



(12)发明专利申请

(10)申请公布号 CN 110516391 A  
(43)申请公布日 2019. 11. 29

(21)申请号 201910823121.9

(22)申请日 2019.09.02

(71)申请人 南京航空航天大学

地址 210000 江苏省南京市江宁区将军大道29号

(72)发明人 郑前钢 刘子赫 汪勇 陈浩颖  
项德威 金崇文 胡忠志 张海波

(74)专利代理机构 北京德崇智捷知识产权代理有限公司 11467

代理人 杨楠

(51)Int.Cl.

G06F 17/50(2006.01)

G06N 3/04(2006.01)

G06N 3/08(2006.01)

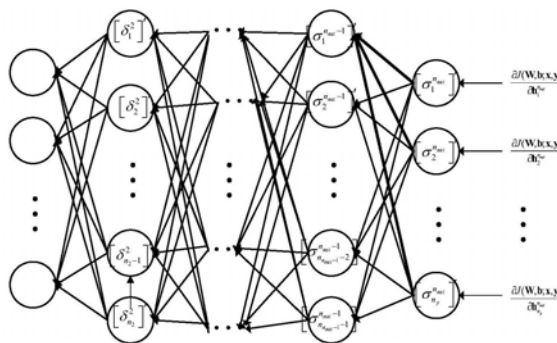
权利要求书1页 说明书7页 附图12页

(54)发明名称

一种基于神经网络的航空发动机动态模型建模方法

(57)摘要

本发明公开了一种基于神经网络的航空发动机动态模型建模方法,利用神经网络构建航空发动机动态模型,所述神经网络为最小批量下降神经网络,其使用最小批量下降法进行神经网络的训练。相比于现有技术,本发明提出的建模方法不仅具有较高的测试精度,而且还有更少的数据存储量、计算复杂度和测试时间,这几个性能指标是能否作为机载模型的重要指标,因而本方法更适合作为航空发机自适应机载动态模型。



1. 一种基于神经网络的航空发动机动态模型建模方法,利用神经网络构建航空发动机动态模型,其特征在于,所述神经网络为最小批量下降神经网络,其使用最小批量下降法进行神经网络的训练。

2. 如权利要求1所述航空发动机动态模型建模方法,其特征在于,所述建模方法包括以下步骤:

步骤一、获得航空发动机动态模型的训练数据;

步骤二、确定神经网络结构;

步骤三、使用最小批量下降法对神经网络进行前传播;

步骤四、使用反向传播计算神经网络梯度,并更新梯度;

步骤五、判断神经网络是否收敛,是则输出模型,否则继续迭代,返回步骤三。

3. 如权利要求2所述航空发动机动态模型建模方法,其特征在于,通过发动机试车实验或/和发动机非线性部件级模型得到所述航空发动机动态模型的训练数据。

4. 如权利要求1~3任一项所述航空发动机动态模型建模方法,其特征在于,将飞行高度、马赫数、燃油流量、尾喷管喉道面积的当前及历史信息,发动机推力、风扇转子转速、压气机转子转速、风扇喘振裕度、压气机喘振裕度、高压涡轮进口温度的历史信息及各个部件退化量作为模型输入,将当前时刻的发动机推力、风扇转子转速、压气机转子转速、风扇喘振裕度、压气机喘振裕度、高压涡轮进口温度作为模型输出。

## 一种基于神经网络的航空发动机动态模型建模方法

### 技术领域

[0001] 本发明涉及航空发动机控制技术领域,尤其涉及一种航空发动机动态模型建模方法。

### 背景技术

[0002] 航空发动机机载模型是实现诸多先进控制技术的基础,其中机载模型分为稳态模型和动态模型。目前航空发动机机载模型建模的方法有很多,根据方法种类主要可以分为两种,一种是具有解析表达式的建模方法,该这类方法具有较强几何意义,保留对象的连续可微,即使在数据较少时也具有很强的泛化能力,常见的有分段线性、多维线性回归、多项式拟合、非线性拟合等;另一种办法是基于人工智能算法,如BP神经网络、ELM、支持向量机等方法,这类方法是直接基于数据建模方法,事先可以不知道被拟合对象的函数形式,具有较强的应用范围,然而该类方法容易出现毛刺和出现过拟合,现流行的支持向量机因为克服传统神经网络易于陷入局部极值而受到广泛关注,然而随着维数的增加和精度要求的提高,必须增加样本采集量,对于大样本数据,支持向量机训练困难,难以应用于大样本数据情况;而且要提高模型的实时性,就必须增加支持向量机的稀疏性,而支持向量机稀疏性问题一直未得到良好的解决。

[0003] 对于发动机动态机载模型,由于要模型保证发动机动态特性,一般需要保留历史输入,模型维数剧烈增加,而且发动机动态特性的非线性程度远大于稳态时候的非线性程度,这些使得单纯形样条不在适用。在上个世纪末,由于神经网络理论上能拟合任何函数而被广泛关注,然而传统的BP神经网络由于易于陷入局部极值,而使得其研究陷入低潮,很多学者转入支持向量回归机在发动机建模上的应用,然而支持向量机不适于大数据复杂训练,这使得它在大包线建模时往往需要建立好几个子模型,这增加数据存储量。最近十几年,随着神经网络关键技术的突破,其应用再次进入学者的视野,传统神经网络训练数据方法采用批量梯度下降法(BGD, batchgradientdescent),该方法在更新神经网络权重时需要计算所有的训练数据,这种办法使得传统神经网络难以应用于大数据情况下,而大样本数据训练是解决神经网络易于陷入局部极值、提高其泛能力和模型精度的良好办法。为此有学者提出随机梯度下降法(SGD, stochastic gradientdescent),该方法在训练网络参数时只需计算一个数据点,该方法适用于大数据,然而,数据往往存在噪声,因而该方法对噪声比较敏感。

### 发明内容

[0004] 本发明所要解决的技术问题在于克服现有技术不足,提出一种基于神经网络的航空发动机动态模型建模方法,该方法使用最小批量下降法进行神经网络的训练,使得神经网络能够适用于大样本数据训练,提高网络泛化能力。

[0005] 本发明具体采用以下技术方案解决上述技术问题:

[0006] 一种基于神经网络的航空发动机动态模型建模方法,利用神经网络构建航空发动

机动态模型,所述神经网络为最小批量下降神经网络,其使用最小批量下降法进行神经网络的训练。

[0007] 优选地,所述建模方法包括以下步骤:

[0008] 步骤一、获得航空发动机动态模型的训练数据;

[0009] 步骤二、确定神经网络结构;

[0010] 步骤三、使用最小批量下降法对神经网络进行前传播;

[0011] 步骤四、使用反向传播计算神经网络梯度,并更新梯度;

[0012] 步骤五、判断神经网络是否收敛,是则输出模型,否则继续迭代,返回步骤三。

[0013] 优选地,通过发动机试车实验或/和发动机非线性部件级模型得到所述航空发动机动态模型的训练数据。

[0014] 优选地,将飞行高度、马赫数、燃油流量、尾喷管喉道面积的当前及历史信息,发动机推力、风扇转子转速、压气机转子转速、风扇喘振裕度、压气机喘振裕度、高压涡轮进口温度的历史信息及各个部件退化量作为模型输入,将当前时刻的发动机推力、风扇转子转速、压气机转子转速、风扇喘振裕度、压气机喘振裕度、高压涡轮进口温度作为模型输出。

[0015] 相比现有技术,本发明技术方案具有以下有益效果:

[0016] 本发明在构建航空发动机动态模型时使用最小批量下降法进行神经网络的训练,使得所提出的建模方法能应用大样本数据,从而提高建模精度和泛化能力。

## 附图说明

[0017] 图1为神经元结构图;

[0018] 图2为神经网络结构图;

[0019] 图3为神经网络反向传播原理图;

[0020] 图4为训练误差与测试误差;

[0021] 图5为MGD NN动态模型的训练相对误差;

[0022] 图6a~图6f为MGD NN和MRR-LSSVR的测试相对误差(H(10km~11km),Ma(1.4~1.5),PLA(65°~75°));

[0023] 图7a~图7o为MGD NN自适应动态模型全包测试。

## 具体实施方式

[0024] 本发明主要针对传统航空发动机动态过程建模方法难以应用于大样本数据情况,提出基于神经网络的航空发动机动态模型建模方法,该方法使用最小批量下降法,使得所提出的建模方法能应用大样本数据,从而提高建模精度和泛化能力。

[0025] 所述建模方法包括以下步骤:

[0026] 步骤一、获得航空发动机动态模型的训练数据;

[0027] 步骤二、确定神经网络结构;

[0028] 步骤三、使用最小批量下降法对神经网络进行前传播;

[0029] 步骤四、使用反向传播计算神经网络梯度,并更新梯度;

[0030] 步骤五、判断神经网络是否收敛,是则输出模型,否则继续迭代,返回步骤三。

[0031] 所述航空发动机动态模型的训练数据可通过发动机试车实验或/和发动机非线性

部件级模型得到,由于试车实验成本高昂,目前普遍通过发动机非线性部件级模型得到发动机动态数据。

[0032] 为了便于公众理解,下面结合附图来对本发明的技术方案进行详细说明:

[0033] 神经网络的基础单元是神经元,是多输入单输出的处理单元,其结构图如图1所示,其中  $W_{i,j}^l$  是  $l+1$  隐含层神经元  $i$  与  $l$  隐含层神经元  $j$  的链接强度,它模拟生物神经元之间的链接,称为神经网络的权重,  $b_i^l$  为神经元的偏执,  $h_j^l$  和  $h_j^{l+1}$  分别表示第  $l$  隐含层神经元  $j$  和第  $l+1$  隐含层神经元  $j$  的输出,  $n_l$  为第  $l$  隐含层的隐含层节点个数,则神经元输出为:

$$[0034] \quad a_i^{l+1} = \sum_{j=1}^{n_l} W_{i,j}^l h_j^l + b_i^l \quad (1)$$

$$[0035] \quad h_i^{l+1} = \sigma(a_i^{l+1}) \quad (2)$$

[0036] 其中  $\sigma(\cdot)$  为激活函数,它反应了神经元的映射能力,在M-P模型中,激活函数为:

$$[0037] \quad \sigma(a_i^l) = \begin{cases} 1, & a_i^l \geq 0 \\ 0, & a_i^l < 0 \end{cases} \quad (3)$$

[0038] 从上式可以看出,该激活函数在零附近不可导,而且拟合能力弱,为此很多学者提出很多其它激活函数,如Sigmoid、Tansig、Relu和PReLU等激活函数。这些激活函数,它们各有不同的优缺点,适用于不同的问题,ReLU以及改进型PReLU在求解梯度的时间较快,节省训练时间,每层网络神经元被激活个数更少,使网络具有稀疏性,保证网络的泛化能力,目前普遍应用图像分类问题。然而,对于发动机这种强非线性回归问题,如果继续使用ReLU和PReLU,需要增加很多的隐含层层数,这极大影响网络的实时性,而Sigmoid和Tanh函数其非线性拟合能力强,只需较少的隐含层层数和隐含层节点就能拟合复杂的非线性问题。

[0039] 神经网络是由很多神经元组成,其结构图如2所示,传统的神经网络一般是一个三层神经网络。

[0040] 神经网络最左边称为输入层,最右边称为输出层,中间层称为隐藏层,将输入到输出的计算称为前向传播,令  $h^1 = x$ ,在给定第  $l$  层的输出值  $h^l$  后,第  $l+1$  的输出值  $h^{l+1}$  有

$$[0041] \quad a^{l+1} = W^{l+1} h^l + b^{l+1} \quad (4)$$

$$[0042] \quad h^{l+1} = \sigma(a^{l+1}) \quad (5)$$

[0043] 以上步骤称为正向传播,是神经网络的计算过程,为了获得网络参数  $W$  和  $b$ ,需要对网络进行学习,学习方式分为监督学习、无监督学习和增强学习,监督学习根据输入样本和期望输出值,通过网络通过训练,使得网络能够最佳地表达输入到输出的关系;无监督学习中并没有显示的教师,只需给出样本,无需给定输出样本的值,系统会对输入样本自动地学习;增强学习为给定一个输入样本,计算出它的输出类别,把它与已知输出做比较,根据差别改善模型性能。本发明把神经网络用于回归问题,因此采用监督学习,监督学习最常采用的为反向传导算法(BP, Back-Propagation),在接下对其进行介绍。

[0044] 神经网络训练方法可以一般采用梯度下降法、共轭梯度法、拟牛顿法等,其中梯度下降法以其计算量小,收敛速度快而被广泛应用,每次迭代如下:

$$[0045] \quad W_{ij}^l = W_{ij}^l + \eta \nabla W_{ij}^l \quad (6)$$

$$[0046] \quad b_i^l = b_i^l + \eta \nabla b_i^l \quad (7)$$

[0047] 其中 $\nabla W_{ij}^l$ 和 $\nabla b_i^l$ 是第1层的神经网络权值和偏置的梯度, $\eta$ 是学习率。

[0048] 反向传播主要是求得神经网络权重和偏置的梯度,其传播原理如图3所示,其中 $[\sigma^l]'$ 为对隐含层节点激活函数的求导, $l = n_{\text{net}} - 1, n_{\text{net}} - 2, \dots, 2, n_{\text{net}}$ 表示为神经网络的层数,假设目标函数对 $\mathbf{a}^{n_{\text{net}}}$ 的偏导 $\delta^{n_{\text{net}}}$ 为:

$$[0049] \quad \delta^{n_{\text{net}}} = \frac{\partial J(\mathbf{W}, \mathbf{b}; \mathbf{x}, \mathbf{y})}{\partial \mathbf{a}^{n_{\text{net}}}} = \frac{\partial J(\mathbf{W}, \mathbf{b}; \mathbf{x}, \mathbf{y})}{\partial \mathbf{h}^{n_{\text{net}}}} \otimes \frac{\partial \mathbf{h}^{n_{\text{net}}}}{\partial \mathbf{a}^{n_{\text{net}}}} = \frac{\partial J(\mathbf{W}, \mathbf{b}; \mathbf{x}, \mathbf{y})}{\partial \mathbf{h}^{n_{\text{net}}}} \otimes [\boldsymbol{\sigma}^{n_{\text{net}}}]' \quad (8)$$

[0050] 其中 $\otimes$ 表示为hadamard乘积,也称为Schur乘积,即 $\mathbf{x} \otimes \mathbf{y} = [x_1 y_1, x_2 y_2, \dots, x_n y_n]^T$ ,  $J(\mathbf{W}, \mathbf{b}; \mathbf{x}, \mathbf{y})$ 为神经网络的训练损失函数,为网络训练目标,常见的有Softmax,交叉熵(Cross Entropy Loss)和欧拉损失(Euclidean Loss),前两种对分类问题效果较好,本发明主要针对回归问题,因而采用后一种损失函数。

