

Model parameters estimation of aero-engine based on hybrid optimization algorithm

LI Qiu-hong, LI Ye-bo, JIANG Dian-wen

(College of Energy and Power Engineering

Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: A hybrid optimization algorithm for the time-domain identification of multivariable, state space model for aero-engine was presented in this paper. The optimization procedure runs particle swarm optimization (PSO) and least squares optimization (LSO) “in series”. PSO starts from an initial population and searches for the optimum solution by updating generations. However, it can sometimes run into a suboptimal solution. Then LSO can start from the suboptimal solution of PSO, and get an optimum solution by conjugate gradient algorithm. The algorithm is suitable for the high-order multivariable system which has many parameters to be estimated in wide ranges. Hybrid optimization algorithm is applied to estimate the parameters of a 4-input 4-output state variable model (SVM) for aero-engine. The simulation results demonstrate the effectiveness of the proposed algorithm.

Key words: aero-engine; state variable model (SVM); particle swarm optimization (PSO); least squares optimization (LSO); hybrid optimization algorithm

CLC number: V233.7

Document code: A

Introduction

Today evolutionary algorithms are widely researched, such as genetic algorithm (GA)^[1], ant colony algorithm^[2], artificial fish swarm algorithm^[3], frog-leaping algorithm^[4], bee swarm algorithm^[5], and particle swarm algorithm^[6]. They search for the solution that maximizes or minimizes a given function in a prescribed search space. Compared with GA which is widely used, particle swarm optimization (PSO) is quite easy to implement and there are few parameters that need to be adjusted. PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas in which GA can be applied.

PSO is a population-based stochastic optimization technique, inspired by the social behavior

of bird flocking or fish schooling. PSO is initialized with a population of random solutions and searches for the optimum by updating generations. The solutions are called particles, and fly through the search space by following the current optimum particles. Recently, different hybrid PSOs have been proposed to overcome the drawback of possible trapping in local optima^[7]. Three optimizers based on GA, differential evolution and PSO are employed “in parallel” to the design of interplanetary trajectories, and the hybrid approach exhibits good performance for most of complex problems with a large number of variables^[8]. In this paper, a hybrid optimization algorithm based on PSO and least squares optimization (LSO) is proposed and applied to estimate the parameters of a 4-input 4-output state variable model (SVM) for aero-engine. The two optimizers run “in series”. We

will use this method to obtain numerical solution of SVM from the simulation data of a component level model (CLM) for aero-engine.

PSO starts from a random initial population and LSO starts from an initial value. If PSO runs into a suboptimal solution, LSO can start from its solution, and reach an optimum.

1 Particle swarm algorithm

PSO learns from the social behavior of swarm animals and uses it to solve optimization problems. In PSO, each single solution is a “bird” and called a “particle”. Particle moves according to its instantaneous velocity which is updated every iteration to change its position and improve the solution. Suppose in a m -dimensional target search space, there are n particles that form a colony. The position of the i th particle ($i = 1, 2, \dots, n$) in the search space can be represented by $\mathbf{X}_i$, then $\mathbf{X}_i$ is a m -dimensional vector, $\mathbf{X}_i = [x_{i1}, x_{i2}, \dots, x_{im}]$. Its velocity can also be represented by another m -dimensional vector, $\mathbf{V}_i = [v_{i1}, v_{i2}, \dots, v_{im}]$. Velocity is changed to move the particle toward the optimum according to that: (1) the best solution $\mathbf{X}_{ibest}^t$ that the particle itself has reached in previous iterations; (2) the best solution $\mathbf{X}_{gbest}^t \in \mathbf{X}_{ibest}^t$ that any particle in the population has reached. Then the particle updates its velocity:

$$\mathbf{v}_i^{t+1} = \omega \mathbf{v}_i^t + (\mathbf{C}_1) \cdot \times (\mathbf{X}_{ibest}^t - \mathbf{X}_i^t) + (\mathbf{C}_2) \cdot \times (\mathbf{X}_{gbest}^t - \mathbf{X}_i^t) \quad (1)$$

where ω is the weighting coefficient, $\mathbf{C}_1$ and $\mathbf{C}_2$ are positive cognitive learning factor and social learning factor. The “ $\times$ ” is the multiplication of corresponding parameters of vectors. $\omega \mathbf{v}_i^t$ is the effect of particle's current velocity. $(\mathbf{C}_1) \cdot \times (\mathbf{X}_{ibest}^t - \mathbf{X}_i^t)$ reflects the cognitive pattern and the memory of the particle itself. $(\mathbf{C}_2) \cdot \times (\mathbf{X}_{gbest}^t - \mathbf{X}_i^t)$ reflects the social pattern and the effect of the swarm.

