Received December 3, 2017, accepted January 2, 2018, date of publication January 10, 2018, date of current version September 7, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2789935

Aero-Engine On-Board Dynamic Adaptive MGD Neural Network Model Within a Large Flight Envelope

QIANGANG ZHENG, HAIBO ZHANG, YONGJIN LI, AND ZHONGZHI HU
Nanjing University of Aeronautics and Astronautics, Jiangsu Province Key Laboratory of Aerospace Power System, Nanjing 210016, China

Corresponding author: Haibo Zhang (zh_zhhb@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 51576096, in part by Q. Lan and the 333 Project, in part by the Funding of Jiangsu Innovation Program for Graduate Education (the Fundamental Research Funds for the Central Universities) under Grant KYLX16_0397, and in part by the Research Funds for Central Universities under Grant NF2018003.

ABSTRACT A novel modeling method, which is based on a min-batch gradient descent neural network (MGD NN), is proposed to establish an adaptive dynamic model of a turbofan engine in a large flight envelope. For establishing a high precision engine dynamic model in a large flight envelope, it always needs a very big training data. This proposed method adopts the MGD algorithm, which is more suitable to train a neural network for big training data due to it consumes much less time to update NN parameters. Dramatically, the huger training data of the MGD NN is the better generalization performance it would be. Furthermore, a regularization strategy, which will also improve the generalization performance of the MGD NN, is applied here. Finally, compared with a popular support vector regression (SVR) modeling method, the proposed method for the adaptive dynamic model of the turbofan engine is validated within a supersonic cruise envelops. The results show that the proposed method has not only much higher precision, but also less data storage and better real-time ability than the SVR method.

INDEX TERMS Neural network, real-time, dynamic adaptive model, support vector regression, data storage.

I. INTRODUCTION

Aero-engine model especially the dynamic model always plays a vital role in aero-engine control systems designed process. As well known, some necessary simulation experiments should be conducted in advance by employing a mathematic aero-engine model instead of the real aero-engine for reducing the costs, decreasing accident risks and shortening development period [1].

With the development of computer science, aero-engine controls are experiencing an evolution from sensor-based type to model-based ones. This model-based control concept will fully utilize the information of the controlled plant as reported in NASA (National Aeronautics and Space Administration) IEC (Intelligent Engine Control) program [2]–[4]. For example, modern aero-engine controls like life extending control [5], fault diagnosis [6], performance seeking control [7] and performance deterioration mitigation control [8] are all implemented with on-board adaptive dynamic model of aero-engine, which is able to accurately track engine parameters for on-line monitoring and self-optimization.

However, aero-engine is a multi-variable, nonlinear and extremely complicated thermodynamics system, operating with changeable environment and large flight envelop. In addition, the performance of its components is certain to gradually degrade during aero-engine service period. So, establishing a high precision dynamic model for aero-engine even though in a small envelope, such as sea level test state, is intractability. Nevertheless, it is still in urgent need to establish on-board adaptive model embedded in FADEC (Full Authority Digital Electronic Controller) to real-time tracking the engine performance, as preliminarily done for famous F119 turbofan engine health management [9], [10].

So far, the most popular method for establishing on-board dynamic model of aero-engine is PLM (Piecewise Linear Modeling). This method has good real-time performance and is suitable for some steady state performance seeking [9]–[12]. However, significant cumulative errors do inevitably exist in the simplified piecewise process.

For the on-line optimization of some transient operations, such as acceleration, deceleration and afterburning on/off, the piecewise linear model is not good enough anymore. And some alternative methods need to provide much better transient precision for some key parameters, such as compressor surge margin and turbine inlet temperature. Besides, when modeling within a large flight envelope, it needs a huge data storage to store vast state variable models.

Hence, a SVR (Support Vector Regression) method was reported and devised on-board engine dynamic models in references [13], [14]. However, unfortunately, along with the increasement of the training data, the training time of SVR will increase rapidly. Moreover, for huge data within a large fight envelope, it is very hard to only set up a single SVR model instead of many sub SVR models [14]–[19].

With the ability of approximating nonlinear function with arbitrarily high precision [20], [21], the NN (Neural Network) had arisen a lot of interests in engine modeling in last three decades [22]–[25]. However, due to poor generalization ability of the conventional NN, SVR methods seems to serve as a more dominant role in engine on-board dynamic model since the beginning of this century [15]–[18], [23]–[27].

The optimization method for the conventional NN training always adopts BGD (Batch Gradient Descent) method. This method needs to compute all the gradients of the entire training data before NN parameters updated. It costs a large amount of time and limits a successful application of NN for huge sample data. In the last decade, Hinton, Le Cun and other researchers proposed some new neural networks [28]–[31], such as DBN (Deep Belief Nets) [28], CNN (Convolutional Neural Networks) [30] and Faster-RCNN [31]. And they made a breakthrough in pattern recognition, phonetic classification and object detection. It greatly reignited an interest in NN field, such as unmanned helicopter [32], [33], robots control [34]–[36].