[0051] 对于 $l = n_{\text{net}} - 1, n_{\text{net}} - 2, \dots, 2$ 层节点的导数 $\delta^l$ 有:

$$[0052] \quad \delta^l = [\mathbf{W}^l]^T \delta^{l+1} \otimes [\boldsymbol{\sigma}^l]' \quad (9)$$

[0053] 其中 $[\sigma^l]'$ 为对激励函数的导数,因此有如下迭代公式:

$$[0054] \quad \nabla W_{ij}^l = \frac{\partial J(\mathbf{W}, \mathbf{b}; \mathbf{x}, \mathbf{y})}{\partial W_{ij}^l} = \mathbf{h}_j^l \otimes \delta_j^{l+1} \quad (10)$$

$$[0055] \quad \nabla b_i^l = \frac{\partial J(\mathbf{W}, \mathbf{b}; \mathbf{x}, \mathbf{y})}{\partial b_i^l} = \delta_i^{l+1} \quad (11)$$

[0056] 至此,通过反向传播,建立了传统神经网络,传统的神经网络一般为三层神经网络,如何增加神经网络层数,使网络变得更深是过去三十年神经网络专家重点研究的领域,主要因为增加网络深度可以增加网络的表达能力,然而深度带来神经网络训练难度急剧增加,主要表现在梯度消失和梯度溢出。

[0057] 计算梯度的另一种办法是对权重和偏执进行扰动,得出损失函数的偏差,在用偏差除以扰动量,得到梯度,即为:

$$[0058] \quad \nabla W_{ij}^l = \frac{J(\mathbf{W}^l + \varepsilon \mathbf{E}_{ij}^l) - J(\mathbf{W}^l)}{\varepsilon} \quad (12)$$

$$[0059] \quad \nabla b_i^l = \frac{J(b^l + \varepsilon \mathbf{e}_i^l) - J(b^l)}{\varepsilon} \quad (13)$$

[0060] 其中 $\mathbf{E}_{ij}^l$ 为第i行j列不为1,其它元素为0的矩阵, $\mathbf{e}_i^l$ 为第i行为1,其它为0的向量, $\varepsilon$ 为值很小的数。这个办法看过去非常有效,容易实现,也容易理解,遗憾的是,这个方法计算复杂度高,假设权重和偏置个数为10000,要计算出 $J(\mathbf{W}^l + \varepsilon \mathbf{E}_{ij}^l)$ 和 $J(b^l + \varepsilon \mathbf{e}_i^l)$ ,则需要计算损失函数10000次,加上计算 $J(\mathbf{W}^l)$ 和 $J(b^l)$ ,总共需要计算损失函数10001次,因此此方法计算量大。而反向传播算法只需前向传播和反向传播各算一次,计算复杂度远远小于该方法,因而该方法为目前主要办法。

[0061] 神经网络往往容易过拟合,所谓过拟合如图4所示,随着迭代次数的增加,训练误差虽然一直减小,然而训练误差随着迭代次数不但不会减小,反而增加。目前避免过拟合最常用的办法有增加训练数据和正则化技术。而增加训练样本的不仅能非常有效地避免过拟合,而且可以有效提高训练精度,然而样本数据的增加,会带来训练的计算复杂度,因此有

必要寻求高效率的训练算法。

[0062] 常见下降法有批量梯度下降法,随机梯度下降法和最小批量梯度下降法,给定训练样本集  $(x_i, y_i)$ ,  $i=1, 2, \dots, N$ ,  $N$ 为训练样本个数。本发明采用最小批量梯度下降法 (MGD, Min-batch Gradient Descent), 它是把训练集随机分为  $M$ 组, 每组具有  $N_b$ 个训练集, 其第  $i$ 组的损失函数如下:

$$[0063] \quad J_i(\mathbf{W}, \mathbf{b}; \mathbf{x}, \mathbf{y}) = \min_{\mathbf{W}, \mathbf{b}} \sum_{j=1}^{N_b} \frac{1}{2} \|\mathbf{h}(\mathbf{x}_{b_i, j}) - \mathbf{y}_{b_i, j}\|^2 \quad (14)$$

[0064] 其中  $i=1, 2, \dots, M$ ,  $\sum b_i = N$ ,  $\bigcup_i \bigcup_j \mathbf{x}_{b_i, j} = \bigcup_i \mathbf{x}_i$ ,  $\bigcup_i \bigcup_j \mathbf{y}_{b_i, j} = \bigcup_i \mathbf{y}_i$ , 它的损失函数是计算训练集中的一个子集, 因此它的计算量比BSG少, 而且梯度方向优化SGD。

[0065] 现流行的支持向量机因为其良好的泛化能力和理论依据而被广泛应用应用到发动机建模中, 其中多输入多输出约简迭代最小二次支持向量机 (MRR-LSSVR) 算法能够将约简技术及迭代策略与标准最小二乘支持向量回归机结合起来, 并考虑多个输出变量对选择支持向量的综合影响, 以对多输出目标贡献最大为筛选准则, 选择更少且更优的支持向量来解决多输出问题, 可以有效地缩短预测时间和增强稀疏性, 因此, 为了验证本发明提出算法的有效性, 进行基于最小批量下降法神经网络 (MGD-NN) 的航空发动机动态模型的仿真实验, 并与MRR-LSSVR方法作比较。

[0066] 为了使非参数实时模型更好地保留原系统动态特性, 采用非线性自回归滑动平均 (NARMA-non-linear autoregressive, moving-average) 模型, 模型训练需充分利用当前及历史时刻的信息, 最终准确地预测出发动机状态。本发明在  $k$ 时刻, 将  $H, Ma, W_{fb}, A_8$  当前及历史信息、 $F, N_f, N_c, S_{mf}, S_{mc}, T_4$  历史信息、及各个部件退化量, 如风扇流量退化量  $\Delta_{W_{a_{2c}}}$ 、压气机流量退化量  $\Delta_{W_{a_{25c}}}$ 、燃烧室燃烧效率退化量  $\Delta_{\eta_{comb}}$ 、高压涡轮效率退化量  $\Delta_{\eta_{Hturb}}$ 、低压涡轮效率退化量  $\Delta_{\eta_{Lturb}}$  作为模型输入, 将当前时刻  $F, N_f, N_c, S_{mf}, S_{mc}, T_4$  作为模型输出, 构建内嵌式预测模型如下:

$$[0067] \quad \begin{cases} \mathbf{y} = \mathbf{f}_{MGDNN}(\mathbf{x}) \\ \mathbf{x} = [H(k), H(k-1), \dots, H(k-m_1); Ma(k), Ma(k-1), \dots, Ma(k-m_2); \\ W_{fb}(k), W_{fb}(k-1), \dots, W_{fb}(k-m_3); A_8(k), A_8(k-1), \dots, A_8(k-m_4); \\ \Delta_{W_{a_{2c}}}(k), \Delta_{W_{a_{2c}}}(k-1), \dots, \Delta_{W_{a_{2c}}}(k-m_5); \Delta_{W_{a_{25c}}}(k), \Delta_{W_{a_{25c}}}(k-1), \dots, \Delta_{W_{a_{25c}}}(k-m_6); \\ \Delta_{\eta_{comb}}(k), \Delta_{\eta_{comb}}(k-1), \dots, \Delta_{\eta_{comb}}(k-m_7); \Delta_{\eta_{Hturb}}(k), \Delta_{\eta_{Hturb}}(k-1), \dots, \Delta_{\eta_{Hturb}}(k-m_8); \\ \Delta_{\eta_{Lturb}}(k), \Delta_{\eta_{Lturb}}(k-1), \dots, \Delta_{\eta_{Lturb}}(k-m_9); \\ F_{in}(k-1), F_{in}(k-2), \dots, F_{in}(k-m_{10}); N_f(k-1), N_f(k-2), \dots, N_f(k-m_{11}); \\ N_c(k-1), N_c(k-2), \dots, N_c(k-m_{12}); S_{mf}(k-1), S_{mf}(k-2), \dots, S_{mf}(k-m_{13}); \\ S_{mc}(k-1), S_{mc}(k-2), \dots, S_{mc}(k-m_{14}); T_4(k-1), T_4(k-2), \dots, T_4(k-m_{15})] \\ \mathbf{y} = [F_{in}(k), N_f(k), N_c(k), S_{mf}(k), S_{mc}(k), T_4(k)]^T \end{cases} \quad (15)$$

