applied sciences

MDPI

Article

Cooperative Multi-UAV Collision Avoidance Based on Distributed Dynamic Optimization and Causal Analysis

Mingrui Lao [1,2] and Jun Tang [1,3,*]

[1] Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China; mingrui_lao@163.com
[2] The School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China
[3] Department of Telecommunication and System Engineering, Universitat Autònoma de Barcelona, Sabadell 08201, Spain
* Correspondence: jun.tang@e-campus.uab.cat; Tel.: +34-688-113-447

Academic Editor: Josep M. Guerrero
Received: 1 December 2016; Accepted: 12 January 2017; Published: 17 January 2017

Abstract: A critical requirement for unmanned aerial vehicles (UAV) is the collision avoidance (CA) capability to meet safety and flexibility issues in an environment of increasing air traffic densities. This paper proposes two efficient algorithms: conflict detection (CD) algorithm and conflict resolution (CR) algorithm. These two algorithms are the key components of the cooperative multi-UAV CA system. The CD sub-module analyzes the spatial-temporal information of four dimensional (4D) trajectory to detect potential collisions. The CR sub-module calculates the minimum deviation of the planned trajectory by an objective function integrated with track adjustment, distance, and time costs, taking into account the vehicle performance, state and separation constraints. Additionally, we extend the CR sub-module with causal analysis to generate all possible solution states in order to select the optimal strategy for a multi-threat scenario, considering the potential interactions among neighboring UAVs with a global scope of a cluster. Quantitative simulation experiments are conducted to validate the feasibility and scalability of the proposed CA system, as well as to test its efficiency with variable parameters.

Keywords: collision avoidance; UAV; 4D trajectory; distributed dynamic optimization; casual analysis

1. Introduction

Recently, the unmanned aerial vehicle (UAV) has received a wide range of urban, civilian and warfare applications [1]. It offers several extraordinary features, especially the acceptance of long-endurance and high-risk missions that could not be reasonably performed by manned aircraft. For any assignment, a common problem is that the UAVs may encounter each other. Consequently, an effective collision avoidance (CA) system is a prerequisite for free flight.

There are several existing widely used or adequately evaluated CA systems for conventional aviation, and these can be applied on UAVs, such as the efficient Medium Term CA approach based on four-dimensional (4D) trajectories (trajectories defined in the three spatial dimensions together with a time-stamp) to resolve conflicts in a terminal maneuvering area (TMA) [2], the Traffic Alert and Collision Avoidance System (TCAS) equipped to issue advisories on how to maneuver vertically to prevent collisions [3], and the Airborne Separation Assurance System (ASAS) that enables the flight crew to maintain separation of an aircraft from one or more other aircraft and provides flight information concerning the surrounding traffic [4]. There are also several auxiliary systems that could provide the aircraft (or UAVs) with precise information about the state of the nearby traffic, such as the Automatic Dependent Surveillance Broadcast (ADS-B) system [5–7].

In addition, an extensive and meticulous survey of various methodologies ranging from abstract concepts to prototype systems for avoiding collisions between UAVs has been summarized in [4]. Typical techniques applied to escort UAV safety include the Monte Carlo approach [8], geometric optimization model [9], symbiotic simulation [10], probabilistic approach [11], consensus [12], evolutionary algorithm [13,14], Kalman filter [15], Markov decision process [16], Dubins curve based algorithm [17], Predictive Control [18], reachable sets based methods [19], and so on. Each method has its own advantages and characteristics, and no matter which is employed, it aims to guarantee minimum separation between UAVs. However, some of the studies only focus on the two-UAV scenarios (not multiple vehicles). Furthermore, most of these methodologies could not support exploring the emergent dynamics between the generated trajectories and the surrounding traffic. Thus, there is a need to check the detailed states of the CA process with a global view to achieve the optimal strategy, while it is essential to understand the cause–effect relationship of each CA action and how the effects of a maneuver impact on the neighboring aircraft. Note that tight couplings between aircraft trajectories in dense scenarios (i.e., flocks applications) can deal with the propagation of non-controllable upstream and downstream maneuvers between trajectories.

The overextended UAV usage of flock applications would lead to a higher number of threats, which could raise the risk of potential collisions. For simplicity, the CA problem is currently being discussed on UAVs (without pilot intervention) that are flying in a segregated airspace. In this paper, it is assumed that a group of cooperative UAVs, which are stabilized by the autopilot, are flying to their own destination and having to avoid collisions with other UAVs in an efficient real-time communication. Each vehicle is equipped with a trajectory control unit which relies on the CA algorithm that is used as a predictive control considering real-time data-link communication with surrounding traffic. In this research, we propose two efficient algorithms: conflict detection (CD) and conflict resolution (CR), which, as two sub-modules, constitute a novel CA system for multi-UAV. The main contributions of this paper are as follows:

- This paper presents an innovative strategy, distributed dynamic optimization approach (DDOA), to provide alternative trajectories for each UAV, by adding some essential constraints (i.e., performance, cost and distance) to make the problem treatable while the feasible trajectory generation for multi-UAV is a non-deterministic polynomial (NP) problem.
- The novel causal analysis is developed to choose a preferred resolution trajectory, performing an analysis of scenario state space in order to explore the dynamic evolution and then determine all reachable states. By decomposing the complex scenario into several common clusters, the solution space is greatly reduced.
- Quantitative measurement experiments are carried out to validate the feasibility and scalability of the CA system for the free flight of multi-UAV. In addition, the average computation time does not grow exponentially as the airspace density increases.

To prevent mid-air collisions between UAVs, the proposed CA system has been developed to serve as the last-resort safety net, which is an on-board CDR system providing resolution advisories. This paper is organized as follows: Section 2 illustrates the CA system functional architecture; Section 3 presents the core algorithm of the CD and CR sub-modules; Section 4 achieves the simulation results using the proposed system; finally, conclusions and future work are summarized in Section 5.

2. The Proposed CA System Architecture

The CA system should be supported by vigorous algorithms to increase the airspace capacity. Figure 1 illustrates the detailed fundamental processes of the complete CA system, in which the input is the 4D (three dimensions (3D) position + time) trajectory information belonging to each UAV; a high speed processor equipped with CD algorithm deals with the trajectories and obtains relevant information to predict whether a collision is going to occur in the future. If a potential collision is detected, DDOA computes the feasible solutions and causal analysis selects the optimal one. The control command is sent to the involved UAVs to resolve current threats and the states improve online.

