Published in IET Control Theory and Applications
Received on 27th May 2009
Revised on 16th December 2009
doi: 10.1049/iet-cta.2009.0256

ISSN 1751-8644

Non-linear dual-mode receding horizon control for multiple unmanned air vehicles formation flight based on chaotic particle swarm optimisation

H.B. Duan S.Q. Liu

National Key Laboratory of Science and Technology on Holistic Control, School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, People's Republic of China
E-mail: hbduan@buaa.edu.cn

Abstract: This study presents a non-linear dual-mode receding horizon control (RHC) approach to investigate the formation flight problem for multiple unmanned air vehicles (UAVs) under complicated environments. A chaotic particle swarm optimisation (PSO)-based non-linear dual-mode RHC method is proposed for solving the constrained non-linear systems. The presented chaotic PSO derives both formation model and its parameter values, and the control sequence is predicted in this way, which can also guarantee the global convergence speed. A dual-model control strategy is used to improve the stability and feasibility for multiple UAVs formation flight controller, and the state-feedback control is also adopted, where the model is based on the invariant set theory. Series experimental results show the feasibility and validity of the proposed control algorithm over other algorithms. The proposed approach is also a promising control strategy in solving other complicated real-world problems.

1 Introduction

In recent years, formation control of multiple unmanned air vehicles (UAVs) has become a challenging interdisciplinary research topic, while autonomous formation flight is an important research area in the aerospace field [1–5]. The main motivation is the wide range of military and civilian applications, where UAVs formations could provide a low cost and efficient alternative to existing technology [6].

Multiple UAVs teams flying in formations with precisely defined geometries have many advantages, such as energy saving when the vortex forces are taken into account [7]. Formation flight can also be used for airborne refuelling and quick deployment of troops and vehicles [8]. Formation flight can be regarded as a complicated control problem, which computes the inputs driving the UAVs along challenging manoeuvres while maintaining relative positions as well as safe distances between each UAV pair. The challenge here lies in designing a formation controller that is computationally simple, yet robust. Several approaches that have been applied to formation problem include extreme seeking feedback [2], decentralised overlapping control [3], constraint forces [8] and vision-based frameworks [9]. Receding horizon control (RHC), also referred to as model predictive control (MPC) or moving horizon control, has also been used in UAVs formation control [10–12]. In [10], Dunbar and Murray theoretically demonstrated the stability of distributed MPC with a sufficiently fast update period. For problems that admit parallelisation, convergence to the centralised solution is guaranteed. The distributed implementation performs differently in general than the centralised implementation [10]. In [11], decentralised RHC is applied to high-level controller design of the organic air vehicle (OAV) as a linear system to achieve formation flight and avoid collisions using emergency switching control. However, the decentralised scheme is not robust enough

because OAVs are very sensitive to wind gust disturbances. The work by Wesselowski and Fierro [12] develops a dual-model MPC algorithm and uses terminal constraints to ensure stability.

RHC is an optimisation-based control method, which has made a significant impact on industrial control engineering, and is being increasingly applied in process controls. RHC is the most natural control method, and in some cases the only methodology for control of systems that are governed by constrained dynamics [13]. RHC rely on an optimisation of a predicted model response with respect to plant input to determine the best input changes for a given state. RHC also supports linear and non-linear control, equality and inequality constraints and offers extensive flexibility in the formulation of the cost function to be minimised, giving RHC a potential advantage over passive state feedback control laws [14]. In view of these advantages, a non-linear dual-mode RHC is applied to design the multiple UAVs formation flight controller in this work.

However, RHC has some inherent disadvantages when applied to multiple UAVs formation control, such as the online computational cost. The large computational burden of the online optimisation has limited RHCs for further applications. Also, in a real-time application, it may be difficult to achieve negligible solution times for the optimisation algorithm that must be run at every step. However, with faster computers and improved methods, RHC may be put in use for the fast-moving world of mobile vehicles, such as formation control of a set of UAVs. Particle swarm optimisation (PSO) is utilised in this paper to solve the optimisation problem in RHC because of its good performance of global exploration and rapid convergence.

PSO algorithm is an optimisation algorithm inspired by social behaviour in nature. PSO has been developed for the non-linear continuous optimisation problem, based on the experience gained from the study of artificial life and psychological research. Like genetic algorithm (GA), the PSO is a population-based optimisation method that searches multiple solutions in parallel [15]. However, PSO employs a cooperative strategy unlike GA, which utilises a competitive strategy. PSO technique finds the optimal solution using a population of particles. Each particle represents a candidate solution to the problem. The particles change their positions by flying around the search space until a relatively unchanging has been encountered, or the stop criteria is satisfied. It is well known that the PSO techniques can provide a high-quality solution with simple implementation and fast convergence [16, 17].

The major contributions of this paper are the formation-controller design for a group of UAVs using a non-linear dual-mode RHC, and chaotic PSO (CPSO) is also adopted in this new approach. The rest of this paper is organised as follows: the next section describes the non-linear UAV model, formation structure and the Virtual Leader employed in our model. Section 3 presents the fundament of basic PSO, and the development of the dual-mode RHC controller, including CPSO and a collision avoidance strategy is presented in Section 4. Section 5 gives the experimental results to verify the performance of the proposed approach for multiple UAVs formation flight problem. Our concluding remarks are contained in Section 6.

2 Leader-following formation model

The point mass model is considered for formation flight. Each UAV is assumed to fly at a constant altitude, parallel to the two-dimensional region to be surveyed. A commonly used non-linear kinematics model that represents a UAV with zero or negligible velocity in the direction perpendicular to the UAV's heading is applied to our model.

$$\begin{aligned} \dot{x} &= v\cos\psi \\ \dot{y} &= v\sin\psi \\ \dot{v} &= u \\ \dot{\psi} &= \omega \end{aligned} \tag{1}$$

where x and y are the Cartesian coordinates of the UAV, v is the velocity and ψ is the heading angle in the (x, y) plane [6, 18]. The acceleration in the longitudinal direction u and angular turn rate ω are assumed to be the control inputs to the UAV. Fig. 1 shows the UAV position and orientation in the plane coordinate system.

In a typical multiple UAVs formation flight, the wingman follows the trajectory of the leader UAV, taking other aircrafts as reference to keep its own position inside the formation. In a large formation, intra-aircraft distances must be kept constant [1]. The formation model in this paper adopts leader mode strategy (as shown in Fig. 2), which means each wingman UAV takes its trajectory references from the leader UAV, while the altitude is the same for all. The leader UAV takes charge of formation trajectory.

