

# 基于 ADE-ELM 的涡轴发动机建模方法

焦 洋, 李秋红, 朱正琛, 廖光煌  
(南京航空航天大学 能源与动力学院  
江苏省航空动力系统重点实验室, 南京 210016)

**摘 要:** 提出了基于自适应微分进化-极端学习机(ADE-ELM)求解平衡方程的高精度涡轴发动机实时部件级模型建立方法. 基于牛顿-拉夫逊(N-R)迭代模型,以迭代计算前模型平衡方程残差为输入,迭代收敛后平衡方程猜值修正量为输出,训练极端学习机,并采用自适应微分进化(ADE)算法优化极端学习机(ELM)参数,提高猜值修正量映射精度. ADE 算法中采用 sigmoid 型自适应缩放因子,提高了微分进化算法的寻优能力. 在涡轴发动机不同飞行状态下的测试结果表明,以 N-R 迭代算法模型为基准,基于 ADE-ELM 的发动机模型,最大建模误差约为一次通过算法的 1/3,运算耗时约为一次通过算法的 1/3,验证了算法的有效性.

**关 键 词:** 涡轴发动机; 建模; 平衡方程; 极端学习机; 微分进化算法

**中图分类号:** V233.7      **文献标志码:** A

## Turbo-shaft engine modeling method based on ADE-ELM

JIAO Yang, LI Qiu-hong, ZHU Zheng-chen, LIAO Guang-huang  
(Jiangsu Province Key Laboratory of Aerospace Power System,  
College of Energy and Power Engineering,  
Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

**Abstract:** A modeling method based on adaptive differential evolution-extreme learning machine (ADE-ELM) was proposed to solve the balance equations for turbo-shaft engine models. The model achieved high accuracy and real time properties. The extreme learning machine (ELM) was trained by the data got from the Newton-Raphson (N-R) iteration model. The inputs were the residuals of balance equations before iteration, and the outputs were the adjustments of guess values after iteration. The parameters of the ELM were optimized by adaptive differential evolution (ADE) algorithm to enhance the mapping accuracy of the guess values adjustments. The sigmoid type adaptive scaling factor was adopted by ADE algorithm to improve its optimization ability. Taking the N-R iteration model as a criterion, the simulation results of ADE-ELM model under different flying states show that, the maximum modeling error is about 1/3 that of the single iteration per time step, and the time consumption is about 1/3 that of the single iteration per time step. The modeling method is valid.

**Key words:** turbo-shaft engines; modeling; balance equations; extreme learning machine; differential evolution algorithm

航空发动机的数学模型在其研制、控制、健康管理、性能寻优中起着非常重要的作用. 以数学模型代替真实发动机开展数值仿真或半物理仿真试验,是降低发动机成本、提高试验安全性的有效手段. 因而发动机数学模型一直是世界各航空强国的研究重点. 早在 20 世纪 50 年代,美国就开始发

动机非线性数学模型的研究,并建立了单轴涡喷以及涡桨发动机模型,此后发动机数学模型的研究一直持续展开.

总体来说,发动机的数学模型可以分为部件级数学模型<sup>[1]</sup>、人工智能映射模型<sup>[2-3]</sup>、线性模型<sup>[4]</sup>.其中部件级数学模型沿发动机气路流程建立发动机各个部件的气动热力学模型,通过求解部件间平衡方程的形式,使得各部件匹配工作.而人工智能映射模型,采用神经网络或支持向量机等智能算法,依据发动机某些参数对其他参数进行映射.线性模型可以表现为状态空间形式或传递函数形式.智能映射模型具有结构简单、实时性好的特点,但其只能用于专门的目的,能够映射的参数有限,且由于发动机的工作包线大、参数多变,很难兼顾到每个动态过程,模型精度受到限制.而线性模型作为小偏差模型,只在其设计点附近具有一定的精度,因而通常用于控制器和 Kalman 滤波器等的的设计过程.能够全面反映发动机稳态和动态工作过程中各个截面参数变化的只有部件级数学模型,且其能够对不可测参数进行计算,一直在发动机数学模型中占据着主导地位,在仿真验证、控制方法、健康管理等<sup>[5]</sup>的研究中得到广泛的应用.但由于其计算过程复杂,实时性差,在机载应用中受到制约.

在部件级模型得到广泛应用以来,提高其实时性的研究一直得到关注.常用的航空发动机平衡方程的求解方法有:牛顿-拉夫逊法(Newton-Raphson, N-R)迭代算法和 Broyden 拟牛顿法等. N-R 法由于需要重复调用发动机部件模型计算雅可比矩阵,而严重影响了模型的实时性;Broyden 拟牛顿法由于采用递推方式更新校正函数,避免了部件模型的重复调用,实时性较好,但其收敛性较差.为提高模型的实时性,有研究提出用一次通过(single iteration per time step, SIPTS)算法代替 N-R 迭代法,以 N-R 法一次计算结果作为共同方程的近似解,其在损失一定的建模精度条件下,大幅提高了模型的运算速度,为目前大多数的发动机部件级模型所采用<sup>[6-7]</sup>.

本文在研究发动机平衡方程的基础上,提出了一种基于极端学习机(ELM)<sup>[8-10]</sup>的发动机平衡方程求解方法,并将其应用于涡轴发动机部件级模型的动态平衡方程求解中,提高了模型的精度和实时性.

# 1 涡轴发动机动态平衡方程

在已知飞行条件、燃油流量的条件下,根据双

转子涡轴发动机工作过程中需满足的流量连续、压力平衡条件进行分析,通常选择动态和稳态情况下均应满足 3 个平衡方程,记  $\epsilon_i, i=1,2,3$  为动态平衡方程残差,3 个动态平衡方程为

1) 燃气涡轮进口流量连续方程

$$\phi_1(x) = (W_{41c} - Q_{41c})/Q_{41c} = \epsilon_1 \quad (1)$$

其中  $W_{41c}$  代表由燃气涡轮部件特性曲线计算出的当前工作条件下燃气涡轮进口相似流量,  $Q_{41c}$  代表按气路流程计算获得的从燃气涡轮导向器进入燃气涡轮的相似流量,在同一截面上,两者应满足流量连续的条件.

