

Research Article Multi-UAV DMPC Cooperative Guidance with Constraints of Terminal Angle and Obstacle Avoidance

Zijie Jiang¹, Xiuxia Yang, Cong Wang, Yi Zhang, and Hao Yu

Naval Aviation University, Yantai 264001, China

Correspondence should be addressed to Zijie Jiang; 306010136@qq.com

Received 13 December 2023; Revised 28 February 2024; Accepted 1 March 2024; Published 20 March 2024

Academic Editor: Binbin Yan

Copyright © 2024 Zijie Jiang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper studies the salvo attack problem for multiple unmanned aerial vehicles (UAVs) against a maneuvering target, and a guidance scheme based on distributed model predictive control (DMPC) is presented to achieve cooperative interception with constraints of terminal impact angle and no-fly zone (or obstacle) avoidance. Firstly, for guaranteeing the synchronization of UAVs in calculating their acceleration commands, the assumed predictive trajectories are introduced, whose deviation from the actual state trajectories is limited by the designed compatibility constraints. Secondly, based on the velocity-obstacle model, the obstacle avoidance constraints are presented, and for guaranteeing the convergence of impact time and impact angles, the auxiliary controller and terminal ingredients are developed, which complete the design of DMPC cooperative guidance scheme. Subsequently, the rigorous proof for the convergence of the proposed guidance scheme is provided. Based on the above design, a complete implementation process of the guidance scheme is presented, in which each UAV uses the particle swarm optimization algorithm to solve the preprocessed local optimization problem, and only the shared information among neighbors is utilized for calculation. Finally, the numerical simulations are conducted under diverse cases, which demonstrate the effectiveness of the proposed guidance scheme when solving cooperative interception problems with terminal angle and obstacle avoidance constraints.

1. Introduction

As an ideal choice for performing dangerous and tedious tasks, UAVs have been widely used in various fields [1], which brings along the rapid development of the cooperative guidance technique [2–4]. Meanwhile, with the advancement of anti-interception methods and target's maneuverability, reaching the consensus on the impact time is no longer the only requirement for cooperative guidance, the coordination in intercept angles and the no-fly zone avoidance have also attracted attention for the purpose of maximizing the damage effect [5–7] and improving the survivability of UAVs in the presence of enemy radars or antiair weapons. Therefore, proposing a method that can meet all the above requirements is of great significance from both theoretical and practical viewpoints.

For the cooperative guidance with impact time constraints, researchers have made outstanding contributions. As the pioneering research in this field, the authors of ref. [8] proposed an impact-time-control guidance (ITCG) law for salvo attack of multiple missiles, which required a predetermined impact time as the global information for all the missiles. Due to the absence of communication among missiles, however, ITCG is not considered as a cooperative guidance law in its real sense by some researchers. For this concern, Zhao and Zhou [9] introduced coordination algorithms into the guidance of multiple missiles and presented a two-level hierarchical cooperative guidance architecture, which can be implemented in either centralized or distributed communication topology. Motivated by ref. [9], the authors of ref. [10] proposed a cooperative proportional navigation (CPN) law which uses the time-varying navigation gain for eliminating the difference in impact times among multiple missiles. Zhao et al. [11] extended CPN to three-dimensional situation and presented different coordination strategies. Because of the utilization of the local

guidance law such as PN, this kind of cooperative guidance is easy to implement. However, due to the fact that most of the above methods are based on the assumption that the velocities of UAVs are constants, the applicability of them is limited to some extent. For this reason, some researchers have managed to introduce the consistency theory into cooperative guidance [12, 13] and divide the design of the guidance law into two stages [14]. The first stage is to design the acceleration command along the direction of the line of sight using the consistency theory, to ensure the consensus on the impact time. The second stage is to design the acceleration command normal to the line of sight for guaranteeing that each UAV precisely hits the target. This idea of design is adopted by lots of researchers [15, 16] and has implications for studies on cooperative guidance with both impact time and impact angle constraints.

The cooperative guidance with impact time and impact angle constraints is widely concerned in recent years [17–20] for achieving the best destroying effect on the target. The authors of ref. [21] presented a novel closed-form guidance law for salvo attack of antiship missiles, based on the biased proportional navigation guidance (BPNG) law. It utilizes the error between the time-to-go and its designated value for feedback control and adds another biased term to BPNG, thereby satisfying both impact time and impact angle constraints. However, the communication among the UAVs is not needed for this method. Enlightened by ref. [14], Dong et al. [16] proposed a fixed-time cooperative guidance law to simultaneously attack a maneuvering target at desired terminal angles. It uses the consistency protocol, fixed-time control theory, and integral sliding mode control to design the tangential acceleration command and normal acceleration command, respectively. In ref. [18], researchers proposed the time- and angle-constrained cooperative guidance law based on the finite-time sliding mode control and the super-twisting control algorithm. Most of the above researches adopted the design idea in ref. [14] and achieved satisfactory guidance effects; however, there are still limitations in some aspects. First, the maneuver of the UAV is achieved through the acceleration command perpendicular to or along its velocity. Most of the above researches failed to take the saturation constraints of normal or tangential acceleration (with respect to velocity) into account. Second, the design in line-of-sight coordinate system is only intended to complete the interception task, while how to avoid entry into no-fly zones or collision with obstacles remains unresolved for the above studies.

No-fly zone (or obstacle) avoidance is one of the hotspots in UAV path planning, and some latest progress has been made on the basis of existing algorithms such as hybrid A * algorithm [22] and artificial potential field method [23]. Despite the great significance of obstacle avoidance in improving the survivability of UAVs when performing interception tasks, only a few studies on guidance or cooperative guidance have taken obstacle avoidance into consideration. Among them, the authors of ref. [24] presented a timeconstrained guidance approach for multimissiles attacking a stationary target based on model predictive control (MPC), which achieved no-fly zone avoidance by adding a safe distance-based penalty term to the local cost function. The authors of ref. [25] proposed a three-dimensional guidance approach for maneuvering target interception and handled obstacle avoidance constraints by decomposing the optimization problem into two subproblems, thus enabling the UAV to avoid obstacles while ensuring the guidance effect. The above researches introduced MPC into guidance and provided new ideas for solving such problems.

