Chapter 3
UAV Modeling and Controller Design

Haibin Duan

Abstract As a complicated multi-input, multi-output, and time-varying nonlinear system, flight control system of unmanned aerial vehicle (UAV), which determines the whole system's performance directly, is crucial for the simulation training system design. This chapter mainly focuses on parameter identification of flight control system based on the modeling of UAVs and a specific controller design for the pendulum-like oscillation in micro aerial vehicle (MAV). A predator–prey particle swarm optimization (PSO) algorithm for identifying parameters of UAV flight control system is presented, with the aim of reducing the workload of the designers during the process of designing complicated UAV control system. Besides, a software environment for UAV controller design was developed based on the UAV model and the proposed method. Then a specific kind of controller design involving the pendulum-like oscillation in the hover and stare state for a MAV in the presence of external disturbances is discussed in detail, since the pendulum-like oscillation caused by uncertainty and external disturbances badly jeopardize the performance of hover and stare, resulting in blurred images or even MAV's overturning. A novel type of PSO-based linear-quadratic regulator (LQR) controller for stabilizing the pendulum-like oscillation is developed, which can enhance the MAV's performance efficiently.

3.1 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 unmanned aerial vehicle (UAV), which also determines the whole system's performance directly (McLean 1990). 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. Presently, cut and

The original version of this chapter was revised. A correction to this chapter is available at https://doi.org/10.1007/978-3-642-41196-0__9

H. Duan and P. Li, *Bio-inspired Computation in Unmanned Aerial Vehicles*,
DOI 10.1007/978-3-642-41196-0_3, © Springer-Verlag Berlin Heidelberg 2014

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 (Zhang and An 2008). In the first part of this chapter, we proposed a parameter identification method for UAV control system based on predator–prey particle swarm optimization (PSO). While using PSO to optimize flight control systems, there are two problems to be solved. Firstly, the UAV 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 UAV controller. Therefore, how to evaluate each particle's quality turns out to be a key issue which must be resolved.

Micro aerial vehicles (MAVs), essentially small-scale flying robots, became an area of interest in the aerospace community with the initiation of the micro UAV (MUAV) program by Defense Advanced Research Projects Agency (DARPA). The technological feasibility of MAVs as one possible solution to new challenging reconnaissance mission scenarios in urban warfare (local, close-up range, hidden reconnaissance, operation between obstacles and maybe even inside buildings) is depending on a bunch of questions, which are only partly answered so far (Johnson and Turbe 2006; Bloss 2009). One of the challenging problems for MAVs is to design a robust flight control system for such a miniaturized "bird," which is generally one order of magnitude smaller than any today's operational UAV. Hover and stare is a key issue to the performance of MAVs and similar kinds of unmanned vehicles, which are designed to perform surveillance and reconnaissance missions. However, pendulum-like oscillation triggered by external disturbances and other uncertain factors will badly impair its performance, thus resulting in blurred images or even overturn of the vehicle (Pflimlin et al. 2010). As a result, control techniques of such a vehicle are becoming more and more important for their wide applications in civil and military fields. The second part of this chapter designed a novel type of pendulum-like oscillation controller for MAV hover and stare state in the presence of external disturbances, which is based on linear-quadratic regulator (LQR) and PSO (Duan and Sun 2013). A linear mathematical model of pendulum phenomenon based upon actual wind tunnel test data representing the hover mode is established, and a hybrid LQR and PSO approach is proposed to stabilize oscillation. PSO is applied for parameter optimization of the designed LQR controller.

3.2 Parameter Identification for UAVs Based on Predator–Prey PSO

3.2.1 Mathematical Model of UAVs

The 6-DOF nonlinear model of UAVs is illustrated in this section, which is the prerequisite for simplifying and linearizing the mathematical model (Zhang 2004).

3.2.1.1 Nonlinear Equations of 6-DOF Modeling

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

$$\begin{cases} \dot{V} = \dfrac{1}{m}\left(P_x \cos\alpha \cos\beta - P_y \sin\alpha \cos\beta + P_z \sin\beta + Z \sin\beta - Q\right) \\ \quad - g\Big(\cos\alpha \cos\beta \sin\vartheta - \sin\alpha\cos\beta\cos\vartheta \cos\gamma - \sin\beta \cos\vartheta \sin\gamma \\ \dot{\alpha} = -\dfrac{1}{mV\cos\beta}\left(P_x \sin\alpha + P_y\cos\alpha + Y\right) + \omega_z - \tan\beta\left(\omega_x\cos\alpha - \omega_y\sin\alpha\right)\dfrac{\delta y}{\delta x} \\ \quad + \dfrac{g}{V\cos\beta}\left(\sin\alpha\sin\vartheta + \cos\alpha\cos\vartheta\cos\gamma\right) \\ \dot{\beta} = \dfrac{1}{mV}\left(-P_x\cos\alpha\sin\beta + P_y\sin\alpha\sin\beta + P_z\cos\beta + Z\right) + \omega_x\sin\alpha \\ \quad + \omega_y\cos\alpha + \dfrac{g}{V}\left(\cos\alpha\sin\beta\sin\vartheta - \sin\alpha\sin\beta\cos\vartheta\cos\gamma\right. \\ \quad \left. + \cos\beta\sin\gamma\cos\vartheta\right) \end{cases} \tag{3.1}$$

where m is the mass of the UAV; 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; and 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:

$$\begin{cases} \dot{\omega}_x = b_{11}\omega_y\omega_z + b_{12}\omega_x\omega_z + \dfrac{I_y\left(M_x + M_{px}\right) + I_{xy}\left(M_y + M_{py}\right)}{I_xI_y - {I_{xy}}^2} \\ \dot{\omega}_y = b_{21}\omega_y\omega_z + b_{22}\omega_x\omega_z + \dfrac{I_{xy}\left(M_x + M_{px}\right) + I_x\left(M_y + M_{py}\right)}{I_xI_y - {I_{xy}}^2} \\ \dot{\omega}_z = \dfrac{I_x - I_y}{I_z}\omega_x\omega_y + \dfrac{I_{xy}}{I_z}\left({\omega_x}^2 - {\omega_y}^2\right) + \dfrac{\left(M_z + M_{pz}\right)}{I_z} \end{cases} \tag{3.2}$$

$$b_{11} = \frac{{I_y}^2 - I_yI_z - {I_{xy}}^2}{I_xI_y - {I_{xy}}^2}, \quad b_{22} = \frac{I_xI_z - {I_x}^2 - {I_{xy}}^2}{I_xI_y - {I_{xy}}^2},$$

$$b_{12} = \frac{I_{xy}\left(I_z - I_y - I_x\right)}{I_xI_y - {I_{xy}}^2}, \quad b_{21} = \frac{I_{xy}\left(I_y - I_z - I_x\right)}{I_xI_y - {I_{xy}}^2}.$$

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

$$\begin{cases} \dot{\gamma} = \omega_x - \tan\vartheta \left(\omega_y \ \cos\gamma - \omega_z \ \sin\gamma\right) \\ \dot{\vartheta} = \omega_y \ \sin\gamma + \omega_z \ \cos\gamma \\ \dot{\psi} = \frac{1}{\cos\vartheta} \left(\omega_y \ \cos\gamma - \omega_z \ \sin\gamma\right) \end{cases} \tag{3.3}$$

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

$$Y = C_y qS, C_y = C_y(\alpha, \delta_z), Z = \sum C_z qS, \sum C_z = C_z(\alpha, \delta_x) + C_z\left(\alpha, \delta_y\right)$$
$$Q = C_x qS, C_x = C_x(\alpha, \delta_z)$$

where $\delta_x, \delta_y, \delta_z$ are the coordinate components of deflection angles of the controlling surface. Aerodynamic moments can be given by

$$M_x = \sum m_x qsl, \sum m_x = {m_x}^{\beta}\beta + m_x(\alpha, \delta_x) + m_x\left(\alpha, \delta_y\right) + {m_x}^{\omega x}\omega_x \frac{l}{2V} + {m_x}^{\omega y}\omega_y \frac{l}{2V}$$

$$M_y = \sum m_y qsl, \sum m_y = {m_y}^{\beta}\beta + m_y(\alpha, \delta_x) + m_y\left(\alpha, \delta_y\right) + {m_y}^{\omega x}\omega_x \frac{l}{2V} + {m_y}^{\omega y}\omega_y \frac{l}{2V}$$

$$M_z = \sum m_z qsb_A, \sum m_z = m_z(\alpha, \delta_z) + {m_z}^{\omega z}\omega_z \frac{b_A}{V} + {m_z}^{\dot{\alpha}}\dot{\alpha}\frac{b_A}{V}.$$

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 ${m_z}^{\omega z}$, ${m_z}^{\dot{\alpha}}$, ${m_x}^{\beta}$, ${m_x}^{\omega x}$, ${m_x}^{\omega y}$, ${m_y}^{\beta}$, ${m_y}^{\omega x}$, ${m_y}^{\omega y}$ are specified values.

