Journal of Intelligent & Robotic Systems (2019) 93:17–32
https://doi.org/10.1007/s10846-018-0802-z

Collision-Free Trajectory Generation and Tracking for UAVs Using Markov Decision Process in a Cluttered Environment

Xiang Yu[1] · Xiaobin Zhou[2] · Youmin Zhang[1]

Received: 14 October 2017 / Accepted: 21 February 2018 / Published online: 8 March 2018

Abstract

A collision-free trajectory generation and tracking method capable of re-planning unmanned aerial vehicle (UAV) trajectories can increase flight safety and decrease the possibility of mission failures. In this paper, a Markov decision process (MDP) based algorithm combined with backtracking method is presented to create a safe trajectory in the case of hostile environments. Subsequently, a differential flatness method is adopted to smooth the profile of the rerouted trajectory for satisfying the UAV physical constraints. Lastly, a flight controller based on passivity-based control (PBC) is designed to maintain UAV's stability and trajectory tracking performance. Simulation results demonstrate that the UAV with the proposed strategy is capable of avoiding obstacles in a hostile environment.

Keywords Collision-free · Differential flatness · Markov decision process (MDP) · Passivity-based control (PBC) · Unmanned aerial vehicle (UAV)

1 Introduction

Unmanned aerial vehicle (UAV), capable of operating autonomously, is a kind of powered flight vehicle without a crew on board [1–3]. Compared with manned aircraft, UAV has vast advantages including lower cost, fewer required personnel, and less support needs. The above-mentioned merits make UAV be of great potential in both civilian and military applications.

However, UAVs have pushed the use of airspace to its limit, manifesting in mid-air collision problem. To guarantee a safe flight and also to integrate UAVs into the National Airspace System, UAVs must possess the capability of sensing and avoiding potential obstacles properly [4]. Therefore, recent research has strengthened the importance of providing an efficient collision-free strategy merging UAVs into the sky.

Normally, a sense and avoid (S&A) system is composed of sensing hardware, decision mechanism, trajectory plan, and flight controller [5], respectively. The sensing hardware such as an inertial navigation system and airborne sensors can gather the information of UAV and intruder. Airborne sensors can be generally grouped into two categories: cooperative sensors (e.g., Traffic Alert and Collision Avoidance System, Automatic Dependent Surveillance Broadcast) [5] and non-cooperative sensors (e.g., airborne sensor, acoustic system, electro-optical system) [6]. According to the information transmitted from the sensing hardware, the decision mechanism predicts the distance between the UAV and intruder for a period of time by advancing the UAV state into future. If the predicted magnitude of the miss distance is smaller than the safety separation, a collision will take place with high possibility between aircraft [7]. Then the mechanism makes the decision to re-plan a safe trajectory. Consequently, the trajectory planner should generate a collision-free trajectory under the constraints of UAV dynamics and fuel economics. In view of UAV

✉ Youmin Zhang
youmin.zhang@concordia.ca

Xiang Yu
xiangyu1110@gmail.com

Xiaobin Zhou
zxb1992@hnu.edu.cn

[1] Department of Mechanical, Industrial and Aerospace Engineering, Concordia University, Montreal, Quebec H3G 1M8, Canada

[2] College of Mechanical and Vehicle Engineering, Hunan University, Changsha, China

aerodynamics, the flight controller needs to adjust actuators accomplishing avoidance maneuvers [8].

The existing methods can be basically divided into five categories: artificial potential field (APF), numerical optimization approaches (NOAs), artificial heuristic approaches (AHAs), graph search approaches (GSAs), and Markov decision process (MDP), respectively.

The APF is originally proposed in [9] to evade collisions in real-time. Within this context, moving obstacles and forbidden regions are modeled by repulsive potential, while the destination is regarded as attractive potential. As a consequence, the UAV moves as a particle reacting to the potential field [10]. A local planner based on APF is developed in [11], by which paths going through narrow areas of free space can be found to guarantee flight safety. However, local minimum may be induced due to the superposition potential effect including the goal potential and obstacles potential. Specifically, when attractive force and repulsive force reach a balance, the UAV will trap in a state instead of the goal.

Nonlinear programming method [12], Pontryagin's minimum principle [13], and mixed integer linear programming [14] can be thought as NOAs. With respect to NOAs, UAV trajectory generation is regarded as a numerical optimization problem, taking UAV aerodynamics constraints and cost function into account. By combining direct collocation with nonlinear programming, a UAV trajectory is produced through approximating the system states with piecewise polynomials [12]. By solving a nonlinear optimal control problem with path constraints [13], a UAV trajectory for avoiding the forbidden region is created correspondingly. The major drawback of NOAs lies in that the computation complexity increases substantially as the amount of aerodynamic constraints increasing.

Genetic algorithm [15, 16], particle swarm optimization (PSO) [17–19], artificial bee colony optimization [20], and biogeography-based optimization [21] are seen as AHAs. Such methods receive considerable attention owing to the convenience and flexibility in planning UAV trajectory. Given a three-dimensional (3D) environment, a collision-free flight trajectory is calculated using modified genetic algorithm and B-spline curves in [15]. PSO is a population-based stochastic search method suitable for trajectory optimization [17]. Depending on the behavior of bird flocks while searching for food, this method is efficient to determine trajectory safety for a UAV [18]. Generally, AHAs is quite advantageous in finding the global optimal solution with great numerical accuracy. However, it is sensitive to parameters and possibly falls into local optima which cannot guarantee the further convergence as the calculation range increases.

GSAs include Voronoi diagram [22, 23], visibility graph [24], and probabilistic roadmap method (PRM) [25, 26], respectively. With employment of GSAs, the free configuration space (flight environment) is formulated as a network diagram. Path planning problem is thereby recast into generating a safe trajectory in the network diagram. PRM is the most typical algorithm which takes random samples from the configuration space and checks their reasonability. A new motion planning is presented by constructing a probabilistic roadmap in a static environment [25]. Despite that it is convenient for GSA to formulate obstacles and targets in a network diagram, the algorithm's convergence rate slows down remarkably in the case of large-scale planning scenarios.

The underlying idea of MDP is to provide a mathematical framework for decision making in discrete time stochastic control process. The optimization problem is resolved by dividing the entire problem into smaller sub-problems until a simple case is reached [27]. The goal of MDP in path planning to find an optimal policy is formulated as deriving the maximal accumulation reward by choosing corresponding actions [28]. Based on the negotiations between different UAVs in a team with the general goal [29], an optimal path of avoiding intruders is generated through MDP. By formulating the collision avoidance problem as an MDP, the problem solving logic is allowed to generate in regard to several specified performance metrics [30]. In order to execute the mission of UAV team persistent search and make decision under uncertainty [31], a decentralized learning framework is developed by virtue of MDP. A new development of collision avoidance strategy based on MDP [32] is presented to ensure the UAV avoid multiple intruders autonomously. A variety of features, including UAV physical constraints, collision avoidance, and wind disturbance, can be incorporated into the framework, which demonstrates the power and flexibility of MDP.

Several levels of success have been achieved in trajectory planning, the UAV based on the aforementioned strategies can avoid static or moving obstacles in a relative simple environment. Current research focus is to navigate UAV in an unknown cluttered environment with moving obstacles. As for part of the conventional methods, the collision avoidance problems are resolved through regarding the moving obstacles as static collision area: 1) discrete the time into single moment; 2) by assuming the velocities of the moving obstacles and UAV are constant, the collision area can be predicted; and 3) design a new trajectory to evade the predicted static collision area. However, the velocity of moving intruder may change in real time which degrades the collision avoidance performance. Additionally,

many kinds of path planning strategies may result in failure of collision avoidance due to local minimum. Lastly, the dynamic constraints of UAV should be maintained within the safety bounds when planning the collision-free trajectory.

This study aims at proposing a collision-free scheme for a UAV, where the aerodynamics constraints, target tracking, local minimum, and path following are explicitly taken into account. An evolutionary strategy is developed on the basis of our previous work [32], and the unique property consist of three aspects:

1. In our previous work, the collision avoidance for moving obstacles is completed by predicting the collision position and then avoiding the static position which restricts the collision avoidance flexibility. This paper expands on existing MDP by introducing time into coordinate space that allows to show the location of the obstacles in a time-space framework. Within the proposed scheme, the UAV can keep away from the moving obstacles in a hostile environment which demonstrates the robustness of the strategy.
2. Local minimum is a severe problem when utilizing MDP to plan trajectory and not resolved in previous works. Considering that local minimum problem may lead to the task failure, backtracking method is integrated with MDP to guarantee a safe flight. Moreover, owing to the developed time-space framework, the UAV is capable of performing tasks of target tracking with the aid of mounted sensors.
3. The physical constraints including maximal roll angle and yaw rate should subject to performance criteria. Hence, a differential flatness approach is employed to avoid aggressive maneuvers and refine the profile of the collision-free trajectory produced by MDP. Moreover, a flight controller using the passivity-based control (PBC) technique is applied to control aileron and rudder deflections. In consequence, the flight trajectory can be followed and avoidance maneuvers can be achieved.

The remainder of this paper is organized as follows. Section 2 includes the problem statement, UAV lateral model, and objectives of this paper. The algorithm of collision prediction and avoidance is presented in Section 3, where the MDP, policy iteration, and differential flatness method are incorporated. Aircraft controller based on PBC technique is designed in Section 4 to stabilize both roll and yaw motions. In Section 5, simulations are conducted to evaluate the proposed algorithm. Finally, the concluding remarks are given in Section 6.

