SCIENCE CHINA
Information Sciences

• **RESEARCH PAPER** •

January 2013, Vol. 56 012202:1–012202:12
doi: 10.1007/s11432-012-4754-9

Parameters identification of UCAV flight control system based on predator-prey particle swarm optimization

DUAN HaiBin[1]*, YU YaXiang[1] & ZHAO ZhenYu[2]

[1]*Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University, Beijing* 100191*, China;*
[2]*Science and Technology on Electron-Optic Control Laboratory, Luoyang* 410079*, China*

Received June 24, 2012; accepted October 30, 2012

Abstract With the improvement of the aircraft flight performance and development of computing science, uninhabited combat aerial vehicle (UCAV) could accomplish more complex tasks. But this also put forward stricter requirements for the flight control system, which are the crucial issues of the whole UCAV system design. This paper proposes a novel UCAV flight controller parameters identification method, which is based on predator-prey particle swarm optimization (PSO) algorithm. A series of comparative experimental results verify the feasibility and effectiveness of our proposed approach in this paper, and a predator-prey PSO-based software platform for UCAV controller design is also developed.

Keywords uninhabited combat aerial vehicle (UCAV), particle swarm optimization (PSO), parameters identification, predator-prey

Citation Duan H B, Yu Y X, Zhao Z Y. Parameters identification of UCAV flight control system based on predator-prey particle swarm optimization. Sci China Inf Sci, 2013, 56: 012202(12), doi: 10.1007/s11432-012-4754-9

1 Introduction

Flight control system, which is a complicated multi-input, multi-output and time-varying nonlinear system, is the core of the simulation training system design of uninhabited combat aerial vehicle (UCAV), which also determines the whole system's performance directly [1,2]. For the existence of strong coupling among the inputs and the nonexistence of mapping relationship between the performance index and the controller parameter, the selection of the controller parameters is a very tough problem in the design of the flight control system [3]. Presently, cut and try method is commonly used to identify all the control loop parameters of flight control system. But this design method is low efficient and, to a great extent, much depends on the experience of the designer while the flight control system will be more complex with the improvement of the aircraft performances, and these are becoming the bottleneck of the flight control system design [4,5].

*Corresponding author (email: hbduan@buaa.edu.cn)

Recently, many researchers focus on the method of automatic finding optimal controller parameters for the system by computer, which has fast operation and strong ability of logic. Identifying controller parameters based on bio-inspired computing methods become a special hot topic. Genetic algorithm (GA) is utilized to optimize the selection of the fixed control parameters of the flight control systems in [6]. Juang [7] designs an intelligent aircraft automatic landing controller that uses recurrent neural networks with genetic algorithms to improve the performance of conventional automatic landing system. In order to reduce the design effort of nonlinear flight control laws, Oosterom [8] studies an automated procedure by using fuzzy clustering to the design of a longitudinal control law in a fly-by-wire flight control system. Qi [9] applies a multivariable proportional-integral-derivative (PID) neural network to design the flight control system of small-scale unmanned helicopter on the hardware platform.

However, the above methods have deficiencies of slow convergence and complex computing procedure. Some researchers try to use other intelligent algorithms to solve this problem. Duan [10] applies an improved ant colony optimization algorithm (ACO) to identify PID parameters for three degree-of-freedom (3-DOF) high precision flight simulator. Sun [11] presents a design method of a large envelope wavelet neural network gain scheduling flight control law based on particle swarm optimization (PSO) algorithm.

PSO is a relatively recent heuristic algorithm inspired by the study of artificial life and psychological research [12]. Eberhart et al. developed PSO in 1995 [13], which is based on the analogy of the swarm of birds and the school of fish. It is well known that the PSO techniques can provide a high-quality solution with simple implementation and fast convergence [14]. As it is very simple, PSO theory is easy to be applied to the optimal problems. Especially, while the flight control system become more complicated, PSO is the most convenient and efficient method to identify the parameters of the control system. According to Angeline, there is no penalty mechanism for particle which reaches a particular level in a search, so the basic PSO has problems of quick converging towards a local optimal solution [15]. In [16,17], a hybrid predator-prey PSO is firstly proposed, which allows the search to escape from local optima and search in different zones of the search space.

While using PSO to optimize flight control systems, there are two problems to be solved. Firstly, the UCAV model selected is very important. Secondly, it is very crucial for PSO algorithm to choose the fitness function, because there is no obvious mapping relationship between the property index and the UCAV controller. Therefore, how to evaluate each particle's quality turns out to be a key issue which must to be resolved.

The remainder of this paper is organized as follows. The next section introduces mathematic model of 5-DOF for the UCAV. Then, in Section 3, we describe how to design the flight control system in MATLAB environment. Predator-prey PSO algorithm is presented in Section 4, and the proposed method is applied to UCAV flight control system design in Section 5. The comparative results and platform are given in Section 6. Finally, our concluding remarks are contained in Section 7.

2 Mathematical model of UCAV

The 6-DOF nonlinear model of UCAV is illustrated in this section, which is the prerequisite for simplifying and linearizing the mathematical model [18].

2.1 Nonlinear equations of 6-DOF modeling

UCAV nonlinear equations of 6-DOF can be deduced by the aerodynamic and kinematical equations as follows.

$$\begin{aligned}\dot{V} &= \frac{1}{m}(P_x\cos\alpha\cos\beta - P_y\sin\alpha\cos\beta + P_z\sin\beta + Z\sin\beta - Q)\\ &\quad -g(\cos\alpha\cos\beta\sin\vartheta - \sin\alpha\cos\beta\cos\vartheta\cos\gamma - \sin\beta\cos\vartheta\sin\gamma, \qquad (1)\\ \dot{\alpha} &= -\frac{1}{mV\cos\beta}\left(P_x\sin\alpha + P_y\cos\alpha + Y\right) + \omega_z - \tan\beta\left(\omega_x\cos\alpha - \omega_y\sin\alpha\right)\frac{\delta y}{\delta x}\\ &\quad +\frac{g}{V\cos\beta}\left(\sin\alpha\sin\vartheta + \cos\alpha\cos\vartheta\cos\gamma\right), \qquad (2)\end{aligned}$$

$$\begin{aligned}\dot{\beta} = {} & \frac{1}{mV}\left(-P_x\cos\alpha\sin\beta + P_y\sin\alpha\sin\beta + P_z\cos\beta + Z\right) + \omega_x\sin\alpha \\ & \omega_y\cos\alpha + \frac{g}{V}\left(\cos\alpha\sin\beta\sin\vartheta - \sin\alpha\sin\beta\cos\vartheta\cos\gamma + \cos\beta\sin\gamma\cos\vartheta\right),\end{aligned} \tag{3}$$

where m is the mass of UCAV; α is attack angle; β is sideslip angle; ϑ is pitch angle; γ is roll angle. P is engine thrust; X, Y, Z are the projections of aerodynamic force in body axis. ω_x, ω_y, ω_z denote the coordinate components of palstance. These three equations already contain three forces in the body axis which are generated by the thrust vector.