At time $t+1$, the position of i th particle will be

$$\mathbf{X}_i^{t+1} = \mathbf{X}_i^t + \mathbf{v}_i^{t+1} \quad (2)$$

The particle keeps adjusting its position in order to find the best solution of the problem according to its own experience and the swarm information.

2 PSO modeling

SVM is very important in linear system theory. Many control and diagnosis methods for aero-engine need SVM. So it is valuable to re-search the estimation of SVM parameters from CLM for aero-engine. There are many modeling methods of SVM, such as partial derivative algorithm^[9], least squares curve fitting algorithm^[10-11], improved least squares algorithm^[12], genetic algorithm^[13], etc. Most of these methods are based on optimization. SVM built here will provide plant for the design of multivariable robust control system of aero-engine. As SVM to be built has more parameters and can not get satisfied solution by the approaches we usually use in Refs. [10-13], which are applied in the parameters estimation of 2-input SVM. Hence we propose a hybrid optimization algorithm to run PSO and LSO “in series” and estimate the parameters from the step responses data of the component level aero-engine model.

SVM is:

$$\begin{aligned} \Delta \dot{\mathbf{x}}_m &= \mathbf{A} \cdot \Delta \mathbf{x}_m + \mathbf{B} \cdot \Delta \mathbf{u} \\ \Delta \mathbf{y} &= \mathbf{C} \cdot \Delta \mathbf{x}_m + \mathbf{D} \cdot \Delta \mathbf{u} \end{aligned} \quad (3)$$

where, Δ is increment symbol. $\mathbf{x}_m = [N_1, N_2]^T$, N_1 and N_2 are fan and compressor rotor speed. $\mathbf{u} = [W_f, A_8, G_{vf}, G_{vc}]^T$, W_f is main burner fuel flow, A_8 is nozzle area, G_{vf} is fan inlet guide vane angle, and G_{vc} is compressor stator vane angle. $\mathbf{y} = [N_1, N_2, T_{46}, P_{it}]^T$, T_{46} is the low pressure turbine exit total temperature, P_{it} is the turbine pressure ratio. Then

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \end{bmatrix}, \\ \mathbf{C} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ c_{31} & c_{32} \\ c_{41} & c_{42} \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ d_{31} & d_{32} & d_{33} & d_{34} \\ d_{41} & d_{42} & d_{43} & d_{44} \end{bmatrix}. \end{aligned}$$

To improve the precision, **A** and **B** are optimized first, and then the matrices **C** and **D**. The particles formed by parameters of **A** and **B** are vectors with 12-dimension. The swarm contains 80 particles. The i th particle is

$$\mathbf{X}_i = [a_{11i}, a_{12i}, a_{21i}, a_{22i}, b_{11i}, b_{12i}, b_{13i}, b_{14i}, b_{21i}, b_{22i}, b_{23i}, b_{24i}].$$

Then from each particle we can get matrices **A** and **B**. When estimating **A** and **B**, let **C** be an identity matrix and **D** be a zero matrix. Through simulation, we can get step responses data of CLM. The steps of the 4-input are done respectively. The step magnitudes of inputs W_f and A_8 are 0.5% of their nominal value. G_{vf} and G_{vc} are zeros at design point, and reach their maximum value, so the steps of G_{vf} and G_{vc} are -0.5° . The output signals are normalized by their design point data and saved as $\tilde{N}_1, \tilde{N}_2, \tilde{T}_{46}, \tilde{P}_{it}$. Apply the step signals with the same magnitude to each SVM formed by particles, and the output data are saved as $\bar{N}_{1p}, \bar{N}_{2p}, \bar{T}_{46p}, \bar{P}_{itp}$. The responses of SVM are linear responses. If the linear responses of SVM formed by the best particle can follow the responses of CLM, SVM is a linearization of aero-engine.

