Available online at www.sciencedirect.com
ScienceDirect
Journal of Bionic Engineering 5 (2008) 340–347

Optimal Formation Reconfiguration Control of Multiple UCAVs Using Improved Particle Swarm Optimization

Hai-bin Duan[1], Guan-jun Ma[1,2], De-lin Luo[3]

1. *School of Automation Science and Electrical Engineering, Beihang University, Beijing* 100191, *P. R. China*
2. *Provincial Key Laboratory for Information Processing Technology, Suzhou University, Suzhou* 215006, *P. R. China*
3. *Center for Systems and Control, Xiamen University, Xiamen* 361005, *P. R. China*

Abstract

Optimal formation reconfiguration control of multiple Uninhabited Combat Air Vehicles (UCAVs) is a complicated global optimum problem. Particle Swarm Optimization (PSO) is a population based stochastic optimization technique inspired by social behaviour of bird flocking or fish schooling. PSO can achieve better results in a faster, cheaper way compared with other bio-inspired computational methods, and there are few parameters to adjust in PSO. In this paper, we propose an improved PSO model for solving the optimal formation reconfiguration control problem for multiple UCAVs. Firstly, the Control Parameterization and Time Discretization (CPTD) method is designed in detail. Then, the mutation strategy and a special mutation-escape operator are adopted in the improved PSO model to make particles explore the search space more efficiently. The proposed strategy can produce a large speed value dynamically according to the variation of the speed, which makes the algorithm explore the local and global minima thoroughly at the same time. Series experimental results demonstrate the feasibility and effectiveness of the proposed method in solving the optimal formation reconfiguration control problem for multiple UCAVs.

Keywords: uninhabited combat air vehicles, particle swarm optimization, control parameterization and time discretization, optimal formation reconfiguration

1 Introduction

Uninhabited Combat Aerial Vehicle (UCAV) is one of inevitable trends of the modern aerial weapon equipments, which develop in the direction of unmanned attendance and intelligence[1]. Research on UCAV directly affects battle effectiveness of the air force, and is also the fatal and fundamental research related to national security.

The advent of multiple UCAVs into the combat arena has led to extensive research activities on the design and control of autonomous UCAVs to achieve specific mission goals. One of the problems of particular interest to researchers is the automatic control of a group of UCAVs flying in close formation[2,3]. Most of the research conducted in recent years has been focused on the coordination and station keeping of multiple UCAVs so as to maintain the relative separations and orientations between the UCAVs in the formation[4] and to track desired flight trajectories[5]. But few studies have been focused on the reconfiguration of UCAV formation[6]. The reconfiguration need to obtain optimal or near-optimal formation performance in the event of a failure, flight-path restriction, or even the total loss of a UCAV. Specifically, failures such as in one or more communication channels[7] might necessitate repositioning of UCAV in order to maintain or maximize the overall benefits from the formation flight. Such failures do not affect the control of individual UCAV in the formation, particularly the ones undergoing reconfiguration. Several theoretical techniques such as graph theory, reconfiguration maps, Dijkstra algorithm[6], or functional optimization[8] have been developed to define the new/optimal positions to be occupied by the UCAVs in the formation.

Multiple UCAVs formation reconfiguration control

Corresponding author: Hai-bin Duan
E-mail: hbduan@buaa.edu.cn

can bring low cost and high efficiency[9]. Formation reconfiguration can be classified as large-scale centralized control problem. We'll obtain the input signals (such as steering angle, throttle, *etc*) of each UCAV by complex calculation to drive the UCAV in a complicated flight maneuver satisfying the constraint that the random distance between two UCAVs must be greater than the safety collision distance and smaller than the communication distance. Ultimately all flight units reach to relative positions of the expectation of demand, forming a new formation.

Time is a very important resource for formation reconfiguration, time-optimal control is one of the main objectives to dynamic system[10,11]. Therefore, studying the time-optimal control of formation reconfiguration is necessary, including fixed terminal state constraints and free terminal state time-optimal problem. Furukawa and Lee *et al.*[12,13] discussed the fix terminal state constraint of time-optimal control in detail. Furukawa *et al.*, Simeon *et al.*, and Saber *et al.*[14–16] discussed the free terminal state of time-optimal control. However, these studies do not consider the flight unit's communication distance and safety collision distance.

Particle Swarm Optimization (PSO) is a population based stochastic optimization technique, which was developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behaviour of bird flocking or fish schooling[17,18]. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA)[19]. The PSO is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators, such as crossover and mutation. It is demonstrated that PSO can get better results in a faster and cheaper way compared with other bio-inspired computational methods. Furthermore, there are fewer parameters to adjust in PSO. In the past few years, PSO has been successfully applied in many fields[20]. One version, with slight modification, works well in a wide range of applications, as well as for specific application focused on a specific requirement[21].