Among the optimization technology in these new neural networks, a SGD (Stochastic Gradient Descent) training method is one choice reported in [37]. It updates the NN parameters using the gradients of one data point instead of the whole training data set. Compared with the conventional BGD, it has much faster convergence rate. This indicates that the new NNs might be suitable for huge data training. Nevertheless, only selecting one training point is always sensitive to the noise and might not be the best gradient direction choice, which results in an increasing number of iterations. Hence, a MGD (Mini-batch Gradient Descent) method was proposed in [38] and [39], which is a compromise scheme between BGD and SGD [39]. It has less computation time than BGD and has the ability to choose a better descent direction than SGD. Moreover, some regulation measures, such as L_1/L_2 regulation [38], [39], were proposed to improve generalization ability of NN.

Therefore, a new modeling method, which adopts MGD neural network, is proposed to establish the adaptive dynamic model of turbofan engine. Simulations of SVR and the proposed method are both conducted in a supersonic cruise envelope in consideration of engine component degradation. The results show that the proposed method has better generalization and real-time ability compared with SVR. And it has less data storage and is able to be utilized in a larger envelope.

The arrangement of this article is described below. In section II, the preliminary study of MGD NN mainly includes the structure of MGD NN, the cost function choosing of BGD, SGD and MGD NNs, regularization strategy and back forward. In Section III, a benchmark test simulation platform of a turbofan engine is clarified. In Section IV, an on-board dynamic engine model based on MGD NN is established. In Section V, simulations of SVR and the proposed method are conducted separately in a super-sonic cruise envelope.

II. THE PRINCIPLE OF MGD NN

In this section, the principle of the MGD NN is presented, containing the structure and cost function of MGD NN, regularization strategy and the back forward algorithm. Each part for the new method is clarified in detail as follows.

A. THE STRUCTURE OF MGD NN

The MGD NN is a non-linear mapping from a input vector $\mathbf{x}_i$ to an output vector $\mathbf{y}_i$, where $i = 1, 2, \cdots, N, N$ is the size of the training set. A multi-layer MGD NN has more than one hidden layer, and each hidden layer is defined as follows:

$$\mathbf{h}_i^k = \boldsymbol{\sigma}(\mathbf{W}^k \mathbf{h}_i^{k-1} + \mathbf{b}^k) \tag{1}$$

where $\mathbf{h}_i^0 = \mathbf{x}_i$ is the input of the neural net, $\mathbf{h}_i^k$ (for $k > 0$) is the output of the k-th hidden layer with a weight matrix $\mathbf{W}^k$ and a bias (or offset) vector $\mathbf{b}^k$, $k = 1, 2, \cdots n_l$, n_l is the number of layers, $\boldsymbol{\sigma}$ is activation function.

B. THE COST FUNCTION OF MGD NN

Any feed-forward NNs can be trained with BP (Back Propagation) algorithm. The differences of them are just to change the cost functions. The most popular cost function is SSE (Summed Squared Error). The cost functions for BGD and SGD are given as follows to show the priority of MGD NN.

And the cost function of BGD, which is usually used by the conventional neural network, can be defined as:

$$J(\mathbf{W}, \mathbf{b}; \mathbf{x}, \mathbf{y}) = \min_{\mathbf{W},\mathbf{b}} \sum_{i=1}^{N} \frac{1}{2} \left\| \mathbf{h}(\mathbf{x}_i) - \mathbf{y}_i \right\|^2 \tag{2}$$

where $\mathbf{h}(\mathbf{x}_i)$ is the n-th hidden layer output of the neural network. From Eq. (2), it can be found that the SSE of the entire training set needs to be computed for updating the traditional NN parameters, which leads to much bigger computation complexity and much longer computation time.

For SGD NN, the cost function is as:

$$J(\mathbf{W}, \mathbf{b}; \mathbf{x}, \mathbf{y}) = \min_{\mathbf{W},\mathbf{b}} \frac{1}{2} \left\| \mathbf{h}(\mathbf{x}_i) - \mathbf{y}_i \right\|^2 \tag{3}$$

It is found that the SGD only uses one training point to calculates the cost function. Hence it costs less time to update neural network. However, its cost function deviates from the real one easily when the training data have noise.

Hence, the MGD algorithm is proposed in [40] and [41]. The training set is randomly divided into M batches, and each batches have the same size N_b. Then the cost function can described as:

$$J_1(\mathbf{W},\mathbf{b};\mathbf{x},\mathbf{y}) = \min_{\mathbf{W},\mathbf{b}} \sum_{j=1}^{N_b} \frac{1}{2}\left\|\mathbf{h}(\mathbf{x}_{b_i,j}) - \mathbf{y}_{b_i,j}\right\|^2 \tag{4}$$

where $i = 1, 2, \cdots M$, $\sum b_i = N$, $\bigcup_i^M \bigcup_j^{b_i} \mathbf{x}_{b_i,j} = \bigcup_i^N \mathbf{x}_i$, $\bigcup_i^M \bigcup_j^{b_i} \mathbf{y}_{b_i,j} = \bigcup_i^N \mathbf{y}_i$. The cost function of MGD is related to sub-training sets. This means it costs less computation time than BGD and has higher precision and more computation stability than SGD.

