MDPI

Real-time Conflict Resolution Algorithm for Multi-UAV Based on Model Predict Control

Hao-Xiang Chen *[ID]**, Ying Nan and Yi Yang**[ID]

College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing 20016, China;
nanying@nuaa.edu.cn (Y.N.); nuaa_yang@nuaa.edu.cn (Y.Y.)

* Correspondence: chenhaoxiang@nuaa.edu.cn; Tel.: +86-131-8295-8309

check for updates

Abstract: A real-time conflict resolution algorithm based on model predictive control (MPC) is introduced to address the flight conflict resolution problem in multi-UAV scenarios. Using a low-level controller, the UAV dynamic equations are abstracted into simpler unicycle kinematic equations. The neighboring UAVs exchange their predicted trajectories at each sample time to predict the conflicts. Then, under the predesignated resolution rule and strategy, decentralized coordination and cooperation are performed to resolve the predicted conflicts. The controller structure of the distributed nonlinear model predictive control (DNMPC) is designed to predict potential conflicts and calculate control variables for each UAV. Numerical simulations of multi-UAV coordination are performed to verify the performance of the proposed algorithm. Results demonstrate that the proposed algorithm can resolve the conflicts sufficiently in real time, while causing no further conflicts.

1. Introduction

Interest in research activities related to techniques using multiple Unmanned Aerial Vehicles (UAVs) has experienced a significant increase in recent decades [1,2]. As the application of cooperative UAVs to execute tasks has gradually increased, the common problem of maintaining a safe separation among all UAVs has aroused much attention from scholars. Therefore, conflict (detection and) resolution algorithms are needed for all the multi-UAV scenarios.

The major problem of conflict (detection and) resolution lies in the accurate collision prediction and efficient adjustment of the flight status among conflicting UAVs [2,3]. Many mythologies and approaches have been applied in order to this problem, among them are two main categories: numerical approaches [4–11] and control approaches [12–15]. As for numerical approaches, Refs. [4,5] considered the conflict resolution (CR) as a replanning problem and applied mixed-integer linear program (MILP). Masci and Tedeschi implemented conflict resolution by using game theory [6]. In Ref. [7], the conflict was resolved by tuning the speed of aerial vehicles based on Monte Carlo optimization, but not all conflicts could be resolved, due to the constraints of velocity change. Other CR methods have been proposed, such as application of geometric algorithm [8], ant colony optimization algorithm [9], multi-objective evolutionary algorithms [10] and differential geometric algorithm [11]. Numerical methods can efficiently resolve the conflicts in some typical scenarios, but they find it difficult to take the actual flight status of the UAV in to account.

Thus, other scholars have adopted control methods in solving the CR problem, a typical form among them is model predictive control (MPC). In Ref. [12], a centralized model predictive controller is designed that considers the physical limitations of the aircraft, although the timeliness of the method is not very good. A modified MPC approach combined with morphing potential field is proposed in Ref. [13], which prohibits UAVs' collision with obstacles and conflicts in real time. Chaloulos et al. [14]

designed a decentralized model predictive control (DMPC) scheme for hierarchical systems, which can provide a preview of the otherwise myopic navigation functions and generate conflict-free trajectories, but without considering the priorities of aircrafts in the system. Meanwhile, all the methods mentioned above focused on resolving current conflicts, while ignoring subsequent conflicts.

In this paper, an improved distributed nonlinear MPC (DNMPC) framework for autonomous UAVs is proposed to resolve the conflicts of flight trajectories that implement decentralized CR based on the communication between the conflicting UAVs. This proposed method is motivated by two main properties: reliable prediction of the potential conflicts and constraint handling for resolving the predicted conflicts by managing the current inputs and states. At the same time, corresponding resolution rule and strategy are proposed to decompose conflicts and resolve them in groups consists of several UAVs. A conflict resolution algorithm is also designed to calculate control variables for each UAV and to ensure no subsequent conflicts will be generated, which leads to a consistent unique decision for conflicting UAVs.

2. Problem Definition

There are N_v UAVs in the airspace at the same height; each UAV can be denoted as U_i (i = 1, . . . , N_v); the flight state variable of each UAV in discrete time sequence $k = 1, 2 \cdots t_k$ is denoted as $x_i(k)$, where $x_i(0)$ and $x_i(t_k)$ represent the initial and final state, respectively. The fundamental purpose of the conflict resolution problem is to design an efficient adjustment strategy to guide each UAV to its target without any collision with other UAVs, and to ensure the global cost minimum [15,16]. To solve this problem, assuming that UAVs within a certain range can receive each other's predicted trajectories between UAVs, the solution scheme can be described as Figure 1 [17]:
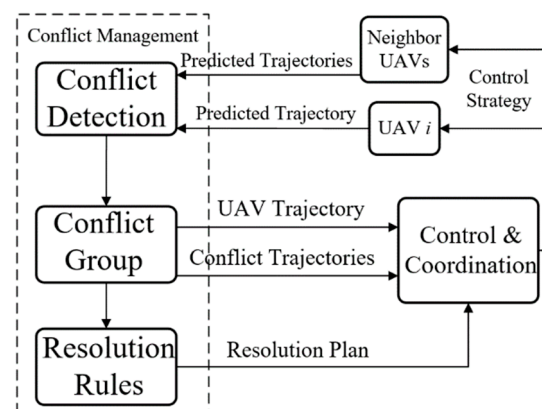

Figure 1. Scheme of solution for conflict resolution problem.

