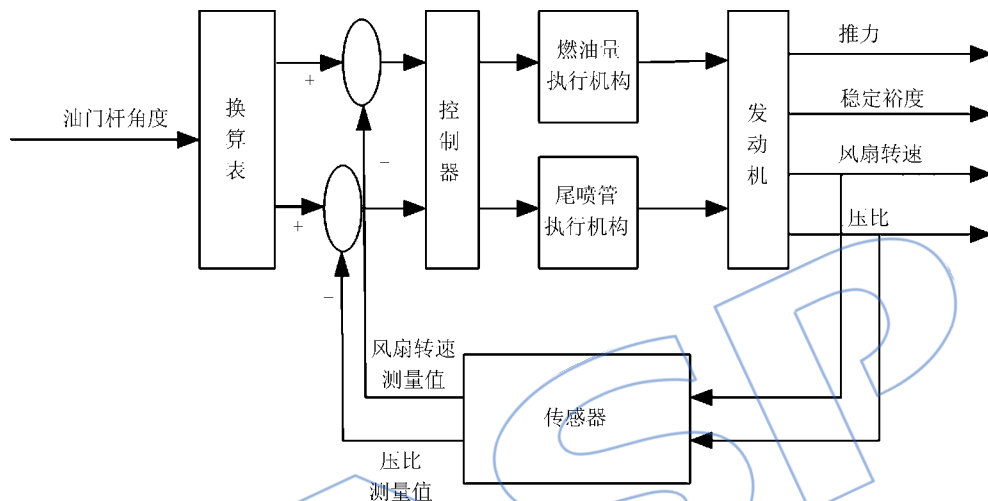


图 1 航空发动机的传统控制系统结构图

Fig. 1 Block diagram of conventional control for aero-engines

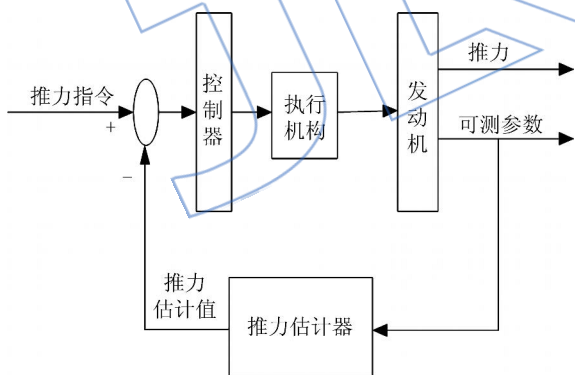


图 2 基于推力估计器的直接推力控制结构图

Fig. 2 Block diagram of direct thrust control for aero-engines

从图 2 可以看出,要实现直接推力控制,首先需要设计一个高精度的推力估计器.为此,本文基于 K -均值聚类和约简最小二乘支持向量回归机设计出了一个高精度的全包线推力估计器.仿真实验表明这种推力估计器能满足直接推力控制的需要.另外,和现有的推力估计器相比较,这种设计的推力估计器无论在实时性或者精度上都占有一定的优势.

支持向量机^[3-4]是一种有力的解决模式识别

和非线性函数估计的机器学习方法.然而,在建立支持向量机的过程当中,由于要实现二次优化求解,因此其训练代价比较高.作为支持向量机的一种变体,最小二乘支持向量机^[5]用等式约束代替不等式约束并把误差的一次方转化为二次方,最后归结为求解一组线性方程组.因而,其训练代价大大降低.但和支持向量机相比较,最小二乘支持向量机的解存在缺乏稀疏性的缺陷^[6].为了实现最小二乘支持向量解的稀疏性,人们提出了很多实现其解稀疏性的方法^[7-10],其中包括最近提出的迭代约简最小二乘支持向量回归机^[10].因此,本文提出用约简最小二乘支持向量回归来设计推力估计器,它既能缩短推力估计器的预测时间又能充分利用全部训练数据中携带的有用信息来满足直接推力控制对精度的需要.但在迭代建模的过程当中,为了使计算数值更稳定,本文用 Cholesky 分解代替原来的 Sherman-Morrison 公式^[11].

1 约简最小二乘支持向量回归机

首先给出标准最小二乘支持向量回归机的数学模型,然后引出约简最小二乘支持向量回归机.