Nomenclature

V	Flight velocity
δ_a	Deflections of the aileron
δ_r	Deflections of the rudder
$\tilde{\Upsilon}$	Tracking errors
Υ_a	Actual angles
Υ_r	Reference angles
$\dot{\theta}_{\min}$	Allowable minimum pitch rate
$\dot{\theta}_{\max}$	Allowable maximum pitch rate
θ	Pitch angle
ψ	Yaw angle
$\dot{\psi}_{max}$	Maximum yaw angle rate
Φ	Roll angle
ϕ_{max}	Maximum roll angle
L	Lift force
W	Gravitational force
g	Gravity acceleration
n_y	Normal overload

2 UAV Model and Problem Statement

2.1 Problem Description

In this paper, a trajectory planning strategy is developed to navigate a UAV from a start point to destination without collision. As shown in Fig. 1, the flight environment is divided into multiple grids. As the UAV is assumed to fly in a horizontal plane, the environment is two dimensional. Therefore, the key point of the research is to find a feasible trajectory $\vec{l}$ to navigate the UAV bypassing obstacle and arriving at the destination.

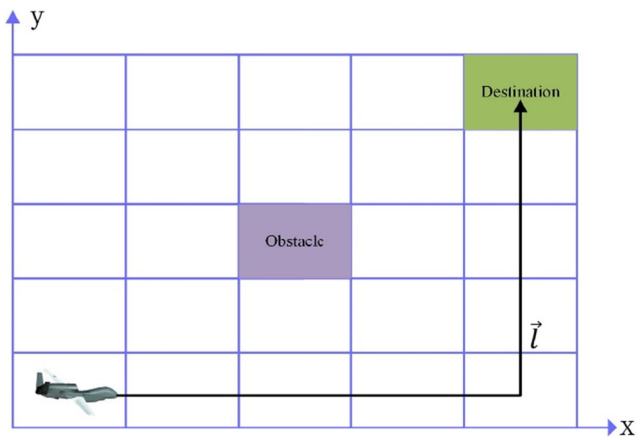

Fig. 1 Illustration of the design problem

2.2 UAV Kinematics Model and Dynamic Model

Generally, the flight velocities relative to the earth coordinate system can be determined as:

$$\begin{cases} \frac{dx}{dt} = V\cos\theta\cos\psi \\ \frac{dy}{dt} = V\cos\theta\sin\psi, \\ \frac{dz}{dt} = V\sin\theta \end{cases} \tag{1}$$

where x, y, and z represent the mass position of UAV with respect to the earth coordinate system, respectively [33]. θ, ψ, and ϕ indicate the pitch angle, yaw angle, and roll angle, which can be referred to Nomenclature table as well. In this study, the collision avoidance problem for a UAV is solved by making a turning maneuver in Cartesian plane (x, y), thus $\theta = 0$. This study focuses on the coordinated turn since it is more efficient for the aileron and the rudder working together to change the aircraft direction. As opposed to bank turn, an advantage of coordinated turn is that the adverse yaw motion resulting from aileron deflections and the induced drag can be counteracted. Besides, the sideslip angle and side force are zero in a coordinated turn. As illustrated in Fig. 2, the relationship of the three attitude angles is described as:

$$\frac{d\psi}{dt} = \frac{g n_y \sin\phi}{V\cos\theta}. \tag{2}$$

When the UAV makes a coordinated turn with constant velocity, n_y can be thereby written as:

$$n_y = \frac{1}{\cos\phi}. \tag{3}$$

Consequently, Eq. 2 can be expressed as:

$$\frac{d\psi}{dt} = \frac{g\tan\phi}{V}. \tag{4}$$

In order to design the flight controller and follow the collision-free path, the aircraft dynamics should be modeled apart from the kinematics model. With respect to the coordinated turn, there exist both roll and yaw. Hence, the rotational dynamics is represented as:

$$\begin{cases} I_x\ddot{\phi} + \frac{\partial F}{\partial\dot{\phi}} = M_{\phi,\delta_a}\delta_a + M_{\phi,\delta_r}\delta_r \\ I_z\ddot{\psi} + \frac{\partial F}{\partial\dot{\psi}} = M_{\psi,\delta_a}\delta_a + M_{\psi,\delta_r}\delta_r \end{cases}, \tag{5}$$

where $I = \begin{bmatrix} I_x & 0 \\ 0 & I_z \end{bmatrix}$ denotes the aircraft inertial relative to the vehicle-carried frame. The deflection amplitudes are bounded by:

$$\begin{cases} -25^\circ < \delta_a < 25^\circ \\ -30^\circ < \delta_r < 30^\circ \end{cases}. \tag{6}$$

$M = \begin{bmatrix} M_{\phi,\delta_a} & M_{\phi,\delta_r} \\ M_{\psi,\delta_a} & M_{\psi,\delta_r} \end{bmatrix}$ is the input matrix and $F(\phi, \psi)$ is the Rayleigh dissipation function due to aerodynamic forces of the UAV [8]. Furthermore, the partial differential of the Rayleigh dissipation function ($\frac{\partial F}{\partial\dot{\phi}}$, $\frac{\partial F}{\partial\dot{\psi}}$) and the products of input matrix with deflections such as $M_{\phi,\delta_a}\delta_a$ are relevant to drag, lift, and side force. The details of such variables are provided in Appendix A.

2.3 Objectives

To achieve the goal of navigating a UAV to evade intruders, the following steps must be completed.

1. Predict the relative distance between the UAV and the intruder for a period of time, thus determining the possibility of collision;
2. Re-plan a feasible collision-free trajectory based on MDP and differential flatness method to evade the intruders; and
3. Design the flight controller to follow the collision-free trajectory.

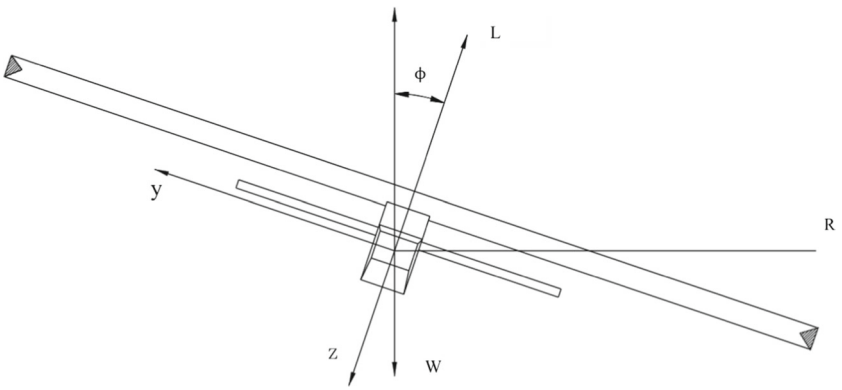

Fig. 2 Illustration of a coordinated turn

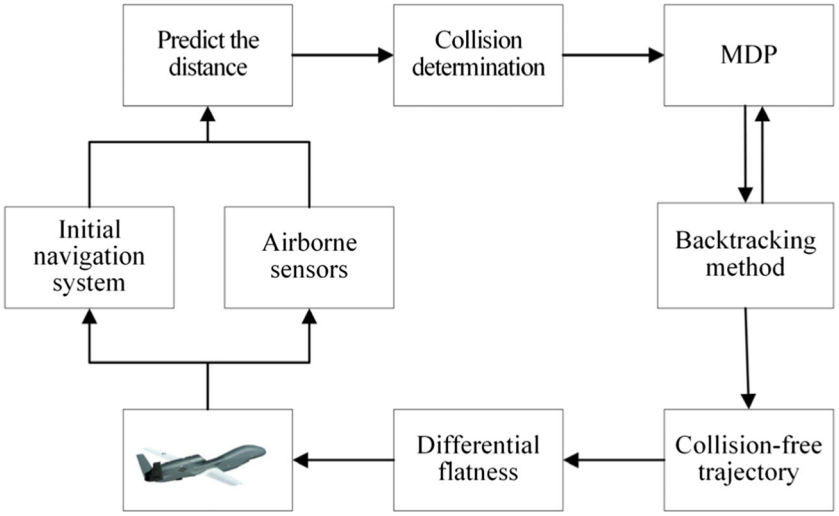

Fig. 3 Illustration of collision prediction and avoidance strategy

3 Collision Prediction and Avoidance Strategy

An assumption is made before presenting the design strategy.

Assumption 1 *The states of the entire environment are observable, which is essential for MDP* [28].

A collision avoidance problem is formulated as an MDP based on Assumption 1. As shown in Fig. 3, the design procedure of the proposed strategy can be divided into three portions.

1. With the information of velocity and position from sensing hardware, the minimum relative distance between the UAV and intruders can be calculated, based on which whether a collision will take place or not can be determined.
2. If a collision is determined, a collision-free trajectory needs to be planned by the proposed improved MDP, where backtracking method is utilized for addressing local minimum problem.
3. With the consideration of UAV physical constraints, a differential flatness method is exploited to guarantee the yaw angle and roll angle within the allowable ranges, contributing to a feasible flight.

3.1 Collision Prediction

As depicted in Fig. 4, a UAV is flying to the destination from the current position while another aircraft heads toward the currently designed trajectory of UAV. When these two aircrafts approach, whether a collision will take place or not can be decided by calculating the minimum distance. If the predicted minimum distance exceeds the safety separation, then a collision will occur with high possibility.