In the scheme, the whole system is divided into two main modules: the conflict management module and the control & coordination module. The conflict management module consists of 3 sub-modules: conflict detection, conflict group (construction) and resolution rule. Conflict detection sub-module receives and analyses predicted trajectories of adjacent UAVs within a certain range, and determines whether there are conflicts, then delivers the conflict information. Conflict group (construction) sub-module sorts the conflicts and establishes conflict groups based on the conflict information. Resolution rules sub-module generates resolution plans for each conflict group based on several collision free rules and constraints. The control & coordination module resolves the current conflicts depending on the resolution plan, local UAV trajectory and conflict trajectories, then deliver corresponding control strategies to UAVs with conflicts and generate new trajectories without collisions in the current time period. The entire coordination process is run once in a one-step cycle, and is continuously cycled to form a rolling real-time online optimization process, which is similar to the conception of model predictive control. Thus, this paper introduces a real-time coordination algorithm based on MPC, and can implement online rolling resolution to flight conflicts for UAVs.

3. Modeling of the Conflict Resolution Problem

3.1. Modeling of UAV

Supposing that all the UAVs are flying in the coordinate system of ground (*Oxyz*), which can be defined as: O is the selected coordinate origin, *Ox* points to true north, *Oz* axis points to true east, and *Oy* axis is determined by the right-hand rule; and all UAVs move within the horizon plane *Oxz*. So that the UAV can be regarded as Dubins' vehicle [18], then by discrete sampling of the flight process of each UAV at the *k*th time, the flight state of UAV can be defined as $x(k) = (x(k), z(k), \psi(k))$. The discretized kinematic equations of UAV can be given as:

$$\begin{cases} x(k+1) - x(k) = v(k)\Delta t[\sin\psi(k+1) - \sin\psi(k)]/\omega(k) \\ z(k+1) - z(k) = v(k)\Delta t[\cos\psi(k) - \cos\psi(k+1)]/\omega(k) \\ \psi(k+1) - \psi(k) = \omega(k)\Delta t \end{cases} \tag{1}$$

where, ψ is the heading angle of UAV, ω is heading angular speed, Δt is the sampling interval.

Rewrite (1) into a matrix form:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \\ \mathbf{y}(k+1) = \mathbf{C}(k+1)\mathbf{x}(k+1) \\ \mathbf{u}(k) = [v(k), \omega(k)]^{\mathrm{T}} \end{cases} \tag{2}$$

where, $\mathbf{y}$ represents for the output vector of UAV, matrixes $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ satisfy the equations below:

$$\mathbf{A}(k) = \mathbf{C}(k) = \mathbf{e}_{3\times3}; \quad \mathbf{B}(k) = \begin{bmatrix} \frac{[\sin\psi(k+1) - \sin\psi(k)]}{\omega(k)} & 0 \\ \frac{[\cos\psi(k+1) - \cos\psi(k)]}{\omega(k)} & 0 \\ 0 & \Delta t \end{bmatrix} \tag{3}$$

The constraints of this problem are mainly composed of UAV flight performance constraints and collision avoidance constraints:

(1) UAV flight performance constraints

$$\begin{cases} 0 \le |\omega(k)| \le \omega_{\max} \\ 0 \le |\dot{v}(k)| \le \mathrm{n}_{\max} \\ v_{\min} \le |v(k)| \le v_{\max} \\ r_{turn} = \frac{v_{\min}}{\omega_{\max}} \le \left|\frac{v(k)}{\omega(k)}\right| \end{cases} \tag{4}$$

where, $v_{\max}$, $\mathrm{n}_{\max}$, $\omega_{\max}$ represent for the maximum speed, overload, angular speed of UAV, r_{turn} represents for the minimum turning radius of UAV.

(2) Collision avoidance constraint Denote the safe flight radius of UAV as r_{safe}, if the distance between any two UAVs (r_{ij}) is less than r_{safe}, these two UAVs are considered to collide [17]. So at *k*th time, the collision avoidance constraint can be given as below:

$$r_{ij}(k) = \sqrt{[x_i(k) - x_j(k)]^2 + [z_i(k) - z_j(k)]^2} \ge r_{safe} \tag{5}$$
$$(\forall i, j = 1, 2, 3, \ldots, N_V)$$