C. REGULARIZATION STRATEGY

Generalization ability, which means how to make NN perform well not only on the training set, but also on the testing set (or new input), is a very important index for evaluating some kind of NN. With a L_2 parameter regularization, the Eq.(4) for MGD NN is rewritten as:

$$\begin{aligned} &J(\mathbf{W},\mathbf{b};\mathbf{x},\mathbf{y}) \\ &= \min_{\mathbf{W},\mathbf{b}} \sum_{j=1}^{N_b} \frac{1}{2}\left\|\mathbf{h}(\mathbf{x}_{b_i,j}) - \mathbf{y}_{b_i,j}\right\|^2 + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}} W_{ji}^l \end{aligned} \tag{5}$$

where s_l is the number of l-th layer note. The second term of Eq. (5) is called a weight decay term that tends to decrease the magnitude of the weights and helps to prevent from overfitting. λ is a decay parameter that modulates weights of cost function.

D. BACK-PROPAGATION ALGORITHM

Back-propagation is actually a propagated gradient descent algorithm, which is efficient to compute the gradients of the weights and bias changes. One iteration to update the parameters $\mathbf{W}$, $\mathbf{b}$ is expressed as follows:

$$W_{ij}^l = W_{ij}^l + \alpha \nabla W_{ij}^l \tag{6}$$

$$b_i^l = b_i^l + \alpha \nabla b_i^l \tag{7}$$

where α is the learning rate, it can decrease as iterations increase. Using the momentum technology, ∇W_{ij}^l and ∇b_i^l can be updated as follows:

$$\nabla W_{ij}^l \leftarrow \eta \nabla W_{ij}^l + (1-\eta)\frac{\partial J(\mathbf{W},\mathbf{b};\mathbf{x},\mathbf{y})}{\partial W_{ij}^l} \tag{8}$$

$$\nabla b_i^l \leftarrow \eta \nabla b_i^l + (1-\eta)\frac{\partial J(\mathbf{W},\mathbf{b};\mathbf{x},\mathbf{y})}{\partial b_i^l} \tag{9}$$

where η is momentum factor.

For the output layer n_l, set

$$\boldsymbol{\delta}^{n_l} = \frac{\partial J(\mathbf{W},\mathbf{b};\mathbf{x},\mathbf{y})}{\partial \mathbf{h}^{n_l}} \tag{10}$$

For $l = n_l - 1, n_l - 2, \cdots, 2$, set

$$\boldsymbol{\delta}^l = \left[\mathbf{W}^l\right]^T \boldsymbol{\delta}^l \left[\boldsymbol{\sigma}^l\right]' \tag{11}$$

Computing the desired partial derivatives, which are given as:

$$\frac{\partial J(\mathbf{W},\mathbf{b};\mathbf{x},\mathbf{y})}{\partial W_{ij}^l} = \mathbf{h}_j^l \boldsymbol{\delta}_j^{l+1} \tag{12}$$

$$\frac{\partial J(\mathbf{W},\mathbf{b};\mathbf{x},\mathbf{y})}{\partial b_j^l} = \boldsymbol{\delta}_j^{l+1} \tag{13}$$

To train MGD NN, the above steps of gradient descent can be repeatedly taken to reduce cost function.

III. SIMULATION PLATFORM

Nowadays, every engine has its benchmark model to validate its controls and design model based controls [35]. The simulation object here is a bench mark model of a validated two spool turbo-fan engine (as shown in Fig. 1). This model is established by CLM (Component Level Model) method. The CLM consists of sub-component models of inlet, fan, compressor, combustion chamber, high compressor turbine, low compressor turbine and nozzle. And these sub components are modeled on the basis of basic thermal dynamics, cooperating with balances of flow, pressure and power. The station numbers for their location are stated in Tab.1.

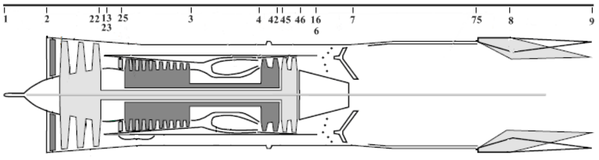


FIGURE 1. Structure of the turbo-fan engine.

TABLE 1. Engine reference stations.

Station	Location	42	High pressure turbine exit
1	Inlet entry	45	Low pressure turbine entry
2	Fan entry	46	Low pressure turbine exit
22	Fan exit	16	Bypass exit
13	Bypass entry	6	Core stream mixer entry
23	Core engine entry	7	Afterburner exit
25	Compressor entry	75	Exhaust nozzle entry
3	Burner entry	8	Exhaust nozzle throat
4	High pressure turbine entry	9	Exhaust nozzle exit

For more details of the CLM, some references can refer to [5] and [42].

The total thermodynamic parameters, air flow and bypass ratio are computed by the engine CLM as:

$$\begin{aligned} &[P_2, T_2, P_{22}, T_{22}, P_3, T_4, P_4, P_{42}, T_{42}, P_6, q_m, bp]^T \\ &\qquad = \begin{cases} f_{en}(H, Ma, W_{fb}, A_8, \alpha_c), & \text{open loop} \\ f_{en}(H, Ma, PLA), & \text{closed loop} \end{cases} \end{aligned}$$

where b_p is bypass ratio, P_2 and $T_2$are the total pressure and total temperature at fan entry, P_{22} and T_{22} are the total pressure and total temperature at fan exit, P_3 is the total pressure at combustion chamber entry, P_4 and T_4 are the total pressure and total temperature at high pressure turbine entry respectively, P_{42} and T_{42} are the total pressure and total temperature at high pressure turbine exit respectively, P_6 is total pressure a low pressure turbine exit, q_m is the engine air flow, PLA represents power level angle in the engine closed loop operation.

