Chinese Journal of Aeronautics, 2013,26(1): 94–103

Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com

An adaptive turbo-shaft engine modeling method based on PS and MRR-LSSVR algorithms

Wang Jiankang [a], Zhang Haibo [a,*], Yan Changkai [a], Duan Shujing [b], Huang Xianghua [a]

[a] *College of Energy and Power Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*
[b] *Aviation Motor Control System Institute, Aviation Industry Corporation of China, Wuxi 214063, China*

Received 22 October 2011; revised 20 December 2011; accepted 19 March 2012
Available online 16 January 2013

KEYWORDS

Adaptive engine model;
Least square support vector regression machine;
Modeling method;
Parameter selection;
Turbo-shaft engine

Abstract In order to establish an adaptive turbo-shaft engine model with high accuracy, a new modeling method based on parameter selection (PS) algorithm and multi-input multi-output recursive reduced least square support vector regression (MRR-LSSVR) machine is proposed. Firstly, the PS algorithm is designed to choose the most reasonable inputs of the adaptive module. During this process, a wrapper criterion based on least square support vector regression (LSSVR) machine is adopted, which can not only reduce computational complexity but also enhance generalization performance. Secondly, with the input variables determined by the PS algorithm, a mapping model of engine parameter estimation is trained off-line using MRR-LSSVR, which has a satisfying accuracy within 5‰. Finally, based on a numerical simulation platform of an integrated helicopter/turbo-shaft engine system, an adaptive turbo-shaft engine model is developed and tested in a certain flight envelope. Under the condition of single or multiple engine components being degraded, many simulation experiments are carried out, and the simulation results show the effectiveness and validity of the proposed adaptive modeling method.

1. Introduction

Unlike a fixed-wing aircraft, a helicopter has a direct mechanical link with its turbo-shaft engine, so the coupling reaction between the helicopter and the engine is obvious, and a dynamic response of the engine will give a significant influence on the helicopter's agility and conventional performance.[1,2] Usually, an engine may inhale large quantities of sand or vapor with salt when a helicopter works in terrible conditions, such as near the ground or a sea-level state. Over time, the deteriorations of engine components caused by various physical faults (e.g., foreign object damage, blade erosion and corrosion, worn seals, excess clearances, and so on) may be ever-increasing so that the operation and control of the engine are inaccurate or misleading, even having a serious influence on the helicopter's flight quality. Therefore, it is remarkably attractive to build an onboard adaptive engine model, which

* Corresponding author. Tel.: +86 25 84892201 8401.
E-mail addresses: wangjiankang16@163.com (J. Wang), zh_zhhb@163.com (H. Zhang).
Peer review under responsibility of Editorial Committe of CJA.

1000-9361
http://dx.doi.org/10.1016/j.cja.2012.12.020

can detect engine components' deteriorations and meantime provide engine performance tracking in real time, including turbine rotor speed, stall margin and power, etc.[3,4]

At home, there have been few formal studies about adaptive modeling methods of turbo-shaft engines. Only Ref. [4] introduces a technique using Kalman filter based on rotor/turbo-shaft engine. Kalman filter is the most popular method for engine parameter estimation abroad, which can acquire a set of tuners that would adapt a state variable model to match actual observations (hence driving the residuals to zero, on the average), derived by the residuals formed by the output of an embedded piecewise simplified engine model and the actual observed measurements.[5] However, due to multi-step iterative calculations, Kalman filter must cost much more time to achieve an optimal estimation. In order to improve real-time ability, some literatures have proposed another method based on data, i.e., neural network, which can finish engine parameter estimation with a short period of computing time.[6] But neural network also has obvious drawbacks: its solution is easier to trap to the local extremum, and its generalization ability also needs to be improved. As a consequence, the applications of Kalman filter and neural network to parameter estimation of aero-engines is somewhat restricted.

In recent decades, another algorithm based on data, named support vector machine (SVM), has been applied in the field of machine learning.[7] In comparison with neural network, SVM is built on a so-called structural risk minimization principle which holds good generalization performance, but its training complexity burden, called the quadratic programming, is expensive. As a variation, the least square support vector machine (LSSVM) has been proposed,[8] which uses equality constraints instead of inequality ones and squared errors as the loss function to mitigate the training complexity. However, LSSVM is not sparse compared with normal SVM, which blocks its predicted speed. On the basis of LSSVM, various derived algorithms such as SMO-based pruning methods for sparse LSSVM (SMO-LSSVM), fast sparse approximation for LSSVM (FSA-LSSVM), reduced least squares support vector regression (R-LSSVR), recursive reduced least squares support vector regression (RR-LSSVR), etc., have been proposed to realize the sparseness or improve real-time ability of algorithms.[9–11] RR-LSSVR gains advantage over common sparse tricks,[10] since it is involved in the whole constraints generated by all training patterns after combining the iterative strategy[11] with the reduced technique[9] in the modeling process. Meantime, RR-LSSVR needs smaller scale subset, which shortens predicting time and strengthens sparseness. Inspired by above illustrations, a deterioration estimator design scheme based on RR-LSSVR is taken into account. However, RR-LSSVR is only suitable for single-output systems. For a multi-output problem, it needs many separate RR-LSSVR modules to perform, which makes the algorithm more complex and increases the number of training patterns. Additionally, it cannot consider the comprehensive actions of multi-output variables to select training patterns. In this paper, an algorithm, called multi-input multi-output recursive reduced least square support vector regression (MRR-LSSVR), is proposed. Compared with RR-LSSVR, MRR-LSSVR can select less and better support vectors to solve multi-output problems owing to considering the comprehensive actions of multi-output variables.

With the aid of MRR-LSSVR, online and real-time nonlinear relationship between an engine's measurable variables (i.e., dependent variables) and its performance deteriorations can be created. However, it is still a key problem which dependent variables can be selected as the inputs of the parameter estimation module. The choice directly affects whether the adaptive model can accurately track or not.

In recent years, parameter selection (PS) has become the focus of numerous research studies in areas where datasets with tens or hundreds of variables are available. The main drawback in PS resides in its combinatorial nature, turning into a non-deterministic and poly-nominal hard problem. The use of an exhaustive search method is unpractical for large numbers of variables. Therefore the design of efficient methods and reliable criteria becomes crucial in the data analysis workflow. Generally, PS algorithms are grouped into two main categories: (1) filter methods, which select variables independently to the predictor, and (2) wrapper methods, in which the way of selecting variables is related to the predictor's performance. Because the purpose of PS is to eliminate irrelevant variables to enhance the generalization performance and curtail the computational complexity, the wrapper methods outperform the filter ones. Ref. [12] put forward a wrapper criterion which ranks variables based on the generalization ability of LSSVR. In this paper, based on multi-input multi-output LSSVR, a PS algorithm with the wrapper criterion for an adaptive engine model is proposed.

As a result, utilizing PS and MRR-LSSVR algorithms, an adaptive turbo-shaft engine model is developed in a certain flight envelope. Through case studies on an integrated helicopter/turbo-shaft engine numerical simulation platform with a high fidelity, it is proved that the established adaptive engine model can track a real engine rapidly not only in a steady state but also during a transient operation, and perform parameter estimation for single or multiple engine performance deteriorations.

2. Simulation platform

An integrated helicopter/turbo-shaft engine system[13] is a sort of cascade systems which have complex coupling relationships. This system is used as the simulation platform in this paper which mainly consists of the following four parts: open-loop model of helicopter, flight controller, open-loop model of turbo-shaft engine, and engine controller. The schematic diagram of the integrated system is depicted in Fig. 1, and the helicopter model and turbo-shaft engine model are introduced as follows.