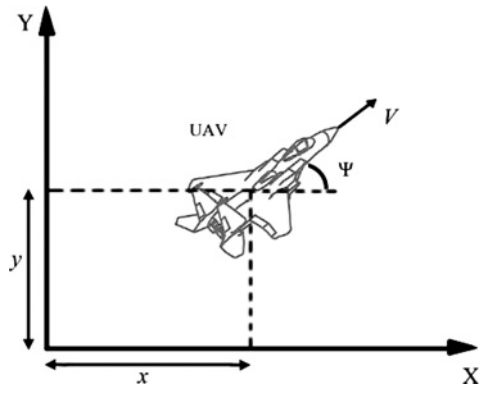

Figure 1 *UAV position and orientation*

doi: 10.1049/iet-cta.2009.0256

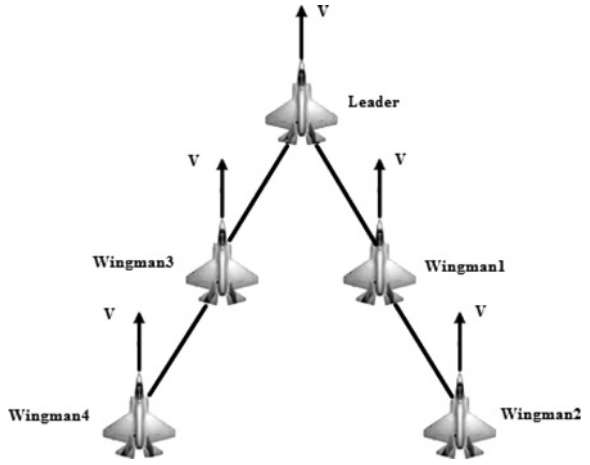

Figure 2 *Multiple UAVs formation*

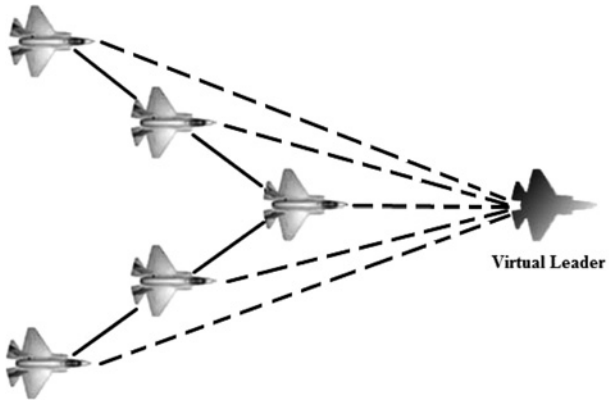

Figure 3 *Multiple UAVs formation with the virtual leader*

The virtual leader is employed in our model to replace the real UAV leader so that UAVs adjust speed and heading angle based on the relative states of virtual leader (as shown in Fig. 3). Then a multiple UAVs formation, defined with respect to all the real UAVs as well as to virtual leader, should be maintained at the same time as the virtual leader tracks its reference trajectory. The key advantage of the virtual UAV leader is that a physical UAV leader is subject to destruction, while the virtual leader can never be damaged. The virtual leader provides a stabile, robust reference for formation control [19, 20].

3 Particle swarm optimisation

The PSO algorithm is an evolutionary computation technique inspired by social behaviour of bird flocking in nature [16]. In 1998, Shi and Eberhart [17, 21] first introduced the inertia weights w into the basic PSO model, by adjusting w to improve the performances of the PSO algorithm. Fig. 4 [22] describes the schematic diagram of PSO.

In Fig. 4, each individual particle i has the following properties: a current position in search space p_i; a current velocity vp_i; and a personal best position in search space

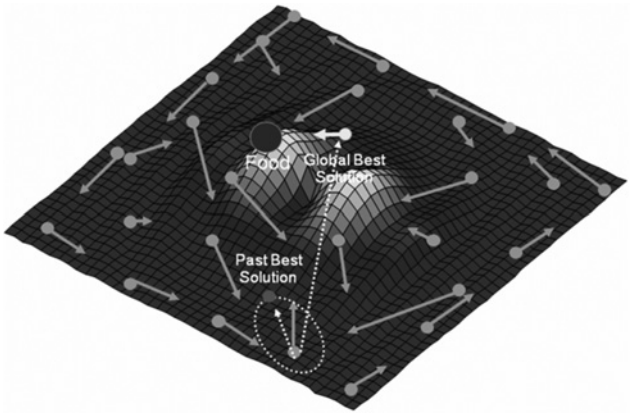

Figure 4 *Schematic diagram of PSO*

$p\text{best}_i$. The personal best position $p\text{best}_i$ corresponds to the position in search space where particle i had the minimisation as determined by the objective function f, assuming a minimisation task. The global best position denoted by $g\text{best}$ represents the position yielding the lowest error among all the $p\text{best}_i$. Equations (2) and (3) define how the personal and global best values are updated at time t, respectively.

$$p\text{best}_i(t+1) = \begin{cases} p\text{best}_i(t), & \text{if } f(p\text{best}_i(t)) \le f(p_i(t+1)) \\ p_i(t+1), & \text{if } f(p\text{best}_i(t)) > f(p_i(t+1)) \end{cases} \tag{2}$$

$$g\text{best}(t) \in \{p\text{best}_1(t),\ p\text{best}_2(t), \ldots, p\text{best}_{ps}(t)\}$$
$$f(g\text{best}(t)) = \min\{f(p\text{best}_1(t)), f(p\text{best}_2(t)), \ldots, f(p\text{best}_{ps}(t))\} \tag{3}$$

Each particle in the swarm is updated during iteration by using (4) and (5). Two pseudo-random sequences, $\text{rand}_1 \sim (0, 1)$ and $\text{rand}_2 \sim (0, 1)$ are used to effect the stochastic algorithm nature. The velocity update is

$$vp_i(t+1) = w \cdot vp_i(t) + c_1 \cdot \text{rand}_1(t) \cdot [p\text{best}(t) - p_i(t)] + c_2 \cdot \text{rand}_2(t) \cdot [g\text{best}(t) - p_i(t)] \tag{4}$$

The new position of the particle is updated by adding the new velocity to the current position

$$p_i(t+1) = p_i(t) + vp_i(t+1) \tag{5}$$

The PSO algorithm consists of repeated application of (2)–(5). In theory, particles of a swarm may benefit from the prior discoveries and experiences of all the members of a swarm when foraging. The key point of PSO is that particles in the swarm share information with each other, which offers some sort of evolutionary advantage. Therefore, owing to the simple concept, easy implementation and quick convergence, PSO has gained much attention and wide applications in solving continuous non-linear optimisation problems [23].

However, the performance of PSO greatly depends on its parameters, and similar to GA and ant colony optimisation, it often suffers from being trapped in local optimum [24, 25].

4 CPSO-based non-linear dual-mode RHC formation-controller design

4.1 Non-linear RHC

Non-linear RHC is that the finite-time optimal control law is computed by solving an online optimisation problem, and linear RHC theory is quite mature so far [13]. Generally, many systems are inherently non-linear, and they are often required to operate over a wide range of operating conditions. Linear models are often inadequate to describe the process dynamics and non-linear models have to be used. This motivates the use of non-linear RHC. This paper focuses on the application of RHC techniques to non-linear formation model.

The optimisation problems over the finite horizon, on which the RHC is based, can be applied to a broad class of systems, including non-linear systems and time-delayed systems. Thus, the RHC has the same broad applications even for non-linear systems.

$$\begin{aligned} &\min_u J = f(x, u; t_c, T_p) \\ &J = \int_{t_c}^{t_c+T_p} F(x, u)\,\mathrm{d}t + \Phi(x_{t_c+T_p}) \\ &\text{subject to} \quad \dot{x} = f(x, u) \\ &L \le \begin{bmatrix} x \\ u \end{bmatrix} \le U \end{aligned} \tag{6}$$

where x and u are, respectively, state vector and control sequence; t_c and T_p represent the control and the prediction horizon with $t_c \le T_p$, L and U are lower and upper bounds and δ is the predicted time step.

Fig. 5 shows the principle of the RCH [26], and in which the online optimisation is typically used to determine the future control strategy. At time t, the controller predicts the future dynamic behaviour of the system over a prediction horizon T_p and determines (over a control horizon T_c) the input, such that a predetermined open-loop performance objective function is optimised [12]. The whole procedure of prediction and optimisation is repeated to find a new input with the control and prediction horizons moving forward.

4.2 Dual-mode formation-controller design

In this sub-section, we will introduce the framework of the multiple UAVs formation flight controller. In our proposed formation flight control strategy, each UAV follows the virtual leader UAV.