2) 动力涡轮进口流量连续方程

$$\phi_2(x) = (W_{45c} - Q_{45cs})/Q_{45c} = \epsilon_2 \quad (2)$$

其中  $W_{45c}$  代表由动力涡轮部件特性曲线计算出的当前工作条件下动力涡轮进口相似流量,  $Q_{45c}$  代表按气路流程计算获得的从高压涡轮导向器进入高压涡轮转子的相似流量,两者同样应满足连续条件.

3) 尾喷管出口静压平衡方程

$$\phi_3(x) = (p_{c7} - p_7)/p_7 = \epsilon_3 \quad (3)$$

其中  $p_{c7}$  代表喷管出口气流静压,  $p_7$  代表环境大气压力.

为求解这 3 个平衡方程,选取猜值  $v = [v_1 \ v_2 \ v_3]^T = [Z_c \ Z_g \ Z_p]^T$ ,  $Z_c$  为压气机压比系数,  $Z_g$  为燃气涡轮落压比系数,  $Z_p$  为动力涡轮落压比系数.其中,  $Z_c$  压比系数定义为

$$Z_c = \frac{\pi_c - \pi_{cb}}{\pi_{cs} - \pi_{cb}} \quad (4)$$

其中  $\pi_c$  为当前转速下相应部件的压比,  $\pi_{cb}$  为其堵塞边界压比,  $\pi_{cs}$  为其喘振边界压比.  $Z_g, Z_p$  定义与  $Z_c$  相似,均通过当前压比与转速线边界点压比进行计算.

采用 N-R 方法求解平衡方程时,第  $k+1$  步的猜值由式(5)得到

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_{k+1} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_k - \begin{bmatrix} \frac{\partial \phi_1}{\partial v_1} & \frac{\partial \phi_1}{\partial v_2} & \frac{\partial \phi_1}{\partial v_3} \\ \frac{\partial \phi_2}{\partial v_1} & \frac{\partial \phi_2}{\partial v_2} & \frac{\partial \phi_2}{\partial v_3} \\ \frac{\partial \phi_3}{\partial v_1} & \frac{\partial \phi_3}{\partial v_2} & \frac{\partial \phi_3}{\partial v_3} \end{bmatrix}_k^{-1} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix}_k \quad (5)$$

为了提高模型精度,通常采用中间差分法来计算雅可比矩阵中的偏导数,即

$$\left. \frac{\partial \phi_j}{\partial v_i} \right|_k = \frac{\phi_j(v_i + \delta v_i) - \phi_j(v_i - \delta v_i)}{2\delta v_i} \quad (6)$$

式中  $\delta v_i$  为在  $v_i$  点附近的扰动量,通常设为一极小的常量.则在计算偏导数的过程中,对于每个猜值需分别正向和负向做小扰动,则需 2 次调用各部件模型,以获得平衡方程残差来计算偏导数,3 个猜值则需调用 6 次部件模型,然后采用高斯法求雅可比矩阵的逆,根据式(5)完成猜值的一次修正.为了满足平衡方程收敛条件(通常设为  $|\epsilon_i| < 10^{-5}, i=1,2,3$ ,需要进行多次的猜值修正,才能得出平衡方程收敛后的猜值修正量  $\Delta v_j = v_j(k+1) - v_j(k), j=1,2,3, v_j(k)$  代表在第  $k$  个时刻的平衡方程猜值,  $v_j(k+1)$  代表在第  $k+1$  个时刻的平衡方程猜值.这严重影响了模型的实时性,为提高模型实时性,大多数模型采用一次通过算法来求解平衡方程,不对平衡方程收敛条件进行检验,只对猜值进行一次修正即进入下一步的计算,这样做在损失一定精度的前提下,节约了猜值的迭代修正时间,提高了模型的实时性.

即便如此,在一次动态计算过程中也需 6 次的部件模型调用.并且,动态精度的损失同样制约了其机载应用.为此本文提出基于极端学习机映射的方法来求解平衡方程,不需多次调用部件模型求雅可比矩阵和进行矩阵求逆,在大幅提高模型实时性的同时可以提高模型的精度.

## 2 基于极端学习机的发动机平衡方程求解

求解发动机共同工作发动机的目的是为了获得发动机下一时刻平衡方程中的猜值,使方程满足收敛条件.在 N-R 法中,猜值的修正量是通过平衡方程残差计算雅可比矩阵得到.由于发动机的非线性,使得平衡方程残差与猜值修正量之间也具有非线性的关系,而极端学习机方法在解决非线性问题方面具有突出的优势,为此本文提出以平衡方程残差来映射猜值修正量来代替式(5)中雅可比矩阵计算部分,获得猜值修正量  $\Delta v_j, j=1,2,3$ .以压气机压比系数  $Z_c$  为例,其极端学

习机映射结构如图 1 所示.

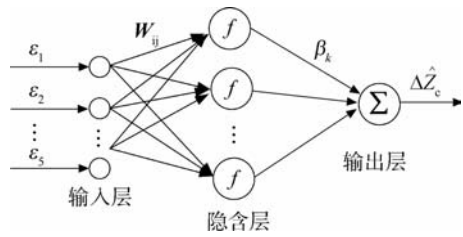


图 1 基于极端学习机的猜值预测结构图

Fig. 1 Guess value prediction diagram based on ELM

图 1 中  $\Delta \hat{Z}_c$  代表对压气机压比系数猜值修正量的估计值.考虑到发动机共同工作要求,映射网络中加入  $\epsilon_4 \epsilon_5$  代表动态工作过程中,部件间功率之差,可以反映出转子的速度变化.其中

$$\begin{aligned} \epsilon_4 &= \frac{P_{gt}}{P_c/\eta_{mc} + P_{ext}} - 1 \\ \epsilon_5 &= \frac{P_{pt}}{P_l/\eta_{ml}} - 1 \end{aligned} \tag{7}$$

其中  $P_{gt}, P_c, P_{pt}, P_l$  分别表示燃气涡轮、压气机、动力涡轮、旋翼的功率,  $P_{ext}$  表示燃气涡轮的抽功量.  $\eta_{mc}, \eta_{ml}$  分别表示燃气涡轮转子和动力涡轮转子的机械效率.