As an emerging control theory, MPC is able to handle control problems with multiple constraints and has been employed into cooperative guidance in recent years [26]. Besides the aforementioned ref. [24, 25], Zhao and Zhou [27] presented a cooperative guidance approach based on distributed model predictive control (DMPC), which could coordinate the impact time of a group of UAV missiles against the stationary target. Kang et al. [28] reformulated the impact time coordination as a consensus on the ranges and range rates of the missiles and presented a MPC-based cooperative guidance law. However, the optimization problem formulated in ref. [28] is solved in a centralized manner. The authors of ref. [29] designed a cooperative guidance algorithm with impact angle constraints based on distributed MPC, which could be applied to salvo attack against a stationary or low-speed maneuvering target. Although there are precedents indicating the successful use of MPC for cooperative guidance, some crucial issues have not yet been resolved, such as the design of terminal ingredients and compatibility constraints; they are all essential for guaranteeing the recursive feasibility and the convergence of the overall system [30, 31]. In addition, up to now, it is still challenging to develop a cooperative guidance approach that simultaneously achieves maneuvering target interception, consensus on the impact time, coordination in impact angles, and obstacle avoidance (for the sake of comparison, all the existing studies on MPC-based cooperative guidance are listed in Table 1).

Therefore, aiming at addressing the issues listed above, we develop a DMPC-based cooperative guidance scheme. The main contributions of this study are as follows.

- (1) Compared to refs [14, 16, 18] which separately design the acceleration commands normal to and along the LOS, we simultaneously design the normal and tangential acceleration commands in virtue of DMPC, and take the saturation of acceleration into consideration
- (2) Compared with refs [24, 27, 29], we further ensure the convergence of the multi-UAV system under DMPC by developing the compatibility constraints and the terminal ingredients. Different from designing a state-related constraint as in refs [32, 33], we derive an obstacle avoidance constraint applied to the control input of each UAV. This work ensures successful obstacle avoidance without reducing the feasible region of solution
- (3) To the best of our knowledge, it is the first time in cooperative guidance that maneuvering target interception, consensus on the impact time, coordination

| Reference | Target type | Consensus on impact time | Coordination in impact angles | No-fly zone or obstacle avoidance | Convergence guarantee |
|-----------|--------------------------|-----------------------------|----------------------------------|-----------------------------------|--------------------------|
| [24] | Stationary | Yes | No | Yes | No |
| [27] | Stationary | Yes | No | No | No |
| [28] | Stationary | Yes | No | No | No |
| [29] | Low-speed maneuvering | Yes | Yes | No | No |

TABLE 1: Comparison of existing studies on MPC-based cooperative guidance.

in impact angles, and obstacle avoidance are all achieved. Besides, different from ref. [28], all the UAVs participating in guidance share the information only with their neighbors and synchronously solve the optimization problem to obtain the acceleration commands

The remainder of this paper is organized as follows. In Section 2, the preliminaries about planar cooperative guidance geometry and DMPC are provided. In Section 3, the detailed design and convergence analysis are conducted for the DMPC-based cooperative guidance scheme. In Section 4, the complete implementation process of DMPC cooperative guidance scheme using particle swarm optimization algorithm is introduced. In Section 5, the numerical simulations are carried out under diverse cases, in which the proposed DMPC cooperative guidance scheme is compared with other methods. Finally, in Section 6, the conclusions are drawn.

2. Preliminary

In this section, some preliminaries about planar cooperative guidance geometry and distributed model predictive control are provided. For simplicity, the following assumptions are adopted in the research process.

Assumption 1. The UAVs and the target are considered as the geometric points in the plane.

Assumption 2. The autopilot dynamics of the UAVs are fast enough that the lag can be neglected.

2.1. Cooperative Guidance Geometry. The planar engagement geometry of multiple UAVs attacking a single maneuvering target is depicted in Figure 1, where M_i $(i = 1, 2, \dots, N_M)$ and T denote the UAV and the target, respectively; $obs_l(l = 1, 2, \dots, N_{obs})$ denotes the obstacle or the no-fly zone, which is modelled as a circle with a radius of $R_{obs\,i}$; and M_iXY and $M_iX_LY_L$ represent the inertial reference frame and the line-of-sight (LOS) reference frame, respectively. Meanings of other symbols are given as follows: θ_{Mi} and θ_T are the flight path angles of M_i and T, v_{Mi} and v_T are the velocities of M_i and T, q_i is the line-of-sight (LOS) angle with respect to M_i and T, r_i is the relative distance between each UAV and the target, $q_{obs\,il}$ is the LOS angle with respect to M_i and obs_{il}.

and d_{il} is the distance between M_i and obs_l . All the above angles are positive in the counterclockwise direction.

Assuming that all the UAVs are thrust controllable, the acceleration of M_i in inertial frame can be decomposed into a_{Mi} and u_{Mi} , where a_{Mi} is the normal acceleration and changes the direction of v_{Mi} and u_{Mi} is the tangential acceleration changing the magnitude of v_{Mi} , while the target is assumed to be passive vehicle with only direction of velocity controllable. Then, the relative motion between M_i and T can be described using Eq. (1), with Eq. (2) describing that between M_i and obs₁.

$$\begin{cases} \dot{r}_{i} = v_{T} \cos \sigma_{Ti} - v_{Mi} \cos \sigma_{Mi}, \\ \dot{q}_{i} = \frac{v_{T} \sin \sigma_{Ti} - v_{Mi} \sin \sigma_{Mi}}{r_{i}}, \\ \dot{\theta}_{Mi} = \frac{a_{Mi}}{v_{Mi}}, \quad \dot{\theta}_{T} = \frac{a_{T}}{v_{T}}, \\ \dot{v}_{Mi} = u_{Mi}, \\ \dot{q}_{il} = -v_{Mi} \cos \sigma_{obs\,il}, \end{cases}$$
(1)

Here,

$$\begin{cases} \sigma_{Ti} = \theta_T - q_i, \\ \sigma_{Mi} = \theta_{Mi} - q_i, \\ \sigma_{\text{obs }il} = \theta_{Mi} - q_{\text{obs }il}. \end{cases}$$
(3)

Remark 3. For the UAV equipped with radar seeker and inertial navigation system, θ_{Mi} , v_{Mi} , r_i , q_i , and $q_{obs,il}$ can be acquired in real time. Using this information, \dot{d}_{il} , $\dot{q}_{obs,il}$, and \dot{q}_i can be directly calculated.