3.2.1.2 Nonlinear Equations of 5-DOF Modeling

Suppose that UAV 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 UAV speed level off (Zhang 2004). At the same time, it is assumed that $I_{xy} \ll I_x I_y - {I_{xy}}^2$, state variable $x = (\alpha,\beta,\omega_x,\omega_y,\omega_z)^T$. The UAV nonlinear equations of 5-DOF are represented as follows:

$$
\begin{cases}
\dot{\alpha} = -\dfrac{1}{mV\ \cos\ \beta}(P\ \sin\ \alpha + Y) + \omega_z - \tan\ \beta\left(\omega_x\ \cos\ \alpha - \omega_y\ \sin\ \alpha\right) \\
\quad + \dfrac{g}{V\ \cos\ \beta}(\sin\ \alpha\ \sin\ \vartheta + \cos\ \alpha\ \cos\ \vartheta\ \cos\ \gamma) \\
\dot{\beta} = \dfrac{1}{mV}(-P\ \cos\ \alpha\ \sin\ \beta + Z) + \omega_x\ \sin\ \alpha \\
\quad + \omega_y\ \cos\ \alpha + \dfrac{g}{V}(\cos\ \alpha\ \sin\ \beta\ \sin\ \vartheta - \sin\ \alpha\ \sin\ \beta\ \cos\ \vartheta\ \cos\ \gamma \\
\quad + \cos\ \beta\ \sin\ \gamma\ \cos\ \vartheta) \\
\dot{\omega}_x = \dfrac{{I_y}^2 - I_y I_z - {I_{xy}}^2}{I_x I_y - {I_{xy}}^2}\omega_y\omega_z + \dfrac{I_{xy}\left(I_z - I_y - I_x\right)}{I_x I_y - {I_{xy}}^2}\omega_x\omega_z + \dfrac{I_y\sum M_x}{I_x I_y - {I_{xy}}^2} \\
\quad b_{11}\omega_y\omega_z + b_{12}\omega_x\omega_z + \dfrac{I_y M_x}{I_x I_y - {I_{xy}}^2} \\
\dot{\omega}_y = \dfrac{I_{xy}\left(I_y - I_z - I_x\right)}{I_x I_y - {I_{xy}}^2}\omega_y\omega_z + \dfrac{I_x I_z - {I_x}^2 - {I_{xy}}^2}{I_x I_y - {I_{xy}}^2}\omega_x\omega_z + \dfrac{I_x\sum M_y}{I_x I_y - {I_{xy}}^2} \\
\quad b_{21}\omega_y\omega_z + b_{22}\omega_x\omega_z + \dfrac{I_x M_y}{I_x I_y - {I_{xy}}^2} \\
\dot{\omega}_z = \dfrac{I_x - I_y}{I_z}\omega_x\omega_y + \dfrac{I_{xy}}{I_z}\left({\omega_x}^2 - {\omega_y}^2\right) + \dfrac{\sum M_z}{I_z}\dfrac{I_x - I_y}{I_z}\omega_x\omega_y \\
\quad + \dfrac{I_{xy}}{I_z}\left({\omega_x}^2 - {\omega_y}^2\right) + \dfrac{M_z}{I_z}
\end{cases}
\tag{3.4}
$$

3.2.1.3 Linearization Modeling

In most cases, the UAV maintains steady straight level flight, and (3.4) can be modeled as linear time invariant state-space perturbation models, with the nominal trajectory being steady-level trimmed flight. The UAV's linear equations are as follows:

$$
\begin{cases}
\Delta\dot{\alpha} = \Delta\omega_z - (Y^{\alpha}\Delta\alpha + \Delta\delta_Z) \\
\Delta\dot{\omega}_z = {M_z}^{\alpha}\Delta\alpha + {M_z}^{\omega_z}\Delta\omega_z + {M_z}^{\delta_z}\Delta\delta_z \\
\Delta\dot{\beta} = \Delta\omega_y + Z^{\delta_x}\Delta\delta_x + Z^{\delta_y}\Delta\delta_y \\
\Delta\dot{\omega}_x = {M_x}^{\beta}\Delta\beta + {M_x}^{\omega_x}\Delta\omega_x + {M_x}^{\omega_y}\Delta\omega_y + {M_x}^{\delta_x}\Delta\delta_x + {M_x}^{\delta_y}\Delta\delta_y \\
\Delta\dot{\omega}_y = {M_y}^{\beta}\Delta\beta + {M_y}^{\omega_x}\Delta\omega_x + {M_y}^{\omega_y}\Delta\omega_y + {M_y}^{\delta_x}\Delta\delta_x + {M_y}^{\delta_y}\Delta\delta_y
\end{cases}
\tag{3.5}
$$

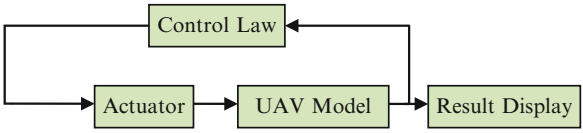

Fig. 3.1 UAV system model (Reprinted from Duan et al. (2013a), with kind permission from Springer Science+Business Media)

We have the state equations.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{3.6}$$

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

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

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

3.2.2 Predator–Prey PSO for Parameter Identification

3.2.2.1 UAV Flight Control System Design in Matlab Environment

Based on control augmentation system of UAVs, the aircraft linear equations are generally obtained by a series of equilibrium points. The flight envelope of this UAV must satisfy $0 \leq H \leq 18km$ and $0.6 \leq M \leq 2.2$ (Zhang 2004). Figure 3.1 shows the schematic diagram of UAV system.

As is obvious in Fig. 3.1, the UAV system is comprised of four subsystems: control law, actuator, mathematical model, and simulation result display.

Matrix 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 Fig. 3.2.

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

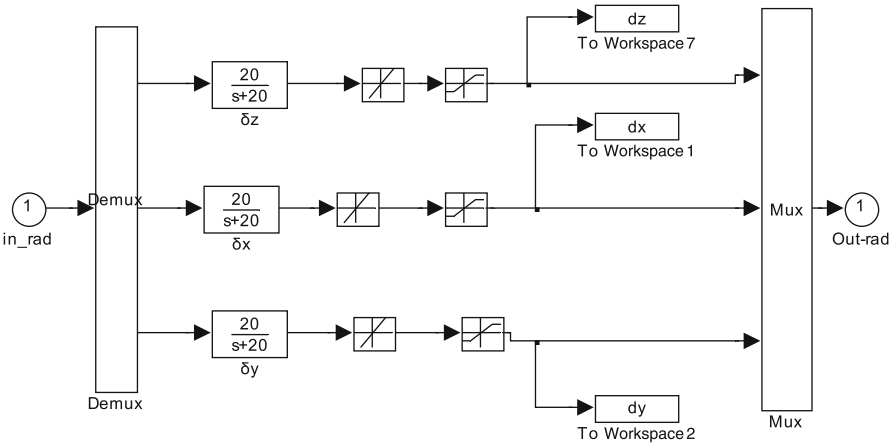

Fig. 3.2 Actuator module of UAVs in Matlab environment (Reprinted from Duan et al. (2013a), with kind permission from Springer Science+Business Media)

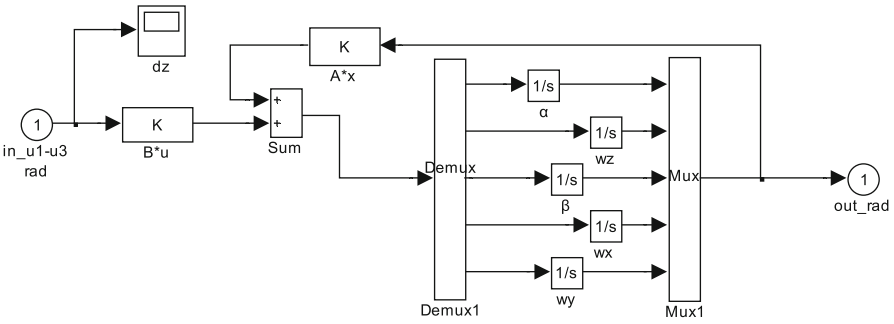

Fig. 3.3 UAV model module in Matlab environment (Reprinted from Duan et al. (2013a), with kind permission from Springer Science+Business Media)

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 $80°/s$. UAV model module is displayed in Fig. 3.3. where matrix A and B are both obtained from the workspace of Matlab. Figure 3.4 shows the simulation result display module.