在 ELM 算法中,给定数据集  $\mathcal{X} = \{x_i, t_i\}^N$ , 其中输入数  $x_i = [x_{i1}, \dots, x_{in}]^T \in \mathbf{R}^n$ , 期望输出数据  $t_i = [t_{i1}, \dots, t_{im}]^T \in \mathbf{R}^m, n, m$  分别为输入层以及输出层节点个数.并设隐含层激励函数为  $h(x)$ , 则含有  $n_h$  个隐含层节点的 ELM 的回归数学模型为

$$\sum_{j=1}^{n_h} h(w_j x_i + b_j) \beta_j = t_i \tag{8}$$

其中  $w_j = [w_{1j}, w_{2j}, \dots, w_{nj}]$  是连接隐含层第  $j$  节点与输入层各节点的权值,  $b_j$  是隐含层第  $j$  个节点的偏置,  $\beta_j = [\beta_{j1}, \dots, \beta_{jm}]^T$  是连接接隐含层第  $j$  节点和输出层各节点的权值.

本文采取多输入单输出的形式来建立映射模型,输入量为 5 个平衡方程残差,输出量为单个猜值修正量,即  $n=5, m=1$ .

令

$$H(w, b, x) = \begin{bmatrix} h(w_1 x_1 + b_1) & h(w_2 x_1 + b_2) & \cdots & h(w_{n_h} x_1 + b_{n_h}) \\ h(w_1 x_2 + b_1) & h(w_2 x_2 + b_2) & \cdots & h(w_{n_h} x_2 + b_{n_h}) \\ \vdots & \vdots & & \vdots \\ h(w_1 x_N + b_1) & h(w_2 x_N + b_2) & \cdots & h(w_{n_h} x_N + b_{n_h}) \end{bmatrix}_{N \times n_h} \tag{9}$$

式(8)的  $N$  个方程可以写为矩阵形式

$$H\beta = T \tag{10}$$

式中  $H$  称为 ELM 的隐含层矩阵,  $T = [t_1, \dots, t_N]^T_{N \times m}$  输出权值  $\hat{\beta}$  可由下式求得

$$\hat{\beta} = H^+ T \tag{11}$$

其中  $H^+$  是神经网络输出层矩阵  $H$  广义逆.可求得预测输出  $Y = [y_1, \dots, y_N]^T_{N \times m}$ , 其中

$$y_i = \sum_{j=1}^{n_h} h(w_j x_i + b_j) \hat{\beta}_j \tag{12}$$

考虑到极端学习机随机生成的隐含层网络权值  $w_j$  和偏置  $b_j$ , 不合适的权值和偏置会影响极端学习机的精度, 为此本文提出自适应微分进化算法来优化网络参数, 提高网络映射精度.

3 自适应微分进化算法

微分进化算法是由美国学者 Storn 和 Price 于 1995 年提出的一种仿生智能计算方法<sup>[11]</sup>, 在解决复杂全局优化问题方面具有比其他进化算法更优的性能, 在收敛速度和稳定性方面超过了其他几种知名的随机算法.

微分进化算法以线性差分策略为主要特征, Price 和 Storn 先后提出了多种不同的策略来生成差分向量以实现算法的变异操作<sup>[12]</sup>, 有学者对自适应融合微分进化算法做出研究<sup>[13-15]</sup>, 结果表明寻优效果得到改善. 经过比较, 本文选择基于最优个体的变异方式, 并采用根据进化代数自适应调整缩放因子的策略, 改进算法的收敛速度, 形成自适应微分进化算法(ADE).

3.1 自适应缩放因子的变异操作

在 DE 算法中, 变异个体的生成用到了父代种群中多个个体的线性组合, 本文选择 3 个个体, 其中一个为最优个体  $x_{best}$ , 另外两个为任选的不同个体  $x_{r_1}, x_{r_2}$ , 则父代种群中任意的一个个体  $x_i$  变异后得到的个体  $x_{imt}$  可由下式得到

$$x_{imt} = x_{best} + \kappa(x_{r_1} - x_{r_2}) \tag{13}$$

其中  $r_1 \neq r_2 \neq i$ ,  $\kappa$  为缩放因子.

本文采用自适应缩放因子, 使得缩放因子不再为一常数. 自适应缩放因子可随着进化代数的增大, 以非线性方式迅速减小, 使得算法在初期具有较强的全局搜索能力, 在后期具有较强的局部寻优能力. 因此采取 sigmoid 函数形式, 其具有平滑的顶部和底部, 可使系统在进化的初期具有较大的缩放因子, 保证其进化速度, 在中期缩放因子快速减小, 进化速度减慢, 后期缩放因子保持在最小值附近, 便于局部寻优.

在第  $g$  代缩放因子为

$$\kappa_g = \kappa_{min} + \frac{(\kappa_{max} - \kappa_{min})}{1 + e^{\alpha(g - g_m)}} \tag{14}$$

其中  $\alpha$  为调整曲线下速率的系数,  $g_m$  为进化代数的一半. 如取  $\kappa_{min} = 0.5, \kappa_{max} = 0.9$ , 进化代数为 100 代时,  $g_m = 50$ , 缩放因子随进化代数和  $\alpha$  的变化曲线如图 2 所示, 可见自适应缩放因子可以通

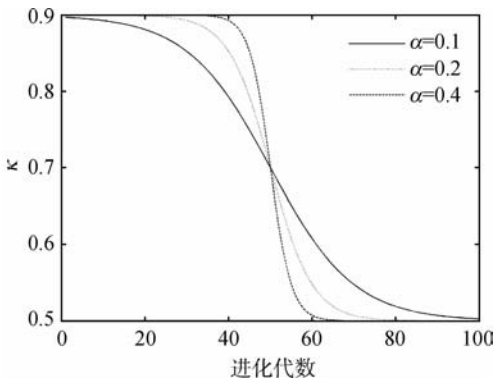


图 2 缩放因子随进化代数变化曲线  
Fig. 2 Scaling factor curves with the evolution

过调整  $\alpha$  获得期望的变化速率.