 $\begin{cases} \dot{q}_{\text{obs }il} = -\frac{v_{Mi} \sin \sigma_{Mi}}{d_{il} + R_{\text{obs }l}}. \end{cases}$

For ease of design, we define $\boldsymbol{u}_i = \begin{bmatrix} a_{Mi} & u_{Mi} \end{bmatrix}^T$, and

$$\tilde{\boldsymbol{u}}_{i} = \begin{bmatrix} \tilde{\boldsymbol{u}}_{i1} \\ \tilde{\boldsymbol{u}}_{i2} \end{bmatrix} = \begin{bmatrix} -\sin \sigma_{Mi} & \cos \sigma_{Mi} \\ \cos \sigma_{Mi} & \sin \sigma_{Mi} \end{bmatrix} \cdot \begin{bmatrix} a_{Mi} \\ u_{Mi} \end{bmatrix}, \quad (4)$$

$$\boldsymbol{w}_{i} = \begin{bmatrix} w_{i1} \\ w_{i2} \end{bmatrix} = \begin{bmatrix} -a_{T} \sin \sigma_{Ti} \\ a_{T} \cos \sigma_{Ti} \end{bmatrix}.$$
 (5)

(2)



FIGURE 1: Planar cooperative guidance geometry.

UAV are described as

Take the derivative of \dot{r}_i and \dot{q}_i . Applying (4) and (5) yields

$$\begin{cases} \ddot{r}_{i} = \dot{q}_{i}^{2} r_{i} + w_{i1} - \tilde{u}_{i1}, \\ \ddot{q}_{i} = -\frac{2\dot{q}_{i}\dot{r}_{i}}{r_{i}} + \frac{w_{i2}}{r_{i}} - \frac{\tilde{u}_{i2}}{r_{i}}. \end{cases}$$
(6)

To estimate the remaining impact time of each UAV, we introduce a new variable called time-to-go, formulated as

$$t_{\rm goi} = -\frac{r_i}{\dot{r}_i}.$$
 (7)

Supposing that the desired terminal LOS angle of each UAV is q_{di} (commonly, there is $q_{di} \neq q_{dj}$ when $i \neq j$), the state variables for the guidance system are selected as

$$\boldsymbol{x}_{i} = \begin{bmatrix} \boldsymbol{x}_{i1} \\ \boldsymbol{x}_{i2} \\ \boldsymbol{x}_{i3} \\ \boldsymbol{x}_{i4} \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_{i} \\ \boldsymbol{t}_{\text{goi}} \\ \boldsymbol{q}_{i} - \boldsymbol{q}_{\text{d}i} \\ \dot{\boldsymbol{q}}_{i} \end{bmatrix}.$$
 (8)

Then, we can get a nonlinear system

$$\dot{\mathbf{x}}_{i} = \begin{bmatrix} -\frac{x_{i1}}{x_{i2}} \\ -1 + x_{i4}^{2} x_{i2}^{2} + \frac{x_{i2}^{2}}{x_{i1}} w_{i1} - \frac{x_{i2}^{2}}{x_{i1}} \tilde{u}_{i1} \\ x_{i4} \\ \frac{2x_{i4}}{x_{i2}} + \frac{w_{i2}}{x_{i1}} - \frac{\tilde{u}_{i2}}{x_{i1}} \end{bmatrix}.$$
(9)

In order to control all the UAVs to simultaneously hit the target at their respective terminal LOS angles without colliding with obstacles, the control objectives of each Guidance : $t = t_{f_i}, x_{i4}(t) = 0$,

Angle tracking :
$$t = t_{fi}, x_{i3}(t) = 0,$$
 (11)

Time consistency : $t = t_{fi}, x_{i2}(t) = x_{j2}(t) = 0, \forall j = 1, 2, \dots, N_M,$

(10)

Obstacle avoidance : $d_{il}(t) > 0, \forall l = 1, 2, \dots, N_{obs}$, (13)

where t_{fi} is the final time of guidance of each UAV, which satisfies $\dot{r}_i(t_{fi}) = 0$.

To achieve all the control objectives of UAVs, a cooperative guidance scheme based on the synchronous distributed model predictive control (DMPC) will be proposed below.

2.2. Distributed Model Predictive Control. With lower computational cost and more structural flexibility [31], DMPC has the potential to solve complex control problems of large-scale systems. Compared to traditional centralized MPC [28], DMPC requires each UAV to independently solve the constrained optimization problem and obtain their control actions, without the need for a central controller to complete all the calculations for all the UAVs.

Among several update strategies of DMPC (sequential DMPC [34], iterative DMPC [35], and synchronous DMPC [30–32, 36, 37]), the synchronous DMPC requires the least communication and computation resources, as it allows all the UAVs to synchronously calculate the control actions and communicate only with their neighbors. Therefore, the synchronous DMPC strategy is a preferred choice for a cooperative guidance problem with multiple constraints.

Prior to designing the cooperative guidance scheme, the basic model of DMPC is given in this subsection. Consider the DMPC problem as follows.

Problem 4. At sampling instant t_k , given the current state of M_i , i.e., $\mathbf{x}_i(t_k)$, the predictive state trajectories of the neighbors of M_i , i.e., $\mathbf{x}_{i-}(\tau|t_k)$, $\tau \in [t_k, t_k + T]$, the predictive

disturbances $\boldsymbol{w}_i(\tau|t_k)$, find the optimal control inputs

$$\boldsymbol{u}_{i}^{*}(\tau|t_{k}) = \underset{\boldsymbol{u}_{i}(\tau|t_{k})}{\arg\min} \int_{t_{k}}^{t_{k}+T} L_{i}(\tau|t_{k},\boldsymbol{x}_{i},\boldsymbol{x}_{i-},\boldsymbol{u}_{i},\boldsymbol{w}_{i}) \mathrm{d}\tau + g_{i}(\boldsymbol{x}_{i}(t_{k}+T|t_{k})),$$
(14)

subject to system equation (9),

$$\boldsymbol{u}_i(\tau|t_k) \in \mathbb{U}_i,\tag{15}$$

$$d_{il}(\tau|t_k) > 0, \forall l = 1, 2, \dots, N_{obs}.$$
 (16)