The control law $u = -Kx$ is used. K denotes the state-feedback gain, and it can be illustrated with the following matrix:

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

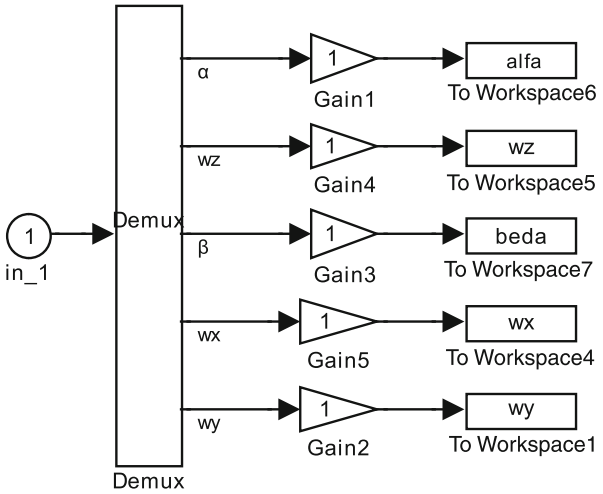

Fig. 3.4 Simulation result display module in Matlab environment (Reprinted from Duan et al. (2013a), with kind permission from Springer Science+Business Media)

3.2.2.2 Predator–Prey PSO for Identifying Controller Parameters

In the Gbest model of PSO, each particle has its current position and velocity in a space of solution. 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 (Shi and Eberhart 1998a)

$$v_{ij}(k+1) = \omega v_{ij}(k) + c_1 r_1 \left[p_i(k) - x_{ij}(k)\right] + c_2 r_2 \left[g_i(k) - x_{ij}(k)\right] \quad (3.7)$$

$$x_{ij}(k+1) = x_{ij}(k) + v_{ij}(k+1) \quad (3.8)$$

where $v_i(k)$ and $x_i(k)$ respectively denote the velocity and position of the i th 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, p_i is the best position of the i th particle, and 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. Predator–prey PSO is a method which takes a cue from the behavior of schools of sardines and pods of killer whales (Higashitani et al. 2006; Wang and Duan 2013). 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 (Duan et al. 2011), and preys escape from predators in multidimensional solution space. After taking a trade-off 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 (Higashitani et al. 2006)

$$v_{dij}(k+1) = \omega_d v_{dij}(k) + c_1 r_1 \left[p_{dij}(k) - x_{dij}(k)\right] + c_2 r_2 \left[g_{dj}(k) - x_{dij}(k)\right] + c_3 r_3 \left[g_j(k) - x_{dij}(k)\right] \tag{3.9}$$

$$\begin{aligned} v_{rij}(k+1) = {} & \omega_r v_{rij}(k) + c_4 r_4 \left[p_{rij}(k) - x_{rij}(k)\right] \\ & + c_5 r_5 \left[g_{rj}(k) - x_{rij}(k)\right] + c_6 r_6 \left[g_j(k) - x_{rij}(k)\right] \\ & - Pasign\left[x_{dIj}(k) - x_{rij}(k)\right] \exp\left[-b\left|x_{dij}(k) - x_{rij}(k)\right|\right] \end{aligned} \tag{3.10}$$

where d and r denote predator and prey, respectively, p_{di} is the best position of predators, g_d is the best position which predators have ever found, p_{ri} is the best position of preys, g_r is the best position which preys have ever found, g is the best position which all the particles have ever found, and ω_d and ω_r can be defined as

$$\omega_d = 0.2 \exp\left(-10 \frac{iteration}{iteration_{\max}}\right) + 0.4 \tag{3.11}$$

$$\omega_r = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{iteration_{\max}} iteration \tag{3.12}$$

PSO can be also improved by a modification of the inertia weight ω_r in (3.12). 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 (3.12) $iteration_{\max}$ is maximum iteration, $\omega_{\max}$ and $\omega_{\min}$ are, respectively, maximum and minimum of ω_r. In our experiment, $\omega_{\max}$ and $\omega_{\min}$ are 0.9 and 0.2, respectively. And the definition of I is given by

$$I = \left\{ k \left| \min_k \left(|x_{dk} - x_{ri}|\right) \right. \right\} \tag{3.13}$$

Then I denotes the number of the ith prey's nearest predator. In (3.10), 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

$$a = x_{span}, \qquad b = \frac{100}{x_{span}} \tag{3.14}$$

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

The parameter identification of the conventional flight controller can be treated as the typical continual spatial optimization problem. 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.

Using the proposed predator–prey PSO algorithm to obtain the optimal parameter combination for the UAV flight control system here, the fitness function is given by

$$J = \frac{1}{2}\int \left(x' Q x + u' R u\right) dt \tag{3.15}$$

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

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

$$x_d = \left(k_1\ k_2\ k_3\ k_4\ k_5\ k_6\ k_7\ k_8\right)$$
$$x_r = \left(k_1\ k_2\ k_3\ k_4\ k_5\ k_6\ k_7\ k_8\right)$$

where x_d and x_r have the constraint of ± 10, which is set according to exact experience.

The process of proposed predator–prey PSO algorithm for solving UAV controller parameter identification can be described with Fig. 3.5 (Duan and Sun 2013).

The above mentioned flow chart of the predator–prey PSO algorithm process can also be illustrated with Fig. 3.6.

The complexity of predator–prey PSO algorithm can be computed, and Table 3.1 shows a comparison of the complexity analysis between basic PSO and predator–prey PSO.

In Table 3.1, $m = m_d + m_r$, and n is the dimension of particle's position. The total complexity of basic PSO and predator–prey PSO can be expressed as

$$T(n)_{basic\,PSO} = O\left(14 N_{\max}\, mn\right) \tag{3.16}$$

$$T(n)_{improved\,PSO} = O\left(N_{\max}\, m_r\, n^2\right) \tag{3.17}$$

3.2.3 *Experiments*

In order to investigate the feasibility and effectiveness of the proposed predator–prey PSO approach for identification of UAV controller parameters, a series of experiments are conducted under some constrained conditions.

PROCEDURE Optimization of UAV 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 x_d and v_d respectively, both of which have the same dimensions m_d by 8. So are x_r and 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 (3.8)-(3.10). Then repeat (2) in **Step 2**.

Step 4: $N \geq N_{\max}$?

(1)Yes: stop and output results;

(2)No: go to **Step 3**.

End

Fig. 3.5 The pseudocode of predator–prey PSO algorithm for UAV flight controller (Reprinted from Duan et al. (2013a), with kind permission from Springer Science+Business Media)

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 $\alpha = 10°, \beta = 10°$, $N_{\max} = 100$, $t = 20s$, mach=0.8, $H = 8000m$, $m_d = 10$, $m_r = 20$ 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 K is illustrated from Fig. 3.7 (a–e). Figure 3.7f shows the evolution curve of the proposed PSO.

The final optimal result is 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 $bestN$ = 100. As indicated in Fig. 3.7, 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

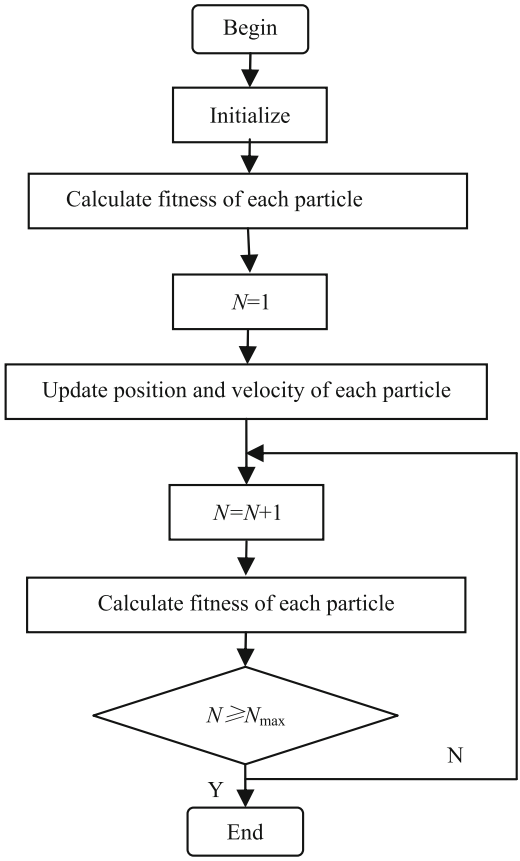

Fig. 3.6 Flow chart of the predator–prey PSO (Reprinted from Duan et al. (2013a), with kind permission from Springer Science+Business Media)

Table 3.1 Comparison of complexity analysis between basic PSO and predator–prey PSO

		Complexity	
Step	Operation	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	$O(1)$	$O(1)$

than LQR. Figure 3.7f also demonstrates that the algorithm can converge to the optimal solution quickly.

Case 2: In this case, the parameter values are $\alpha = 10°, \beta = 10°$, $N_{\max} = 100$, $t = 20s$, $mach = 1.2$, $H = 15000m$, $m_d = 10$, $m_r = 20$ 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 Fig. 3.8a–e. Figure 3.8f shows the evolution curve of the proposed PSO.