2.1. Helicopter model

In this paper, a real-time helicopter model is built based on UH-60A Black Hawk helicopter data. This model is an unsteady nonlinear aerodynamic model, which contains fuselage, main rotor, tail rotor, horizontal tail, and vertical tail. Among these components, there are many complex close-coupling actions (see Fig. 1). Through a series of equilibrium computations and calculations of dynamic equations, such as balancing rotor model, fuselage model, and so on, the required power of the helicopter can be worked out and sent to the turbo-shaft engine together with some flight parameters. In Ref. [13], a great amount of tests were performed to check the accuracy of this helicopter model. The results prove that this model

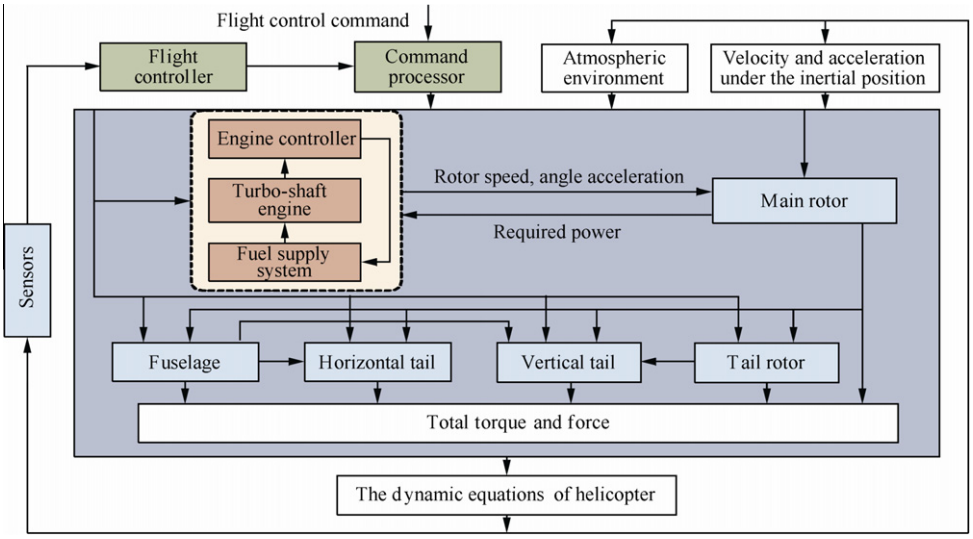

Fig. 1 Schematic diagram of the integrated helicopter/engine system.

has a good fidelity and is able to carry out digital flight simulations of routine missions.

2.2. Turbo-shaft engine model

Fig. 2 depicts the structure of a component-level model of the turbo-shaft engine, which is established based on T700 engine data set. In Fig. 2, the engine station numbers represent the inlet or outlet of engine components.

During the engine modeling procedure, every component model of the engine is created using the engine's thermodynamic characteristic and typical experimental data at first. Next, with the benefits of power balance, flow equilibrium, pressure equilibrium, and rotor dynamics equations, the balance equations among engine components can be constructed and calculated. Finally, the Newton–Raphson iterative method and once-pass-through algorithm are adopted in order to solve the steady-state engine model and dynamic engine model, respectively.

3. Adaptive engine model

Fig. 3 shows the structure diagram of the adaptive engine model. This system consists of a host computer and a slave computer of the two-level system. The host computer is to simulate the actual integrated helicopter/turbo-shaft engine system. The slave computer is to replace the adaptive engine model, which is used for tracking the engine's state and estimating its performance deterioration in real time, including a MRR-LSSVR estimation module, an onboard nonlinear engine model, and a nonlinear calculation module, where the input of the estimation module is determined by a PS algorithm. In Fig. 3, HPP represents required power of rotor and SMC the compressor stall margin.

When the integrated helicopter/engine system is working under non-nominal condition, the outputs of the MRR-LSSVR module, i.e., the values of engine performance deteriorations, will be calculated, and the onboard engine model will be corrected immediately. In this section, firstly, the multi-input multi-output LSSVR algorithm is concisely described, and then the PS algorithm with the wrapper criterion, the MRR-LSSVR algorithm, and the design of the parameter estimation module are introduced one by one as follows.

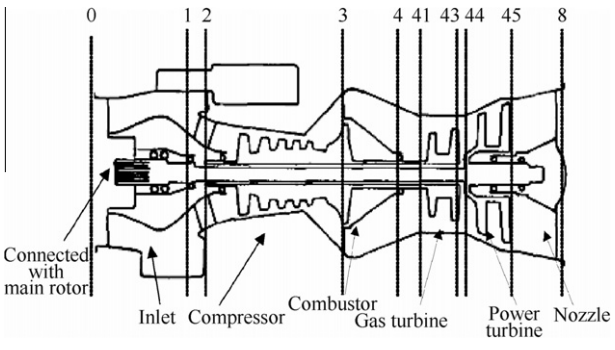

Fig. 2 Structure of the turbo-shaft engine.

3.1. LSSVR algorithm

For a multi-input multi-output system, considering the training sample set $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$ of size N, where $\boldsymbol{x}_i$ is the input pattern, $\boldsymbol{y}_i = [y_{i,1}\ y_{i,2} \cdots y_{i,M}]$ is the corresponding target, and M is the number of output variables, the mathematical model of LSSVR is obtained[14,15]:

$$\begin{aligned} \min_{\boldsymbol{w}_m, e_{i,m}} J(\boldsymbol{w}_m, e_{i,m}) &= \frac{1}{2}\sum_{m=1}^{M} \boldsymbol{w}_m^{\mathrm{T}} \boldsymbol{w}_m + \frac{\gamma}{2}\sum_{m=1}^{M}\sum_{i=1}^{N} e_{i,m}^2 \\ \text{s.t. } y_{i,m} &= \boldsymbol{w}_m^{\mathrm{T}} \varphi_m(\boldsymbol{x}_i) + b_m + e_{i,m} \\ (i &= 1, 2, \cdots, N;\ m = 1, 2, \cdots, M) \end{aligned} \tag{1}$$

where $\boldsymbol{w}_m$ represents the model complexity, $e_{i,m}$ represents the error between actual output and predictive value, b_m is the offset, $\gamma \in \mathbf{R}^+$ is a regularization parameter which can control the tradeoff between the flatness of the model and the closeness to the training data, and $\varphi_m(\cdot)$ is a nonlinear mapping which can transform the input data into a high-dimensional feature

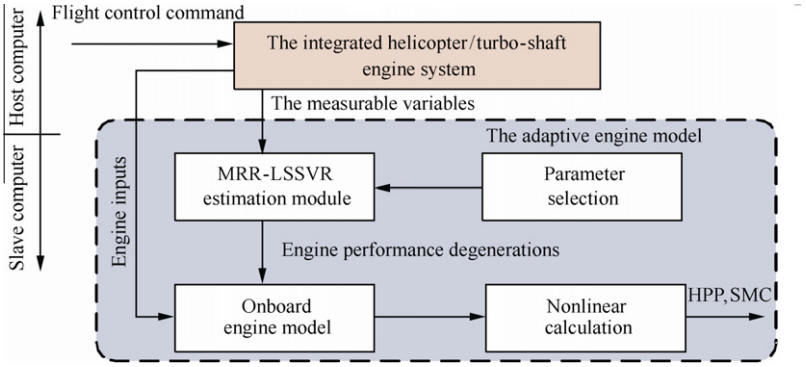

Fig. 3 Diagram of the adaptive engine model.

space. In order to solve the above optimization problem, the Lagrange function without constraints can be constructed as:

$$L(\boldsymbol{w}_m, b_m, e_{i,m}, \alpha_{i,m}) = J - \sum_{m=1}^{M}\sum_{i=1}^{N}\alpha_{i,m}\{\boldsymbol{w}_m^{\mathrm{T}}\varphi_m(\boldsymbol{x}_i) + b_m + e_{i,m} - y_{i,m}\} \tag{2}$$

where $\alpha_{i,m}$ is Lagrange multiplier. Hence, Karush–Kuhn–Tucker (KKT) conditions of Eq. (1) can be expressed as:

$$\begin{cases} \dfrac{\partial L}{\partial \boldsymbol{w}_m} = \boldsymbol{0} \rightarrow \boldsymbol{w}_m = \displaystyle\sum_{i=1}^{N}\alpha_{i,m}\varphi_m(\boldsymbol{x}_i) \\ \dfrac{\partial L}{\partial b_m} = 0 \rightarrow \displaystyle\sum_{i=1}^{N}\alpha_{i,m} = 0 \\ \dfrac{\partial L}{\partial e_{i,m}} = 0 \rightarrow \alpha_{i,m} = \gamma e_{i,m} \\ \dfrac{\partial L}{\partial \alpha_{i,m}} = 0 \rightarrow \boldsymbol{w}_m^{\mathrm{T}}\varphi_m(\boldsymbol{x}_i) + b_m + e_{i,m} - y_{i,m} = 0 \end{cases} \tag{3}$$