3.2 交叉以及选择操作

DE 算法交叉操作的目的是通过变异个体  $x_{imt}$  和原个体  $x_i$  的各组成元素的随机重组以提高种群个体的多样性. 经过交叉操作后生成的新个体中, 每个元素可能由变异个体提供, 也可能由原个体提供

$$x_{icr,j} = \begin{cases} x_{imt,j}, & p \leq p_{cr} \\ x_{i,j}, & \text{else} \end{cases} \tag{15}$$

其中  $p$  为随机生成的交叉后新个体的第  $j$  个元素选择为变异个体第  $j$  个元素的概率,  $p_{cr}$  为设定概率上限, 当  $p \leq p_{cr}$  时, 则第  $j$  个元素从变异个体中获得, 否则从原个体中获得.  $p_{cr}$  越大, 算法的收敛速度越快.

接下来, 对交叉获得个体  $x_{icr}$  和原个体  $x_i$  的适应度进行比较, 当且仅当  $x_{icr}$  的适应度值比原个体  $x_i$  的适应度值更好时,  $x_{icr}$  才会成为下一代进化中的新个体. 否则,  $x_i$  仍将保留在下一代的种群中, 则在第  $g+1$  代种群中, 第  $i$  个个体为

$$x_i^{g+1} = \begin{cases} x_{icr}^g & \Phi(x_{icr}^g) > \Phi(x_i^g) \\ x_i^g & \text{else} \end{cases} \tag{16}$$

其中  $\Phi$  为适应度值计算函数.

3.3 自适应微分进化算法验证

为验证自适应微分进化算法(ADE)的寻优能力, 选取 3 种常见的智能优化算法的测试函数: Schaffer, Eggcrate, Rosenbrock 开展仿真验证, 并与标准 DE(differential evolution)算法进行比较. 优化过程中参数设置如表 1 所示, 为了避免随机性, 每种算法独立运行 50 次, 在表 2 中给出了不同缩放因子和自适应缩放因子的平均优化结果.

测试函数如下:

1) Eggcrate 函数

$$\min f(x) = x_1^2 + x_2^2 + 25(\sin^2 x_1 + \sin^2 x_2)$$
$$x \in [-2\pi, 2\pi]$$

2) Rosenbrock 函数

$$\min f(x) = 100 \times (x_1^2 - x_2)^2 + (1 - x_1)^2$$
$$x \in [-10, 10]$$

3) Schaffer 函数

$$\min f(x) = \frac{[\sin^2 \sqrt{(x_1^2 + x_2^2)} - 0.5]}{[1 + 0.001(x_1^2 + x_2^2)]^2} - 0.5$$
$$x \in [-5, 5]$$

表 1 DE/ADE 参数设置表

Table 1 DE/ADE parameter setting

| 测试函数       | $N_v$ | $g_{\max}$ | $p_{\text{cr}}$ | $\min f(x)$ |
|------------|-------|------------|-----------------|-------------|
| Eggcrate   | 10    | 20         | 0.6             | 0           |
| Rosenbrock | 15    | 50         | 0.6             | -1          |
| Schaffer   | 30    | 100        | 0.6             | 0           |

其中  $N_v$  表示种群规模,  $g_{\max}$  代表最大迭代次数,  $\min f(x)$  为测试函数目标值.

表 2 函数测试结果表

Table 2 Function test results

| 算法  | Eggcrate |                         | Schaffer |                     | Rosenbrock |                          |
|-----|----------|-------------------------|----------|---------------------|------------|--------------------------|
|     | $\kappa$ | $\min f(x)$             | $\kappa$ | $\min f(x)/10^{-1}$ | $\kappa$   | $\min f(x)$              |
| DE  | 0.2      | 9.4994                  |          |                     | 0.2        | 4.8499                   |
|     | 0.3      | 9.4882                  |          |                     | 0.3        | 2.5741                   |
|     | 0.4      | 9.4882                  | 0.4      | -9.9007             | 0.4        | $8.8121 \times 10^{-1}$  |
|     | 0.5      | 9.4882                  | 0.5      | -9.9016             | 0.5        | $2.2753 \times 10^{-2}$  |
|     | 0.6      | 9.4882                  | 0.6      | -9.9028             | 0.6        | $3.1500 \times 10^{-8}$  |
|     | 0.7      | $2.6769 \times 10^{-3}$ | 0.7      | -9.9994             | 0.7        | $2.4837 \times 10^{-7}$  |
|     | 0.8      | $4.3131 \times 10^{-5}$ | 0.8      | -9.9912             | 0.8        | $2.1762 \times 10^{-5}$  |
|     | 0.9      | $2.0701 \times 10^{-4}$ | 0.9      | -9.9028             | 0.9        | $3.3710 \times 10^{-7}$  |
|     | 1.0      | $8.3198 \times 10^{-3}$ | 1.0      | -9.9025             | 1.0        | $1.1682 \times 10^{-5}$  |
|     | 1.1      | $8.5076 \times 10^{-3}$ | 1.1      | -9.9020             |            |                          |
|     | 1.2      | $1.1455 \times 10^{-2}$ | 1.2      | -9.9028             |            |                          |
| ADE | 1.2~0.1  | $8.4580 \times 10^{-6}$ | 1.2~0.1  | -10.000             | 1.2~0.1    | $1.5646 \times 10^{-12}$ |

由表 2 可见,DE 算法在不同缩放因子情况下寻优结果各不相同,在某个缩放因子下具有较优的结果.而 ADE 算法较固定缩放因子算法,具有明显的优越性,所寻得的结果具有更优的性能.特别对于 Schaffer 这样的典型多峰函数,ADE 算法获得了其最优解.

