Aero-engine Model Correction Technology Based on Adaptive Neural Network*

Chenghan Pu1, Wenxiang Zhou2*

1 Energy and Power College Department, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu Province, 210000, China
2 Energy and Power College Department, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu Province, 210000, China

*Corresponding author’s e-mail: zhouwx@nuaa.edu.cn

Abstract. In this paper, a neural network-based algorithm is proposed to adapt the performance maps of engine component models for the mismatches between aero-engine simulation models and actual engine characteristics. Based on the general characteristics data of rotating components in GasTurb, a neural network capable of calculating the efficiency and mass flow of rotating components is trained. This neural network is introduced into the engine nonlinear component model to calculate the deviation between the output parameter of each section of the engine component model and the real engine performance indicators. The linear relationship between the parameters of the nonlinear model is solved by applying perturbation theory. The error between the output characteristics of the neural network and the real engine component characteristics is derived based on the simulation error, which makes the neural network further optimized so that it can track the current real engine performance. In this paper, a model of one turboshaft engine is used as the simulation object, and the simulation of component model building and performance maps adaptation is carried out. The simulation results show that the proposed performance maps adaptation algorithm can effectively improve the accuracy of the component-level model of the turboshaft engine, and is applicable to model correction of various types of gas turbine engines.

1. Introduction

Currently, the core ideas of aero-engine component correction techniques can be divided into two main categories. The first category is the optimization technique of component performance maps correction factor based on limited experiment data. The core idea can be summarized as defining a set of correction factors for component performance maps, minimizing the simulation error of the model, using different optimization algorithms to optimize the best set of component characteristic correction factors so that the model output can accurately match the experiment data.

In 1990, Stamatis A et al. of the National Technical University of Athens proposed a method to automatically correct component performance maps based on actual engine experiment data. This method defined correction coefficients around the design point, aiming at minimizing the error between the model simulation output and the engine experiment data, and completed the correction of component performance maps through iteration [1]. In 2005, Y G Li et al. from Cranfield University, UK, proposed a method to match measurable important performance parameters with experiment data and solve the correction coefficients of unmeasurable parameters such as efficiency by using Newton Rafelson's algorithm to finally obtain more accurate true values of unmeasurable parameters in response to the...
problem of inaccurate calculation of engine component-level model design points\cite{3}. In 2005, Wu Hu et al. from Northwestern Polytechnic University proposed an adaptive correction method for the performance maps of engine components, which used a simplex algorithm to implement the correction of the fan and core machine components\cite{5}.

The second category is based on massive engine experiment data for component characteristic correction. In 2003, C Kong et al. at Howon University in South Korea first applied the system identification method to the correction of aero-engine component performance maps, which corrected design and off-design points to varying degrees, and used the Canadian Pratt & Whitney PT6A-62 turboprop engine as an example to reconstruct the component performance maps by using the system identification correction factor to prove the effectiveness of the method\cite{9}. In 2006, C Kong found that the compressor flow rate can be expressed as a cubic polynomial in the pressure ratio, and the efficiency can be expressed as a cubic polynomial in the flow rate, and based on a large amount of experimental data, a cubic polynomial curve was fitted by a genetic algorithm. The study showed that this method could obtain a more accurate map of the compressor component performance maps\cite{10}. In 2007, C Kong improved the fitness function in the genetic algorithm and carried out simulation analysis with the PW206C turboshaft engine, showing that the method can reduce the reliance on experiment data and greatly reduce the calculation time and improve the model correction efficiency\cite{11}.

With the development of deep learning, a novel data-based correction technology is proposed. In 2004, Chen Ce et al. of the College of Engineering, Air Force Engineering University applied the BP neural network algorithm to the correction of turbofan engine component performance maps, and through the powerful learning ability of the neural network, the accurate extrapolations of fan performance maps under two operating conditions of low and high speed were completed, which initially verified the feasibility of using the BP neural network calculation module to replace the interpolation calculation of engine component performance maps\cite{13}. Based on this research, a deep learning-based engine model correction algorithm is proposed in this paper. First, the generic component characteristic curve in Gasturb is used as a pre-training sample for the neural network to compare the deviation between the engine design performance requirements and the nonlinear model of the engine, and deduce the error between the compressor component performance maps before the model revision and the real component performance maps reversely. Then a new training sample was generated. Finally, the original data is augmented and iterated. The neural network can be automatically updated to fit the real engine performance maps.