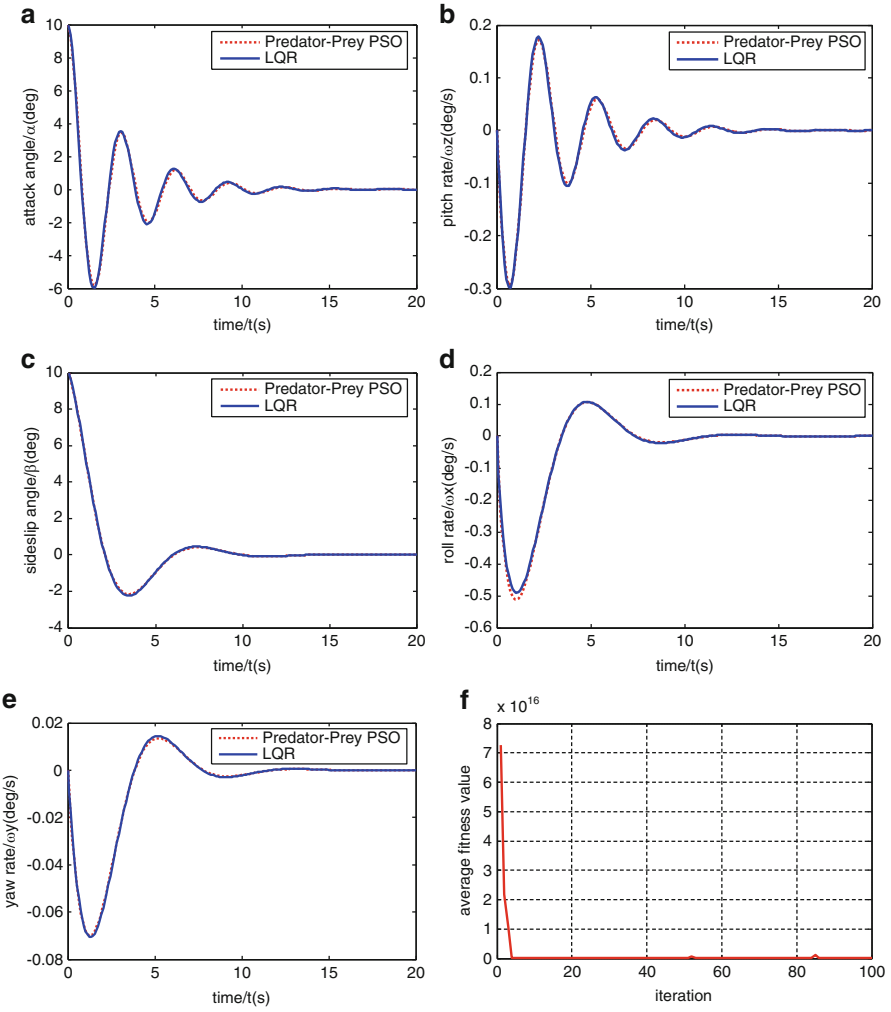

Fig. 3.7 Results of identifying controller parameters for UAVs based on predator–prey PSO in Case 1 (**a**) Comparison of attack angle responses. (**b**) Comparison of pitch rate responses. (**c**) Comparison of sideslip angle responses. (**d**) Comparison of roll rate responses. (**e**) Comparison of yaw rate responses. (**f**) Evolution curve of predator–prey PSO (Reprinted from Duan et al. (2013a), with kind permission from Springer Science+Business Media)

The final optimal results are $K = [0.0416, 0.8298, -2.7208, 1.4825, -1.0683, 1.9002, -0.2273, 2.6742]$, the minimum fitness value $J_{\min} = 68.3361$, and the best iteration $bestN = 99$. Although the initial conditions are much different from those of experiment 1, the improved algorithm can find the optimal solution.

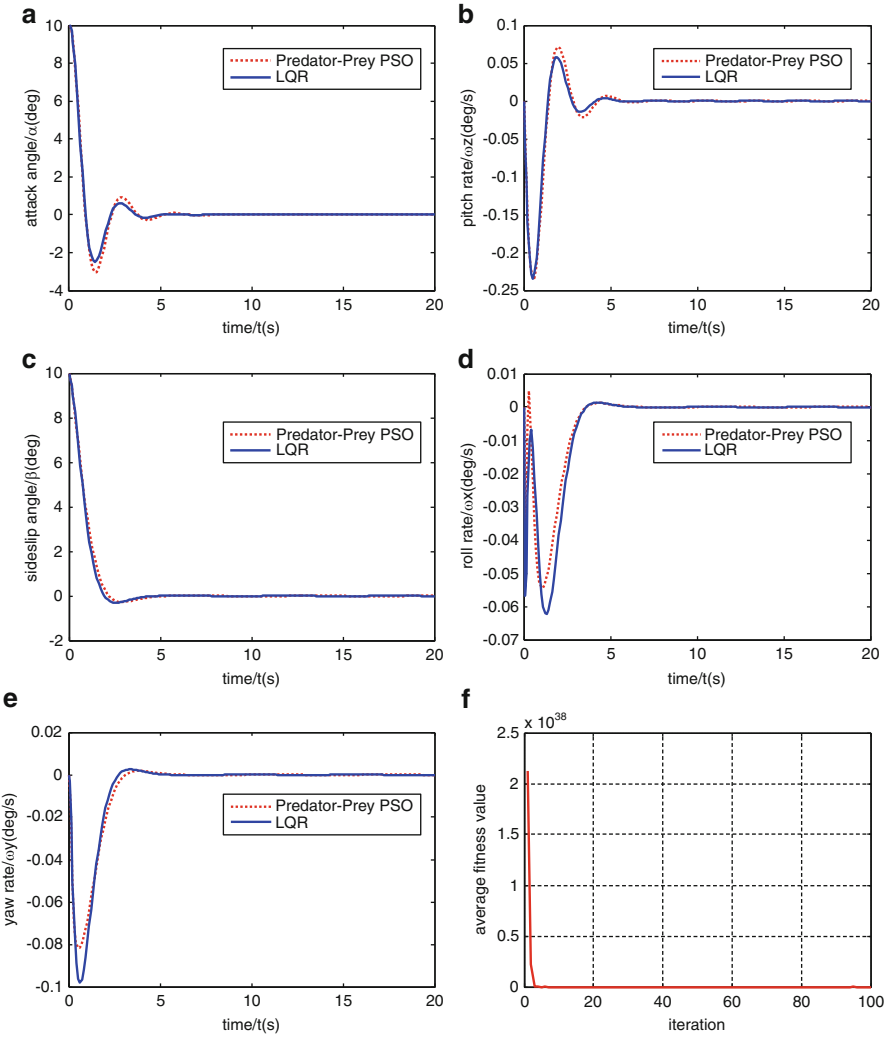

Fig. 3.8 Results of identifying controller parameters for UAVs based on predator–prey PSO in Case 2 (**a**) Comparison of attack angle responses. (**b**) Comparison of pitch rate responses. (**c**) Comparison of sideslip angle responses. (**d**) Comparison of roll rate responses. (**e**) Comparison of yaw rate responses. (**f**) Evolution curve of predator–prey PSO (Reprinted from Duan et al. (2013a), with kind permission from Springer Science+Business Media)

From these two experimental results, it is obvious that the proposed predator–prey PSO approach could make the UAV 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 UAV model and the proposed predator–prey PSO algorithm, we developed a software platform of UAV controller design. The graphical user interfaces (GUI) of this platform are shown in Fig. 3.9.

3.3 PSO Optimized Controller for Unmanned Rotorcraft Pendulum

MAV offer several advantages as autonomous UAVs (Pflimlin et al. 2010; Sun and Duan 2013; Duan and Sun 2013). They can be very small with a compact layout. Many are capable of high-speed flight in addition to the normal hover and vertical take-off and landing capabilities. These features make them well suited for a variety of missions, especially in urban environments. A recent announcement by the US Government of plans to greatly increase the number of unmanned aircraft on station in Iraq and Afghanistan helps drive the surge in interest for unmanned vehicles. US Defense Secretary Robert Gates (Bloss 2009) commented that unmanned aircraft is essential to provide real-time video of insurgent activity and argued that the need is growing at 300 % per year. Nowadays, many companies are functioning towards research and development of MAVs, and the representative achievements are Cypher series by Sikorsky and MAVs by Honeywell etc.

Hover and stare is of the paramount importance to the performance of MAVs, as they are designed to perform surveillance and reconnaissance missions. However, pendulum-like oscillation (see Fig. 3.10) triggered by external disturbances and other uncertain factors will badly impair its performance, resulting in blurred images or even MAVs' overturning. As a result, control techniques of MAVs are becoming more and more important for their wide applications in civil and military fields, with special regard to the hover and stare state for better performances of surveillance and reconnaissance missions. This section focuses on a particular kind of MAV, which is driven by a rotor and a ducted fan, and proposes a combination control law design approach to stabilize the hover and stare pendulum-like oscillation based on LQR, in which an improved PSO algorithm is utilized for parameter optimization of matrix Q and R in the linear-quadratic regulator. In this way, the MAV's dynamic properties can be ameliorated efficiently while executing surveillance missions that requires perfect stability and rapid responses.