[0068] 预测模型既要保证合适的动静态精度又要使各输入参数尽可能少, 因而  $m_1, m_2, \dots, m_{15}$  的确定至关重要, 一般通过可以调试确定, 这里确定为 2。

[0069] 本发明的仿真环境都是在 Windows 7 Ultimate with Service Pack 1 (x64) 的操作系统, CPU 为主频为 3.30GHz 的 Intel (R) Core (TM) i5-4590, 内存为 8G, 具体运行软件为 MATLAB2016b 或者 VC++6.0。

[0070] 本发明选定在超音速巡航包线进行仿真,既H从9km到13km、Ma从1.2到1.6、PLA从65°到75°,退化量 $\Delta_{W_{a2c}}$ ,  $\Delta_{W_{a2sc}}$ ,  $\Delta_{\eta_{comb}}$ ,  $\Delta_{\eta_{Hhub}}$ 和 $\Delta_{\eta_{Lhub}}$ 的变化范围为从0%到5%。对部件级模型进行充分激励得到1587597个训练数据。

[0071] 由于MRR-LSSVR对如此大的数据无法进行训练,因而需要对该包线进行分区,本发明在H=9.5km、10.5km、11.5km、或12.5km和Ma=1.25、1.35、1.45、或1.55等共 $4 \times 4 = 16$ 个点建立16个MRR-LSSVR动态子模型。

[0072] MGD NN的训练相对误差如图5所示,很显然 $N_f$ 和 $N_c$ 的相对误差都在8%以下, $F_{in}$ 的相对误差小于2%,另外三个的相对误差少于3%,可以看出训练相对误差在整个训练集都比较小。图6给出了基于MRR-LSSVR和MGD NN而分别建立起来的发动机动态模型的在H(10km~11km)、Ma(1.4~1.5)、PLA(65°~75°)的测试相对误差,图中黑色线为MRR-LSSVR的误差,红色线为MGD NN的误差,从图中可以看出,MGD NN的测试精度大于MRR-LSSVR,其中MRR-LSSVR动态模型的 $F_{in}$ 、 $N_f$ 、 $N_c$ 、 $T_4$ 、 $S_{mf}$ 、 $S_{mc}$ 的最大测试相对误差分别为1.4%、0.5%、0.8%、1.2%、2%、3.5%,而MGD NN动态模型的 $F_{in}$ 、 $N_f$ 、 $N_c$ 、 $T_4$ 、 $S_{mf}$ 、 $S_{mc}$ 的最大测试相对误差分别为0.3%、0.05%、0.1%、0.3%、0.9%、2%,可以看出相比于MRR-LSSVR,MGD NN的 $F_{in}$ 、 $N_f$ 、 $N_c$ 、 $T_4$ 、 $S_{mf}$ 、 $S_{mc}$ 的最大误差分别降低了3.67、9.0、7.0、3.0、1.2和0.75倍。这主要因为MGD NN能够应用大样本数据建模,越多的训练数据使得MGD NN模型具有更好的泛化能力。MRR-LSSVR对于大样本数据训练困难,从而导致其泛化能力受限。

[0073] 图7给出MGD NN自适应航空发动机动态模型在超音速巡航的测试仿真,其中 $F_{in}$ 、 $N_f$ 、 $N_c$ 、 $T_4$ 、 $S_{mf}$ 、 $S_{mc}$ 的值都已经归一化,图7(a)~(i)分别给出了H、Ma、 $W_{fb}$ 、 $A_8$ 、 $\Delta_{W_{a2c}}$ 、 $\Delta_{W_{a2sc}}$ 、 $\Delta_{\eta_{comb}}$ 、 $\Delta_{\eta_{Hhub}}$ 和 $\Delta_{\eta_{Lhub}}$ 的变化曲线,图7(j)~(o)发动机参数 $T_4$ 、 $F_{in}$ 、 $N_f$ 、 $N_c$ 、 $S_{mf}$ 和 $S_{mc}$ 的响应曲线,从图中可看出,在超音速巡航包线内,MGD NN动态模型可以良好地预测发动机各个参数,从而验证了给模型在该包线内具有良好精度。

[0074] 表1 MGD NN和MRR-LSSVR比较

[0075]

	数据存储量 (double)	模型复杂度	平均测试时间
MRR-LSSVR	960,096	61,006	1.8ms
MGD NN	4,276	8,476	0.31ms

[0076] 表1给出了MGD NN模型和MRR-LSSVR模型的数据存在量、模型复杂度和平均测试时间,通过调试 $m_1, m_2, \dots, m_{15}$ 确定为3, MRR-LSSVR的支持量机个数为1000, MGD-NN的网络结构为[54, 70, 6]。每次更新网络权重的最小批量个数为3000个,学习率为 $\alpha = 0.001$ ,并随迭代次数增加而减小。

[0077] MRR-LSSVR的数据存储量为960,096((1000支持向量机个数 $\times$ 54维+(1000支持向量机个数 $\times$ 6输出)权重+1偏置 $\times$ 6输出) $\times$ 16子模型)。

[0078] MGD NN的数据存储量为4,276((54维数+1偏置) $\times$ 70隐含层个数+(70隐含层个数+1偏置) $\times$ 6输出)。

[0079] MRR-LSSVR的计算复杂度为12,0000((54 $\times$ 1000减法+53 $\times$ 1000加法+1000除法)+(1000乘法+999加法+1偏置) $\times$ 6输出)。



[0080] MGD NN的计算复杂度为8,476 ((54乘法+54加法+1偏置) × 70隐含层个数+ (70乘法+70加法+1偏置) × 6输出)。

[0081] 两个程序运行环境一样, MRR-LSSVR和MGD NN的平均测试时间分别为1.8毫秒和0.31毫秒。

[0082] 相比于目前流行的MRR-LSSVR, 本发明提出的建模方法不仅具有较高的测试精度, 而且还有更少的数据存储量、计算复杂度和测试时间, 这几个性能指标是能否作为机载模型的重要指标, 因而本方法更适合作为航空发动机自适应机载动态模型。