## 4 ADE-ELM 算法以及仿真验证

### 4.1 ADE-ELM 算法

为了减少隐含层节点的数目,提高 ELM 算法的精度,本文将 ADE 算法用于 ELM 隐含层权值和偏置的寻优过程,形成 ADE-ELM 算法.其流程如下:

**Step 1** 设置初始化参数,给定 ELM 的激励函数以及隐含层节点数  $n_h$ .随机产生 ADE 的初

始种群,种群中的每个个体  $x_i$  包括输入至隐含层的权值和偏置.

**Step 2** 利用式(11)计算出输出偏置后,即可计算得到网络实际预测输出  $\mathbf{Y}$ ,这里将适应度函数  $\Phi$  定义为预测输出  $\mathbf{Y}$  与期望输出  $\mathbf{T}$  平方根误差,并取适应度函数最小的个体为最优个体,即  $\mathbf{x}_{\text{best}}$ .

**Step 3** 在最优个体  $\mathbf{x}_{\text{best}}$  之外随机选择另外两个不同的个体  $\mathbf{x}_{r1}, \mathbf{x}_{r2}$ .按式(13)对  $\mathbf{x}_i$  执行变异操作,产生变异个体  $\mathbf{x}_{\text{int}}$ ,其中放缩因子按式(14)计算得到.

**Step 4** 按式(15)对  $\mathbf{x}_i$  和  $\mathbf{x}_{\text{int}}$  执行交叉操作,产生交叉个体  $\mathbf{x}_{\text{icr}}$ .

**Step 5** 按式(16)对针对种群中的交叉个体进行选择操作,获得下一代种群.

**Step 6** 检查是否满足终止条件,若满足则

结束,否则转 Step 2,利用新生成的种群,构成极端学习机,继续迭代计算直到最大迭代次数  $g_{\max}$ .

4.2 ADE-ELM 映射平衡方程猜值修正量验证

本文在直升机/涡轴发动机综合系统仿真平台上开展仿真验证工作. 仿真平台中,涡轴发动机模型是基于 T700 数据用部件法建立的非线性实时数学模型<sup>[16-17]</sup>. 该综合仿真模型经过大量的仿真验证,能够较为准确地反映系统动静态特性.

针对动态平衡方程的 3 个猜值,设计 3 个极端学习机,输入都是 5 个残差  $\epsilon_i, i=1,2,3,4,5$ ,输出为对应的猜值修正量  $\Delta v_j, j=1,2,3$ ,隐含层节点数选为  $n_h=20$  个,对应的待优化连接权值和偏置个数之和为  $5 \times 20 + 20$  个,每个待优化个体中含有 120 个元素,采用 ADE 算法对 ELM 的输入至隐含层权值和偏置进行优化,形成 ADE-ELM 算法. 考虑到过大数据量会使得极端学习机在矩阵求广义逆计算过程中精度下降,作者仅选择包线内具有代表性的工作点进行采样训练. 在高度 800,900,1000 m,前飞速度 0~75 m/s 范围内,采用 N-R 迭代算法对模型进行求解直至满足平衡方程收敛条件,每隔 5 m/s 采集一组动态数据. 其中包含 N-R 法迭代计算前的平衡方程残差和迭代收敛后的猜值修正量,共获得 3024 组数据,以迭代计算前的残差  $\epsilon_i, i=1,2,\cdots,5$  为输入,迭代收敛后猜值修正量  $\Delta v$  为输出,训练 ADE-ELM 网络. 设置优化代数为  $g_{\max}=200$ ,个体中每个元素的取值范围选为 0~1,缩放因子中  $k_{\max}=1, k_{\min}=0.1$ ,调整系数  $\alpha=0.2$ ;变异概率  $p_{cr}=0.6$ . ADE-ELM 算法对 3 个猜值修正量的训练误差如图 3~图 5 所示,各包含 3024 组训练数据点. 图中  $\Delta Z_c$  为压气机压比系数的修正量;  $\Delta Z_g$  为燃气涡轮落压比系数的修正量;  $\Delta Z_p$  为动力涡轮落压比系数的修正量;训练误差定义为预测猜值修正量  $\Delta \hat{v}_j$  与期望猜值修正量  $\Delta v_j$  的相对误差,即

$$e = \left| \frac{\Delta v_j - \Delta \hat{v}_j}{\Delta v_j} \right| \times 100\% \tag{17}$$

由图 3~图 5 可见,各修正量的训练误差均小于 0.5%.

4.3 基于 ADE-ELM 的部件级模型仿真验证

将离线训练获得的 ADE-ELM 网络参数,代入上述直升机/涡轴发动机综合系统仿真平台,开展基于 ADE-ELM 算法的部件级模型仿真验证,并将结果与 N-R 迭代算法及一次通过(single iteration per time step, SIPTS) 算法进行比较. 为