This paper presents a hybrid approach based on Control Parameterization and Time Discretization (CPTD) PSO algorithm, which can solve the free terminal state of multiple UCAVs formation reconfiguration optimal time control problem effectively.

The remainder of this paper is organized as follows: Section 2 presents the mathematical description of multiple UCAVs optimal formation reconfiguration time-optimal control. Subsequently, the UCAVs formation reconfiguration time-optimal control discrete based on CPTD method is proposed in Section 3, and the time-optimal control of formation reconfiguration based on an improved PSO is also presented in this section. Then, simulation results are given to verify the feasibility and effectiveness of the proposed method in Section 4. Our concluding remarks and future work are contained in Section 5.

2 Mathematical description of multiple UCAVs formation reconfiguration time-optimal control

Assume that the number of UCAVs in a formation is N. Without losing generality, we consider the multiple UCAVs formation is moving at a certain high level. Assume the control vector initial time $t = 0$, terminal time $t = T$. Define the i-th UCAV's control input (including steering angle, velocity *etc*) is

$$u_i = \{u_i(t) \mid \forall t \in [0,T]\} \in \Re^{r_i},\ \forall i \in \{1,\cdots,N\}.$$

Then the formation's control input vector is $\boldsymbol{U} = (u_1,\ldots,u_N)$, the continuous control input vector of the formation can be further described as

$$\boldsymbol{U} = (u_1,\cdots,u_N) = \{\boldsymbol{U}(t) \mid \forall t \in [0,T]\}.$$

Define the i-th UCAV's state variable as

$$\boldsymbol{x}_i = [y_i, z_i, \theta_i]^{\mathrm{T}} \in \Re^3,\ \forall i \in \{1,\cdots,N\},$$

where (y_i, z_i) denotes the i-th UCAV coordinates, and θ_i denotes the heading angle of the i-th UCAV. Therefore, the formation system state variables are defined as

$$\boldsymbol{X} = \left(\boldsymbol{x}_1^{\mathrm{T}},\cdots,\boldsymbol{x}_N^{\mathrm{T}}\right)^{\mathrm{T}} \in \Re^{3*N}.$$

The formation system dynamic can be described as

$$\dot{\boldsymbol{X}}(t) = f(t, \boldsymbol{X}(t), U(t)). \tag{1}$$

Give a continuous control input $\boldsymbol{U}$ and the initial state $\boldsymbol{X}(0) = \boldsymbol{X}_0$, then the state of the whole system at any time $t \in (0,T]$ can be determined uniquely in the following form

$$X(t) = X(0) + \int_0^{t^-} f(\tau, X(\tau), U(\tau))\ \mathrm{d}\tau \quad (2)$$

This means that, given the initial state $X(0)$, the state $X(t)$ at any time t can be specified only by the control function U. When this needs to emphasized, the state $X(t)$ can be written with the form $X(t\,|\,U)$ when the control function is defined[22].

Generally, the standard of payoff function can be expressed with the following equation

$$J(U) = \Phi_0(X(T\,|\,U)) + \int_0^T L_0(t, X(t\,|\,U), U(t))\mathrm{d}t\,.$$

The problem may also be subject to a variety of other constraints, generally in the form

$$\begin{cases} g_i(U) = \Phi_i(X(\tau_i\,|\,U)) + \int_0^{\tau_i} L_i(t, X(t\,|\,U), U(t))\mathrm{d}t \le 0 \\ \forall i \in \{1, \cdots, M\} \end{cases}.$$

For a single UCAV system, the optimal control problem can be formulated as finding the continuous control input U and terminal time T that minimizes a payoff function $J(U)$:

$$\min_{u_1, T} \cdots \min_{u_N, T} J(U)\ , \quad (3)$$

$$J(U) = T\ . \quad (4)$$

The function U and time T are normally constrained with the following equation

$$U_{\min} \le U(t) \le U_{\max},\ \forall t \in [0, T),\ 0 < T. \quad (5)$$

Free terminal constraints can be defined as

$$\begin{aligned} g_1(U, \Delta t) = \sum_{i=1}^{N} \{ & [(z_i(T) - z_m(T)) - z_i^m]^2 + \\ & [(y_i(T) - y_m(T)) - y_i^m]^2 + \\ & [(\theta_i(T) - \theta_m(T)) - \theta_i^m]^2 \} = 0, \end{aligned} \quad (6)$$

where $m \in \{1, \cdots, N\}$, defines the m-th UCAV as the central UCAV, $[y_i^m, z_i^m, \theta_i^m]^{\mathrm{T}}$ represents the desired relative coordinates of i-th UCAV with respect to UCAV m.