In Problem 4, $t_k = t_0 + k\delta$, $k = 0, 1, \dots, \delta$ denotes the sampling period, *T* represents the control horizon. $L_i(\tau | t_k, \mathbf{x}_i, \mathbf{x}_{i-}, \mathbf{u}_i, \mathbf{w}_i)$ is the stage cost given as in (17), which is established based on the control objectives of each UAV. $g_i(\mathbf{x}_i (t_k + T | t_k))$ is the terminal cost, and \mathbb{U}_i is the admissible set of the control inputs, whose specific form is given as (18). It should be noted that the terminal cost and the other terminal ingredients are essential in guaranteeing the convergence of the multi-UAV system and will be specifically designed in Section 3.2.

$$L_{i}(\tau|t_{k}, \mathbf{x}_{i}, \mathbf{x}_{i-}, \mathbf{u}_{i}, \mathbf{w}_{i}) = \lambda_{1i} \left(\underbrace{x_{i3}^{2}(\tau|t_{k}) + x_{i4}^{2}(\tau|t_{k})}_{\text{angle tracking}} \right) + \lambda_{2i} \underbrace{\left(\sum_{j \in \mathbb{N}_{i}} \left(x_{i2}(\tau|t_{k}) - x_{j2}(\tau|t_{k}) \right)^{2} \right)}_{\text{time consistency}},$$

$$(17)$$

$$\mathbb{U}_{i} = \left\{ \left[a_{Mi}, u_{Mi} \right]^{T} \middle| a_{Mi} \in \left[a_{Mi}^{\min}, a_{Mi}^{\max} \right], u_{Mi} \in \left[u_{Mi}^{\min}, u_{Mi}^{\max} \right] \right\}.$$
(18)

In Eq. (17), \mathbb{N}_i is the neighbor index set of M_i , and λ_{1i} and λ_{2i} are the weighting scalars given beforehand.

By solving the above constrained optimization problem, each UAV will obtain its real-time guidance command u_i $(\tau) = u_i^*(\tau|t_k), \ \tau \in [t_k, t_{k+1})$. As the system states evolve along the dynamic equation (9) from t_k to t_{k+1} , the above problem will be renewed, which ensures the real-time feedback and correction of DMPC. However, considering that the future system states are currently unknown, the following issues should be addressed before solving Problem 4:

- (1) The actual state trajectories of neighbors $\mathbf{x}_{i-}(\tau|t_k)$ cannot be acquired in advance
- (2) The exact value of disturbance $\boldsymbol{w}_i(\tau|t_k)$ cannot be obtained beforehand

For the former issue, we introduce the assumed predictive state trajectories replacing actual ones and employ the compatibility constraints to limit the deviation between the two trajectories, thereby ensuring the convergence of the overall multi-UAV system. For the latter issue, we propose a prediction method to estimate the target's states within the control horizon, which helps to compensate the disturbances caused by target maneuvering. All detailed analysis and design will be presented in the next section.

3. Design and Analysis of DMPC Cooperative Guidance Scheme

Based upon the previous studies on how to implement synchronous DMPC in cooperative guidance [24, 27, 29], in this section, we further present the associated constraints and terminal ingredients to ensure the convergence of the overall multi-UAV system. Additionally, our work also achieves both obstacle avoidance and cooperative interception, which has not yet been addressed in relevant researches.

3.1. Assumed Predictive State Trajectories and Prediction for System Disturbances. In order to synchronously calculate the control action, each UAV needs to know the predictive state trajectories of its neighbors according to (14) and (17). However, since any UAV cannot obtain its actual state trajectory before solving the optimization problem (Problem 4), an assumed predictive state trajectory is needed as a substitute.

To distinguish the two different trajectories, we denote the actual (optimal) control inputs as $\boldsymbol{u}_i^*(\tau|t_k)$ and the assumed control inputs as $\hat{\boldsymbol{u}}_i(\tau|t_k)$. The actual predictive state trajectory [30] refers to the system state trajectory obtained by implementing the optimal control inputs $\boldsymbol{u}_i^*(\tau|t_k)$ and is denoted as $\boldsymbol{x}_i^*(\tau|t_k)$, while the assumed predictive state trajectory $\hat{\boldsymbol{x}}_i(\tau|t_k)$ is obtained by implementing $\hat{\boldsymbol{u}}_i(\tau|t_k)$.

The assumed control inputs are constructed as (19), which consist of the optimal control inputs at the previous sampling instant, i.e., $\boldsymbol{u}_i^*(\tau|t_{k-1})$, and the virtual control inputs generated by the auxiliary controller, i.e., $\boldsymbol{u}_i^{\kappa}(\boldsymbol{x}_i^*(\tau|t_{k-1}))$. The auxiliary controller is the crucial guarantee of the system stability and will be designed in Section 3.2.

$$\widehat{u}_{i}(\tau|t_{k}) = \begin{cases} u_{i}^{*}(\tau|t_{k-1}), & \tau \in [t_{k}, t_{k-1} + T), \\ u_{i}^{\kappa}(x_{i}^{*}(\tau|t_{k-1})), & \tau \in [t_{k-1} + T, t_{k} + T). \end{cases}$$
(19)

Accordingly, the assumed predictive state trajectory can be represented as

$$\widehat{x}_{i}(\tau|t_{k}) = \begin{cases} x_{i}^{*}(\tau|t_{k-1}), & \tau \in [t_{k}, t_{k-1} + T), \\ x_{i}^{\kappa}(\tau|t_{k-1}), & \tau \in [t_{k-1} + T, t_{k} + T). \end{cases}$$
(20)

In which, $\mathbf{x}_{i}^{\kappa}(\tau|t_{k-1})$ is the system state obtained by implementing $\mathbf{u}_{i}^{\kappa}(\mathbf{x}_{i}^{*}(\tau|t_{k-1}))$.