After eliminating $\boldsymbol{w}_m$ and $e_{i,m}$, the following linear function is obtained:

$$\begin{bmatrix} 0 & \boldsymbol{1}^{\mathrm{T}} \\ \boldsymbol{1} & \overline{\boldsymbol{K}} \end{bmatrix}\begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{Y} \end{bmatrix} \tag{4}$$

where

$$\boldsymbol{Y} = [\,\boldsymbol{Y}_1 \quad \boldsymbol{Y}_2 \quad \ldots \quad \boldsymbol{Y}_M\,] = \begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,M} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,M} \\ \vdots & \vdots & & \vdots \\ y_{N,1} & y_{N,2} & \cdots & y_{N,M} \end{bmatrix}$$

$$\boldsymbol{\alpha} = [\,\boldsymbol{\alpha}_1 \quad \boldsymbol{\alpha}_2 \quad \ldots \quad \boldsymbol{\alpha}_M\,] = \begin{bmatrix} \boldsymbol{\alpha}_{1,1} & \boldsymbol{\alpha}_{1,2} & \ldots & \boldsymbol{\alpha}_{1,M} \\ \alpha_{2,1} & \alpha_{2,2} & \ldots & \alpha_{2,M} \\ \vdots & \vdots & & \vdots \\ \alpha_{N,1} & \alpha_{N,2} & \ldots & \alpha_{N,M} \end{bmatrix}$$

$$\boldsymbol{b} = [\,b_1 \quad b_2 \quad \cdots \quad b_M\,]$$

$$\boldsymbol{1} = [\,1 \quad 1 \quad \cdots \quad 1\,]_{N\times 1}$$

$$\boldsymbol{0} = [\,0 \quad 0 \quad \cdots \quad 0\,]_{M\times 1}$$

$\overline{\boldsymbol{K}}$ is the matrix whose element $\overline{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi^{\mathrm{T}}(\boldsymbol{x}_i)\varphi(\boldsymbol{x}_j) + \delta_{ij}/\gamma$ with $\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$. Among all the kernel functions, Gaussian kernel $k(\boldsymbol{x}_i,\boldsymbol{x}_j) = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/(2\upsilon^2))$ is the most popular choice, where υ represents the kernel parameter. In this paper, Gaussian kernel is used, and after solving Eq. (4), $\boldsymbol{\alpha}$ and $\boldsymbol{b}$ can be computed. For a new pattern $\boldsymbol{x}$, the predictor of multi-input multi-output LSSVR is gotten as follows[14,15] :

$$f_m(\boldsymbol{x}) = \sum_{i=1}^{N}\alpha_{i,m}k(\boldsymbol{x}_i, \boldsymbol{x}) + b_m \ (m = 1, 2, \ldots, M) \tag{5}$$

3.2. PS algorithm

As for Eq. (5), a problem is still not solved, that is, which measurable variables of the engine will be used as input variables for the adaptive module. In this section, a wrapper criterion for ranking variables is introduced.[12]

After combining Eqs. (2) and (3), the equation without constraints for the Wolfe dual optimization problem can be attained:

$$\min\left\{\overline{L}(b_m, \alpha_{i,m}) = \sum_{m=1}^{M}\left[\frac{1}{2}\sum_{i,j=1}^{N}\alpha_{i,m}\alpha_{j,m}k(\boldsymbol{x}_i,\boldsymbol{x}_j) + \frac{1}{2\gamma}\sum_{i=1}^{N}\alpha_{i,m}^2 - \sum_{i=1}^{N}\alpha_{i,m}y_{i,m} + b_m\sum_{i=1}^{N}\alpha_{i,m}\right]\right\} \tag{6}$$

And then Eq. (6) can be reformulated for convenience:

$$\min\left\{\overline{L}(b_m, \boldsymbol{\alpha}_m) = \sum_{m=1}^{M}\left[\frac{1}{2}[\,b_m \quad \boldsymbol{\alpha}_m^{\mathrm{T}}\,]\begin{bmatrix} 0 & \boldsymbol{1}^{\mathrm{T}} \\ \boldsymbol{1} & \overline{\boldsymbol{K}} \end{bmatrix}\begin{bmatrix} b_m \\ \boldsymbol{\alpha}_m \end{bmatrix} - [\,b_m \quad \boldsymbol{\alpha}_m^{\mathrm{T}}\,]\begin{bmatrix} 0 \\ \boldsymbol{Y}_m \end{bmatrix}\right]\right\} \tag{7}$$

Next, let $\bar{\boldsymbol{\alpha}}_m = [\,b_m \quad \boldsymbol{\alpha}_m^{\mathrm{T}}\,]^{\mathrm{T}}$ and $\mathrm{d}\overline{L}/\mathrm{d}\bar{\boldsymbol{\alpha}}_m = \boldsymbol{0}$, where $m = 1,2,\ldots,M$, the following equation is obtained:

$$\begin{bmatrix} b_m \\ \boldsymbol{\alpha}_m \end{bmatrix} = \begin{bmatrix} 0 & \boldsymbol{1}^{\mathrm{T}} \\ \boldsymbol{1} & \overline{\boldsymbol{K}} \end{bmatrix}^{-1}\begin{bmatrix} 0 \\ \boldsymbol{Y}_m \end{bmatrix} \tag{8}$$

Finally, substituting Eq. (8) into Eq. (7), the optimal value $\overline{L}^*$ of Eq. (7) is gotten in the following:

$$\overline{L}^* = \sum_{m=1}^{M}\left\{-\frac{1}{2}\begin{bmatrix}0 & \boldsymbol{Y}_m^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}0 & \mathbf{1}^{\mathrm{T}}\\ \mathbf{1} & \overline{\boldsymbol{K}}\end{bmatrix}^{-1}\begin{bmatrix}0\\ \boldsymbol{Y}_m\end{bmatrix}\right\}$$
$$= \sum_{m=1}^{M}\left\{-\frac{1}{2}\begin{bmatrix}0 & \boldsymbol{Y}_m^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}b_m\\ \boldsymbol{\alpha}_m\end{bmatrix}\right\} = -\frac{1}{2}\sum_{m=1}^{M}\boldsymbol{Y}_m^{\mathrm{T}}\boldsymbol{\alpha}_m \tag{9}$$

If the ith variable is removed, $\overline{L}^*(i) = -\frac{1}{2}\sum_{m=1}^{M}\left(\boldsymbol{Y}_m^{\mathrm{T}}\boldsymbol{\alpha}_m(i)\right)$, where $\boldsymbol{\alpha}_m(i)$ is the solution of Eq. (8) without the ith input variable. Thus, a wrapper criterion for ranking variables can be put forward[12] :

$$\Delta(i) = \sum_{m=1}^{M}\left|\boldsymbol{Y}_m^{\mathrm{T}}(\boldsymbol{\alpha}_m - \boldsymbol{\alpha}_m(i))\right| \tag{10}$$

In the process of computation, if the value of $\Delta(i)$ is smaller than the value of $\Delta(j)(j \neq i)$, the ith variable is considered to make less contribution to the optimal value $\overline{L}^*$ than the jth variable. As for the turbo-shaft engine in this paper, there are 20 measurable variables (i.e., input $\boldsymbol{x}$) for the selection, and output variables are $\boldsymbol{y} = [y_1 \quad y_2 \quad y_3] = [D_{\mathrm{com}} \quad D_{\mathrm{gas}} \quad D_{\mathrm{pow}}]$, where D_{com} represents compressor flow deterioration, D_{gas} represents gas turbine efficiency deterioration, and D_{pow} represents power turbine flow deterioration. If $\overline{L}$ acquires the optimal value $\overline{L}^*$, the estimation accuracy of the variable (D_{com}, D_{gas}, D_{pow}) will be the best. And then, by removing the ith input variable, $\Delta(i)$ can be calculated based on Eq. (10). According to the sequence $\Delta(i)$, we can rank the variables, discard those variables with small values of $\Delta(i)$, and select input variables which make more contributions to $\overline{L}^*$ (see Table 1) as input variables of the MRR-LSSVR estimation module.

During the selecting process, it is needed to analyze the variables' measurability at first, which shows that the variables' values are easily acquired. In the list of Table 1, the values of T_4, p_4, T_{41}, and p_{41} are hardly acquired because the sensor is generally not allowed to set in high-temperature parts. The remaining variables are 20 measurable variables. Considering T_2 and P_2 are worked out with H and V_x, and T_8 is equal to T_{46}, T_2, p_2, and T_8 also have no use for selection. And then, based on the above PS algorithm, we can choose the variables which have more contributions to variable estimation from all remaining variables. At last, utilizing the debugging method, eight input variables are selected, which are $\boldsymbol{x} = [x_1 \quad x_2 \quad \ldots \quad x_8] = [H \quad V_x \quad N_{\mathrm{g}} \quad N_{\mathrm{p}} \quad p_3 \quad p_{44} \quad T_{44} \quad T_{45}]$.