The predicted minimum distance vector $\vec{d}_m$ about the intruder with respect to UAV can be defined as:

$$\vec{d}_m = \hat{V}_r \times \left(\vec{d} \times \hat{V}_r\right), \tag{7}$$

where $\vec{d}$ and $\hat{V}_r$ are the relative distance vector and the unit relative velocity vector of intruder with respect to the UAV $\vec{V}_r$, respectively. $\hat{V}_r$ and $\vec{V}_r$ can be expressed as:

$$\hat{V}_r = \frac{\vec{V}_r}{\left\|\vec{V}_r\right\|}. \tag{8}$$

$$\vec{V}_r = \vec{V}_2 - \vec{V}_1, \tag{9}$$

where $\vec{V}_1$ and $\vec{V}_2$ are the velocities of UAV and intruder, respectively. When the following condition can be satisfied, a collision will occur:

$$\left|\overrightarrow{d_m}\right| \leq r_{\text{safe}}, \tag{10}$$

where r_{safe} indicates the safety separation.

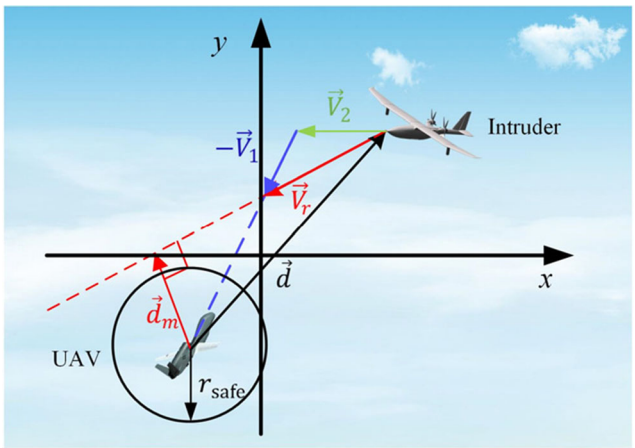

Fig. 4 The calculation of the minimum distance between the UAV and intruder

3.2 MDP-Based Collision Avoidance Strategy

The optimal trajectory is formed using an MDP-based approach. MDP is well suited to the problem due to its power and flexibility to directly incorporate obstacles and UAV physical constraints into its framework. The trajectory generation strategy is detailed in the following three subsections.

3.2.1 The Structure of MDP

The collision avoidance system aims to build a transparent learning frame from interaction between actions and environment to achieve an optimal policy. A finite MDP is composed of four components, including state S, rewards $R^a_{ss'}$, action set $A(s), s \in S$, and transition probabilities $P^a_{ss'}$, respectively. As can be seen in Fig. 5, the state S refers to a list of squared grids which are marked with a number sequence from 1 to 25. Moreover, the numbers in bracket indicate the immediate rewards. It should be noted that the destination and obstacle are assigned with $+\mho$ and $-\mho$, respectively ($\mho \gg 0$). The other grids are assigned with -0.1 to denote the cost stepping into another grid. It should be noted that the destination can be replaced with a moving target. Owing to the developed time-space configuration, the flexibility of MDP is improved to achieve specific missions including target tracking.

Figure 6 highlights the basic UAV actions which are designed to be eight movements, $a = 1, 2, \ldots, 8$. The entire 360° is equally separated by these actions and all angles between two adjacent actions are 45°. The UAV moves toward the front grid with 0.9 probability in the horizontal plane when it selects a random action, at the same time the movement may lead to the UAV stepping into both sideward grids with lower probability.

1(-0.1)	6(-0.1)	11(-0.1)	16(-0.1)	21(+℧) Target
2(-0.1)	7(-0.1)	12(-0.1)	17(-0.1)	22(-0.1)
3(-0.1)	8(-0.1)	13(-℧) Obs	18(-0.1)	23(-0.1)
4(-0.1)	9(-0.1)	14(-0.1)	19(-0.1)	24(-0.1)
5(-0.1)	10(-0.1)	15(-0.1)	20(-0.1)	25(-0.1)

Fig. 5 The specific selections of the UAV basic actions

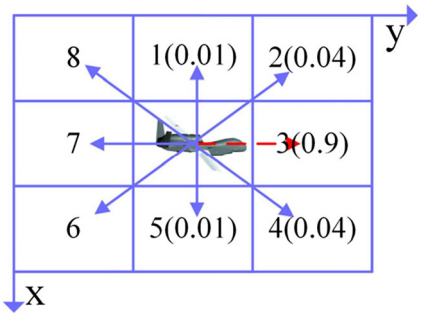

Fig. 6 The scenario of basic action and transition probability

The UAV selects action a_t repetitively at each of a sequence of discrete time steps $t = 0, 1, 2, \ldots$ then the UAV enters into the next possible state s_{t+1} with transition probability $P^a_{ss'}$. Given the state s and action a, the transition probability corresponding to the next state s' can be represented as:

$$P^a_{ss'} = P_r\left(s_{t+1} = s' | s_t = s, a_t = a\right). \tag{11}$$

The environment responds to this action and presents developmental state s_{t+1} to the UAV, meanwhile the UAV receives immediate reward r_{t+1} from the environment. Similarly, given the state s_t and action a_t, together with the next state s', the reward is represented as:

$$R^a_{ss'} = E\left\{r_{t+1} | s_t = s, a_t = a, s_{t+1} = s'\right\}. \tag{12}$$

The destination or target position is assigned with the only positive immediate reward, $+\mho$ which is also named as absorbing state. MDP implements a link (policy π) at a sequence of discrete time steps from the current state s to the next possible state s'. Therefore, the generation of UAV collision-free trajectory is transformed into searching for the optimal policy π^* and selecting the optimal action set.

3.2.2 Policy Iteration

The generation process of optimal policy is named as policy iteration. The optimal policy π^* can be achieved by searching for the maximum expected reward $V^*(s)$. The state value function $Q^\pi(s, a, t)$ and value function $V^\pi(s, t)$ are introduced as significant tools:

$$\begin{aligned} Q^\pi(s, a, t) &= E_\pi\{R | s_t = s, a_t = a\} \\ &= E_\pi\left\{\sum_{k=0}^{\infty} \lambda^k r_{t+k+1} | s_t = s, a_t = a\right\}, \end{aligned} \tag{13}$$

$$\begin{aligned} V^\pi(s, t) &= E_\pi\{R | s_t = s\} \\ &= E_\pi\left\{\sum_{k=0}^{\infty} \lambda^k r_{t+k+1} | s_t = s\right\}. \end{aligned} \tag{14}$$

where $\lambda = 0.6$ is the discount factor to indicate the influence of future reward on current value. E_π denotes the excepted value under policy π. $Q^\pi(s, a, t)$ represents the expected reward when the UAV takes action a in state s according

to policy π. For every $Q^{\pi}(s,a,t)$ in same state and same time, there exists eight values because of eight actions $a = 1, 2, \ldots, 8$. The value function $V^{\pi}(s,t)$ is the maximum of the eight values:

$$V^{\pi}(s,t) = max\, Q^{\pi}(s,a,t) \tag{15}$$

The initial approximation V_0 is chosen arbitrarily (generally 0). The successive approximation can be obtained by iteration solution method:

$$\begin{aligned} V_{k+1}(s,t) &= E_{\pi}\{r_{t+1} + \gamma V_k(s_{t+1}, t+1) \,|s_t = s\} \\ &= \sum\nolimits_a \pi(s,a) \sum\nolimits_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma V_k\left(s', t+1\right)\right]. \end{aligned} \tag{16}$$

For each $s \epsilon S$, in order to produce successive approximation V_{k+1} from V_k, the new value is obtained through the old value of the successor state s' and then replace the old value of state s. Iteration solution method evaluates the expected rewards along the one-step transition until $V_{k+1}(s,t) = V_k(s,t)$. $V_k = V^{\pi}$ is assured since the Bellman equation guarantees the equality in this case [28]. Equation 16 indicates that the sequence $\{V_k\}$ can converge to V^{π} as $k \to \infty$.

After the determination of the value function V^{π} for a policy π, a problem should be resolved that whether another action $a \neq \pi(s)$ can be chosen to improve the policy π. The solution of selecting another action $a \neq \pi(s)$ and obtaining a better policy π' is called policy improvement. If inequality (17) holds, then there exists a policy π' better than the current one π. Similarly, inequality (18) can be obtained:

$$Q^{\pi}\left(s, \pi'(s), t\right) \geq V^{\pi}(s,t), \tag{17}$$

$$V^{\pi'}(s,t) \geq V^{\pi}(s,t). \tag{18}$$

With reference to inequality (18), the value function $V^{\pi'}$ can be calculated as:

$$\begin{aligned} V^{\pi}(s,t) &\leq Q^{\pi}\left(s, \pi'(s), t\right) \\ &= E_{\pi'}\left\{r_{t+1} + \gamma V^{\pi}(s_{t+1}) \,|s_t = s\right\} \\ &\leq E_{\pi'}\left\{r_{t+1} + \gamma Q^{\pi}\left(s_{t+1}, \pi'(s_{t+1})\right) |s_t = s\right\} \\ &= E_{\pi'}\left\{r_{t+1} + \gamma E_{\pi'}\left(r_{t+2} + \gamma V^{\pi}(s_{t+2})\right) |s_t = s\right\} \\ &\cdots \\ &\leq E_{\pi'}\left\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} \ldots |s_t = s\right\} \\ &= V^{\pi'}(s,t). \end{aligned} \tag{19}$$

With reference to the greedy policy, the better policy π' is produced to replace the policy π as:

$$\begin{aligned} \pi'(s,t) &= argmax_a\, Q^{\pi}(s,a,t) \\ &= argmax_a\, E\left\{r_{t+1} + \gamma V^{\pi}(s_{t+1}, t) \,|s_t = s, a_t = a\right\} \\ &= argmax_a \sum\nolimits_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma V^{\pi}(s', t)\right]. \end{aligned} \tag{20}$$

Generally, the greedy policy is considered to evaluate a change at all states with allowable actions, thus selecting the optimal actions for all states and extracting the improved policy. Once a better policy π' is yielded through policy improvement, the value function $V^{\pi'}(s,t)$ can be calculated and again be used to yield an even better policy π''. This method is named policy iteration, which can be described as:

$$\pi_0 \xrightarrow{PE} V^{\pi_0} \xrightarrow{PI} \pi_1 \xrightarrow{PE} V^{\pi_1} \xrightarrow{PI} \pi_2 \ldots \xrightarrow{PI} \pi^* \xrightarrow{PE} V^*. \tag{21}$$

In the iterative flow path, PE and PI represent policy evaluation and policy improvement, respectively. Typically, an optimal policy π^* for all states can be derived by policy iteration. For more details about MDP, please refer to Appendix B.