The ith UAV state vector and control input sequence in (1) are

$$\mathbf{x}_i = (v_i, \Psi_i, x_i, y_i) \quad \mathbf{u}_i = (u_i, \ \omega_i) \tag{7}$$

The virtual leader state is x_{VL} and control inputs are $\mathbf{u}_{VL}$. According to the leader UAV, the ith UAV relative state is $\mathbf{x}_{ri} = \mathbf{x}_i - \mathbf{x}_{VL}$. We define formation state and input sequence as

$$\begin{aligned} &X = (\mathbf{x}_{VL}, \mathbf{x}_1, \ \ldots, \mathbf{x}_N) \quad X_r = (\mathbf{x}_{r1}, \ \ldots, \mathbf{x}_{rN}) \\ &U = (\mathbf{u}_{VL}, \mathbf{u}_1, \ \ldots, \mathbf{u}_N) \end{aligned} \tag{8}$$

In non-linear RHC, the input applied to the system is usually given by the solution of the following finite-horizon optimal control problem according to (6), which is solved at every

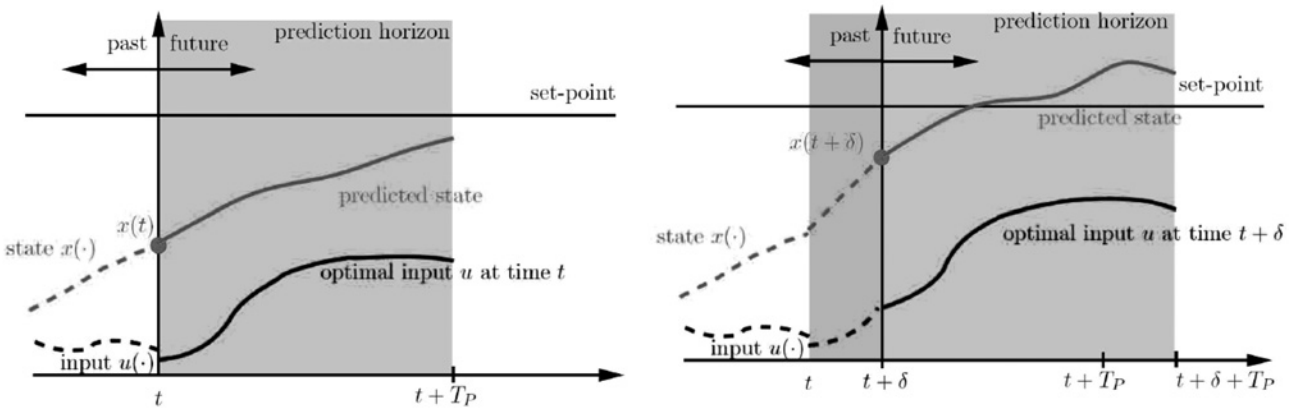

Figure 5 *RHC scheme*

a Current time $t_c = t$
b Current time $t_c = t + \delta$

doi: 10.1049/iet-cta.2009.0256

sampling instant

$$J = \min_{U} \int_{t}^{t+T_p} F(X_r(\tau), U(\tau))\,d\tau + V(X_r(t+T_p)) \quad (9)$$

subject to

$$\begin{aligned} &\dot{X}(\tau) = f(X(\tau),\ U(\tau)) \\ &U(\tau) \in U, \quad \forall \tau \in [t,\ t+t_c] \\ &X(\tau) \in X, \quad \forall \tau \in [t,\ t+T_p] \\ &X(t+T_p) \in \mathbf{\Omega} \end{aligned} \quad (10)$$

where

$$F(X_r, U) = X_r^T Q X_r + U^T R U \quad (11)$$

$$V(X_r(t+T_p)) = X(t+T_p)_r^T P X(t+T_p)_r \quad (12)$$

The deviation from the desired values is weighted by the positive-definite matrices Q, R and P; the time step is δ.

V is the terminal penalty.

$$\mathbf{\Omega} = \{X | X^T P X \le \alpha\} \quad (13)$$

The terminal region $\mathbf{\Omega}$ is chosen such that it is invariant for the non-linear system control by using a linear state feedback. As control systems become more complex and performance requirements more demanding, the invariant sets are widely employed to design stabilising controllers and, in particular, for applying RHC strategies. In order to enlarge the solution range and make search process of PSO easier, the dual-mode control strategy is chosen in this paper, as this strategy provides an efficient way to guarantee the stability of RHC with input constraints. The basic idea is to use a finite horizon of allowable control inputs to steer the state into an invariant set. The terminal region $\mathbf{\Omega}$ and terminal penalty matrix P can be determined offline [27].

A local linear control law that stabilises the non-linear system in $\mathbf{\Omega}$ is obtained as follows

$$\dot{X} = \frac{\partial f}{\partial x}(0,\ 0)X + \frac{\partial f}{\partial u}(0,\ 0)u \quad (14)$$

Substitute the linear state feedback $u = KX$ into (14), we can obtain

$$\dot{X} = \phi X \quad (15)$$

$$\phi = \frac{\partial f}{\partial x}(0,\ 0) + \frac{\partial f}{\partial u}(0,\ 0)K \quad (16)$$

Define the following Lyapunov equation

$$\phi P + P\phi^T + Q^* = 0 \quad (17)$$

where $Q^* = Q + K^T R K$ and the solution P is a positive-definite symmetric matrix. For any vector $X \in R^n$, $\|X\|$ denotes Euclidean norm. There exists a constant $\alpha \in (0,\ \infty)$ to fix the terminal region $\mathbf{\Omega}$ at the origin as (13). The constant α satisfies $KX \in \mathbf{U}$ for all $x \in \mathbf{\Omega}$ and the following condition, according to (13) and $u = KX$

$$\begin{aligned} \|KX\| &= \|K(\alpha^{1/2}P^{-1/2})(\alpha^{-1/2}P^{1/2}X)\| \\ &\le \|K(\alpha^{1/2}P^{-1/2})\| \cdot \|\alpha^{-1/2}P^{1/2}X\| \\ &\le \|K\alpha^{1/2}P^{-1/2}\| = \alpha K^T P K \end{aligned} \quad (18)$$

It follows from the input constraints that

$$\alpha K^T P K \le u_{max}^2 \quad \|u\| \le u_{max} \quad (19)$$

As multiple UAVs formation state X enter the terminal region $\mathbf{\Omega}$, X will be kept in this region all the while, and tends to the origin gradually.

4.3 Collision avoidance

In multiple UAVs formation flight system, each UAV moves in an environment in which there are obstacles and other UAVs. Thus, the multiple UAVs, at the same time, have to consider the problem of formation control and collisions avoidance. Collisions avoidance is assumed to be the most important task: only when UAV is at safe distance from the other UAVs and the obstacles, it can take care of maintaining the formation.

To achieve collision avoidance with other UAVs, a priority indexing scheme is used [28, 29]: all UAVs are tagged and the UAV with a lower index creates an imaginary obstacle around the UAV with a higher index (as seen in Fig. 6) and tries to avoid it. Thus, collision avoidance is achieved.

The UAVs with a lower index must react rapidly when neighbouring UAVs with a higher index approach within unsafe range or when obstacles are detected as they appear within the sensor range, to avoid any collision. Consequently, a multiple UAVs formation control strategy that ensures avoidance of collisions is achieved by adding a

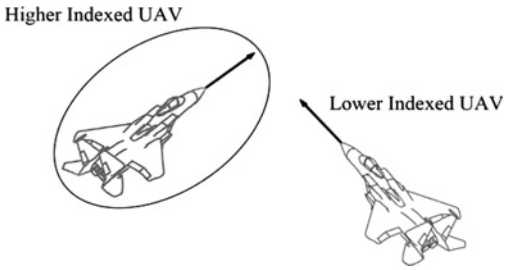

Figure 6 *Collision avoidance*

constraint to (9).

$$J_1 = J + \text{Penalty}_c \times \sum_{j=1}^{\text{Co}_{\text{num}}} \text{Dis}_{\text{constraint}}(d_{ij}) \tag{20}$$

$$\text{Dis}_{\text{constraint}}(d_{ij}) = \begin{cases} 1, & d_{ij} \le d_{\text{safe}} \\ 0, & d_{ij} > d_{\text{safe}} \end{cases} \tag{21}$$

where Penalty_c is the collision penalty coefficient, Co_{num} is the total account of collisions that UAV_i has to avoid, d_{ij} is the distance between UAV_i and the jth collision centre (The obstacle shape is specified as a circle). As long as UAV_i spatial horizon overlaps the jth obstacle, $\text{Dis}_{\text{constraint}}(d_{ij}) = 1$ and the value of (20) will be very large. Therefore this constraint is adopted for effectively avoiding collisions.