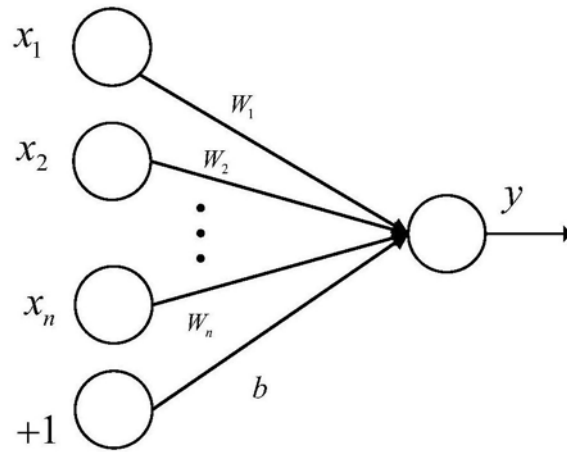


图1

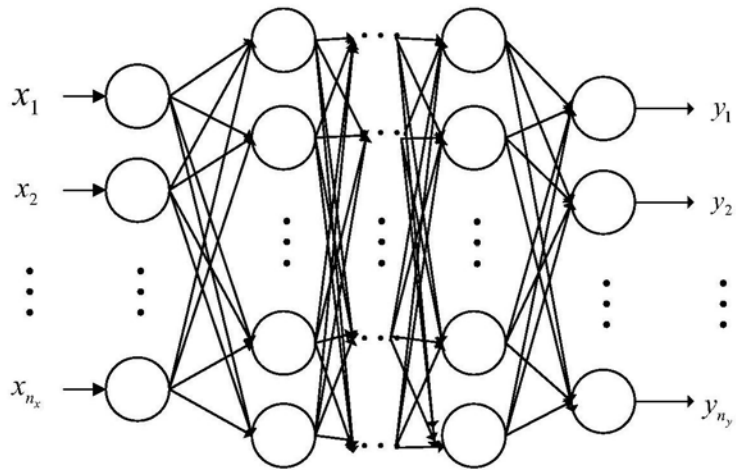


图2

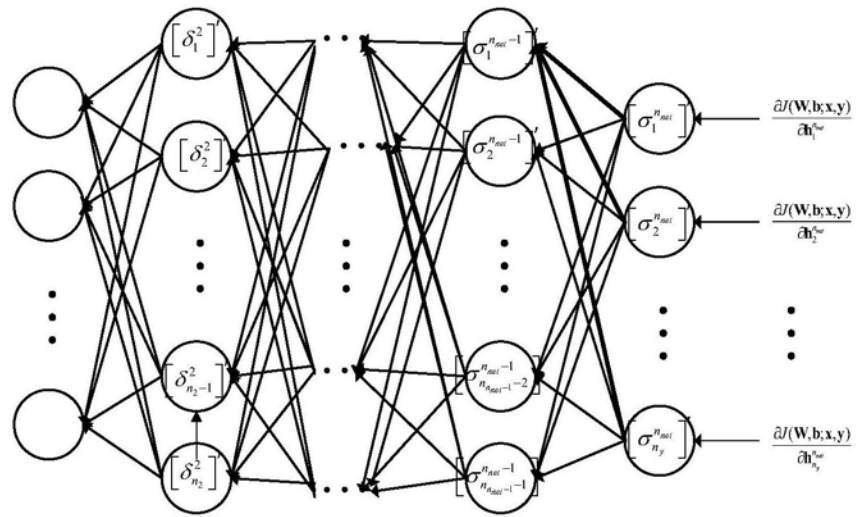


图3

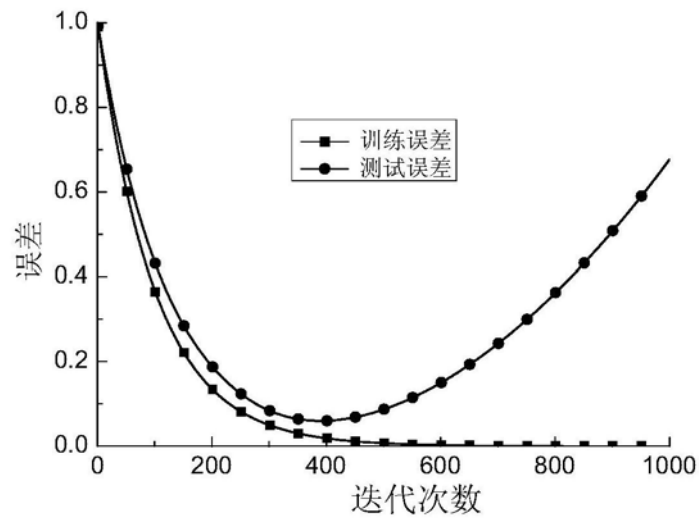


图4

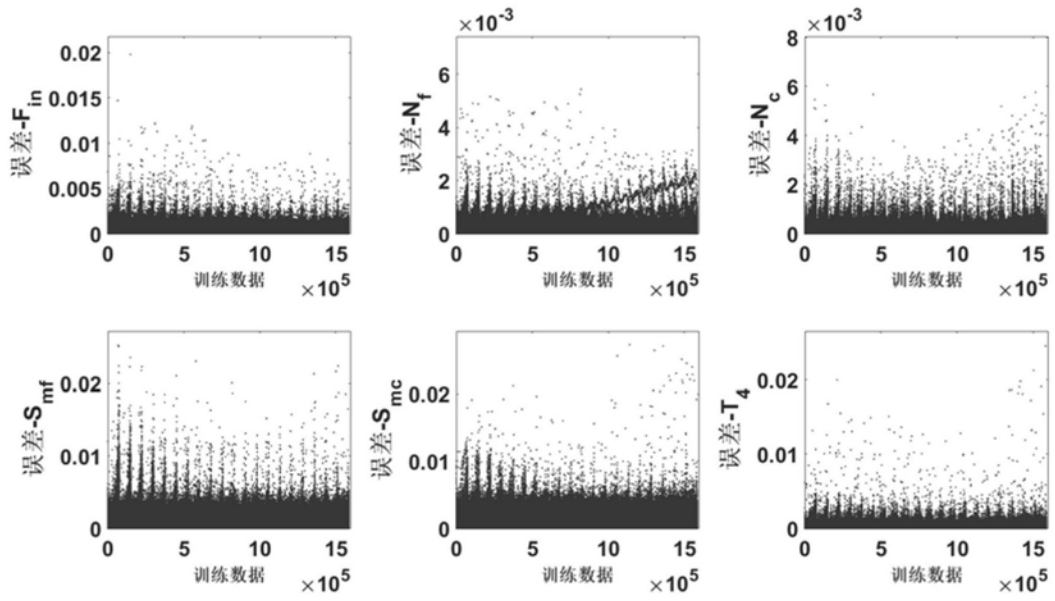


图5

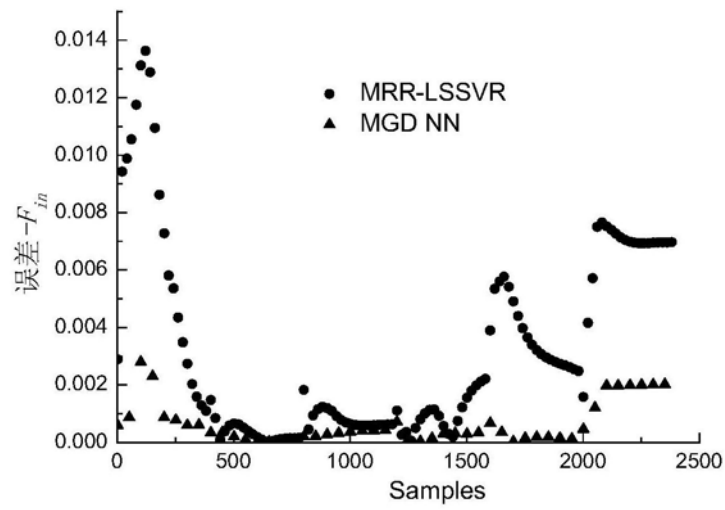


图6a

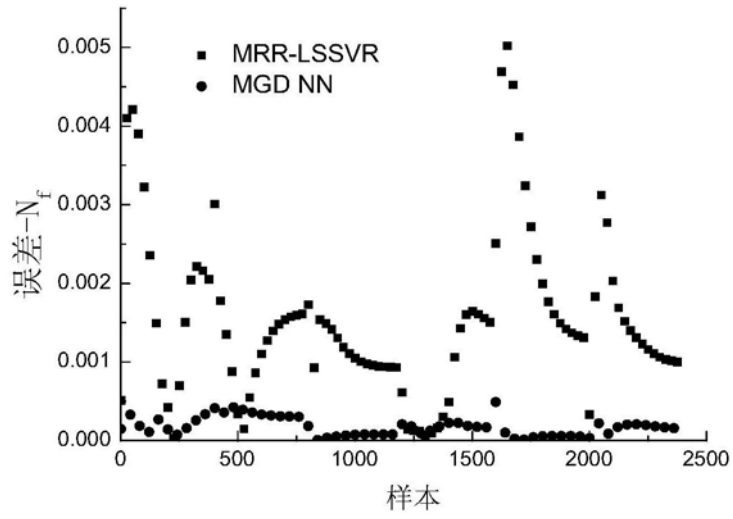