Applying the assumed state trajectories, UAVs are able to synchronously compute control actions as long as their neighbors send the assumed trajectories to them in advance; accordingly, the stage cost will be rewritten as follows:

$$L_{i}(\tau|t_{k}, \mathbf{x}_{i}, \widehat{\mathbf{x}}_{i-}, \mathbf{u}_{i}, \mathbf{w}_{i}) = \lambda_{1i} \Big(x_{i3}^{2}(\tau|t_{k}) + x_{i4}^{2}(\tau|t_{k}) \Big) \\ + \lambda_{2i} \left(\sum_{j \in \mathbb{N}_{i}} \big(x_{i2}(\tau|t_{k}) - \widehat{x}_{j2}(\tau|t_{k}) \big)^{2} \right).$$
(21)

However, the introduction of assumed state trajectories induces the uncertain deviation from the actual ones; in order for eliminating the effect of such deviation and ensuring the convergence of the overall system, a compatibility constraint will be given in the following.

According to (21), the only assumed state variable involved in stage cost is x_{i2} ; thus, the deviation between the assumed predictive state trajectory and the actual one is defined as $\varepsilon_i(\tau|t_k) = \hat{x}_{i2}(\tau|t_k) - x_{i2}(\tau|t_k)$. Then, the compatibility constraint is given by

$$C_i^{\text{CB}}(\tau|t_k) = |\varepsilon_i(\tau|t_k)| - \upsilon_i(t_k) \le 0,$$
(22)

$$v_i(t_k) = \frac{-b_i(t_k) + \sqrt{b_i^2(t_k) - 4a_i \cdot c_i(t_k)}}{2a_i},$$
 (23)

$$\begin{cases} a_i = \lambda_{2i} |N_i| [T + (\gamma_i - 1)\delta] + 2(T - \delta) \sum_{j \in \mathbb{N}_i} \lambda_{2j}, \\ b_i(t_k) = 2(T - \delta) \sum_{j \in \mathbb{N}_i} \lambda_{2j} \varphi_{ji}(t_k), \\ c_i(t_k) = -\delta \gamma_i \lambda_{2i} \sum_{j \in \mathbb{N}_i} \zeta_{ij}^2(t_k), \end{cases}$$
(24)

Wherein $|N_i|$ is the number of the neighbors of M_i , φ_{ii}

 $\begin{aligned} &(t_k) = \varphi_{ji}(t_k) = \max_{\tau \in [t_k, t_k + T]} |\widehat{x}_{i2}(\tau|t_k) - \widehat{x}_{j2}(\tau|t_k)|, \ \zeta_{ij}(t_k) = \zeta_{ji}(t_k) \\ &= \min_{\tau \in [t_k, t_k + T]} |\widehat{x}_{i2}(\tau|t_k) - \widehat{x}_{j2}(\tau|t_k)|, \text{ and } \gamma_i \in (0, 1) \text{ is the preset scalar.} \end{aligned}$

By employing compatibility constraints, the action taken by the UAV will not deviate significantly from that it planned and told its neighbors, and the role of which in ensuring the convergence of the overall system will be detailed in Section 3.3.

Due to the absence of interaction like that between the UAV and its neighbors, the UAV is unable to receive the predictive state trajectory from the target, so the only means to acquire the future states of the target is via prediction. From equations (1) and (5), the disturbances in system (9) are caused by the change of the target's flight path angle; hence, the prediction object is chosen as $e = \theta_T$.

For the prediction object, we assume the following.

Assumption 5. *e* is *N*-order differentiable on interval $[t_{k-N+1}, t_k + T]$.

Assumption 6. e and its arbitrary-order derivatives are bounded.

Applying Assumptions 5 and 6, the following theorem can be derived.

Theorem 7. Given $(t_{k-N+1}, e(t_{k-N+1})), \dots, (t_k, e(t_k))$, there exist $\alpha_0, \alpha_1, \dots, \alpha_{N-1}$ such that, for any $i = 0, 1, \dots, N-1$, \tilde{e} $(t_{k-i}|t_k) - e(t_{k-i}) = 0$ and $\lim_{\tau \to -t_k^+} |\widehat{e}(\tau|t_k) - e(\tau)| = 0$, where

$$\tilde{e}(t_{k-i}|t_k) = \sum_{j=0}^{N-1} \alpha_j (t_{k-i} - t_k)^j,$$
(25)

$$\widehat{e}(\tau|t_k) = \sum_{j=0}^{N-1} \alpha_j (\tau - t_k)^j, \qquad (26)$$

whose proof follows.

Proof of Theorem 7. Using Taylor's mean value theorem and Assumption 5, for $\forall \tau \in [t_{k-N+1}, t_k + T]$, there holds

$$e(\tau) = \sum_{j=0}^{N-1} \frac{e^{(j)}(t_k)}{j!} (\tau - t_k)^j + R_{N-1}(\tau - t_k).$$
(27)

Here, $R_{N-1}(\tau - t_k) = (e^{(N)}(\tilde{t})/N!)(\tau - t_k)^N$ is the N-1-order Lagrange remainder, and \tilde{t} is a certain value between τ and t_k .

Let $\Delta_i = t_{k-i} - t_k$, $i = 0, 1, \dots, N - 1$. Constructing matrix

$$\boldsymbol{\Xi} = \begin{bmatrix} 1 & \cdots & \boldsymbol{\Delta}_{0}^{N-1} \\ \vdots & & \vdots \\ 1 & \cdots & \boldsymbol{\Delta}_{N-1}^{N-1} \end{bmatrix},$$
(28)

and vectors

$$\boldsymbol{e} = [e(t_k), \cdots, e(t_{k-N+1})]^T,$$

$$\boldsymbol{\beta} = \left[e(t_k), \cdots, \frac{e^{(N-1)}(t_k)}{(N-1)!}\right],$$

$$\boldsymbol{R} = [R_{N-1}(0), \cdots, R_{N-1}(t_{k-N+1} - t_k)]^T,$$

(29)

yields

$$\boldsymbol{e} = \boldsymbol{\Xi} \cdot \boldsymbol{\beta} + \boldsymbol{R}. \tag{30}$$

Let

$$\boldsymbol{\alpha} = \left[\alpha_0, \alpha_1, \cdots, \alpha_{N-1}\right]^T = \boldsymbol{\Xi}^{-1} \cdot \boldsymbol{e}.$$
(31)