Define the distance between any two UCAVs as $d^{i,j}(x_i(t), x_j(t))$, $i, j \in \{1, \cdots, N\}$, and

$$d^{i,j}(x_i(t), x_j(t)) = \sqrt{\left(x_i(t) - x_j(t)\right)^2 + \left(y_i(t) - y_j(t)\right)^2 + \left(z_i(t) - z_j(t)\right)^2}$$

In order to avoid collision, $d^{i,j}(x_i(t), x_j(t))$ must be greater than the safety collision distance D_{safe}

$$\begin{cases} d^{i,j}(x_i(t), x_j(t)) \ge D_{\mathrm{safe}} \\ \forall t \in [0, T],\ \forall_{i \ne j} i, j \in \{1, \cdots, N\} \end{cases}. \quad (7)$$

To ensure the real-time communication and update the combat situation of the formation, $d^{i,j}(x_i(t), x_j(t))$ must be smaller than the communication distance D_{comm}

$$\begin{cases} d^{i,j}(x_i(t), x_j(t)) \le D_{\mathrm{comm}} \\ \forall t \in [0, T],\ \forall_{i \ne j}\ i, j \in \{1, \cdots, N\} \end{cases}. \quad (8)$$

Generally, the mathematical description of multiple UCAVs formation reconfiguration time-optimal control can be summarized as: meeting the restrictive conditions Eq. (1) and Eqs. (5) to (8), finding a continuous control input U and terminal time T that satisfy Eqs. (3) and (4).

3 Improved PSO algorithm in multiple UCAVs optimal formation reconfiguration time-optimal control

PSO is a population based stochastic optimization technique inspired by social behaviour of bird flocking or fish schooling. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. Each particle keeps track of its coordinates in the problem space which are associated with the best solution, the fitness value, it has achieved so far. This value is called p_{best}. Another "best" value that is tracked by the particle swarm optimizer is the best value obtained so far by any particle of the neighbours of the particle. This location is called l_{best}. When a particle takes all the population as its topological neighbours, the best value is a global best and is called g_{best}. The PSO concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its p_{best} and l_{best} locations (local version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward p_{best} and l_{best} locations.

The use of PSO to solve optimization problem is not constrained by whether the objective function is linear or not, suitable for solving optimal formation reconfiguration problem. However, the control inputs of each flight unit are continuous, PSO can not solve the continuous control inputs. Therefore, first, the control inputs of each fight unit are piecewise linearized, using the approximation piecewise linearization control inputs substitute for the continuous inputs, then using PSO to find the global optimal solution, obtained piecewise linearization control inputs. Based on the above ideas, this paper adopts CPTD method to obtain the approximate payoff function and constraint condition, simplify the problem in description and handling, and then uses improved PSO algorithm to find the optimal control inputs U's approximate solution $\hat{\boldsymbol{U}}(t;n_p,\boldsymbol{\Omega})$ until meet constraints Eqs. (5), (7), (8), (13), and (14).

3.1 Formation reconfiguration time-optimal control discrete based on CPTD method

The CPTD method is characterized by three features:

(1) The continuous control inputs u_i is approximated by a piecewise function with a set of static parameters[14]. The terminal time T is first partitioned into n_p time interval, the partition is conducted to introduce a piecewise function with n_p constants that approximates each continuous control inputs.

(2) The terminal time T is the function of each time interval Δt_p.

(3) The static control parameter sets and the time-step interval are found by minimizing the payoff function with a standard non-linear parametric optimization method (such an improved PSO).

In formation reconfiguration time-optimal control discrete based on CPTD method, it can be divided into the following three steps:

Step 1: The division of the terminal time T

The terminal time T is partitioned into $n_p \in \{1,2,\cdots\}$ time intervals, and each $\Delta t_p \in \Re^+$, so

$$T = n_p \cdot \Delta t_p \tag{9}$$

In each time interval Δt_p, according to the corresponding control inputs, the formation motion Eq. (1) conducts numerical integration.