3.2. Modeling of Conflict Management

In this study, a distributed control model is adopted for UAVs to communicate. For all the UAVs in a certain range, they broadcast their own predicted trajectories to each other. If at *k*th time, there

exist two or more UAVs with distance between them of less than the safe range, then these UAVs will be predicted as "conflicting". The predicted conflict UAV set is defined as below:

$$Cf_\varepsilon(k) = \left\{ (S_i, S_j) \Big| i, j \in \{1 \cdots N_v\}, \exists k \in \left\{1 \cdots t_f\right\}, r_{ij}(k) < r_{safe} \right\}$$
$$\varepsilon \in \left\{1 \cdots N_{Cf}\right\} \tag{6}$$

where, N_{Cf} is the number of conflict groups at kth time, S_i, S_j are the predicted trajectories of UAVs with conflict, as shown in Figure 2. Thus, all the resolutions are calculated and maintained by the members in the coordination group independently, based on the information broadcast by the group members.
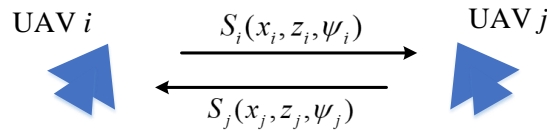

Figure 2. Information exchange of distributed UAV control system.

The concept of a state machine is used to model the UAV flight process. The state machine is a mathematical model that represents a limited number of states and behaviors, such as transitions and actions between these states. It mainly consists of the following parts: states, events, actions, and transitions, as shown in Figure 3.
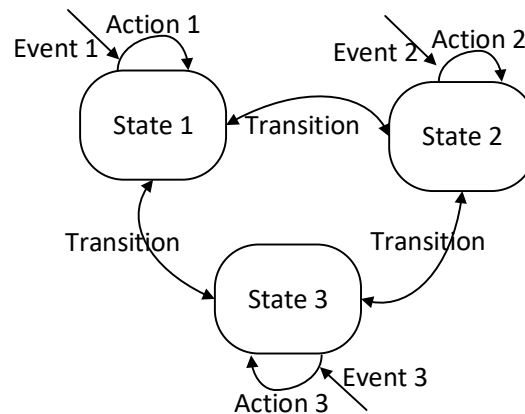

Figure 3. Principle of state machine.

Considering the issue of multi-UAV trajectory coordination [19], suppose that UAVs in flight have only two states, namely: (1) flying towards the target mission point; (2) performing adjustment for the conflict resolution. To describe the switching condition of the state machine between these two states [20], a reference early warning distance variable r_{dec} is defined. As shown in Figure 4, when the distance between two UAVs is less than r_{dec}, the UAV will change from the cruising state to the adjustment state, as shown in Figure 5.
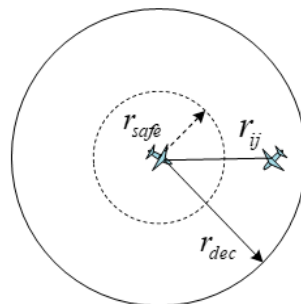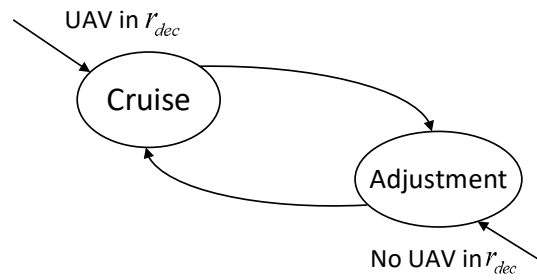

Figure 4. Reference distance of UAV.

Figure 5. State Machine of UAV flight.

Therefore, the process for establishing a multi-UAV conflict group is given as follows:

Process 1:

Step1. At kth time, get all UAV state variables:

$$\boldsymbol{x_i}(k) = [x_i(k), z_i(k), \psi_i(k)], \quad i = 1, 2, 3, \ldots, N_V;$$

Step2. For UAV i, identify all UAV j so that

$$\sqrt{[x_i(k) - x_j(k)]^2 + [z_i(k) - z_j(k)]^2} \leq r_{dec}$$

Construct the conflict set $Cf_\varepsilon = \left\{ S_i, S_j \right\}, (\varepsilon = i);$

Step3. If $Cf_\varepsilon = \varnothing$, then the UAV i flies towards the target point according to the original flight status; Otherwise, adjustment optimizations are performed on the UAV i and j;

Step4. For $i = i + 1; (\varepsilon = \varepsilon + 1)$, repeat Step 2–3, until the maximum number of UAVs.

3.3. Conflict Resolution Strategy Based on Synergetic Heading Angle Control Rule

After maintaining the predicted conflict groups, in order not to cause chain reactions of new conflicts, conflicting UAVs are sorted by their flight status [21]. Define the following rules: (1) The UAV with the earlier arrival at the conflict position has the higher priority; (2) The UAV closer to its target has the higher priority; (3) The UAV with higher speed has the higher priority. In the process of resolution, the UAV with the highest priority can fly straight to its target (if it does not cause subsequent conflicts), while the UAV with the lowest priority needs to receive the trajectory of the UAV with the highest priority as a reference track and maneuver until all conflicts are resolved.

