

Article

Affine Formation Maneuver Control for Multi-Agent Based on Optimal Flight System

Chao Kang *, Jihui Xu and Yuan Bian 

School of Equipment Management and Unmanned Aerial Vehicle Engineering, Air Force Engineering University, Xi'an 710043, China; skyline6018@126.com (J.X.); bianyuanlara@163.com (Y.B.)

* Correspondence: kc1015584136@163.com

Abstract: The use of affine maneuver control to maintain the desired configuration of unmanned aerial vehicle (UAV) swarms has been widely practiced. Nevertheless, the lack of capability to interact with obstacles and navigate autonomously could potentially limit its extension. To address this problem, we present an innovative formation flight system featuring a virtual leader that seamlessly integrates global control and local control, effectively addressing the limitations of existing methods that rely on fixed configuration changes to accommodate real-world constraints. To enhance the elasticity of an algorithm for configuration change in an obstacle-laden environment, this paper introduces a second-order differentiable virtual force-based metric for planning local trajectories. The virtual field comprises several artificial potential field (APF) forces that adaptively adjust the formation compared to the existing following control. Then, a distributed and decoupled trajectory optimization framework that considers obstacle avoidance and dynamic feasibility is designed. This novel multi-agent agreement strategy can efficiently coordinate the global planning and local trajectory optimizations of the formation compared to a single method. Finally, an affine-based maneuver approach is employed to validate an optimal formation control law for ensuring closed-loop system stability. The simulation results demonstrate that the proposed scheme improves track accuracy by 32.92% compared to the traditional method, while also preserving formation and avoiding obstacles simultaneously.

Keywords: multi-agent formation; affine; obstacle collision; virtual field; local control



Citation: Kang, C.; Xu, J.; Bian, Y. Affine Formation Maneuver Control for Multi-Agent Based on Optimal Flight System. *Appl. Sci.* **2024**, *14*, 2292. <https://doi.org/10.3390/app14062292>

Academic Editor: Yutaka Ishibashi

Received: 30 January 2024

Revised: 5 March 2024

Accepted: 7 March 2024

Published: 8 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The use of unmanned aerial vehicle (UAV) swarms [1,2] has significantly expanded the range of applications and possibilities in the unmanned field, surpassing the capabilities of single-vehicle systems. Formation-based control algorithms aim to guide all vehicles to form specific configurations while maintaining relative velocity and distance. However, relying on fixed formation patterns limits the adaptability of formation control in different environments. Hence, the notion of dynamically altering configurations in UAV formation control has garnered significant attention from researchers, particularly the maneuver control for the purpose of achieving continuous formation change [3–6]. This approach possesses the ability to autonomously navigate aerial formations in cluttered and wild environments, effectively providing assistance during natural disasters like earthquakes and floods. Maneuver control can aid in formation search and guidance [7]. Additionally, researchers can inspect restricted areas such as documenting structural and cultural artifacts within historical buildings [8] or autonomously searching subterranean environments [9], which can gain situational awareness and assist specialized personnel in specific missions, such as assessing the structural integrity of collapsed buildings, tunnels, mines, exploration of a newly discovered branch in a cave network, and searching for lost persons. Formation configurations can be associated with different tasks and environments, such as collective animal behavior [10] in nature or reconnaissance and strike missions [11,12]. However,

continuous configuration changes increase system complexity and communication requirements among UAV agents, particularly when adjusting and preserving the position and orientation of each agent in motion [13]. Additionally, the effectiveness of configuration change may be hindered by various external and internal constraints.

Therefore, various methods combining formation control with configuration changes have been extended to multi-agent systems, resulting in more versatile forms of unmanned vehicles, including maneuvering [13,14], surveillance [15], reconnaissance [16], and traffic monitoring and management [17]. In addition to researching practical constraints and scenarios, distributed methods have proven to be more adaptable and consider mutual autonomy coordination for configuration changes, in contrast to centralized approaches [3,17]. This suggests that the characteristics of the distributed strategy are better suited for the dynamic configuration changes of formations. Distributed formation control strategies closely related to configurations can be broadly categorized into position, displacement, and distance-based methods, based on the types of sensed and controlled variables used thus far. However, these methods have limited effectiveness in accurately tracking the target formation [18], which restricts configuration changes to variations in time, scales, and orientation of the target configuration [19–21]. From a control perspective, the dynamic change of configuration can be specified at the global formation level [19–22] or the local level [23,24]. The global approach involves presetting the path points of the formation's center of gravity for agents to follow, maintaining the rigid body structure by preserving relative positions and transitioning between different formation shapes. On the other hand, the local approach focuses on agent-level control, which is more adaptable to tasks and collision avoidance. The first approach involves maintaining the rigid structure of the configuration with constant constraints, requiring a configuration with a predefined change scale and nominal configuration. Ref. [6] proposed an affine-based generation method for target configuration using an interaction law with both designed positive and negative weights (different from consensus control law). Ref. [22] extended the affine-based method by introducing a stress matrix to enable general linear transformations involving translation, rotation, scaling, shear, or compositions of these transformations. These methods ensure that the configuration maintains the rigid graph structure in each step, preserving collinearity and distance ratios, while also enabling maneuver control. The second approach is more adaptable, aiming to maximize the observation capabilities of the robot team and facilitate obstacle avoidance. This method focuses on actively moving mobile agents to improve the accuracy of target detection with a dynamically adjustable geometry. Ref. [25] studied a formation loop that forms an un-rigid configuration using a distributed non-linear model predictive controller (DNMPC) and an estimator module. Ref. [26] discussed the use of artificial potential fields (APF) as a cost function in a non-linear model predictive controller (NMPC) with a modified A* path planner method for configuration changes. These methods exhibit generality, allowing adaptation to both static and dynamic environments, where the value and direction of velocity can be adjusted based on the current cost.

However, these methods have limitations in terms of generality. The first method fails to address sudden changes in an agent's velocity in terms of value and direction. The velocity is directly designed in each step of the configuration transformation, relying on prior information about the environment. The second approach heavily relies on strong communication interaction, which may lead to reconstruction failures in the event of breakdowns in one or more agents. Therefore, the goal of formation control with continuous configuration changes is to ensure low communication costs, an adaptable change model, as well as equilibrium and stability within the system.

Thus, this paper presents an algorithm that integrates APF-based leader control [27–30] with the affine formation, while utilizing behavior optimization for local trajectory calculation. This algorithm enables efficient navigation for multi-agents within a formation by incorporating artificial potentials to guide agent movement, a virtual leader for formation path planning, and geometric pattern modeling for obstacle avoidance. Moreover, the agents within the

formation are initially placed randomly and are adjusted applying additional forces to move individual agents to their new equilibrium positions in various configurations.

The objective of employing potential functions is to guide agents away from obstacles and towards or away from the virtual leader, while also ensuring cohesion among agents as they navigate through the environment [31–33]. In other words, the configuration of the formation can be adjusted based on the error in the velocity vector of agents and the virtual leader. Therefore, the potential functions used in our methodology consist of virtual leader forces and obstacle forces. As the virtual leader's position moves, the artificial potential relationships guide the group along the path defined by the virtual leader's motion. By relying on the global motion of a virtual leader, the complexity of planning multiple collision-free paths for numerous vehicles is reduced to planning just one collision-free path. Consequently, the interacting agents enter optimization procedures, where the explicit fourth-order Runge–Kutta methods [34] are introduced into the local control law. This method is commonly used to solve ordinary differential equations (ODE) and estimates the value of the next step by utilizing the slope of the midpoint with truncation error $O(L^4)$ in each step size. Subsequently, the local trajectory [35] of each agent estimates the value of the function at the next step by calculating the slopes of multiple intermediate points to obtain the numerical solution.

The virtual leader abstraction is utilized to guide the formation through the environment. It is important to note that the virtual leader is not a physical vehicle, but an imaginary point used as a reference for the movement of the formation. The construction of the virtual leader is similar to Leonard [3]. The virtual leader abstraction offers a significant advantage by eliminating the need for individual node trajectories, focusing solely on planning the path of the virtual leader through areas of interest for the entire team. This approach greatly reduces communication requirements. Additionally, the motion of the virtual leader is limited to a distance that prevents any of the vehicles from experiencing excessive total forces in the direction of the virtual leader's motion.

Furthermore, the communication requirements among agents may vary depending on the level of precision needed for configuration changes. Each agent updates its trajectory within the formation based on the positions of other agents, obstacles, and the virtual leader each time step. The position of the virtual only needs to be transmitted to each agent once per update at most, regardless of whether there is a need to recalculate the position to adjust the local trajectory. This approach ensures that the bandwidth requirements for all agents interacting with their positions remain manageable in small formations (less than 20 agents), as it considers only horizontal motion consensus. However, for larger groups, the position information would be limited to agents within a nearby radius.

This paper investigates the generality of formation-desired configurations and proposes an adaptable algorithm for local trajectory planning. The algorithm ensures the continuous change of geometric patterns and improves the consistency of local trajectory planning with realistic constraints. Consequently, the formation control relies less on prior information and enhances the ability to avoid obstacles and other agents in real-time, particularly in situations involving sudden obstacles. Additionally, the combination of formation control methods and optimal techniques expands the utilization of formation configuration variations and definitions to adapt to more complex environments.

The contributions of this article could be summarized as follows:

1. A novel non-linear and dynamical system for formation configurations changing is created by the combination of the global and local control law, which could relieve the dependence on the rigid body of formation.
2. An APF-based adjustment metric is developed for local trajectory planning through various forces. It could solve the problem of local minimum value of APF and update the local navigation to filter the discrete value due to a sudden change of velocity.
3. A novel optimal framework has been integrated, combining a virtual leader to consider dynamic feasibility, obstacle avoidance, and flexibility. This framework can improve the elasticity of configuration variations in potential collision.

The paper is organized as follows. Section 2 describes the preliminaries. In Section 3, the description of the interaction forces among agents, virtual leader, and obstacles is presented. The description and proof of the main results for local control laws are given in Section 4. The results of the simulation and discussions are drawn in Section 5. Section 6 concludes the entire paper and provides a glimpse into future prospects.

2. Preliminaries

This section mainly introduces the preliminary knowledge of unmanned aerial vehicles, containing graph theory to depict the inner relationship of the system and the state-space equation of the UAV system.

2.1. Graph Theory

A weighted undirected graph is denoted as $G = (V, E, A)$, where $V = (v_1, v_2, \dots, v_n)$ represents the set of all vertices in the undirected graph, and each vertex represents an agent. $E = \{e_{ij} = (v_i, v_j)\} \subseteq v \times v$ represents the edge set between vertices, and $e_{ij} = (v_i, v_j)$ represents the edge from v_i vertex to v_j vertex; an edge connection between two vertices indicates that there is an information interaction between these two vertices. The graph is undirected if it allows two-way communication; otherwise, it is directed. $A = [a_{ij}]_{n \times n}$ represents the adjacency matrix indicating the relationship between agents, where a_{ij} is the weight of the side $e_{ij} = (v_i, v_j)$ and where the diagonal elements of the matrix are all 0. For $i, j = 1, 2, 3, \dots, n (i \neq j)$, if the agents v_i and v_j can receive information from each other, then the elements in the adjacency matrix are $a_{ij} = a_{ji} > 0$; otherwise, the element in the adjacency matrix is 0. This paper has only considered an undirected graph, in which $e_{ij} \in E \Leftrightarrow e_{ji} \in E$.