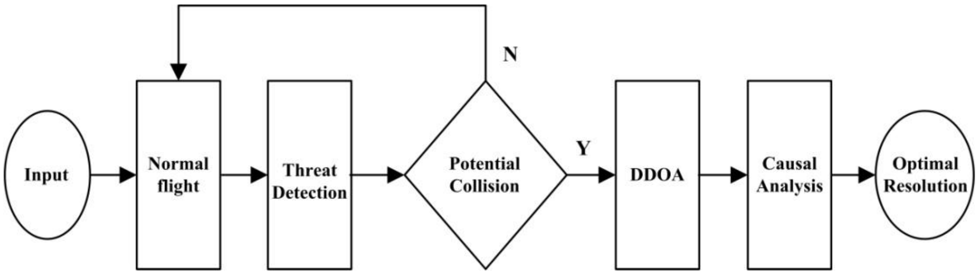

Figure 1. Functional architecture of the novel collision avoidance (CA) system.

The core of the operational CA system consists of two components:

- The CD module that analyzes the flight information depending on 4D trajectory has two different roles: generating the intuitionistic representation of threats and informing the CR module.
- The CR module integrates CD and resolves the detected conflicts in two stages: DDOA aims to avoid collisions by generating alternative trajectories for each UAV with a local optimization scope; causal analysis focuses on exploring the emergent dynamics between the alternative trajectories generated by DDOA and the rest involved trajectories. Based on the detected threats, the CR should be informed with the geographical coordinates where the collision would occur, the time when the collision would occur, and the involved UAVs. Though CR is a combinatorial problem in exponential growth with the increasing number of UAVs, in our research several strategies such as setting constraints are needed to narrow the solution space progressively to get the final optimal solution. Figure 2 conceptually illustrates the solution procedure. The DDOA generates several locally-optimal CA trajectories considering different constraints, e.g., the two UAVs should meet the performance requirements, satisfy the state equation, and keep a safe distance, etc. This causal analysis allows taking into consideration all the possible solutions that ensure the completeness of the solution space, and then reduces the space once again based on some specific constraints (e.g., the least cost).

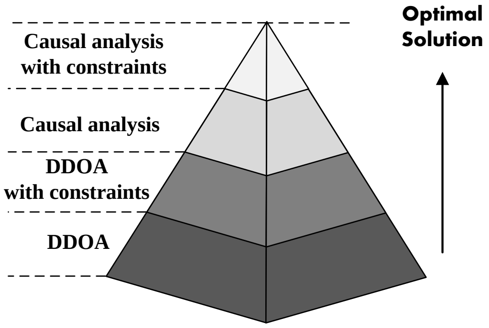

Figure 2. Step-by-step optimal process.

The hardware architecture of the novel CA system consists of the following major components: CA processor—performs airspace surveillance, intruder tracking, own UAV state tracking, conflict detection and conflict resolution, and it uses discrete status inputs of involved UAVs to control the CA logic parameters that determine the protection volume around own UAV; State transmitter—is used to provide air-to-air data exchange between UAVs so that coordinated, complementary CA trajectories can be issued when required; Control unit—guarantees the multiple cooperative UAVs following the generated optimal trajectory to resolve the conflicts.

3. Core Algorithm Analysis

3.1. Representation of 4D Trajectory and Conflict Detection

The continuous trajectory of UAV can be modeled as a sequence of discrete points (waypoints) expediently for computer processes. There is a direction between every two adjacent waypoints due to the velocity vector. The position of 3D and corresponding time—called spatial-temporal information—makes up the 4D trajectory of each UAV. For $UAVi\,(i = 1, 2, ..., n)$, its dynamic characteristics can be described in a Cartesian coordinate system [20]. The region formed by x and y axes indicates the horizontal plane, and z stands for the altitude.

$$p_t^i = \begin{bmatrix} x_t^i \\ y_t^i \\ z_t^i \end{bmatrix}, \ v_t^i = \frac{dp_t^i}{dt} = \begin{bmatrix} v_{x,t}^i \\ v_{y,t}^i \\ v_{z,t}^i \end{bmatrix} = \begin{bmatrix} v_t^i \cos\varphi_t^i \cos\theta_t^i \\ v_t^i \cos\varphi_t^i \sin\theta_t^i \\ v_t^i \sin\varphi_t^i \end{bmatrix} \\ \frac{d\varphi_t^i}{dt} = \omega_t^i, \quad 0 < \theta_t^i < 2\pi, \quad -\frac{\pi}{2} \le \varphi_t^i \le \frac{\pi}{2} \tag{1}$$

The formula defines p_t^i and v_t^i, respectively, as the position and velocity of *UAVi* in 3D; let θ_t^i represent the course angle, the orientation of velocity vector v_t^i in the x-y plane (measured from the x axis in counter-clockwise direction); let φ_t^i delegate the pitch angle, the orientation of velocity vector v_t^i in the vertical plane (measured from the horizontal plane up as positive and down as negative); and ω_t^i is the angular velocity.

Considered simply, each UAV is supposed to keep its own velocity $v_i, i \in \{1, 2, \cdots, n\}$ during regular flight. In the detection sub-module, the UAV that detects the conflict will read out the sensed data from sensor unit and check whether any intruder enters its safety zone. At each moment t, the relative distance between *UAVi* and nearby *UAVj* in the airspace can be simply calculated as:

$$R_t^{ij} = \sqrt{\left(x_t^i - x_t^j\right)^2 + \left(y_t^i - y_t^j\right)^2 + \left(z_t^i - z_t^j\right)^2} \tag{2}$$

In this research, the conflict detection logic can be defined as follows: if the current relative distance of the two UAVs is smaller than the R_{cf} ($R_t^{ij} < R_{cf}$), and they have a trend to be closer, i.e., when the value of R_t^{ij} that is the relative distance between *UAVi* and nearby *UAVj* is getting smaller, the resolution maneuver fires regardless of whether the coming closest point of approach (CPA) [21] is in collision volume R_{cl} or not. Despite the fact that this way certainly may touch off several unnecessary resolution maneuvers, the collision risk is highly reduced since the situation in whichthe intruder suddenly changes its direction near the CPA is prevented.