假如有一个训练数据集 $\left\{ \left(x_i, d_i \right) \right\}_{i=1}^N$, 其中 $x_i \in \mathbf{R}^n$ 代表系统的输入量, $d_i \in \mathbf{R}$ 代表系统的输出量, 最小二乘支持向量回归机的数学模型如下:

$$\min_{w, e} J(w, e) = \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^N e_i^2 \quad (1)$$

$$\text{s. t. } d_i = w^T \phi(x_i) + b + e_i, \\ i = 1, \dots, N$$

其中 $C \in \mathbf{R}^+$ 是正则化参数, $e = [e_1, \dots, e_N]^T$, $\phi(\cdot)$ 是一个能把 x_i 从输入空间映射到高维特征空间中的非线性映射. 式(1)的 Lagrange 函数为

$$L(w, b, e; \bar{a}) = \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^N e_i^2 + \sum_{i=1}^N \bar{a}_i [d_i - w^T \phi(x_i) - b - e_i] \quad (2)$$

其中 $\bar{a} = [\bar{a}_1, \dots, \bar{a}_N]^T$ 为 Lagrange 乘子. 优化的 KKT (Karush-Kuhn-Tucker) 条件为

$$\begin{cases} \frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^N \bar{a}_i \phi(x_i) \\ \frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^N \bar{a}_i = 0 \\ \frac{\partial L}{\partial e_i} = 0 \rightarrow \bar{a}_i = C e_i \\ \frac{\partial L}{\partial \bar{a}_i} = 0 \rightarrow w^T \phi(x_i) + b + e_i - d_i = 0 \end{cases} \quad (3)$$

从式(3)中消去 w 和 e , 可以得到如下线性方程组:

$$\begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & \bar{K} \end{bmatrix} \begin{bmatrix} b \\ \bar{a} \end{bmatrix} = \begin{bmatrix} 0 \\ d \end{bmatrix} \quad (4)$$

其中

$$\mathbf{1} = [1, \dots, 1]^T,$$

$$\bar{a} = [\bar{a}_1, \dots, \bar{a}_N]^T,$$

$$d = [d_1, \dots, d_N]^T,$$

$$\bar{K}_{ij} = k(x_i, x_j) = \phi^T(x_i) \phi(x_j) + \frac{\delta_{ij}}{C}.$$

在这里

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad (i, j = 1, \dots, N)$$

$k(\cdot, \cdot)$ 为满足 Mercer 条件的核函数. 常用的核函数为 Gaussian 核 $k(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / 2\gamma)$ 等. 通过解方程组(4)可以得到如下回归机:

$$f(x) = w^T x + b = \sum_{i=1}^N \bar{a}_i k(x_i, x) + b \quad (5)$$

由式(5)可知, 此回归机是非稀疏性的, 即几

乎每一个 $\bar{a}$ 都不等于 0, 这对预测是不利的, 因为预测时间和支持向量的数目成正比, 支持向量的数目越多, 模型的预测时间越长. 因而, 在对预测时间要求比较高的场合实现回归机的稀疏性是非常必要的. 在这里, 令

$$w = \sum_{i \in S} \alpha_i k(x_i, \cdot)$$

其中 S 为选择的支持向量的索引的集合, 把 w 代入到式(1)中, 可以得到约简最小二乘支持向量回归机的数学模型

$$\min_{\alpha, e} J(\alpha, e) = \frac{1}{2} \alpha^T K \alpha + \frac{C}{2} \sum_{i=1}^N e_i^2 \quad (6)$$

$$\text{s. t. } d_i = \sum_{j \in S} \alpha_j k(x_j, x_i) + b + e_i \\ (i = 1, \dots, N)$$

由式(6)可知, 和一般的实现最小二乘支持向量回归机解的稀疏性的方法不同之处在于: 约简最小二乘支持向量回归机不仅实现了稀疏性也同时考虑到了其他非支持向量产生的约束对目标函数的影响, 而一般实现稀疏性的方法仅仅用训练数据集的一个子集来进行最小二乘支持向量回归机的建模, 把其他非支持向量产生的约束对目标函数的影响给完全抛弃掉了. 由于丢掉了部分有用的信息, 所以一般来说在支持向量数目相同的情况下, 约简最小二乘支持向量回归机的泛化能力要好于实现稀疏性的一般方法^[10]. 从式(6)中消去 e , 可以得到如下表达式:

$$\min_{\alpha, b} J(b, \alpha) = \frac{1}{2} \alpha^T K \alpha + \frac{C}{2} \sum_{i=1}^N \left[d_i - \sum_{j \in S} \alpha_j k(x_j, x_i) - b \right]^2 \quad (7)$$