2. Neural network-based modeling of engine component-level models

In this paper, component-level modeling is carried out for a turboshaft engine (whose structure contains the intake, compressor, burner, gas turbine, power turbine and Nozzle, as shown in figure 1), with the engine sections defined as shown in Table 1.
Table 1. Definition of the turboshaft engine section.

| Section number | Section definition | Section number | Section definition |
|----------------|--------------------|----------------|--------------------|
| 0              | Undisturbed airflow | 41             | Gas turbine inlet  |
| 1              | In-take inlet      | 44             | Gas turbine outlet |
| 2              | In-take outlet     | 45             | Power turbine inlet | |
| 21             | Compressor inlet  | 5              | Power turbine outlet |
| 3              | Compressor outlet | 6              | Nozzle inlet      |
| 31             | Burner inlet      | 7              | Nozzle outlet      |
| 4              | Burner outlet     |                |                    |

Several simulations have shown [8-12] that the performance maps of the aero-engine compression components have a large impact on the accuracy of each section parameters. Therefore, the purpose of this paper is to correct the performance maps of the aero-engine compressor components.

In contrast to the traditional modeling approach of calculating the mass flow and efficiency of compressor components, this paper uses a neural network to replace the two-dimensional interpolation function to calculate the compressor flow rate $W_{a21}$ and efficiency $\eta_{cp}$ (as shown in figure 2). The neural network is set up as a dual-input, dual-output structure that takes the corrected speed $n_{cp,cor}$ and the pressure ratio $\pi_{cp}$ of the compressor as inputs, and outputs the compressor conversion flow rate $W_{a21}$ and efficiency $\eta_{cp}$. In this paper, the generic compressor performance maps data in Gasturb are used as the initial training samples for the neural network, and the initial training samples are expanded using interpolation to ensure that the neural network can thoroughly learn the compressor performance maps. Finally, a total of 5760 training samples were obtained to pre-train the neural network in this paper.

![Figure 2. Comparison of two nonlinear component-level model calculations](image)

Table 2. Neural network hyper-parameter settings.

| Parameter                   | Parameter values | Parameter                   | Parameter values |
|-----------------------------|------------------|-----------------------------|------------------|
| Number of hidden layers     | 1                | Learning Rate               | 0.05             |
| Number of node in hidden layer | 80           | Training algorithms         | Adam algorithm   |
| Training batch size         | 64               | Loss function               | L2 loss function |
| Hidden layer activation function | Relu          | Output layer activation function | Linear mapping |
After replacing compressor component performance maps with the neural network, the engine component model conversion flow rates $W_{a21}$ and efficiency $\eta_{cp}$ are calculated as shown in figure 3.

The literature [13] and [19] stated that a single-layer neural network could use the corrected speed $n_{cp,cor}$ and the pressure ratio $\pi_{cp}$ of a compressor as inputs to accurately calculate its conversion flow rate $W_{a21}$ and efficiency $\eta_{cp}$. However, it has not been verified that the accuracy of the engine component-level model can be guaranteed by using neural networks to replace the compressor performance maps calculation. To verify the feasibility, three steady-state points were selected for comparison between the two modeling approaches in this paper (Table 3). The deviation of the neural network model output from the standard value is calculated by using the traditional component model output as the standard value. The maximum deviation of the outputs of the two modeling methods was tested to be no more than 0.3%, meeting the model accuracy requirements.

| Sample source | ng (%) | $T_{at5}$ (K) | $W_2$ (kg/s) | $\pi_{cp}$ | $N_e$ (kW) |
|---------------|--------|--------------|--------------|------------|-------------|
| Traditional model output | 99.98 | 1179.24 | 4.717 | 17.596 | 1485.99 |
| Neural network model output | 99.96 | 1180.48 | 4.713 | 17.587 | 1485.29 |
| Steady-state error (%) | 0.02 | -0.10 | 0.08 | 0.05 | 0.04 |

| Sample source | ng (%) | $T_{at5}$ (K) | $W_2$ (kg/s) | $\pi_{cp}$ | $N_e$ (kW) |
|---------------|--------|--------------|--------------|------------|-------------|
| Traditional model output | 98.98 | 1169.95 | 4.650 | 17.271 | 1439.62 |
| Neural network model output | 99.0 | 1169.22 | 4.654 | 17.280 | 1440.31 |
| Steady-state error (%) | -0.02 | 0.06 | -0.08 | 0.05 | -0.04 |

| Sample source | ng (%) | $T_{at5}$ (K) | $W_2$ (kg/s) | $\pi_{cp}$ | $N_e$ (kW) |
|---------------|--------|--------------|--------------|------------|-------------|
| Traditional model output | 95.36 | 1138.91 | 4.405 | 16.093 | 1275.46 |
| Neural network model output | 95.37 | 1138.24 | 4.409 | 16.100 | 1276.00 |
| Steady-state error (%) | -0.01 | 0.05 | -0.09 | -0.04 | -0.04 |
Figure 4 shows the learning effect of the neural network for the compressor performance maps in Gasturb. It can be seen that the Mass Flow-Efficiency and Mass Flow-Pressure Ratio curves output by the neural network largely overlap with the training data.