3.3 The Combination of Optimal Policy with Initial Position

In practice, the derived optimal policy π^* is a number sequence, while each grid in the environment is attached with a number. Notice that the number sequence indicates the optimal movements in each grid of the environment. In other words, the optimal actions can be obtained by combining the optimal policy with current position.

As indicated in Fig. 7a, a UAV is searching for an optimal trajectory toward the destination in the left bottom of a simplified grid environment. Therefore, the UAV firstly has to decide which action is the optimal choice in the current grid. In order to clearly present the policy iteration process with the actual action selection, the potential actions of the UAV in initial position are shown in Fig. 7d–f as the convergence of iterative policy. There are eight allowable actions in the current position. The potential actions are associated with the state value function of $Q^{\pi}(s,a,t)$, the maximum value can be approximated according to greedy policy. By this way, the optimal action is determined after convergence. After a sequence of policy iteration to all grids in the environment, the optimal actions can be derived as in Fig. 7b. Finally, the trajectory of the UAV toward destination is generated as shown in Fig. 7c.

Remark 1 Through modeling decision-making process, the MDP addresses the optimization problem with dynamic programming. With the aid of the greedy policy to state value function $Q^{\pi}(s,a,t)$ and value function $V^{\pi}(s,t)$, the maximal state value function $Q^{\pi^*}(s,a,t)$ and optimal policy π^* can be obtained after convergence of iterative policy evaluation.

Remark 2 The optimal policy π^* refers to the optimal action set in all grids of the environment as shown in

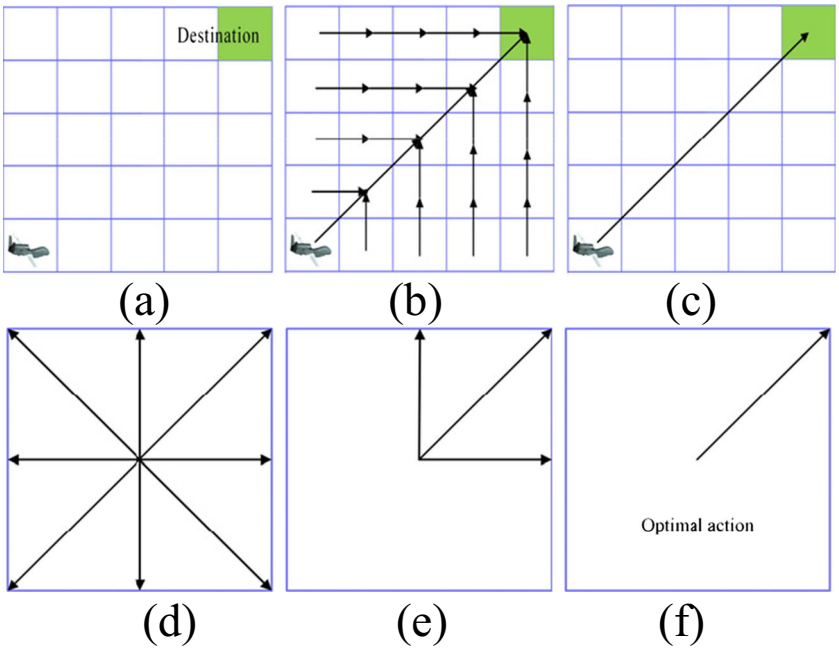

Fig. 7 The process of UAV searching for optimal actions toward the destination

Fig. 7b. After determining all optimal actions in the grid environment, the trajectory is actually composed of a sequence of actions from initial position to destination, as depicted in Fig. 7c.

Remark 3 It should be noted that every actions and states are assigned with sequential time. Specifically, the next UAV action is decided by the policy of next moment, $a = \pi(s, t)$. Consequently, there are two features in this work: 1) the target or destination reacts like an attractive potential which navigates the UAV for adjusting direction to approach them and 2) the UAV can respond to the unpredicted movement of obstacle significantly which increases the collision avoidance performance. The introduction of time factor sets this study apart from general MDP.

Remark 4 While the MDP is adopted to generate a trajectory, the UAV may trap into a local minimum and terminate there without achieving the final goal. In order to address this issue, the backtracking algorithm is employed. The basic idea behind this method is to check whether the derived policy satisfies specific conditions before the determination of the next step. The conditions include: 1) the adjacent actions a_{t1} and a_{t2} satisfy $|a_1 - a_2| \leq 1$ and 2) the action will not incur collision. If not, another policy must be derived from the rest of potential solutions. The backtracking algorithm treats the local minimum problem through recursive search. The main procedure is provided in Appendix C.

Remark 5 With respect to MDP-based methods, the information of current environment states should be available. In this sense, MDP is named as fully observable Markov decision making process. If the environment states cannot be directly or fully observed, a partially observable MDP can be adopted to address the problem [34].

3.4 Integrating the UAV Constraints into the Trajectory

As stated previously, a collision-free trajectory for the UAV can be derived through the improved MDP method. However, the derived trajectory may not be feasible if the aerodynamic constraints are not considered explicitly. It is of paramount importance to drive the UAV flying from one position to another without exceeding the allowable aerodynamic range. In this study, the differential flatness method is adopted to achieve this goal.

For nonlinear dynamic systems, if there exist flat system outputs $F \in R^m$ with ability to express the system state variables and control input variables, the systems must be flat. Commonly, a dynamic system can be expressed as:

$$\begin{cases} \dot{x} = f(x, u) \\ y = h(x) \end{cases}, \tag{22}$$

where $x \in R^n$ and $u \in R^m$ represent system states and control inputs, respectively. The sufficient condition for a flatness system [35] can be defined as:

$$F = g\left(x, u, \dot{u}, \ddot{u}, \ldots, u^n\right), \tag{23}$$

where F represents the system outputs. Consequently, the above equation can be translated into:

$$\begin{cases} x = x_f(F, \dot{F}, \ddot{F}, \ldots, F^n) \\ u = u_f(F, \dot{F}, \ddot{F}, \ldots, F^n) \end{cases}. \tag{24}$$

Equation 24 indicates the state variables $x \in R^n$ and control inputs $u \in R^m$ can be expressed with flat outputs, $F \in R^m$. Furthermore, x_f and u_f link the flat outputs and their derivates to the system input and state variables. This system is flat with flat outputs $F_1 = x$, $F_2 = y$, and $F_3 = \phi$. In this study, the coordinated turn is closely associated with rolling motion. The maximal roll angle ϕ_{max} is considered as a dynamic constraint which is resolved through differential flatness method. According to Eq. 4, the parameterization of roll angle can be derived as:

$$\phi(t) = atan\frac{\upsilon\dot{\psi}(t)}{g}. \tag{25}$$

To determine the time corresponding to the maximal roll angle, it is necessary to calculate the extreme of the roll angle. In terms of Eq. 25, the time derivative of the roll angle is:

$$\dot{\phi}(\mathrm{t}) = \frac{\upsilon\ddot{\psi}(t)}{g}\frac{g^2}{g^2+\upsilon^2\dot{\psi}^2}. \tag{26}$$

Then the calculation of the maximal roll angle turns to calculate the first and second derivatives of the yaw angle. The reference trajectory is designed as:

$$F_i(t) = \left[1-(1+\omega_n t)\,e^{-\omega_n t}\right]R_i + R_i^0;\ i = 1,2,3, \tag{27}$$

where ω_n, R_i, and R_i^0 indicate the natural frequency, the amplitude of flat outputs, and the initial value of flat outputs, respectively. Therefore, the first and second derivatives of the yaw angle are:

$$\dot{\psi}(t) = \omega_n^2 t R_3 e^{-\omega_n t}, \tag{28}$$

$$\ddot{\psi}(t) = \omega_n^2 R_3 e^{-\omega_n t}[1-\omega_n t]. \tag{29}$$

Besides, the relationship between the settling time t_s and natural frequency ω_n can be derived from Eq. 27. The settling time is designed to be the time when output signal enters and remains the final value within 5% error band. In view of the first-order control system, the relationship can be represented as:

$$(1+\omega_n t)\,e^{-\omega_n t} = 0.05. \tag{30}$$

Based on the solution of the above equation, the approximation of natural frequency is:

$$\omega_n \approx \frac{5.83}{t_s}. \tag{31}$$

In terms of Eqs. 28, 29, and 31, the derivate of the roll angle can be written as:

$$\begin{aligned}\dot{\phi}(t) &= \frac{\upsilon\ddot{\psi}(t)}{g}\frac{g^2}{g^2+\upsilon^2\dot{\psi}^2}\\ &= \left[1-\frac{5.83t}{t_s}\right]\frac{5.83^2 R_3\upsilon}{gt_s^2}e^{-\frac{5.83t}{t_s}}\\ &\quad\times\frac{g^2}{g^2+\upsilon^2\frac{5.83^4 t^2 R_3^2}{t_s^4}e^{-2\frac{5.83t}{t_s}}}.\end{aligned} \tag{32}$$