4.4 Chaotic particle swarm optimisation

In RHC, the cost function (20) plays a role of an evaluation function in PSO. The future control input sequence U is obtained by minimising (20) via a PSO. In PSO design, the optimisation concepts based on chaotic sequences can be a good alternative to provide diversity in PSO populations. The application of chaotic sequences instead of random sequences in PSO is a powerful strategy to diversify the population of particles and to improve the PSO's performance in preventing premature convergence to local minimum [24, 25]. Chaos optimisation is realised through chaos variables, which can be obtained by many ways. One of the simplest maps which is brought to the attention of scientists by May [30] in 1976, which appears in non-linear dynamics of biological population evidencing chaotic behaviour is logistic map

$$Z_{n+1} = \mu Z_n (1 - Z_n) \tag{22}$$

where Z_n is the nth chaotic number where n denotes the iteration number. Obviously, $Z_n \in (0, 1)$ under the conditions that the initial $Z_0 \in (0, 1)$, and $Z_0 \notin \{0.0, 0.25, 0.5, 0.75, 1.0\}$. $\mu = 4$ has been used in our algorithm.

The population diversity measured with the average particle distance [31], which describes population diversity with dispersion degree between particles [32]. Assume L is the maximum length of search space, ps is the population of particles, ln is the dimensions of solution space, p_{id} is the dth coordinate of particle p_i, P_d is the average of the dth coordinate, and the average particles distance $D(\text{iter}_k)$ at the kth iteration is defined as follows

$$D(\text{iter}_k) = \frac{1}{ps \times L} \sum_{i}^{ps} \sqrt{\sum_{d=1}^{ln} (p_{id} - P_d)^2} \tag{23}$$

In our CPSO approach, the PSO algorithm is first run to find the global best position as a candidate solution, and once particles collide $D(\text{iter}_k) < \varepsilon$, where ε is a positive constant. Then, the better solution generated from chaotic systems substitute random numbers for the PSO particles, where it is necessary to make a random-based choice. In this way, the global convergence can be improved and falling into local best solution can be prevented.

As mentioned in Section 3, PSO can also be improved by a modification of the inertia weight w in (4). The inertia weight can be used to balance the local and global search during the optimisation process. If the inertia weight is big, it is possible to enhance global search. Otherwise, smaller inertia weight will enhance the local search. While the value of w is made to decrease gradually with the increase in the number of iterations by the following equation at the kth iteration iter_k

$$w = w_{\max} - \text{iter}_k \frac{w_{\max} - w_{\min}}{\text{iter}_{\max}} \tag{24}$$

where $\text{iter}_{\max}$ is maximum iteration, $w_{\max}$ and $w_{\min}$ are separately maximum and minimum of w.

In order to guarantee the stability and enhance the efficiency of the control algorithm, the initial value of each particle choose the last control input sequence after the achievement of primary sequence, such as the lth particle at a time t is initialised by $U(t - \delta)$.

The process of our proposed non-linear dual-mode RHC method based on CPSO for solving multiple UAVs formation flight problem can be described as follows:

Step 1: Initialise UAVs states X_0 and the non-linear dual-mode RHC parameters used in formation system.

Step 2: Evaluate the terminal region $\mathbf{\Omega}$ and terminal penalty matrix P by (14)–(19).

Step 3: Detect if formation state $X(t)$ enters the terminal region $\mathbf{\Omega}$ or not by (13)? If it is true, then go to Step 8; else go to Step 4.

Step 4: Initialise particle swarm adopted with the last predictive control sequence $U(t - \delta)$ optimised by PSO, while particle swarm is initialised randomly at the first time.

Step 5: Evaluate the value of each particle by computing the cost function (20), update the particle swarm and the global best particle $g$best according to (2)–(5).

Step 6: Detect if PSO precociously convergence to local minima with (23)? If it is true, then go to next step; else go to Step 8.

doi: 10.1049/iet-cta.2009.0256

Step 7: Use chaotic systems to generate a better solution to substitute random numbers for the PSO particles in next iteration and then go to Step 4.

Step 8: Detect the PSO terminate conditions (reaching the maximal generation or finding the idea optimum). If the terminate conditions is met, end the

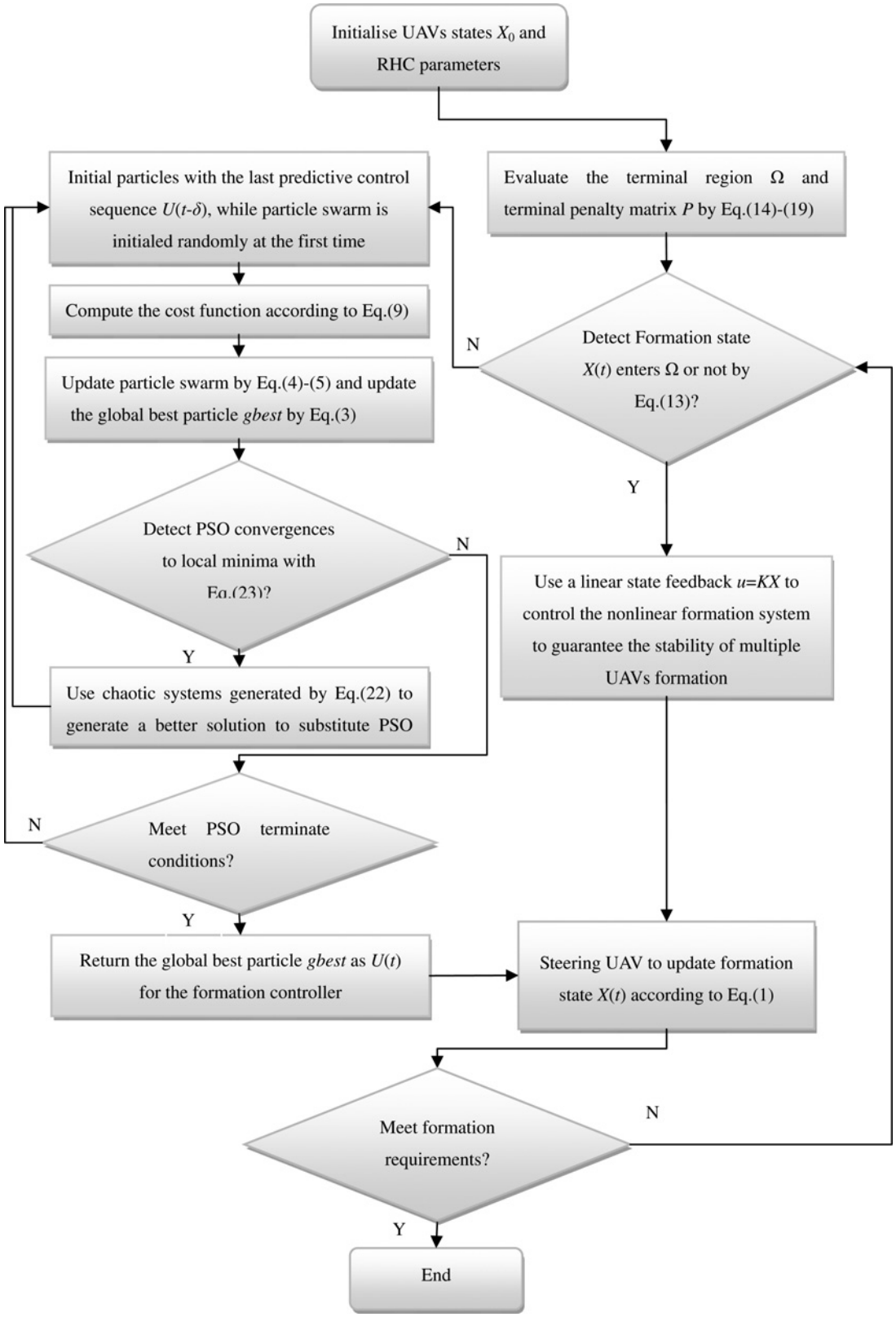

Figure 7 *CPSO-based RHC formation control scheme for multiple UAVs formation flight*

PSO algorithm and return the global best particle *g*best as the control sequence $U(t)$ or continue the computation.