When calculating **A** and **B**, we use only the responses of N_1 and N_2 . Then model parameters of **A** and **B** can be found by minimizing the 2-norm of the responses errors between CLM and SVM. The cost function is,

$$J = \min_{\mathbf{X}_i} \sum_{p=1}^4 \sum_{k=1}^L \left\{ \frac{(\tilde{N}_{1p}[k] - \bar{N}_{1p}[k])^2}{g_{1p}^2} + \frac{(\tilde{N}_{2p}[k] - \bar{N}_{2p}[k])^2}{g_{2p}^2} \right\} \quad (4)$$

where the subscript $p = 1, 2, 3, 4$ is corresponding to step responses of W_f, A_8, G_{vf}, G_{vc} respectively. L is an integer value related to the time. $g_{1p} = \tilde{N}_{1p}[L]$ and $g_{2p} = \tilde{N}_{2p}[L]$ are scaling factors, i. e. the static value of responses.

If the cost function is minimized, then the responses errors between CLM and SVM are minimized in 2-norm. Let the weighting coefficient ω in Eq. (1) vary with time,

$$\omega(t) = \omega_{\max} - t \times (\omega_{\max} - \omega_{\min}) / t_{\text{all}} \quad (5)$$

where $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum weighting coefficients. t_{all} is the evolution time. Then at the start of the evolution, the weighting coefficient is relatively big and the particles can converge to the best solution quickly. At the end of search, the weighting coefficient is relatively small and the particles can search carefully around the best solution.

To avoid getting into local optimum, after certain time, small magnitude random disturbance is added to the position of particles. The probability of position disturbance is 50%,

$$\mathbf{X}_i^{t+1} = \mathbf{X}_i^{t+1} \times [\text{ones}(1, m) - \mathbf{E}],$$

$$\mathbf{E} = [\epsilon_1, \epsilon_2, \dots, \epsilon_m],$$

$$|\epsilon_j| < \delta, \quad j = 1, 2, \dots, m \quad (6)$$

where $\text{ones}(1, m)$ is a unit vector of 1-row and m -column. ϵ_j is a random number. δ is a very small positive number, here $\delta = 0.005$.

3 LSO modeling

Least squares estimator is widely used and offers computational and theoretical advantages. As a classical data processing method, least squares algorithm is applied to lots of engineering domain, and it is a main approach to estimate parameters from the test data.

In a nonlinear least squares problem,

$$\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_q(\mathbf{x})]^T$$

where $f_i(\mathbf{x})$ ($i = 1, 2, \dots, q$) is a vector function of $\mathbf{x} \in \mathbf{R}^m$.

Finding a vector $\mathbf{x}$ that

$$\min_{\mathbf{x} \in \mathbf{R}^m} S(\mathbf{x}) \quad (7)$$

$$S(\mathbf{x}) = \sum_{i=1}^q f_i^2(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|_2^2$$

This is called a nonlinear least squares minimization problem.

If there exists $\mathbf{x}^*$ that makes $f(\mathbf{x}^*) \approx 0$, then

$$\|\mathbf{F}(\mathbf{x}^*)\|_2^2 \approx 0$$

For

$$\forall \mathbf{x} \in \mathbf{R}^m, \quad S(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|_2^2 \geq 0,$$

$$S(\mathbf{x}^*) = \min_{\mathbf{x} \in \mathbf{R}^m} S(\mathbf{x}) = \min_{\mathbf{x} \in \mathbf{R}^m} \|\mathbf{F}(\mathbf{x})\|_2^2 \quad (8)$$

Then $S(\mathbf{x})$ gets the minimum value at $\mathbf{x}^*$. $\mathbf{x}^*$ is the least squares solution of $\mathbf{F}(\mathbf{x}) = 0$.

To get the least squares solution, let

$$\nabla S(\mathbf{x}) = 0 \quad (9)$$

where $\nabla S(\mathbf{x}) = 2\mathbf{D}\mathbf{F}(\mathbf{x})^T \mathbf{F}(\mathbf{x})$, then

$$\mathbf{D}\mathbf{F}(\mathbf{x})^T \mathbf{F}(\mathbf{x}) = 0 \quad (10)$$

Which can be written as

$$[\nabla f_1(\mathbf{x}), \dots, \nabla f_q(\mathbf{x})] \mathbf{F}(\mathbf{x}) = 0 \quad (11)$$

The least squares solution in Eq. (11) is the optimum solution of the least squares problem in Eq. (7). In nonlinear problem, the main methods are Newton-Raphson (N-R) algorithm, Gauss-Newton (G-N) algorithm, and Levenberg-Marquarat (L-M) algorithm. Those methods all belong to conjugate gradient algorithm. They all must start from certain initial value $\mathbf{x}_0 \in \mathbf{R}^m$, and process iterative search following the steps:

- (1) Select the descent direction $\mathbf{P}_0$ that makes $\nabla S(\mathbf{x}_0)^T \mathbf{P}_0 < 0$;
- (2) Select step factor α , let $\mathbf{x}_1 = \mathbf{x}_0 + \alpha \mathbf{P}_0$, and $S(\mathbf{x}_1) < S(\mathbf{x}_0)$;
- (3) Take $\mathbf{x}_1$ as the first approximation of $\mathbf{x}^*$, and replace $\mathbf{x}_0$ with $\mathbf{x}_1$, go back to step (2) until convergence rule is satisfied.

The initial value $\mathbf{x}_0$ directly influences the convergence velocity and the optimum solution obtained.

Let

$$\begin{aligned} \mathbf{x} &= [a_{11}, a_{12}, a_{21}, a_{22}, \\ &b_{11}, b_{12}, b_{13}, b_{14}, b_{21}, b_{22}, b_{23}, b_{24}] \\ f_i(\mathbf{x}) &= \delta_{ijkp}(\mathbf{x}) \\ j &= 1, 2; \quad k = 1, \dots, L; \quad p = 1, 2, 3, 4 \\ \delta_{1kp}(\mathbf{x}) &= (\tilde{N}_{1p}[k] - \bar{N}_{1p}[k]) / g_{1p} \\ \delta_{2kp}(\mathbf{x}) &= (\tilde{N}_{2p}[k] - \bar{N}_{2p}[k]) / g_{2p} \end{aligned} \quad (12)$$

Then the least squares index of Eq. (12) equals to Eq. (4) and from its best solution $\mathbf{x}^*$, we can get matrices $\mathbf{A}$ and $\mathbf{B}$.

4 Simulation

Let PSO runs up to 1 000 generations, set the maximum weighting coefficient $\omega_{\max} = 0.9$, the minimum weighting coefficient $\omega_{\min} = 0.4$,

the maximum value of particle position $\mathbf{X}_{\max} = 5 \times \text{ones}(1, 12)$, the minimum value of particle position $\mathbf{X}_{\min} = -5 \times \text{ones}(1, 12)$, the maximum velocity $\mathbf{v}_{\max} = 0.2 \times \text{ones}(1, 12)$, the minimum velocity $\mathbf{v}_{\min} = -0.2 \times \text{ones}(1, 12)$, where $\text{ones}(1, 12)$ is a unit vector of 1-row and 12-column. The positions and velocities of the initial generation particles are generated randomly within these ranges. Then we can start our PSO. The cost function changing with the iteration is shown in Fig. 1. Because the cost function varies in a large range, the vertical scale is the logarithm of the cost function. Referring to Fig. 1, after the evolution runs up to 500 generations, the cost function doesn't change much, and the best $J = 19.68$. It implies that PSO converges to its optimum. We can get matrices $\mathbf{A}$ and $\mathbf{B}$ from the optimum particle. Let $\mathbf{C}$ be an identity matrix and $\mathbf{D}$ be a zero matrix, then we get a SVM with N_1 and N_2 as its outputs. N_1 responses of CLM and SVM are shown in Fig. 2. N_2 responses of CLM and SVM are shown in Fig. 3. In these figures, “~” denotes the response of CLM and “—” denotes the response of SVM. Referring to Fig. 2 and Fig. 3, the responses errors between SVM and CLM are not acceptable.