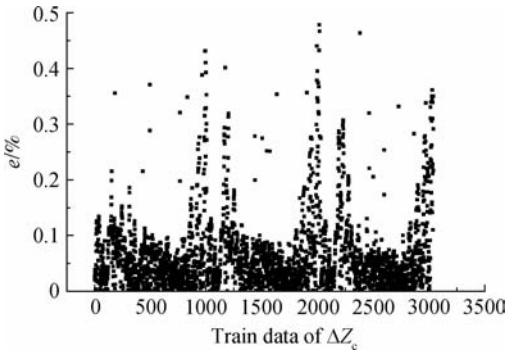


图 3  $\Delta Z_c$  猜值修正量训练误差

Fig. 3 Training error of guess value adjustment  $\Delta Z_c$

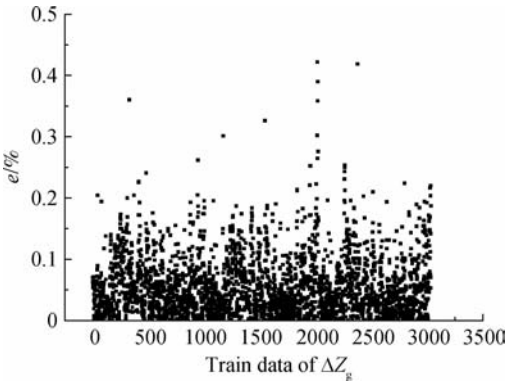


图 4  $\Delta Z_g$  猜值修正量训练误差

Fig. 4 Training error of guess value adjustment  $\Delta Z_g$

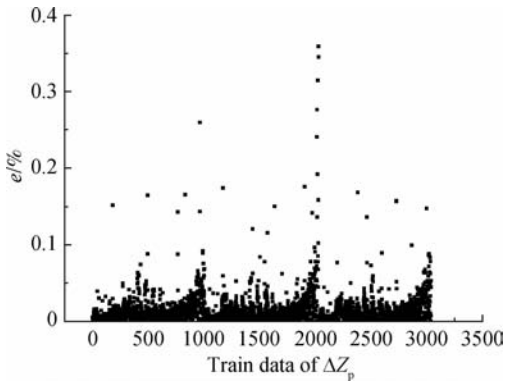


图 5  $\Delta Z_p$  猜值修正量训练误差

Fig. 5 Training error of guess value adjustment  $\Delta Z_p$

了充分验证算法的有效性,在包线范围内,开展了大量的仿真验证,通过表 3 给出了 3 组仿真中相对 N-R 迭代算法模型在动态计算中的最大偏差,并给出了第 2 种仿真情况下的模型输出曲线,如图 6 和图 7 所示. 其他飞行区域和发动机工作状态下的仿真结果类似.  $n_p$  为动力涡轮轴转速,  $n_g$  为燃气涡轮轴转速,  $M$  为动力涡轮输出转矩,  $W_f$  为燃油流量,  $T_{44}$  为动力涡轮进口总温.

表 3 不同状态下发动机模型输出最大误差对比

Table 3 Maximum error comparison of engine model outputs at different states

| 状态         | 高度 $H/\text{m}$ |         |       |         |            |         |       |
|------------|-----------------|---------|-------|---------|------------|---------|-------|
|            | 900             |         | 1 100 |         | 0          |         |       |
|            | 爬升速度            |         | 前飞速度  |         | 前飞速度       |         |       |
| 指令信号变化     | 无退化             |         | 无退化   |         | 压气机流量退化 5% |         |       |
| 发动机健康参数    | 无退化             |         | 无退化   |         | 压气机流量退化 5% |         |       |
| 算法         | SIPTS           | ADE-ELM | SIPTS | ADE-ELM | SIPTS      | ADE-ELM |       |
| 相对<br>误差/% | $n_{\text{p}}$  | 0.034   | 0.003 | 0.218   | 0.018      | 0.195   | 0.014 |
|            | $n_{\text{g}}$  | 0.120   | 0.007 | 0.607   | 0.040      | 0.546   | 0.034 |
|            | $M$             | 0.159   | 0.046 | 1.279   | 0.276      | 1.223   | 0.304 |
|            | $W_{\text{f}}$  | 0.809   | 0.034 | 4.643   | 0.199      | 4.774   | 0.189 |
|            | $T_{44}$        | 0.276   | 0.028 | 1.730   | 0.158      | 1.900   | 0.143 |
|            | $Z_{\text{c}}$  | 0.091   | 0.006 | 0.869   | 0.069      | 0.910   | 0.024 |
|            | $Z_{\text{p}}$  | 0.218   | 0.006 | 1.329   | 0.044      | 1.540   | 0.049 |
|            | $Z_{\text{g}}$  | 2.104   | 0.069 | 12.470  | 0.544      | 13.741  | 0.450 |

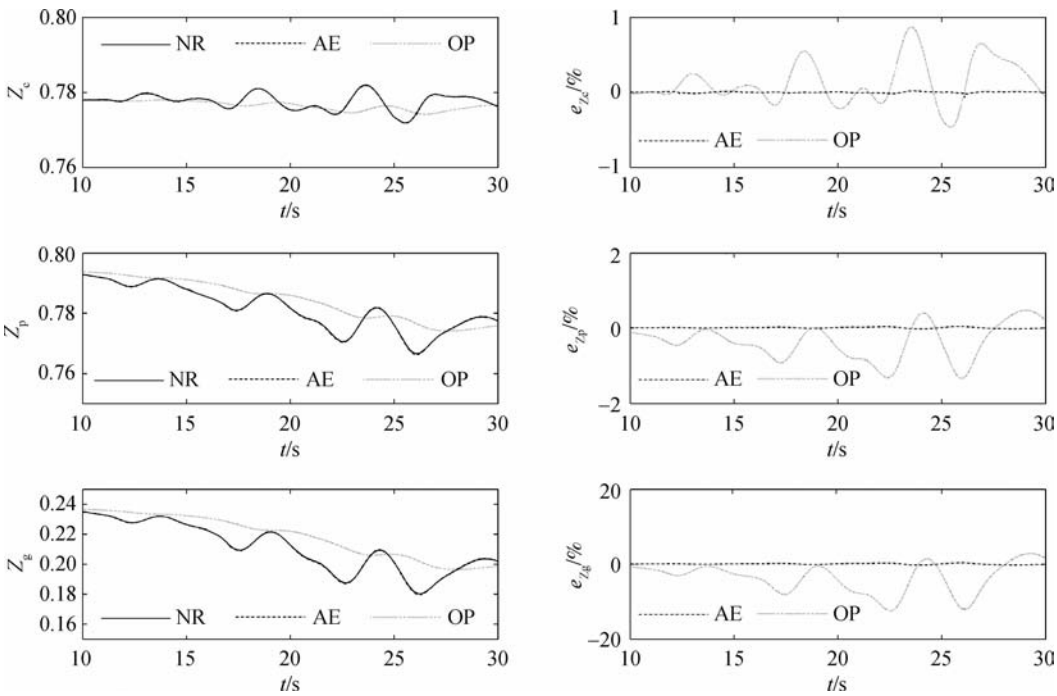


图 6 动态过程中猜值变化及相对 N-R 法误差

Fig. 6 Variation of guess values during transition state and the relative errors to N-R method