2.2. Kinematic and Dynamic Models of UAV

The multi-agent formation is denoted as (G, p) , where the vertex v_i within graph G is mapped to point p_i . In addition, it is assumed that the first agents n_l are leaders and the rest n_f are followers. Let $V_l = \{v_1, v_2, \dots, v_l\}$ and $V_f = \{v_{l+1}, v_{l+2}, \dots, v_n\}$ be the sets of leaders and followers, respectively, where $l, f \in n$.

A mass-point [18,36] can be used to describe the UAV agent and the motion of the formation. The fixed-wing agent model is described as Figure 1 shows:

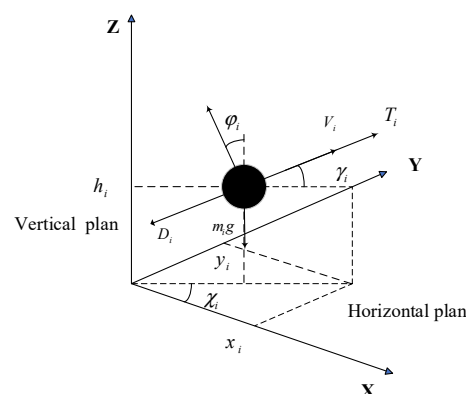


Figure 1. Fixed-wing agent model.

It is shown that all these variables are presented to depict the agent motion from the inertial coordinate frame (x, y, z) ; thus, the kinematic motion equations for UAV can be expressed as

$$\begin{aligned}\dot{x}_i &= V_i \cos \gamma_i \cos \chi_i \\ \dot{y}_i &= V_i \cos \gamma_i \sin \chi_i \\ \dot{h}_i &= V_i \sin \gamma_i\end{aligned}\quad (1)$$

Meanwhile, the dynamic of UAV is given by

$$\begin{aligned}\dot{V}_i &= \frac{T_i - D_i}{m_i} - g \sin \gamma_i \\ \dot{\gamma}_i &= \frac{g}{V_i} (n_{gi} \cos \phi_i - \cos \gamma_i) , \\ \dot{\chi}_i &= \frac{L_i \sin \phi_i}{m_i V_i \cos \gamma_i}\end{aligned}\quad (2)$$

where x_i and y_i are the range displacement projected on the X and Y axes separately, h_i is the flight altitude, and V_i is the speed which is assumed as ground speed in the paper, χ_i and γ_i are the heading angle and the flight one. T_i is the engine thrust vector directed with the velocity vector, D_i is a general term for drag. Additionally, $m_i g$ is used to describe the gravity of the agent, the g load $n_{gi} = L_i / m_i g$ is contained by an elevator, ϕ_i the banking angle and the engine thrust T_i are considered as the control inputs.

Therefore, by the use of the feedback linearization technology, the non-linear UAV model Formulation (1) and (2) can be simplified to a linear constant system

$$\begin{cases} x_i(t) = u_{xi}(t) \\ y_i(t) = u_{yi}(t) \\ h_i(t) = u_{hi}(t) \end{cases}\quad (3)$$

However, $u_{xi}(t)$, $u_{yi}(t)$, and $u_{hi}(t)$ are virtual control inputs to describe the accelerations in each way. The actual control input is given as

$$\begin{aligned}\phi_i &= \tan^{-1} \left[\frac{u_{yi} \cos \chi_i - u_{xi} \sin \chi_i}{\cos \gamma_i (u_{hi} + g) - \sin \gamma_i (u_{xi} \cos \chi_i - u_{yi} \sin \chi_i)} \right] \\ n_{gi} &= \frac{\cos \gamma_i (u_{hi} + g) - \sin \gamma_i (u_{xi} \cos \chi_i - u_{yi} \sin \chi_i)}{g \cos \phi_i} , \\ T_i &= [\sin \gamma_i (u_{hi} + g) + \cos \gamma_i (u_{xi} \cos \chi_i + u_{yi} \sin \chi_i)] m_i + D_i\end{aligned}\quad (4)$$

where the heading angle χ_i can be expressed as $\tan \chi_i = \dot{y}_i / \dot{x}_i$, the flight one γ_i is $\sin \gamma_i = \dot{h}_i / V_i$. Thus, in terms of the state-space representation, denoting $\mathbf{p}_i = [x_i \ y_i \ h_i]$, $\mathbf{v}_i = [\dot{x}_i \ \dot{y}_i \ \dot{h}_i]$, and $\mathbf{u}_i = [u_{xi} \ u_{yi} \ u_{hi}]$, which are the position, velocity, and acceleration control vector, the double-integrator system is reduced as

$$\begin{cases} \dot{\mathbf{p}}_i(t) = \mathbf{v}_i(t) \\ \dot{\mathbf{v}}_i(t) = \mathbf{u}_i(t) \end{cases}\quad (5)$$

where $\mathbf{p}_i \in R^n$, $\mathbf{v}_i \in R^n$, $\mathbf{u}_i \in R^n$. Consider a flock of n agents in R^d and $\mathbf{p}_i \in R^d$ as the position of agent i and $\mathbf{p} = [p_1^T, p_2^T, \dots, p_n^T] \in R^{dn}$ is the configuration of all the agents, where $d \geq 2$ and $n \geq d + 1$. Let m be the number of edges in an undirected graph and the orientation of which is the direction assignment to each edge.

Given the substantial difference in time constants between the horizontal and vertical trajectory controllers, configuration control can be concentrated primarily on achieving consensus in the horizontal positional and velocity motion. Meanwhile, the control input is composed of anticipated configuration, reference trajectory, velocity, and state information of adjacent entities.

It is presupposed that all followers possess a restricted situational awareness, and their capability to detect fellows is constrained by their sensing range. Supposing all Unmanned Aerial Vehicles (abbreviated as UAV) possess an equivalent efficacious sensing range, thus the neighbor set of agent i can be depicted as:

$$N_i = \{j : \|\mathbf{p}_{ij}\| < R, j = 1, \dots, n; j \neq i\}\quad (6)$$

where $\mathbf{p}_{ij} = \mathbf{p}_i - \mathbf{p}_j$ is the position vector from UAV i to UAV j , $\|\cdot\|$ is the Euclidean norm, R is the sensing range.

Let U_{ij} denote the artificial potential between UAV i and j , R_0 and R_1, R_2 are sensing range and desired distance, respectively. $\otimes$ is the Kronecker product and $\|\cdot\|$ as the

Euclidean norm of vector notation, $I_n \in \mathbb{R}^{n \times n}$ is the identity matrix of order n , and $\dim(\cdot)$ is the dimension of a linear space. For any vector x , $\text{diag}(x)$ denotes the diagonal matrix whose i th diagonal entry is the i th entry of x . Z_i denotes the total artificial potential of UAV i .

2.3. Affine Transformation Description

Based on the undirected graph G to describe the communication topology of multi-agent formation, denote the $R = [R_l, R_f]^T = [r_1, r_2, \dots, r_N]^T$ as the nominal configuration and let S be the affine span. Let the dimension be $d = 2$ and $P = \{p_i\}_{i=1}^n$ in $\mathbb{R}^2$ be the affine image set and the associated definitions and lemmas about affine transformation are presented as follow [22].

Definitions 1 (Affine Span). A linear space can be obtained by any affine span. Thus, provided with the nominal configuration $\{r_i\}_{i=1}^N$ for formation G , the affine span S is:

$$S = \left\{ \sum_{i=1}^N \alpha_i r_i : \alpha_i \in \mathbb{R}^2 \text{ and } \sum_{i=1}^N \alpha_i = 1 \right\} \quad (7)$$

Lemma 1 (Affine Span with Rank condition). The nominal configuration $\{r_i\}_{i=1}^N$ affinely span $\mathbb{R}^2$ if and only if $N \geq 3$ and $\text{rank}(\bar{R}(r)) = 3$, where $\bar{R}(r) = [R, 1_N] \in \mathbb{R}^{N \times 3}$.

Definitions 2 (Affine Image). The affine image set consisting of all the affine transformations of the nominal configuration $\{r_i\}_{i=1}^N$, which is defined as follows:

$$I(r) = \left\{ P = [p_1^T, p_2^T, \dots, p_N^T]^T \in \mathbb{R}^{2N} \mid p_i = A' r_i + b \right\} \quad (8)$$

where $I(R)$ is the predefined affine image and can be applied to all time-varying target configurations, and a linear subspace because it is closed under addition and scalar multiplication, $A' \in \mathbb{R}^{2 \times 2}$ and $b \in \mathbb{R}^2$ are the linear affine transformation.

Lemma 2 (General Rigid). The nominal configuration is universally rigid if and only if the stress matrix is positive semi-definite and satisfies $\text{rank}(\Omega) = N - 3$.

Definitions 3 (Stress Matrix). The stress matrix can be viewed as generalized graph Laplacian matrices with both positive and negative edge weights. Let graph stress denote as the scalar $\{\omega_{ij} \mid \omega_{ij} = \omega_{ji}, \omega_{ij} \in \mathbb{R}\}_{ij \in E}$, $\omega_{ij} > 0$ represents the attractive force and $\omega_{ij} < 0$ the opposite, and denote if:

$$\sum_{j \in N_i} \omega_{ij} (p_j - p_i) = 0 \quad (9)$$

the equilibrium set $\Omega \in \mathbb{R}^{N \times N}$ satisfies:

$$[\Omega]_{ij} = \begin{cases} 0, & i \neq j, (i, j) \notin E, \\ -\omega_{ij} & i \neq j, (i, j) \in E, \\ \sum_{k \in N_i} \omega_{ik} & i = j. \end{cases} \quad (10)$$

Equation (10) can be expressed in a matrix form as $(\Omega \otimes I_2)R = 0$, denote the $\bar{\Omega} = \Omega \otimes I_2$:

$$\bar{\Omega} = \Omega \otimes I_2 = \begin{bmatrix} \bar{\Omega}_{ll} & \bar{\Omega}_{lf} \\ \bar{\Omega}_{fl} & \bar{\Omega}_{ff} \end{bmatrix} \quad (11)$$

Definitions 4 (Affine Localizability). Denote the nominal formation (G, r) as *affinely localizable* if and only if $\{r_i\}_{i=1}^N$ affinely span $\mathbb{R}^2$ (Lemma 1) and any $p = [p_l^T, p_f^T]^T \in I(r)$, p_f^T can be uniquely determined by p_l^T .

Lemma 3 (Affine Localizability of Nominal Formation). If the nominal configuration $\{r_i\}_{i=1}^N$ affinely span $\mathbb{R}^2$ and Ω is positive semi-definite, (G, r) is affinely localizable if and only if Ω_{ff} is nonsingular.