图6b

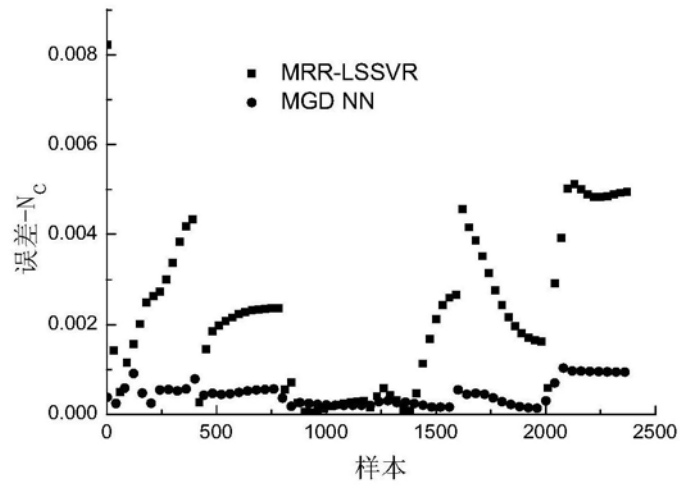


图6c

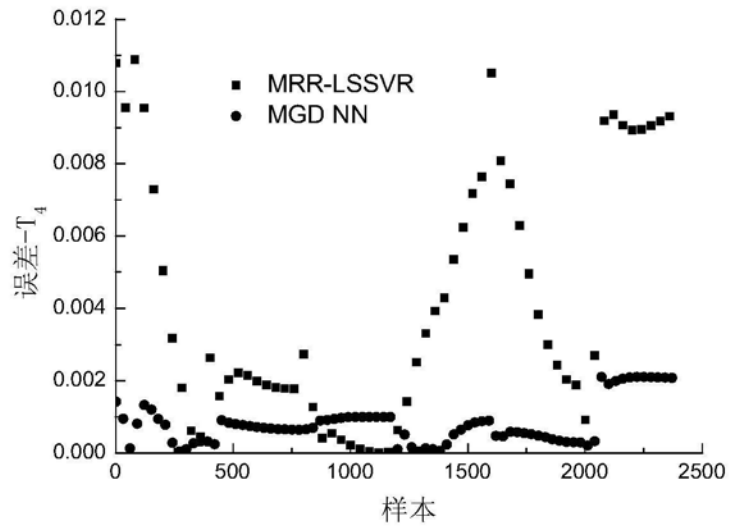


图6d

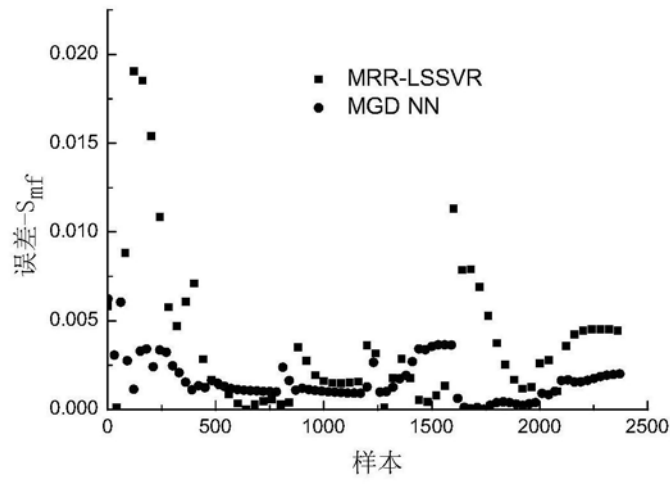


图6e

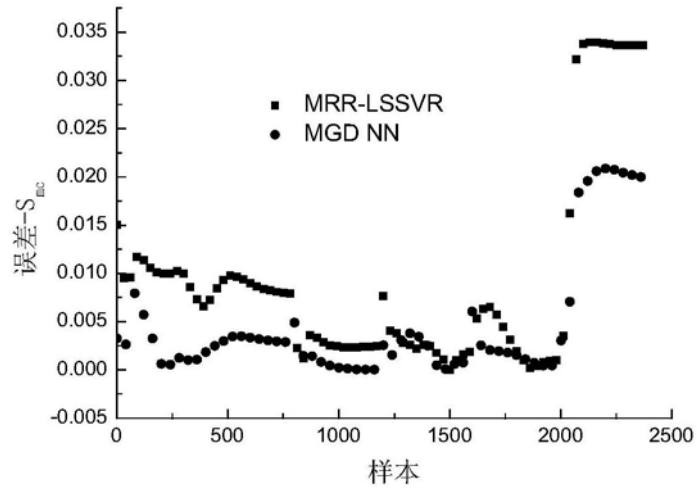


图6f

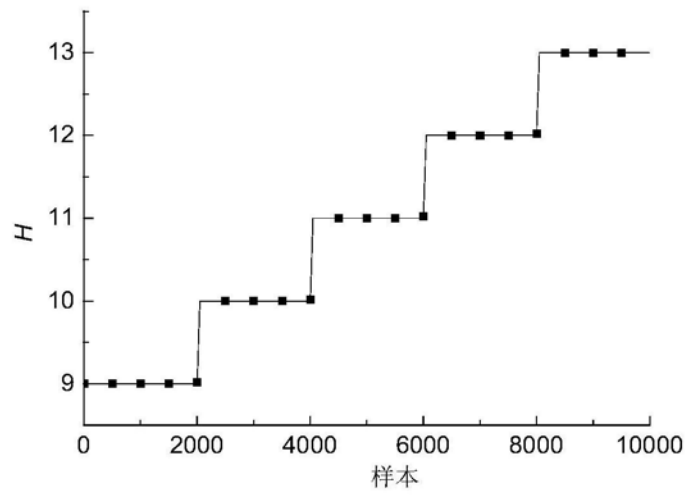


图7a

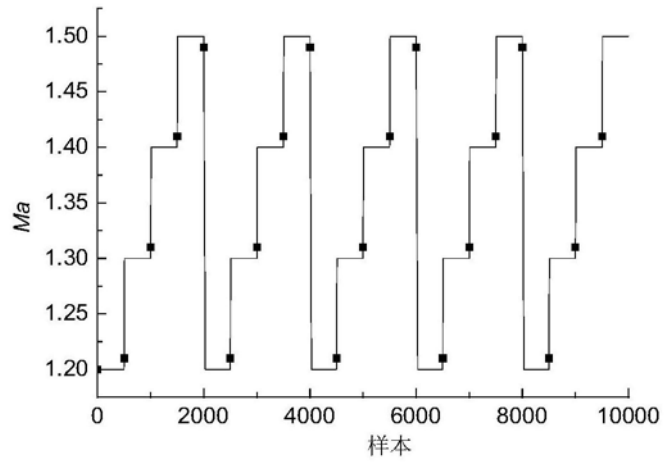


图7b