1) 在 ADE-ELM 训练数据采集区域内不同飞行状态下进行测试。

采集训练数据时,模拟直升机水平前飞状态。在本次测试中,验证其爬升工作状态。选择  $H=900\text{ m}$ ,改变爬升速度,由  $0\text{ m/s}$  间隔  $1\text{ m/s}$  变化到  $-4\text{ m/s}$  的动态过程。如表 3 可见,在不同于采集训练数据的飞行条件下,基于 ADE-ELM 算法的猜值预测相对误差均小于  $0.1\%$ 。同时基于 ADE-

ELM 算法的发动机模型各主要输出参数与 N-R 迭代模型的输出误差最大值也在  $0.05\%$  以内,远小 SIPTS 算法最大约  $0.8\%$  的误差。

2) 在 ADE-ELM 训练数据采集区域附近相同飞行状态下进行测试。

选择  $H=1100\text{ m}$ ,前飞速度由  $10\text{ m/s}$  间隔  $5\text{ m/s}$  变化到  $30\text{ m/s}$  的动态过程。仿真结果见图 6 和图 7,图中 NR 代表采用牛顿-拉夫逊迭代算法

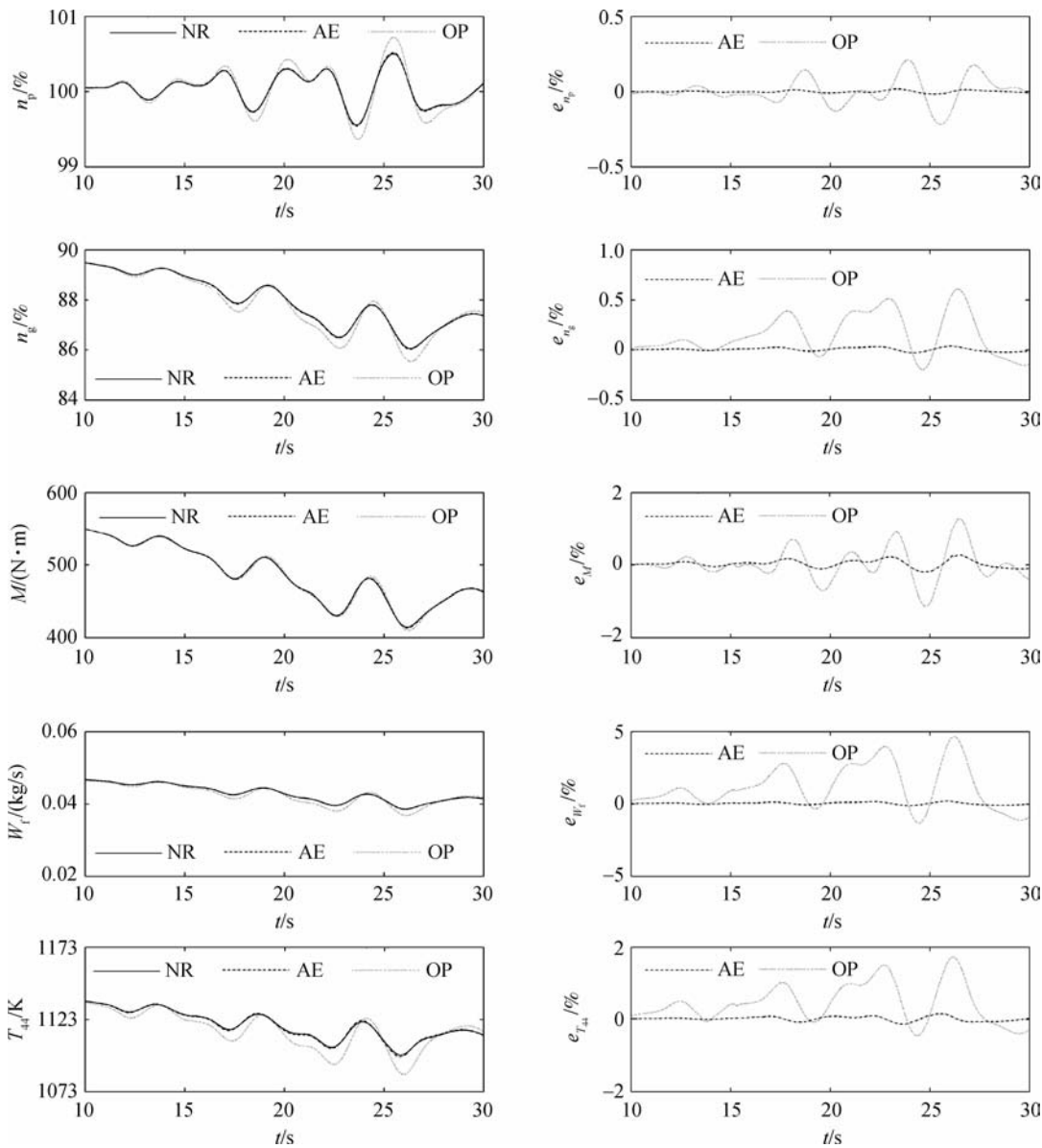


图 7 动态过程中模型输出变化及相对 N-R 法误差

Fig. 7 Variation of model outputs during transition state and the relative errors to N-R method

模型的输出, AE 代表基于 ADE-ELM 算法模型输出, OP 代表一次通过算法模型输出,  $e$  代表相对于 N-R 法的百分比输出误差,  $e$  下标代表对应的变量. 由图 6 可见 ADE-ELM 算法模型猜值预测误差在 0.6% 以内. 由图 7 可见, 发动机模型各主要输出参数与 N-R 迭代模型的输出误差最大值也在 0.3% 以内, 远小于 SIPTS 算法最大约 4.6% 的误差.

3) 在远离 ADE-ELM 训练数据采集区域进行退化发动机的测试.

在地面工作点  $H=0\text{ m}$ , 模拟压气机流量退化 5%, 前飞速度由 10 m/s 间隔 5 m/s 变化到 30 m/s 的动态过程. 由表 3 可见, 在远离 ADE-ELM 训练数据采集点, 部件有较大性能退化的工作状态下,

ADE-ELM 算法的猜值预测仍然能够很好地跟踪 N-R 迭代计算的结果, 猜值预测误差小于 0.5%. 同时基于 ADE-ELM 算法的发动机模型各主要输出参数与 N-R 迭代模型的输出误差最大值在 0.3% 以内, 远小于 SIPTS 算法大约 4.8% 的误差.