3.3.1 Mathematical Model of Pendulum Oscillation for MAVs

The MAV in this section adopts the axial symmetrical layout. Owing to the fact that the suspension center moves freely with rotor wings in the plane, pendulum-like oscillation is nonlinear, strongly coupled, and of high order (Pflimlin et al. 2010). In this section, a mathematical model representing the pendulum-like oscillation

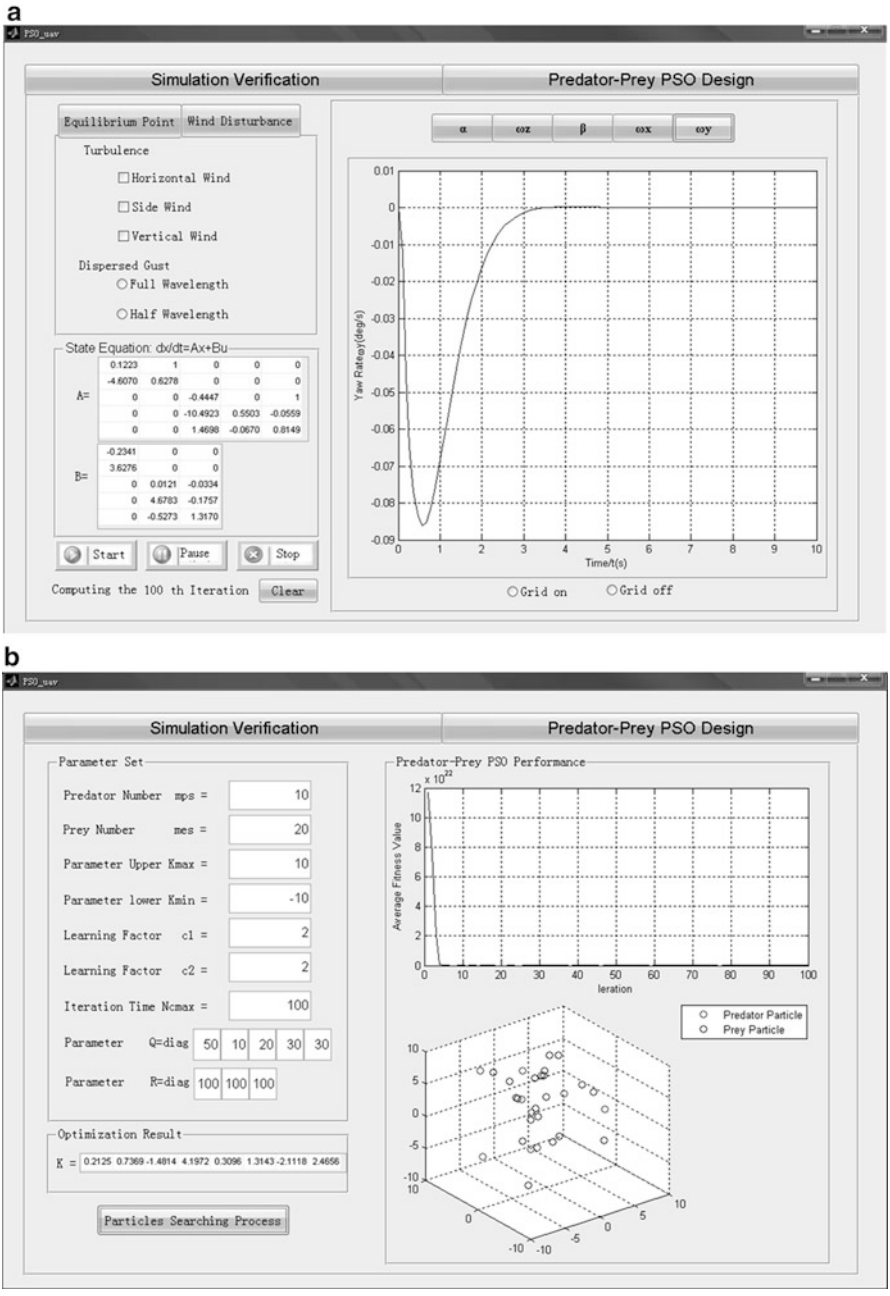

Fig. 3.9 UAV controller design software platform based on predator–prey PSO. (**a**) Yaw rate response interface. (**b**) Predator–prey PSO convergence process interface (Reprinted from Duan et al. (2013a), with kind permission from Springer Science+Business Media)

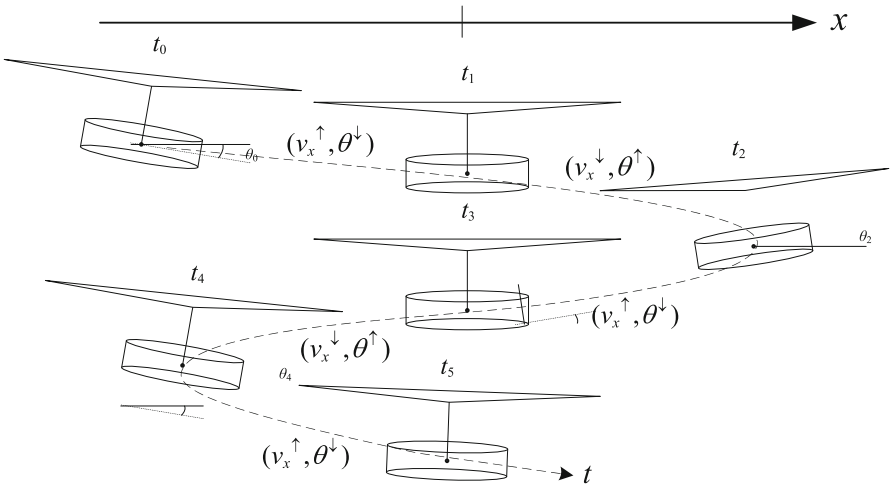

Fig. 3.10 Pendulum-like oscillation in actual flight (x-axis) (Reprinted from Duan and Sun (2013), with kind permission from Springer Science+Business Media)

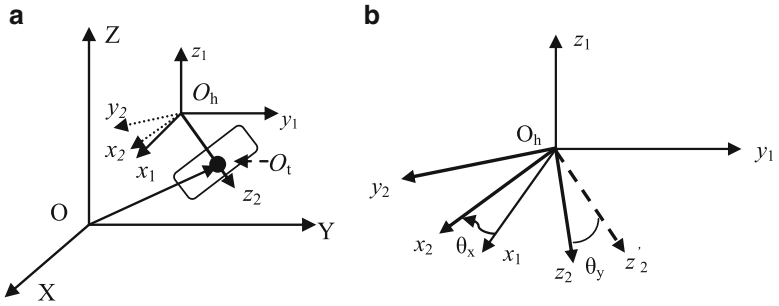

Fig. 3.11 Coordinates of the pendulum model and decomposition of oscillation angle. (**a**) Coordinates of the pendulum model. (**b**) Decomposition of oscillation angle (Reprinted from Duan and Sun (2013), with kind permission from Springer Science+Business Media)

in the hover and stare state is obtained by using the Lagrangian method, and the corresponding linearized model is obtained in the neighborhood of the hover and stare equilibrium.

3.3.1.1 Nonlinear Mathematical Model of MAV Pendulum-Like Oscillation

The pendulum can be abstracted as a system consisting of the suspension point O_h, the pendulum rod O_hO_t, and the pendulum mass O_t (see Fig. 3.11).

In Fig. 3.11, $OXYZ$ is the ground-fixed coordinate system, $O_h x_1 y_1 z_1$ represents a mobile ground coordinate system, with the point O_h as the origin and parallel to $OXYZ$, while $O_h x_2 y_2 z_2$ is defined as the pendulum-body coordinate frame, which originates from O_h, and axis z_2 points downward along the pendulum rod $O_h O_t$. Decompose the two-dimensional pendulum-like oscillation to one in the direction of axis X and Y, and the transition matrix R from $O_h x_2 y_2 z_2$ to $O_h x_1 y_1 z_1$ is obtained as follows:

$$R = \begin{bmatrix} \cos\theta_y & \sin\theta_x \sin\theta_y & \cos\theta_x \sin\theta_y \\ 0 & \cos\theta_x & -\sin\theta_x \\ -\sin\theta_y & \sin\theta_x \cos\theta_y & \cos\theta_x \cos\theta_y \end{bmatrix} \tag{3.18}$$

Suppose the position vector of the suspension point O_h in the coordinate frame of $OXYZ$ and the pendulum mass O_h in the coordinate frame of $O_h x_2 y_2 z_2$ are presented with $r_h = [x,y,z]^T$, $r'_t = [0,0,l]^T$, respectively. Then, the coordinate of point O_t in $O_h x_1 y_1 z_1$ can be calculated according to the following equation:

$$r_{ht} = R \cdot r'_t = \left[l \cos\theta_x \sin\theta_y, -l \sin\theta_x, l \cos\theta_x \cos\theta_y \right]^T \tag{3.19}$$

where l denotes the distance from O_h to O_t, i.e., the length of the pendulum.