To ensure that the two conflicting UAVs are able to handle the conflict autonomously, and to guarantee the consistency of the action, Ref. [15] introduced a control rule of synergetic heading angle for cruising UAVs with constant speed. During the process of conflict resolution, UAVs with higher and lower priorities (UAV H.P. and UAV L.P.) are commanded to turn to the same side of their current directions under this rule. Denote $\Delta\psi = \psi_L - \psi_H$, shown in Figure 6, the rule can be described as: if $\Delta\psi \in [0, \pi]$, then increase the heading angles of both UAVs; if $\Delta\psi \in [\pi, 2\pi]$, then decrease the heading angles of both UAVs.
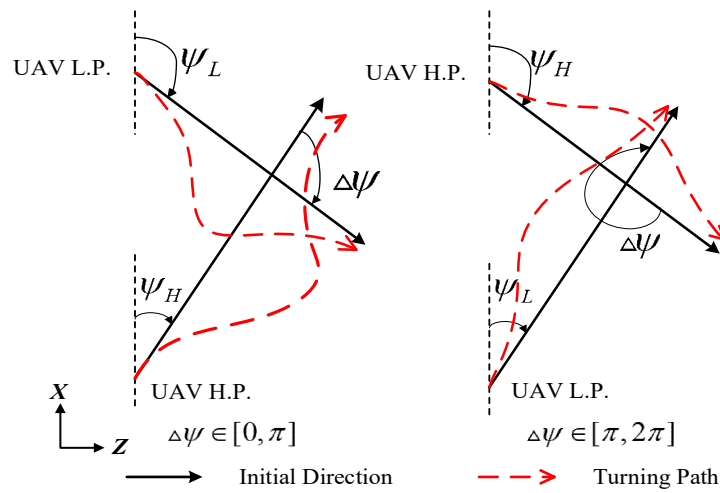
Figure 6. Synergetic heading angle control rule.

4. Conflict Resolution Algorithm Based on DNMPC

Model predictive control is a model-based finite time domain open-loop optimal algorithm. The current optimal control is executed in the next sampling period, and the process is repeated. In this study, a distributed nonlinear MPC algorithm (DNMPC) is introduced to coordinate the conflicting UAVs' trajectories.

4.1. Performance Indicators for Conflict Resolution

When executing tasks in airspace without threats, UAVs should fly along their own line with the mission in order to complete the mission as quickly as possible. Therefore, in the conflict resolution problem, the UAV should adjust its trajectory as little as possible while meeting the collision avoidance requirements [22]. In this paper, the normalized performance indicators are designed as follows:

(1) Total flight distance indicator:

$$dis_i = \|x_i(k) - x_{it}, z_i(k) - z_{it}\|_2$$
$$J_{i1}(k) = \sum_{i=1}^{N_V} (dis_i / dis_0) = = \sum_{i=1}^{N_V} (\|x_i(k) - x_{it}, z_i(k) - z_{it}\|_2 / \|x_i(0) - x_{it}, z_i(0) - z_{it}\|_2) \tag{7}$$

where, $P_{ti} = [x_{ti}, z_{ti}]$ represents for the target of UAV i.

(2) Trajectory adjustment cost indicator:

$$c_i(k) = |\psi_i(k) - \psi_i(k-1)| / |\psi_i(k-1)| + |v_i(k) - v_i(k-1)| / |v_i(k-1)|$$
$$J_{i2}(k) = \sum_{i=1}^{N_V} c_i(k) \tag{8}$$

(3) Conflict penalty function: Rewrite Constraint (5) as a penalty function for conflict:

$$f_i(\mathrm{k}) = \sqrt{[x_i(k) - x_j(k)]^2 + [z_i(k) - z_j(k)]^2} - r_{safe} \geq 0$$
$$F_i(\mathrm{k}) = \begin{cases} 0 & f_i(\mathrm{k}) \geq 0 \\ |f_i(\mathrm{k})| & f_i(\mathrm{k}) < 0 \end{cases} \tag{9}$$

In summary, the entire performance indicator for conflict resolution in period $[k, k+N]$ can be given as:

$$\min_{\mathbf{U}_i} J_i = \sum_{q=1}^{N_k} [J_{i1}(k+q) + J_{i2}(k+q) + \rho \cdot \mathrm{F}_i(\mathrm{k})] \tag{10}$$

where, $\mathrm{F}_i(\cdot)$ is the conflict penalty function and ρ is the coefficient for the penalty function.