$$\dot{\omega}_x = b_{11}\omega_y\omega_z + b_{12}\omega_x\omega_z + \frac{I_y\left(M_x + M_{px}\right) + I_{xy}\left(M_y + M_{py}\right)}{I_xI_y - I_{xy}^2}, \tag{4}$$

$$\dot{\omega}_y = b_{21}\omega_y\omega_z + b_{22}\omega_x\omega_z + \frac{I_{xy}\left(M_x + M_{px}\right) + I_x\left(M_y + M_{py}\right)}{I_xI_y - I_{xy}^2}, \tag{5}$$

$$\dot{\omega}_z = \frac{I_x - I_y}{I_z}\omega_x\omega_y + \frac{I_{xy}}{I_z}\left(\omega_x^2 - \omega_y^2\right) + \frac{\left(M_z + M_{pz}\right)}{I_z}, \tag{6}$$

$$b_{11} = \frac{I_y^2 - I_yI_z - I_{xy}^2}{I_xI_y - I_{xy}^2}, \tag{7}$$

$$b_{22} = \frac{I_xI_z - I_x^2 - I_{xy}^2}{I_xI_y - I_{xy}^2}, \tag{8}$$

$$b_{12} = \frac{I_{xy}\left(I_z - I_y - I_x\right)}{I_xI_y - I_{xy}^2}, \tag{9}$$

$$b_{21} = \frac{I_{xy}\left(I_y - I_z - I_x\right)}{I_xI_y - I_{xy}^2}, \tag{10}$$

where I_x, I_y, I_z and M_x, M_y, M_z denote the coordinate components of inertia moment and resultant moment respectively.

In the body axis, we have

$$\dot{\gamma} = \omega_x - \tan\vartheta\left(\omega_y\cos\gamma - \omega_z\sin\gamma\right), \tag{11}$$

$$\dot{\vartheta} = \omega_y\sin\gamma + \omega_z\cos\gamma, \tag{12}$$

$$\dot{\psi} = \frac{1}{\cos\vartheta}\left(\omega_y\cos\gamma - \omega_z\sin\gamma\right), \tag{13}$$

where ψ is drift angle. Aerodynamic equations can be described as

$$\begin{aligned}& Y = C_yqS, \quad C_y = C_y(\alpha,\delta_z), \quad Z = \sum C_zqS, \quad \sum C_z = C_z(\alpha,\delta_x) + C_z(\alpha,\delta_y), \\ & Q = C_xqS, \quad C_x = C_x(\alpha,\delta_z),\end{aligned}$$

where δ_x, δ_y, δ_z are the coordinate components of deflection angles of the controlling surface. Aerodynamic moments can be given by

$$\begin{aligned}& M_x = \sum m_xqsl, \quad \sum m_x = m_x^\beta\beta + m_x\left(\alpha,\delta_x\right) + m_x\left(\alpha,\delta_y\right) + m_x^{\omega_x}\omega_x\frac{l}{2V} + m_x^{\omega_y}\omega_y\frac{l}{2V}, \\ & M_y = \sum m_yqsl, \quad \sum m_y = m_y^\beta\beta + m_y\left(\alpha,\delta_x\right) + m_y\left(\alpha,\delta_y\right) + m_y^{\omega_x}\omega_x\frac{l}{2V} + m_y^{\omega_y}\omega_y\frac{l}{2V}, \\ & M_z = \sum m_zqsb_A, \quad \sum m_z = m_z\left(\alpha,\delta_z\right) + m_z^{\omega_z}\omega_z\frac{b_A}{V} + m_z^{\dot{\alpha}}\dot{\alpha}\frac{b_A}{V}.\end{aligned}$$

When the height and mach are fixed, aerodynamic coefficients $C_y(\alpha,\delta_z)$, $C_x(\alpha,\delta_z)$, $C_z(\alpha,\delta_x)$, $C_z(\alpha,\delta_y)$, $m_x(\alpha,\delta_x)$, $m_x(\alpha,\delta_y)$, $m_y(\alpha,\delta_x)$, $m_y(\alpha,\delta_y)$, $m_z(\alpha,\delta_z)$ are the functions of the height, mach, attack angle and control surface. Aerodynamic derivatives $m_z^{\omega_z}$, $m_z^{\dot{\alpha}}$, m_x^β, $m_x^{\omega_x}$, $m_x^{\omega_y}$, m_y^β, $m_y^{\omega_x}$, $m_y^{\omega_y}$ are specified values.