Finally, we have the coordinate of O_t in coordinate frame $OXYZ$ presented as in (3.20):

$$r_t = r_h + R \cdot r' \tag{3.20}$$

The Lagrange function of the pendulum system:

$$\begin{aligned} L &= T_0 - V = T_m + T'_M + T''_M - V \\ &= \frac{1}{2} m \dot{r}_h^T \dot{r}_h + \frac{1}{2} m_t V_t^T V + \frac{1}{2} J_{x2} \omega_{x2}^2 + \frac{1}{2} J_{y2} \omega_{y2}^2 \\ &\quad - \left(-m_t g l \cos\theta_x \cos\theta_y + (m + m_t)(z - z_0) g \right) \\ &= \frac{1}{2} (m + m_t) \left(\dot{x}^2 + \dot{y}^2 + \dot{z}^2 \right) + \frac{1}{6} m_t l \left[\left(3 + \cos^2\theta_y \right) l \dot{\theta}_x^2 + \left(1 + 3\cos^2\theta_x \right) l \dot{\theta}_y^2 \right] \\ &\quad + m_t l \left(\dot{x} \dot{\theta}_y \cos\theta_x \cos\theta_y - \dot{x} \dot{\theta}_x \sin\theta_x \sin\theta_y - \dot{y} \dot{\theta}_x \cos\theta_x - \dot{z} \dot{\theta}_x \sin\theta_x \cos\theta_y \right. \\ &\quad \left. - \dot{z} \dot{\theta}_y \cos\theta_x \sin\theta_y \right) + m_t g l \cos\theta_x \cos\theta_y - (m + m_t)(z - z_0) g \end{aligned} \tag{3.21}$$

where T_m, T'_M, T''_M represent, respectively, kinetic energy of the suspension center O_h, translation kinetic energy of the pendulum mass O_t, and rotational kinetic energy of the pendulum rod around the centroid of O_t and V is potential energy of the system, choosing the initial state of the pendulum-like oscillation as the zero-potential energy surface.

Table 3.2 Main parameters of the MAV system structure

Symbol	Value	Physical meaning
m	10 kg	Suspension center quality
m_t	20 kg	Pendulum quality
l	0.86 m	Pendulum length
g	9.8 m/s^2	Gravity acceleration
θ_x, θ_y	–	Pendulum angle around axis x, y
ϕ_x, ϕ_y	–	Pitch and roll angle
a_x, a_y	–	Suspension center acceleration
u_x, u_y	–	Control force on O_h

Therefore, from the Lagrange equation it can be deduced:

$$\begin{cases} \dfrac{d}{dt}\left(\dfrac{\partial L}{\partial \dot{\theta}_x}\right) - \dfrac{\partial L}{\partial \theta_x} = 0 \\ \dfrac{d}{dt}\left(\dfrac{\partial L}{\partial \dot{\theta}_y}\right) - \dfrac{\partial L}{\partial \theta_y} = 0 \end{cases} \tag{3.22}$$

Considering all the above equations (3.18, 3.19, 3.20, 3.21, and3.22), the nonlinear mathematical model of the MAV pendulum-like oscillation is finally obtained as follows:

$$\begin{cases} \ddot{\theta}_x = \Big(3\ddot{x}\sin\theta_x\sin\theta_y + 3\ddot{y}\cos\theta_x + 3\ddot{z}\sin\theta_x\cos\theta_y + 2l\dot{\theta}_x\dot{\theta}_y\cos\theta_x\sin\theta_y \\ \qquad - 3l\dot{\theta}_y^2\sin\theta_x\cos\theta_x - 3g\sin\theta_x\cos\theta_x\Big)/\left(\cos^2\theta_y + 3\right)l \\ \ddot{\theta}_y = \Big(-3\ddot{x}\cos\theta_x\cos\theta_y + 3\ddot{z}\cos\theta_x\sin\theta_y + 6l\dot{\theta}_x\dot{\theta}_y\sin\theta_x\cos\theta_x \\ \qquad - l\dot{\theta}_x^2\sin\theta_y\cos\theta_y - 3g\cos\theta_x\sin\theta_y\Big)/\left(\cos^2\theta_x + 3\right)l \end{cases} \tag{3.23}$$

Table 3.2 gives the main parameters of the MAV system structure in this section.

3.3.1.2 Linearization of Mathematical Model

Step 1: Change the pendulum angle around axis x, y (θ_x,θ_y) to the pitch and roll angle (ϕ_x, ϕ_y) according to $\phi_x = \theta_y$, $\phi_y = -\theta_x$, and then (3.23) can be described as

$$\begin{cases} \ddot{\phi}_y = \Big(3\ddot{x}\sin\phi_x\sin\phi_y - 3\ddot{y}\cos\phi_y + 3\ddot{z}\sin\phi_y\cos\phi_x + 2l\dot{\phi}_x\dot{\phi}_y\cos\phi_y\sin\phi_x \\ \qquad - 3l\dot{\phi}_x^2\sin\phi_y\cos\phi_y - 3g\sin\phi_y\cos\phi_y\Big)\Big/\left(\cos^2\phi_x + 3\right)l \\ \ddot{\phi}_x = \Big(-3\ddot{x}\cos\phi_y\cos\phi_x + 3\ddot{z}\cos\phi_y\sin\phi_x + 6l\dot{\phi}_x\dot{\phi}_y\sin\phi_y\cos\phi_y \\ \qquad - l\dot{\phi}_y^2\sin\phi_x\cos\phi_x - 3g\cos\phi_y\sin\phi_x\Big)\Big/\left(\cos^2\phi_y + 3\right)l \end{cases} \tag{3.24}$$

Step 2: Choose the state variables $X_e = \left[x, \phi_x, \dot{x}, \dot{\phi}_x, y, \phi_y, \dot{y}, \dot{\phi}_y, z, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}\right]^T$, the input of the system $u = [\ddot{x}, \ddot{y}]^T$. Expand (3.24) into Taylor series in the vicinity of the equilibrium point, and the linear model of the pendulum-like oscillation is finally obtained as follows:

$$\begin{bmatrix} \dot{X}_x \\ \dot{X}_y \end{bmatrix} = \begin{bmatrix} A_x & 0 \\ 0 & A_y \end{bmatrix} \begin{bmatrix} X_x \\ X_y \end{bmatrix} + \begin{bmatrix} B_x & 0 \\ 0 & B_y \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{3.25}$$

where the state vector $X_x = \left[x, \phi_x, \dot{x}, \dot{\phi}_x\right]^T$, $X_y = \left[y, \phi_y, \dot{y}, \dot{\phi}_y\right]^T$, and

$$A_x = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & -\frac{3g}{4l} & 0 & 0 \end{bmatrix}, A_y = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & -\frac{3g}{4l} & 0 & 0 \end{bmatrix}, B_x = \begin{bmatrix} 0 \\ 0 \\ 1 \\ -\frac{3}{4l} \end{bmatrix}, B_y = \begin{bmatrix} 0 \\ 0 \\ 1 \\ -\frac{3}{4l} \end{bmatrix}.$$

3.3.2 Oscillation Controller Design Based on LQR and PSO

3.3.2.1 Characteristics of Pendulum-Like Oscillation

From the linear model, the MAV pendulum-like oscillation in the direction of X and Y is no longer coupled with each other. The pendulum system can be reduced to two four-order subsystems, which are independent of one another. Due to similarity between matrix A_x and A_y, we choose either of the two subsystems to analyze characteristics of the MAV pendulum-like oscillation.

Considering pendulum motion only in the direction of axis X, the state-space equation of the linearized X subsystem is presented as follows:

$$\begin{cases} \dot{X} = AX + BU \\ Y = CX \end{cases} \tag{3.26}$$

where $X = [x_1, x_2, x_3, x_4]^T = \left[x, \phi_x, \dot{x}, \dot{\phi}_x\right]^T$, $A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -0.6500 & 0.6500 \\ 0 & -8.547 & 0.5669 & -0.3924 \end{bmatrix}$,

$B = \begin{bmatrix} 0 \\ 0 \\ 1 \\ -0.872 \end{bmatrix}$, $C = \begin{bmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \end{bmatrix}$.