Then, we obtain $\Xi \cdot \alpha - e = \Xi \cdot \Xi^{-1}e - e = 0$; that is, for any $i = 0, 1, \dots, N-1$, $\tilde{e}(t_{k-i}|t_k) - e(t_{k-i}) = 0$. Moreover, constructing $\tau = [1, \dots, (\tau - t_k)^{N-1}]$ yields $e(\tau) = \tau \cdot \beta + R_{N-1}(\tau - t_k)$ and

$$\widehat{e}(\tau|t_{k}) - e(\tau) = \tau(\alpha - \beta) + R_{N-1}(\tau - t_{k}) \\
= \tau \cdot \Xi^{-1} R + R_{N-1}(\tau - t_{k}) = \begin{bmatrix} 1, \cdots, (\tau - t_{k})^{N-1} \end{bmatrix} \\
\cdot \begin{bmatrix} 1 & 0 & \cdots & 0 \\ & * \end{bmatrix} \cdot \begin{bmatrix} R_{N-1}(0) \\ \vdots \\ R_{N-1}(t_{k-N+1} - t_{k}) \end{bmatrix} \\
+ R_{N-1}(\tau - t_{k}) = \begin{bmatrix} 1, \cdots, (\tau - t_{k})^{N-1} \end{bmatrix} \\
\cdot \begin{bmatrix} R_{N-1}(0) \\ & * \end{bmatrix} + R_{N-1}(\tau - t_{k}).$$
(32)

Applying Assumption 6, we get $\lim_{\tau \longrightarrow t_k^+} |\hat{e}(\tau|t_k) - e(\tau)| = 0$; thus, Theorem 7 holds.

From Theorem 7, if T is set small, (26) and (31) can be used for predicting the value of e within the control horizon, and the derivative of e can be predicted with

$$\widehat{\dot{e}}(\tau|t_k) = \begin{cases} \alpha_1 & \tau = t_k, \\ \dot{\bar{e}}(\tau|t_k) & \tau > t_k. \end{cases}$$
(33)

Then, using (1), (3), and (5), we can calculate the predictive disturbances $\boldsymbol{w}_i(\tau|t_k)$.

Remark 8. According to Theorem 7, by updating the observation data in the manner of sliding window, the above designed prediction method can provide a relatively accurate estimation and offset the effect induced by the disturbances within a short horizon. Therefore, the estimation error and the associated robustness constraints [38] will not be discussed specifically in the following sections.

3.2. Design of the Auxiliary Controller, Terminal Ingredients, and Obstacle Avoidance Constraints. Avoidance of obstacles or no-fly zones is the primary assurance for safely accomplishing the interception tasks. Nevertheless, it is rarely studied in previous literature on cooperative guidance, which is mainly because the associated problem cannot be integrated into the framework of the traditional control theory. In this subsection, we tackle this issue by employing the obstacle avoidance constraints under DMPC.

According to the traditional VO model depicted in Figure 2, the UAV is able to avoid the obstacle as long as its flight path angle meet:

$$\sigma_{\text{obs}\,il}^2 \ge \left(\arcsin \frac{R_{\text{obs}\,l}}{d_{il} + R_{\text{obs}\,l}} \right)^2. \tag{34}$$

Based on (34), we define

$$f_{il}(t) = \sigma_{\text{obs}\,il}^2(t) - \left(\arcsin\frac{R_{\text{obs}\,l}}{d_{il}(t) + R_{\text{obs}\,l}}\right)^2 \qquad (35)$$

and provide the following theorem.



FIGURE 2: Planar velocity-obstacle model.

Theorem 9. Suppose there exists $t^* > t$ such that $d_{il}(t^*) = 0$. If there are $d_{il}(\tau) > 0$ and $f_{il}(\tau) < 0$ for any $\tau \in [t, t^*)$, then the sufficient condition for the UAV to successfully avoid obstacles at t^* is as follows: $\dot{f}_{il}(\tau) \ge (\dot{d}_{il}(\tau)/d_{il}(\tau))f_{il}(\tau)$ holds for any $\tau \in [t, t^*)$.

Proof of Theorem 9. Under the condition of Theorem 9, there is

$$\frac{\dot{f}_{il}(\tau)}{f_{il}(\tau)} \le \frac{\dot{d}_{il}(\tau)}{d_{il}(\tau)}.$$
(36)

Integrating (36) from τ to t^* yields

$$\int_{\tau}^{t^*} \frac{\dot{f}_{il}}{f_{il}} \mathrm{d}\tau \le \int_{\tau}^{t^*} \frac{\dot{d}_{il}}{d_{il}} \mathrm{d}\tau.$$
(37)

Namely,

$$\ln\left(\frac{f_{il}(t^*)}{f_{il}(\tau)}\right) \le \ln\left(\frac{d_{il}(t^*)}{d_{il}(\tau)}\right).$$
(38)

Following the monotonicity of ln (\cdot) and applying $f_{il}(\tau)$ < 0 yields

$$f_{il}(t^*) \ge \frac{d_{il}(t^*)}{d_{il}(\tau)} f_{il}(\tau) = 0.$$
(39)

According to (34), the UAV is able to avoid the obstacle at t^* , which completes the proof.

Based on Theorem 9, the obstacle avoidance constraint and its triggering condition are derived as follows:

$$\begin{cases} a_{\mathrm{M}i}(\tau|t_k) \ge C_{il}^{\mathrm{OA}}(\tau|t_k), & \text{for } \sigma_{\mathrm{obs}\,il}(\tau|t_k) \ge 0, f_{il}(\tau|t_k) < 0, \\ a_{\mathrm{M}i}(\tau|t_k) \le C_{il}^{\mathrm{OA}}(\tau|t_k), & \text{for } \sigma_{\mathrm{obs}\,il}(\tau|t_k) < 0, f_{il}(\tau|t_k) < 0, \end{cases}$$

$$\tag{40}$$



FIGURE 3: Comparison between the designed obstacle avoidance constraint and the traditional VO method used in [32, 33]: (a) obstacle avoidance trajectories; (b) feasible solution region during the obstacle avoidance process.