通过表 3 中发动机各特性参数输出最大误差对比可知, 基于 ADE-ELM 的涡轴发动机部件级模型最大建模误差约为一次通过算法的 1/3. 在实时性方面, 基于上述直升机/涡轴发动机综合仿真平台, 作者对 ADE-ELM 和 SIPTS 两种算法所建立发动机模型进行动态过程计算 10 万次. 仿真环境: CPU 主频 2.2 GHz, 内存 2 GB. 在循环过程中, 通过燃油流量  $W_f$  不断变化来调整发动机工作状态. 采用 CLOCK 时钟计时, ADE-ELM 算法 10 万次模型

计算平均耗时为 21.17 s,而一次通过算法 10 万次模型计算平均耗时为 60.56 s.可见,ADE-ELM 算法所用时间大概为一次通过算法的 1/3,有效说明 ADE-ELM 算法在实时性方面的优越性.

5 结 论

本文基于智能映射的方法实现了涡轴发动机平衡方程的求解.提出用基于 Sigmoid 函数变化缩放因子的自适应 DE 算法,提高了算法的寻优能力,并将其应用于极端学习机隐含层权值和偏置的优化,实现了基于 ADE-ELM 算法的涡轴发动机高精度实时模型.与一次通过算法相比,ADE-ELM 算法节约了模型调用与矩阵求逆运算时间,且 ELM 映射获得模型收敛后平衡方程猜值修正量,提高了模型的精度.仿真结果表面,基于 ADE-ELM 的涡轴发动机部件级模型最大建模误差为一次通过算法的 1/3,模型运行耗时约为一次通过算法的 1/3.涡轴发动机动态模型由于平衡方程较少,基于 ADE-ELM 的模型求解取得了满意的效果,采用此种算法求解其他类型发动机模型平衡方程的适用性尚需进一步验证.

参考文献:

[ 1 ] 姚文荣. 涡轴发动机/旋翼综合建模控制及优化研究[D]. 南京:南京航空航天大学,2008.  
YAO Wenrong. Research on integrated modeling, control and optimization of turboshaft engine/rotor[D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2008. (in Chinese)

[ 2 ] 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)

[ 3 ] 文陆军. 用实测数据建立发动机的神经网络模型[D]. 南京:南京航空航天大学,2005.  
WEN Lujun. Aeroengine modeling with neural network by using practice measure data[D]. Nanjing: Nanjing University of Aeronautics and Astronautics,2005. (in Chinese)

[ 4 ] 胡宇,杨月诚,张世英,等. 基于改进拟合法的涡扇发动机状态变量模型建立方法[J]. 推进技术,2013,34(2):405-410.  
HU Yu,YANG Yuecheng,ZHANG Shiyong,et al. Establishment of turbofan engine state variable model based on improved fitting method[J]. Journal of Propulsion Technology,2013,34(2):405-410. (in Chinese)

[ 5 ] 姜彩虹,孙志岩,王曦. 航空发动机预测健康管理系统设计的关键技术[J]. 航空动力学报,2009,24(11):2589-2594.  
JIANG Caihong,SUN Zhiyan,WANG Xi. Critical technologies for aero-engine prognostics and health management systems development [J]. Journal of Aerospace Power, 2009,24(11):2589-2594. (in Chinese)

[ 6 ] DeCastro J A,Litt J S,Frederick D K. A modular aero-propulsion system simulation of a large commercial aircraft engine[R]. NASA/TM-2008-215303,2008.

[ 7 ] May R D,Csank J,Lavelle T M,et al. A high-fidelity simulation of a generic commercial aircraft engine and controller[R]. NASA/TM-2010-216810,2010.

[ 8 ] Huang G B,Ding X,Zhou H. Optimization method based extreme learning machine for classification[J]. Neurocomputing,2010,74(1/2/3):155-163.

[ 9 ] Suresh S,Saraswathi S,Sundararajan N. Performance enhancement of extreme learning machine for multi-category sparse data classification problems[J]. Engineering Applications of Artificial Intelligence,2010,23(7):1149-1157.

[10] Liu N,Wang H. Ensemble based extreme learning machine [J]. IEEE:Signal Processing Letters,2010,17(8):754-757.

[11] Storn R,Price K. Differential evolution; a simple and efficient adaptive scheme for global optimization over continuous spaces[M]. Berkeley,CA:International Computer Science Institute,1995.

[12] 杨启文,蔡亮,薛云灿. 差分进化算法综述[J]. 模式识别与人工智能,2008,21(4):506-513.  
YANG Qiwen,CAI Liang,XUE Yuncan. A survey of differential evolution algorithms[J]. Pattern Recognition and Artificial Intelligence,2008,21(4):506-513. (in Chinese)

[13] Brest J,Greiner S,Boskovic B,et al. Self-adapting control parameters in differential evolution; a comparative study on numerical benchmark problems[J]. IEEE Transactions on Evolutionary Computation,2006,10(6):646-657.

[14] Brest J,Boskovic B,Greiner S. Performance comparison of self-adaptive and adaptive differential evolution algorithms soft computing; a fusion of foundations[J]. Methodologies and Applications,2007,11(7):617-629.

[15] Noman N,Iba H. Accelerating differential evolution using an adaptive local search[J]. IEEE Transactions on Evolutionary Computation,2008,12(1):107-125.

[16] 张海波,王健康,陈国强. 单旋翼直升机/涡轴发动机通用综合模型的构建方法:中国,102520620[P]. 2012-06-27.

[17] 张海波,姚文荣,陈国强. 涡轴发动机/直升机综合控制仿真平台设计[J]. 推进技术,2011,32(3):383-390.  
ZHANG Haibo,YAO Wenrong,CHEN Guoqiang. Design of a numeric simulation platform for integrated turbo-shaft engine/helicopter control system[J]. Journal of Propulsion Technology,2011,32(3):383-390. (in Chinese)