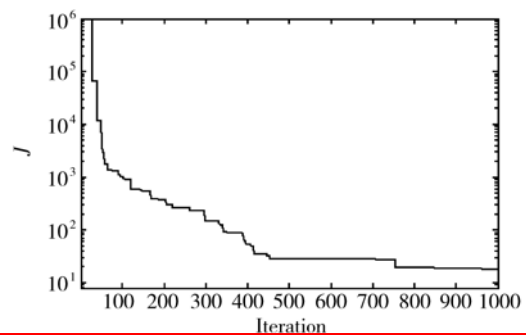
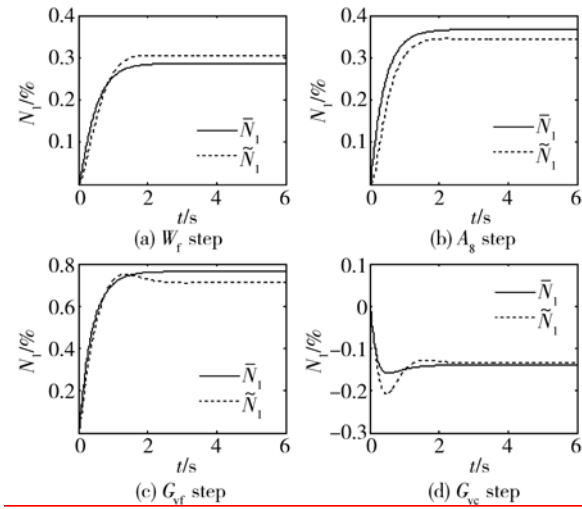
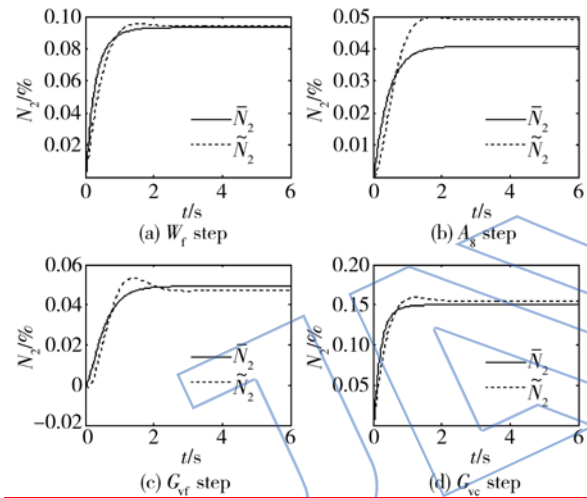


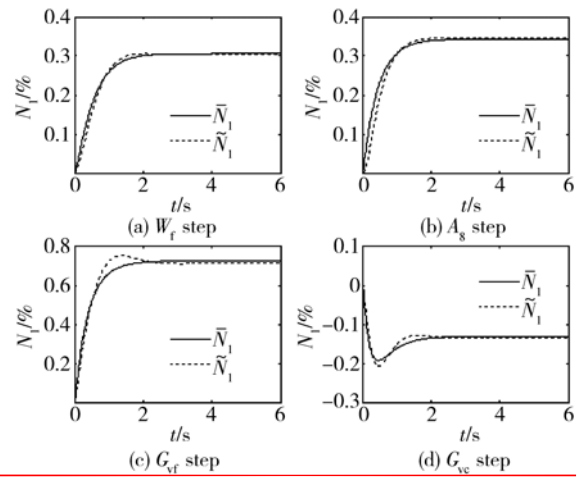
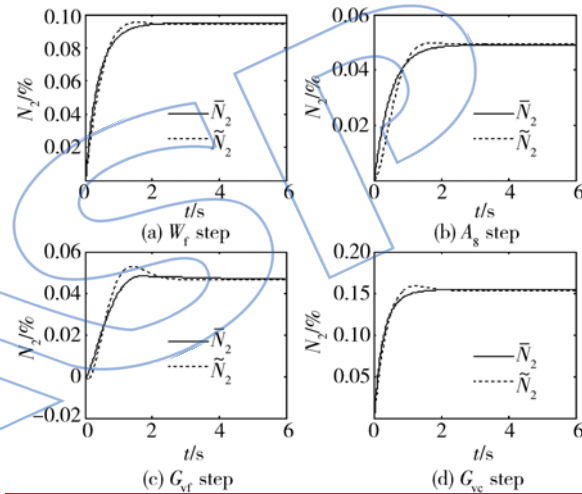
Fig. 1 Cost function of PSO

LSO is sensitive to its initial value. If the initial value is unreasonable, it is difficult to get an optimum solution. When there are many parameters to be estimated, it is difficult to set them all reasonable. Therefore, LSO is adopted “in series” with PSO, and takes PSO optimum solution as its initial value. This forms a hybrid optimization algorithm.