2.2 Nonlinear equations of 5-DOF modeling

Suppose that UCAV equations can be simplified into nonlinear equations of 5-DOF if the thrust and the resistance of the aircraft are the same. Without consideration of the thrust vector, we have $\dot{V}=0,\ P=Q,\ P_x=P\ll G,\ P_y=P_z=0$. On this condition, the equations of the UCAV speed level off [18]. At the same time, it is assumed that $I_{xy}\ll I_xI_y-I_{xy}^2$, state variable $\boldsymbol{x}=(\alpha,\beta,\omega_x,\omega_y,\omega_z)^{\mathrm{T}}$. The UCAV nonlinear equations of 5-DOF are represented as follows:

$$\dot{\alpha}=-\frac{P\sin\alpha+Y}{mV\cos\beta}+\omega_z-\tan\beta\left(\omega_x\cos\alpha-\omega_y\sin\alpha\right)+\frac{g}{V\cos\beta}\left(\sin\alpha\sin\vartheta+\cos\alpha\cos\vartheta\cos\gamma\right), \tag{14}$$

$$\begin{aligned}\dot{\beta}=&\frac{Z-P\cos\alpha\sin\beta}{mV}+\omega_x\sin\alpha+\omega_y\cos\alpha\\&+\frac{g}{V}\left(\cos\alpha\sin\beta\sin\vartheta-\sin\alpha\sin\beta\cos\vartheta\cos\gamma+\cos\beta\sin\gamma\cos\vartheta\right),\end{aligned} \tag{15}$$

$$\dot{\omega}_x=\frac{I_y^2-I_yI_z-I_{xy}^2}{I_xI_y-I_{xy}^2}\omega_y\omega_z+\frac{I_{xy}\left(I_z-I_y-I_x\right)}{I_xI_y-I_{xy}^2}\omega_x\omega_z+\frac{I_y\sum M_x}{I_xI_y-I_{xy}^2}=b_{11}\omega_y\omega_z+b_{12}\omega_x\omega_z+\frac{I_yM_x}{I_xI_y-I_{xy}^2}, \tag{16}$$

$$\dot{\omega}_y=\frac{I_{xy}\left(I_y-I_z-I_x\right)}{I_xI_y-I_{xy}^2}\omega_y\omega_z+\frac{I_xI_z-I_x^2-I_{xy}^2}{I_xI_y-I_{xy}^2}\omega_x\omega_z+\frac{I_x\sum M_y}{I_xI_y-I_{xy}^2}=b_{21}\omega_y\omega_z+b_{22}\omega_x\omega_z+\frac{I_xM_y}{I_xI_y-I_{xy}^2}, \tag{17}$$

$$\dot{\omega}_z=\frac{I_x-I_y}{I_z}\omega_x\omega_y+\frac{I_{xy}}{I_z}\left(\omega_x^2-\omega_y^2\right)+\frac{\sum M_z}{I_z}=\frac{I_x-I_y}{I_z}\omega_x\omega_y+\frac{I_{xy}}{I_z}\left(\omega_x^2-\omega_y^2\right)+\frac{M_z}{I_z}. \tag{18}$$

2.3 Linearization modeling

In most cases, the UCAV maintains steady straight level flight, Eqs. (14)–(18) can be modeled as linear time invariant state-space perturbation models, with the nominal trajectory being steady-level trimmed flight. The UCAV's linear equations are as follows:

$$\Delta\dot{\alpha}=\Delta\omega_z-\left(Y^{\alpha}\Delta\alpha+\Delta\delta_Z\right), \tag{19}$$

$$\Delta\dot{\omega}_z=M_z^{\alpha}\Delta\alpha+M_z^{\omega_z}\Delta\omega_z+M_z^{\delta_z}\Delta\delta_z, \tag{20}$$

$$\Delta\dot{\beta}=\Delta\omega_y+Z^{\delta_x}\Delta\delta_x+Z^{\delta_y}\Delta\delta_y, \tag{21}$$

$$\Delta\dot{\omega}_x=M_x^{\beta}\Delta\beta+M_x^{\omega_x}\Delta\omega_x+M_x^{\omega_y}\Delta\omega_y+M_x^{\delta_x}\Delta\delta_x+M_x^{\delta_y}\Delta\delta_y, \tag{22}$$

$$\Delta\dot{\omega}_y=M_y^{\beta}\Delta\beta+M_y^{\omega_x}\Delta\omega_x+M_y^{\omega_y}\Delta\omega_y+M_y^{\delta_x}\Delta\delta_x+M_y^{\delta_y}\Delta\delta_y. \tag{23}$$

We have the state equations

$$\dot{\boldsymbol{x}}=\boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{u},\quad \boldsymbol{y}=\boldsymbol{C}\boldsymbol{x},$$

where the state variable $\boldsymbol{x}=(\alpha,\omega_z,\beta,\omega_x,\omega_y)^{\mathrm{T}}$ and the control surface $\boldsymbol{u}=(\delta_z,\delta_x,\delta_y)^{\mathrm{T}}$, and $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ can be denoted by

$$\boldsymbol{A}=\begin{bmatrix}-Y^{\alpha}&1&0&0&0\\M_z^{\alpha}&M_z^{\omega_z}&0&0&0\\0&0&Z^{\beta}&0&1\\0&0&M_x^{\beta}&M_x^{\omega_x}&M_x^{\omega_y}\\0&0&M_y^{\beta}&M_y^{\omega_x}&M_y^{\omega_y}\end{bmatrix},\quad \boldsymbol{B}=\begin{bmatrix}-Y\delta_z&0&0\\M_z^{\delta_z}&0&0\\0&Z^{\delta_x}&Z^{\delta_y}\\0&M_x^{\delta_x}&M_x^{\delta_y}\\0&M_y^{\delta_x}&M_y^{\delta_y}\end{bmatrix},\quad \boldsymbol{C}=\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix}.$$

3 UCAV flight control system design in MATLAB environment

Based on control augmentation system of UCAV, the aircraft linear equations are generally obtained by a series of equilibrium points. The flight envelope of this UCAV must satisfy $0\leqslant H\leqslant 18$ km and $0.6\leqslant M\leqslant 2.2$ [18]. Figure 1 shows the schematic diagram of UCAV system.

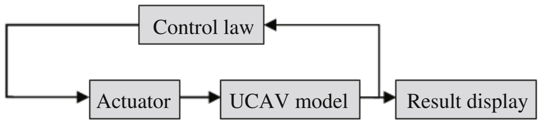


Figure 1 UCAV system model.