It is obvious that the extreme value of the roll angle ϕ_{ext} can be determined at the time $t = \frac{t_s}{5.83}$:

$$\phi_{ext} = \mathrm{atan}\left(\frac{5.83\upsilon R_3 e^{-1}}{gt_s}\right). \tag{33}$$

By comparing all the values of the roll angle including its beginning, end and extreme values, the maximum is the extreme value. Consequently, to guarantee that the maximum roll angle within the allowable range, the following condition must be satisfied:

$$|\phi_{ext}| \le \phi_{max}. \tag{34}$$

$$t_s \ge \frac{5.83\upsilon R_3 e^{-1}}{gtan(\phi_{max})}. \tag{35}$$

In accordance with Eq. 35, the condition for settling time is $t_s \ge \frac{5.83\upsilon R_3 e^{-1}}{gtan(\phi_{max})}$ which satisfies the constraints about roll angle.

Since settling time t_s is already determined, it is of importance to design the reference trajectories $F_i, i = 1,2,3$. In this study, the Bessel curve is adopted to smooth the profile of the trajectory based on the settling time. A general Bessel function of degree n is:

$$F_i = a_n t^n + a_{n-1}t^{n-1} + \cdots + a_1 t^1 + a_0, i = 1,2, \tag{36}$$

where a_i $(i = 0,1,2,\ldots)$ are constant coefficients which can be calculated by the initial and the final conditions. The trajectory is smoother as the increase of n, meanwhile the computation load for trajectory planning becomes heavier. After overall consideration, the Bessel function is designed as degree of 4 in this study:

$$F_i = a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t^1 + a_0, (i = 1,2). \tag{37}$$

The main goal of flatness is to drive the UAV from the initial position $(x(t_0), y(t_0))$ to final position $(x(t_s), y(t_s))$ without violating dynamic constraints. The initial position and final position are already generated based on the improved MDP, meanwhile, the initial constant t_0 and the final constant t_s can be easily obtained:

$$F_i(t_o) = a_4 t_0^4 + a_3 t_0^3 + a_2 t_0^2 + a_1 t_0^1 + a_0, (i = 1,2), \tag{38}$$

$$F_i(t_s) = a_4 t_s^4 + a_3 t_s^3 + a_2 t_s^2 + a_1 t_s^1 + a_0, (i = 1,2). \tag{39}$$

Based on the initial conditions and the final conditions, the coefficients a_j^i can be calculated.

4 Path Following Controller Design

After the flatness of reference trajectory, it is necessary to design a lateral controller to track roll and yaw commands for achieving the lateral movement stabilization. Therefore, a controller based on PBC is designed to cope with the

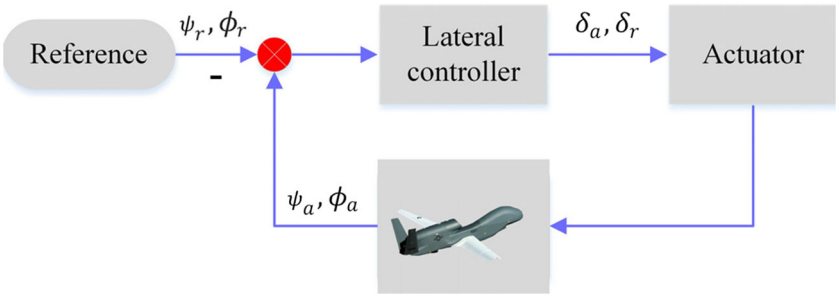

Fig. 8 The structure of the lateral controller

coordinated turn and automatically follow the reference trajectory. The structure of the lateral controller is illustrated as in Fig. 8. One objective of this controller is to eliminate the roll and yaw tracking error.

In this study, the generalized coordinate of lateral dynamics is simplified as $\Upsilon = [\phi, \psi]^T$ and the input is described as $u = [\delta_a, \delta_r]^T$. The focus of the controller is to eliminate the tracking errors of roll and yaw angles, $\tilde{\Upsilon}$ between Υ_a and Υ_r which can be expressed as:

$$\tilde{\Upsilon} = \Upsilon_a - \Upsilon_r. \tag{40}$$

Based on the tracking errors, an error function Ω is defined as:

$$\Omega = I\ddot{\tilde{\Upsilon}} + k_1\dot{\tilde{\Upsilon}} + k_2\tilde{\Upsilon}. \tag{41}$$

where k_1 and k_2 are positive definite matrices. Therefore, an object of the work is to make $\Omega \rightarrow 0$. The control input is designed as follows:

$$u = M^{-1}\left[I\ddot{\Upsilon}_r + \frac{\partial F}{\partial \dot{\Upsilon}} - k_1\dot{\tilde{\Upsilon}} - k_2\tilde{\Upsilon}\right]. \tag{42}$$

Substituting Eq. 42 into Eq. 5 achieves:

$$\begin{aligned} I\ddot{\Upsilon} + \frac{\partial F}{\partial \dot{\Upsilon}} &= Mu \\ &= I\ddot{\Upsilon}_r + \frac{\partial F}{\partial \dot{\Upsilon}} - k_1\dot{\tilde{\Upsilon}} - k_2\tilde{\Upsilon}. \end{aligned} \tag{43}$$

Table 1 Parameters of path planning

Parameter	Value
UAV cruising velocity (m/s)	30
Maximum roll angle (°)	−30, 30
Moving obstacle velocity (m/s)	30
Moving target velocity (m/s)	6
Basic action-state cost	-10^{-10}
Cost for targets or destination	20
Cost for obstacles	−20
Maximum aileron deflection (°)	−25, 25
Maximum rudder deflection (°)	−30, 30
Minimum safety distance (m)	50
Time interval (s)	1

On the basis of Eq. 43, another solution can be derived as:

$$\begin{aligned} I\ddot{\Upsilon} + \frac{\partial F}{\partial \dot{\Upsilon}} - \left(I\ddot{\Upsilon}_r + \frac{\partial F}{\partial \dot{\Upsilon}} - k_1\dot{\tilde{\Upsilon}} - k_2\tilde{\Upsilon}\right) &= I\ddot{\tilde{\Upsilon}} + k_1\dot{\tilde{\Upsilon}} + k_2\tilde{\Upsilon} \\ &= 0, \end{aligned} \tag{44}$$

$$\Omega = I\ddot{\tilde{\Upsilon}} + k_1\dot{\tilde{\Upsilon}} + k_2\tilde{\Upsilon} = 0. \tag{45}$$

Besides, a stored energy function based on the energy balance is established as:

$$H\left(\tilde{\Upsilon}, \dot{\tilde{\Upsilon}}\right) = \frac{1}{2}\tilde{\Upsilon}^T k_2\tilde{\Upsilon} + \frac{1}{2}\dot{\tilde{\Upsilon}}^T I\dot{\tilde{\Upsilon}}. \tag{46}$$

In order to make the error function Ω converged, the derivate of the stored energy function must be not positive. The derivate of storage function $H\left(\tilde{\Upsilon}, \dot{\tilde{\Upsilon}}\right)$ is:

$$\dot{H}\left(\tilde{\Upsilon}, \dot{\tilde{\Upsilon}}\right) = k_2\tilde{\Upsilon}\dot{\tilde{\Upsilon}} + I\dot{\tilde{\Upsilon}}\ddot{\tilde{\Upsilon}}. \tag{47}$$

By substituting Eq. 45 into Eq. 47, a solution can be obtained as:

$$\dot{H}\left(\tilde{\Upsilon}, \dot{\tilde{\Upsilon}}\right) = (k_2\tilde{\Upsilon} + I\ddot{\tilde{\Upsilon}})\dot{\tilde{\Upsilon}} = -k_1\dot{\tilde{\Upsilon}}^2 \leq 0. \tag{48}$$

Remark 6 The tracking error is modeled with an error function Ω which creates a map $\Omega \rightarrow \dot{\tilde{\Upsilon}}$ with the stored energy function, $H\left(\tilde{\Upsilon}, \dot{\tilde{\Upsilon}}\right)$. By resorting to the energy function, the controller is utilized to test the energy state of the whole system. The energy function, $H\left(\tilde{\Upsilon}, \dot{\tilde{\Upsilon}}\right)$, is positive definite while $\dot{H}\left(\tilde{\Upsilon}, \dot{\tilde{\Upsilon}}\right)$ is negative semi-definite. Consequently, the error function Ω will converge to and the control system maintains global asymptotic stability in the equilibrium point $\dot{\tilde{\Upsilon}} = 0$ according to the Lyapunov theorem.