Step 9: Apply the first part of the optimal control sequence to update formation state $X(t)$, and then go to Step 1.

Step 10: Use a linear state feedback $u = KX$ to control the non-linear formation system to guarantee the stability of multiple UAVs formation.

Step 11: Detect the formation stability conditions. If the certain conditions are achieved, end the formation control method or go to Step 2.

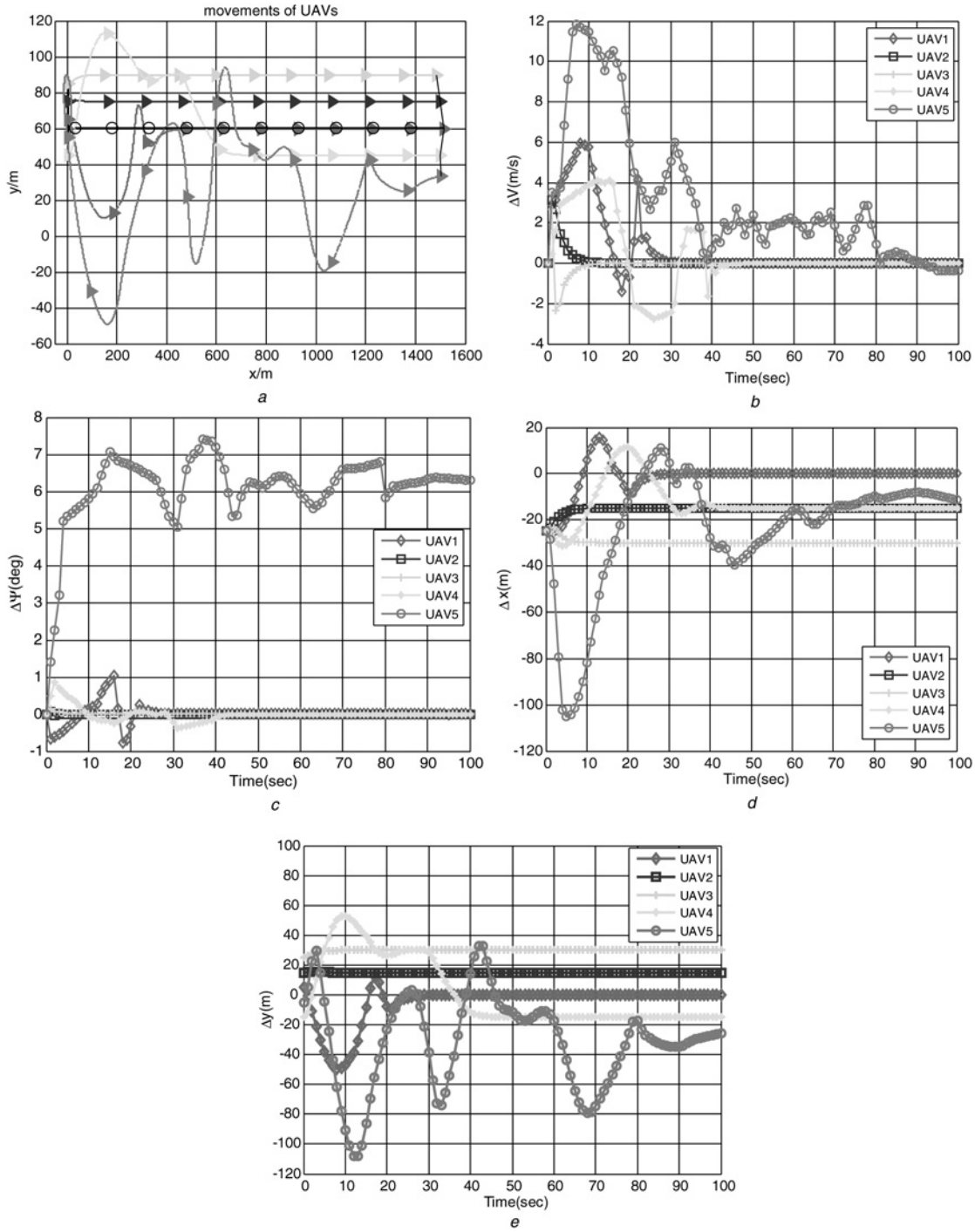

Figure 8 *Detailed results generated by the control sequence optimised by GA*

a Five UAVs merges to a V-formation while following a virtual leader
b Relative velocities of five UAVs
c Relative heading angles
d Relative distances in the *x*-direction
e Relative distances in the *y*-direction

IET Control Theory Appl., 2010, Vol. 4, Iss. 11, pp. 2565–2578

doi: 10.1049/iet-cta.2009.0256

Fig. 7 displays the flowchart of the CPSO-based non-linear dual-mode RHC formation control scheme for multiple UAVs formation flight.

5 Simulation and results

In this section, series experiments have been performed to investigate the performance of the proposed CPSO-based non-linear dual-mode RHC formation control scheme for multiple UAVs formation flight. We use (1) to represent states of UAV model, respectively. The multiple UAVs group consists of five agents, with input constraints $[-5, 5]$ m/s^2 for the acceleration u and $[-\pi/18, \pi/18]$ rad/s for the angular turn rate ω. To improve performance and avoid collisions, a safe distance between UAVs is defined, $d_{safe} = 3$. Collisions between UAVs are solved with the