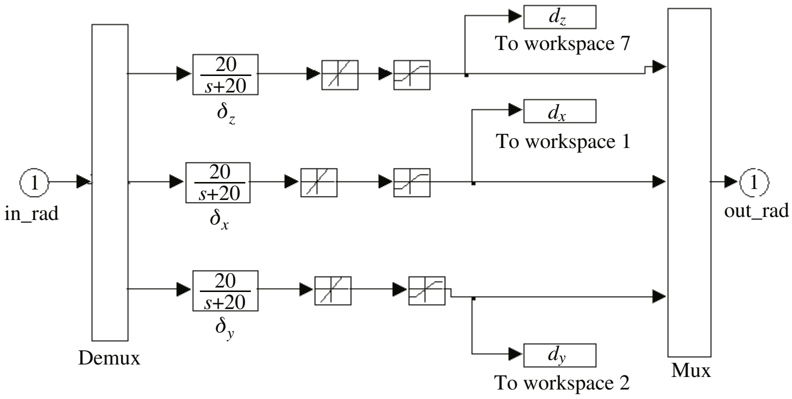


Figure 2 Actuator module of UCAV in MATLAB environment.

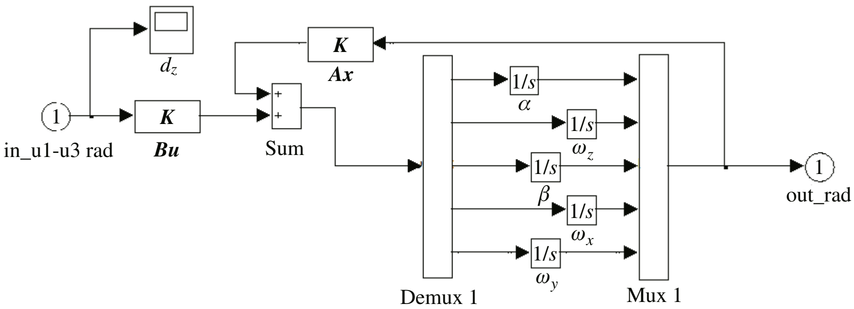


Figure 3 UCAV model module in MATLAB environment.

As is obvious in Figure 1, the UCAV system is comprised of four subsystems: control law, actuator, mathematical model and simulation result display.

Matrix $\boldsymbol{K}$ for the function of control law can be obtained from MATLAB main program. Actuator module is responsible for control adjusting, which can also limit the amplitudes of control surface deflection angle. The actuator module can be shown with Figure 2.

The actuator module requires that deflection angle of the elevator must be less than that of aileron and rudder. In order to prevent the deflection angles of the control surface from deflecting too fast, the angle authority of the elevator is set at $-18°$ to $12°$, aileron and rudder at $-25°$ to $25°$, and the angle rate authority of elevator and aileron at $50°$/s, and rudder is at $80°$/s. UCAV model module is displayed in Figure 3.

Matrix A and B in Figure 3 are both obtained from the workspace of MATLAB. Figure 4 shows the simulation result display module.

The control law $\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x}$ is used. $\boldsymbol{K}$ denotes the state feedback gain, and it can be illustrated with the following matrix

$$\boldsymbol{K} = \begin{bmatrix} k_1 & k_2 & 0 & 0 & 0 \\ 0 & 0 & k_3 & k_4 & k_5 \\ 0 & 0 & k_6 & k_7 & k_8 \end{bmatrix}.$$

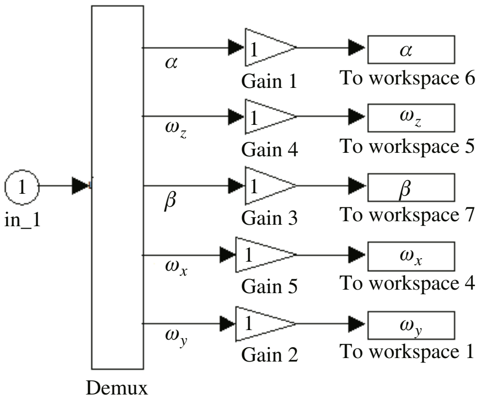


Figure 4 Simulation result display module in MATLAB environment.

4 Predatory-prey PSO

In the Gbest-model of PSO, each particle has its current position and velocity in a space of solution [16]. The best solution found so far are Pbest and Gbest. Each particle aims to get a global optimal solution by current velocity, Pbest and Gbest. Gbest-model can be expressed as

$$v_{ij}(k+1) = \omega v_{ij}(k) + c_1 r_1 [p_i(k) - x_{ij}(k)] + c_2 r_2 [g_i(k) - x_{ij}(k)], \tag{24}$$
$$x_{ij}(k+1) = x_{ij}(k) + v_{ij}(k+1), \tag{25}$$

where $\boldsymbol{v}_i(k)$ and $\boldsymbol{x}_i(k)$ respectively denote the velocity and position of the ith particle at step k, j is the dimension of particle i, c_1 and c_2 are weight coefficients, r_1 and r_2 are random numbers between 0 and 1, $\boldsymbol{p}_i$ is the best position of the ith particle and $\boldsymbol{g}_i$ is the best position which particles have ever found.

Generally, the basic PSO algorithm mentioned above is easily falling to local optimal solutions. In this case, the concept of predator-prey behavior is proposed to improve the basic PSO [16,17]. Predator-prey PSO is a method which takes a cue from the behavior of schools of sardines and pods of killer whales. In this model, particles are divided into two categories, predator and prey. Predators show the behavior of chasing the center of preys' swarm; they look like chasing preys. And preys escape from predators in multidimensional solution space. After taking a tradeoff between predation risk and their energy, escaping particles would take different escaping behaviors. This helps the particles avoid the local optimal solutions and find the global optimal solution.

The velocities of the predator and the prey in the improved PSO can be defined by [16].

$$v_{dij}(k+1) = \omega_d v_{dij}(k) + c_1 r_1 [p_{dij}(k) - x_{dij}(k)] + c_2 r_2 [g_{dj}(k) - x_{dij}(k)] + c_3 r_3 [g_j(k) - x_{dij}(k)], \tag{26}$$
$$\begin{aligned} v_{rij}(k+1) = {} & \omega_r v_{rij}(k) + c_4 r_4 [p_{rij}(k) - x_{rij}(k)] + c_5 r_5 [g_{rj}(k) - x_{rij}(k)] + c_6 r_6 [g_j(k) - x_{rij}(k)] \\ & - P_{\text{asign}} [x_{dij}(k) - x_{rij}(k)] \exp[-b|x_{dij}(k) - x_{rij}(k)|], \end{aligned} \tag{27}$$

where d and r denote predator and prey respectively, $\boldsymbol{p}_{di}$ is the best position of predators, $\boldsymbol{g}_d$ is the best position which predators have ever found. $\boldsymbol{p}_{ri}$ is the best position of preys, $\boldsymbol{g}_r$ is the best position which preys have ever found. $\boldsymbol{g}$ is the best position which all the particles have ever found. ω_d and ω_r can be defined as

$$\omega_d = 0.2 \exp\left(-10 \frac{\text{iteration}}{\text{iteration}_{\max}}\right) + 0.4, \tag{28}$$
$$\omega_r = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{\text{iteration}_{\max}} \text{iteration}. \tag{29}$$