3. Aero-engine Model Correction Based on Adaptive Neural Network

To avoid the problem that neural networks can only simulate the general performance maps of engines, this chapter, based on the previous analysis, compares the actual experiment data of aero-engines, deduces the deviation between neural network and engine design requirements, and enhances and iterates the original data, so that the corrected neural network can simulate the performance maps of the compression part of real engines.

3.1. Back transfer of component-level model error

As shown in figure 5, the characteristic correction method of aero-engine compression parts based on the neural network can be divided into two levels in principle:

Figure 5. General scheme of aero-engine model correction based on adaptive neural network

Level 1: According to the steps in Chapter 2, the mathematical models of neural network output and component-level model output parameters are established, the error transfer equation is derived according to the gradient, and the model error is transferred to the neural network output data of each compression component.
Level 2: the corresponding neural network is trained according to the neural network output parameter deviation of each compression component. The network weights are updated to modify the characteristic curve and reduce the simulation calculation deviation of the engine model.

In the first level, based on the design point simulation data and engine experiment data, the small perturbation method is used to calculate the deviation, and the specific steps are as follows:

Step 1: The neural network output data $x_0$ and the component-level model output data $y_0$ are obtained by iterative calculation using the joint working equations of components and models, Newton-Lavesen algorithm and small perturbation method. Small perturbations are made around the neural network output data, and the engine model is iteratively calculated to obtain new neural network output data $x_k$ and component-level model output data $y_k$.

According to the output data $x_k$ and $y_k$, the equations are solved, and the partial derivative coefficient matrix from the output data of the component-level model to the output data of the neural network is obtained using formula (1) and formula (2).

$$
J = \left[ \begin{array}{ccc}
\frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial y_n}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_n}
\end{array} \right] \quad (2)
$$

The vector $y$ is the output data of the component-level model, the vector $x$ is the output data of the neural network, and the matrix $J$ is the partial derivative coefficient matrix.

According to formula (3) and formula (4), the influence factor $k$ of different compression components on the output parameter deviation of the component-level model is calculated.

$$
\Delta y = y - y' \\
k = f_1(num, \Delta y) \quad (3)
$$

In which vector $y'$ is the real sensor measurement data, vector $\Delta y$ is the deviation between output data of component-level model and experiment data, code $num$ represents the corresponding compression component, and $f_1(num,\Delta y)$ is a function for calculating the influence factors of different compression components on the deviation of output parameters of the component-level model.

According to formula (5) and formula (6), the deviation of engine model simulation results and experiment data is calculated, and the deviation to be corrected is assigned to each compression part according to the influence factor $k$ calculated above, then it is brought into the partial derivative coefficient matrix $J$ respectively to calculate the deviation of neural network output data.

$$
\Delta y_{num} = k\Delta y \\
\Delta x_{num} = J^{-1}\Delta y_{num} \quad (5, 6)
$$

The vector $\Delta y_{num}$ is the deviation assigned to the corresponding compression component to be corrected, and the vector $\Delta x_{num}$ is the deviation of neural network output data calculated by the corresponding component.

Step 2: Modify the neural network based on the deviation of the output parameters of the neural network.

In Step 1, the deviation $\Delta x_{num}$ between the performance maps of compression parts simulated by neural network and those of real engine compression parts is output, and in Step2, the output is used as the compensation amount of training samples, which is input to the samples to enhance the data. The enhanced data formula is shown in Formula (7), Formula (8) and Formula (9). That is, according to the current engine conversion speed, extract the equal speed line of the characteristic curve under the current conversion speed, and give a training set according to the extracted data:

$$
x_{num} = \Delta x_{num} + x \\
x = (x_1, x_2, ..., x_k) \\
D = \{ (m_1, x_1), (m_2, x_2), ..., (m_k, x_k) \} \quad (7, 8, 9)
$$

where $x$ is the output data of equal speed line before neural network training, $x_{num}$ is the training
data of neural network, and vector $\mathbf{m}_i((i = 1, 2, \ldots, k))$ is the input data of the neural network. Set the mean square error of neural network on samples $(\mathbf{m}_i, \mathbf{x}_i)$ as shown in formula (10), and adjust the weights and thresholds of the neural network by the negative gradient method.

$$E = \frac{1}{2} \| (\Delta \mathbf{x}_{num}) \|_2^2$$  \hspace{1cm} (10)

At this point, this paper has completed the iterative algorithm of simulating the real engine characteristic data by the neural network, and iterated and enhanced the engine component level model several times until the output parameter deviation $\Delta \mathbf{y}$ is less than the given value. Finally, this paper records the input and output data sets of the neural network that meet the requirements of deviation range, and applies them to the multi-point correction method.