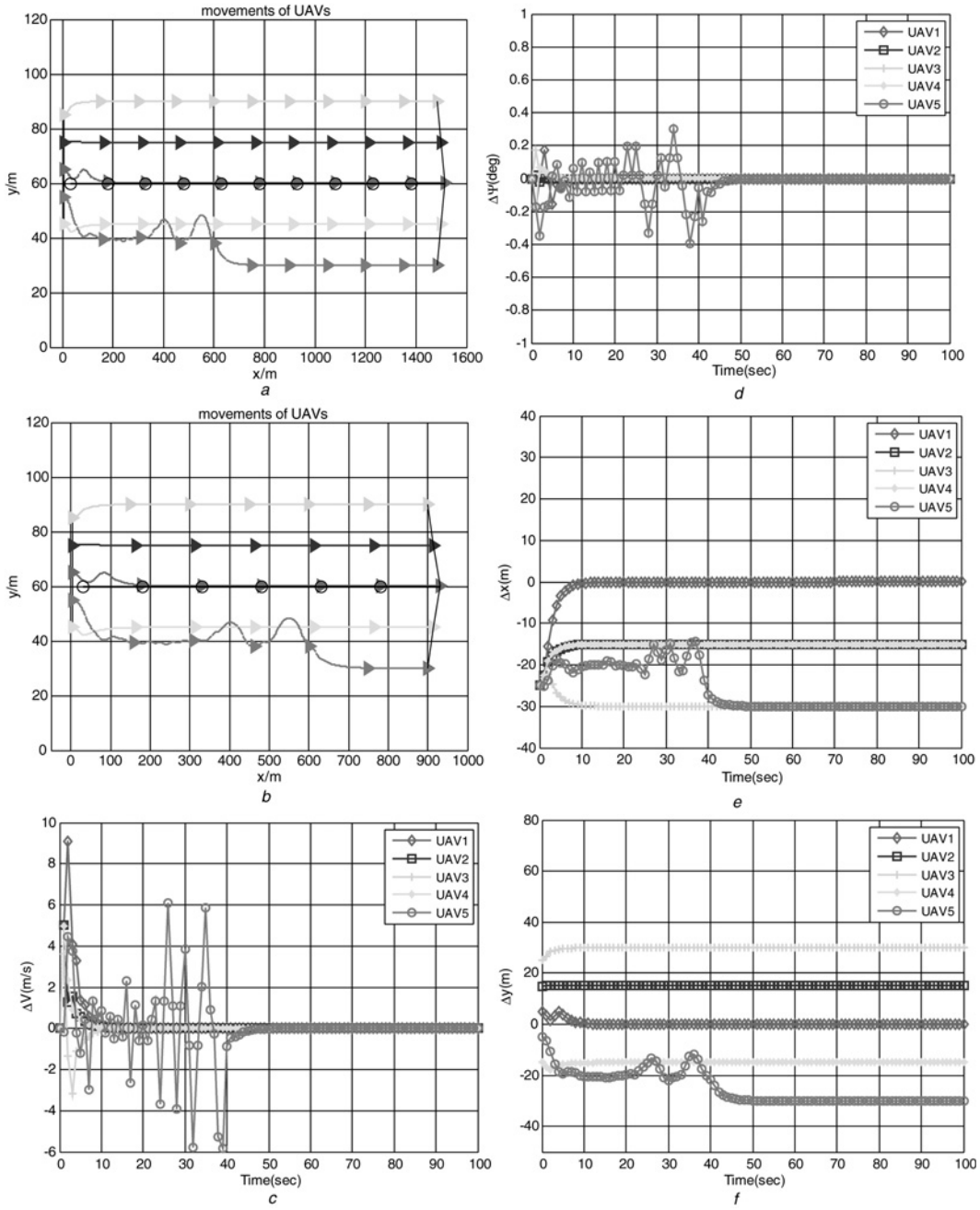

Figure 9 *Detailed results generated by the control sequence optimised by CPSO*

a Formation trajectory under the same initial conditions
b Formation trajectory during the first 60 time steps
c Relative velocities of UAVs
d Relative heading angles
e Relative distances in the *x*-direction
f Relative distances in the *y*-direction

priority index strategy. Each UAV is tagged with a serial number.

The initial conditions of the non-linear dual-mode formation controller are prediction horizon $T_p = 8$ s, time step $\delta = 1$ s (the simulation time), weighting matrices $Q = \text{diag}(1, 1, 1, 1)$, $R = \text{diag}(1, 1)$, $P = (0.0596\ 0\ -0.0063\ 0;\ 0\ 0.0596\ 0\ -0.0063;\ -0.0063\ 0\ 0.0129\ 0;\ 0\ -0.0063\ 0\ 0.0129)$. The improved PSO parameters setting is the size of the particle swarm $p_s = 20$, inertia weight

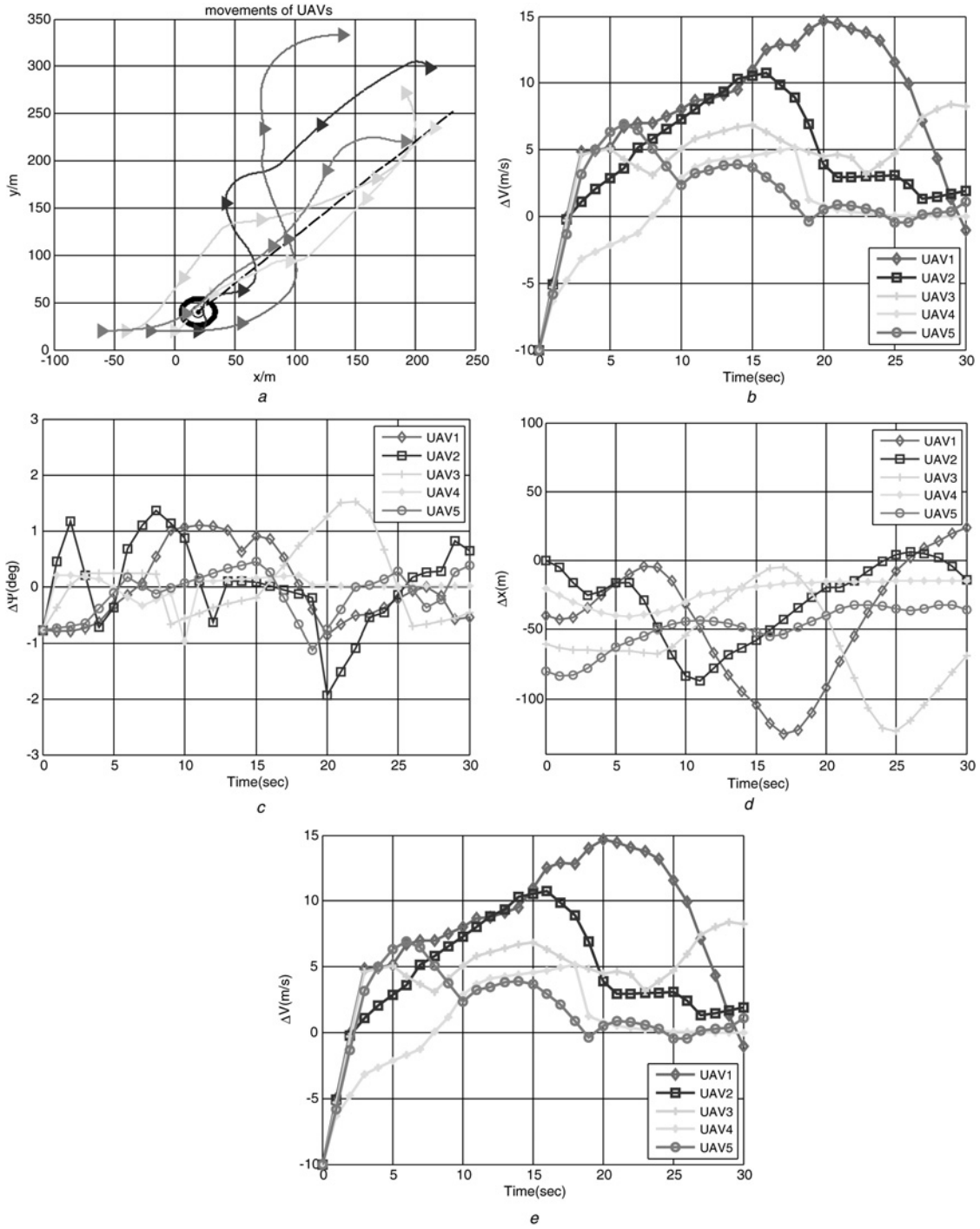

Figure 10 *Detailed results of GA for the second experiment*

a Formation trajectory in a complicated environment with obstacle
b Relative velocities of UAVs
c Relative heading angles
d Relative distances in the *x*-direction
e Relative distances in the *y*-direction

IET Control Theory Appl., 2010, Vol. 4, Iss. 11, pp. 2565–2578
doi: 10.1049/iet-cta.2009.0256

$w_{max} = 1.2$, $w_{min} = 0.1$, particle maximum velocity $vp_{max} = 4$, $c_1 = 0.5$, $c_2 = 0.5$ and the maximum iteration $iter_{max} = 200$.

In order to fully illustrate the efficiency of the proposed algorithm, we compare its performance with the standard GA under the same conditions. In both CPSO and GA, the population size is 20. The GA is real-valued with random initialisation, and updates the population and search for the optimum with random techniques. Crossover and mutation probabilities are set as 0.8 and $1/n$, respectively, where n is the dimension of the problem.