IV. ON-BOARD REAL-TIME ADAPTIVE DYNAMIC MODEL

In order to embody the engine dynamic characteristics completely, the inputs and outputs of the on-board dynamic adaptive turbofan engine model are set carefully as follows.

The inputs consist of current and past values of H, Ma, W_{fb}, A_8, the degradations of fan air flow $\Delta_{Wa_{2c}}$, compressor air flow $\Delta_{Wa_{25c}}$, combustor efficiency $\Delta_{\eta_{comb}}$, high pressure turbine efficiency $\Delta_{\eta_{Hturb}}$ low pressure turbine efficiency $\Delta_{\eta_{Lturb}}$ and past values of F, N_f, N_c, S_{mf}, S_{mc}, T_4. The outputs contain the current value of F, N_f, N_c, S_{mf}, S_{mc} and T_4. Then, the on-board real-time adaptive dynamic model of turbofan engine is formulated as:

$$\mathbf{y} = \mathbf{f}_{MGDNN}(\mathbf{x})$$

$$\times \left\{ \begin{array}{l} \mathbf{x} = [H(k), H(k-1), \ldots, H\,(k-m_1); \\ \quad Ma(k), Ma(k-1), \ldots, Ma\,(k-m_2); \\ \quad W_{fb}(k), W_{fb}(k-1), \ldots, W_{fb}\,(k-m_3); \\ \quad A_8(k), A_8(k-1), \ldots, A_8\,(k-m_4); \\ \quad \Delta_{Wa_{2c}}(k), \Delta_{Wa_{2c}}(k-1), \cdot\because . \Delta_{Wa_{2c}}\,(k-m_5); \\ \quad \Delta_{Wa_{25c}}(k), \Delta_{Wa_{25c}}(k-1), \cdot\because\cdot, \Delta_{Wa_{25c}}\,(k-m_6); \\ \quad \Delta_{\eta_{comb}}(k), \Delta_{\eta_{comb}}(k-1), \cdot\because\cdot, \Delta_{\eta_{comb}}\,(k-m_7); \\ \quad \Delta_{\eta_{Hturb}}(k), \Delta_{\eta_{Hturb}}(k-1), \cdot\because\cdot, \Delta_{\eta_{Hturb}}\,(k-m_8); \\ \quad \Delta_{\eta_{Lturb}}(k), \Delta_{\eta_{Lturb}}(k-1), \cdot\because\cdot, \Delta_{\eta_{Lturb}}\,(k-m_9); \\ \quad F(k-1), F(k-2), \cdots\ldots, F\,(k-m_{10}); \\ \quad N_f(k-1), N_f(k-2), \cdots, N_f\,(k-m_{11}); \\ \quad N_c(k-1), N_c(k-2), \cdot\because\cdot, N_c\,(k-m_{12}); \\ \quad S_{mf}(k-1), S_{mf}(k-2), \cdot\therefore\cdot\ , S_{mf}\,(k-m_{13}); \\ \quad S_{mc}(k-1), S_{mc}(k-2), \cdot\therefore\cdot\ , S_{mc}\,(k-m_{14}); \\ \quad T_4(k-1), T_4(k-2), \cdot\because\cdot, T_4\,(k-m_{15})] \\ \mathbf{y} = [F(k), N_f(k), N_c(k), S_{mf}(k), S_{mc}(k), T_4(k)]^T \end{array} \right. \tag{14}$$

where $m_1, m_2, \cdots, m_{15}$ is chosen by trial and error, $\mathbf{f}_{MGDNN}$ is the nonlinear mapping of MGD NN to express the engine dynamics.

V. MGD NN MODELING AND SIMULATIONS

To verify the effectiveness of the proposed method, an on-board adaptive dynamic model of turbofan engine with a large flight envelope is set up and validated. Moreover, the performance of the engine is bound to degrade during its service time. Hence, it's necessary to simulate the process while engine under degradation, and the above CLM has the functions to simulate component degradations. For comparison, the same simulation of the popular modeling method SVR [18] will also be utilized with a same data samples set herein.

The super-sonic cruise envelope is H (9km~13km), Ma(1.2~1.6), PLA (65°~75°). The degradations of engine are choose $\Delta_{Wa_{2c}}$, $\Delta_{Wa_{25c}}$, $\Delta_{\eta_{comb}}$, $\Delta_{\eta_{Hturb}}$ and $\Delta_{\eta_{Lturb}}$. They are all set as 0~5%. The training data of the MGD NN is obtained when the engine CLM is sufficiently excited in the super-sonic cruise envelope under engine degradation. As shown in Fig. 2, the numbers of the training data are more than 1.5 million.