 Fig. 2 N_1 responses of PSO

 Fig. 3 N_2 responses of PSO

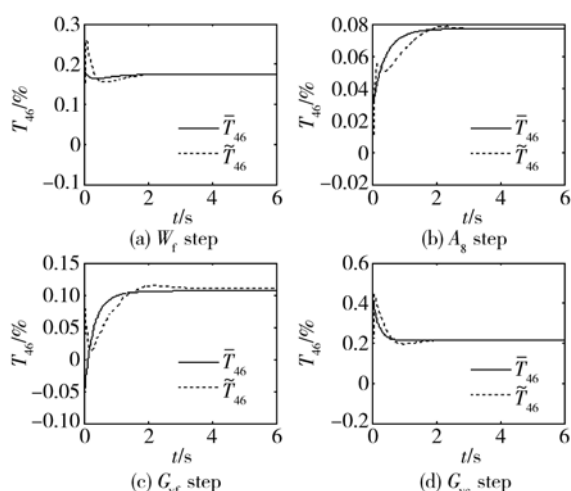
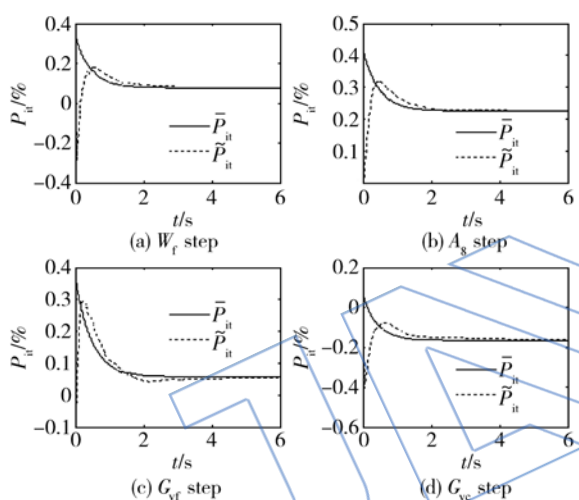
L-M algorithm is an extension of G-N optimization technique. It is more powerful and seldom stuck in the local minima. So we use the standard L-M algorithm to solve the least squares problem in Eq. (12). Then matrices $\mathbf{A}$ and $\mathbf{B}$ can be gotten from the optimum solution of hybrid optimization algorithm, and $J=5.37$. N_1 responses of nonlinear and linear model are shown in Fig. 4. N_2 responses of nonlinear and linear model are shown in Fig. 5. We can see that the dynamic responses errors between the nonlinear and the linear are reduced and acceptable.

When estimating the parameters of $\mathbf{C}$ and $\mathbf{D}$, the matrices $\mathbf{A}$ and $\mathbf{B}$ are already gotten by hybrid optimization algorithm. The particle


 Fig. 4 N_1 responses of hybrid optimization algorithm

 Fig. 5 N_2 responses of hybrid optimization algorithm

formed by the parameters of $\mathbf{C}$ and $\mathbf{D}$ is $\mathbf{X}_i = [c_{31i}, c_{32i}, c_{41i}, c_{42i}, d_{31i}, d_{32i}, d_{33i}, d_{34i}, d_{41i}, d_{42i}, d_{43i}, d_{44i}]$. Then $\mathbf{C}$ and $\mathbf{D}$ can be obtained by hybrid optimization algorithm and form the final SVM with $\mathbf{A}$ and $\mathbf{B}$. The responses of SVM and CLM of T_{46} and P_{it} are shown in Fig. 6 and Fig. 7. Referring to Fig. 6 and Fig. 7, we can see that the responses of SVM fit the responses of CLM well. SVM obtained by hybrid optimization algorithm is valid.

To demonstrate the ability of hybrid optimization algorithm, we do some further tests about the approach. Here we use PSO, LSO and hybrid optimization algorithm to estimate the parameters of aero-engine within different ranges. When estimating parameters of $\mathbf{A}$ and $\mathbf{B}$, let the parameters vary with $[-10, 10]$, PSO gets $J=$

Fig. 6 T_{46} responses by hybrid optimization algorithmFig. 7 P_{ti} responses by hybrid optimization algorithm

40.02. Hybrid optimization algorithm gets $J = 5.37$. Run LSO 20 times with a random initial value within $[-10, 10]$, the list of J gotten by LSO is shown in Table 1, where “Inf” refers to the value greater than 10^{10} .

When the parameters vary within $[-20, 20]$, PSO gets $J = 53.65$, hybrid optimization algorithm gets $J = 5.37$, the list of J gotten by LSO starts from the random initial value within $[-20, 20]$ is shown in Table 2.

When the parameters vary within $[-100, 100]$, PSO gets $J = 476.35$, hybrid optimization algorithm gets $J = 5.42$, and LSO can not finish 20 times search with program error arising.