PSO can be also improved by a modification of the inertia weight ω_r in Eq. (29). The inertia weight, whose value is between 0 and 1 (adaptive particle swarm optimization), can be used to balance the local and global search during the optimization process. If the inertia weight is big, it is possible to enhance global search. Otherwise, smaller inertia weight will enhance the local search. In Eq. (29) $\text{iteration}_{\max}$ is maximum iteration, $\omega_{\max}$ and $\omega_{\min}$ are respectively maximum and minimum of ω_r. In this paper, $\omega_{\max}$ and $\omega_{\min}$ are 0.9 and 0.2 respectively. And the definition of I is given by

$$I = \left\{ k | \min_k (|x_{dk} - x_{ri}|) \right\}. \tag{30}$$

Then I denotes the number of the ith prey's nearest predator. In Eq. (27), P is used to decide if the prey escapes or not ($P = 0$ or $P = 1$), and a, b are the parameters which decide the difficulty of the preys escaping from the predators. The closer the prey and the predator, the harder the prey escapes from the predator. a, b are denoted by

$$a = x_{\text{span}}, \quad b = 100/x_{\text{span}}, \tag{31}$$

where x_{span} is the span of the variable.

5 Predatory-prey PSO for identifying controller parameters

The parameters identification of the conventional flight controller can be treated as the typical continual spatial optimization problem [19]. PSO is a novel way for solving the problem. PSO, which is a bio-inspired computation algorithm, can be applied to flight system control to reduce the workload of conventional designer. The bounds of the control gain parameters are set, and PSO searches for the corresponding space automatically to find the optimal parameters. The process in conventional design is conducted manually, now it can be done automatically. Bio-inspired computation can be applied to promote the automation of conventional controller design [20].

Using the proposed predator-prey PSO algorithm to obtain the optimal parameters combination for the UCAV flight control system here, the fitness function is given by

$$J = \frac{1}{2} \int (\boldsymbol{x}' \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{u}' \boldsymbol{R} \boldsymbol{u}) \mathrm{d}t, \tag{32}$$

where $\boldsymbol{x}$ and $\boldsymbol{u}$ are respectively the state vector and the control vector. $\boldsymbol{Q}$ and $\boldsymbol{R}$ are diagonal positive matrix. Here the weighting matrices is chosen as $\boldsymbol{Q} = \text{diag}(50, 10, 20, 30, 30)$, $\boldsymbol{R} = \text{diag}(100, 100, 100)$. The smaller J, the better the particle.

The position vector of the predator and the prey is defined by

$$\boldsymbol{x}_d = (k_1 \ k_2 \ k_3 \ k_4 \ k_5 \ k_6 \ k_7 \ k_8), \quad \boldsymbol{x}_r = (k_1 \ k_2 \ k_3 \ k_4 \ k_5 \ k_6 \ k_7 \ k_8),$$

where $\boldsymbol{x}_d$ and $\boldsymbol{x}_r$ have the constraint of ± 10, which is set according to exact experience.

The process of proposed predator-prey PSO algorithm for solving UCAV controller parameters identification can be described with Figure 5.

The above-mentioned flow chart of the predator-prey PSO algorithm process can also be illustrated with Figure 6.

The complexity of predator-prey PSO algorithm can be computed, and Table 1 shows a comparison of the complexity analysis between basic PSO and predator-prey PSO.

In Table 1, $m = m_d + m_r$, n is the dimension of particle's position. The total complexity of basic PSO and predator-prey PSO can be expressed as

$$T(n)_{\text{basic\_PSO}} = O\left(14 N_{\max} m n\right), \tag{33}$$

$$T(n)_{\text{improved\_PSO}} = O\left(N_{\max} m_r n^2\right). \tag{34}$$

PROCEDURE Optimization of UCAV controller parameter based on the predator-prey PSO
BEGIN
Step 1: Initialization
Set the maximum iteration number $N_{\max}$, the number of the predators m_d and the number of the preys m_r. Initialize randomly the positions and velocities of the predators $\boldsymbol{x}_d$ and $\boldsymbol{v}_d$ respectively, both of which have the same dimensions m_d by 8. So are $\boldsymbol{x}_r$ and $\boldsymbol{v}_r$. And run simulation module of Simulink to compute fitness of each particle. After that, find out the minimum fitness value of the predators as $pbest_d(0)$, that of the preys as $pbest_r(0)$, and that of all the particles as $gbest(0)$.
Step 2:
(1) Let $N = 1$;
(2) Calculate the fitness value of all the particles in iteration k through running simulation module, and then find out the minimum fitness value of the predators as $pbest_d(N)$, that of the preys as $pbest_r(N)$, and that of all the particles as $gbest(N)$.
Step 3:
(1) Let$N \leftarrow N + 1$;
(2) Update all the positions and the velocities according to Eqs. (25)–(27). Then repeat (2) in Step 2.
Step 4: $N \geqslant N_{\max}$?
(1) Yes: stop and output results;
(2) No: go to Step 3.
END

Figure 5 The pseudocode of predator-prey PSO algorithm for UCAV flight controller.

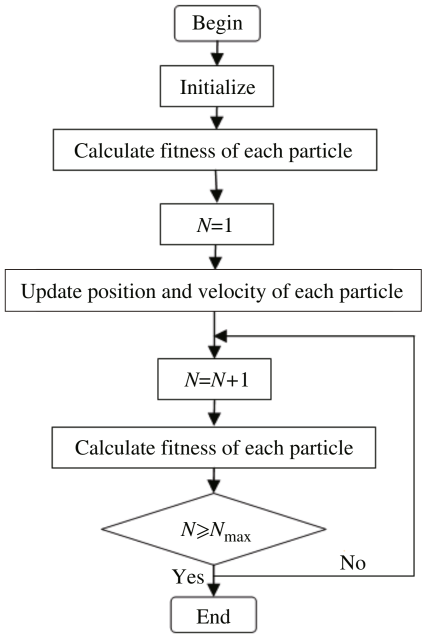


Figure 6 Flow chart of the predator-prey PSO.