From Figure 2, it can be seen that Figure 2b–d are obtained by rotating, scaling, and shearing the original formation Figure 2a, respectively. Figure 2e is obtained from Figure 2d by reducing the scale in the vertical direction. Figure 2f, where the four points are collinear, is obtained from Figure 2e by reducing the scale in the vertical direction to zero.

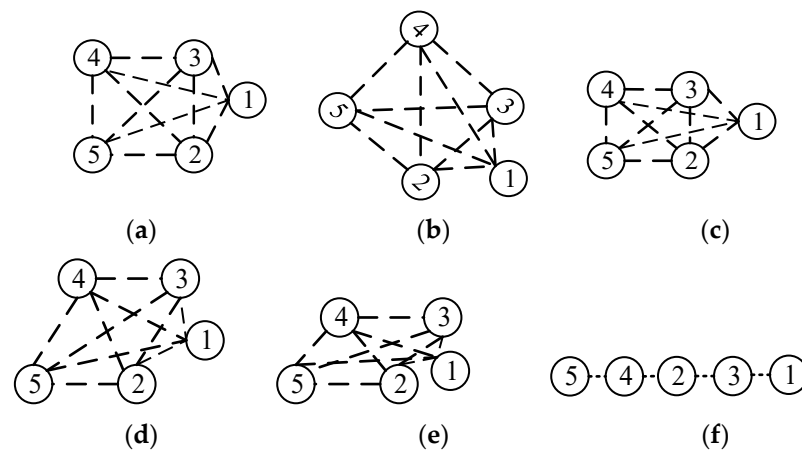


Figure 2. Affine-based predefined configurations patterns for illustration: (a) Nominal; (b) rotation and scaling; (c) scaling; (d) shear; (e) shear and scaling; (f) linear style. It should be noted that in this paper, the dotted line segments only represent the outlines of the shape, but not the connectivity of the graph. Agents are sequentially numbered from 1 to 5 (marked at the center of the agent).

Due to the simplification of the communication complexity, only leaders need to be aware of the affine transformation (A, b) and controlled only. Thus, the leader–follower strategy can be well adopted as the control law, and the leaders' positions will have a one-to-one correspondence to the affine transformation $[A(t), b(t)]$.

3. Agent Motion by Forces

In this section, we propose the concept of an artificial potential function (APF)-based virtual leader strategy with the extraordinary ability to abruptly alter the position of others, which is absolutely essential for executing intricate maneuvers within the swarm.

3.1. APF-Based Virtual Leader Forces

Previous research on affine-based formation control [21,22] primarily relies on prior information about obstacles. However, when realistic constraints are introduced into the control law, such as communication loss or sudden situations, the proposed algorithm may not successfully navigate. This is due to the unknown obstacles causing the loss of rigidity during the transition to a predefined configuration. To address this issue, this paper introduces the APF-based virtual leader, which is more suitable for driving a large group of UAVs to make sudden changes in flight direction. The APF assists agents in local trajectory planning, ensuring that the force range between the virtual leader and the real agents remains appropriate.

The dynamic generation of control forces for smooth group navigation is achieved through the use of APF, which effortlessly enables the swarm to navigate around obstacles

while maintaining cohesion. The APF of the virtual leader is determined by each agent's position and the current placement of the virtual leader. The initial position of the virtual leader can be defined as the centroid of the real leaders group. To achieve the desired formation with a rigid body and ensure equilibrium for all agents, the virtual leader should be placed at the centroid, which can be defined as follows:

$$p_{vl} = 1/N \sum_{i=1}^N p_i \quad (12)$$

As illustrated in Figure 3, our approach enhances the maneuverability of the affine-based formation by incorporating a virtual leader alongside conventional leader. This enables agents to change their flight direction and replan their local trajectories. Consequently, agents can interact with multiple obstacles, adaptively adjusting their local trajectories while iteratively evaluating collision possibilities. This concept allows for responsive behavior to environmental changes, as the APF directs agents away from obstacle positions towards new equilibrium positions. The equilibrium position for each agent is determined by following the steepest gradient towards the geometric point where the sum of these virtual forces is zero. The followers then track the plan of the virtual leader at the local level, seamlessly integrating trajectory planning with formation control. The virtual leader is initially positioned at the centroid of the leaders.

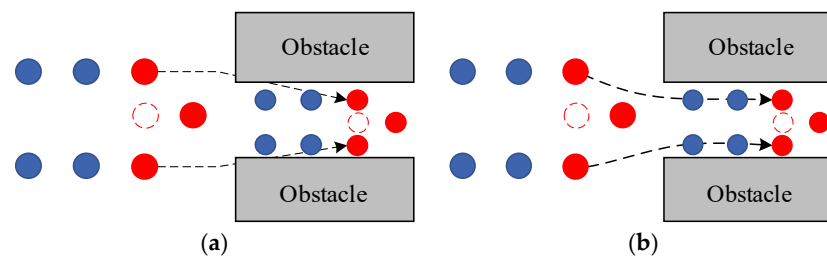


Figure 3. Illustration for passing through the rectangle obstacle by rotating and scaling the original formation. (a) Formation may collide the obstacles at preset path. (b) Passing through without collision. The leader is red, virtual leader red-dotted, followers blue, and obstacles are gray.

The use of artificial potential between the virtual leader and the agents is illustrated in Figure 4. The virtual leader potential function is simply a function of each agent's position and the virtual leader's current placement. Let $U_{ij}(\|p_{ij}\|)$ represent a non-negative smooth pairwise artificial potential function associated with the relative distance between agent (i, j) . It governs the collective and individual movement within the swarm, which satisfies:

$$U_{ij}(\|p_{ij}\|) = \begin{cases} a \ln(\|p_{ij}\|) + \frac{b}{c(\|p_{ij}\|)^2} & \|p_{ij}\| \leq d_{sen} \\ a \ln(r) + \frac{b}{c(r)^2} & \|p_{ij}\| > d_{sen} \end{cases} \quad (13)$$

where $i, j = 1, 2, \dots, N, i \neq j$, a , b and c are positive constants, $U_{ij}(\|p_{ij}\|)$ is bounded and continuously differentiable over interval $[0, +\infty)$. $\nabla(b/(\|p_{ij}\|)^2)$ describes the gradient negative cube odd function, which is not defined at 0 point. Therefore, $U_{ij}(\|p_{ij}\|)$ would obtain its boundary maximum value U_0 when $\|p_{ij}\| \rightarrow 0$; minimum U_d when (i, j) satisfies the desired position; that is, $\|p_{ij}\| = d$; and the constant U_r when $\|p_{ij}\| > d_{sen}$. Formulation (13) can describe both the fundamental forces of attraction and repulsion between agents, agents and virtual leaders, and agents and obstacles, a , b , and c are the positive constants.

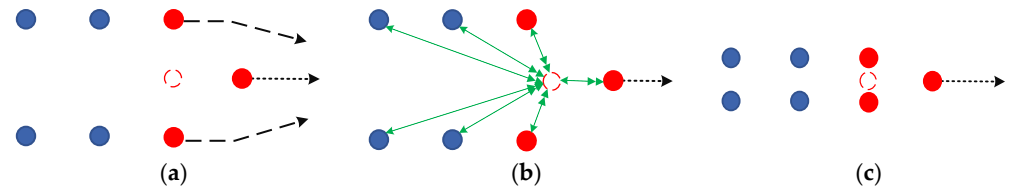


Figure 4. Illustration for virtual forces between agents and virtual leader: (a) Formation flight by desired path; (b) agent local planning; (c) flight in corresponding configuration. The green line represents the gradient descent direction of APF. The leader is red, virtual leader red-dotted, followers blue. (Both arrows show the direction).

Meanwhile, the obstacles could generate a repulsive force solely to prevent collisions and enable the formation to traverse smoothly among multiple obstacles. The obstacles then produce a repulsive force between the agents and polygon edges of obstacles (inversely related to the distance), as follows:

$$U_{ob} = \sum_{k=1}^m \frac{b}{c(\|p_{io}\|)^2} (b, c)_{ob} \quad (14)$$

It is important to note that the system does not include a specific obstacle detection law. Instead, it assumes that information about obstacle position and shape is available, for example, through system perception and 3D modeling. However, it is observed that the value and direction can change suddenly to potential forces within the formation. As a result, some may find that their newly calculated trajectory could potentially lead to collisions with other agents or obstacles, when the formation is striving to reach an equilibrium point. Therefore, trajectory planning should be incorporated into the motion of the agents at each time step.

3.2. Local Trajectory Planning

In order to avoid the system oscillations resulted from the sudden change of velocity, the receding horizon strategy is utilized to reduce the settling time in formation control of UAVs swarm navigation in the presence of obstacles. Therefore, the Runge–Kutta of order 4 (RK4) method is introduced in the paper for kinematic discretization description to update the agents' state.

The purpose of employing this method is to facilitate the adaptability of local trajectories, meaning that new trajectories are smoothed and discretized within the time step prior to the agents reaching new target points. As a result, the agents ultimately converge towards an equilibrium point, where the total sum of virtual forces becomes zero, effectively preventing any potential collisions. The description of local control law is as follows:

$$u_{local}^i = \underbrace{-\sum_{j=1, j \neq i}^N \nabla_i [\lambda_{ij} U_{ij}(\|\tilde{p}_{ij}\|) + \lambda_{io} U_{io}(\|\tilde{p}_{io}\|) + \lambda_{iv} U_{iv}(\|\tilde{p}_{iv}\|)]}_{f_i^p} + \underbrace{C_1 \sum_{j=1, j \neq i}^N (v_i - v_j) + (v_i - v_{vl})}_{f_i^v} \quad (15)$$

where C_1 is the constant value, λ_{ij} , λ_{io} , and λ_{iv} are different spring constants. f_i^p is the gradient-based term, which regulates position consensus by employing local repulsive and attractive forces, ensuring collision avoidance and cohesion maintenance, and f_i^v means the agent's velocity is adjusted by utilizing the weighted average velocity of its neighbors as a reference. Denote $X\{p_i, v_i\}$ as the state of the agent, and the state is updated within a time step. Thus, the process of updating is described as follows:

$$X_{n+1} = X_n + h_{n+1} \sum_{i=1}^s a_i k_i, i = 1, 2, \dots, s \quad (16)$$

$$k_i = f(t_s, x_n^j, x_{n+1}^j), j = 1, 2, \dots, s \quad (17)$$

where the k_i is a slope of four points in the time step interval, calculated at both ends and two midpoints of the interval. Hence, input u_i^j is initially derived through the application of the local control law. Subsequently, the acquired position and velocity are discretized to facilitate the update of the local trajectory as Figure 5 shows.