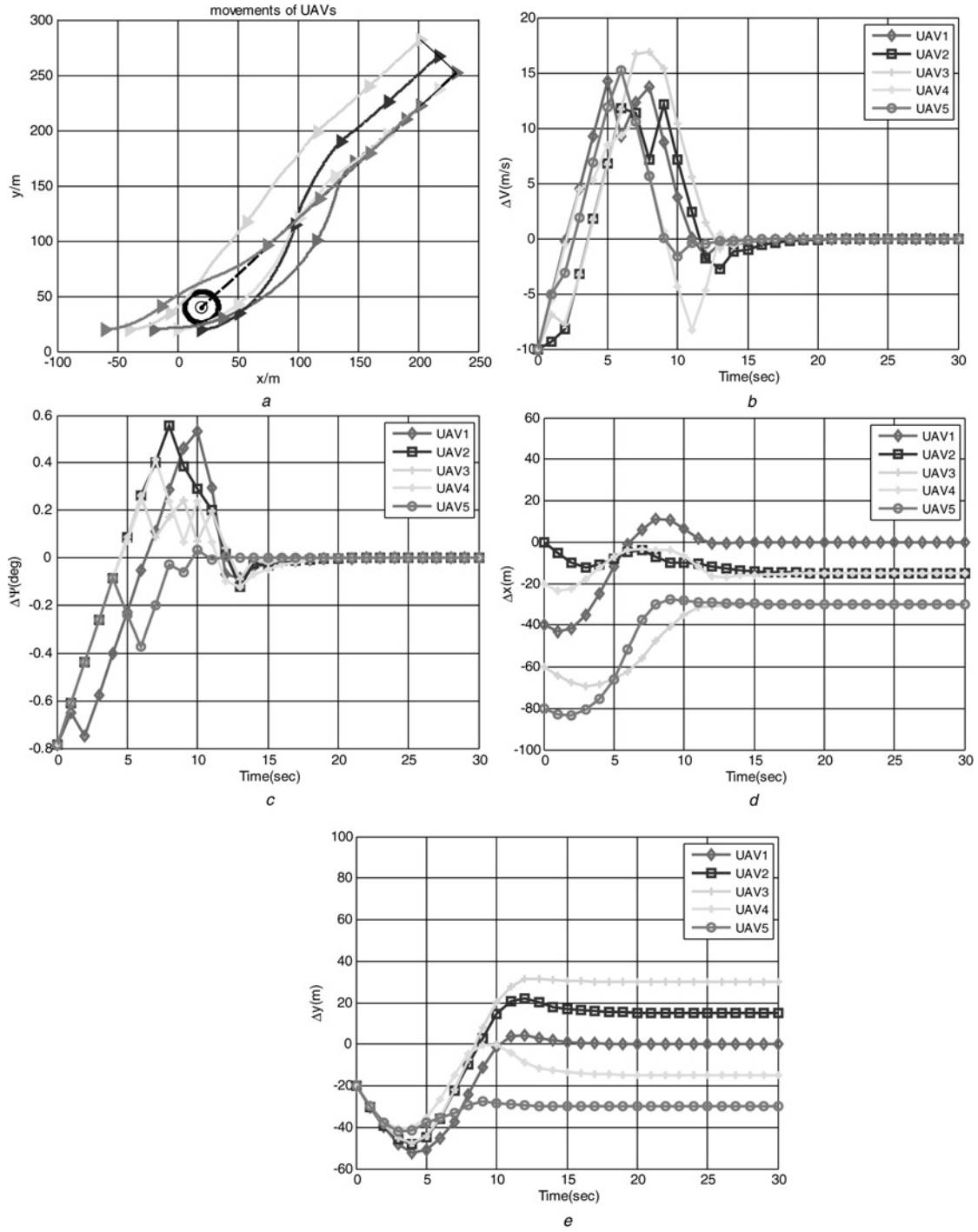

Figure 11 *Detailed results of CPSO for the second experiment*

a Formation trajectory in a complicated environment with obstacle
b Relative velocities of UAVs
c Relative heading angles
d Relative distances in the *x*-direction
e Relative distances in the *y*-direction

5.1 Formation control

Multiple UAVs states are initialised as $\mathbf{x}_{VL} = (15, 0, 30, 60)$, $\mathbf{x}_1 = (15, 0{,}5, 65)$, $\mathbf{x}_2 = (15, 0, 5, 75)$, $\mathbf{x}_3 = (15, 0, 5, 85)$, $\mathbf{x}_4 = (15, 0, 5, 45)$, $\mathbf{x}_5 = (15, 0, 5, 55)$, and the relative distances are $\mathbf{x}_{r1} = (0, 0, 0, 0)$, $\mathbf{x}_{r2} = (0{,}0, -15, 15)$, $\mathbf{x}_{r3} = (0, 0, -30, 30)$, $\mathbf{x}_{r4} = (0, 0, -15, -15)$, $\mathbf{x}_{r5} = (0, 0, -30, -30)$. The virtual leader is marked with '○', whereas UAV is '△'. The five UAVs reconfigure from a '|' initial shape to forming a 'V' formation. Assume that the virtual leader speed is 15 m/s in the x-direction and its heading angle is 0 rad during the simulations. Figs. 8 and 9 show the detailed results generated by the control sequence optimised by GA and CPSO, respectively.

The UAV group has to follow the virtual leader as seen in Fig. 8*a*, which shows the (x, y) positions of UAVs generated by using GA to optimise the control sequence. Note that the UAV group is travelling from left to right in the figure. The results are shown in Fig. 8, which illustrates that both the path tracking and formation maintenance tasks are not achieved. Figs. 8*b*–*e* show the multiple UAVs convergence time. It is obvious that the fifth UAV cannot move to the initial relative position to follow the virtual leader, while other UAVs can converge to designated position. The same experiment is tested five times with similar results that the satisfactory tasks are not achieved.

In Fig. 9, UAVs follow the virtual leader with a constant-velocity in x direction and the desired separation between UAVs is 15 m in both the x and y directions. Under the control inputs optimised by CPSO, the multiple UAVs converge to the desired formation from the same initial configuration. Formation results are presented in Fig. 9*a*. The formations converge in 50 s as seen in Fig. 9*b*. Compared with the results generated by the control sequence optimised by GA, CPSO performs better for its comprehensive ability to search and high precision, which demonstrates that CPSO is suitable for the dual-mode RHC controller.

5.2 Formation control with an obstacle

The second experiment illustrates the effectiveness of the proposed method in Section 4.3 for five UAVs performing obstacle avoidance. The parameters in the dual-mode RHC controller are chosen to be exactly the same as in the first experiment. The second experiment is initialised: $\mathbf{x}_{VL} = (15, 0.7854, 20, 40)$, $\mathbf{x}_1 = (5, 0, -20, 20)$, $\mathbf{x}_2 = (5, 0, 20, 20)$, $\mathbf{x}_3 = (5, 0, -40, 20)$, $\mathbf{x}_4 = (5, 0, 0, 20)$, $\mathbf{x}_5 = (5, 0, -60, 20)$, and the relative distances are $\mathbf{x}_{r1} = (0, 0, 0, 0)$, $\mathbf{x}_{r2} = (0, 0, -15, 15)$, $\mathbf{x}_{r3} = (0, 0, -30, 30)$, $\mathbf{x}_{r4} = (0, 0, -15, -15)$, $\mathbf{x}_{r5} = (0, 0, -30, -30)$. Also, we change the virtual leader's heading angle from 0 to $0.7854(\pi/4)$ and add a circular obstacle in the simulation environment. The obstacle centre is located at (20, 40) and the radius is 12 m. Fig. 10 shows the results generated by the control sequence optimised by GA, whereas Fig. 11 shows the results generated by the proposed algorithm.

The optimal obstacle avoidance trajectory with the dual-mode controller is generated assuming that UAVs can sense the circle obstacle. The trajectory of five UAVs after 20 time steps is presented in Figs. 10*a* and 11*a*. The results generated by control sequence optimised by GA as seen in Figs. 10*b*–*e* are far from satisfactory. Fig. 11*a* shows the formation trajectory after 20 time steps. The results are shown in Figs.11*c* and *d*, which demonstrates that both the path tracking and formation maintenance tasks are successfully achieved. Figs. 11*c*–*e* represent the multiple UAVs convergence time. It is obvious that it converges in 20 s. It clearly shows the superiority of the proposed algorithm over GA.

The simulation results obtained by applying the proposed dual-mode RHC algorithm show that the UAVs are capable of flying in the desired formation. Simulations with different conditions are conducted to verify the feasibility and effectiveness of the proposed controller.

6 Conclusion

In this paper, we have described multiple UAVs formation flight using a non-linear dual-mode RHC based on CPSO. The attractive aspect of non-linear RHC is the ability to incorporate state and control as hard or soft constraints in the optimisation formulation and utilise future information to generate an optimal control sequence at each time step. However, the two major disadvantages of non-linear RHC are a control stability problem and high computational time. During our implementation, we employ a dual-mode strategy with the terminal region to guarantee the stability of non-linear RHC and we have shown that non-linear RHC is a feasible control approach in our simulations. The convergence of the RHC optimisations time problem is very well solved by the proposed CPSO, which can ensure that a dual-mode non-linear RHC is accepted as a reliable formation control algorithm.