3.2. Iteration and correction results of the algorithm

To verify the effectiveness of the algorithm, the characteristic curves of the combined compressor parts of the engine model with the conversion speed of 0.9-1.0 are offset. The offset characteristic curves are taken as the actual performance maps of the engine compressor parts. The engine simulation results based on this characteristic diagram are taken as the reference values. It can be seen from Table 3.1 that before the model is revised, the simulation error of the operating point between 0.9 and 1.0 is quite different from the reference value. To correct the error between the simulation output of the current engine model and the reference value, this paper uses the model correction method in 3.1 to fix the current model characteristic diagram. The simulation error after model correction is shown in Table 4.

| Sample source                | ng(%) | $T_\text{at}$ (K) | $W_2$ (kg/s) | $\pi_{cp}$ | $N_e$ (kW) |
|-----------------------------|-------|-------------------|--------------|------------|-----------|
| Reference value             | 97.11 | 1109.22           | 4.588        | 16.578     | 1317.17   |
| Simulation results before correction | 95.37 | 1138.24           | 4.409        | 16.100     | 1276.00   |
| Corrected simulation results | 97.35 | 1112.00           | 4.568        | 16.532     | 1311.41   |
| Steady-state error before correction (%) | 1.8  | -2.6              | 3.9          | 2.9        | 3.1       |
| Corrected steady-state error (%) | 0.25 | -0.2              | 0.4          | 0.2        | 0.4       |

| Sample source                | ng(%) | $T_\text{at}$ (K) | $W_2$ (kg/s) | $\pi_{cp}$ | $N_e$ (kW) |
|-----------------------------|-------|-------------------|--------------|------------|-----------|
| Reference value             | 95.85 | 1081.41           | 4.532        | 16.500     | 1249.73   |
| Simulation results before correction | 93.51 | 1125.00           | 4.270        | 15.474     | 1191.94   |
| Corrected simulation results | 96.13 | 1083.24           | 4.522        | 16.141     | 1247.60   |
| Steady-state error before correction (%) | 2.4  | -4.0              | 5.8          | 4.2        | 4.6       |
| Corrected steady-state error (%) | 0.3  | -0.1              | 0.2          | 0.05       | 0.2       |

| Sample source                | ng(%) | $T_\text{at}$ (K) | $W_2$ (m3/h) | $\pi_{cp}$ | $N_e$ (kW) |
|-----------------------------|-------|-------------------|--------------|------------|-----------|
| Reference value             | 93.41 | 1073.65           | 4.342        | 15.381     | 1153.16   |
| Simulation results before correction | 91.65 | 1109.65           | 4.128        | 14.836     | 1106.59   |
| Corrected simulation results | 93.00 | 1066.83           | 4.347        | 15.549     | 1156.12   |
| Steady-state error before correction (%) | 1.9  | -3.4              | 5.0          | 3.5        | 4.0       |
| Corrected steady-state error (%) | 0.4  | 0.6               | -0.1         | -0.1       | -0.3      |
It can be seen from figure 6 that the neural network-based characteristic correction technology proposed in this paper effectively corrects the operating points with a large deviation between the initial characteristic and the reference value, significantly reduces the simulation error of the engine component-level model, and can control the steady-state error of the engine within 0.5%.

Figure 6. Comparison of errors before and after model correction (T45 error, W2 error, PIC error, Ne error)

Figure 7 shows the correction effect of the compression component characteristic correction method proposed in this paper on the compressor characteristic curve. Among them, the reference characteristic curve is the compressor characteristic curve after being biased, and the characteristic curve before being corrected is the characteristic curve without being biased. It can be seen that, after iterative calculation by the optimization algorithm, the corrected characteristic curve is gradually close to the characteristic curve after deflection.