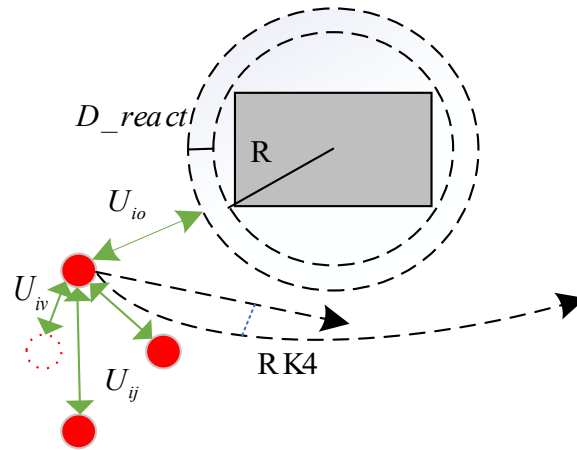


Figure 5. Illustration for one agent's local trajectory planning in formation. The forces in green are between agents, agents and virtual leaders, and agents and obstacles. The dotted line in black represents the original and iterated trajectory. (To simplify the calculations of obstacle forces, the obstacles are enclosed within a circle). The agent is red, virtual leader red-dotted (Both arrows show the direction).

As depicted in Figure 5, the agent interacts with other agents, obstacles (with d-react), and the virtual leader, which generates the three kinds of artificial potential forces U_i to obtain the fast gradient descent direction $\nabla(U_i)$ to equilibrium state X_i . Then, the state of the formation would be constantly updated X_{n+1}^i in a receding horizon to avoid the collision. The following Algorithm 1 summarizes the local control.

Algorithm 1: Local Trajectory Planning

Count the error sum $\tilde{p}_i, \tilde{v}_i$ in time step

$$p_{io} = \vec{p}_{io} * (d_{react} / ||p_{io}||)^2 \quad p_{iv} = \vec{p}_{iv} * ||p_{iv}||$$

$$p_{ij} = \vec{p}_{ij} * ||p_{ij}||$$

$$v_{ij} = v_i - v_j, \quad v_{iv} = v_i - v_{vl}$$

Calculate equilibrium location p_i

$$\nabla_i[U_{ij}(||\tilde{p}_{ij}||) + U_{io}(||\tilde{p}_{io}||) + U_{iv}(||\tilde{p}_{iv}||)]$$

Calculate the state $X_{n+1}\{p_i, v_i\}$ in next time

$$X_{n+1} = X_n + h \sum_{i=1}^4 a_i k_i$$

Check

$$||p_{io}|| < d_{min}$$

$$||v_i|| < v_{min}$$

end

Where d_{react} is the sensing distance based on the radius of the obstacles, d_{min} is the minimum distance between agents and obstacles, and v_{min} is the lower bounds of velocity, which can be described as:

$$\begin{cases} v_{min} \leq ||v_i|| \leq v_{max} \\ u_{min} \leq u_i \leq u_{max} \end{cases} \quad (18)$$

Furthermore, to mitigate the impact of local minima limitations in the implementation of APF, such as a formation navigating through narrow openings between obstacles, it

is essential to reduce the velocity of the virtual leader as the forces applied to the real agents increase.

4. Formation Control with Virtual Leader

4.1. Algorithm Description

If we assume that only a subset of agents in the formation can receive global information, then a widely used distributed controller protocol with a three-component controller protocol can be defined as follows:

$$\mathbf{u}_i = \underbrace{\mathbf{f}_i^p + \mathbf{f}_i^v}_{\mathbf{u}_{local}} + \mathbf{f}_i^d \quad (19)$$

where the first component $\mathbf{f}_i^p$ and the second one $\mathbf{f}_i^v$ are the same as (15) in local formation control for input and $\mathbf{f}_i^d$ is the navigational feedback which would drive the agent to track the global path.

Therefore, the distributed control protocol for an agent based on virtual leader and APF is designed as follows:

$$\mathbf{u}_i = \underbrace{-\sum_{j=1, j \neq i}^N \nabla_i(U_{ij} + U_{io} + U_{iv})}_{\mathbf{f}_i^p} + \underbrace{C_1 \sum_{j=1, j \neq i}^N (\mathbf{v}_i - \mathbf{v}_j) + (\mathbf{v}_i - \mathbf{v}_{vl})}_{\mathbf{f}_i^v} + \underbrace{f(\mathbf{u}_d, \mathbf{v}_d, \mathbf{p}_d)}_{\mathbf{f}_i^d} \quad (20)$$

$$f(\mathbf{u}_d, \mathbf{v}_d, \mathbf{p}_d) = \mathbf{u}_d - C_2(\mathbf{p}_i - \mathbf{p}_d) - C_3(\mathbf{v}_i - \mathbf{v}_d) \quad (21)$$

where $f(\mathbf{u}_d, \mathbf{v}_d, \mathbf{p}_d)$ represents the global path, in which $\mathbf{u}_d$ and $\mathbf{v}_d$ are desired velocity and obtained from desired path $\mathbf{p}_d$. The global path $\mathbf{p}_d$ is designed by the affine-based control algorithm and the corresponding configurations must be pre-defined.

From Equation (8), the constant is defined as follows:

$$\mathbf{p}^*(t) = [I_n \otimes A(t)]\mathbf{r} + I_n \mathbf{b}(t) \quad (22)$$

where $A(t) \in \mathbb{R}^{d \times d}$ and $\mathbf{b}(t) \in \mathbb{R}^d$ are continuous of t . The desired position $\mathbf{p}^*(t)$ of agent $i \in V$ in the target formation is $\mathbf{p}_i^*(t) = A(t)\mathbf{r}_i + \mathbf{b}(t)$, where $\mathbf{r}$ is the constant configuration and $(G, \mathbf{r})$ the nominal formation for a typical geometric pattern, such as the standard geometric graph, as illustrated in Figure 2. Thus, the desired control input $\mathbf{u}_d$ would be:

$$\mathbf{u}_d = -1/\gamma_i \sum_{j \in N_i} \Omega_{ij}(\dot{\mathbf{p}}_{id} - \dot{\mathbf{p}}_{jd}) \quad (23)$$

where $\gamma_i = \sum_{j \in N_i} \omega_{ij}$, motion consensus is only considered in the paper; thus, we have $\mathbf{v}_{1d} = \dots = \mathbf{v}_{Nd} = \mathbf{v}_d$, $\mathbf{p}_{1d} = \dots = \mathbf{p}_{Nd} = \mathbf{p}_d$. Formulations (22) and (23) describe the pre-defined configuration and the formation moves to the desired position $\mathbf{p}_i(t) \rightarrow \mathbf{p}^*(t)(t \rightarrow \infty)$. This strategy primarily focuses on ensuring that various pre-defined configurations can be achieved even in the presence of obstacles. Additionally, the locally planned trajectory can be adjusted to avoid collisions if necessary and the sudden change of velocity. The following Algorithm 2 summarizes the desired input control.

This paper creatively integrates the local control and affine-based global control into a unique optimal control framework. Hence, the approach involves using an affine-based method to generate multiple geometric patterns for all agents, followed by online calculation of local trajectories to update the trajectory and avoid collisions. The aim of this model would achieve collective maneuvers simultaneously and efficiently.

Algorithm 2: Formation Control

Pre-define the nominal configurations image
 $I(r) = \{P = [p_1^T, p_2^T, \dots, p_N^T]^T \in \mathbb{R}^{2N} \mid p_i = A^T r_i + b\}$
Regulate the change of formation
 $p^*(t) = [I_n \otimes A(t)]r + I_n b(t)$
Calculate the desired inputs u_d for leaders' path
 $u_d = -1/\gamma_i \sum_{j \in N_i} \Omega_{ij}(\dot{p}_i - \dot{p}_j)$
Calculate virtual leader trajectory in single-integrator
 $u_{vl} = \dot{p}_{t+1} - \dot{p}_t$
Calculate the real input for followers
 $u_i = f_i^p + f_i^v + f_i^l$
Optimize the trajectory (state) in Algorithm 1
 $X_{n+1} = X_n + h \sum_{i=1}^4 a_i k_i$
Nonlinear programming for the spring constant
 $\lambda_{ij}, \lambda_{io}$ and λ_{iv} by *fmincon* in Matlab toolboxes
end

Moreover, the use of virtual leader forces has been considered as an adjustment metric to prevent the collision from agents and obstacles, which would enhance the realistic functional and generality of UAV formation flight. This method can also solve the problem of much communication and feedback between leaders and followers to some extent, as Figure 5 shows.

The following assumptions are made in this paper:

1. The obstacle avoidance maneuvers are primarily in the horizontal plane.
2. To simplify the calculations of obstacle forces, the obstacles are enclosed within a circle.
3. The communication among UAVs is not limited by the information transmission distance.
4. The nominal configuration is affinely spanned $\mathbb{R}^d$ and designed to be generically universally rigid, leaders must satisfy the desired condition of $d + 1$ agents affinely spanning $\mathbb{R}^d$ at least.
5. The formation (G, p) maintains the connection relation within the graph $G = (V, E, A)$ through the whole flight.

4.2. Proof of Main Results

In this subsection, the primary outcomes of the motion consensus and formation control protocol for a UAV swarm with virtual agents are presented, supported by the following theorem. The formation stability of the novel control protocol is theoretically analyzed and proved, as stated in the following theorem.

Theorem 1. *Supposing a disordered swarm of nagents is described by (5) in dynamics and steered by protocol (20) in formation control, if the interaction between agents within the graph (G, p) is undirected and connected for the whole time, Z_0 is denoted as the sum of total artificial potential (formation, virtual agent and obstacles), and then the following conclusions are:*

1. *The graph is connected mutually if $Z_0 < U_r$ for all $t \geq 0$.*
2. *Obstacles collision can be prevented between the agent and obstacles if $Z_t < U_0$.*
3. *The trajectories of all agents converge to the desired values asymptotically.*

Proof of Theorem 1. We firstly prove part 1 of Theorem 1.

For the sake of simplicity and without sacrificing generality, it is assumed that $M(1 \leq M \leq N)$ agents function as leaders. Therefore, the kinematic error of the leaders in the system is defined as follows:

$$\begin{cases} \tilde{p}_i = p_i - p_d \\ \tilde{v}_i = v_i - v_d \end{cases}, i = 1, 2, \dots, M \quad (24)$$

based on the fact that the global trajectory is twice differentiable with respect to time and all agents have access to global information. It also satisfies that $\dot{\mathbf{p}}_d = \mathbf{v}_d$, $\dot{\mathbf{v}}_d = \mathbf{u}_d$ and we have $\tilde{\mathbf{p}}_{ij} = \tilde{\mathbf{p}}_i - \tilde{\mathbf{p}}_{j'}$, $\tilde{\mathbf{p}}_i = \tilde{\mathbf{v}}_i$, $\tilde{\mathbf{v}}_i = \tilde{\mathbf{u}}_i$.