Let $p_t^{ij} = p_t^i - p_t^j$ be the distance vector in 3D space between *UAVi* and nearby *UAVj* at time t; let $v_t^{ij} = v_t^i - v_t^j$ be the relative velocity (closing speed) between the UAVs at time t. Distinctly, when the two UAVs are getting closer to each other (i.e., $p_t^{ij} \cdot v_t^{ij} > 0$), a conflict is detected and it has to be resolved immediately; when the two UAVs are getting further apart (i.e., $p_t^{ij} \cdot v_t^{ij} < 0$), there is no risk to deteriorate into a collision at all.

3.2. Conflict Resolution Algorithm

This section focuses on the CR algorithm consisting of two subsections: DDOA and causal analysis. Different alternative trajectories for each involved UAV are generated by DDOA in a collision, and the globally optimal one is selected by causal analysis considering the follow-up effects.

3.2.1. Distributed Dynamic Optimization Approach

The *UAVi* trajectory is discrete and the sampling interval is Δt. Therefore, in the m-th moment, the related position and pitch angle of *UAVi* are $x_i(m), y_i(m), z_i(m), \varphi_i(m), m \in N$. The discrete kinematic model of *UAVi* is:

$$\begin{bmatrix} x_i(m+1) \\ y_i(m+1) \\ z_i(m+1) \\ \varphi_i(m+1) \end{bmatrix} = \begin{bmatrix} x_i(m) + v_i \cos(\varphi_i(m)) \cos\theta_i \cdot \Delta t \\ y_i(m) + v_i \cos(\varphi_i(m)) \sin\theta_i \cdot \Delta t \\ z_i(m) + v_i(\sin\varphi_i(m)) \cdot \Delta t \\ \varphi_i(m) + \omega_t^i \cdot \Delta t \end{bmatrix} \tag{3}$$

In order to generate an amending trajectory for CA, it is better if each UAV could adjust its track as few times as possible, execute the trajectory as shorter as possible and reach the target point as soon as possible. Therefore, at time m, the objective function $O_i(m)$ of *UAVi* comprises three aspects above:

$$O_i(m) = w_1 C_i(m) + w_2 D_i(m) + w_3 T_i(m) \tag{4}$$

In this formula, w_1, w_2, w_3 are the corresponding weight coefficients of the three indexes $C_i(m)$, $D_i(m)$, $T_i(m)$. Their values reflect different application contexts, e.g., $w_3 > w_1$ indicates that the controllers prefer shorter time to achieve the destination even if the fuel cost is higher. Before the computation, normalization processing is needed. The three indexes can be calculated using the following formulas.

$$C_i(m) = \begin{cases} |\varphi_i(m) - \varphi_i(m-1)| & if\ m \geq 1 \\ 0 \quad if\ m = 0 \end{cases} \tag{5}$$

where $C_i(m)$ expresses the track adjustment cost of *UAVi*, which is defined as the change value of pitch angle.

$$D_i(m) = \|p_i(m) - p_i^d\|_2 = \left[(x_i(m) - x_i^d)^2 + (y_i(m) - y_i^d)^2 + (z_i(m) - z_i^d)^2\right]^{1/2} \tag{6}$$

where $D_i(m)$ indicates the distance from current location to its destination $p_i^d = (x_i^d, y_i^d, z_i^d)^T$.

$$T_i(m) = m \cdot \Delta t + D_i(m)/v_i \tag{7}$$

where $T_i(m)$ represents the total flight time of *UAVi.*

Assume that there are N intervals from time m to the destination, thus the position and corresponding pitch angle sequences of *UAVi* in $[m, m+N-1]$ are respectively:

$$\begin{aligned} & P_i \triangleq [p_i(m), p_i(m+1), \cdots, p_i(m+r), \cdots, p_i(m+N-1)]^T \\ & \Psi_i \triangleq [\varphi_i(m), \varphi_i(m+1), \cdots, \varphi_i(m+r), \cdots, \varphi_i(m+N-1)]^T \\ & \forall r \in [0, N-1] \end{aligned} \tag{8}$$

The objective function for progressive optimization is:

$$\min O_i(m+r) = \min\left(\sum_{r=0}^{N-1} (w_1 D_i(m+r) + w_2 T_i(m+r) + w_3 C_i(m+r))\right) \tag{9}$$

Note that it should meet the following constraints in the optimization process:

$$q(\varphi_i(m+r)) = \begin{bmatrix} 1 \\ -1 \end{bmatrix} C_i(m+r) - \begin{bmatrix} B \\ -A \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} (\varphi_i(m+r) - \varphi_i(m+r-1)) - \begin{bmatrix} B \\ -A \end{bmatrix} \leq 0 \tag{10}$$

$$e_i(p_i(m+r)) = \begin{bmatrix} x_i(m+r) - [x_i(m+r-1) + v_i \cos(\varphi_i(m+r-1)) \cos\theta_i \cdot \Delta t] \\ y_i(m+r) - [y_i(m+r-1) + v_i \cos(\varphi_i(m+r-1)) \sin\theta_i \cdot \Delta t] \\ z_i(m+r) - [z_i(m+r-1) + v_i(\sin\varphi_i(m+r-1)) \cdot \Delta t] \\ \varphi_i(m+r) - \varphi_i(m+r) \end{bmatrix} = 0 \quad (11)$$

$$d_i(p_i(m+r), p_j(m+r)) = d_0 - \|p_i(m+r) - p_j(m+r)\|_2 \leq 0 \quad (12)$$

In the above formulas, $q(\varphi_i(m+r)) \leq 0$ explains that the value change of *UAVi* pitch angle should be in the variation range of *(A,B)*; $e_i(p_i(m+r)) = 0$ shows that the calculation of *UAVi* position and pitch angle at time $m + r$ is based on the corresponding information at $m + r - 1$; $d_i(p_i(m+r), p_j(m+r)) \leq 0$ illustrates that the distance between *UAVi* and *UAVj* must not be shorter than the minimum safe distance d_0.

Integrating the objective function with constraints, the next waypoint $(p_i(m+1), \varphi_i(m+1))$ can be calculated based on the current state $(p_i(m), \varphi_i(m))$. Repeating the optimization procedure, the complete amending trajectory made up of a series of discrete waypoints is generated.