其中

$$K_{ij} = k(x_i, x_j), \quad i, j \in S$$

令

$$\frac{\partial J}{\partial b} = 0 \text{ 和 } \frac{\partial J}{\partial \alpha} = 0$$

可得如下线性方程组:

$$\begin{bmatrix} R & Z Z^T \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = Z d \quad (8)$$

其中

$$R = \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & K \end{bmatrix}, \quad Z = \begin{bmatrix} \mathbf{1}^T \\ K \end{bmatrix},$$

$$K_{ij} = k(x_i, x_j), \quad i \in S, \quad j = 1, \dots, N$$

如果 $R + Z Z^T$ 奇异, 可以对其进行正则化为

$R + ZZ^T + 10^{-10} I$. 通过解式(8), 可以得到一个稀疏的回归机为

$$f(\mathbf{x}) = \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b \quad (9)$$

到目前为止, 还有一个问题没有解决, 那就是怎么来确定支持向量的集合 S . 在这里本文采用文献[10]中的迭代方法来确定集合 S . 首先将式(7)展开为

$$\min_{\alpha, b} \left\{ J(\mathbf{b}, \mathbf{a}) = \begin{bmatrix} \mathbf{b} & \mathbf{a}^T \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0}^T \\ \mathbf{0} & K/C \end{bmatrix} + \begin{bmatrix} \mathbf{1}^T \\ K \end{bmatrix} \begin{bmatrix} \mathbf{1} & K^T \end{bmatrix} \begin{bmatrix} \mathbf{b} \\ \mathbf{a} \end{bmatrix} - 2 \begin{bmatrix} \mathbf{1}^T \\ K \end{bmatrix} \mathbf{d} \right\} \quad (10)$$

将式(8)展开

$$\left(\begin{bmatrix} \mathbf{0} & \mathbf{0}^T \\ \mathbf{0} & K/C \end{bmatrix} + \begin{bmatrix} \mathbf{1}^T \\ K \end{bmatrix} \begin{bmatrix} \mathbf{1} & K^T \end{bmatrix} \right) \begin{bmatrix} \mathbf{b} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{1}^T \\ K \end{bmatrix} \mathbf{d} \quad (11)$$

即

$$\left(\begin{bmatrix} \mathbf{0} & \mathbf{0}^T \\ \mathbf{0} & K/C \end{bmatrix} + \begin{bmatrix} N & \mathbf{1}^T K^T \\ K \mathbf{1} & K K^T \end{bmatrix} \right) \begin{bmatrix} \mathbf{b} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{1}^T \\ K \end{bmatrix} \mathbf{d} \quad (12)$$

如果在第 $(n+1)$ 步迭代时, 训练数据 $(\mathbf{x}_q, d_q)$ 被选中作为支持向量, 令

$$U^{n+1} = \left(\begin{bmatrix} \mathbf{0} & \mathbf{0}^T & \mathbf{0} \\ \mathbf{0} & K_n/C & K_q \\ \mathbf{0} & K_q^T & k_{qq} \end{bmatrix} + \begin{bmatrix} N & \mathbf{1}^T K^T & \mathbf{1}^T K_q \\ K \mathbf{1} & K K^T & K K_q \\ K_q^T \mathbf{1} & K_q^T K^T & K_q^T K_q \end{bmatrix} \right)^{-1} \quad (13)$$

其中

$$k_q = \begin{bmatrix} k(\mathbf{x}_q, \mathbf{x}_i) \\ \vdots \\ k(\mathbf{x}_q, \mathbf{x}_j) \end{bmatrix}, \quad (i, \dots, j \in S),$$

$$k_q = \begin{bmatrix} k(\mathbf{x}_q, \mathbf{x}_1) \\ \vdots \\ k(\mathbf{x}_q, \mathbf{x}_j) \end{bmatrix}, \quad (i = 1, \dots, N),$$

$$k_{qq} = k(\mathbf{x}_q, \mathbf{x}_q)$$

若在第 n 步时

$$U^n = \left(\begin{bmatrix} \mathbf{0} & \mathbf{0}^T \\ \mathbf{0} & K_n/C \end{bmatrix} + \begin{bmatrix} N & \mathbf{1}^T K^T \\ K \mathbf{1} & K K^T \end{bmatrix} \right)^{-1}$$

已知, 那么就可以利用迭代策略来解式(13)以达到避免对 U^{n+1} 进行直接求解减少计算量的目的.