Given that Z_t represents the summation of the total relative kinetic energy of an agent at time t , it can be deduced that:

$$Z(t) = 1/2 \sum_{i=1, i \in O}^N \left\{ [U_i(\|\tilde{\mathbf{p}}_{ij'}\|)] + C_2(\mathbf{p}_i - \mathbf{p}_d)^T(\mathbf{p}_i - \mathbf{p}_d) + C_3(\mathbf{v}_i - \mathbf{v}_d)^T(\mathbf{v}_i - \mathbf{v}_d) \right\} \quad (25)$$

Denote $V_i = \sum_{i=1, i \in O}^N \left\{ \tilde{\mathbf{p}}_i^T \tilde{\mathbf{p}}_i + U(\|\tilde{\mathbf{p}}_i\|) \right\}$ as the total artificial of agent i . Then in the error system, $Z(t)$ can be re-written as follows:

$$Z(t) = \frac{1}{2} \sum_{i=1, i \in O}^N [V_i + \tilde{\mathbf{v}}_i^T \tilde{\mathbf{v}}_i] \quad (26)$$

It can be easily demonstrated that $Z(t)$ is positive semi-definite, making it a suitable candidate function for Lyapunov analysis. Considering the symmetry of U_{ij} with respect to $\tilde{\mathbf{p}}_{ij'}$, we have the following:

$$\frac{\partial U_i}{\partial \tilde{\mathbf{p}}_{ij'}} = \frac{\partial U_i}{\partial \tilde{\mathbf{p}}_i} = -\frac{\partial U_i}{\partial \tilde{\mathbf{p}}_{j'}} \quad (27)$$

Thus, we have

$$\frac{d}{dt} \left(\frac{1}{2} \sum_{i=1, i \in O}^N V_i \right) = \sum_{i=1, i \in O}^N \left\{ \tilde{\mathbf{v}}_i^T (\nabla_{\mathbf{p}_i} U_i) + C_2 \tilde{\mathbf{v}}_i^T \tilde{\mathbf{p}}_i \right\} \quad (28)$$

where $U_i = \sum_{i \in O}^N U_{ij'}(\|\tilde{\mathbf{p}}_{ij'}\|)$, j' means other agents, obstacles, and virtual leaders. Then, obtaining the derivative to time t for $Z(t)$ as follows:

$$\dot{Z}(t) = \sum_{i=1, i \in O}^N (\nabla_{\mathbf{p}_i} U_i) + C_2 \tilde{\mathbf{v}}_i^T \tilde{\mathbf{p}}_i + \tilde{\mathbf{v}}_i^T \dot{\tilde{\mathbf{v}}}_i \quad (29)$$

thus, via substituting (20) to (30), we might have:

$$\dot{Z}(t) = \sum_{i=1}^N \tilde{\mathbf{v}}_i^T \sum_{j=1, j \neq i}^N C_{1ij}(\mathbf{v}_{ij} - \mathbf{v}_{iv}) + \sum_{i=1}^N C_3 \tilde{\mathbf{v}}_i^T \dot{\tilde{\mathbf{v}}}_i \quad (30)$$

and Equation (31) can be expressed in matrix form as follows:

$$\dot{Z}(t) = -\tilde{\mathbf{v}}^T [(C_1 Y(t) + C_3 \Omega(t)) \otimes I_n] \tilde{\mathbf{v}} \quad (31)$$

where $\Omega(t)$ is the stress matrix, $Y(t) = \text{diag}[\gamma_1, \gamma_2, \dots, \gamma_N]$, $\tilde{\mathbf{v}} = (\tilde{v}_1^T, \tilde{v}_2^T, \dots, \tilde{v}_N^T)^T$. Apparently, $\Omega(t)$ and $Y(t)$ are positive semi-definite, these two factors would explain that $Z(t \geq 0)$ is non-increasing function with $\dot{Z}(t) \leq 0$.

Thus, the level set $\psi = \left\{ [\tilde{\mathbf{p}}^T, \tilde{\mathbf{v}}^T]^T \in \mathbb{R}^{2nN} : Z(t) \leq Z(0), t \geq 0 \right\}$ can be seen as a positively invariant set. Then, supposing that $Z(0) < U_r$, the equation can possibly meet $U_i(\|\tilde{\mathbf{p}}_{ij'}\|) \leq Z(t) \leq Z(0) < U_r$ ($i \in O, t \geq 0$) in each edge $[e_{ij}]$. Therefore, the connectivity of graph G of all agents $\mathbf{p}_i$ for swarm is shown within $t \geq 0$ and $i \in O$; that is to say, the part 1 of Theorem 1 is proved.

As for part 2 of Theorem 1, it is obvious that $U_i(\|\tilde{\mathbf{p}}_{ij'}\|) \leq Z(t) \leq Z(0) < U_r < U_0$ from above; thus, the obstacles collision is proved.

The part 3 of the theorem, $\|\tilde{\mathbf{p}}_i\|$ and $\|\tilde{\mathbf{v}}_i\|$ are all bounded and level set ψ is tight from the above analysis. Moreover, all trajectories of agents would finally converge and satisfy the invariant set $\psi' = \left\{ [\tilde{\mathbf{p}}^T, \tilde{\mathbf{v}}^T]^T : \dot{Z}(t) = 0, t \geq 0 \right\}$. As $Y(t)$ and $\Omega(t)$ are positive semi-definite from the definition, based on Formulation (31) and as $\dot{Z}(t) = 0$, we could obtain $\tilde{\mathbf{v}}^T [Y(t) \otimes I_n] \tilde{\mathbf{v}} = 0$ and $\tilde{\mathbf{v}}^T [\Omega(t) \otimes I_n] \tilde{\mathbf{v}} = 0$. Since the stress matrix $\Omega(t)$ can be considered as generalized graph Laplacian matrices with both positive and negative edge weights, $\Omega(t)$

can thus also meet the sum of squares property [30]; therefore, $\tilde{v}^T[(C_1 Y(t) + C_3 \Omega(t)) \otimes I_n] \tilde{v}$ can be equivalent to $\tilde{v}_1 = \tilde{v}_2 = \dots = \tilde{v}_N$. To be more specific, this result can demonstrate that the errors of all agents' velocity would converge to 0 asymptotically. Thus, it can imply the $\tilde{v}_1 = \tilde{v}_2 = \dots = \tilde{v}_N = 0$, such that part 3 of the Theorem 1 is proved. $\square$

Remark 1. After examining the proof process, it has been determined that parts 1–3 of the theorem align with the fundamental principles of swarm algorithms, namely cohesion, separation, and alignment as proposed by Reynolds. Furthermore, these essential criteria have been employed to validate the algorithm presented in this paper.

Remark 2. Despite the similarities between certain terms in Equation (20) and existing protocols, the swarm behavior guided by Equation (20) would be distinctively unique. This is due to the presence of a virtual agent serving as an evaluation index through the local trajectory. Consequently, each agent is required not only to track the global trajectory but also to actively adjust its own local trajectory.

Remark 3. Although the paper does not specifically analyze the impact of acceleration and turning rate as inputs, it is important to note that these two factors are influenced by the agents themselves. Therefore, assuming that the initial distance and interaction range between agents meet the required range, $Z(0) < U_r$ and $Z(0) < U_0$ can effectively avoid collisions within the agent swarm and ensure the overall stability of the system.

The controller protocol (20) offers several advantages to a certain extent. Firstly, it integrates protocols for leaders/followers and virtual agents, enabling simultaneous operation. This distinguishes it from existing control protocols and enhances the adaptability and scalability of the system. Secondly, the local adjustment eliminates the need for additional feedback information interaction among leaders and followers, thereby greatly reducing communication consumption. Thirdly, based on the aforementioned local protocol, each UAV can be dynamically and randomly assigned as a leader in complex environments, ensuring continuity in the event that leaders are shot down or destroyed.

5. Simulation and Analysis

In this section, we proceed to conduct numerical simulations to illustrate the theoretical results and validate the effectiveness of the novel formation control algorithm.

The whole experimental process is shown in Figure 6, and the algorithm in two-dimensional is described as follows:

1. The randomly distributed generated agents denoted as $\{p | p_{ij} \in \mathbb{R}^d, \forall (i, j) \in \mathbb{R}\}$, $(i, j) \in (1, \dots, n)$ are steered to rendezvous and make a swarm formation as the nominal configuration $p_0 \in (P, r)$. The leaders are set and then time-varying velocity $v(t)$ (10) starts to change in this stage for all agents u_i (20).
2. Desired geometric patterns $A(r)$ are predefined and generated with desired trajectories and corresponding positions of the formation $p^*(t)$ based on the knowledge of obstacle position and shape available.
3. A trajectory through all the barycenter of leaders of each predefined configuration; thus, this virtual agent motives with constant velocity $X(v_{vl}, p_{vl})$ in a single-integrator differentiable.
4. Based on the available knowledge of obstacles, using constant interaction with other agents and obstacle constraints (20) to trigger local trajectory planning in Algorithm 1 (15) and (16).
5. The formation operated the (20) control law to adjust the trajectories of agents possibly entering collide zones of obstacles while changing the configuration of formation $p(t) \rightarrow p^*(t)$, which can demonstrate the tracking ability of the algorithm (21).

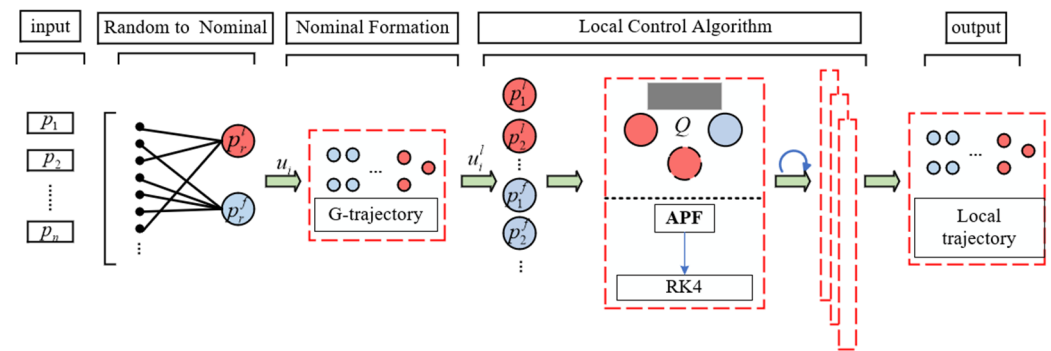


Figure 6. Flow chart of proposed algorithm experimental process. The leader is red, virtual leader red-dotted, followers blue.

It is noted that only the leaders possess global information regarding the desired trajectory (u_d). The focus on the position error relative to the reference trajectory is solely on the leaders. Consequently, the control laws for the leaders are not explicitly designed in this paper.

In order to evaluate the performance of our proposed methods, all the experiments are simulated on PC with MATLAB R2020a Windows 10. Protocol (20) is used to steer seven UAV agents moving in a two-dimensional space with five processes above through Algorithms 1 and 2.

The number of agents $N = 7$, the sensing range $d_{react} = 30$, desired distance from the virtual leader $d_0 = 30$, obstacles radius $obs_{rad} = 20$, maximum velocity $v_{max} = 80$ m/s, max acceleration $u_{max} = 2$ m/s², max velocity error $e_{vmax} = 0.5$ m/s, max trajectory error $e_{xmax} = 10$ m, the timestep $d_t = 0.1$ s. Other parameters were set as $a = 1.5$, $b = 2$, $c = 0.5$, $\delta = 0.6$, $\omega_{ij} = 1$. All the above parameters remained fixed throughout the simulation.