4.2. Controller Design of DNMPC

Figure 7 is the solution framework for DNMPC conflict resolution controller. k is the current moment, and N is the prediction time domain and the control time domain length. $[x_i(k+1|k), \ldots, x_i(k+N|k)]$ is the predicted state vector of UAV i, $[x_{ir}(k+1|k), \ldots, x_{ir}(k+N|k)]$ is the reference state vector for UAV i, $[U_i(k|k), \ldots, U_i(k+N-1|k)]$ is the input sequence of control vector. Each UAV solves its own optimization problem independently and communicates with other UAVs only when necessary; thus, this framework describes a distributed controller [14]. Meanwhile, the analysis and justification of closed-loop stability can be referred to in Refs. [23,24].
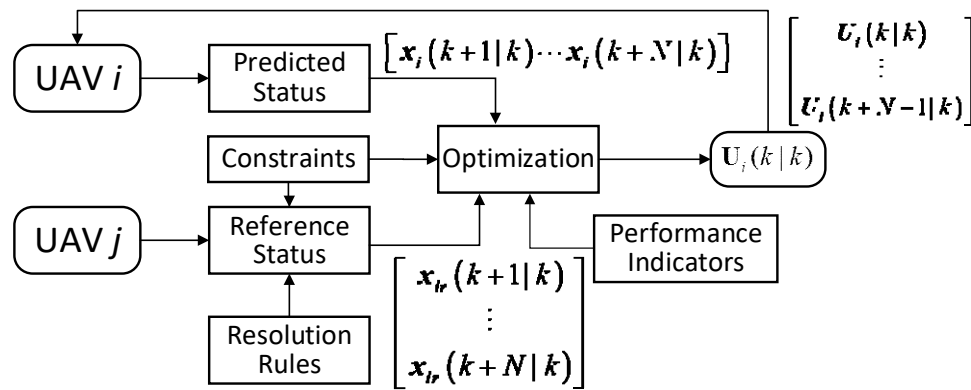

Figure 7. Solution framework for DNMPC.

4.3. Algorithm Description for Conflict Resolution

According to the idea of distributed MPC and receding horizon optimization, for the multi-UAV conflict resolution problem, the specific collision avoidance and trajectory coordination algorithm for any given UAV can be described as follows:

Algorithm 1:

Step1.   Set the sampling time k, the status of UAV i is $x_i(k)$. The current input control vector is $u_i(k)$, in the period of prediction $[k, k + N]$, based on the model of UAV, the predicted flight status can be given as:

$$\widetilde{X}_i(k + N) = [x_i(k + 1|k), x_i(k + 2|k), x_i(k + 3|k), \dots, x_i(k + N|k)]^T;$$

Step2.   If conflict group for UAV i $Cf = \phi$, UAV i need no trajectory coordination; else, set $j \in Cf$, denote the predicted flight status of UAV j as:

$$\widetilde{X}_j(k + N) = \left[x_j(k + 1|k), x_j(k + 2|k), x_j(k + 3|k), \dots, x_j(k + N|k)\right]^T;$$

Step3.   According to $\widetilde{X}_j$ and the resolution rule in Section 3.3, combined with Constraints (4) and (5), plan a conflict-free reference path for UAV i, then generate the predicted reference status:

$$\widetilde{X_{ir}}(k + N) = [x_{ir}(k + 1|k), x_{ir}(k + 2|k), x_{ir}(k + 3|k), \dots, x_{ir}(k + N|k)]^T$$

Step4.   According to $\widetilde{X_{ir}}$ and $\widetilde{X}_i$, combined with constraints (4) (5) and performance indicators (10), solve the optimal control vector $U_i = [u_i(k|k), u_i(k + 1|k), \dots, u_i(k + N - 1|k)]$. Then feedback optimal control vector U_i to UAV i.

Step5.   Execute the first item $u_i(k|k)$ in U_i, then get the status of UAV i in $k+1$th time; and again sends the updated predicted flight status $\widetilde{X}_i{}'$ to other associated UAVs.

Step6.   If UAV i has reached the specified state, i.e., it reaches the specified target point, the calculation is ended; otherwise, let $k = k + 1$, and repeat Steps 1–5.

In the resolution process, if some UAVs reach the designated target point earlier than other UAVs, then these UAVs are removed from the optimization list, that is, the next round of process will not consider these UAVs. The flow chart of the process for the conflict resolution algorithm is constructed as Figure 8:
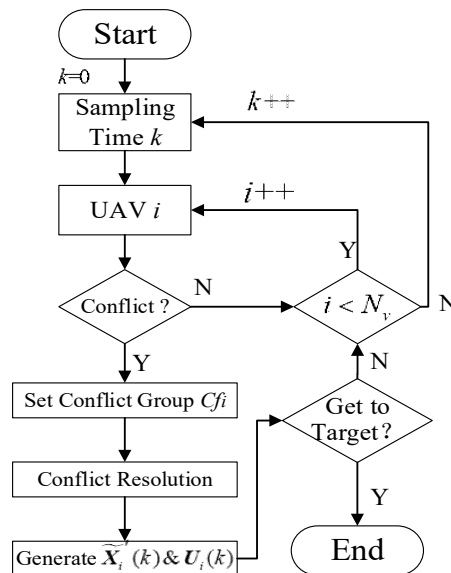

Figure 8. Flow chart of the conflict resolution algorithm.

5. Simulation Results and Analysis

To verify the performance of the discussed conflict resolution algorithm, numerical simulation results and analysis are presented in this section. The whole simulation system and the UAV objects

were built using Visual C++ (ver. 10.0), and the results figures were created using MATLAB. The entire simulation was performed on a workstation consisting of a 3.5 GHz Intel Core-i7 CPU and 16 GB of physical RAM running 64-bit Microsoft Windows 10. The sampling interval Δt of DNMPC is set as 1 s, the maximum iteration step is set as 200, while the prediction time domain is set as 10 steps. The formulation and parameters tuning method are referred to the traditional form of DMC mentioned in Refs. [25,26]. It is assumed that the UAVs obtain their state information completely accurately, that is, the status acquired by the UAVs at the current time in the simulations is equal to the current status of the calculation output of DNMPC at the previous time.