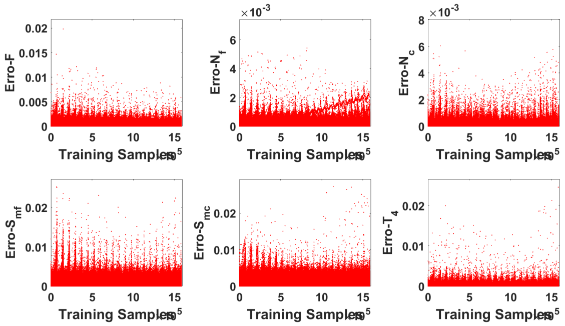
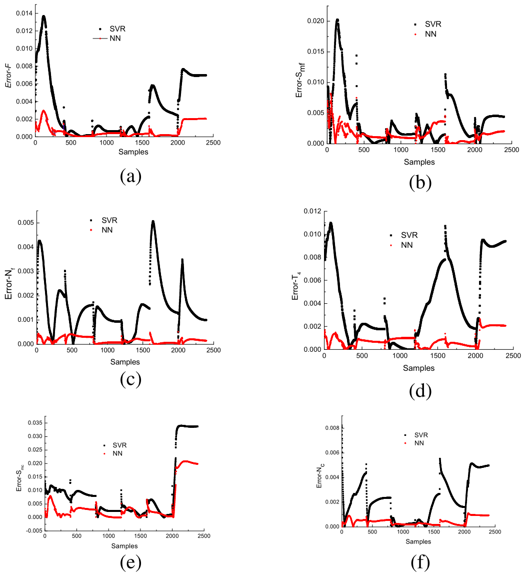


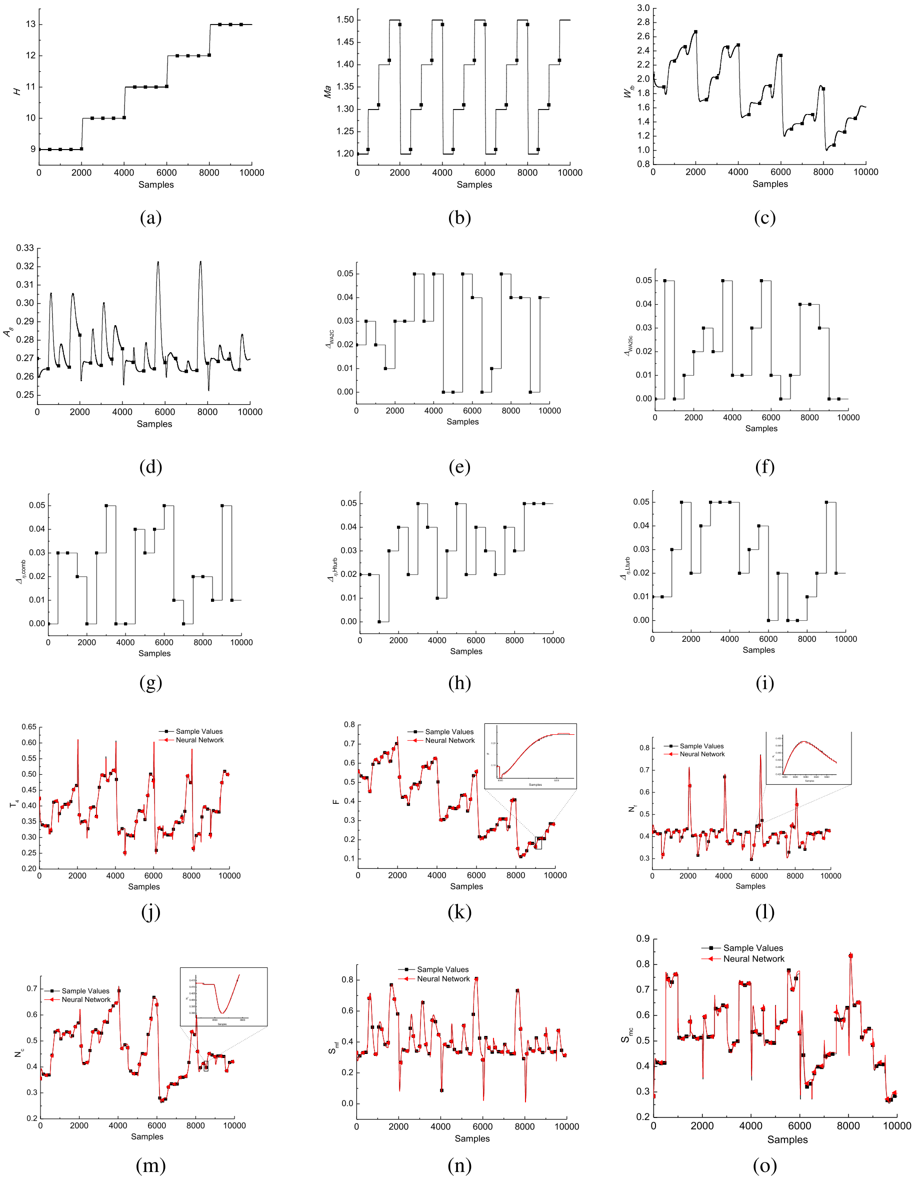
FIGURE 2. **The training errors of engine parameters using MGD NN.**



(a) (b) (c) (d) (e) (f)