Besides, the algorithm for generating CA waypoints is robust enough to resolve successive threats. For example, if *UAV*1 encounters *UAV*2 and *UAV*3 seperately in a short interval, the two threats could be resolved via a compositive CA trajectory withonly a distance constraint between *UAV*1 and *UAV*3 added to the optimization calculation.This section calculates minimum deviation of the planned trajectory by an objective function integrated with track adjustment, distance, and time costs, taking into account the vehicle performance, state and separation constraints.

3.2.2. Causal Analysis

The size (computational complexity) of the conflict resolution problem is typically very large, which represents a real challenge to deal with the exponential growth of the state space in a realistic environment containing dozens of UAVs. A highly efficient technique to reduce the problem size is clustering [22]. The scenario can be decomposed into several sets of independent clusters based on the interaction between planned trajectories for a period of time. The partition principle is mainly in view of the distances between the neighboring UAVs. These independent clusters can be processed in parallel to significantly improve efficiency. Figure 3 is the concept illustration.

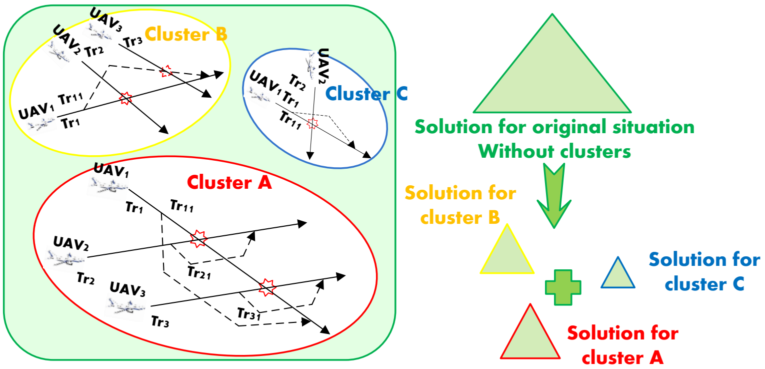

Figure 3. Concept illustration of clustering.

The original problem is decomposed into several clusters with treatable sizes with few trajectories. Thus, the overall solution is a combination of conflict-free solutions obtained by a causal analysis for these regional clusters. The core idea of the causal analysis is to explore the state space of the CA scenario to achieve the optimal resolution. It utilizes the alternative trajectories and relevant cost generated by the DDOA, which includes the track adjustment, distance, and time costs. Therefore, for n UAVs involved in the same scenario, the total cost of the trajectories amendment can be defined as:

$$Cost = \sum_{i=1}^{n} \sum_{r=0}^{N-1} C_i(m+r) + D_i(m+r) + T_i(m+r) \tag{13}$$

The process state could be described by a specific four-tuple:

$$\{conflict, cost, (active \quad UAV), (resolution \quad trajectories)\} \tag{14}$$

An instance state a = $\{2,1,(\underline{1},2,\underline{3}),(11,\underline{111},\underline{31},\underline{32})\}$ is used to explain the tuple.

- Conflict: indicates the threat that will be resolved (conflict 2 is active in state a).
- Accumulated cost: Cost summation for the respective resolution trajectories (cost = 1 in state a).
- Active UAVs: is a processed tuple and represents all the UAVs in this scenario, among them the active one would be underlined ($UAV1$ and $UAV3$ are active to have conflict 2 in state a).
- Active resolution trajectories: used to store solution trajectories and the active resolution trajectories are represented with underlines to resolve the active collision (111, 31 or 32 can be used to resolve conflict 2 in state a). Here the number is composed with the aircraft identification i and amending strategy descend/climb (1/2).

The causal exploration can be summarized with the following steps in Figure 4:

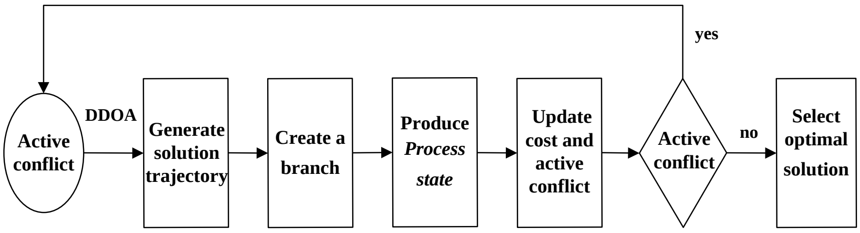

Figure 4. Causal exploration flow chart.

As an explanatory case, Figure 5 illustrates the whole feasible solutions of a case scenario, and Figure 6 shows its reachability tree. Using the causal exploration algorithm, there are six final process states as the feasible solutions. Evidently, state d is preferred to be utilized than the others due to the minimum cost.

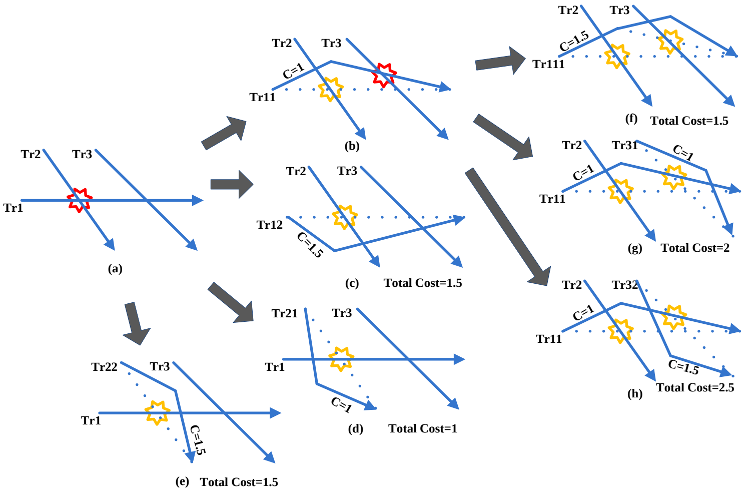

Figure 5. The feasible solutions.