Table 2 Initial conditions of the UAV

Simulation	Case 1	Case 2	Case 3
Initial position (m)	(60,300)	(30,30)	(30,750)
Target position (m)	(2340,300)	changeable	(1450,750)
Initial speed (m/s)	30	30	30
Heading angle (°)	0	45	0

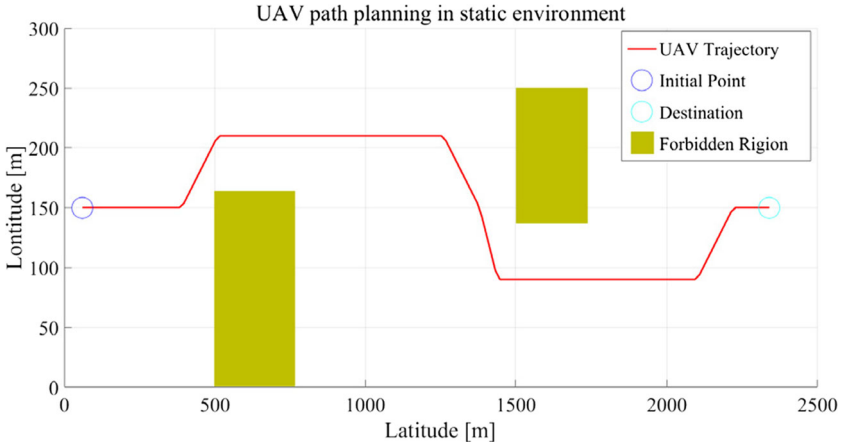

Fig. 9 UAV collision avoidance in static environment

Remark 7 The system's energy function is modified based on PBC technique and Lyapunov theorem, which aims to completing collision avoidance maneuvers. By maintaining the energy balance of the system, the lateral controller is employed to accomplish two objectives: 1) follow the roll and yaw reference trajectory and 2) provide roll and yaw stabilization.

5 Simulation and Result Assessment

5.1 Simulation Scenarios

The developed strategy is tested in MATLAB R2015 on the computer configured with a 2.8 GHz Intel Core I7 processor with 8 GB RAM. Relevant parameters of three simulations are listed in Tables 1 and 2. In order to adequately demonstrate the collision avoidance ability in the UAV, the simulation studies involve both moving intruder and forbidden regions. The forbidden regions are represented with rectangles. If the UAV continues flying without appropriate collision avoidance maneuvers, the collision will take place. Case 1 illustrates the static obstacles avoidance performance of UAV with the proposed strategy. The collision free trajectory generation and target tracking ability are demonstrated in Case 2. In order to further illustrate the ability of the proposed algorithm to avoid both static and moving obstacles in a cluttered environment, another simulation test is shown in Case 3. A band-limited white noise which mimics the sensor noise is injected into measurement channels.

5.2 Simulation Results and Performance Assessment

5.2.1 Collision Avoidance Tests and Analysis

As indicated in Fig. 9, the collision avoidance simulation scenario is implemented in a static grid environment. The flying horizontal plane is defined as XY plane. Each grid is a 30 m × 30 m square, and the UAV can fly to a neighboring grid from the current position at every time interval. On the basis of [36], the time interval is defined as 1 s. As for Case 1, the computation time for every decision step is less than 1 s demonstrating that the defined time interval is

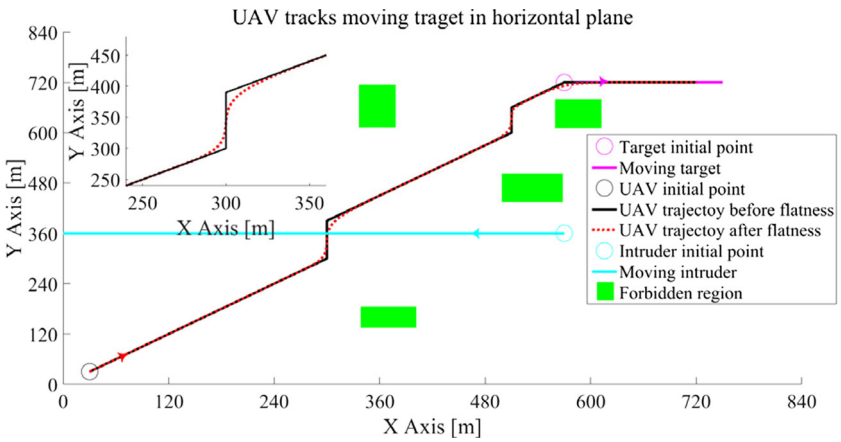

Fig. 10 UAV collision avoidance and target tracking

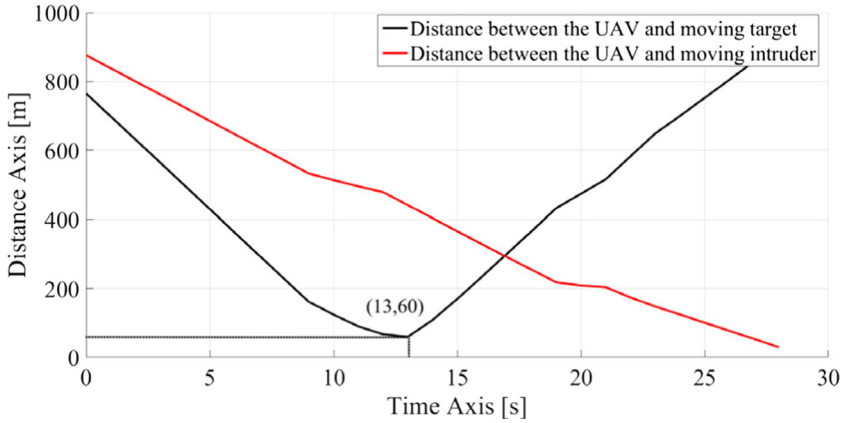

Fig. 11 The collision avoidance and target tracking performance analysis

compliant with the time required for the simulation to run. Note that the computation cost will increase exponentially as the planning space grows which makes MDP based algorithm not suitable for path planning in a sufficiently large space [37]. As shown in Fig. 10, the UAV approaches the target through achieving larger reward. Due to the negative basic action-state cost, the UAV attempts to fly closer to the target with fewer actions. However, a moving intruder is predicted to arrive at the neighboring area of the UAV. Moreover, several forbidden regions are existed in the surroundings. Consequently, a collision-free trajectory is re-planned to avoid the moving obstacle and to prevent from negative reward by virtue of MDP-based strategy. The same strategy is applied to deal with the forbidden areas. Finally, the UAV changes its direction before colliding with intruder and static obstacles, arriving at the same horizontal line to continue tracking the moving target. The black line represents the trajectory with basic actions which exceeds the constraints about roll angle and yaw rate in practice. Thus the differential flatness method is adopted to resolve this problem and the red line is the desired trajectory. The subplot is the comparison of the trajectories before and after flatness.

As can be observed from Fig. 11, the UAV approaches the moving intruder before the 13th second and then the distance between them increases. The minimum distance in the 13th second is 60 m, which implies the collision avoidance between the UAV and moving intruder is completed. Figure 11 also highlights that the UAV gets closer to the moving target and their distance decreases continually, which confirms the feasibility of UAV for tracking moving target. In conclusion, the proposed collision avoidance strategy is suitable for UAVs to track moving target while avoid obstacles in complex environment.

Considering that the flight environment in practice may be hostile, the problem needs to be resolved for a safe flight. In order to demonstrate the robustness of the proposed strategy adequately, a complicated environment is introduced in Fig. 12. In Case 3, A UAV is flying toward destination along the initial trajectory while many static obstacles are in the way. Those blocks inducing collision will be associated with negative immediate reward. After the

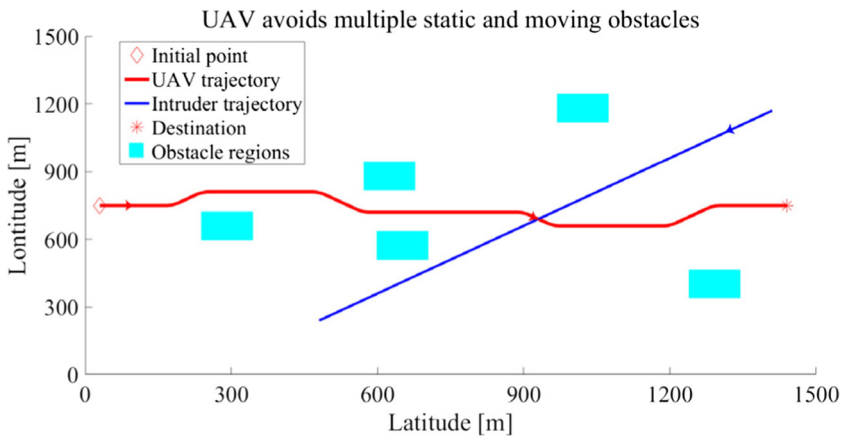

Fig. 12 Trajectories generation in a hostile environment

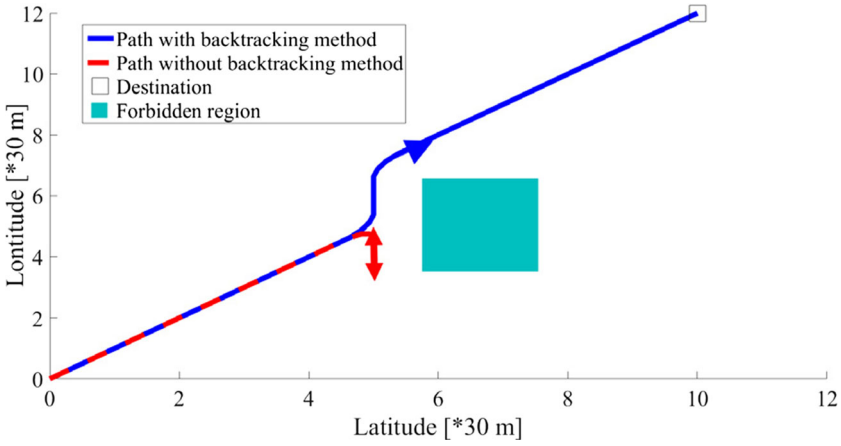

Fig. 13 Path planning with and without backtracking method

UAV passed several blocks along the red trajectory, another moving obstacle, an aircraft, is flying toward the UAV and may collide with the UAV. Obviously, the UAV changes the flight direction (red arrow) and bypass the moving obstacle in due course. Consequently, the UAV with the proposed algorithm can avoid multiple obstacles following the designed trajectory safely in a hostile environment.