Step 2: The piecewise linear of control inputs

For the n_p time interval, define $r_i \times n_p$ constants for the i-th system as

$$\boldsymbol{\Omega}_i = \left\{\sigma_j^i \in \Re^{r_i} \mid \forall j \in \left\{1,\cdots,n_p\right\}\right\},\forall i \in \left\{1,\cdots,N\right\}.$$

Then, each of the continuous control inputs for the i-th system can be approximated by a piecewise function with constants as follows

$$\hat{u}_i(t;n_p,\boldsymbol{\Omega}_i) = \sum_{j=1}^{n_p} \sigma_j^i \chi_j(t) \cong u_i(t), \tag{10}$$

where $\chi_j(t)$ is expressed with the following equation

$$\chi_j(t) = \begin{cases} 1 & (j-1)\cdot \Delta t_p \le t \le j \cdot \Delta t_p \\ 0 & \text{otherwise} \end{cases}. \tag{11}$$

Define the set of all piecewise constants for all systems as $\Omega = \{\Omega_1,\ldots,\Omega_N\}$. The set of approximated control inputs for all the systems can by written as

$$\hat{\boldsymbol{U}}(t;n_p,\boldsymbol{\Omega}) = \left\{\hat{u}_1(t;n_p,\boldsymbol{\Omega}_1),\cdots,\hat{u}_N(t;n_p,\boldsymbol{\Omega}_N)\right\}.$$

Finding $\hat{\boldsymbol{U}}(t;n_p,\boldsymbol{\Omega})$ therefore results in finding the parameter set $\boldsymbol{\Omega}$. Most important for this approximation in practical implementations is an appropriate choice for n_p. Increasing n_p results in an exponential increase in computation time, reducing n_p results in loss of accuracy.

Step 3: Approximate parametric

The approximation

$$\hat{\boldsymbol{U}}(t;n_p,\boldsymbol{\Omega}) = \left\{\hat{u}_1(t;n_p,\boldsymbol{\Omega}_1),\cdots,\hat{u}_N(t;n_p,\boldsymbol{\Omega}_N)\right\},$$

can be derived form specification of $\boldsymbol{\Omega}$ and Δt_p. The fact that finding $\hat{\boldsymbol{U}}(t;n_p,\boldsymbol{\Omega})$ and T is equivalent to finding $\boldsymbol{\Omega}$ and Δt_p introduces an approximate payoff function and constraint function J. As a result, the dynamic optimization problem can be transformed into the following static optimization problem

$$J \cong \min_{\boldsymbol{\Omega},\Delta t_p}(n_p \cdot \Delta t_p) \tag{12}$$

subject to the following bounds

$$\begin{cases} (u_{\min})_i \le \sigma_j^i \le (u_{\max})_i \\ \forall i \in \left\{1,\cdots,N\right\},\ \forall j \in \left\{1,\cdots,n_p\right\},\ 0 < \Delta t_p \end{cases}, \tag{13}$$

and the free terminal constraint

$$\begin{aligned} \hat{g}_1(\boldsymbol{\Omega},\Delta t) = \sum_{i=1}^{N}\{&[(z_i(T)-z_m(T))-z_i^m]^2 + \\ &[(y_i(T)-y_m(T))-y_i^m]^2 + \\ &[(\theta_i(T)-\theta_m(T))-\theta_i^m]^2\} = 0. \end{aligned} \tag{14}$$

The state of the whole system can be appropriated as follows

$$\dot{\boldsymbol{X}}(t) \cong f(t, \boldsymbol{X}(t), \hat{\boldsymbol{U}}(t; n_p, \boldsymbol{\Omega})). \quad (15)$$

3.2 Time-optimal control of formation reconfiguration based on PSO

Using CPTD piecewise linear control inputs, the PSO algorithm can be used to resolve time-optimal control of formation reconfiguration problem.

(1) Construction of particles' position

Vector $\boldsymbol{\Omega} = \{\boldsymbol{\Omega}_1, \cdots, \boldsymbol{\Omega}_N\}$ combines with Δt_p as the particles' position vector by using float decimal encoding method. Thus, the position of each particle can be expressed as $P = \boldsymbol{\Omega}_1 \boldsymbol{\Omega}_2 \cdots \boldsymbol{\Omega}_N \Delta t_p$. The control parameter $\boldsymbol{\Omega}_i$ is a constant array, which can be expressed as follows

$$\boldsymbol{\Omega}_i = \begin{bmatrix} \sigma_{11}^i & \sigma_{21}^i & \cdots & \sigma_{n_p 1}^i \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1 r_i}^i & \sigma_{2 r_i}^i & \cdots & \sigma_{n_p r_i}^i \end{bmatrix}, \begin{cases} \forall i \in \{1, 2, \cdots N\} \\ \forall j \in \{1, 2, \cdots n_p\} \\ \forall k \in \{1, 2, \cdots r_i\} \end{cases},$$