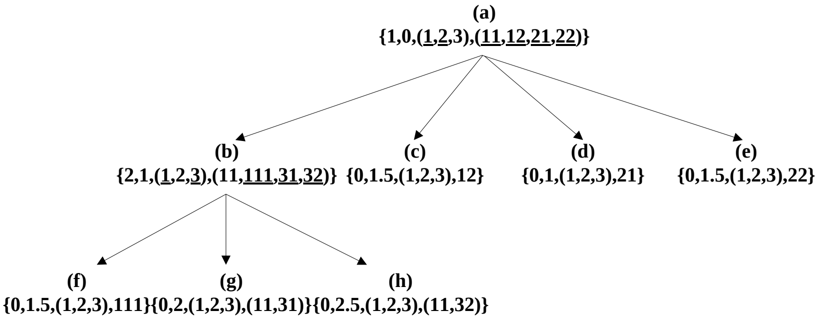

Figure 6. The CA reachability tree.

4. Simulation and Results

In this section, we carry out simulation experiments to evaluate the efficacy of the proposed algorithms for the CA system. The computational results represent collision-free maneuvers for multiple UAVs that are modeled with detailed dynamics.

4.1. A Case Scenario

To verify the feasibility of the proposed algorithms, our simulation experiments are performed on a manually generated scenario of five UAVs (with the corresponding trajectories Tr1, Tr2, Tr3, Tr4 and Tr5), illustrated in Figure 7. This scenario includes the following characteristics:

(1) It is a fairly complex scenario with multiple UAVs. Indeed, among the first three trajectories, Tr1 imposes sequent threats to both Tr2 and Tr3; moreover, an encounter emerges between Tr4 and Tr5, which further increases the complexity of the scenario.

(2) It involves the interrelationship (also called domino effect) between neighboring threats. For instance, a new secondary threat between $UAV3$ and $UAV5$ may emerge in the process of resolving their respective previous threat.

(3) It resolves successive threats via a compositive CA trajectory, i.e., $UAV1$ is detected to encounter $UAV2$ and $UAV3$ separately in a short continuous time period and the two successive conflicts can be considered overall rather than one by one.

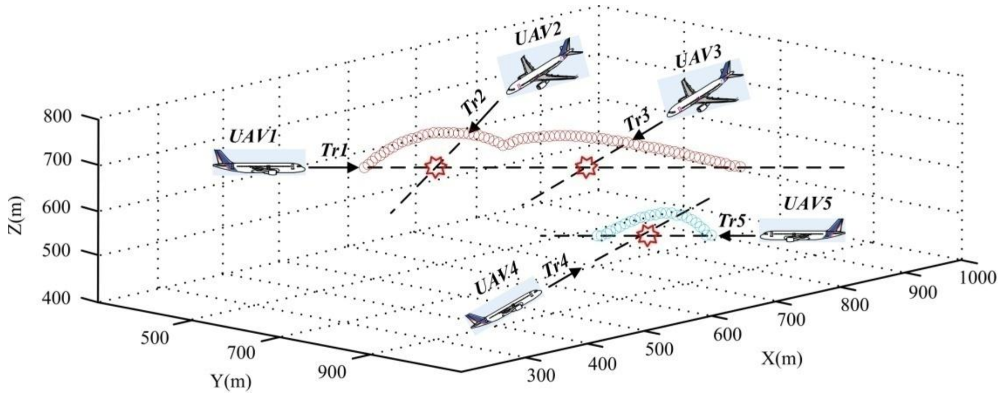

Figure 7. Five-UAV scenario with the generated optimal resolution.

In the above scenario, each UAV is assumed to be heading towards its own target position along the straight line. Simulations start with the initial information about the position and velocity of the UAVs in a scenario. For the local area, the UAV states have been treated as broadcast information with efficient real-time communication. The assumptions and the parameter setup values are given in

Table 1. The proposed algorithms are coded in C++ and the simulation is performed on an EliteBook laptop with a processor Intel i5 of 2.6 GHz and 4 GB of RAM.

Table 1. The assumptions of parameters and unmanned aerial vehicle (UAV) properties.

R_{cf} (km)	R_{cl} (m)	Velocity (m/s)	Δt (s)	A (rad)	B (rad)
1.2	100	30	1	−1.047	1.047
		$w_1 = w_2 = w_3$			

In this causal analysis, the cost of various feasible CA trajectories assists the decision making of solution selection. After executing the proposed CA system, the optimal strategy, that Tr1 synthetically amends upward and *UAV*5 climbs, is implemented.

The entirety of the reachable states of this multi-UAV scenario are displayed in Figure 8. From the proposed multi-threat resolution point of view, solo *UAV*1 climbs (Tr11) or *UAV*2 descends (Tr21) to resolve the first threat (see state a). However there would generate a secondary threat with Tr3 if trajectory Tr11 is applied (see state b). Then *UAV1* could climb again and a composite CA trajectory (Tr111) is generated (see state d). Meanwhile, *UAV*3 could descend (Tr31) to resolve the new encounter (see state e). For resolving the threat between *UAV*4 and *UAV*5, *UAV*4 amends downward (Tr41) or *UAV*5 turns upward (Tr51). In state d, there would be no domino effect between the neighboring UAVs (see state h and state i), while a new threat would appear between the descending *UAV*3 and climbing *UAV*5 (see state k) and have to be resolved through the following heading change (see state p and state q).

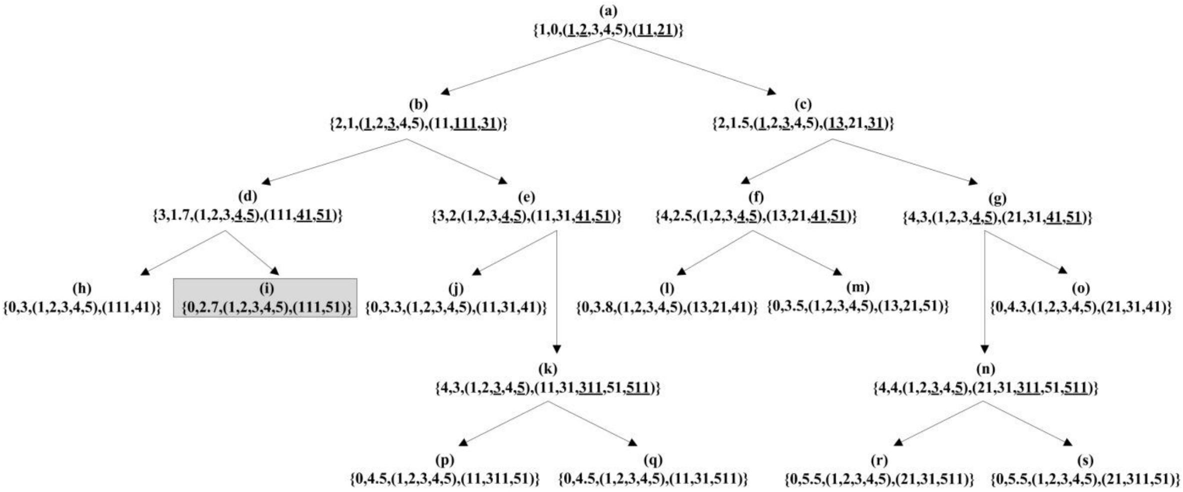

Figure 8. The reachability tree of this multi-UAV scenario.