The results show that when the parameters vary in a wide ranges, neither PSO nor LSO can get satisfied solution, but hybrid optimization

algorithm can get a satisfied solution.

Table 1 List of J gotten by LSO with random initial value within $[-10, 10]$

No	1	2	3	4	5
J	7.58	5.65	Inf	23.99	5.76
No	6	7	8	9	10
J	5.52	334.81	Inf	22.80	9.77
No	11	12	13	14	15
J	10.73	5.42	17.57	5.59	9.00
No	16	17	18	19	20
J	5.37	5.69	15.11	5.38	7.24

Table 2 List of J gotten by LSO with random initial value within $[-20, 20]$

No	1	2	3	4	5
J	1986.59	16.03	31.93	49.60	Inf
No	6	7	8	9	10
J	Inf	21.35	Inf	Inf	21.22
No	11	12	13	14	15
J	30.66	14.39	22.51	21.34	33.24
No	16	17	18	19	20
J	49.83	25.18	59.30	100.70	Inf

5 Conclusion

A hybrid optimization algorithm based on PSO and LSO is proposed in this paper to estimate the parameters of SVM for aero-engine. The global searching ability of PSO is used to get an initial value for LSO. LSO starts from its initial value and follows L-M algorithm to get an optimum solution of SVM. When the single optimization algorithm can not get a satisfied result, hybrid optimization algorithm is a good alternative. When applied to parameters estimation of SVM for aero-engine, it gets satisfied result. This hybrid optimization algorithm can be used to other parameters optimization problems especially in complex problems with a larger number of variables to be optimized and variables varied in wide ranges.

References :

- [1] ZHAO Jiaqing, WANG Ling. Center based genetic algorithm and its application to the stiffness equivalence of the aircraft wing[J]. Expert Systems with Applications, 2011, 38(5):6254-6261.
- [2] YANG Jingan, ZHUANG Yanbin. An improved ant colony optimization algorithm for solving a complex combinatorial optimization problem[J]. Applied Soft Computing, 2010, 10(2):653-660.
- [3] MA Hai, WANG Yanjiang. An artificial fish swarm algorithm based on chaos search[C]// The 5th International Conference on Natural Computation (ICNC'09). Tianjin: IEEE, 2009:118-121.
- [4] Rahimi-Vahed A, Mirzaei A H. A hybrid multi-objective shuffled frog-leaping algorithm for a mixed-model assembly line sequencing problem[J]. Computers & Industrial Engineering, 2007, 53(4):642-666.
- [5] Mohammadi A, Ziarati K. A powerful bee swarm optimization algorithm[C]// The 13th IEEE International Multi-topic Conference. Islamabad: IEEE, 2009:1-6.
- [6] ZHAN Zhihui, ZHANG Jun, LI Yun, et al. Adaptive particle swarm optimization[J]. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 2009, 39(6):1362-1381.
- [7] Ling S H, Iu H, Leung F H F, et al. Improved hybrid particle swarm optimized wavelet neural network for modeling the development of fluid dispensing for electronic packaging[J]. Industrial Electronics, IEEE Transactions on, 2008, 55(9):3447-3459.
- [8] Sentinella M R, Casalino L. Hybrid evolutionary algorithm for the optimization of interplanetary trajectories[J]. Journal of Spacecraft and Rockets, 2009, 46(2):365-372.
- [9] Sugiyama N. Derivation of ABCD system matrices from nonlinear dynamic simulation of jet engines[R]. AIAA-1992-3319, 1992.
- [10] FENG Zhengping, SUN Jianguo. A new method for establishing a state space model of aeroengine[J]. Journal of Aerospace Power, 1998, 13(4):435-438. (in Chinese)
- [11] FENG Zhengping, SUN Jianguo. Modeling of small perturbation state variable model for aeroengines[J]. Journal of Propulsion Technology, 2001, 22(1):54-57. (in Chinese)
- [12] YUAN Chunfei, YAO Hua, YANG Gang. On-board real-time adaptive model of aero-engine[J]. Chinese Journal of Aeronautics, 2006, 27(4):561-564. (in Chinese)
- [13] LI Qiu-hong, SUN Jianguo. Aero-engine state variable modeling based on the genetic algorithm[J]. Journal of Aerospace Power, 2006, 21(2):427-431. (in Chinese)