The experiment is divided into three phases: Firstly, the desired path of the formation, the position of configuration, and the obstacles are pre-defined. The experiment could be conducted by affine maneuvering by single-integrator control law from (20) in Zhao [22]. Secondly, the obstacle is randomly located in the trajectory of formation to test the effect of a local trajectory planning method (Algorithm 1). Finally, the formation with seven agents is operated as Algorithm 2 and the experiment process acts as Figure 6.

5.1. Phase I

With the available information regarding obstacles, it becomes feasible to pre-define the desired path and its corresponding configuration. The purpose of this part is to generate the nominal configurations during each time interval, which can provide the desired path for the leaders of the formation and the desired tracking velocity. The positions of agents are randomly selected from the box $[-3 \text{ m}, 3 \text{ m}] \times [-3 \text{ m}, 3 \text{ m}]$, as Figure 7b shows. Their velocities of leaders are set as constant $v_d = 45$ m/s and directions in intervals $[-\pi/2, \pi/2]$. Agents 1, 2, and 3 are assigned as leaders illustrated in red through the whole simulation and the four change forms of configuration are calculated, scaling, rotation, shear, and translation. It is shown that the affine-based leader/follower consensus protocol can form the continuous configuration change for the formation. The nominal configuration is established prior to 6 s. At the moment $t = 6$ s and $t = 13$ s the formation is expected to undergo scaling in the y-direction, with shrinking and enlargement occurring, respectively. Starting from $t = 19$ s and $t = 24$ s, the formation initiates rotation. Additionally, there is shearing at $t = 28$ s and $t = 36$ s, while the formation translates during other time intervals within the given range.

One can see that the velocity for followers would have suddenly changed to meet the requirement in each time interval of the different configuration changes. The scale of change could not be controlled effectively in any case.

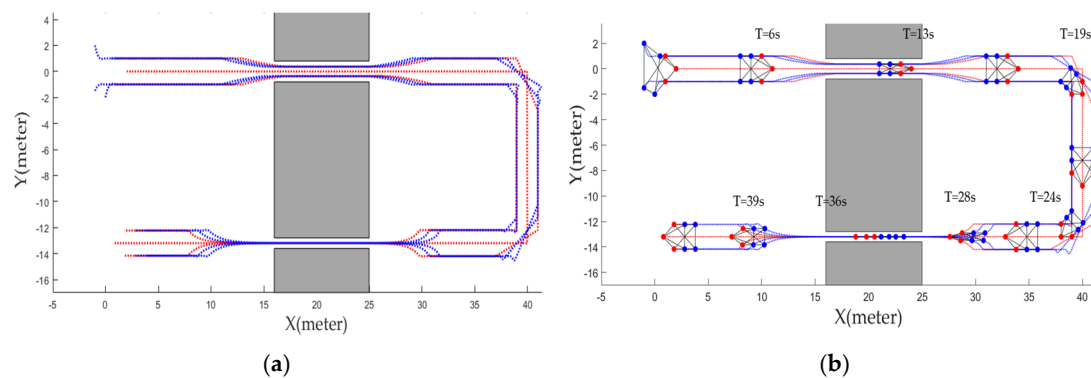


Figure 7. Affine-based predefined configurations patterns for illustrations: (a) pre-defined path; (b) pre-defined configuration. The leader is red dot, followers blue dot, and obstacles are gray. Red and blue dotted lines show the desired trajectory for leaders and followers, respectively.

5.2. Phase II

The aim of the second part is mainly to test the effect of the algorithm of local trajectory planning. The position of the obstacle is located randomly at $[-10, -5, 60, 6]$ (x, y, width, height) and agents are from the box $[-50 \text{ m}, -40 \text{ m}] \times [-60 \text{ m}, -50 \text{ m}]$ as Figure 8a shows; the acceleration, velocity, and the angle of drift are in intervals $[0 \text{ m/s}^2, 2 \text{ m/s}^2]$, $[1 \text{ m/s}, 15 \text{ m/s}]$ and $[-\pi/2, \pi/2]$, the velocity of the virtual leader is $v^l = [5 \text{ m/s}, 5 \text{ m/s}]$. The formation has started at the nominal configuration and encounters the obstacle along the desired path x_d , as Figure 8b shows. The state X_{vl} of the virtual leader is the desired state for agents, as Figure 8b shows, and the formation would generate Algorithm 1 to steer agents to avoid obstacles. The velocity and its error related to the virtual leader are given in Figure 8c,d; as can be seen, the velocity fluctuates greater at $t = 10 \text{ s}$ and $t = 20 \text{ s}$ moving through the obstacle and the velocity change interval is from $v = 5 \text{ m/s}$ to $v = [1 \text{ m/s}, 14 \text{ m/s}]$. Figure 8d illustrates the change in velocity error during the same time interval. The primary reason for this is that the desired trajectory, as moved by the virtual leader, would directly pass through the obstacle. However, the formation manages to avoid collision and maintain its configuration due to the forces exerted by the virtual leader. In Figure 8e, the trajectory error of all agents is depicted, with the note that the X and Y axes have different scales. It can be observed that the leaders and followers exhibit smaller average position errors, while the overall formation has a relatively larger error. The desired trajectory is not fully matched due to the influence of uninformed followers on the leaders, but it remains within an acceptable range. Lastly, Figure 8f demonstrates the total velocity error for all agents, indicating that the protocol will converge eventually.

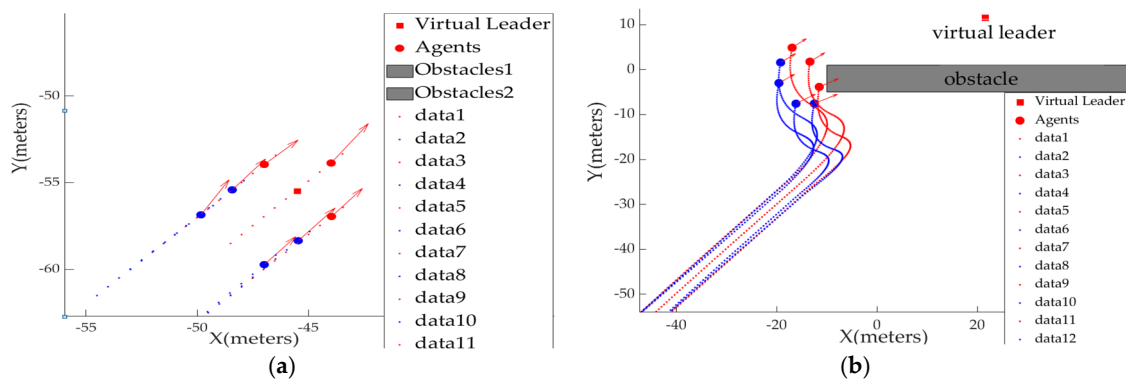


Figure 8. Cont.

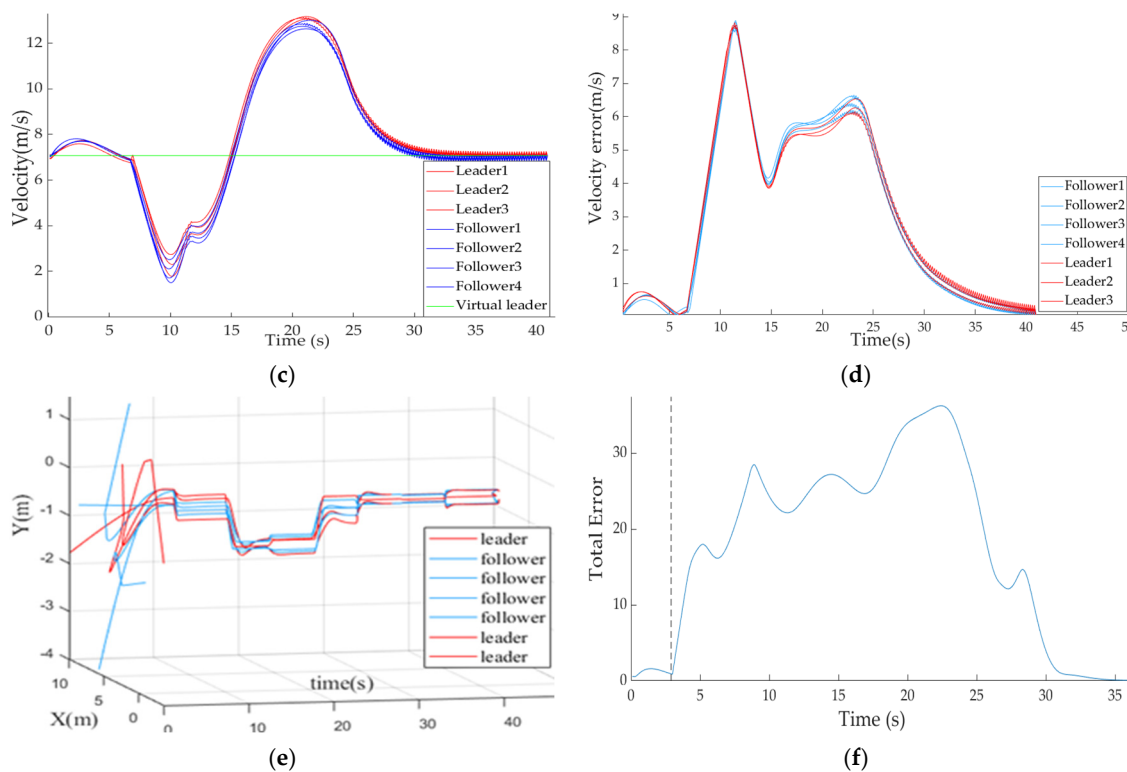


Figure 8. Simulation results of local trajectory planning: (a) initial state; (b) snapshot of the formation at avoiding collision; (c) velocity of agents; (d) velocity error of agents; (e) trajectory error of agents; (f) total velocity error of all agents.

5.3. Phase III

The purpose of this part is to test the affine-based local trajectory planning algorithm in the pre-defined positions of obstacles to follow the desired path without collision. The positions of three obstacles are located at $[-10, -5, 30, 6]$, $[-10, -30, 30, 50]$, and $[-10, -40, 30, 6]$. The initial velocity of the virtual leader is set $v^l = [40 \text{ m/s}, 0 \text{ m/s}]$ and it moves with constant velocity through all the paths with direction intervals $[-\pi/2, \pi/2]$. Meanwhile, based on the velocity of the virtual leader $v^l = [45 \text{ m/s}, 0 \text{ m/s}]$, the velocities of agents in the formation are randomly generated with a standard normal distribution. The initial acceleration of the agents is set $[0 \text{ m/s}^2]$. The formation starts at the random positions $[p_1, p_2, \dots, p_7]$ to follow the desired path p_d , planning the local trajectory. From Figure 9a, it is known that all the accelerations of agents have changed drastically and unpredictably, especially in the interval of $t = 200 \text{ s}$ – 300 s , and would converge in the final. The main reason is that input of agents would increase or decrease suddenly when the forces among agents undergo changes. Figure 9b depicts the velocity error of the agents, indicating that the reference velocity is well tracked by both leaders and followers throughout the trajectory tracking process. The experimental results particularly demonstrate that the proposed algorithm enables the asymptotic tracking of the desired trajectory and remains effective even when the trajectory changes suddenly due to the formation steering local planning to avoid collisions during the motion process. Figure 9c illustrates that the minimum distance between agents consistently remains greater than 0, indicating successful collision avoidance for the formation and the maintenance of the desired configuration within the acceptable error range. Additionally, Figure 9d indicates that trajectory tracking is achieved satisfactorily.