　　　Simulation 1: 3 cruising UAVs rendezvous, requiring conflict resolution. Supposing that all 3 UAVs have the same characteristics of flight, the initial conditions and other simulation parameters are listed as Table 1. The safe distance r_{safe} = 3 km and detection distance r_{dec} = 8 km. Since the UAVs are cruising, their velocities remain constant, and the synergetic heading angle control rule is used in this simulation. In this simulation, the maximum and average cycling time of DNMPC is 0.47 s and 0.31 s, respectively, which is less than $\Delta t = 1s$. The conflict resolution results are shown in Figures 9–11.

Table 1. Initial conditions and parameter values for Simulation 1.

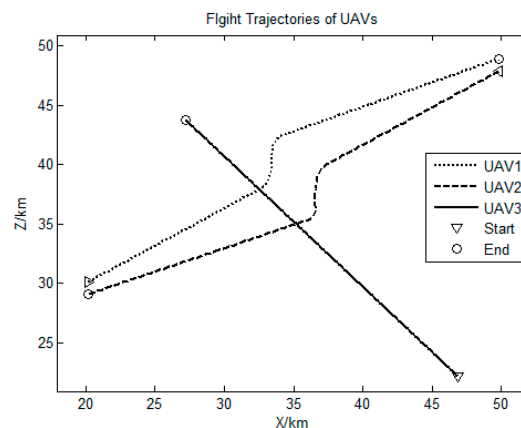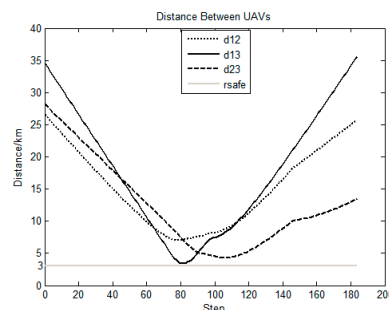| UAV	Start Points/km	Destination/km	Velocity/m·s^{-1}	ψ/deg	$\dot{\psi}$/deg s^{-1}
1	(20, 30)	(50, 49)	180	32.5	(−10,10)
2	(50, 48)	(20, 29)	180	212.4	(−10,10)
3	(45, 20)	(25, 42)	200	132.3	(−10,10)

Figure 9. Fight trajectories of 3 UAVs.

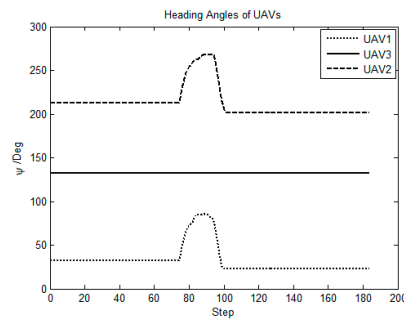**Figure 10.** Distance between UAVs.

Figure 11. Heading angles of 3 UAVs.

According to the initial conditions, the potential conflicts are detected as (UAV1-3,69s) and (UAV2-3,72s), which are within one prediction time domain. Thus, the DNMPC deals with the 3 UAVs in one conflict group, and UAV3 has the highest level due to its flight state. It is shown in Figures 9 and 11 that UAV3 flies past the conflict area and straight towards its destination, while UAV1 and UAV2 adjust their flight trajectories based on the synergetic heading angle control rule. Figure 10 shows that the distances between each of the UAVs are above the safe distance, which indicates that the adjustment plan calculated by DNMPC resolves potential conflicts sufficiently, and causes no further conflicts.

Simulation 2: Conflict resolution of 4 UAVs. In this simulation, the safe distance r_{safe} = 3 km and the detection distance r_{dec} = 15 km. The initial conditions and other simulation parameters are listed as Table 2, and the other settings of DNMPC is the same as in Simulation 1. In this simulation, the maximum and average cycling times of DNMPC are 0.66 s and 0.52 s, respectively, and are thus less than $\Delta t = 1s$. The conflict resolution results are shown in Figures 12 and 13.

Table 2. Initial conditions and parameter values for Simulation 2.

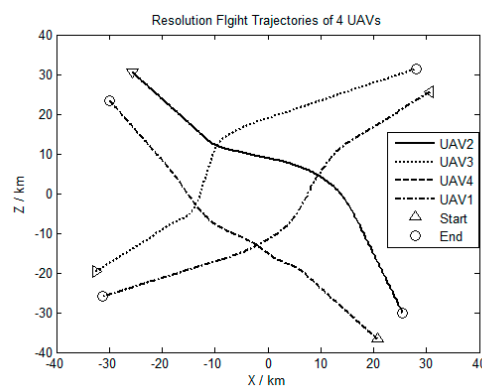| UAV	Start Points/km	Destination/km	Velocity/m·s^{-1}	ψ/deg	$\dot{\psi}$/deg s^{-1}
1	(30, 25)	(−31, −26)	200	219.9	(−10,10)
2	(−26, 31)	(25, −30)	200	310.9	(−10,10)
3	(−32, −20)	(28, 31)	200	40.3	(−10,10)
4	(21, −36)	(−30, 23)	200	130.8	(−10,10)