Table 1 List of variables selection.

Symbol	Definition	Choice
H (m)	Flight altitude	√
V_x (m · s^{-1})	Forward speed	√
N_{g} (%)	Gas turbine rotor relative speed	√
N_{p} (%)	Power turbine rotor relative speed	√
T_2 (K)	Compressor inlet total temperature	×
p_2 (Pa)	Compressor inlet total pressure	×
T_3 (K)	Compressor outlet total temperature	×
p_3 (Pa)	Compressor outlet total pressure	√
T_4 (K)	Combustor outlet total temperature	×
p_4 (Pa)	Combustor outlet total pressure	×
T_{41} (K)	Gas turbine inlet total temperature	×
p_{41} (Pa)	Gas turbine inlet total pressure	×
T_{42} (K)	Gas turbine outlet total temperature	×
p_{42} (Pa)	Gas turbine outlet total pressure	×
T_{43} (K)	Air-entraining section total temperature after gas turbine	×
p_{43} (Pa)	Air-entraining section total pressure after gas turbine	×
T_{44} (K)	Power turbine inlet total temperature	√
p_{44} (Pa)	Power turbine inlet total pressure	√
T_{45} (K)	Power turbine outlet total temperature	×
p_{45} (Pa)	Power turbine outlet total pressure	√
T_{46} (K)	Air-entraining section total temperature after power turbine	×
p_{46} (Pa)	Air-entraining section total pressure after power turbine	×
T_8 (K)	Nozzle outlet total temperature	×
p_8 (Pa)	Nozzle outlet total pressure	×

3.3. MRR-LSSVR algorithm

From Eq. (5), it is clear that every training sample is support vector, so LSSVR is not sparse. Using the reduction strategies presented in Ref. [9], after letting $\boldsymbol{w}_m = \sum_{i\in S}\alpha_{i,m}k(\boldsymbol{x}_i,\cdot)$ and substituting it into Eq. (1) where the subset $\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i\in S} \subset \{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i=1}^{N}$, and S is the index class of the useful subset, we get the corresponding form as follows:

$$\min\left\{L(b_m,\boldsymbol{\alpha}_m) = \frac{1}{2}\sum_{m=1}^{M}\boldsymbol{\alpha}_m^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\alpha}_m + \frac{\gamma}{2}\sum_{m=1}^{M}\sum_{i=1,i\in S}^{N}\left(y_{i,m} - \sum_{j\in S}\alpha_{j,m}\varphi_m^{\mathrm{T}}(\boldsymbol{x}_j)\varphi_m(\boldsymbol{x}_j) - b_m\right)^2\right\} \tag{11}$$

where $\boldsymbol{K}_{i,j} = k(\boldsymbol{x}_i,\boldsymbol{x}_j), i,j \in S$. Let $\partial L/\partial b_m = 0$ and $\partial L/\partial \alpha_{i,m} = 0$, the following linear equation is gained:

$$(\boldsymbol{R} + \boldsymbol{Z}\boldsymbol{Z}^{\mathrm{T}})\begin{bmatrix}\boldsymbol{b}\\ \boldsymbol{\alpha}\end{bmatrix} = \boldsymbol{Z}\boldsymbol{Y} \tag{12}$$

where $\boldsymbol{R} = \begin{bmatrix}0 & \mathbf{0}^{\mathrm{T}}\\ \mathbf{0} & \boldsymbol{K}/\gamma\end{bmatrix}$ and $\boldsymbol{Z} = \begin{bmatrix}\mathbf{1}^{\mathrm{T}}\\ \widehat{\boldsymbol{K}}\end{bmatrix}$ with $\widehat{\boldsymbol{K}} = k(\boldsymbol{x}_i,\boldsymbol{x}_j)$, $i \in S$, $j = 1,2,\ldots,N$. In Eq. (13), if $\boldsymbol{R} + \boldsymbol{Z}\boldsymbol{Z}^{\mathrm{T}}$ is singular, a small change $\boldsymbol{R} + \boldsymbol{Z}\boldsymbol{Z}^{\mathrm{T}} + 10^{-8}\boldsymbol{I}$ will find the solution.[10] Therefore, for a new sample $\boldsymbol{x}$, we can obtain R-LSSVR as follows:

$$f_m(\boldsymbol{x}) = \sum_{i\in S}\alpha_{i,m}k(\boldsymbol{x}_i,\boldsymbol{x}) + b_m \ (m = 1,2,\ldots,M) \tag{13}$$

Because the subset $\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i\in S}$ in Eq. (13) is selected randomly, R-LSSVR is lack of sparseness or the generalization ability is degraded, so it is important to select the subset. With the iterative strategy presented in Ref. [11], we can pick up the patterns which make more contributions to the optimization target to form the subset.[10] Reformulating Eq. (11), we have:

$$\min\left\{L = \sum_{m=1}^{M}\left[\begin{bmatrix}b_m & \boldsymbol{\alpha}_m^{\mathrm{T}}\end{bmatrix}\left(\begin{bmatrix}0 & \mathbf{0}^{\mathrm{T}}\\ \mathbf{0} & \boldsymbol{K}/\gamma\end{bmatrix} + \begin{bmatrix}\mathbf{1}^{\mathrm{T}}\\ \widehat{\boldsymbol{K}}\end{bmatrix}\begin{bmatrix}\mathbf{1} & \widehat{\boldsymbol{K}}^{\mathrm{T}}\end{bmatrix}\right)\begin{bmatrix}b_m\\ \boldsymbol{\alpha}_m\end{bmatrix} - 2\left(\begin{bmatrix}\mathbf{1}^{\mathrm{T}}\\ \widehat{\boldsymbol{K}}\end{bmatrix}\boldsymbol{Y}_m\right)^{\mathrm{T}}\begin{bmatrix}b_m\\ \boldsymbol{\alpha}_m\end{bmatrix}\right]\right\} \tag{14}$$

And Eq. (12) can be unfolded as follows:

$$\left(\begin{bmatrix}0 & \mathbf{0}^{\mathrm{T}}\\ \mathbf{0} & \boldsymbol{K}/\gamma\end{bmatrix}+\begin{bmatrix}N & \mathbf{1}^{\mathrm{T}}\widehat{\boldsymbol{K}}^{\mathrm{T}}\\ \widehat{\boldsymbol{K}}\mathbf{1} & \widehat{\boldsymbol{K}}\widehat{\boldsymbol{K}}^{\mathrm{T}}\end{bmatrix}\right)\begin{bmatrix}\boldsymbol{b}\\ \boldsymbol{\alpha}\end{bmatrix}=\begin{bmatrix}\mathbf{1}^{\mathrm{T}}\\ \widehat{\boldsymbol{K}}\end{bmatrix}\boldsymbol{Y} \tag{15}$$

Let $\boldsymbol{U}^n=\left(\begin{bmatrix}0 & \mathbf{0}^{\mathrm{T}}\\ 0 & \boldsymbol{K}/\gamma\end{bmatrix}+\begin{bmatrix}N & \mathbf{1}^{\mathrm{T}}\widehat{\boldsymbol{K}}^{\mathrm{T}}\\ \widehat{\boldsymbol{K}}\mathbf{1} & \widehat{\boldsymbol{K}}\widehat{\boldsymbol{K}}^{\mathrm{T}}\end{bmatrix}\right)^{-1}$, where n stands for the nth iteration. If the qth training pattern $\boldsymbol{x}_q$ is chosen at the $(n+1)$th iteration, we can get $\boldsymbol{U}^{n+1}$ using the Sherman–Morrison formula:

$$\boldsymbol{U}^{n+1}=\begin{bmatrix}\boldsymbol{U}^n & \mathbf{0}\\ \mathbf{0}^{\mathrm{T}} & 0\end{bmatrix}+\zeta\begin{bmatrix}\boldsymbol{\beta}\\ -1\end{bmatrix}\begin{bmatrix}\boldsymbol{\beta}^{\mathrm{T}} & -1\end{bmatrix} \tag{16}$$

where $\zeta=\left(k_{qq}+\hat{\boldsymbol{k}}_q^{\mathrm{T}}\hat{\boldsymbol{k}}_q-\left[\hat{\boldsymbol{k}}_q^{\mathrm{T}}\mathbf{1}\quad\hat{\boldsymbol{k}}_q^{\mathrm{T}}\widehat{\boldsymbol{K}}^{\mathrm{T}}+\boldsymbol{k}_q^{\mathrm{T}}\right]\boldsymbol{\beta}\right)^{-1}$,

$\boldsymbol{\beta}=\boldsymbol{U}^n\begin{bmatrix}\mathbf{1}^{\mathrm{T}}\hat{\boldsymbol{k}}_q\\ \boldsymbol{k}_q+\widehat{\boldsymbol{K}}\hat{\boldsymbol{k}}_q\end{bmatrix}$, $\boldsymbol{k}_q=\begin{bmatrix}k(\boldsymbol{x}_q,\boldsymbol{x}_i)\\ k(\boldsymbol{x}_q,\boldsymbol{x}_{i+1})\\ \vdots\\ k(\boldsymbol{x}_q,\boldsymbol{x}_j)\end{bmatrix}$, $\hat{\boldsymbol{k}}_q=\begin{bmatrix}k(\boldsymbol{x}_q,\boldsymbol{x}_1)\\ k(\boldsymbol{x}_q,\boldsymbol{x}_2)\\ \vdots\\ k(\boldsymbol{x}_q,\boldsymbol{x}_i)\end{bmatrix}$

with $i,i+1,\ldots,j\in S$.