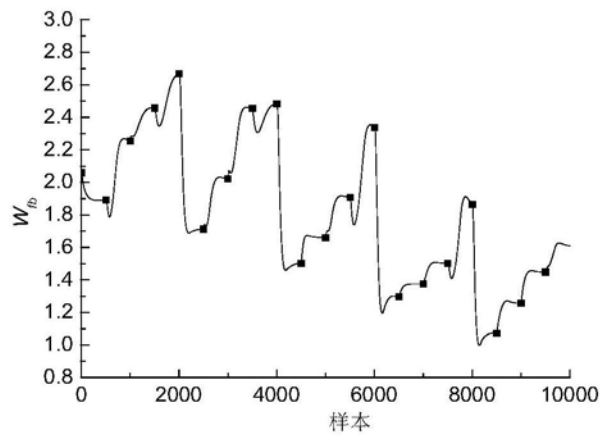


图7c

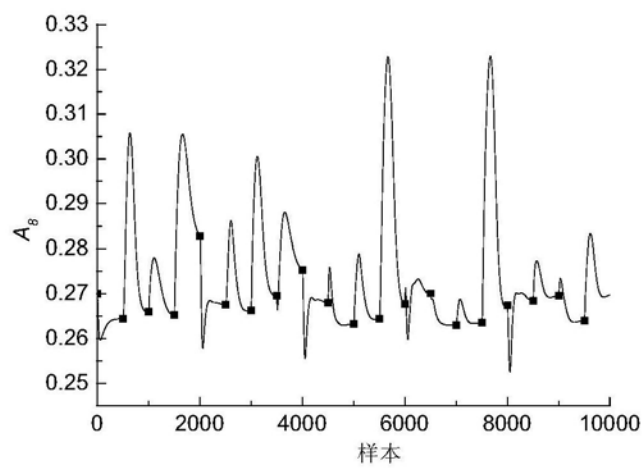


图7d



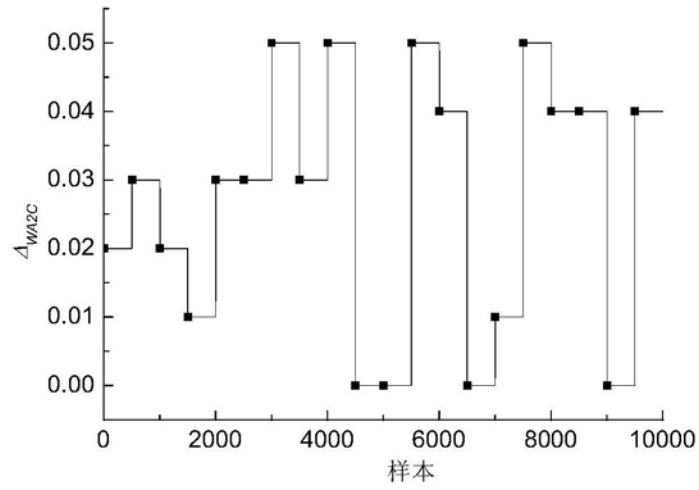


图7e

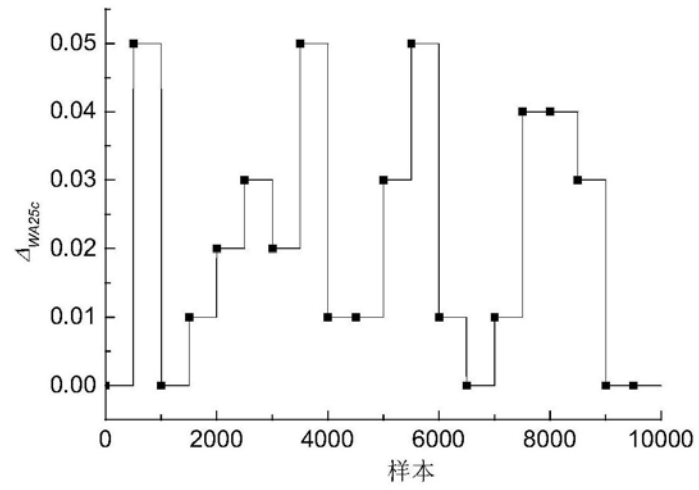


图7f

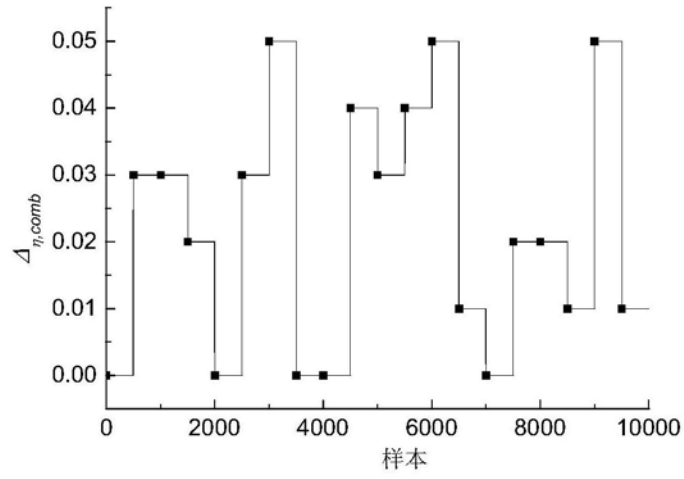


图7g

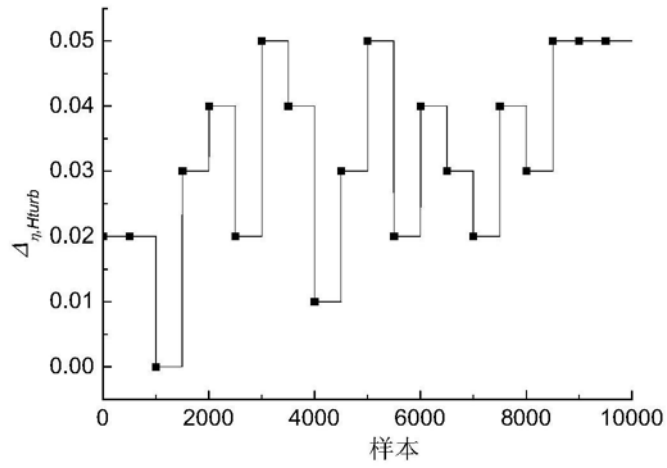


图7h

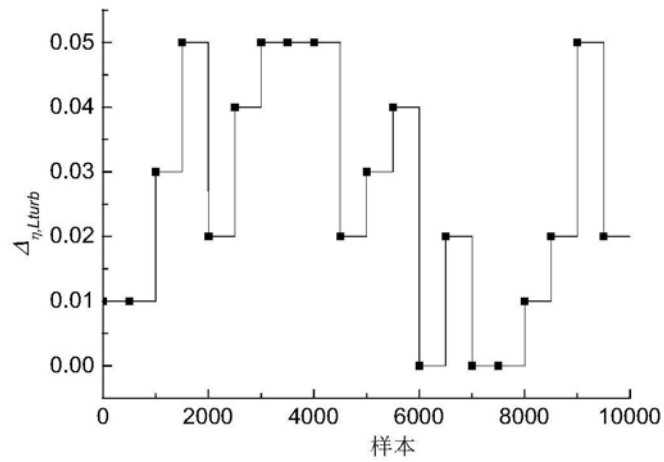