On the other branch, the first threat is resolved by the amendment of Tr2 (see state c), and two options are considered to solve the encounter between *UAV*1 and *UAV*3 (see state f and state g). In state f, the collision between *UAV*4 and *UAV*5 could be avoided directly without domino effect (see state l and state m). However in state g, a new threat (see state n) would appear between the descending *UAV*3 and climbing *UAV*5 and it could be resolved by one of the involved trajectories (see state r and state s).

Since the cost criterion is taken into consideration in the decision-making process, it is clear that state i (*UAV*1 climbs twice for the sequent threats with *UAV*2 and *UAV*3, and *UAV*5 amends upward to avoid the collision with *UAV*4) is the best one of the feasible scenarios. The amended trajectories of *UAV*1 and *UAV*5are represented in Figure 7.

Table 2 summarizes the amended waypoints of *UAV*1 and *UAV*5 (for the sake of simplicity, the data are recorded every two seconds). *UAV*1 starts to change its headings at 17:17:48 and returns to the original trajectories at 17:18:18, while *UAV5* starts at 17:18:04 and ends at 17:18:24.

Table 2. *UAV*1 and *UAV*5 modified trajectories.

Time	ID	Amended Trajectory			Time	ID	Amended Trajectory		
		X (m)	*Y* (m)	*Z* (m)			*X* (m)	*Y* (m)	*Z* (m)
17:17:48	*UAV*1	370.43	510.68	669.57	17:18:10	*UAV*1	593.19	733.44	721.53
17:17:50	*UAV*1	387.85	528.10	686.68		*UAV*5	628.79	779.87	572.48
17:17:52	*UAV*1	406.44	564.10	701.14	17:18:12	*UAV*1	614.07	754.32	716.18
17:17:54	*UAV*1	425.92	566.17	713.00		*UAV*5	614.12	754.47	578.76
17:17:56	*UAV*1	446.44	586.69	720.60	17:18:14	*UAV*1	633.94	774.19	705.67
17:17:58	*UAV*1	467.54	597.79	723.66		*UAV*5	599.14	728.52	580.34
17:18:00	*UAV*1	488.67	628.92	721.03	17:18:16	*UAV*1	651.64	791.89	689.14
17:18:02	*UAV*1	509.24	649.50	713.75		*UAV*5	584.16	702.58	578.76
17:18:04	*UAV*1	529.85	670.10	720.90	17:18:18	*UAV*1	667.72	807.97	669.57
	*UAV*5	668.57	848.77	532.96		*UAV*5	569.50	677.18	572.48
17:18:06	*UAV*1	550.95	691.20	723.99	17:18:20	*UAV*5	555.03	652.12	564.58
	*UAV*5	656.73	828.26	551.38	17:18:22	*UAV*5	541.56	628.78	551.38
17:18:08	*UAV*1	572.14	712.39	725.25	17:18:24	*UAV*5	529.72	608.28	532.96
	*UAV*5	643.26	804.93	564.58					

Figure 9 depicts the relative distance of each pair of UAVs that are involved in a detected threat. From this figure, we can see that the minimum separation region of each UAV is never intruded upon, thanks to the favorable effect attained by the CA algorithm.

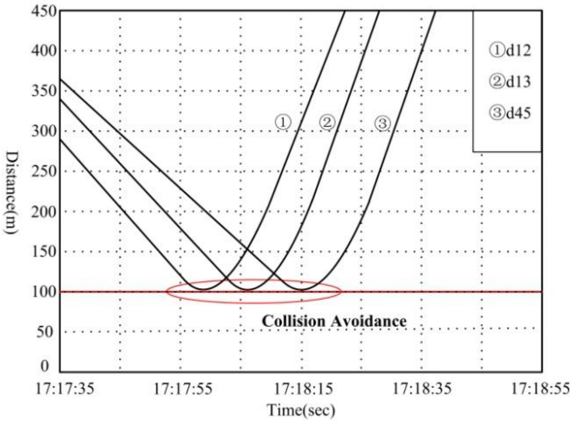

Figure 9. Relative distance.

Figure 10 illustrates the pitch angle profiles of *UAV*1 and *UAV*5 that amending their headings for the avoidance of corresponding threats in this scenario. At 17:17:48, *UAV*1 follows the control input that changes its pitch angle from 0 to 0.71 and climbs subsequently. At 17:17:58, *UAV*1 and *UAV*2 reach CPA and *UAV*1 begins to return to its original path. However, *UAV1* climbs again at 17:18:02 because of the encounter with *UAV*3. At 17:18:18, *UAV*1 recoveries its course when both the involved threats are resolved. And meanwhile, the neighboring collision between *UAV*4 and *UAV*5 is avoided via the upward amendment of *UAV*5. Using this CA advisory, there is no domino effect between the two sub-scenarios.

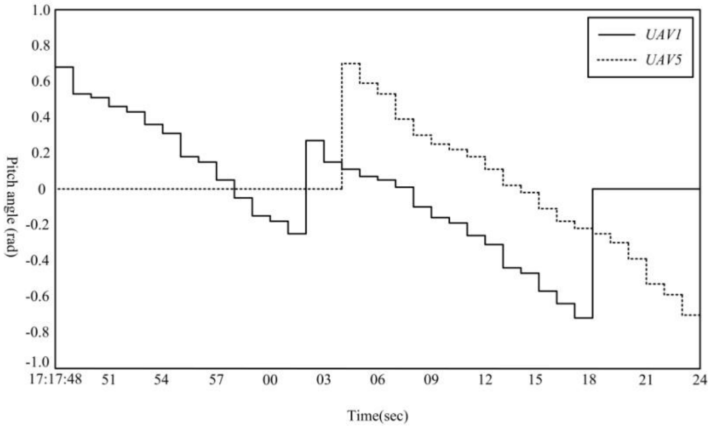

Figure 10. Pitch angle profile of both UAVs.