And then, with the equation $\begin{bmatrix}b_m^n\\ \boldsymbol{\alpha}_{S,m}^n\end{bmatrix}=\boldsymbol{U}^n\begin{bmatrix}\hat{0}\\ \widehat{\boldsymbol{Y}}_{S,m}\end{bmatrix}$, Eq. (17) will be obtained:

$$\begin{aligned}\begin{bmatrix}b_m^{n+1}\\ \boldsymbol{\alpha}_{S,m}^{n+1}\\ \alpha_{q,m}^{n+1}\end{bmatrix}&=\boldsymbol{U}^{n+1}\begin{bmatrix}\hat{0}\\ \widehat{\boldsymbol{Y}}_{S,m}\\ \hat{y}_{q,m}\end{bmatrix}=\begin{bmatrix}\boldsymbol{U}\begin{bmatrix}\hat{0}\\ \widehat{\boldsymbol{Y}}_{S,m}\end{bmatrix}\\ 0\end{bmatrix}\\&\quad+\zeta\left(\boldsymbol{\beta}^{\mathrm{T}}\begin{bmatrix}\hat{0}\\ \widehat{\boldsymbol{Y}}_{S,m}\end{bmatrix}-\hat{y}_{q,m}\right)\begin{bmatrix}\boldsymbol{\beta}\\ -1\end{bmatrix}\\&=\begin{bmatrix}b_m^n\\ \boldsymbol{\alpha}_{S,m}^n\\ 0\end{bmatrix}+\gamma\left(\boldsymbol{\beta}^{\mathrm{T}}\begin{bmatrix}\hat{0}\\ \widehat{\boldsymbol{Y}}_{S,m}\end{bmatrix}-\hat{y}_{q,m}\right)\begin{bmatrix}\boldsymbol{\beta}\\ -1\end{bmatrix}\end{aligned} \tag{17}$$

where $\hat{0}=\sum_{i=1}^{N}y_{i,m}$, $\widehat{\boldsymbol{Y}}_m=\widehat{\boldsymbol{K}}\boldsymbol{Y}_m, m=1,2,\ldots,M$.

In such way, $\boldsymbol{U}$, $\boldsymbol{\alpha}_m$, and b_m can be updated efficiently, and the target function Eq. (14) becomes:

$$L^n=\sum_{m=1}^{M}\left[\begin{bmatrix}b_m^n & \left(\boldsymbol{\alpha}_m^n\right)^{\mathrm{T}}\end{bmatrix}\left(\begin{bmatrix}0 & \mathbf{0}^{\mathrm{T}}\\ \mathbf{0} & \boldsymbol{K}_{SS}/\gamma\end{bmatrix}+\begin{bmatrix}\mathbf{1}^{\mathrm{T}}\\ \widehat{\boldsymbol{K}}_S\end{bmatrix}\begin{bmatrix}\mathbf{1} & \widehat{\boldsymbol{K}}_S^{\mathrm{T}}\end{bmatrix}\right)\begin{bmatrix}b_m^n\\ \boldsymbol{\alpha}_m^n\end{bmatrix}-2\left(\begin{bmatrix}1\\ \widehat{\boldsymbol{K}}_S\end{bmatrix}\boldsymbol{Y}_m\right)^{\mathrm{T}}\begin{bmatrix}b_m^n\\ \boldsymbol{\alpha}_m^n\end{bmatrix}\right]$$

If $\boldsymbol{\alpha}_m^n$ and b_m^n are fixed, and $i\in P=\{1,2,\ldots,N\}-S$, we can get:

$$L^{n+1}=\min_{\alpha_{i,m}}\left\{\sum_{m=1}^{M}\left[\frac{1}{2}\left(k_{ii}/\gamma+\hat{\boldsymbol{k}}_i^{\mathrm{T}}\hat{\boldsymbol{k}}_i\right)\alpha_{i,m}^2+r_{i,m}^n\alpha_{i,m}\right]\right\} \tag{18}$$

where $r_{i,m}^n$ is used to denote the approximate error about $\hat{y}_{i,m}$ with $\boldsymbol{\alpha}_m^n$ and b_m^n, which is expressed in the following:[10]

$$r_{i,m}^n=\begin{cases}-\hat{y}_{i,m}, & n=0\\ \left[\hat{\boldsymbol{k}}_i^{\mathrm{T}}\mathbf{1}\quad\hat{\boldsymbol{k}}_i^{\mathrm{T}}\widehat{\boldsymbol{K}}^{\mathrm{T}}+\boldsymbol{k}_{S,i}^{\mathrm{T}}/\gamma\right]\begin{bmatrix}b_m^n\\ \boldsymbol{\alpha}_m^n\end{bmatrix}-\hat{y}_{i,m}, & n>0\end{cases} \tag{19}$$

So the optimal value of Eq. (18) is :

$$L^{n+1}=-\sum_{m=1}^{M}\frac{\left(r_{i,m}^n\right)^2}{2\left(k_{ii}/\gamma+\hat{\boldsymbol{k}}_i^{\mathrm{T}}\hat{\boldsymbol{k}}_i\right)} \tag{20}$$

Finally, if training sample $\boldsymbol{x}_i$ in P subset meets the demand of $\min_{i\in P}L^{n+1}$,$\boldsymbol{x}_i$ will be selected as support vector. At last, terminate selecting patterns from P subset while $\max\left(\sum_{m=1}^{M}\left|r_{P,m}^n\right|\right)<\varepsilon$ (ε is a small positive constant) or until reaching the predefined size of support vectors (define as Q).

In the above way, for a multi-input multi-output system, after combining the iterative computation of the kernel matrix inversion with the strategy of selecting a reduced subset, the MRR-LSSVR algorithm is able to be realized, which takes the multi-output comprehensive influences on selecting support vectors into consideration. Additionally, compared with RR-LSSVR at a cost of $O(2uMN)$, a multi-input multi-output problem can be solved with a better real-time ability using MRR-LSSVR at a cost of $O(2uN)$, where $O(\cdot)$ is the computation complexity, $M>1$, and u is the number of input variables.

For the MRR-LSSVR estimation module in this paper, output variables are $y=[y_1\ y_2\ y_3]=[D_{\mathrm{com}}\ \ D_{\mathrm{gas}}\ \ D_{\mathrm{pow}}]$, and eight input variables $\boldsymbol{x}=[x_1\ x_2\ \ldots\ x_8]=[H\ \ V_x\ \ N_{\mathrm{g}}\ \ N_{\mathrm{p}}\ \ p_3\ \ p_{44}\ \ T_{44}\ \ T_{45}]$ are determined by the PS algorithm. The training dataset is picked based on the turbo-shaft engine component-level model, where the sample size $N=2160$. Additionally, it is needed to predefine the size of final support vectors or threshold value ε. At last, via the simulation debugging method, the training parameters of MRR-LSSVR with Gaussian kernel function are fixed as follows:

The size of final support vectors $Q=300$;
The threshold value $\varepsilon=0$;
The regulator $\gamma=2^{18}$;
The kernel parameter $\upsilon=1.2$.