Table 1 Comparison of complexity analysis between basic PSO and predator-prey PSO

Step	Operation	Complexity	
		Basic PSO	Predator-prey PSO
1	Initialize	$O\,(2mn)$	$O\,[2(m_d + m_r)n]$
2	Calculate the fitness value of all the particles	$O\,(7m)$	$O\,(7(m_d + m_r))$
3	Update all the positions and the velocities	$O\,(14mn)$	$O\,(14m_d n + (31 + n)m_r n)$
4	Stop and output result	$O\,(1)$	$O\,(1)$

6 Experimental results and platform

In order to investigate the feasibility and effectiveness of the proposed predator-prey PSO approach for identification of UCAV controller parameters, a series of experiments are conducted under some

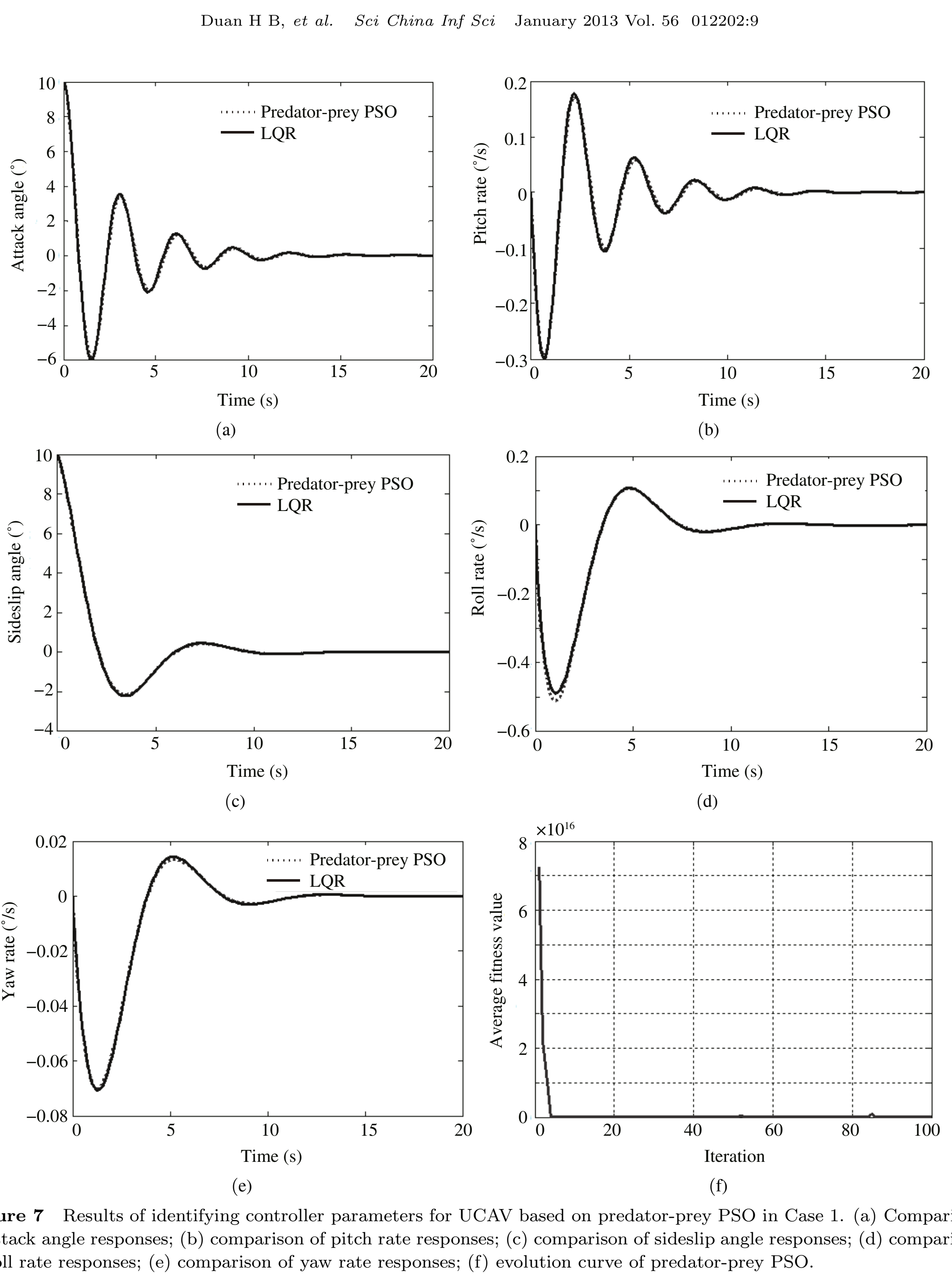


Figure 7 Results of identifying controller parameters for UCAV based on predator-prey PSO in Case 1. (a) Comparison of attack angle responses; (b) comparison of pitch rate responses; (c) comparison of sideslip angle responses; (d) comparison of roll rate responses; (e) comparison of yaw rate responses; (f) evolution curve of predator-prey PSO.

constrained conditions.

The predator-prey PSO algorithm is implemented in a MATLAB 2008a programming environment with an Intel Core 2 PC running Windows XP SP2. No commercial PSO tools are used in these experiments.

Case 1: In this case, the parameter values of predator-prey PSO are set to $\alpha = 10°$, $\beta = 10°$, $N_{\max} = 100$, $t = 20$ s, mach $= 0.8$, $H = 8000$ m, $m_d = 10$, $m_r = 20$ where t is the simulating time of the controller. Comparison of the experiment results between the improved PSO and the LQR method which could directly compute the state feedback gain $\boldsymbol{K}$ are illustrated from Figure 7 (a) to (e). Figure 7(f) shows the evolution curve of the proposed PSO.