where σ_{jk}^i is the k-th component of $\hat{u}_i(t; n_p, \boldsymbol{\Omega}_i)$ at the j-th time interval. As each column of $\boldsymbol{\Omega}_i$ represents the control parameter of i-th UCAV at a time interval, we can expand the $\boldsymbol{\Omega}_i$ by column, and combine it with Δt_p, eventually straightened into a length of $N \times n_p \times r_i + 1$ floating point code series. The particle's position vector can be expressed as follows

$$\boldsymbol{x} = [((\sigma_{11}^1, \sigma_{12}^1, \cdots, \sigma_{1 r_i}^1), \cdots, (\sigma_{n_p 1}^1, \sigma_{n_p 2}^1 \cdots, \sigma_{n_p r_i}^1)), \cdots \\ , ((\sigma_{11}^N, \sigma_{12}^N, \cdots, \sigma_{1 r_i}^N), \cdots, (\sigma_{n_p 1}^N, \sigma_{n_p 2}^N \cdots, \sigma_{n_p r_i}^N)), \Delta t_p].$$

(2) Initialization of the population

Given the population size and the max iteration, initialize the position vector and velocity randomly.

(3) Computation of the payoff function

Considering of the time-optimal control constraints[23], define the extended payoff function as follows

$$J_{\text{extend}} = \min_{\boldsymbol{\Omega}, \Delta t_p} \{(n_p \cdot \Delta t_p) + \sigma^* \cdot \hat{g}_1(\boldsymbol{\Omega}, \Delta t) + \\ \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} [\sigma_{ij} \cdot \max(0, D_{\text{safe}} - d^{i,j}(x_i(t), x_j(t))) \\ + \sigma_{ij}^{'} \cdot \max(0, \ d^{i,j}(x_i(t), x_j(t)) - D_{\text{comm}})]\}, \quad (16)$$

where σ_{ij} and $\sigma_{ij}^{'}$ are the safety distance punishment constant coefficient, and communication distance punishment constant coefficient, respectively. σ^* is the punishment constant coefficient of terminal constraint. If σ_{ij}, $\sigma_{ij}^{'}$, σ^* are big enough (must be the positive number), then the primitive payoff function Eq. (12) and the constraints condition Eqs. (7), (8), and Eq. (14) are equal to expression Eq. (16).

(4) Position and velocity updating rule can be expressed as follows

$$\begin{cases} v_i = w v_i + c_1 \cdot r_1 \cdot (p_{\text{best}i} - x_i) + c_2 \cdot r_2 \cdot (g_{\text{best}i} - x_i) \\ x_i = x_i + v_i \end{cases}, \quad (17)$$

where w is an inertia weight, v_i is the previous velocity value, c_1 and c_2 denote different acceleration coefficients, r_1 and r_2 are random values between 0 and 1, $p_{\text{best}i}$ is the best position found by the particle, $g_{\text{best}i}$ is the particle to be followed.

(5) Improved PSO model

When particles are exploring the search space, if some particles have found the current best position, the others will fly toward it. If the best position is a local optimum, particles cannot explore over again in the definite search space. In this occasion, the PSO algorithm will be trapped into the local optimum, which is called premature convergence phenomenon. The higher dimension of the optimized function is, the easier the algorithm is to appear to this phenomenon. Therefore, this work integrates the various velocity variations. The core idea is that when the payoff function does not change apparently more than 20 iterations, the variation process starts according to the following equation

$$\begin{cases} |v_{i,d}| = \min\{|v_{1,d}|, |v_{2,d}|, \ldots\ldots |v_{m,d}|\} \\ i \in \{1, 2, 3, \ldots\ldots, m\}, \ d \in \{1, 2, 3, \ldots\ldots, D\} \end{cases}. \quad (18)$$

If $Rand < Rate$, $v_{i,d}$ = random values between $V_{\max}$ and $-V_{\max}$, where $V_{\max}$ denotes the maximum velocity of the particles, $Rate$ denotes the mutation probability.

(6) In order to take the global search and local search into account, we assume that the speeds of all dimensions are independent, and each dimension has a threshold. When too many particle velocities achieve this value, it can decline by adjust the speed threshold automatically, then can dynamically adjust the speeds of the various particles. This scheme can be described as follows

$$F_d(t) = F_d(t-1) + \sum_{i=1}^{\text{size}} b_{id}(t) , \tag{19}$$

$$b_{id}(t) = \begin{cases} 0, & v_{id}(t) > T_d \\ 1, & v_{id}(t) < T_d \end{cases} , \tag{20}$$

$$\begin{cases} \text{if} & F_d(t) > k_1 \\ \text{then} & F_d(t) = 0;\ T_d = T_d / k_2 \end{cases} , \tag{21}$$

where $F_d(t)$ is the frequency used to record the escaping times of d-th dimension, T_d is the threshold, k_1 and k_2 are constants, k_2 controls the decline rate of threshold. By limiting the particles flying speed, the algorithm effectively coordinates the relationship of global search and local search, increases the global optimization capacity when accelerates the convergence speed.