In order to prove the availability of the parameter estimation module, the relative errors of a test subset are dis-played in Fig. 4. From Fig. 4, it can be seen that the relative errors of

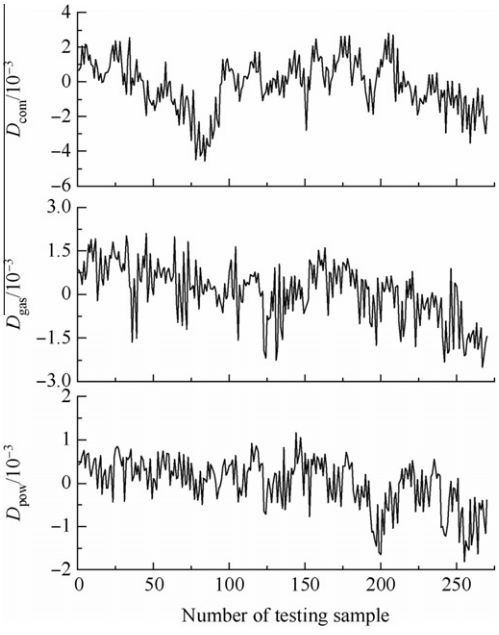

Fig. 4 Relative errors of a test subset.

D_{com}, D_{gas}, and D_{pow}, i.e., error-D_{com} are all lower than 5‰, which achieve a satisfying accuracy level.

3.4. Adaptive engine module

In this paper, the MRR-LSSVR estimation module is designed in a flight envelope of an altitude $H \leqslant 1000$ m and a forward speed $V_x \leqslant 50$ m/s, and meanwhile the flight envelope is divided with $\Delta H = 100$ m and $\Delta V_x = 5$ m/s for training network. In every engine steady-state operation point, performance degenerations of the engine are set as follows:

$$D_{com} = 0, -0.01, -0.02, -0.03$$
$$D_{gas} = 0, -0.01, -0.02, -0.03$$
$$D_{pow} = 0, 0.01, 0.02, 0.03$$

Here, one thing should be pointed out: for an actual adaptive engine model, it is necessary to take six condition parameters (flow and efficiency deterioration) into consideration, and in this paper only three condition parameters (D_{com}, D_{pow}, and D_{gas}) are considered as an example. These three condition parameters represent the deteriorations of three main engine components (compressor, gas turbine, and power turbine). With the groups of single or multiple deteriorations, the dynamic training subset can be gained based on the integrated helicopter/engine system. And then, through normalizing the input $\boldsymbol{x}$ of a training sample which is added with the white noise, the MRR-LSSVR mapping module, i.e., the nonlinear mapping relationship between the input $\boldsymbol{x}$ and the output $\boldsymbol{y}$, will be constructed based on MRR-LSSVR. According to the prominent memory and generalization ability of MRR-LSSVR, the model of engine parameter estimation can be built in every steady-state operation point.

During the realization of the MRR-LSSVR predictor, the module is turned on to track the actual engine state at first. After normalizing the above eight engine measurable parameters and putting them as the inputs of the MRR-LSSVR estimation module, the outputs of the predictor, i.e., D_{com}, D_{gas}, and D_{pow}, are worked out. With this MRR-LSSVR estimator (see Fig. 5), the onboard engine model has a good adaptive capability in this certain flight envelope.

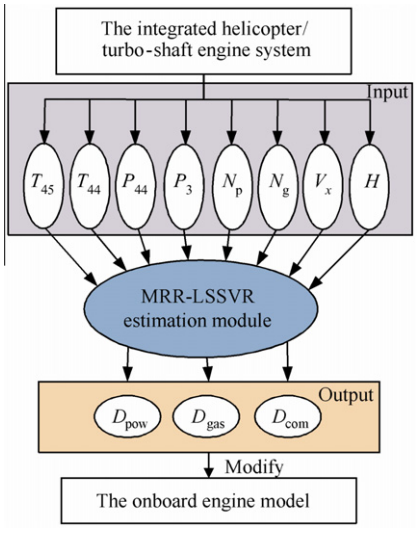

Fig. 5 Realization of the MRR-LSSVR estimator.

4. Case studies

For the proposed adaptive engine model, the accuracy and generalization ability are the main problems that need careful consideration. In this paper, based on the integrated helicopter/turbo-shaft engine simulation system, lots of simulation experiments are carried out to verify the adaptive ability on a computer with an Intel i5 M460 (2.53 + 2.53 GHz) processor and 2.0 GB memory, and the simulation step is 50 ms. In the simulation process, the engine performance deteriorations are imitated by changing the flow or efficiency of different engine components together. The simulation results under different operating conditions are listed as follows, where the parameters with subscript "r" represent the outputs of the actual engine model and the ones with no subscript represent the outputs of the adaptive engine model.

4.1. Simulations with single engine performance deterioration

In this section, on the ground, the simulation tests with single engine component degenerate are implemented. Due to the limited space, we only introduce the simulation of compressor flow deterioration, whose results are similar to those of gas turbine efficiency deterioration or power turbine flow deterioration.

Fig. 6 is the response of compressor flow deterioration. Firstly, at the moment of $t = 5$ s, engine component degenerations are set as follows: $D_{com,r} = -0.03$, $D_{gas,r} = 0$, and $D_{pow,r} = 0$. From Fig. 6(a), it can be seen that the estimated error of D_{com} is only 1.2%, and D_{gas} and D_{pow} are close to zero. Subsequently, after correcting the onboard engine model with the estimated error of engine health parameters, the state variables in Fig. 6(b) can be obtained, where N_g and N_p are gas turbine and power turbine rotor relative speeds of the adaptive engine model. From Fig. 6(b), we can see that the state of the adaptive engine model can nearly keep in step with the actual engine state, and the dynamic response time of system is about 9 s. Finally, engine performance parameters SMC and HPP can be calculated by the nonlinear calculation module of the adaptive engine model. From Fig. 6(c), it can be seen that the relative steady-state errors of SMC and HPP are 0.1% and 0.02%, respectively, which achieve a good accuracy level. Additionally, computing time of the MRR-LSSVR module is short, below 1 ms in a simulation step, which is enough to meet the requirement of real-time ability.

4.2. Simulations with multiple engine performance deteriorations

Through above single deterioration simulations, it is obvious that the adaptive engine model has a fine adaptive ability. For multiple engine components being degraded together, some digital simulations are performed in this section.

Firstly, at the operation point of $H = 0$ m and $V_x = 0$ m/s, $D_{com,r} = -0.01$, $D_{gas,r} = -0.02$, $D_{pow,r} = 0.03$ are set at the moment of $t = 5$ s. The simulation results are shown in Fig. 7. The dynamic response time of system is about 8 s. Compared with the preinstall deteriorations, the estimated values of D_{com}, D_{gas}, and D_{pow} all attain a high accuracy level, whose maximum relative steady-state error is 1.87%. This precision achieves the simulation level in Ref. [16]. Additionally, like the

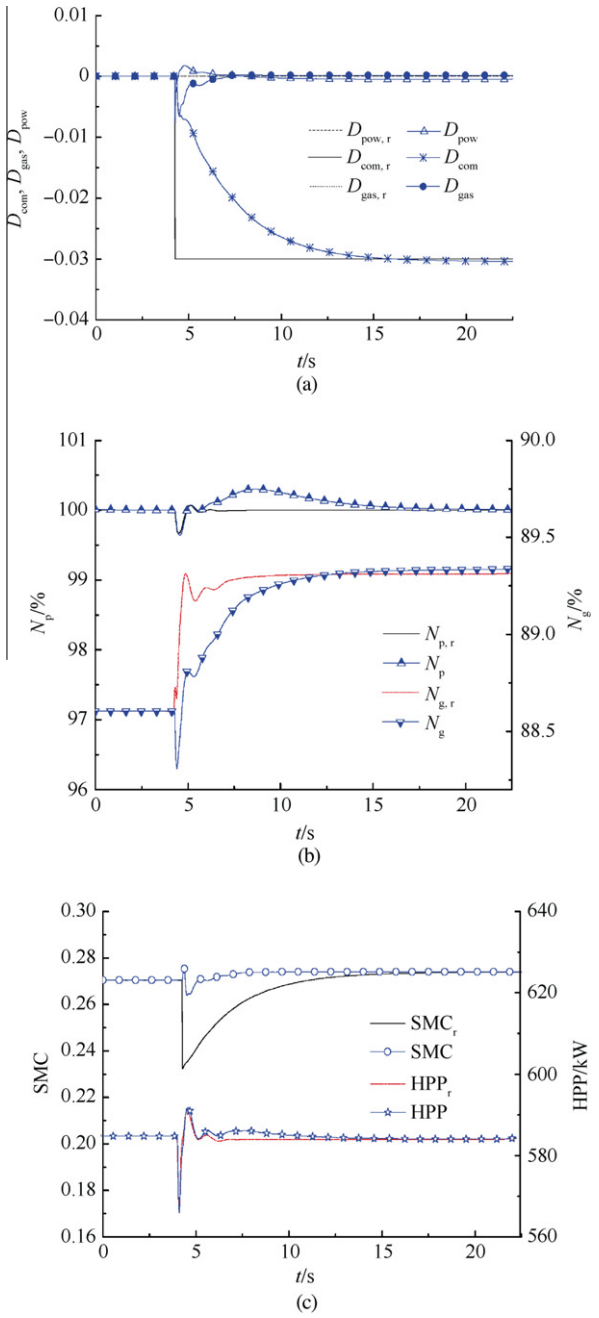

Fig. 6 Simulations at $H = 0$ m, $V_x = 0$ m/s with $D_{com,r} = -0.03$, $D_{gas,r} = 0$, $D_{pow,r} = 0$.