图7i

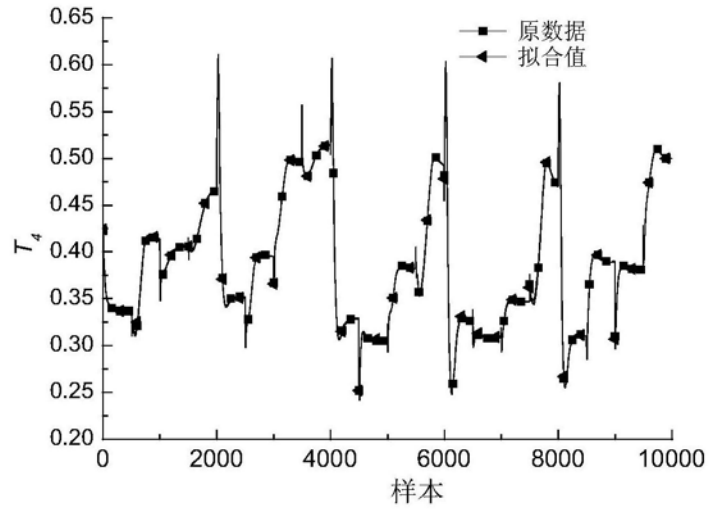


图7j

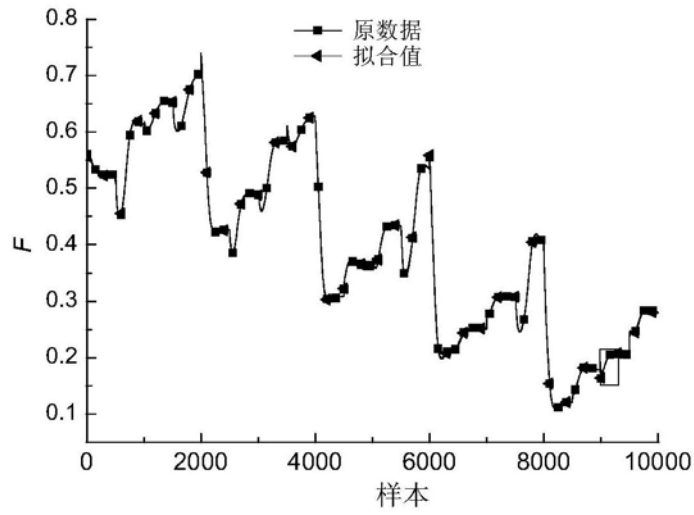


图7k

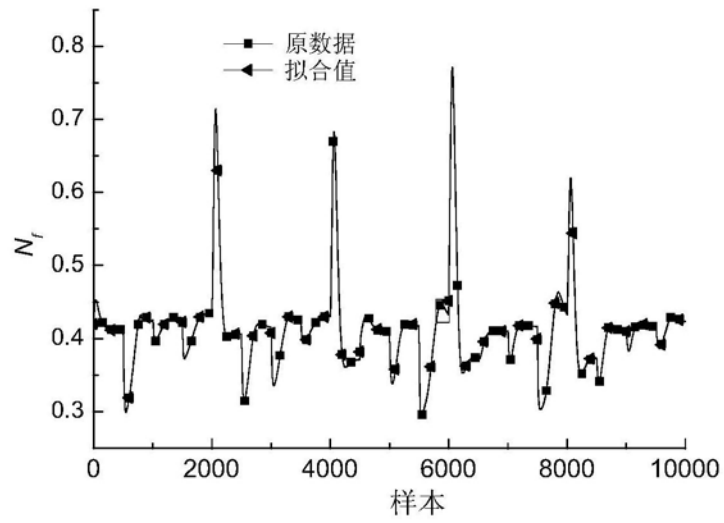


图71

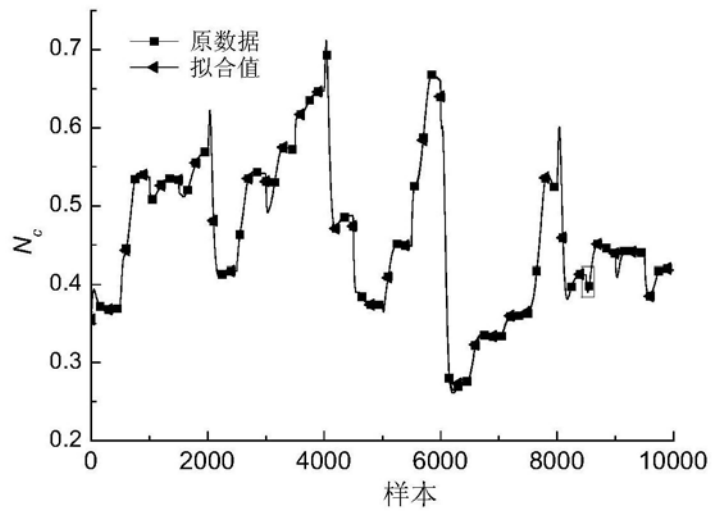


图7m

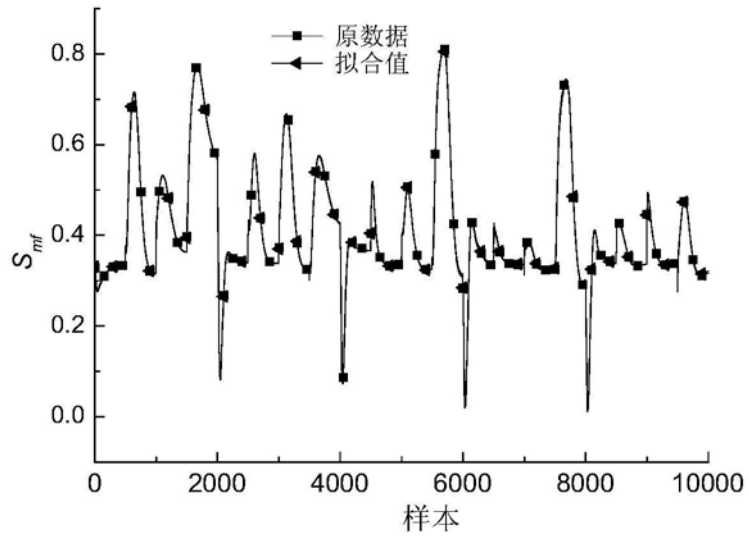


图7n

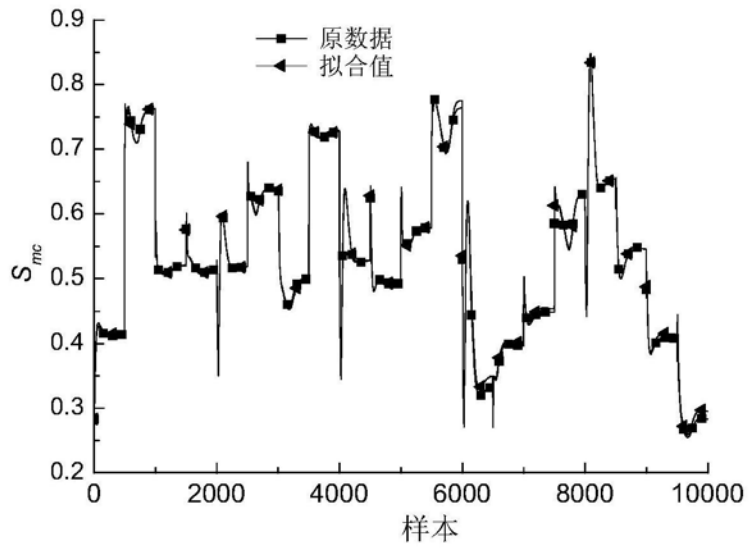


图7o