4.2. Further Investigation

In this section, we conduct additional experiments to test the scalability of the proposed algorithms. In [23], scalability is defined as the ability of an algorithm, system, or model to deal with increasing numbers of missions at a reasonable cost, or its ability to be effectively enlarged to commensurate that growth.

In practical applications, we may encounter different settings of UAVs, which depend on different values of densities, variable R_{cf} and sense range. For instance, with the increase of applications such as surveillance in which various UAVs owning different capabilities and properties would cooperate and compete for a certain target, it is expected that the airspace traffic density will grow considerably in certain reduced areas during short time periods. Since the CA system is applied independently to each cluster, the increasing UAV number in the whole airspace may improve the encounter possibility, thereby raising the number of clusters, rather than the number of UAVs involved in a specific cluster.

To validate the scalability, we use the computation time as a parameter to record the performance of the proposed CA system in handling different-density scenarios. Based on the simulation results in Table 3, the average computation time (over 100 runs of random flight tests) for low density (20 UAVs), medium density (60 UAVs), and high density (100 UAVs) traffic do not grow exponentially as the airspace density increases, attributing the high scalability to the independent clusters, which can be frequently processed.

Table 3. Time taken in different densities.

UAV Number	Computation Time (s)
20	0.92
60	5.28
100	17.62

In addition, several extensive simulations with varying values of tuning parameters R_{cf} and sense range (SR) are constructed to observe the performance in terms of efficiency. The CA maneuvers should be optimal so that UAVs can arrive at their respective destination in minimum possible time. Define t_i^e as the expectant flight time (not to make any deviation of the planned trajectory) and t_i^r as the realistic flight time (fully follow the CA maneuvers) of $UAVi\ (i = 1, 2, .., n)$, and the efficiency is given as [24]:

$$Efficiency = \frac{1}{n}\sum_{i=1}^{n}\frac{t_i^e}{t_i^r} \tag{15}$$

Figure 11 depicts the simulation results of efficiency with variable parameters. We could conclude the following:

- The efficiency is decreasing as the R_{cf} value increases, because extensive conflict volume leads to more threats that would be detected by the CD algorithm, thus more waypoints have to be amended to avoid potential collisions resulting in the higher realistic flight time.
- Compared to SR = 9 km, the efficiency is lower when SR is equal to three kilometers with the same R_{cf}, because the restricting sensor range may ignore the neighboring threats that could be synthetically resolved, and lately detect the intruder UAV, causing the number of amended waypoints to be higher.
- With the same parameters of R_{cf} and sense range, high efficiency is obtained if the complete CR algorithm including the causal analysis is utilized rather than only the DDOA. With the positive impact of causal analysis, the average increment of efficiency is respectively 2.83% and 3.47% for SR = 9 km and SR = 3 km. Consequently, the causal analysis is effective for the selection of optimal resolution trajectory.

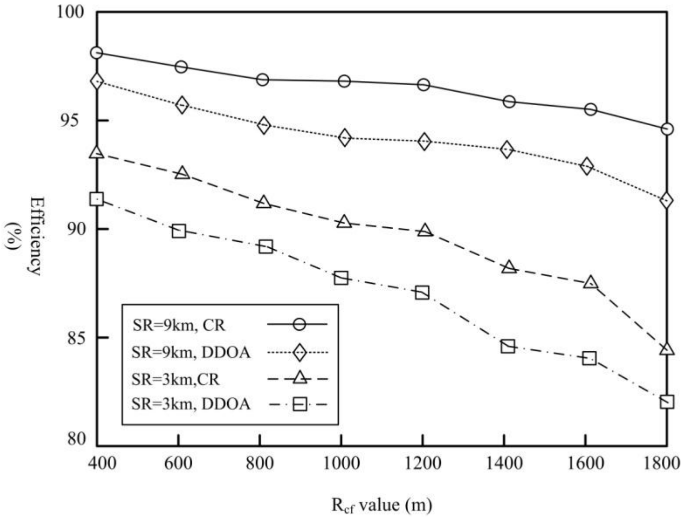

Figure 11. Variation of efficiency with R_{cf} and sense range for the traffic density of 20 UAVs.

5. Conclusions

This paper has presented two efficient algorithms: CD algorithm and CR algorithm, which constitute a novel CA system for multiple cooperative UAVs aiming to detect and resolve threats by generating an optimal trajectory with an overall minimum cost in 3D airspace. The CD module analyzes the spatial-temporal information composing a set of discrete 4D trajectories to generate the intuitionistic representation of detected conflicts. The CR algorithm integrates with the CD portion and solves the detected threats through DDOA and causal analysis. The DDOA provides new feasible alternative trajectories for each UAV involved in an encounter with a local optimization scope, and it generates alternative avoiding trajectories by an objective function (integrating track adjustment, cost, distance, and time) in consideration of several constraints (i.e., performance, state, and distance). The causal analysis explores the emergent dynamics between the alternative trajectories generated by DDOA and the surrounding traffic. It has numerous advantages such as splitting the scenario to separate clusters, reducing the size of solution space, and exploring the global optimal solution. Based on the quantitative simulation results, the proposed CA system has been validated to be feasible and effective in resolving conflicts for multiple UAVs and has high scalability for different traffic densities; the efficiency decreases as the value of conflict volume increases, and increases as the sense range increases; the causal analysis is effective for the selection of an optimal resolution trajectory.

Future research will be dedicated to: (1) using a heuristic-based algorithm to improve our proposed approach, which could tackle the collision avoidance problem of more UAVs in high-density scenarios, while it also may require more computation time; (2) making a comprehensive review of the existing best performing methods that consider UAV factors and integrating them to our system in order to improve its performance; (3) considering several disturbances (e.g., wind) to improve the current system logic in order to hold the complex field situations; (4) integrating UAVs into(manned) general aviation to operate in a non-segregated airspace.

Acknowledgments: The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China (71601181), the Military Scientific Research Project (162017160500A21XXX), and the State Key Laboratory of Management and Control for Complex Systems-Open Project (20160109).