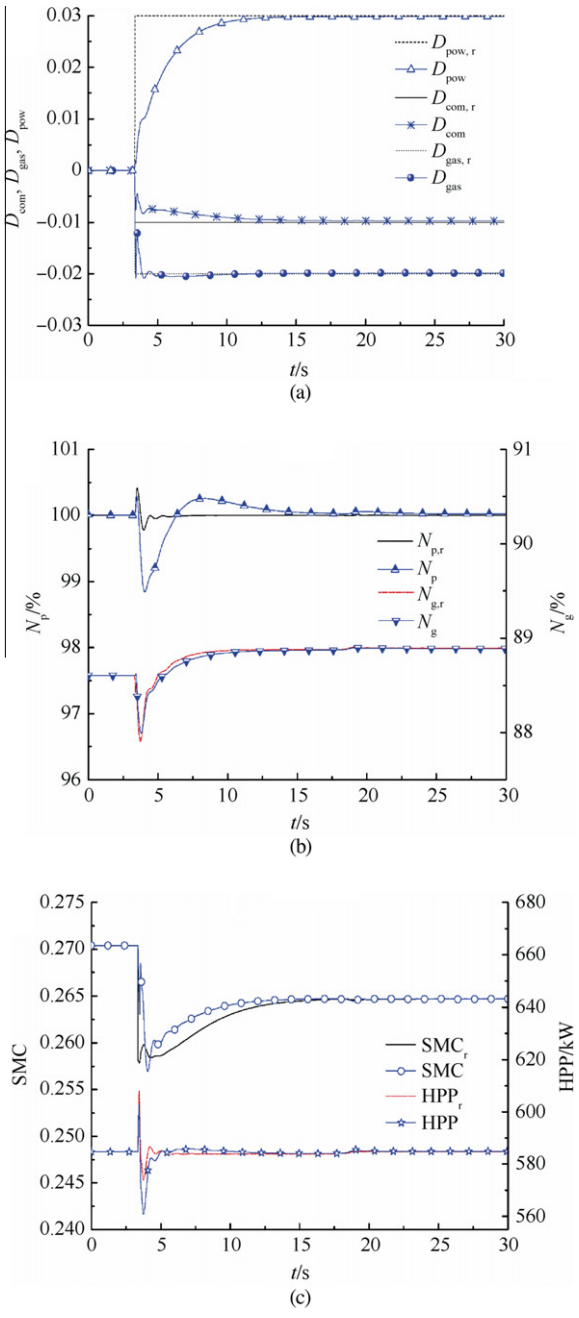

Fig. 7 Simulations at $H = 0$ m, $V_x = 0$ m/s with $D_{com,r} = -0.01$, $D_{gas,r} = -0.02$, $D_{pow,r} = 0.03$.

case of single deterioration, the computing time of the MRR-LSSVR module is also about 1 ms. Ultimately, the state variables N_g, N_p and the performance parameters SMC, HPP of the adaptive engine model can all be in step with the state of the actual engine.

Secondly, at another operating point of $H = 800$ m, $V_x = 30$ m/s with $D_{com,r} = -0.02$, $D_{gas,r} = -0.03$, $D_{pow,r} = 0.01$, the simulation test was performed (see Fig. 8). From Fig. 8, it can be seen that the adaptive model is connected at the 5th second. The relative errors of estimated deteriorations are all close to zero, where the maximum error is 1.35%. In comparison with the steady-state of the actual engine, the precisions of N_g,N_p, SMC, and HPP of the adaptive engine model are satisfying.

Putting the facts of Figs. 7 and 8 together, we can find that the adaptive model has a good self-correct ability both on the ground and in the high altitude.

Furthermore, in order to check the generalization ability of the proposed adaptive model in a certain flight envelope, the simulation tests at different operation points are implemented and the test results are listed in Table 2, where the operating points at $H = 100$ m, $V_x = 12$ m/s and $H = 700$ m, $V_x = 44$ m/s are out of training samples. In Table 2, $E_{1,max}$ presents the maximum relative steady-state error of estimated

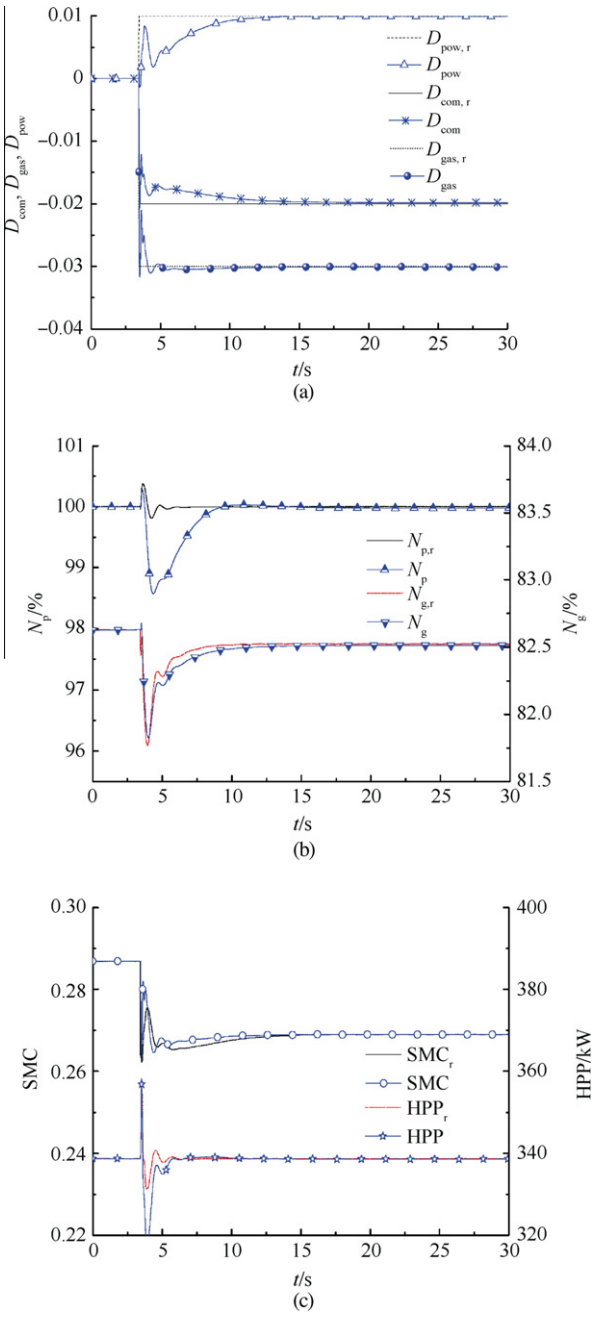

Fig. 8 Simulations at $H = 800$ m, $V_x = 30$ m/s with $D_{com,r} = -0.02$, $D_{gas,r} = -0.03$, $D_{pow,r} = 0.01$.

engine degenerations values, and $E_{2,max}$ stands for the maximum relative steady-state error of engine performance parameters. From simulation results of the operating points at $H = 300$ m, $V_x = 25$ m/s and $H = 1000$ m, $V_x = 50$ m/s, the relative errors of estimated engine degenerations values are about 1%, and those of engine performance parameters are almost zero. For the operating points out of training samples, i.e., $H = 100$ m, $V_x = 12$ m/s and $H = 700$ m, $V_x = 44$ m/s, $E_{1,max}$ is also below 2%, and $E_{2,max}$ is close to zero. These prove that the adaptive model has a good robustness and fault-tolerant capability, which can estimate the engine deteriorations successfully in the certain flight envelope of

Table 2 Simulation results of the adaptive engine model.