In an attempt to overcome the limitation of local minimum, a backtracking method is integrated with MDP. The comparative result is presented in Fig. 13, where two trajectories are generated with and without backtracking method to avoid forbidden region. It can be seen that unlike the latter method, the UAV with the first scenario arrives at the destination safely while the latter cannot avoid the obstacle and move back and forth around the obstacle. It exemplifies that the backtracking method can be used to resolve the local minimum problem in MDP-based strategy.

5.2.2 Path Following Result and Performance Assessment

The path following performance to target tracking test is demonstrated in this section. As can be seen in Fig. 14, the adopted controllers guarantee that the roll and yaw angles track the reference signals. The tracking errors for yaw angle and roll angle are within ±3°, which illustrate that the controller designed in this study is capable of satisfying the limit of performance criterion. From Fig. 15, the control inputs, which consist of aileron deflection and

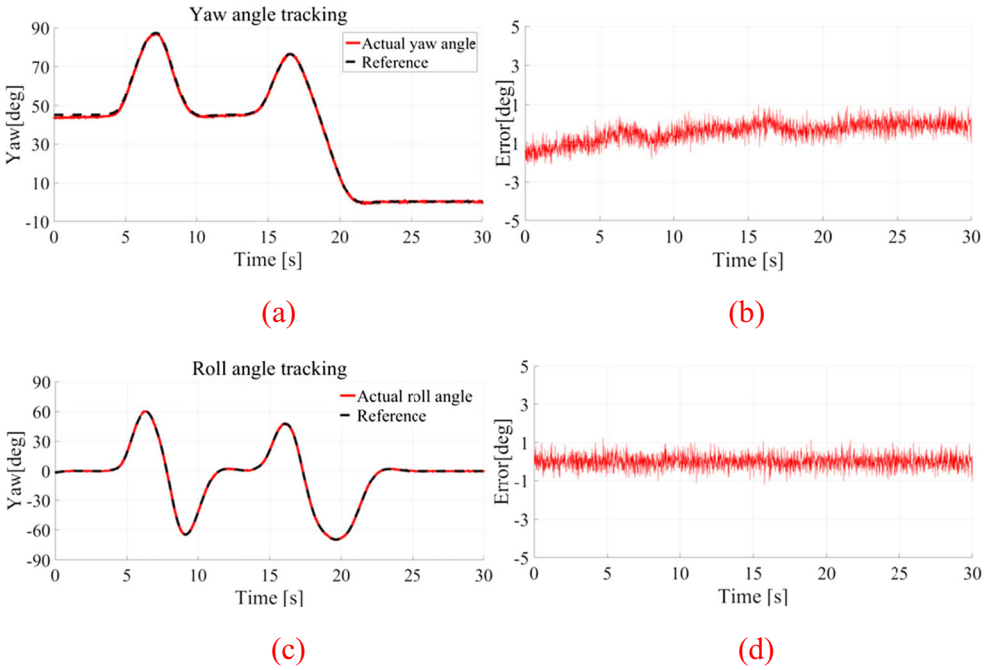

Fig. 14 UAV outputs in response to reference roll and yaw angles

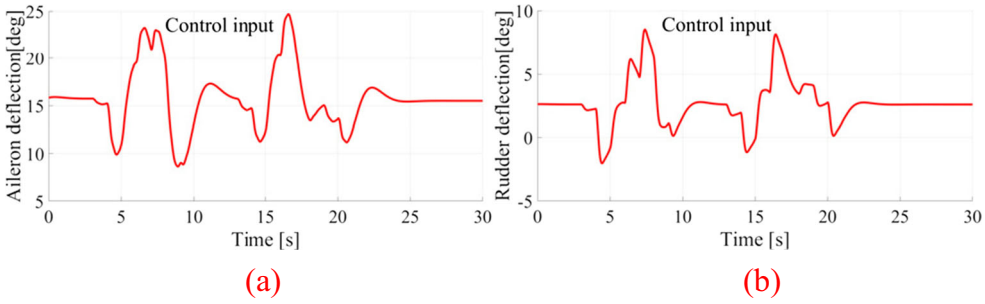

Fig. 15 Aileron and rudder deflection angles of the UAV

rudder deflection, are within the allowable range of UAV dynamic constraints, which attribute to the differential flatness.

6 Conclusion

The large-scale development of UAV in both civilian and military domains has inspired the need to design high-efficiency collision-free strategies. In order to guarantee the flight safety, a MDP-based method is proposed to navigate the UAV and re-plan the path in the case of collision prediction and detection. Three cases involving collision avoidance in static, dynamic environment, and target tracking are performed to illustrate the effectiveness of the proposed MDP-based strategy. Moreover, target tracking is another kind of superiority which has great application potential. The uniqueness of the developed strategy includes: 1) based on the time-space configuration, the UAV with the proposed MDP-based strategy will keep away from the obstacles; 2) local minimum can be avoided by adding backtracking method into the MDP-based strategy, while the UAV will respond to target tracking mission; and 3) a flatness-based trajectory planning method and a lateral controller for trajectory tracking are appropriately utilized to improve the applicability of the developed algorithm. Due to the target tracking ability, the proposed method can be adopted to specific applications including surveillance and reconnaissance. The integration of lateral controller and differential flatness method can further expand the application of the proposed method especially in a cluttered environment.

However, the algorithm is applied currently in a horizontal plane instead of a three-dimensional space. Besides, the environment state information has to be fully observed due to the requirement of MDP. These two factors need to be explicitly considered in the future work.

Acknowledgments This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, in part by the National Natural Science Foundation of China under Grant 61573282 and Grant 61603130. The authors would like to express their sincere gratitude to the Editor-in-Chief, the Guest Editors, and the anonymous reviewers whose insightful comments have helped to improve the quality of this paper considerably.

Appendix A

Based on the condition that slide angle $\beta = 0$ and pitch angle $\theta = 0$, the partial differential of the Rayleigh dissipation function ($\frac{\partial F}{\partial \dot{\phi}}$, $\frac{\partial F}{\partial \dot{\varphi}}$) are as follows:

$$\begin{aligned}\frac{\partial F}{\partial \dot{\phi}} &= -1051.05\cos(\varphi)\left(6.6(-11\dot{\phi}+5\dot{\varphi}\cos(\phi))\sin(\phi)/V\right)V^2\\ &\quad +1051.05\cos(\phi)\sin(\varphi)(-0.4461-92.4424\dot{\varphi}\sin(\phi)/V\\ &\quad +0.11(1.0656+24.6016\left(\frac{\dot{\varphi}\sin(\phi))}{V}+0.143V^2\right)\\ &\quad -1051.05(\sin(\phi)\sin(\varphi))\left(6.6\left(1.7\dot{\phi}-11.5\varphi\cos(\phi)\right)/V\right)V^2\\ \frac{\partial F}{\partial \dot{\varphi}} &= -1051.05\cos(\varphi)\left(6.6(-11\dot{\phi}+5\dot{\varphi}\cos(\phi))/V\right)V^2\\ &\quad -1051.05\sin(\phi)(-0.4461-92.4424\dot{\varphi}\sin(\phi)\cos(\theta)/V\\ &\quad +0.11(1.0656+24.6016\left(\frac{\dot{\varphi}\sin(\phi))}{V}+0.143V^2\right)\\ &\quad -1051.05\cos(\phi)\left(\frac{6.6(-11.5\dot{\varphi}\cos(\phi))}{V}\right)V^2\end{aligned}$$

Input matrix $M = \begin{bmatrix} M_{\phi,\delta_a} & M_{\phi,\delta_r} \\ M_{\varphi,\delta_a} & M_{\varphi,\delta_r} \end{bmatrix}$ which are due to drag, lift and side force can be expressed as follows:

$$\begin{aligned}M_{\phi,\delta_a} &= -630.63V^2\cos(\varphi)\\ M_{\phi,\delta_r} &= 231.231V^2\cos(\varphi)-634.413V^2\sin(\phi)\sin(\varphi)\\ M_{\varphi,\delta_a} &= 0\\ M_{\varphi,\delta_r} &= -634.413V^2\cos(\phi)\end{aligned}$$

Appendix B

A. Procedure of Policy Iteration

1. Initialization

$V(s) \in R$ and $\pi(s) \in A(s)$ arbitrarily for all $s \in S$

2. Policy Evaluation
Repeat
$\vartheta \longleftarrow 0$
For each $s \in S$:
$v \leftarrow V(s)$
$V(s) \leftarrow \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma V(s') \right]$
$\vartheta \longleftarrow max\,(\vartheta, |v - V(s)|)$
until $\vartheta < \theta$ (a small positive number)
3. Policy Improvement
Policy stable ← true
For each $s \in S$:
$b \leftarrow \pi(s)$
$\pi(s) \leftarrow arg\ max_a \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma V(s') \right]$
If $b \neq \pi\,(s)$, then Policy stable ← false
If policy stable, then stop; else, go to 2.