Author Contributions: Mingrui Lao and Jun Tang designed the distributed dynamic optimization approach and causal analysis, conducted the simulations of the core algorithms and co-wrote the initial manuscript. Besides, Jun Tang helped improve the system design and put forward several construction suggestions. The final manuscript was read and approved by all authors.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

3D	Three dimensional
4D	Four dimensional
ADS-B	Automatic Dependent Surveillance Broadcast
ASAS	Airborne Separation Assurance System
CA	Collision avoidance
CD	Conflict detection
CPA	Closet point of approach
CR	Conflict resolution
DDOA	Distributed dynamic optimization approach
NP	Non-deterministic polynomial
TCAS	Traffic Alert and Collision Avoidance System
TMA	Terminal Maneuvering Area
UAV	Unmanned aerial vehicle

References

1. Borra-Serrano, I.; Peña, J.M.; Torres-Sánchez, J.; Mesas-Carrascosa, F.J.; López-Granados, F. Spatial Quality Evaluation of Resampled Unmanned Aerial Vehicle-Imagery for Weed Mapping. *Sensors* **2015**, *15*, 19688–19708. [CrossRef] [PubMed]
2. Ruiz, S.; Piera, M.A.; Del Pozo, I. A medium term conflict detection and resolution system for terminal maneuvering area based on spatial data structures and 4D trajectories. *Transp. Res. C Emerg. Technol.* **2013**, *26*, 396–417. [CrossRef]
3. Netjasov, F.; Vidosavljevic, A.; Tosic, V.; Everdij, M.H.; Blom, H.A. Development, validation and application of stochastically and dynamically coloured Petri net model of ACAS operations for safety assessment purposes. *Transp. Res. C Emerg. Technol.* **2013**, *33*, 167–195. [CrossRef]
4. Brooker, P. Airborne Separation Assurance Systems: Towards a work programme to prove safety. *Saf. Sci.* **2004**, *42*, 723–754. [CrossRef]
5. Lai, C.P.; Ren, Y.J.; Lin, C. ADS-B based collision avoidance radar for unmanned aerial vehicles. In Proceedings of the IEEE MTT-S International Microwave Symposium Digest, Boston, MA, USA, 7–9 June 2009; pp. 85–88.
6. Lin, Y.; Saripalli, S. Sense and avoid for Unmanned Aerial Vehicles using ADS-B. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 6402–6407.
7. BaraldiSesso, D.; Vismari, L.F.; Camargo, J.B. An approach to assess the safety of ADS-B based unmanned aerial systems. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 669–676.

8. Wolf, T.B.; Kochenderfer, M.J. Aircraft collision avoidance using Monte Carlo real-time belief space search. *J. Intell. Rob. Syst.* **2011**, *64*, 277–298. [CrossRef]
9. Tang, J.; Fan, L.; Lao, S. Collision Avoidance for Multi-UAV Based on Geometric Optimization Model in 3D Airspace. *Arabian J. Sci. Eng.* **2014**, *39*, 8409–8416. [CrossRef]
10. Holt, J.; Biaz, S.; Yilmaz, L.; Aji, C.A. A symbiotic simulation architecture for evaluating UAVs collision avoidance techniques. *J. Simul.* **2014**, *8*, 64–75. [CrossRef]
11. Prandini, M.; Hu, J.; Lygeros, J.; Sastry, S. A Probabilistic Approach to Aircraft Conflict Detection. *IEEE Trans. Intell. Transp. Syst.* **2000**, *1*, 99–220. [CrossRef]
12. Manathara, J.G.; Ghose, D. Rendezvous of multiple UAVs with collision avoidance using consensus. *J. Aerosp. Eng.* **2011**, *25*, 480–489. [CrossRef]
13. Mujumdar, A.; Padhi, R. Reactive Collision Avoidance of Using Nonlinear Geometric and Differential Geometric Guidance. *J. Guid. Control Dyn.* **2011**, *34*, 303–311. [CrossRef]
14. 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. Man Cybern. B* **2003**, *33*, 898–912. [CrossRef] [PubMed]
15. Luo, C.; McClean, S.I.; Parr, G.; Teacy, L.; De Nardi, R. UAV position estimation and collision avoidance using the extended Kalman filter. *IEEE Trans. Veh. Technol.* **2013**, *62*, 2749–2762. [CrossRef]
16. Ragi, S.; Chong, E.K. UAV path planning in a dynamic environment via partially observable markov decision process. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *49*, 2397–2412. [CrossRef]
17. Lin, Y.; Saripalli, S. Path planning using 3D Dubins Curve for Unmanned Aerial Vehicles. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 296–304.
18. Shim, D.H.; Sastry, S. An Evasive Maneuvering Algorithm for UAVs in See-and-Avoid Situations. In Proceedings of the 2007 American Control Conference, New York, NY, USA, 11–13 July 2007; pp. 3886–3891.
19. Lin, Y.; Saripalli, S. Collision avoidance for UAVs using reachable sets. In Proceedings of the 2015 International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015; pp. 226–235.
20. Netjasov, F.; Vidosavljevic, A.; Tosic, V.; Everdij, M.H.C.; Blom, H.A.P. Stochastically and dynamically coloured Petri net model of ACAS operations. In Proceedings of the 4th International Conference on Research in Air Transportation (ICRAT), Budapest, Hungary, 1–4 June 2010; pp. 449–456.
21. Kochenderfer, M.J.; Holland, J.E.; Chryssanthacopoulos, J.P. Next-generation airborne collision avoidance system. *Lincoln Lab. J.* **2012**, *19*, 17–33.
22. Ruiz, S.; Piera, M.A.; Nosedal, J.; Ranieri, A. Strategic de-confliction in the presence of a large number of 4D trajectories using a causal modeling approach. *Transp. Res. C Emerg. Technol.* **2014**, *39*, 129–147. [CrossRef]
23. Ross, A.M.; Rhodes, D.H.; Hastings, D.E. Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value. *J. Syst. Eng.* **2008**, *11*, 246–262. [CrossRef]
24. George, J.; Ghose, D. A reactive inverse PN algorithm for collision avoidance among multiple unmanned aerial vehicles. In Proceedings of the 2009 American Control Conference, St. Louis, MO, USA, 10–12 June 2009; pp. 3890–3895.