The proposed method was applied to five UAVs formation flight experiments. The first case concerns the control of a set of UAVs to achieve the desired formation shape by tracking the virtual leader. Each follower UAV adjusts states relative to the virtual leader. The second case concerns the control of UAVs to avoid an obstacle during the progress of dynamic reconfiguration. In this case, the dual-mode non-linear RHC approach does consider the obstacle information to ensure the UAVs' security. Consequently, both presented cases illustrate the commendable ability of the controller to meet the formation requirements from different initialisations.

Our future work will focus on applying the new approach proposed in this paper to real-world mobile robots formation. In a real-time application, it is difficult to neglect solution times for the optimisation algorithm that must be run at every step. However, with faster computers and improved technology, the proposed approach can be a good candidate

doi: 10.1049/iet-cta.2009.0256

for real-time formation control of robots under more complicated environments.

7 Acknowledgments

This work was supported by the National High Technology Research and Development Program of China (863 Program), Natural Science Foundation of China (NSFC) under grant #60975072 and #60604009, Aeronautical Science Foundation of China under grant #2006ZC51039, Beijing NOVA Program Foundation of China under grant #2007A017 and 'New Scientific Star in Blue Sky' Talent Program of Beihang University of China. The authors are grateful to the anonymous referees for their valuable comments and suggestions for the better presentation of this paper.

8 References

[1] GIULIETTI F., POLLINI L., INNOCENTI M.: 'Autonomous formation flight', *IEEE Control Syst. Mag.*, 2000, **20**, (6), pp. 34–44

[2] BINETTI P., ARIYUR K.B., KRSTIC M., BERNELLI F.: 'Formation flight optimization using extremum seeking feedback', *J. Guid. Control Dyn.*, 2003, **26**, (1), pp. 132–142

[3] PACHTER M., D'AZZO J.J., PROUD A.W.: 'Tight formation flight control', *J. Guid. Control Dyn.*, 2001, **24**, (2), pp. 246–254

[4] DONG W., FARRELL J.A.: 'Formation control of multiple underactuated surface vessels', *IET Control Theory Appl.*, 2008, **2**, (12), pp. 1077–1085

[5] SHAN J.: 'Six-degree-of-freedom synchronised adaptive learning control for spacecraft formation flying', *IET Control Theory Appl.*, 2008, **2**, (10), pp. 930–949

[6] STIPANOVI D.M., INALHAN G., TEO R., TOMLIN C.J.: 'Decentralized overlapping control of a formation of unmanned aerial vehicles', *Automatic*, 2004, **40**, pp. 1285–1296

[7] LAVRETSKY E.: 'F/A-18 autonomous formation flight control systems design'. Proc. AIAA Guidance, Navigation, and Control Conf. Exhibit, August 2002, paper no. AIAA-2002–4757

[8] ZOU Y.F., PAGILLA P.R., RATLIFF R.T.: 'Distributed formation flight control using constraint forces', *J. Guid. Control Dyn.*, 2009, **32**, (1), pp. 112–120

[9] DAS A.K., FIERRO R., KUMAR V., OSTROWSKI J.P., SPLETZER J., TAYLOR C.J.: 'A vision-based formation control framework', *IEEE Trans. Robot. Autom.*, 2002, **18**, (5), pp. 813–825

[10] DUNBAR W.B., MURRAY R.M.: 'Distributed receding horizon control for multi-vehicle formation stabilization', *Automatica*, 2005, **42**, pp. 549–558

[11] KEVICZKY T., BORRELLI F., FREGENE K., GODBOLE D., BALAS G.J.: 'Decentralized receding horizon control and coordination of autonomous vehicle formations', *IEEE Trans. Control Syst. Technol.*, 2008, **16**, (1), pp. 19–33

[12] WESSELOWSKI K., FIERRO R.: 'A dual-mode model predictive controller for robot formations'. Proc. 42nd IEEE Conf. on Decision and Control, Hawaii, USA, 2003, pp. 3615–3620

[13] KWON W.H., HAN S.: 'Receding horizon control: Model predictive control for state models' (Springer, London, 2005)

[14] FINDEISEN R., FINDEISEN F.: 'An introduction to nonlinear model predictive control'. Proc. 21st Benelux Meeting Systems and Control, Veldhoven, The Netherlands, 2002

[15] KENNEDY J., EBERHART R.C.: 'Particle swarm optimization'. Proc. 1995 IEEE Int. Conf. on Neural Networks, vol. 4, pp. 1942–1948

[16] SHI Y., EBERHART R.C.: 'Parameter selection in particle swarm optimization'. Proc. Seventh Annual Conf. on Evolutionary Programming, New York, 1998, pp. 591–600

[17] SHI Y., EBERHART R.C.: 'Comparison between genetic algorithms and particle swarm optimization'. Proc. Seventh Annual Conf. on Evolutionary Programming, New York, 1998, pp. 611–616

[18] LESLEY A.W., HURTADO J.E.: 'Decentralize cooperative-control design for multivehicle formations', *J. Guid., Control Dyn.*, 2008, **31**, (4), pp. 970–979

[19] SUN Z., LI B., CAI W.C., LIAO X.H., SONG Y.D.: 'Virtual leader based robust adaptive formation control of multi-unmanned ground vehicles (UGVs)'. Proc. 2007 American Control Conf., New York, USA, 2007, pp. 1876–1881

[20] PRICE C.R.: 'The virtual UAV leader'. Proc. AIAA Infotech@Aerospace 2007 Conf. and Exhibit, Rohnert Park, California, 2007, paper no. AIAA-2007–2785

[21] SHI Y., EBERHART R.C.: 'A modified particle swarm optimizer'. Proc. IEEE World Conf. on Computation Intelligence, Anchorage, 1998, pp. 69–73

[22] http://210.60.61.21/Course/95_1/CI/Lecture/Lecture%207-%20PSO.ppt

[23] EBERHART R.C., SHI Y.: 'Particle swarm optimization: developments, applications and resources'. Proc. 2001 Congress on Evolutionary Computation, 2001, pp. 81–86

[24] ALATAS B., AKIN E., OZER A.B.: 'Chaos embedded particle swarm optimization algorithms', *Chaos Solutions Fractals*, 2009, **40**, pp. 1715–1734

[25] LIU B., WANG L., JIN Y.H., TANG F., HUANG D.X.: 'Improved particle swarm optimization combined with chaos', *Chaos Solutions Fractals*, 2005, **25**, pp. 1261–1271

[26] http://www.ee.unimelb.edu.au/people/mcrotk/courses/notes/finc_ln_20060419.pdf

[27] CHEN H., ALLGOWER F.A.: 'Quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability', *Automatica*, 1998, **14**, (10), pp. 1205–1217

[28] WANG X.H., YADAV V., BALAKRISHNAN S.N.: 'Cooperative UAV formation flying with obstacle/collision avoidance', *IEEE Trans. Control Syst. Technol.*, 2007, **15**, (4), pp. 672–679

[29] GAGNON E., RABBATH C.A., LAUZON M.: 'Receding horizons with heading constraints for collision avoidance'. Proc. AIAA Guidance, Navigation, and Control Conf. and Exhibit, San Francisco, California, 2005, paper no. AIAA-2005–6369

[30] MAY R.M.: 'Simple mathematical models with very complicated dynamics', *Nature*, 1976, **261**, pp. 459–467

[31] DANILO F.C., CARMELO J.A.B.: 'Clan particle swarm optimization', *Int. J. Intell. Comput. Cybern.*, 2009, **2**, (2), pp. 197–227

[32] KRINK T., VESTERSTORM J.S., RIGET J.: 'Particle swarm optimization with spatial particle extension'. Proc. 2002 IEEE Congress on Evolutionary Computation (CEC), Honolulu, 2002, pp. 1474–1479

doi: 10.1049/iet-cta.2009.0256