4 Experimental results

In order to investigate the feasibility and effectiveness of the proposed PSO approach to multiple UCAVs' optimal formation reconfiguration control, a series of experiments have been conducted under complex combat field environment. The proposed approach has been coded in Matlab language and implemented on PC-compatible with 1024 MB of RAM under the Microsoft Windows XP.

Consider N UCAVs, the pose of the i-th UCAV is denoted as $x_i = [y_i, z_i, \theta_i]^{\mathrm{T}} \in \Re^3$, $i = 1, 2, \cdots, N$, where y_i and z_i are the coordinates of the centre of the rear-axle and θ_i is the heading angle. The control inputs (u_i) to the UCAV are the velocity at the centre of the rear-axle v_i and the average steering angle γ_i. The equations of motion for the i-th UCAV are

$$\begin{cases} \dot{y}_i(t) = v_i(t) \cdot \cos(\theta_i) \\ \dot{z}_i(t) = v_i(t) \cdot \sin(\theta_i) \\ \dot{\theta}_i(t) = \dfrac{v_i(t)}{l_i} \tan \gamma_i(t) \end{cases} ,$$

where l_i is the length of the i-th UCAV, $i = 1, 2, \cdots, N$.

In our experiment, $N = 5$, it means there are five UCAVs. $l_i = 3.15$ m, $v_{\min} = 0$, $v_{\max} = 4.4$ Ma, $\gamma_{\min} = -53°$, $\gamma_{\max} = 53°$. Assume that the 3rd vehicle is the centre of the formation, $D_{\text{safe}} = 5$ km, $D_{\text{comm}} = 45$ km. Giving the arbitrary initial state and the relative state, after the optimal control, the UCAVs can move to the desired relative V-shape formation.

Fig. 1 describes the relationship of the payoff function and iterations. Fig. 2 describes the formation reconfiguration trajectory by improved PSO. From Fig. 2, it is clear that the UCAVs successfully moved to the desired relative V-shape formation. Fig. 3 shows the distance between any two UCAVs is much smaller than the communication distance D_{comm} and greater than the safety distance D_{safe}. Figs. 4 and 5 show the computed optimal velocities and steering angles for all UCAVs respectively. The steering angle figure shows that many UCAVs attempt to drive at their maximum or minimum steering angle.

It is clearly that the improved PSO can always find group of optimal solutions to meet the payoff function requirements and various formations system constraints to achieve multiple UCAVs formation reconfiguration.

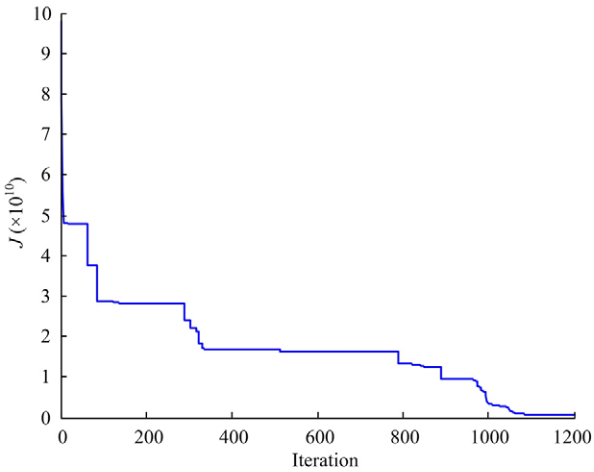

Fig. 1 Evolution curve of improved PSO.

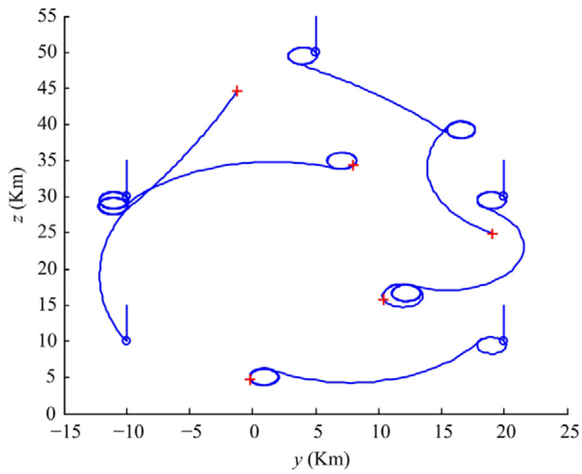

Fig. 2 The 5 UCAVs' reconfiguration trajectory. "O" is the initial state, and "+" is the desired relative formation state.