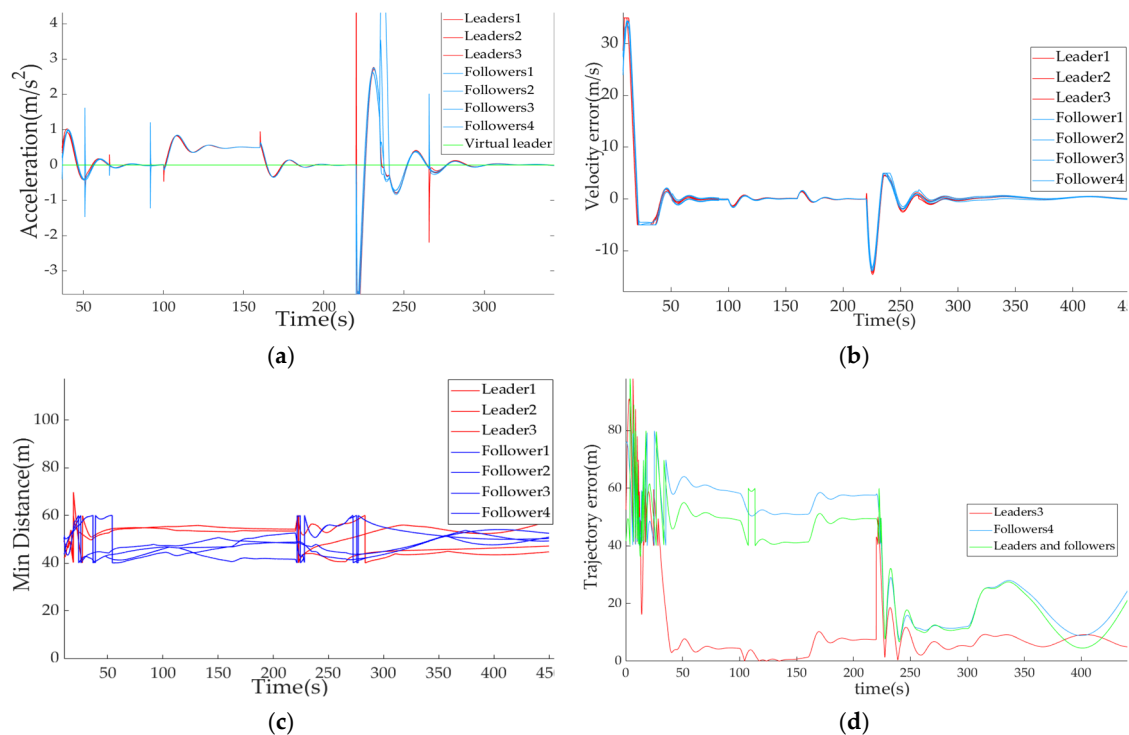


Figure 9. Simulation results of global algorithm: (a) acceleration of agents; (b) velocity error of agents; (c) minimum distance between agents; (d) trajectory error of agents.

5.4. Discussion

The experiments primarily focused on determining the optimal trajectory of each agent for affine maneuver control, revealing that a single mode of control may lack generality in complex environments to some extent. The proposed algorithm has been validated to be effective, particularly in addressing situations involving colliding obstacles. From the experiments, it was observed that each agent in the swarm autonomously carried out obstacles' detection, judgment, and decision-making, compared with the original algorithm. The virtual leader mainly influenced an agent if it was likely to enter the zone where obstacles react, while the other agents continued to track desired trajectories with less effect from the virtual leader forces and reach predefined positions. Notably, even in the presence of abrupt changes in trajectories or obstacles, the proposed algorithm remained effective.

From Figures 7–9, it is demonstrated that the purpose of the proposed algorithm is to guarantee the configuration change to avoid the collision autonomously and maneuver continuously to track the predefined trajectory of an affine image set $A(r)$. Thus, Figure 7b shows that the random formation is generated at the beginning and starts to form the nominal configuration and then reach each $A(r)$ ($[p_i^*]^T$ from $[p_i]^T$). However, a pre-defined trajectory would not avoid the obstacles effectively and could not provide the capability of changing velocity v suddenly.

Therefore, the proposed model is introduced to deal with the problem. It is shown that 11 desired positions have been pre-defined in Figure 7, and the original path shows the possibility of collision. Note that system of detecting and steering each agent is determined by distance $\|p_i - p_{obstacle}\|$ only. In any case, the formation steered by an affine-based maneuvering method would fall apart and lose the body integrity when encountering the obstacles. Moreover, a virtual leader is designed to operate based on the pre-defined trajectory of the centroid of the pre-defined configurations as Figure 3 demonstrates, denoted as p_d^{vl} (12).

The advantage of this virtual strategy lies in its ability to enhance system functionality without significantly increasing complexity. Firstly, this strategy does not interfere with the formation motion. Secondly, the trajectory of the virtual agent is constructed based on the existing pre-defined trajectory. Lastly, this virtual agent can be seamlessly integrated into

the control protocol, serving as an adjustment metric. Thus, the virtual leader introduced into the algorithm can modify the formation motion and improve the effectiveness of formation maneuvering.

In order to evaluate the effectiveness of the proposed method, we use two indexes to evaluate the motion results. Table 1 intuitively shows the average value of all the experimental results with motion speed and track accuracy. It can be seen from Table 1 that the proposed method had the highest accuracy value. However, the time has been taken larger than that of the affine-based maneuvering method, which is due to the complexity of the proposed being higher than the affine-based maneuvering method ($O(L^2)$ and $O(2L)$).

Table 1. Evaluation tests (mean value).

Methods	Speed (s)	Track Accuracy (%)
Affine-based maneuvering	40	67.34
The proposed method	450	89.51

The second term Speed shows that the affine-based maneuvering took up 40 s, and the proposed method used 450 s. Obviously, the proposed method would use more time than the classic one with higher computational cost. Firstly, based on $p^*(t) = [I_n \otimes A(t)]r + I_n b(t)$ (22), the classic method basically solved the changeable problem of formation configurations for multi-agent. Then, the proposed method took more within each time step due to the calculation from Algorithm 1 (15).

The obstacle added to the experiment scenario is depicted in Figure 10a, which is based on Figure 7a. As illustrated in Figure 7b, the new obstacle is inserted between $t = 13$ s to $t = 19$ s. The experimental results primarily focus on the formation maneuvering $t = 0$ s–40 s and are obtained as follows.

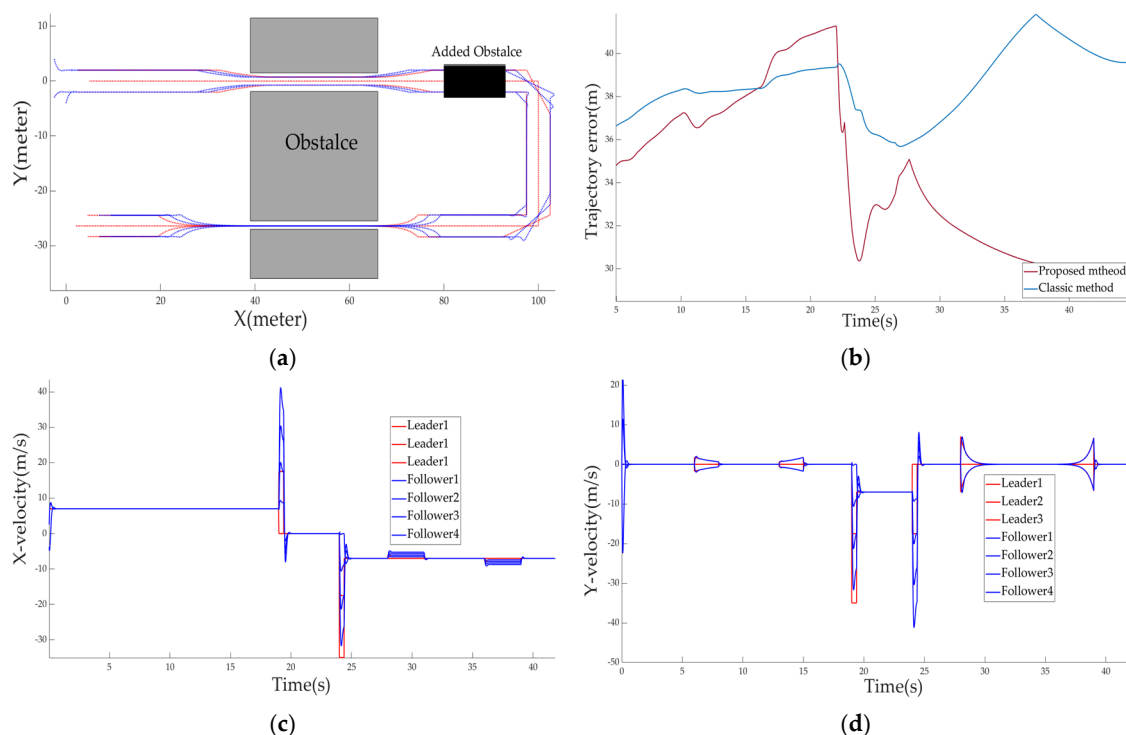


Figure 10. Cont.

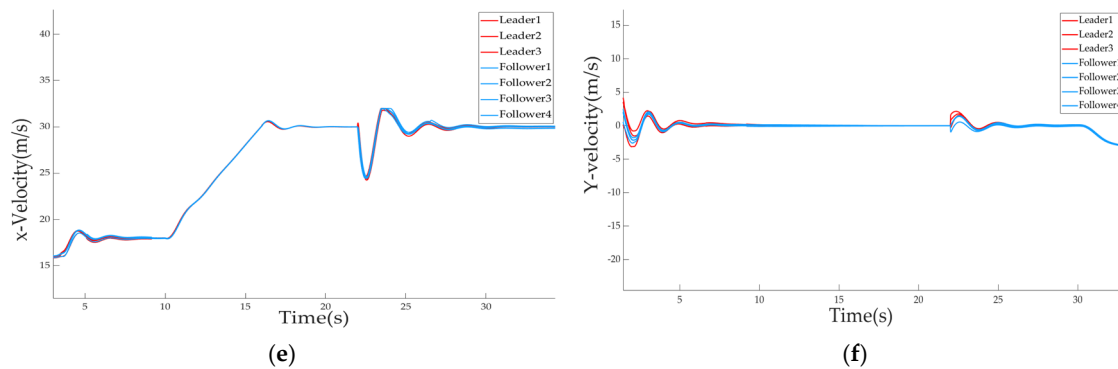


Figure 10. Simulation results of contrast test: (a) experimental scenario; (b) trajectory error; (c) velocity of the classic method in X axis; (d) velocity of the classic method in Y axis; (e) velocity of the proposed method in X axis; (f) velocity of the proposed method in Y axis.