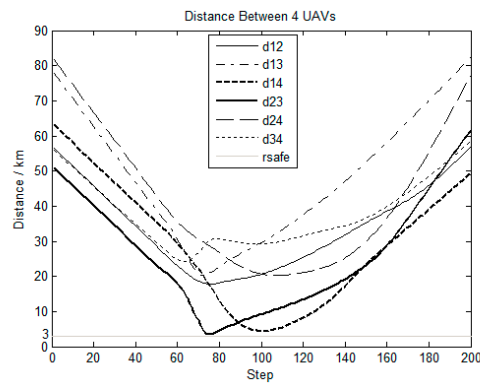**Figure 12.** Resolution flight trajectories of 4 UAVs.

Figure 13. Distance between UAVs.

The simulation results show that, according to the initial conditions, conflict occurs between each pair of UAVs. Thus, DNMPC manages all conflict UAVs into two conflicts groups: (UAV2-3,76s) and (UAV1-4,89s) based on the proposed resolution rule, and resolves the conflicts separately. It is shown in Figures 12 and 13 that the potential conflicts are resolved sufficiently and without any further conflicts.

To further demonstrate the merits of the proposed algorithm, regarding the conditions of Simulation 2, 150 Monte-Carlo simulations were carried out for comparison between the proposed method (denoted as A) and the method of Ref. [15] (denoted as B). Both methods were implemented under the same simulation settings. The following statistic indexes are listed in Table 3 for comparison: final performance indicator of equation (10) (FPI), total running time (TRT), and total number of further conflicts (TFC).

Table 3. Statistical indexes for both methods.

| Indicator | | FPI | | | TRT (s) | | | TFC (s) | |
Method	Best	Worst	Avg.	Best	Worst	Avg.	Best	Worst	Avg.
A	171.6	184.2	177.1	121.9	136.5	129.7	0	2	0.08
B	180.4	193.1	186.3	137.6	149.9	142.4	2	21	5.3

It is shown in Table 3 that the final performance indicator of the proposed method is better than B. Also, the proposed method resolved the problem in less time and caused much less further conflict than B, which indicates that the proposed method enjoys superiority in terms of effectiveness and computation efficiency compared with common methods.

6. Conclusions and Future Work

In this paper, we have presented a CR algorithm that solves potential trajectory conflicts among conflicting UAVs cooperatively under the frameworks of DNMPC. The proposed algorithm adjusts the path of each conflicting UAV by coordinating their flight status while maintaining their velocities in order to avoid the detected conflict. The main advantage of this algorithm lies in that it can generate stable and feasible solutions to the CR problem without creating any further conflicts. The proposed algorithms were validated on the basis of many numerical simulations under an environment without the influence of noise, communication delay and aero turbulence. Experimental results showed that the proposed algorithm is effective and performs better than traditional CR algorithm and method.

Future research should focus on the resolution of some undetected sudden conflict and the elimination of the influence of calculation and communication delay on the resolution quality. On the other hand, new models will be established to handle different sources of uncertainty in sensors, dynamic models, aero turbulence, etc. This uncertainty should be considered when predicting the aerial vehicle trajectory in the conflict detection.