FIGURE 3. **The Testing errors of engine parameters using MGD NN and SVR (H (10km~11km), *Ma* (1.4~1.5) and *PLA* (65°~75°)). a.The testing error of *F*. b.The testing error of S_{mf}. c.The testing error of N_f. d.The testing error of T_4. e.The testing error of S_{mc}. f. The testing error of N_c.**

Whereas for SVR method [18], the same huge data can not be trained for modeling. Therefore, the SVR usually needs to establish enough sub models for the super-sonic cruise envelope. The envelope is divided into 16 sub-envelopes here to establish a SVR engine model.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

FIGURE 4. The test of the on-board real-time adaptive dynamic model (in the whole super-sonic cruise envelope). a. The change of *H*. b. The change of *Ma*. c. The control variable of W_{fb}. d. The control variable of A_8 e. The change of $\Delta_{Wa_{2c}}$. f. The change of $\Delta_{Wa_{25c}}$. g. The change of $\Delta_{\eta_{comb}}$. h. The change of $\Delta_{\eta_{Hturb}}$. i. The change of $\Delta_{\eta_{Lturb}}$. j. The response of T_4. k. The response of *F*. l. The response of N_f. m. The response of N_c. n. The response of S_{mf}. o. The response of S_{mc}.

A. MODEL PRECISION VALIDATION

The training errors of the MGD NN are shown in figure 2. It is clearly shows that the errors of N_f, N_c are all within 8%, and the others are less than 3%. For the testing errors, the proposed method is compared with the SVR modeling method at the sub envelope H (10km∼11km), Ma (1.4∼1.5) and PLA (65°∼75°). The testing errors of MGD NN and SVR are respectively shown in figure 3. It can be seen that the proposed method has much higher precision than SVR. The main reason is that the proposed method is established based on a huge data, which can dramatically improve the generalization performance. On the contrary, the training effect of the SVR is significantly limited on the huge data.

Though debugging, the structure of MGD-NN is chosen as [6], [54], [70]. The min-batch number is chosen as 3000. The decay parameter is $\lambda = 10^{-5}$. Learning rate is $\alpha = 0.001$. Momentum factor is $\eta = 0.5$. The SVR has 1000 support vectors and $m_1, m_2, \cdots, m_{15}$ are set to 3.

Fig. 4 shows the predictive ability of the proposed method in the super-sonic cruise envelope. The value of F, N_f, N_c, S_{mf}, S_{mc} and T_4 have been normalized. Fig. 4 (a∼i) show the changes of the engine model inputs. Fig. 4 (j∼o) show the model predictive ability of the engine model outputs. It shows clear that the dynamic model has high precision for the whole super-sonic cruise envelope.

B. MODEL REAL-TIME PERFORMANCE VALIDATION

Tab.2 gives the data storage, computation complexity and average testing time of SVR and the proposed method.

TABLE 2. Comparison for MGD NN and SVR.

	Data storage (double)	Computation complexity	Average testing time
SVR	960,096	61,006	1.8ms
NN	4,276	8,476	0.31ms

The data storage of SVR is 960,096 (1000 support vectors × 54 dimensions + (1000 support vectors × 6 output) weight +1 bias×6 output) × 16 sub envelope).

The data storage of the proposed method is 4,276 ((54 dimension + 1 bias) × 70 numbers of hidden layer nodes + (70 numbers of hidden layer nodes +1 bias) × 6 output).

The computation complexity of SVR is 61,006 ((54 × 1000 + 1000) kernel support vector + (1000 weights + 1 bias) × 6) output (((54 dimension + 1 bias) × 1000 support vectors +1 bias) × 6 output).

And the computation complexity of MGD NN is 8,476 ((54 product+54 plus+1bias) × 70 numbers of hidden layer nodes + (70 product + 70 plus + 1bias) × 6output).

The testing environments of these two programs are entirely the same as: the operating system is Windows 7 Ultimate with Service Pack 1 (x64); the Central Processing Unit (CPU) is Intel(R) Core(TM) i5-4590 and its main frequency is 3.30GHz; the random-access memory (RAM) is 8G. The testing time of the SVR and the proposed method are 1.8 millisecond and 0.31 millisecond respectively.

Therefore, compared with the SVR, the proposed method not only has much higher testing precision, but also has less data storage, calculating amount and testing time. All of these performance indexes are the most importance indexes that decide whether it can be applied to become the on-board model or not. Hence, the proposed method can be applied as on-board real-time adaptive dynamic model.

VI. CONCLUSIONS

Through the simulation tests of on-board adaptive dynamic model of turbofan engine with the MGD NN and SVR method, some conclusions are followed by:

(1) The proposed method can be applied to a much larger envelope and has better generalization performance than the popular SVR. It is caused by the improvement of the optimization technology (min-batch gradient descent) and some adopted regularization technologies. The MGD NN could be established based on the huge data. The more training data is, the generalization performance is. Therefore, the proposed method can be applied to a bigger flight envelope with huge modeling data samples.

(2) Compared with the SVR, the proposed method has less data storage, computation complexity and testing time which are the main indexes for an on-board dynamic model. Therefore, the simulation results show that the proposed method is more suitable for establishing an on-board dynamic turbofan engine adaptive model.