In the actual flight, external disturbances such as crosswinds jeopardize stability of hover and stare state and lead to a considerate degradation of the surveillance performance and result in even false intelligence information. Assume there is a

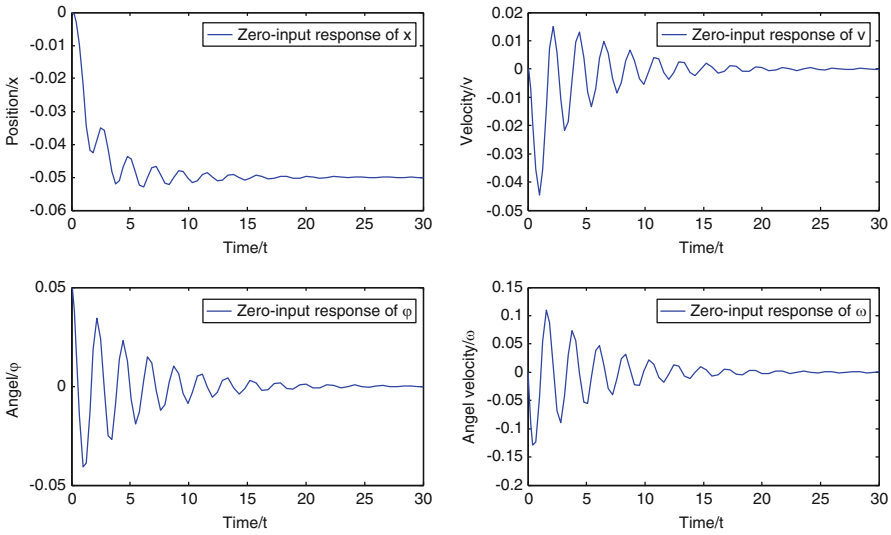

Fig. 3.12 Responses of the MAV pendulum-like oscillation system (Reprinted from Duan and Sun (2013), with kind permission from Springer Science+Business Media)

sudden gust, the pendulum system gets an initial state $\phi_x(0) = 0.05$rad. The inherent characteristics of the MAV pendulum-like oscillation system and the responses are given in Fig. 3.12.

As shown in Fig. 3.12, due to external disturbances of crosswinds, which result in an initial state $\phi_x(0) = 0.05$rad, the pendulum-like oscillation occurs in MAV that is required to be stable enough to carry out hover and stare missions. However, the actual fact shown by the analysis results states clearly that the position of the suspension, denoted by x in the Fig. 3.12, does not remain in the original place but moves to another site in 20 s, which may bring about deviation from the ideal monitoring precision. Furthermore, the MAV's body swings back and forth just in the way as a pendulum does and eventually converges to the equilibrium point after 25 s.

3.3.2.2 Control Law Design Based on LQR

As mentioned above, the hover and stare state is inherently unstable, and external disturbances would give rise to pendulum-like oscillation depicted as in Fig. 3.12. A novel type of optimal control law based on LQR and PSO is designed in this section, which is used to eliminate pendulum-like oscillation with the shortest duration. In this controller, the position of the suspension point x and the pendulum angle ϕ_x are the two main state variables to be controlled to the desired value. In order to ensure the robustness and optimality of the close loop, the LQR design technique is applied due to the fact that it has a very nice robustness property

and has been widely used in many applications. The key issue of LQR controller design is how to select an appropriate control vector $u(t)$ so that the given quadratic performance index (see (3.27)) obtains the minimum value. It is proved that the performance index (10) can reach its minimum by the designed linear control law in (3.28).

$$J = \int_0^\infty \left(X^T QX + u^T Ru\right) dt \tag{3.27}$$

$$u(t) = -KX(t) = R^{-1}B^T PX(t) \tag{3.28}$$

and the optimal matrix P can be calculated from the following Algebraic Riccati Equation:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{3.29}$$

Taking all factors into account, including the performance of the control system and restrictions on the total energy consumed, matrix Q and R can be defined in the form of $Q = \text{diag}(q_{11},q_{22},0,0)$, $R = 1$, in which parameters q_{11} and q_{22} are crucial for a splendid dynamic response. As a result, the proposed algorithm takes advantage of PSO's high operating efficiency, fast convergence speed, and model simplicity, which is used to search the appropriate parameter setting of the LQR control law design approach.

Let the input u be the control force acting on the suspension center, and the corresponding pendulum-holding back control law from the LQR is described as follows:

$$u = -KX = -\left(k_1x_1 + k_2x_2 + k_3x_3 + k_4x_4\right) \tag{3.30}$$

where K denotes the feedback parameters obtained for the LQR and X denotes the states of the system, i.e., the position of the suspension center, the angle of the pendulum, the speed of the suspension center, and the angular velocity of the pendulum, respectively.

The PSO-based LQR controller for prohibiting MAV pendulum-like oscillation of the hover and stare state in presence of external disturbances can be illustrated with Fig. 3.13.

3.3.2.3 Key Settings for PSO

1. *Fitness Function f*

   The fitness function f chosen in LQR controller can be described as follows:

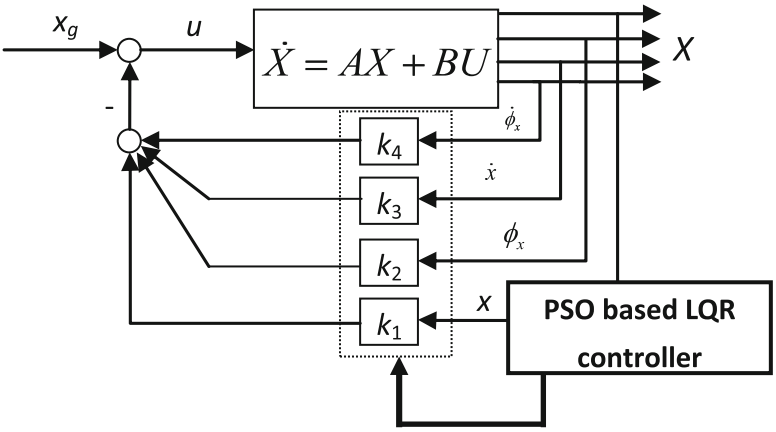

Fig. 3.13 Structure of pendulum-like oscillation control system (Reprinted from Duan and Sun (2013), with kind permission from Springer Science+Business Media)

$$f = \frac{1}{\int_0^{\infty} \left(X^T Q X + u^T R u\right) dt} \tag{3.31}$$

where $u = -(k_1x_1 + k_2x_2 + k_3x_3 + k_4x_4)$, $K = [k_1,k_2,k_3,k_4] = lqr(A,B,Q,R)$, $Q = \text{diag}(q_{11},q_{22},0,0)$.

2. *Inertia Weight* ω
   The inertia weight ω controls the exploration properties of the algorithm, with larger values facilitating a more global behavior and smaller values facilitating a more local behavior; thus results of the algorithm depend largely on ω selection (Duan and Liu 2010; Liu et al. 2012; Duan and Sun 2013). Generally, there are two ways to choose the inertia weight, namely, constant ω and time-variant ω.Shi suggested using $0.8 < \omega < 1.4$, which starts with bigger ω values (a more global search behavior) that is dynamically reduced (a more local search behavior) during the optimization. In this section, ω can be declined linearly from 1.4 to 0.8 in the former 75 % phylogenetic scale and keep constant in the rest time.
3. *Population Size m*
   According to the scale of the exact optimization problem, m is set between 40 and 150. Here we choose $m = 100$.
4. *Acceleration Constants* c_1 *and* c_2
   Shi and Eberhart (1998b) suggests $c_1 = c_2 = 2$. Related work showed that having each particle put slightly more trust in the swarm (larger c_2 value) and slightly less trust in itself (smaller c_1 value), which seems to act better for the structural design problems. According to experiences, we choose $c_1 = 1.8$ and $c_2 = 1.3$.

Table 3.3 Control parameters of the PSO-based LQR approach

Parameter	Optimized value
Matrix Q	Diag(245.6,250.3 0,0)
Feedback vector K	[15.6709, −17.1806, 8.6616, 2.2921]

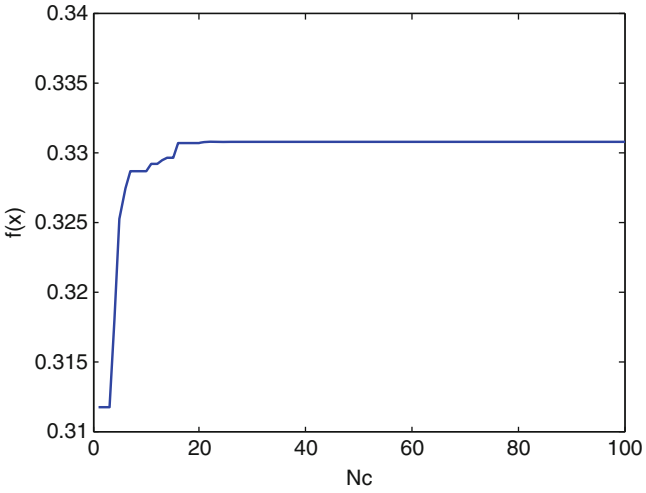

Fig. 3.14 The average evolution curves of the PSO (Reprinted from Duan and Sun (2013), with kind permission from Springer Science+Business Media)

3.3.3 Experiments

Assume the designed MAV executes a surveillance mission using the hover and stare mode in the actual flight. Considering external disturbances, we take crosswinds, for example, the system gets an initial state $x = [0, 0.05, 0, 0]$, which gives rise to pendulum oscillation without an appropriate control.

Using the proposed controller described in Fig. 3.1 and experimental settings given above, the control parameters which include the optimal weight matrix Q and the resulting feedback vector K are obtained, which are shown in Table 3.3.

Fig. 3.14 shows the evolution curve of the PSO algorithm for optimizing the weight parameters of the designed LQR controller.