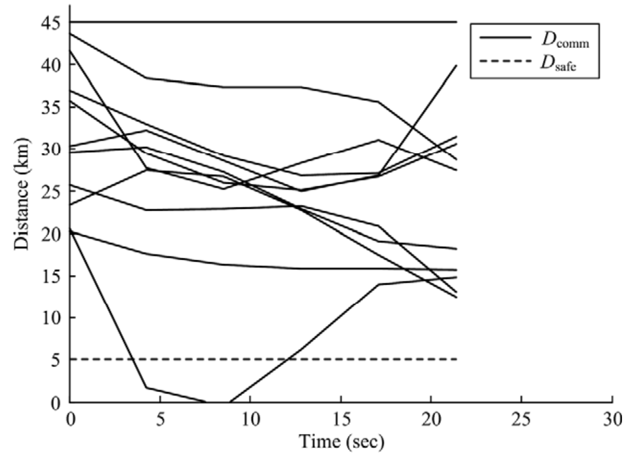

Fig. 3 The change curve of spacing interval of UCAVs.

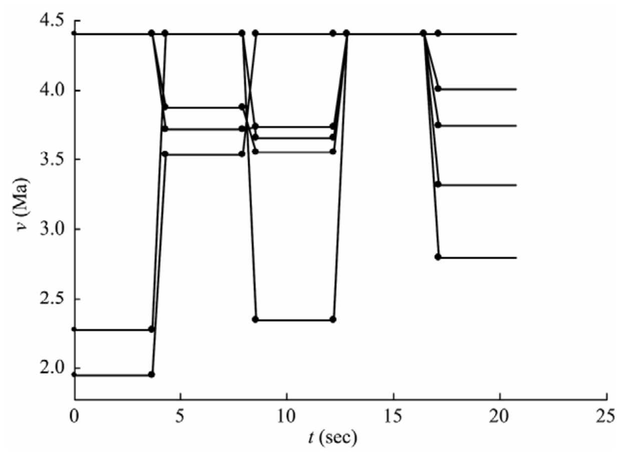

Fig. 4 Computed optimal velocity.

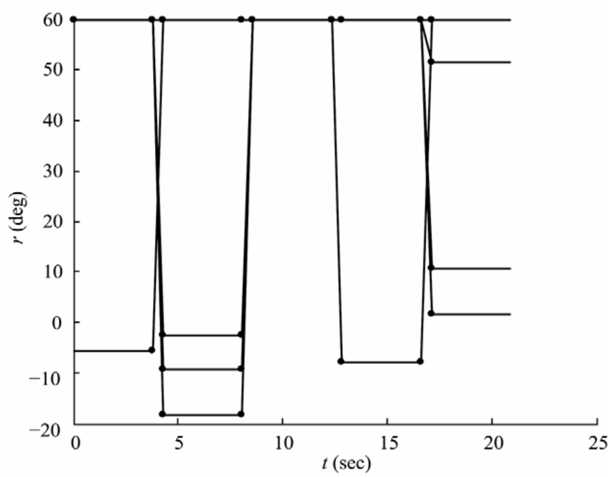

Fig. 5 Computed optimal steering angle.

5 Conclusions and future work

An effective approach to the problem of coordinated optimal formation reconfiguration control for multiple UCAVs has been presented in this paper. Experimental results show that the proposed PSO algorithm can solve the optimal control of the multiple UCAVs formation reconfiguration problem effectively.

The improved PSO algorithm can not only solve the single-formation reconfiguration problem, the minimum energy control, the shortest time and minimum energy integrated control problems, but also solve the centralized control of complex systems, such as multiple robots formation reconfiguration, and the multiple unmanned ground/underground vehicles coordinate control problems. Our future work will focus on the exact application of the proposed method in optimal formation reconfiguration control of multiple UCAVs, while the multiple UCAVs communication is another problem in this field.

Acknowledgement

This work is supported by the Natural Science Foundation of China (Grant No.60604009), the Aeronautical Science Foundation of China (Grant No. 2006ZC51039), the Beijing NOVA Program Foundation of China (Grant No. 2007A017), and the Open Fund of the Provincial Key Laboratory for Information Processing Technology, Suzhou University (Grant No. KJS0821). The authors are grateful to the anonymous referees for their valuable comments and suggestions.