The increasement of the hidden layers would greatly enhance the nonlinear fitting capacity and improve the model precious with less nodes of hidden layer [41]. Therefore, the future research works of the authors will concentrate on the deep neural network.

APPENDIX

NOMENCLATURE

Symbol	Explanation
H	Height
Ma	Mach number
PLA	Power level angle
W_{fb}	Fuel flow
A_8	Exhaust nozzle throat area
$\Delta_{Wa_{2c}}$	The degradation of fan air flow
$\Delta_{Wa_{25c}}$	The degradation of compressor air flow
$\Delta_{\eta_{comb}}$	The degradation of combustor efficiency
$\Delta_{\eta_{Hturb}}$	The degradation of high pressure turbine efficiency
$\Delta_{\eta_{Lturb}}$	The degradation of low pressure turbine efficiency
F	Engine thrust
T_4	High pressure turbine inlet temperature
N_f	Fan rotor speed
N_c	Compressor rotor speed
S_{mf}	Fan surge margin
S_{mc}	Compressor surge marge

REFERENCES

[1] S. Jianguo, "Prospects of the aero-engine control development in the early time of the 21st century," *J. Aerosp. Power*, vol. 16, no. 2, pp. 97–102, 2001.

[2] S. Garg, "Propulsion controls and diagnostics research in support of NASA aeronautics and exploration mission programs," NASA, Cleveland, OH, USA, Tech. Rep. TM 2011-216939, 2011.

[3] J. Csank, R. D. May, S. J. Litt, and T.-H. Guo, "Control design for a generic commercial aircraft engine," NASA, Cleveland, OH, USA, Tech. Rep. TM-2010-216811.

[4] S. Garg, "Propulsion controls and health management research at NASA Glenn Research Center," NASA, Cleveland, OH, USA, Tech. Rep. NASA/TM 2002-211590, 2002.

[5] Q. Zheng *et al.*, "On-board real-time optimization control for turbofan engine life extending," *Int. J. Turbo Jet-Engines*, vol. 34, no. 4, pp. 321–332, 2016, doi: 10.1515/tjj-2016-0015.

[6] D. L. Simon and J. B. Armstrong, "An integrated approach for aircraft engine performance estimation and fault diagnostics," ASME, New York, NY, USA, Tech. Rep. TGT-2012-69905, 2012.

[7] F. Sun, L. Miao, and H. Zhang, "A study on the installed performance seeking control for aero-propulsion under supersonic state," *Int. J. Turbo Jet-Engines*, vol. 33, no. 4, pp. 341–351, 2016.

[8] S. Garg, "Aircraft turbine engine control research at NASA Glenn research center," NASA, Cleveland, OH, USA, Tech. Rep. NASA/TM-2013-217821, 2013.

[9] T. Brotherton, A. Volponi, R. Luppold, and D. L. Simon, "eSTORM: Enhanced self tuning on-board real-time engine model," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2003, pp. 3075–3086.

[10] A. Vloponi and D. L. Simon "Enhanced self tuning on-board real-time model (eSTORM) for aircraft engine performance health tracking," NASA, Cleveland, OH, USA, Tech. Rep. NASA/CR-2008-215272, 2008.

[11] J. S. Litt, "An optimal orthogonal decomposition method for Kalman filter-based turbofan engine thrust estimation," *J. Eng. Gas Turbines Power*, vol. 130, no. 1, pp. 1–12, 2008.

[12] T. U. J. Gronstedt, "Identifiability in multi-point gas turbine parameter estimation problems," in *Proc. ASME Turbo Expo Power Land, Sea Air Amer. Soc. Mech. Eng.*, 2002, pp. 9–17.

[13] C.-W. Fei and G.-C. Bai, "Wavelet correlation feature scale entropy and fuzzy support vector machine approach for aeroengine whole-body vibration fault diagnosis," *Shock Vib.*, vol. 20, no. 2, pp. 341–349, 2013.

[14] Q. Zheng *et al.*, "On-board real-time optimization control for turbofan engine thrust under flight emergency condition," *Proc. Inst. Mech. Eng. I, J. Syst. Control Eng.*, vol. 231, no. 7, pp. 554–566, 2017.

[15] H. C. Qu and X. B. Ding, "Civil aeroengine fault diagnosis based on fuzzy least square support vector machine," *Appl. Mech. Mater.*, vols. 130–134, pp. 2047–2050, Oct. 2012.

[16] M. M. Adankon and M. Cheriet, "Support vector machine," in *Encyclopedia of Biometrics*. Boston, MA, USA: Springer, 2015, pp. 1504–1511.

[17] Y. Zhao and J. Sun, "Recursive reduced least squares support vector regression," *Pattern Recognit.*, vol. 42, no. 5, pp. 837–842, 2009.

[18] W. Jiankang, Z. Haibo, Y. Changkai, D. Shujing, and H. Xianghua, "An adaptive turbo-shaft engine modeling method based on PS and MRR-LSSVR algorithms," *Chin. J. Aeronaut.*, vol. 26, no. 1, pp. 94–103, 2013.

[19] B. Gu, V. S. Sheng, Z. Wang, D. Ho, S. Osman, and S. Li, "Incremental learning for ν-support vector regression," *Neural Netw.*, vol. 67, pp. 140–150, Jul. 2015.