References

1. Cai, G.; Wang, B.; Chen, B.M.; Tong, H.L. Design and Implementation of a Flight Control System for an Unmanned Rotorcraft using. *Asian J. Control* **2013**, *15*, 95–119. [CrossRef]
2. Conde, R.; Alejo, D.; Cobano, J.A.; Viguria, A.; Ollero, A. Conflict Detection and Resolution Method for Cooperating Unmanned Aerial Vehicles. *J. Intell. Rob. Syst.* **2012**, *65*, 495–505. [CrossRef]
3. Pallottino, L.; Feron, E.M.; Bicchi, A. Conflict resolution problems for air traffic management systems solved with mixed integer programming. *IEEE Trans. Intell. Transp. Syst.* **2002**, *3*, 3–11. [CrossRef]
4. Hwang, I.; Kim, J.; Tomlin, C. Protocol-Based Conflict Resolution for Air Traffic Control. *Air Traffic Control Q.* **2002**, *15*, 1–34. [CrossRef]
5. Masci, P.; Tedeschi, A. Modelling and Evaluation of a Game-Theory Approach for Airborne Conflict Resolution in Omnet++. In Proceedings of the International Conference on Dependability, Athens, Glyfada, Greece, 18–23 June 2009; pp. 162–165.
6. Vela, A.; Solak, S.; Singhose, W.; Clarke, J.-P. A mixed integer program for flight-level assignment and speed control for conflict resolution. In Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, Shanghai, China, 15–18 December 2009; pp. 5219–5226.
7. Omer, J. A space-discretized mixed-integer linear model for air-conflict resolution with speed and heading maneuvers. *Comput. Oper. Res.* **2015**, *58*, 75–86. [CrossRef]
8. Carbone, C.; Ciniglio, U.; Corraro, F.; Luongo, S. A Novel 3D Geometric Algorithm for Aircraft Autonomous Collision Avoidance. In Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, USA, 13–15 December 2006; pp. 1580–1585.
9. Durand, N.; Alliot, J.M. Ant Colony Optimization for Air Traffic Conflict Resolution. *J. Intell. Rob. Syst.* **2009**.
10. Lamont, G.B.; Slear, J.N.; Melendez, K. UAV Swarm Mission Planning and Routing using Multi-Objective Evolutionary Algorithms. In Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making, Honolulu, HI, USA, 1–5 April 2007; pp. 10–20.
11. Mujumdar, A.; Padhi, R. Reactive Collision Avoidance of Using Nonlinear Geometric and Differential Geometric Guidance. *J. Guidance Control Dyn.* **2015**, *34*, 69. [CrossRef]
12. Chaloulos, G.; Hokayem, P.; Lygeros, J. Hierarchical Control with Prioritized MPC for Conflict Resolution in Air Traffic Control. *IFAC Proc. Volumes* **2011**, *44*, 1564–1569. [CrossRef]
13. Stastny, T.J.; Garcia, G.A.; Keshmiri, S.S. Collision and Obstacle Avoidance in Unmanned Aerial Systems Using Morphing Potential Field Navigation and Nonlinear Model Predictive Control. *J. Dyn. Syst. Meas. Control* **2015**, *137*, 014503. [CrossRef]
14. Chaloulos, G.; Hokayem, P.; Lygeros, J. Distributed hierarchical MPC for conflict resolution in air traffic control. In Proceedings of the 2010 American Control Conference, Baltimore, MD, USA, 30 June–2 July 2010; pp. 3945–3950.
15. Izadi, H.A.; Gordon, B.W.; Zhang, Y. Rule-Based Cooperative Collision Avoidance Using Decentralized Model Predictive Control. In Proceedings of the Infotech@Aerospace Conference, St. Louis, MO, USA, 29–31 March 2011.
16. Albaker, B.M.; Rahim, N.A. Unmanned aircraft collision detection and resolution: Concept and survey. In Proceedings of the IEEE Conference on Industrial Electronics and Applications, Taichung, Taiwan, 15–17 June 2010.

17. Yoo, J.; Devasia, S. Application of provably-safe conflict resolution for air traffic control. In Proceedings of the 2012 IEEE 51st IEEE Conference on Decision and Control, Maui, HI, USA, 10–13 December 2012.

18. Anderson, R.P.; Milutinovic, D.A. Stochastic Approach to Dubins Vehicle Tracking Problems. *IEEE Trans. Autom. Control* **2014**, *59*, 2801–2806. [CrossRef]

19. Tomlin, C.; Mitchell, I.; Ghosh, R. Safety verification of conflict resolution maneuvers. *IEEE Trans. Intell. Transp. Syst.* **2001**, *2*, 110–120. [CrossRef]

20. Mayne, D.Q.; Rawlings, J.B.; Rao, C.V.; Scokaert, P.O.M. Constrained model predictive control: Stability and optimality. *Automatica* **2000**, *36*, 789–814. [CrossRef]

21. Kuriki, Y.; Namerikawa, T. Formation Control with Collision Avoidance for a Multi-UAV System Using Decentralized MPC and Consensus-Based Control. In Proceedings of the 2015 European Control Conference (ECC), Linz, Austria, 15–17 July 2015.

22. Menon, P.K.; Sweriduk, G.D.; Sridhar, B. Optimal Strategies for Free-Flight Air Traffic Conflict Resolution. *J. Guidance Control Dyn.* **1999**, *22*, 202–211. [CrossRef]

23. Sorcia-Vázquez, F.D.J.; Garcia-Beltran, C.D.; Valencia-Palomo, G.; Guerrero Ramírez, G.; Adam-Medina, M.; Escobar-Jiménez, R. Distributed Model Predictive Control Applied to a Four Interconnected Tank Process. Revista Iberoamericana de Automática eInformática Industrial RIAI. **2015**, *12*, 365–375. Available online: http://doi.org/10.1016/j.riai.2015.07.002 (accessed on 22 February 2019). (In Spanish)

24. Espinosa, J.; Marruedo, D.; Marquardt, W.; Schutter, B.; Valencia, F. A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark. *J. Process Control* **2011**, *21*, 800–815.

25. Garriga, J.L.; Soroush, M. Model Predictive Control Tuning Methods: A Review. *Ind. Eng. Chem. Res.* **2010**, *49*, 3505–3515. [CrossRef]

26. Gutiérrez-Urquídez, R.C.; Valencia-Palomo, G.; Rodríguez-Elias, O.M.; Trujillo, L. Systematic selection of tuning parameters for efficient predictive controllers using a multi-objective evolutionary algorithm. *Appl. Soft Comput.* **2015**, *31*, 326–338. [CrossRef]