It has been shown that the proposed method improves the maneuvering ability of the formation. From Figure 10b, it is evident that the trajectory error is reduced by the proposed method. Furthermore, the velocity of the X axis in the classic method undergoes more drastic changes compared to the proposed method, as observed in Figure 10c,e. Similarly, Figure 10d,f demonstrate that the velocity of the Y axis in the proposed method changes smoothly.

6. Conclusions

The changes in formation configuration have been confirmed to be feasible. However, the challenge of maneuvering the formation while avoiding obstacles and converging to desired values has not been adequately addressed to enhance practical application. Moreover, the concept of a rigid body may not suffice in diverse environmental situations. Although the affine technique has somewhat resolved this issue by continuously providing multiple geometric patterns for configuration changes, it requires an adjustable environment and the maintenance of a rigid body at all times. As a result, this method lacks some generality for practical application, particularly in complex environments.

Therefore, this paper introduces a new consensus protocol based on a hybrid control model to enhance the effectiveness of formation maneuver control and obstacle avoidance in complex environments. The protocol employs distributed control laws with a virtual leader for double-integrator control and establishes the system's stability. The presented algorithm showcases the capability to autonomously adjust each local trajectory of a formation based on potential forces when encountering obstacle colliding zones. Crucially, the control law is confined to each agent's local reference frame, thereby improving the adaptability and scalability of the formation in realistic environments. Furthermore, the results show that the formation, steered by the proposed control algorithm, can still reach the predefined configuration even in the presence of violent trajectory changes. Additionally, the algorithm effectively reduces communication consumption.

In the future, there are several important topics that require attention. These include investigating more intricate graph structures, dynamic leader reassignment, and transitioning between distributed and decentralized control laws to enhance autonomy. Moreover, it is crucial to develop more autonomous control laws for adapting to environmental changes, improve deep learning techniques to establish rule-based response mechanisms, explore the interconnections between agents in complex networks, utilize the MPC method for local environment prediction, and delve into other related areas.

Author Contributions: Conceptualization, C.K. and J.X.; methodology and writing—original draft, C.K.; writing—review and editing, Y.B.; visualization C.K.; supervision J.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

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

References

- Purta, R.; Dobski, M.; Jaworski, A. A Testbed for Investigating the UAV Swarm Command and Control Problem Using DDDAS. *Procedia Comput. Sci.* **2013**, *18*, 2018–2027. [\[CrossRef\]](#)
- Quan, L.; Yin, L.; Zhang, T.; Wang, M.; Wang, R.; Zhong, S.; Cao, Y.; Xu, C.; Gao, F. Formation Flight in Dense Environments. *arXiv* **2022**, arXiv:2210.04048.
- Oh, K.K.; Park, M.C.; Ahn, H.S. A survey of multi-agent formation control. *Automatica* **2015**, *53*, 424–440. [\[CrossRef\]](#)
- Zhao, S.; Zelazo, D. Bearing Rigidity and Almost Global Bearing-Only Formation Stabilization (Vol.). *IEEE Trans. Autom. Control* **2016**, *61*, 1255–1268. [\[CrossRef\]](#)
- Anderson, B.; Yu, C.; Fidan, B.; Hendrickx, J. Rigid Graph Control Architectures for Autonomous Formations. *IEEE Control Syst.* **2008**, *28*, 48–63.
- Lin, Z.; Wang, L.; Chen, Z.; Fu, M. Necessary and sufficient graphical conditions for affine formation control. *IEEE Trans. Autom. Control* **2016**, *61*, 2877–2891. [\[CrossRef\]](#)
- Rabta, B.; Wankmüller, C.; Reiner, G. A drone fleet model for last-mile distribution in disaster relief operations. *Int. J. Disaster Risk Reduct.* **2018**, *28*, 107–112. [\[CrossRef\]](#)
- Petracek, P.; Kratky, V.; Baca, T.; Petrlik, M.; Saska, M. New Era in Cultural Heritage Preservation: Cooperative Aerial Autonomy for Fast Digitalization of Difficult-to-Access Interiors of Historical Monuments. *arXiv* **2023**, arXiv:2303.02962. [\[CrossRef\]](#)
- Petrlik, M.; Petracek, P.; Kratky, V.; Musil, T.; Stasinchuk, Y. UAVs Beneath the Surface: Cooperative Autonomy for Subterranean Search and Rescue in DARPA SubT. *J. Field Robot* **2023**, *3*, 1–68. [\[CrossRef\]](#)
- Marconi, L.; Melchiorri, C.; Beetz, M.; Pangercic, D.; Siegwart, R.; Leutenegger, S. The sherpa project: Smart collaboration between humans and ground aerial robots for improving rescuing activities in alpine environments. In Proceedings of the 2012 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), College Station, TX, USA, 5–8 November 2012.
- Mahdoui, N.; Mahdoui, V.; Natalizio, E. Communicating multi-uav system for cooperative slam-based exploration. *J. Intell. Robot Syst.* **2020**, *98*, 325–343. [\[CrossRef\]](#)
- Dorling, K.; Heinrichs, J.; Messier, G. Vehicle routing problems for drone delivery. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *47*, 70–85. [\[CrossRef\]](#)
- Oh, K.K.; Ahn, H.S. Formation control of mobile agents based on inter-agent distance dynamics. *Automatica* **2011**, *47*, 2306–2312. [\[CrossRef\]](#)
- Cao, Y.; Ren, W. Distributed formation control for fractional-order systems: Dynamic interaction and absolute/relative damping. *Syst. Control Lett.* **2010**, *59*, 233–240. [\[CrossRef\]](#)
- Saska, M.; Hert, D.; Baca, T.; Kratky, V.; Nascimento, T. Formation control of unmanned micro aerial vehicles for straitened environments. *Auton. Robot* **2020**, *44*, 991–1008. [\[CrossRef\]](#)
- Marzat, J.; Bertrand, S.; Eudes, A.; Sanfourche, M.; Moras, J. Reactive MPC for autonomous MAV navigation in indoor cluttered environments: Flight experiments. *IFAC Pap.* **2017**, *50*, 15996–16002. [\[CrossRef\]](#)
- Quan, L.; Yin, L.; Xu, C.; Gao, F. Distributed Swarm Trajectory Optimization for Formation Flight in Dense Environments. In Proceedings of the 2022 International Conference on Robotics and Automation, Philadelphia, PA, USA, 23–27 May 2022.
- Ai, X.L.; Yu, J.Q.; Chen, Y.B.; Chen, F.Z.; Shen, Y.C. Optimal Formation Control with Limited Communication for Multi-Unmanned Aerial Vehicle in an Obstacle-Laden Environment. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2016**, *47*, 979–997. [\[CrossRef\]](#)
- Dong, W. Robust formation control of multiple wheeled mobile robots. *J. Intell. Robot Syst.* **2011**, *62*, 547–565. [\[CrossRef\]](#)
- Hengster-Movrić, K.; Bogdan, S.; Draganjac, I. Multi-agent formation control based on bell-shaped potential functions. *J. Intell. Robot Syst.* **2010**, *58*, 165–189. [\[CrossRef\]](#)
- Liu, Y.; Jia, Y. An iterative learning approach to formation control of multi-agent systems. *Syst. Control Lett.* **2012**, *61*, 148–154. [\[CrossRef\]](#)
- Zhao, S. Affine Formation Maneuver Control of Multiagent Systems. *IEEE Trans. Autom. Control* **2018**, *63*, 4140–4155. [\[CrossRef\]](#)
- Tavares, A.; Madruga, S.P.; Brito, A.V.; Nascimento, T.P. Dynamic leader allocation in multi-robot systems based on nonlinear model predictive control. *J. Intell. Robot Syst.* **2019**, *98*, 359–376. [\[CrossRef\]](#)
- Li, D.; Ma, G.; Xu, Y.; He, W.; Ge, S.S. Layered Affine Formation Control of Networked Uncertain Systems: A Fully Distributed Approach Over Directed Graphs. *IEEE Trans. Cybern.* **2021**, *51*, 6119–6130. [\[CrossRef\]](#) [\[PubMed\]](#)
- Monteriu, A. Nonlinear decentralized model predictive control for unmanned vehicles moving in formation. *Inf. Technol. Control* **2015**, *44*, 89–97. [\[CrossRef\]](#)
- Kuriki, Y.; Namerikawa, T. Formation control with collision avoidance for a multi-uav system using decentralized MPC and consensus-based control. *SICE J. Control. Meas. Syst. Integr.* **2015**, *8*, 285–294. [\[CrossRef\]](#)
- Dong, L.; Chen, Y.; Qu, X. Formation Control Strategy for Nonholonomic Intelligent Vehicles Based on Virtual Structure and Consensus Approach. *Procedia Eng.* **2016**, *137*, 415–424. [\[CrossRef\]](#)

28. Chen, L.; Mei, J.; Li, C.; Ma, G. Distributed Leader–Follower Affine Formation Maneuver Control for High-Order Multiagent Systems. *IEEE Trans. Autom. Control* **2020**, *65*, 4941–4948. [[CrossRef](#)]
29. Zhou, X.; Wen, X.; Wang, Z.; Gao, Y.; Li, H.; Wang, Q.; Yang, T.; Lu, H.; Cao, Y.; Xu, C.; et al. Swarm of micro flying robots in the wild. *Sci. Robot* **2022**, *7*, eabm5954. [[CrossRef](#)]
30. Arul, S.H.; Manocha, D. Dcad: Decentralized collision avoidance with dynamics constraints for agile quadrotor swarms. *IEEE Robot Autom. Lett.* **2020**, *5*, 1191–1198. [[CrossRef](#)]
31. Saffre, F.; Hildmann, H.; Karvonen, H.; Lind, T. Monitoring and Cordoning Wildfires with an Autonomous Swarm of Unmanned Aerial Vehicles. *Drones* **2022**, *6*, 301. [[CrossRef](#)]
32. Castrillo, V.U.; Manco, A.; Pascarella, D.; Gigante, G. A Review of Counter-UAS Technologies for Cooperative Defensive Teams of Drones. *Drones* **2022**, *6*, 65. [[CrossRef](#)]
33. Ruseno, N.; Lin, C.-Y.; Chang, S.-C. UAS Traffic Management Communications: The Legacy of ADS-B, New Establishment of Remote ID, or Leverage of ADS-B-Like Systems? *Drones* **2022**, *6*, 57. [[CrossRef](#)]
34. Montijano, J.I.; Randez, L.; Calvo, M. Explicit runge-kutta methods for the numerical solution of linear inhomogeneous ivps. *J. Comput. Appl. Math.* **2023**, *425*, 115083. [[CrossRef](#)]
35. Turpin, M.; Michael, N.; Kumar, V. Capt: Concurrent assignment and planning of trajectories for multiple robots. *Ind Robot.* **2014**, *33*, 98–112. [[CrossRef](#)]
36. Wang, J.; Xin, M. Integrated Optimal Formation Control of Multiple Unmanned Aerial Vehicles. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 1731–1744. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.