where

$$C_{il}^{OA}(\tau|t_{k}) = \frac{\nu_{Mi}(\tau|t_{k})\dot{d}_{il}(\tau|t_{k})f_{il}(\tau|t_{k})}{2\sigma_{obs\,il}(\tau|t_{k})\dot{d}_{il}(\tau|t_{k})} + \nu_{Mi}(\tau|t_{k})\dot{q}_{obs\,il}(\tau|t_{k}) - \frac{\nu_{Mi}(\tau|t_{k})\dot{d}_{il}(\tau|t_{k})R_{obs\,l} \arcsin\left(R_{obs\,l}/d_{il}(\tau|t_{k}) + R_{obs\,l}\right)}{\sigma_{obs\,il}(\tau|t_{k})[d_{il}(\tau|t_{k}) + R_{obs\,l}]\sqrt{(d_{il}(\tau|t_{k}) + R_{obs\,l})^{2} - R_{obs\,l}^{2}}}.$$

$$(41)$$

Figure 3 shows the difference between the obstacle avoidance constraints adopted by refs. [32, 33] and those designed in this paper. By employing the latter ones, the feasible region is significantly augmented, and the saturation of normal acceleration is alleviated. Besides, according to (40), the obstacle avoidance constraint is directly imposed on the control input a_{Mi} , so we can redefine the admissible set of the control inputs as

$$\begin{aligned}
& \bigcup_{i}(\tau|t_{k}) = \bigcap_{l=1}^{N_{obs}} \mathbf{U}_{il}(\tau|t_{k}), \end{aligned}$$

$$\begin{aligned}
& \mathbf{U}_{i}(\tau|t_{k}) = \begin{cases} a_{Mi}, u_{Mi} \end{bmatrix}^{T} \begin{vmatrix} a_{Mi} \in [a_{Mi}^{\min}, a_{Mi}^{\max}], u_{Mi} \in [u_{Mi}^{\min}, u_{Mi}^{\max}], & \text{for } f_{il}(\tau|t_{k}) \ge 0 \\ a_{Mi} \in [\max\{a_{Mi}^{\min}, C_{il}^{OA}(\tau|t_{k})\}, a_{Mi}^{\max}], u_{Mi} \in [u_{Mi}^{\min}, u_{Mi}^{\max}], & \text{for } \sigma_{obs\,il}(\tau|t_{k}) \ge 0, f_{il}(\tau|t_{k}) < 0 \\ a_{Mi} \in [a_{Mi}^{\min}, \min\{a_{Mi}^{\max}, C_{il}^{OA}(\tau|t_{k})\}], u_{Mi} \in [u_{Mi}^{\min}, u_{Mi}^{\max}], & \text{for } \sigma_{obs\,il}(\tau|t_{k}) < 0, f_{il}(\tau|t_{k}) < 0 \\ a_{Mi} \in [a_{Mi}^{\min}, \min\{a_{Mi}^{\max}, C_{il}^{OA}(\tau|t_{k})\}], u_{Mi} \in [u_{Mi}^{\min}, u_{Mi}^{\max}], & \text{for } \sigma_{obs\,il}(\tau|t_{k}) < 0, f_{il}(\tau|t_{k}) < 0 \\ \end{cases}
\end{aligned}$$

$$(42)$$

Obviously, $\mathbb{U}_i(\tau | t_k)$ is a compact set, and we suppose

$$\mathbb{U}_{i}(\tau|t_{k}) = \left\{ \left[a_{Mi}, u_{Mi} \right]^{T} \middle| a_{Mi} \in \left[a_{Mi}^{\text{OAmin}}(\tau|t_{k}), a_{Mi}^{\text{OAmax}}(\tau|t_{k}) \right], u_{Mi} \in \left[u_{Mi}^{\min}, u_{Mi}^{\max} \right] \right\}.$$
(44)

Thereupon, the design of obstacle avoidance constraints is completed.

For a conventional MPC problem, the auxiliary controller and other terminal ingredients are of great significance in ensuring the recursive feasibility and system stability; nonetheless, in most studies on DMPC cooperative guidance [27–29], their development remains a challenging issue.

For the auxiliary controller and terminal ingredients, the following requirements should be satisfied [30, 31].

Definition 10. The terminal set $\Omega_i(t_k)$ and the auxiliary controller $u_i^{\kappa}(\cdot)$ are such that for any $x_i(t|t_k) \in \Omega_i(t_k)$ and $\tau \ge t$, by implementing $u_i(\tau) = u_i^{\kappa}(x_i(\tau|t_k))$

$$\boldsymbol{x}_i(\tau|\boldsymbol{t}_k) \in \boldsymbol{\Omega}_i(\boldsymbol{t}_k),\tag{45}$$

$$\boldsymbol{u}_{i}^{\kappa}(\boldsymbol{x}_{i}(\tau|t_{k})) \in \mathbb{U}_{i}(\tau|t_{k}), \tag{46}$$

$$\sum_{i=1}^{N_{\mathrm{M}}} (\dot{g}_i(\boldsymbol{x}_i(\tau|t_k)) + L_i(\tau|t_k, \boldsymbol{x}_i, \boldsymbol{x}_{i-}, \boldsymbol{u}_i, \boldsymbol{w}_i)) \le 0.$$
(47)

In Definition 10, conditions (45)–(47) represent the positive invariance, control input admissibility, and local convergence, respectively. Based on the above, we design

the terminal ingredients and the auxiliary controller as follows.