Parameter	$H = 100$ m, $V_x = 12$ m/s	$H = 300$ m, $V_x = 25$ m/s	$H = 700$ m, $V_x = 44$ m/s	$H = 1000$ m, $V_x = 50$ m/s
$D_{com,r}$	−0.01	−0.02	−0.03	−0.03
$D_{gas,r}$	−0.03	−0.01	−0.02	−0.03
$D_{pow,r}$	0.02	0.03	0.01	0.03
D_{com}	−0.01013	−0.01996	−0.02997	−0.03004
D_{gas}	−0.03014	−0.00994	−0.02034	−0.03023
D_{pow}	0.02025	0.02981	0.00987	0.02984
$E_{1,max}$ (%)	1.25	0.63	1.70	0.77
SMC_r	0.25987	0.29188	0.28130	0.27076
HPP_r (kW)	498.34822	356.27485	334.47487	357.22745
SMC	0.25985	0.29186	0.28106	0.27070
HPP (kW)	498.22486	356.32690	334.25888	357.10163
$E_{2,max}$ (‰)	0.3	0.2	0.9	0.4
Computing time	About 1 ms	About 1 ms	About 1 ms	About 1 ms

Table 3 Simulation comparisons between BP-NN and MRR-LSSVR.

Parameter	$H = 0$ m, $V_x = 0$ m/s		$H = 550$ m, $V_x = 33$ m/s	
	BP-NN	MRR-LSSVR	BP-NN	MRR-LSSVR
$D_{com,r}$	−0.03	−0.03	−0.03	−0.03
$D_{gas,r}$	−0.03	−0.03	−0.03	−0.03
$D_{pow,r}$	0.03	0.03	0.03	0.03
D_{com}	−0.02986	−0.02966	−0.02734	−0.02986
D_{gas}	−0.03046	−0.02997	−0.02996	−0.03007
D_{pow}	0.03053	0.03041	0.03078	0.02979
$E_{1,max}$ (%)	1.7	1.4	8.9	0.7
SMC_r	0.26028	0.26028	0.28091	0.28091
HPP_r (kW)	584.77101	584.77101	328.55633	328.55633
SMC	0.26007	0.26040	0.28064	0.28104
HPP (kW)	584.63942	584.73087	328.67836	328.47652
$E_{2,max}$ (‰)	0.8	0.5	1.0	0.5
Computing time	Below 1 ms	About 1 ms	Below 1 ms	About 1 ms

$H \leqslant 1000$ m and $V_x \leqslant 50$ m/s. And moreover, the computing time of the MRR-LSSVR estimation module is short, only about 1 ms. It is obvious that the adaptive model based on MRR-LSSVR can meet the requirement of real-time capability in the certain flight envelope.

Finally, as is well known, conventional Kalman filter is the estimated method depending on a state variable model, which needs much time to finish parameter estimations because of multi-step iterative calculations. Compared with Kalman filter, back propagation neural network (BP-NN) and MRR-LSSVR, as the learning method based on data, can both improve the computing time of engine parameter estimation dramatically. In Table 3, simulation comparisons between BP-NN and MRR-LSSVR are listed, where the working point of $H = 0$ m, $V_x = 0$ m/s is the design operation point, and $H = 550$ m, $V_x = 33$ m/s is out of training samples. At the design point, both BP-NN and MRR-LSSVR give high estimating precisions, $E_{1,max}$ about 1.5% and $E_{2,max}$ about zero. However, at the working point of $H = 550$ m, $V_x = 33$ m/s, in comparison with BP-NN, MRR-LSSVR can achieve a higher precision with similar computing time due to its good generalization ability. As a result, $E_{1,max}$ using MRR-LSSVR is

only 0.7%, but 8.9% using BP-NN, and therefore the adaptive engine model based on MRR-LSSVR has a stronger robustness than the one based on BP-NN.

5. Conclusions

(1) In order to perform parameter estimation considering interactions between engine performance deteriorations while multiple engine components being degraded together, and meantime improve real-time capability of the adaptive module, the MRR-LSSVR algorithm is realized depending on the RR-LSSVR algorithm.
(2) Based on multi-input multi-output LSSVR, the PS algorithm with the wrapper criterion is designed to choose reasonable inputs of the MRR-LSSVR estimation module.
(3) Combining MRR-LSSVR and PS, an adaptive modeling method of turbo-shaft engines is proposed based on the simulation platform of an integrated helicopter/turbo-shaft engine system.
(4) Through plenty of simulation experiments, it is proved that the proposed adaptive engine model can track the actual engine state rapidly and exactly, where the maximum relative errors of estimated engine degenerations are below 2% and the deviations of engine performance parameters are almost close to zero. In comparison with BP-NN, the proposed method can achieve a higher estimating precision with similar computing time about 1 ms.

Acknowledgments

This study was co-supported by Aeronautical Science Foundation of China (No. 2010ZB52011) and Funding of Jiangsu Innovation Program for Graduate Education (No. CXLX11_0213).

References

1. Smith BJ, Zagranski RD. Closed loop bench testing of the next generation control system for helicopter engines. In: *AHS International 58th Annual Forum*, Montreal, Canada; 2002.
2. Desai MC, Crainic C. Adaptive thermodynamic engine model for the next generation control system for helicopter engines. In: *AHS International 58th Annual Forum*, Montreal, Canada; 2002.
3. Luppold RH, Roman JR, Gallops GW, Kerr LJ. *Estimating in-flight engine performance variations using Kalman Filter concepts*; 1989. Report No.: AIAA-89-2584.
4. Zou XQ, Sun JG, Zhang HB. Adaptive model of rotor/turbo-shaft engine. *J Beijing Univ Aeronaut Astronaut* 2009;**35**(10):1192–6 [Chinese].
5. Kobayashi T, Simon DL, Litt JS. Application of a constant gain extended Kalman Filter for in-flight estimation of aircraft engine performance parameters; 2005. Report No.: NASA/TM 2005-213865.
6. Chen T, Sun JG. Aeroengine gas path fault diagnosis using rough sets and neural networks. *J Aerospace Power* 2006;**21**(1):207–12 [Chinese].
7. Zhang ZS, Li LJ, He ZJ. Based on support vector machine fault diagnosis. *J Xian Jiaotong Univ* 2002;**36**(3):618–20.
8. Suykens JAK, Vandewalle J. Least squares support vector machine classifiers. *Neural Process Lett* 1999;**9**(3):293–300.
9. Lin KM, Lin CJ. A study on reduced support vector machines. *IEEE Trans Neural Netw* 2003;**14**(6):1449–59.
10. Zhao YP, Sun JG. Recursive reduced least squares support vector regression. *Pattern Recognit* 2009;**42**(5):837–42.
11. Jiao L, Bo L, Wang L. Fast sparse approximation for least squares support vector machine. *IEEE Trans Neural Netw* 2007;**18**(3):685–97.
12. Zhao YP, Sun JG. Thrust estimator design based on least squares support vector regresson machine. *J Harbin Inst Technol New Ser* 2010;**17**(4):578–83.
13. Zhang HB, Yao WR, Chen GQ. Design of a numeric simulation platform for integrated turbo-shaft engine/helicopter control system. *J Propul Technol* 2011;**32**(3):383–90 [Chinese].
14. Lu F, Huang JQ, Qiu XJ. Application of multi-outputs LSSVR by PSO to the aero-engine model. *J Syst Eng Electron* 2009;**20**(5):1153–8.
15. Li LJ. The study of modeling algorithm based on LS-SVM and predictive control algorithm [dissertation]. Hangzhou: Zhejiang University; 2008 [Chinese].
16. Espana MD, Gilyard GB. On the estimation algorithm used in adaptive performance optimization of turbofan engines; 1993. Report No.: NASA-TM-4551.

Wang Jiankang received his M.S. degree in the aerospace propulsion theory and engineering field from Nanjing University of Aeronautics and Astronautics in 2010. Since then, he has been working toward a Ph.D. degree there. His research interests include modeling, control, and fault diagnosis of aero-engines.

Zhang Haibo is an associate professor in the College of Energy and Power Engineering at Nanjing University of Aeronautics and Astronautics. He received his Ph.D. degree from the same university in 2005. His main research interests are aero-engine modeling and control.