The zero-input responses of the MAV pendulum-like oscillation with an initial pendulum angle of 0.05rad, which is brought about by crosswinds in the actual flight environment (See Fig. 3.15).

Compared with the responses without control in Fig. 3.13, the dynamic behaviors of x and ϕ_x state obviously that the closed loop eliminates the pendulum-like oscillation and the state converges to the equilibrium point with an average time of 6 s.

Furthermore, considering a constant interference force acting upon the pendulum system besides the initial state, the resulting responses are given in Fig. 3.16. The interference value is taken as 0.2m/s^2, and the control framework and relative parameters remain the same as mentioned above in Fig. 3.13 and Table 3.3.

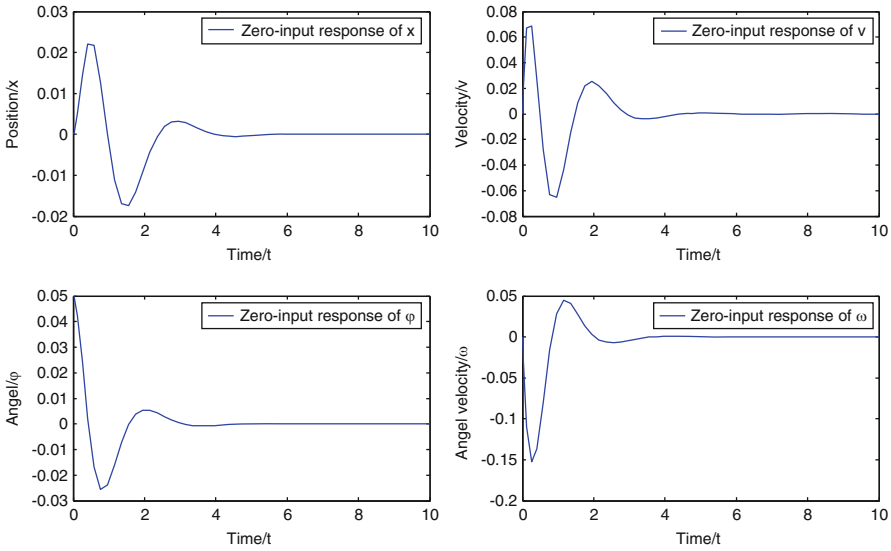

Fig. 3.15 The zero-input response of the MAV pendulum-like oscillation system (Reprinted from Duan and Sun (2013), with kind permission from Springer Science+Business Media)

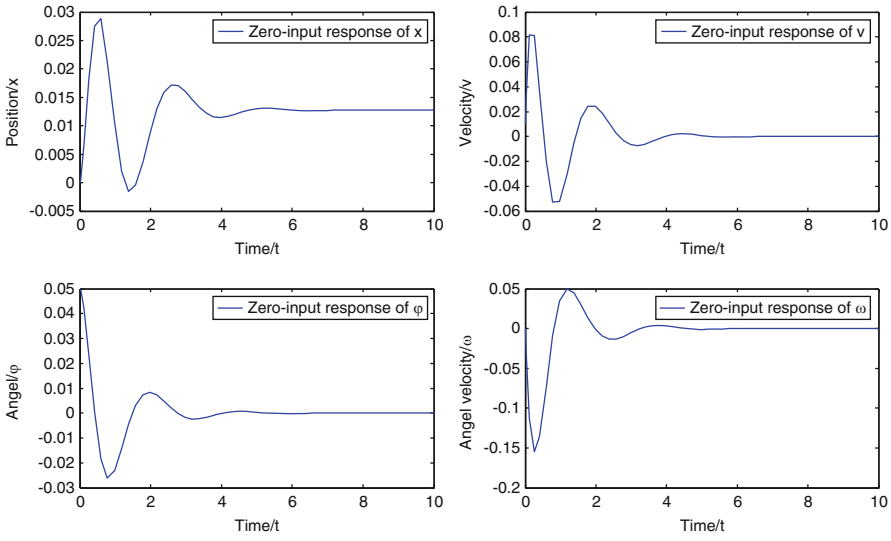

Fig. 3.16 Responses of the MAV pendulum system with a constant disturbance (Reprinted from Duan and Sun (2013), with kind permission from Springer Science+Business Media)

As presented in Fig. 3.16, the PSO-based LQR controller can eliminate the interference of the constant disturbance and the initial state in 6 s. The MAV is stabilized to a state of $x = [0.0128, 0, 0, 0]$, which means that the LQR control design technique based on PSO is robust for external disturbances.

3.4 Conclusions

As a key component of UAV system, controller acts as the brain of UAVs. So the selection of controller parameters is crucial in the design of the flight control system. However, for the existence of strong coupling among the inputs and the nonexistence of mapping relationship between the performance index and the controller parameter, it is a tough work. The 6-DOF nonlinear model of UAVs is illustrated, which is the prerequisite for simplifying and linearizing the mathematical model. Then UAV equations can be simplified into nonlinear equations of 5-DOF with the assumption that the thrust and the resistance of the aircraft maintains the same. And it can be modeled as linear time invariant state-space perturbation models, with the nominal trajectory being steady-level trimmed flight. To reduce the workload of the designers during the process of designing complicated UAV control system, a predator–prey PSO algorithm for identifying parameters of UAV flight control system is presented.

The performance of hover and stare is the key issue to MAVs when carrying out new challenging reconnaissance missions in urban warfare (local, close-up range, hidden reconnaissance, operation between obstacles, and maybe even inside buildings). However, pendulum-like oscillation caused by uncertainty will badly impair the performance of hover and stare, resulting in blurred images or even overturn. However, pendulum-like oscillation caused by uncertainty and external disturbances badly jeopardize the performance of hover and stare, resulting in blurred images or even MAV's overturning. So the second part of this chapter mainly deals with control issue of pendulum-like oscillation in an MAV's hover and stare state in the presence of external disturbances; a novel type of PSO-based LQR controller for stabilizing the pendulum-like oscillation is developed, which can enhance the MAV's performance efficiently. Simulation results verify the feasibility, effectiveness, and robustness of our proposed approach, which provides a more effective way for control law design.

References

Bloss R (2009) Latest unmanned vehicle show features both innovative new vehicles and miniaturization. Ind Robot Int J 36(1):13–18

Duan H, Liu S (2010) Non-linear dual-mode receding horizon control for multiple unmanned air vehicles formation flight based on chaotic particle swarm optimisation. IET Control Theory Appl 4(11):2565–2578

Duan H, Sun C (2013) Pendulum-like oscillation controller for micro aerial vehicle with ducted fan based on LQR and PSO. Sci China Technol Sci 56(2):423–429

Duan H, Zhang X, Xu C (2011) Bio-inspired computing. Science Press, Beijing

Duan H, Yu Y, Zhao Z (2013a) Parameters identification of UCAV flight control system based on predator–prey particle swarm optimization. Sci China Inf Sci 56(1):012202

Duan H, Luo Q, Ma G, Shi Y (2013b) Hybrid particle swarm optimization and genetic algorithm for multi-UAVs formation reconfiguration. IEEE Comput Intell Mag 8(3):16–27

Higashitani M, Ishigame A, Yasuda K (2006) Particle swarm optimization considering the concept of predator–prey behavior. In: Proceedings of IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada. IEEE, pp 434–437

Johnson EN, Turbe MA (2006) Modeling, control, and flight testing of a small-ducted fan aircraft. J Guid Control Dyn 29(4):769–779

Liu F, Duan H, Deng Y (2012) A chaotic quantum-behaved particle swarm optimization based on lateral inhibition for image matching. Optik 123(21):1955–1960

McLean D (1990) Automatic flight control systems. Prentice Hall, New York

Pflimlin J-M, Binetti P, Soueres P, Hamel T, Trouchet D (2010) Modeling and attitude control analysis of a ducted-fan micro aerial vehicle. Control Eng Pract 18(3):209–218

Shi Y, Eberhart R (1998a) A modified particle swarm optimizer. In: Proceedings of the 1998 IEEE International Conference on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Anchorage, AK. IEEE, pp 69–73

Shi Y, Eberhart RC (1998b) Parameter selection in particle swarm optimization. In: Proceedings of 7th International Conference on Evolutionary Programming, California. Springer, Berlin/Heidelberg, pp 591–600

Sun C, Duan H (2013) Artificial Bee colony optimized controller for MAV pendulum. Aircr Eng Aerosp Technol 85(2):104–114

Wang X, Duan H (2013) Predator–prey biogeography-based optimization for bio-inspired visual attention. Int J Comput Intell Syst 6(6):1151–1162

Zhang WY (2004) Study of intelligent robust design approach of large flight envelope flight control system. Master thesis, Beihang University, Beijing

Zhang M, An J (2008) Application of intelligent computation to promote the automation of flight control design. In: Proceedings of International Conference on Intelligent Computation Technology and Automation (ICICTA), Hunan. IEEE, pp 991–994