In Figure 7, two working points with $W_I$ values of 83.5% and 88.5% are selected to show the working point status before and after the characteristic correction. Among them, reference points A and B are the working conditions calculated by the engine simulation model, points $A'$ and $B'$ are the working conditions calculated by engine simulation model when the characteristic curve is not biased, and points A and B are the working conditions calculated by revised engine simulation model. It can be seen that after iterative calculation by the optimization algorithm, the working conditions calculated by the engine simulation model are closer to the reference working conditions.
4. Conclusion

In this paper, a method of correcting the characteristic curve of aero-engine compression parts based on neural network is proposed, which solves the problem of low accuracy of engine model caused by the inaccurate characteristic curve in the past. Compared with the ordinary interpolation method, the proposed neural network method improves the accuracy of the characteristic curve, and ensures the stability and generalization ability of the model. Moreover, it can greatly improve the accuracy of the characteristic curve with less computation, which effectively solves the problem that the characteristic curve of compression parts is inaccurate or cannot be corrected, and is suitable for the model correction of various types of gas turbines.

References

[1] Stamatis A, Mathioudakis K, Papailiou K D. (1990) Adaptive simulation of gas turbine performance. Journal of Engineering for Gas Turbines and Power., 112(2): 168–175.
[2] Li Y G, Pilidis P, Newby M A. (2005) An adaptation approach for gas turbine design-point performance simulation. Turbo Expo: Power for Land, Sea, and Air., 47284: 95-105.
[3] Xie Guanghua, Zeng Qingfu, Zhang Yandong. (1998) Adaptive parameter correction of the aero-engine simulation model. Journal of Aerospace Power., 13(1): 37–40.
[4] Duan Shoufu, Fan Siqi, Lu Yan. (1999) Research on adaptive modeling technology of aero-engine. Journal of Aerospace Power., 14(4): 440–443.
[5] Wu Hu, Xiao Hong, Jiang Jianjun. (2005) Adaptive simulation of turbofan engine component characteristics. Propulsion technology., 26(5): 430–433.
[6] Chen Yuchun, Huang Xing(2008). Method for determining the adaptive model of turbine engine component characteristics. Propulsion Technology., 29(2): 214–218.
[7] Xiao Hong, Liu Zhenxia, Lian Xiaochn(2008). Comparison of two adaptive models for turbofan engine components. Journal of Civil Aviation University of China., 26(3): 17–19.
[8] Dong Zhen, Zhou Wenxiang, Pan Muxua, Huang Kaiming(2018). Method for correcting and updating the characteristics of turboshift engine components. Aero-engine., 44 (06):11-16.
[9] C Kong, J Ki, M Kang(2003). A new scaling method for component maps of a gas turbine using system identification. Journal of Engineering for Gas Turbines and Power., 125(7): 979–985.
[10] C Kong, S Kho, J Ki(2006). Component map generation of a gas turbine using genetic algorithms. Journal of Engineering for Gas Turbines and Power., 128(5): 92–96.
[11] C Kong, J Ki(2007). Components map generation of gas turbine engine using genetic algorithms and engine performance deck data. Journal of Engineering for Gas Turbines and Power., 129(2): 312–317.
[12] Y G Li, P Pilidis(2011). M A Newby. Improved multiple point nonlinear genetic algorithm-based
performance adaptation using the least square method. Journal of Engineering for Gas Turbines and Power., 134(1): 202–207.

[13] Chen Ce, Li Jun(2004). BP network research on the characteristics of turbofan engine components. Journal of Aerospace Power., 19(1): 61–64.

[14] Chen Min, Tang Hailong, Zhang Jin(2005). Real-time performance simulation model of a twin-shaft afterburner turbofan engine. Journal of Aerospace Power., v20 (1):13–17.

[15] Cao Yuan, Jin Xianlong, Meng Guang(2004). Review and Prospect of Aero-engine System-level Simulation Research. Journal of Aerospace Power., v19 (4):562-571.

[16] Cao Yuan, Jin Xianlong, Meng Guang(2005). Nonlinear modular modeling and simulation of aero-engine. Journal of Computer-Aided Design & Computer Grap., v17 (3): 505-510.

[17] Taylor B L, Oppenheimer F L(1950). Investigation of Frequency-Response Characteristics of Engine Speed for a Typical Turbine-Propeller Engine. NACA TN-2184.

[18] Delio G J(1952). Evaluation of Three Methods for Determining Dynamic Characteristics of a Turbojet Engine. NACA TN-2634.

[19] Lei Jihong, Chen Ce, Li Jun, etc(2003). Component characteristics of turbofan engine based on BP neural network, Gas Turbine Test and Research, v16 (4):15-17.