文献[10]中利用 Sherman-Morrison 公式来迭代求解 U^{n+1} , 但这种迭代方法容易积累误差造成数值计算的不稳定性, 这对于设计高精度的推力估计器来说并不适合, 因而本文采用使数值计算更稳定的 Cholesky 分解^[11]来迭代求解 U^{n+1} . 具体方法如下:

令

$$(\mathbf{L}^n)^T \mathbf{L}^n = (\mathbf{U}^n)^{-1} \quad (14)$$

则

$$\mathbf{L}^{n+1} = \begin{bmatrix} \mathbf{L}^n & \mathbf{l}_1^{n+1} \\ \mathbf{0} & l_2^{n+1} \end{bmatrix} \quad (15)$$

其中

$$(\mathbf{L}^{n+1})^T \mathbf{L}^{n+1} = (\mathbf{U}^{n+1})^{-1}$$

$$\mathbf{l}_1^{n+1} = \left[(\mathbf{L}^n)^T \right]^{-1} \begin{bmatrix} \mathbf{1}^T K_q \\ K_q + K K_q \end{bmatrix}$$

$$l_2^{n+1} = \sqrt{K_q^T K_q + k_{qq} - (\mathbf{l}_1^{n+1})^T \mathbf{l}_1^{n+1}}$$

在得到 $\mathbf{L}^{n+1}$ 后, 用追赶法很容易得到

$$\begin{bmatrix} b^{n+1} \\ \alpha_s^{n+1} \\ \alpha_q^{n+1} \end{bmatrix} = (\mathbf{L}^{n+1})^{-1} \left\{ \begin{bmatrix} (\mathbf{L}^{n+1})^T \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ d_s \\ d_q \end{bmatrix} \right\} \quad (16)$$

其中

$$\mathbf{0} = \sum_i^N d_i, \quad \mathbf{d} = K \mathbf{d}$$

然而, 到现在为止, 用何种策略来确定 $(\mathbf{x}_q, d_q)$ 作为支持向量并没有给出, 现讨论如下:

若 α^n 和 b^n 已知, 则式(10)变为

$$J^n = \begin{bmatrix} b^n & (\alpha^n)^T \end{bmatrix} \cdot \left(\begin{bmatrix} \mathbf{0} & \mathbf{0}^T \\ \mathbf{0} & K_{SS}/C \end{bmatrix} + \begin{bmatrix} \mathbf{1}^T \\ K_S \end{bmatrix} \begin{bmatrix} \mathbf{1} & K_S^T \end{bmatrix} \right) \begin{bmatrix} b^n \\ \alpha^n \end{bmatrix} - 2 \begin{bmatrix} \mathbf{1}^T \\ K_S \end{bmatrix} \mathbf{d} \begin{bmatrix} b^n \\ \alpha^n \end{bmatrix} \quad (17)$$

如果固定 α^n 和 b^n , 则在第 $(n+1)$ 步, 式(10)可以简化为

$$J^{n+1} = \min_{\alpha_i} \left\{ \frac{1}{2} \left(\frac{k_{ii}}{C} + k_i^T k_i \right) \alpha_i^2 + r_i^n \alpha_i \right\},$$

$$i \in P = \{1, \dots, N\} - S \quad (18)$$

其中

$$r_i^n = \begin{cases} -\lambda_{d_i} & n = 0 \\ \left[k_i^T \mathbf{1} & k_i^T K^T + \frac{K_{Si}^T}{C} \right] \begin{bmatrix} b^n \\ \alpha^n \end{bmatrix} - \lambda_{d_i} & n > 0 \end{cases}$$

由式(18), J^{n+1} 很容易求得

$$J^{n+1} = - \frac{\left(r_i^n\right)^2}{2\left[\frac{k_{ii}}{C} + \mathbf{k}_i^T \mathbf{k}_i\right]} \tag{19}$$