[20] Z. Qiangang, Z. Haibo, and L. Yongjin, "Research on simplex B-splines algorithm in on-board steady state modeling of turbofan engines," *J. Propuls. Technol.*, vol. 12, no. 36, pp. 1887–1894, 2015.

[21] Z. Zhao, X. Wang, and C. Zhang, "Neural network based boundary control of a vibrating string system with input deadzone," *Neurocomputing*, vol. 275, pp. 1021–1027, Jan. 2017.

[22] M. Maggiore, R. Ordonez, K. M. Passino, S. Adibhatlam, "Estimator design in jet engine applications," *Eng. Appl. Artif. Intell.*, vol. 16, nos. 7–8, pp. 579–593, 2003.

[23] Y. Oussar, I. Rivals, L. Personnaz, and G. Dreyfus, "Training wavelet networks for nonlinear dynamic input-output modeling," *Neurocomputing*, vol. 20, no. 1, pp. 173–188, 1998.

[24] C. Yang, X. Wang, Z. Li, Y. Li, and C.-Y. Su, "Teleoperation control based on combination of wave variable and neural networks," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 8, pp. 2125–2136, Aug. 2016.

[25] S. Hussain, M. Mokhtar, and J. M. Howe, "Sensor failure detection, identification, and accommodation using fully connected cascade neural network," *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1683–1692, Mar. 2015.

[26] K. Xu, M. Xie, L. C. Tang, and S. L. Ho, "Application of neural networks in forecasting engine systems reliability," *Appl. Soft Comput.*, vol. 2, no. 4, pp. 255–268, 2003.

[27] M. Zedda and R. Singh, "Gas turbine engine and sensor fault diagnosis using optimization techniques," *J. Propuls. Power*, vol. 18, no. 5, pp. 1019–1025, 2002.

[28] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[29] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[31] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[32] Y. Jiang, C. Yang, S.-L. Dai, and B. Ren, "Deterministic learning enhanced neutral network control of unmanned helicopter," *Int. J. Adv. Robot. Syst.*, vol. 13, no. 6, pp. 1–12, 2016.

[33] Z. Zhao, Y. Liu, W. He, and F. Luo, "Adaptive boundary control of an axially moving belt system with high acceleration/deceleration," *IET Control Theory Appl.*, vol. 10, no. 11, pp. 1299–1306, 2016.

[34] C. Yang, Y. Jiang, Z. Li, W. He, and C.-Y. Su, "Neural control of bimanual robots with guaranteed global stability and motion precision," *IEEE Trans. Ind. Informat.*, vol. 3, no. 13, pp. 1162–1171, Jun. 2017.

[35] W. He, Y. Chen, and Y. Zhao, "Adaptive neural network control of an uncertain robot with full-state constraints," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 620–629, Mar. 2016.

[36] W. He, Y. Dong, and C. Sun, "Adaptive neural impedance control of a robotic manipulator with input saturation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 3, pp. 334–344, Mar. 2016.

[37] S. Zhang, A. E. Choromanska, and Y. LeCun, "Deep learning with elastic averaging SGD," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 685–693.

[38] J. Duchi, E. Hazan, and Y. Singer, "Adaptive sub-gradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Jul. 2011.

[39] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 9–48.

[40] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[41] Y. Bengio, I. J. Goodfellow, and A. Courville, "Deep learning," in *An MIT Press Book in Preparation. Draft Chapters*. Cambridge, MA, USA: MIT Press, 2015. [Online]. Available: http://www.iro.umontreal.ca/?bengioy/dlbook

[42] S. Jianguo, V. Vasilyev, and B. Ilyasov, *Advanced Multivariable Control Systems of Aeroengines*. Beijing, China: Beijing University of Aeronautics and Astronautics, 2005.

QIANGANG ZHENG received the Ph.D. degree from the Nanjing University of Aeronautics and Astronautics (NUAA), in 2018. He is currently a Teacher with the College of Energy and Power Engineering, NUAA. His research interests include modeling, fault diagnosis, optimization control, and model predictive control of aero-engines.

HAIBO ZHANG received the Ph.D. degree from the Nanjing University of Aeronautics and Astronautics (NUAA), in 2005. He is currently a Professor with the College of Energy and Power Engineering, NUAA. His main research interests include modeling, fault diagnosis, and the optimization control of aero-engines.

YONGJIN LI is currently pursuing the Ph.D. degree in aerospace propulsion theory and engineering at the College of Energy and Power Engineering, Nanjing University of Aeronautics and Astronautics. His research interests include modeling, fault diagnosis, and the optimization control of aero-engines.

ZHONGZHI HU received the B.S. degree in automatic control from the Beijing Institute of Technology, China, the M.S. degree, and the Ph.D. degree in electrical engineering from The University of British Columbia, Canada. He has published over 40 refereed conference and journal papers, and holds five U.S. patents. His research interests are in gas turbine modeling, simulation, controls, and PHM. He has almost 30 years of extensive experience in modeling and simulation, controls and PHM for gas turbine, wind turbine, and other industry applications, as a Researcher and an Engineer, including 15 years of research and development and engineering positions working for General Electric. He is currently a Professor with the Nanjing University of Aeronautics and Astronautics, Nanjing, China.