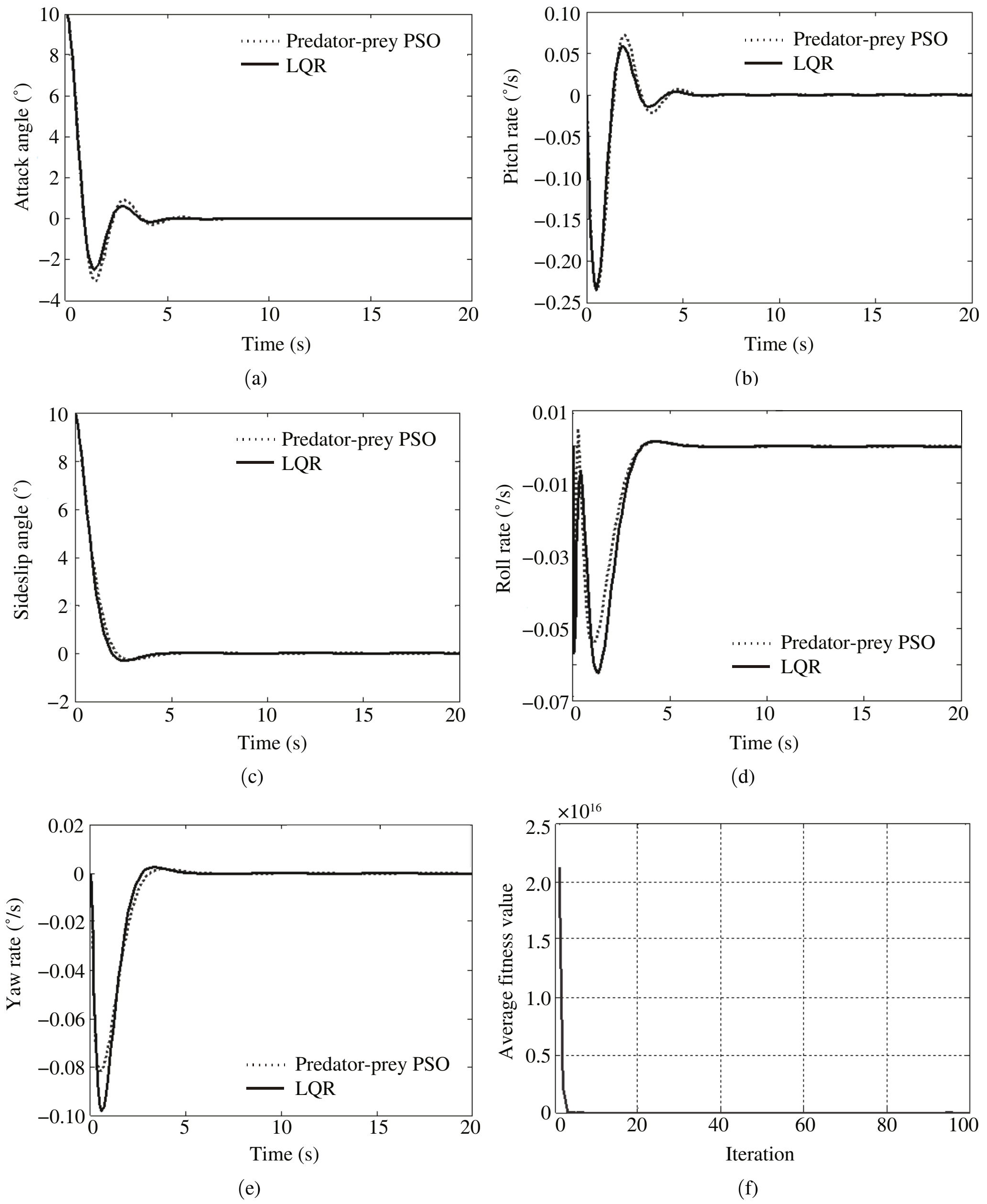
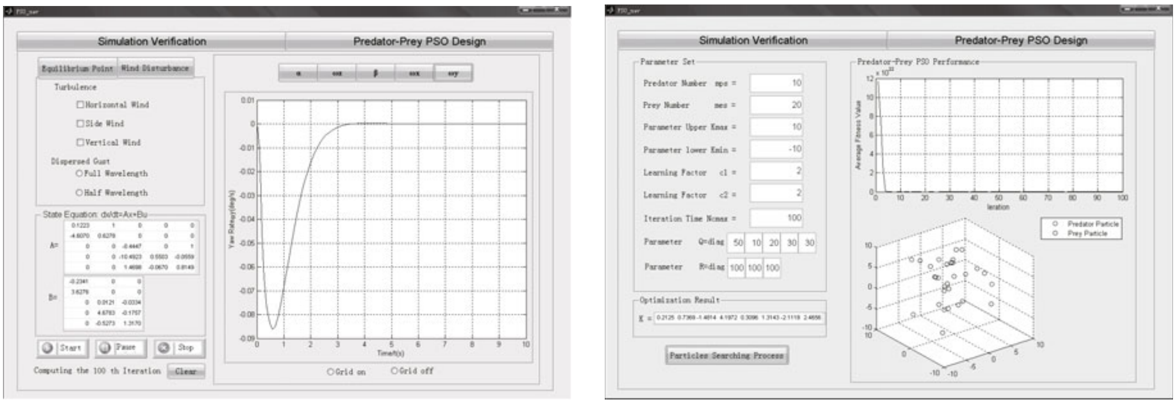


Figure 8 Results of identifying controller parameters for UCAV based on predator-prey PSO in Case 2. (a) Comparison of attack angle responses; (b) comparison of pitch rate responses; (c) comparison of sideslip angle responses; (d) comparison of roll rate responses; (e) comparison of yaw rate responses; (f) evolution curve of predator-prey PSO.

The final optimal result is $\boldsymbol{K} = [-0.1471,\ -0.4012,\ 1.2393,\ -0.4847,\ 0.1373,\ 1.1180,\ -0.3967,\ 4.9621]$, the minimum fitness value $J_{\min} = 223.0907$ and the best iteration best $N = 100$. As indicated in Figure 7, as expected, the proposed algorithm can guarantee that the achieved states are almost the same as the ones obtained by LQR. And the realization of the proposed method is simpler than LQR. Figure 7(f) also demonstrates that the algorithm can converge to the optimal solution quickly.