并且

$$q = \arg \min_{i \in P} \left\{ - \frac{\left(r_i^n\right)^2}{2\left[\frac{k_{ii}}{C} + \mathbf{k}_i^T \mathbf{k}_i\right]} \right\} \tag{20}$$

由式(18)~式(20)可知,若达到预先指定的选择支持向量的数目或者 $\max(|r_p^n|) < \varepsilon$ (ε 为预先定义的一个正数来控制支持向量的数目),那么就可以停机. 在确定了 (x_q, d_q) 作为支持向量以后,若达到了停机条件就停机,否则令 $S = S \cup \{q\}$ 进行下一步的迭代求解. 详细的算法过程、收敛性证明和复杂度分析见文献[10]. 下面用迭代约简最小二乘支持向量回归机(recursive reduced least squares support vector regression, RRLSSVR)来设计全包线推力估计器.

2 全包线推力估计器设计

如果以航空发动机的可测参数为约简最小二乘支持向量回归机的输入量,以推力为输出量,那么就可以设计出一个推力估计器,从而实现航空发动机的直接推力控制. 为了提高推力估计器的精度,文献[12]提出了一种按高度对全包线进行划分然后在每一个子区域中设计子推力估计器的设计方案,但这种划分包线方法有一定的不合理之处,那就是即使在同一高度,开加力与不开加力推力相差也是很大的. 在引起相同推力估计误差的情况下,开加力时推力估计的相对误差较小而不开加力时推力估计的相对误差较大. 为了避免这种情况的发生,借鉴模式识别中的聚类思想,本文用聚类的方法将全包线的数据聚成几个类,然后用每一个类中的数据设计一个子推力估计器. 一般来说,在每一个类中,推力都是相近的,不会有太大的波动,这就为设计高精度的推力估计器打下了基础. 另外,每一个类中包含的包线区域可能不连通,但这没有关系,只要这些区域产生的推力相近就可以了. 那么用何种聚类方法来将全包线中的数据进行分类呢? 本文用模式识别中的 K -均值聚类方法将全包线内的数据根据 Euclidean 距离进行聚类[13]. K -均值聚类的结果就是产生 K 个类,每一个类有一个质心,然后计算当前状态与 K 个质心的 Euclidean 距离,将当前状态归属于离 K 个质心 Euclidean 距离最近的类中,

最后用该类中的子推力估计器估计出当前航空发动机所提供的推力,其结构如图 3 所示.

本文以某型混合排气涡扇发动机非线性部件级模型(如图 4 所示)为研究对象设计一个全包线

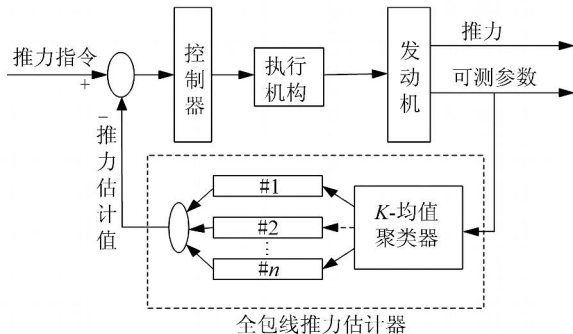


图 3 基于 K -均值聚类的推力估计器结构图
Fig.3 Block diagram of thrust estimator based on K -means clustering

推力估计器. 推力估计器的输入为 7 个变量^[12]: 高度 (H), 马赫数 (Ma), 外涵出口总压 (p_{16}), 喷管喉道总温 (T_8), 主供油量 (W_{fb}), 加力供油量 (W_{fa}), 发动机温比 (E_{tr}). 在包线内采集了覆盖全包线的 5 059 个训练数据(等高度间隔等马赫数间隔), 然后用 K -均值聚类器将全部训练数据聚成 10 个类, 然后用每一个类中的数据设计一个子推力估计器. 在用 RRLSSVR 设计子推力估计器的过程当中, Gaussian 函数

$$k(x_i, x_j) = \exp\left[-\frac{\|x_i - x_j\|^2}{2\gamma^2}\right]$$

被选择为核函数,核参数 γ 和正则化参数 C 可以通过交叉验证的方法取得. 为了便于比较,定义了一个性能参数如下:

$$\text{相对误差} = \left| \frac{\text{估计值} - \text{实际值}}{\text{实际值}} \right|$$

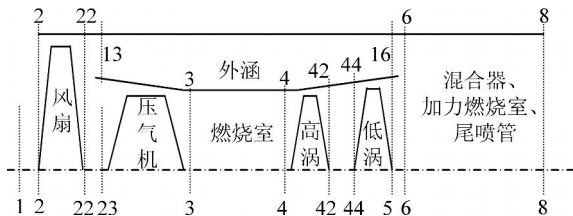


图 4 发动机部件及截面示意图
Fig.4 Illustration of turbofan engine with mixing exhaust

关于每一个子推力估计器的具体情况,见表 1. 在每一个子推力估计器都设计好以后,又在包线内采集了覆盖全包线的 1 009 个数据(等高

度间隔等马赫数间隔)进行测试.测试的效果如图 5 所示.

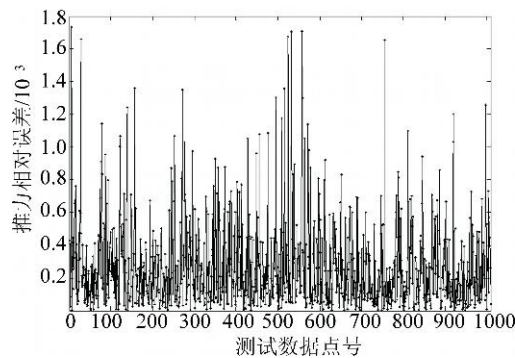


图 5 测试数据测试的结果

Fig.5 Experimental result of testing data set

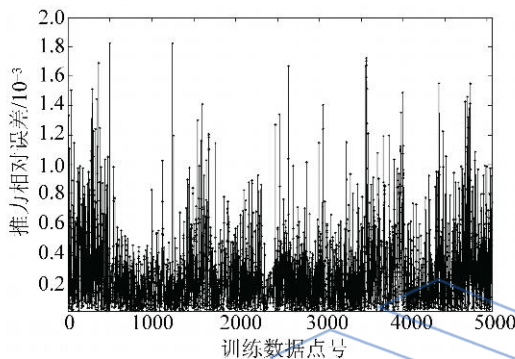


图 6 训练数据测试的结果

Fig.6 Experimental result of training data set

然后,又用学习过的训练数据进行测试,测试的效果如图 6 所示.

为了说明用约简最小二乘支持向量回归机设计推力估计器的优越性,本文又用标准最小二乘支持向量回归机(LSSVR)来设计推力估计器,现将这两种方法设计推力估计器的统计结果列于表 2 当中.

由表 2 可知,虽然用标准 LSSVR 和 RRLSSVR 方法设计的推力估计器都能使相对误差的最大值小于 2 %,能满足直接推力控制对精度的需要(相对误差的最大值不大于 5 %),但是用标准 LSSVR 设计的推力估计器的预测时间为 10.66 ms(本文的时间是在 Matlab 7.1 ,CPU 为 AMD3200 + (2.01 G) ,内存为 512 M 的环境下测得的),这远远大于用 RRLSSVR 设计的推力估计器的预测时间 1.05 ms.一般来说,要把推力估计器的预测时间控制在 2 ms 之内,预测时间越短越好,这样才能给控制器和执行机构留下充分的时间来控制发动机.另外,对于用 RRLSSVR 设计的推力估计器来说,训练数据估计推力的标准差和均值均小于测试数据的标准差和均值,这主要是因为训练数据推力估计器学习过而测试数据推力估计器没有学习过的缘故.虽然测试数据相对误差的最大值小于训练数据相对误差的最大值,但这只是个别现象,主要是看均值和方差.最

表 1 关于各个子推力估计器的具体情况

Table 1 Details about each subestimator

子推力估计器编号	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8	# 9	# 10
训练数据	513	926	495	548	505	500	514	390	381	287
测试数据	112	185	101	97	100	98	103	81	75	57
支持向量数目	45	75	40	55	50	40	40	40	95	35
C	2^{20}	2^{20}	2^{20}	2^{20}	2^{20}	2^{20}	2^{20}	2^{20}	2^{20}	2^{20}
γ	0.6	0.4	0.4	0.4	0.5	0.8	0.6	0.6	0.3	0.5