Appendix C

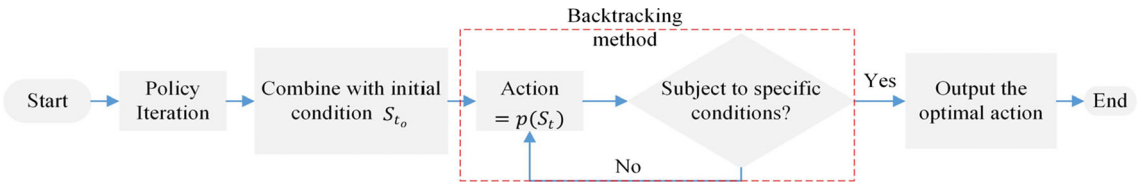

Fig. 16 The integration of backtracking method with MDP

References

1. Gundlach, J.: Designing Unmanned Aircraft Systems: A Comprehensive Approach. American Institute of Aeronautics and Astronautics, Reston (2012)
2. Halit, E., Kemal, L.: 3D path planning for multiple UAVs for maximum information collection. J. Intell. Robot. Syst. **73**(1–4), 737–762 (2014)
3. Angelov, P.: Sense and Avoid in UAS: Research and Applications. Wiley, Hoboken (2012)
4. Yang, K., Gan, S.K., Sukkarieh, S.: An efficient path planning and control algorithm for RUAV's in unknown and cluttered environment. J. Intell. Robot. Syst. **57**, 101–122 (2010)
5. Yu, X., Zhang, Y.M.: Sense and avoid technologies with applications to unmanned aircraft systems: Review and prospects. Prog. Aerosp. Sci. **74**, 152–166 (2015)
6. Gui, Y., Guo, P., Zhang, H., Lei, Z., Du, X., Du, J., Yu, Q.: Airborne vision-based navigation method for UAV accuracy landing using infrared lamps. J. Intell. Robot. Syst. **72**(2), 197–218 (2013)
7. Kuchar, J.K., Yang, L.C.: A review of conflict detection and resolution modeling methods. IEEE Trans. Intell. Trans. Syst. **1**(4), 179–189 (2000)
8. Akmeliawati, R., Mareels, I.M.Y.: Nonlinear energy-based control method for aircraft automatic landing systems. IEEE Trans. Control Syst. Technol. **18**(4), 871–884 (2010)
9. Khatib, O.: Real time obstacle avoidance for manipulators and bile Robots. Int. J. Rob. Res. **5**(1), 90–99 (1986)
10. Lavalle, S.: Planning Algorithms. Cambridge University Press, Cambridge (2006)
11. Chuang, J.H., Ahuja, N.: An analytically tractable potential field model of free space and its application in obstacle avoidance. IEEE Trans. Syst. Man. Cybern. B. Cybern. **28**, 729–736 (1998)
12. Geiger, B., Horn, J., Delullo, A., Niessner, A., Long, L.: Optimal path planning of UAV using direct collocation with nonlinear programming. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference, Keystone, Colorado (2006)
13. Sridhar, B., Ng, H.K., Chen, N.Y.: Aircraft trajectory optimization and contrails avoidance in the presence of winds. J. Guid. Control. Dyn. **34**(5), 1577–1583 (2011)
14. Schrijver, A.: Theory of Linear and Integer Programming. Wiley, Hoboken (1998)
15. Nikolos, I.K., Valavanis, K.P., Tsourveloudis, N.C., Kostaras, A.N.: Evolutionary algorithm based offline/online path planner for UAV navigation. IEEE Trans. Syst. B Man. Cybern. **33**(6), 898–912 (2003)
16. Son, Y.S., Baldick, R.: Hybrid coevolutionary programming for nash equilibrium search in games with local optima. IEEE Trans. Evol. Comput. **8**(4), 305–315 (2004)
17. Kim, D.H., Shin, S.: Self-organization of decentralized swarm agents based on modified particle swarm algorithm. J. Intell. Robot. Syst. **46**(2), 129–149 (2006)
18. Mauro, P., Conway, B.A.: Particle swarm optimization applied to space trajectories. J. Guid. Control Dyn. **33**(5), 1429–1441 (2010)
19. Pinto, A.M., Moreira, A.P., Costa, P.G.: A localization method based on map-matching and particle swarm optimization. J. Intell. Robot. Syst. **77**(2), 313–326 (2015)
20. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. J. Global. Optim. **39**(3), 459–471 (2007)
21. Fu, Y., Zhang, Y.M., Yu, X.: An advanced sense and collision avoidance strategy for Unmanned Aerial Vehicles in landing phase. IEEE Aerosp. Electron. Syst. Mag. **31**(9), 40–52 (2016)
22. Bhattacharya, P., Gavrilova, M.L.: Roadmap-based path planning-using the Voronoi diagram for a clearance-based shortest path. IEEE Robot. Autom. Mag. **15**(2), 58–66 (2008)
23. Pehlivanoglu, Y.V.: A new vibrational generic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV. Aerosp. Sci. Technol. **16**(1), 47–55 (2012)
24. Sridharan, K., Priya, T.K.: The design of a hardware accelerator for real-time complete visibility graph construction and efficient FPGA implementation. IEEE Trans. Ind. Electron. **52**(4), 1185–1187 (2005)

25. Kavraki, L.E., Švestka, P., Latombe, J.C.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans. Robot. Autom. **12**, 566–580 (1994)
26. Kavraki, L.E., Svestka, P., Latombe, J.C., Overmars, M.H.: Randomized preprocessing of configuration for fast path planning. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2138–2146, San Diego (1994)
27. Puterman, M.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley, Hoboken (2005)
28. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
29. Lian, Z.T., Deshmukh, A.: Performance prediction of an unmanned airborne vehicle multi-agent system. Eur. J. Oper. Res. **172**(2), 680–695 (2006)
30. Billingsley, T.B., Kochenderfer, M.J., Chryssanthacopoulos, J.P.: Collision avoidance for general aviation. IEEE Aerosp. Electron. Syst. Mag. **27**(7), 1–17 (2011)
31. Ure, N.K., Chowdhary, G., Chen, Y.F., How, J.P., Vian, J.: Distributed learning for planning under uncertainty problems with heterogeneous teams. J. Intell. Robot. Syst. **74**(1–2), 529–544 (2014)
32. Fu, Y., Yu, X., Zhang, Y.M.: Sense and collision avoidance of Unmanned Aerial Vehicles using Markov decision process and flatness approach. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 714–719. Lijiang (2015)
33. Miele, A.: Flight Mechanics: Theory of Flight Paths. Courier Dover Publications, New York (2016)
34. Bai, H., Hsu, D., Kochenderfer, M.J., et al.: Unmanned aircraft collision avoidance using continuous-state POMDPs. Robot. Auton. Syst. **1**, 1–8 (2012)
35. Chamseddine, A., Zhang, Y.M., Rabbath, C.A., Theilliol, D.: Trajectory planning and re-planning strategies applied to a quadrotor unmanned aerial vehicle. J. Guid. Control. Dyn. **35**(5), 1667–1671 (2012)
36. Yao, P., Wang, H., Su, Z.: Real-time path planning of unmanned aerial vehicle for target tracking and obstacle avoidance in complex dynamic environment. Aerosp. Sci. Technol. **47**, 269–279 (2015)
37. Powell, W.B.: Approximate Dynamic Programming. Wiley-Interscience, Hoboken (2008)

Xiang Yu received the B.S., M.S., and Ph.D. degrees in automatic control from Northwestern Polytechnical University, Xi'an, China, in 2003, 2004, and 2008, respectively. From 2009 to 2013, he was a postdoctoral research fellow at The University of Western Ontario, London, ON, Canada. He is currently working at Concordia University. His main research interests include GNC for aerospace engineering systems and fault-tolerant control design.

Xiaobin Zhou received the B.S. degree from Northestern University, Shenyang, China, in 2015. He is currently pursuing his doctoral degree at Hunan University, Changsha, China. His research interests include collision avoidance, as well as guidance, navigation, and control of unmanned aerial/ground/surface vehicles.

Youmin Zhang (M'99-SM'07) received the B.S., M.S., and Ph.D. degrees in automatic control from Northwestern Polytechnical University, Xi'an, China, in 1983, 1986, and 1995, respectively. Dr. Zhang is currently a Professor with the Department of Mechanical, Industrial & Aerospace Engineering, Concordia University. His current research interests include fault diagnosis and fault-tolerant (flight) control systems, cooperative GNC of unmanned aerial/space/ground/surface vehicles. He has authored four books, over 460 journal and conference papers, and book chapters. Dr. Zhang is a Fellow of CSME, a Senior Member of AIAA and IEEE, Vice-President of International Society of Intelligent Unmanned Systems, and a member of the Technical Committee for several scientific societies. He is an Editorial Board Member, Editor-in-Chief, Editor-at-Large, Editor or Associate Editor of several international journals. He has served as the General Chair, the Program Chair, and IPC Member of several international conferences.