Case 2: In this case, the parameter values are $\alpha = 10°$, $\beta = 10°$, $N_{\max} = 100$, $t = 20$ s, mach $= 1.2$, $H = 15000$ m, $m_d = 10$, $m_r = 20$ where t is the simulating time of the controller. Comparisons of the experiment results between the improved PSO and the LQR method are illustrated from Figure 8 (a) to (e). Figure 8(f) shows the evolution curve of the proposed PSO.

The final optimal results are $\boldsymbol{K} = [0.0416,\ 0.8298,\ -2.7208,\ 1.4825,\ -1.0683, 1.9002,\ -0.2273,\ 2.6742]$,



(a) (b)

Figure 9 UCAV controller design software platform based on predator-prey PSO. (a) Yaw rate response interface; (b) predator-Prey PSO convergence process interface.

the minimum fitness value $J_{\min} = 68.3361$ and the best iteration best $N = 99$. Although the initial conditions are much different from those of Case 1, the improved algorithm can find the optimal solution.

From these two experimental results, it is obvious that the proposed predator-prey PSO approach could make the UCAV controlling system obtain better performance than the conventional LQR method. The state feedback gain obtained according to the predator-prey PSO can guarantee fast response, precise control, and strong robustness.

Based on the UCAV model and the proposed predator-prey PSO algorithm, we developed a software platform of UCAV controller design. The graphical user interfaces (GUI) of this platform are shown in Figure 9.

7 Conclusions

In this paper, a predator-prey PSO algorithm for identifying parameters of UCAV flight control system is presented which can reduce the workload of the designers during the process of designing complicated U-CAV control system. In this approach, a new fitness function, which is proposed during design procedure, is proven appropriate and efficient by a series of comparative experimental results.

In the future, we will conduct further study on real-time and rapidity performance [21] of this proposed bio-inspired computing method for the hypersonic vehicle flight control system.

Acknowledgements

This work was partially supported by Natural Science Foundation of China (Grant Nos. 61273054, 60975072, 60604009), Program for New Century Excellent Talents in University of China (Grant No. NCET-10-0021) and Aeronautical Foundation of China (Grant No. 20115151019).

References

1 Jin G D, Gu L X. UAV simulator-based simulation of flight control system, In: Proceedings of the 2009 International Workshop on Intelligent Systems and Applications, 2009, Wuhan. 1–4
2 Samuel B L, Antonios T, Brian A W, et al. Terrain based co-operative UAV mapping of complex obstacles using 2-D splinegon. Int J Intell Syst Comput Cybern, 2012, 5: 248–292
3 Duan H B, Shao S, Su B W, et al. New Development thoughts on the bio-inspired intelligence based control for unmanned combat aerial vehicle. Sci China Tech Sci, 2010, 53: 2025–2031
4 Duan H B, Li P. Progress in control approaches for hypersonic vehicle. Sci China Tech Sci, 2012, 55: 2965–2970
5 Li G W, Zhang W G, Li J, et al. Particle swarm optimization based on uniform design and its application in flight control systems. Comput Appl, 2004, 27: 727–729

6 Li C Y, Jing W X, Gao C S. Adaptive back-stepping-based flight control system using integral filters. Aerosp Sci Technol, 2009, 13: 105–113
7 Juang J G, Chiou H K, Chien L H. Analysis and comparison of aircraft landing control using recurrent neural networks and genetic algorithms approaches. Neurocomputing, 2008, 71: 3224–3228
8 Oosterom M, Babuška R. Design of a gain-scheduling mechanism for flight control laws by fuzzy clustering. Control Eng Practice, 2006, 14: 769–781
9 Qi G P, Song P, Li K J. Multivariable PID neural network based flight control system for small-scale unmanned helicopter. In: Proceedings of the 2009 IEEE International Conference on Information and Automation, 2009, Macau. 1331–1335
10 Duan H B, Liu S Q, Wang D B, et al. Design and realization of hybrid ACO-based PID and LuGre friction compensation controller for three degree-of-freedom high precision flight simulator. Simul Model Pract Theory, 2009, 17: 1160–1169
11 Sun X, Zhang W G, Yin W, et al. Design of a large envelope wavelet neural network gain scheduling flight control law based on PSO algorithm. Meas Sci Technol, 2007, 26: 34–36
12 Liu F, Duan H B, Deng Y M. A chaotic quantum-behaved particle swarm optimization based on lateral inhibition for image matching. Optik, 2012, 123: 1955–1960
13 Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, Perth, 1995. 1942–1948
14 Duan H B, Ma G J, Luo D L. Optimal formation reconfiguration control of multiple UCAVs using improved particle swarm optimization. J Bionic Eng, 2008, 5: 213–225
15 Jin S H, Kwang Y L, Raul G R. Multiobjective control of power plants using particle swarm optimization techniques. IEEE Trans Energy Convers, 2006, 21: 552–561
16 Mitsuharu H, Atsushi I, Keiichiro Y. Particle swarm optimization considering the concept of predator-prey behavior. In: Proceedings of the IEEE Conference on Evolutionary Computation, 2006, Vancouver. 434–437
17 Di GesùV, Lenzitti B, Lo Bosco G, et al. Comparison of different cooperation strategies in the prey-predator problem. In: Proceedings of the International Workshop on Computer Architecture for Machine Perception and Sensing, 2006, Montreal. 108–112
18 Zhang W Y. Study of Intelligent Robust Design Approach of Large Flight Envelope Flight Control System. Dissertation for the Master Degree. Beijing: Beihang University, 2004.
19 Duan H B. Ant Colony Algorithms: Theory and Applications. Beijing: Science Press, 2005
20 Zhang M T, An J W. Application of intelligent computation to promote the automation of flight control design. In: Proceedings of the International Conference on Intelligent Computation Technology and Automation, 2008, Hunan, Vol. 1: 991–994
21 Li X D, Xian B, Diao C, et al. Output feedback control of hypersonic vehicles based on neural network and high gain observer. Sci China Inf Sci, 2011, 54: 429–447