表 2 推力估计器预测结果的信息统计

Table 2 Information statistics of thrust estimation results

数据集	方法	相对误差的 最大值/ 10^{-3}	相对误差的 中间值/ 10^{-5}	相对误差的 标准差/ 10^{-5}	相对误差的 平均值/ 10^{-5}	预测时间/ ms
测试	标准 LSSVR	1.86	4.86	14.1	8.54	10.66
数据集	RRLSSVR	1.73	19.9	26.7	27.1	1.05
训练	标准 LSSVR	1.48	4.21	8.70	6.78	10.66
数据集	RRLSSVR	1.82	18.0	23.6	24.5	1.05

后,将本文基于 K -均值聚类和 RRLSSVR 设计的推力估计器和文献[12]中设计的推力估计器相比较,相对误差的最大值 1.82 % 小于文献[12]中的 1.93 %,预测时间 1.05 ms 小于文献[12]中的 1.37 ms,这在估计精度和实时性上都占有一定的优势。

3 结 论

为了实现直接推力控制,需要设计一个高精度的推力估计器。由于在航空发动机全包线范围内推力的变化范围比较大,而又要使估计出来的推力精度比较高,这就给设计推力估计器带来了一定的困难。因此,本文提出了一种基 K -均值聚类和约简最小二乘支持向量回归机的推力估计器设计方法,基于 K -均值聚类将全包线范围内的数据进行聚类,然后在每一个类中用 RRLSSVR 设计一个子推力估计器。最后,仿真实验表明,本文提出的推力估计设计方案能满足直接推力控制的需求,并和现有的推力估计器设计方案比较起来也占有一定的优势。另外,本文提出的设计方案不仅能用来设计推力估计器同时也能用在其他方面进行预测。

参考文献:

- [1] Gastineau Z, Happawana G. Robust model-based control for jet engines[R]. AIAA-98-3752, 1998.
- [2] Maggiore M, Ordonez R, Passino K M, et al. Estimator design in jet engine applications[J]. Engineering Applications of Artificial Intelligence, 2003, 16(7-8): 579-593.
- [3] Vapnik V N. The nature of statistical theory[M]. New York: Springer-Verlag, 1995.
- [4] Cristiannini N, Shawe-Taylor J. An introduction to support vector machines[M]. Cambridge, UK: Cambridge University Press, 2000.
- [5] Suykens J A K, Vandewalle J. Least squares support vector machine classifiers[J]. Neural Process Letter, 1999, 9(3): 293-300.
- [6] Suykens J A K, De Brabanter J, Lukas L, et al. Weighted least squares support vector machines: robustness and sparse approximation[J]. Neurocomputing, 2002, 48(1-4): 85-105.
- [7] Jiao L, Bo L, Wang L. Fast sparse approximation for least squares support vector machine[J]. IEEE Transactions on Neural Networks, 2007, 18(3): 685-697.
- [8] 赵永平, 孙健国. 基于线性无关度的稀疏最小二乘支持向量回归机[J]. 中国图象图形学报, 2009, 14(6): 1136-1140. ZHAO Yongping, SUN Jianguo. Sparse least squares support vector regression machine based on the scale of linear independency[J]. Journal of Image and Graphics, 2009, 14(6): 1136-1140. (in Chinese)
- [9] 赵永平, 孙健国. 一种快速稀疏最小二乘支持向量回归机[J]. 控制与决策, 2008, 23(12): 1347-1352. ZHAO Yongping, SUN Jianguo. Fast method for sparse least squares support vector regression machine[J]. Control and Decision, 2008, 23(12): 1347-1352. (in Chinese)
- [10] ZHAO Yongping, SUN Jianguo. Recursive reduced least squares support vector regression[J]. Pattern Recognition, 2009, 42(5): 837-842.
- [11] Zhang X D. Matrix analysis and applications[M]. New York: Springer, 2004.
- [12] 赵永平, 孙健国. 最小二乘支持向量回归机在发动机推力估计中的应用[J]. 航空动力学报, 2009, 24(6): 1420-1425. ZHAO Yongping, SUN Jianguo. Aeroengine thrust estimation using least squares support vector regression machine[J]. Journal of Aerospace Power, 2009, 24(6): 1420-1425. (in Chinese)
- [13] Duda R O, Hart P E, Stork D G. Pattern classification[M]. UK: John Wiley & Sons Inc., 2001.