References

[1] Zheng C W, Li L, Xu F J. Evolutionary route planner for unmanned air vehicles. *IEEE Transactions on Robotics and Automation*, 2005, **21**, 609–620.

[2] Blake W, Multhopp D. Design, performance and modeling considerations for close formation flight. *Proceedings of the* 1998 *AIAA Guidance*, *Navigation*, *and Control Conference*, Reston, USA, 1998, 476–486.

[3] Dargan J, Pachter M, Dazzo J. Automatic formation flight control. *Proceedings of the* 1992 *AIAA Guidance*, *Navigation*, *and Control Conference*, Washington DC, USA, 1992, 838–857.

[4] Fierro R, Belta C, Desai J, Kumar V. On controlling aircraft formations. *Proceedings of the* 40*th IEEE Conference on Decision and Control*, Arlington, USA, 2001, **2**, 1065–1070.

[5] Koo T, Shahruz S. Formation of a group of unmanned aerial vehicles (UAVs). *Proceedings of the American Control*

Conference, Arlington, USA, 2001, 69–74.

[6] Giuletti F, Pollini L, Innocenti M. Autonomous formation flight. *IEEE Control Systems Magazine*, 2000, **20**, 34–44.

[7] Pollini L, Giuletti F, Innocenti M. Robustness to communication failures within formation flight. *Proceedings of the American Control Conference*, Arlington, USA, 2002, 2860–2866.

[8] Chichka D F, Speyer J, Park C. Peak-seeking control with application to formation flight. *Proceedings of the* 38*th IEEE Conference on Decision and Control*, Arlington, USA, 1999, **3**, 2463–2470.

[9] Shannon Z, John K T, Shankar S. Hybrid system design for formations of autonomous vehicles. *Proceedings of the* 42*nd IEEE Conference on Decision and Control*, Maui, Hawaii, USA, 2003, 1–6.

[10] Teo K L. *A Unified Computational Approach to Optimal Control Problems*, Longman Scientific and Technical, New York, 1991.

[11] Lee Y D, Lee B H, Kim H G. An evolutionary approach for time optimal trajectory planning of a robotic manipulator. *Information Sciences*, 1999, **113**, 245–260.

[12] Furukawa T. Time-subminimal trajectory planning for discrete non-linear systems. *Engineering Optimization*, 2002, **34**, 219–243.

[13] Lee K Y, Dissanayake G. Numerical solution for a near minimum time trajectory for two coordinated manipulators. *Engineering Optimization*, 1998, **30**, 227–247.

[14] Furukawa T, Durrant-Whyte H F, Bourgault F, Dissanayake G. Time-optimal coordinated control of the relative formation of multiple vehicles. *Proceedings the* 2003 *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Kobe, Japan, 2003, 259–264.

[15] Simeon T, Leroy S, Lauumond J P. Path coordination for multiple mobile robots: A resolution-complete algorithm. *IEEE Transactions on Robotics and Automation*, 2002, **18**, 42–49.

[16] Saber R F, Dunbar W B, Murray R M. Cooperative control of multi-vehicle systems using cost graphs and optimization. *Proceedings of the* 2003 *American Control Conference*, Denver, USA, 2003, 2217–2222.

[17] Kennedy J, Eberhart R. Particle swarm optimization. *Proceedings of the* 1995 *IEEE International Conference on Neural Network*s, Perth, Australia, 1995, 1942–1948.

[18] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. *Proceedings of the* 6*th International Symposium on Micro-Machine and Human Science*, Nagoya, Japan, 1995, 39–43.

[19] Fu G J, Wang S M, Liu S Y. An improved velocity mutation particle swarm optimizer. *Computer Engineering and Application*, 2006, **13**, 48–51.

[20] Hao R, Wang Y J, Wang Q. An improved particle swarm optimization based on self-adaptive escape velocity. *Journal of Software*, 2005, **16**, 2036–2044.

[21] Carlos C A, Gregprop T P, Maximino S L. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 2004, **8**, 256–279.

[22] Chen Y M, Chang S H. An agent-based simulation for multi-UAVs coordinative sensing. *International Journal of Intelligent Computing and Cybernetics*, 2008, **1**, 269–284.

[23] Xiong W, Chen Z J, Zhou R. Optimization for multiple flight vehicles formation reconfiguration using hybrid genetic algorithm. *Proceeding of the first Chinese Guidance*, *Navigation and Control Conference*, Beijing, China, 2007, 501–506.