The terminal cost is defined as

$$g_{i}(\mathbf{x}_{i}(\tau|t_{k})) = \rho_{1i} \left[(x_{i3}(\tau|t_{k}) + \mu_{i}x_{i4}(\tau|t_{k}))^{2} + x_{i4}^{2}(\tau|t_{k}) \right] + \rho_{2i}(x_{i2}(\tau|t_{k}) - x_{ri}(\tau))^{2},$$
(48)

where $x_{ri}(\tau)$ is the reference trajectory, selected as

$$x_{\rm ri}(\tau) = \frac{\sum_{i=1}^{N_{\rm M}} x_{i2}(0)}{N_{\rm M}} - \tau.$$
(49)

The terminal set is given as

$$\Omega_{i}(t_{k}) = \left\{ x_{i}(\tau|t_{k}) \middle| \begin{array}{l} (x_{i3}(\tau|t_{k}) + \mu_{i}x_{i4}(\tau|t_{k}))^{2} + x_{i4}^{2}(\tau|t_{k}) \leq \Lambda_{1i} \\ \wedge (x_{i2}(\tau|t_{k}) - x_{ri}(\tau))^{2} \leq \Lambda_{2i} \end{array} \right\}.$$
(50)

The auxiliary controller is designed in the form of

$$\boldsymbol{u}_{i}^{\kappa}(\boldsymbol{x}_{i}(\tau|t_{k})) = \begin{bmatrix} -\sin\sigma_{\mathrm{M}i}(\tau|t_{k}) & \cos\sigma_{\mathrm{M}i}(\tau|t_{k}) \\ \cos\sigma_{\mathrm{M}i}(\tau|t_{k}) & \sin\sigma_{\mathrm{M}i}(\tau|t_{k}) \end{bmatrix} \cdot \tilde{\boldsymbol{u}}_{i}^{\kappa}(\boldsymbol{x}_{i}(\tau|t_{k})), \tag{51}$$

$$\tilde{\boldsymbol{u}}_{i}^{\kappa}(\boldsymbol{x}_{i}(\tau|t_{k})) = \begin{bmatrix} \tilde{\boldsymbol{u}}_{i1}^{\kappa}(\boldsymbol{x}_{i}(\tau|t_{k})) \\ \tilde{\boldsymbol{u}}_{i2}^{\kappa}(\boldsymbol{x}_{i}(\tau|t_{k})) \end{bmatrix} = \begin{bmatrix} x_{i1}(\tau|t_{k})x_{i4}^{2}(\tau|t_{k}) + k_{2i}\frac{x_{i1}(\tau|t_{k})}{x_{i2}^{2}(\tau|t_{k})}(x_{i2}(\tau|t_{k}) - x_{ri}(\tau)) \\ \frac{2x_{i1}(\tau|t_{k})x_{i4}(\tau|t_{k})}{x_{i2}(\tau|t_{k})} + k_{1i}x_{i1}(\tau|t_{k})x_{i3}(\tau|t_{k}) + \frac{1 - k_{1i}(\mu_{i}^{2} + 1)}{\mu_{i}}x_{i1}(\tau|t_{k})x_{i4}(\tau|t_{k}) \end{bmatrix} + \boldsymbol{w}_{i}(\tau|t_{k}). \tag{52}$$

Remark 11. In this paper, the local convergence of each UAV subsystem is mainly guaranteed by the terminal cost and the auxiliary controller, so the terminal region can be determined conservatively by setting Λ_{1i} and Λ_{2i} large. If a less conservative terminal region is required, Λ_{1i} and Λ_{2i} should be properly determined adopting methods proposed in ref. [36, 37]. Besides, different from ref. [39] which adopts dual-mode MPC strategy, our strategy is the same as ref. [30]; i.e., the control action will be obtained by solving the optimization problem regardless of whether the states are inside the terminal region or not, rather than applying the auxiliary controller as the actual control action.

In order for making the above items meet the conditions in Definition 10, the following assumption is required.

Assumption 12. Given weighting scalars λ_{1i} and λ_{2i} (in stage cost (17)) and μ_i (in terminal cost (48)), there exist positive scalars ρ_{1i} , ρ_{2i} , k_{1i} , and k_{2i} such that (53)–(55), and there always exists an auxiliary controller satisfying $u_i^{\kappa}(x_i(\tau|t_k)) \in \mathbb{U}_i(\tau|t_k)$ for any $\mathbf{x}_i(\tau|t_k) \in \mathbf{\Omega}_i(t_k)$.

$$\rho_{1i} > \frac{\lambda_{1i} (\mu_i^2 + 1)^2 + \lambda_{1i} \mu_i^2}{2\mu_i},$$
(53)

$$\frac{\lambda_{1i}}{2\rho_{1i}\mu_i} < k_{1i} < \frac{2\rho_{1i} - \lambda_{1i}\mu_i}{2\rho_{1i}(\mu_i^2 + 1)^2},\tag{54}$$

$$k_{2i} > \frac{\lambda_{2i}|N_i| + \sum_{j \in \mathbb{N}_i} \lambda_{2j}}{\rho_{2i}}.$$
(55)

Remark 13. In some cases, the above assumption may not hold due to the contradiction between (46) and (54), especially when the obstacle avoidance constraints in (40) are triggered. The priority in these cases is to ensure that condition (46) is met, while as a cost, the convergence of the overall system may be broken and restored soon after the above contradiction disappears.

Now, the design of auxiliary controller and terminal ingredients is completed. By substituting the terminal, compatibility, and obstacle avoidance constraints into Problem 4, the optimization problem to be solved by each UAV can be expressed as follows.

Problem 14. At sampling instant t_k , given the current state of M_i , i.e., $\mathbf{x}_i(t_k)$, the assumed predictive state trajectories of the neighbors of M_i , i.e., $\hat{\mathbf{x}}_{i-}(\tau|t_k)$, $\tau \in [t_k, t_k + T]$, the predictive disturbances $\mathbf{w}_i(\tau|t_k)$, find the optimal control inputs.

$$\boldsymbol{u}_{i}^{*}(\boldsymbol{\tau}|\boldsymbol{t}_{k}) = \operatorname*{arg\,min}_{\boldsymbol{u}_{i}(\boldsymbol{\tau}|\boldsymbol{t}_{k})} J_{i}(\boldsymbol{t}_{k}, \boldsymbol{x}_{i}, \boldsymbol{\hat{x}}_{i-}, \boldsymbol{u}_{i}, \boldsymbol{w}_{i}), \quad (56)$$

subject to dynamic equation (9), compatibility constraint (22), admissibility constraint of control inputs (considering obstacle avoidance) (57), and terminal set constraint (58).

$$\boldsymbol{u}_i(\tau|\boldsymbol{t}_k) \in \mathbb{U}_i(\tau|\boldsymbol{t}_k), \tag{57}$$

$$\boldsymbol{x}_i(t_k + T|t_k) \in \boldsymbol{\Omega}_i(t_k), \tag{58}$$

wherein,

$$\begin{split} J_{i}(t_{k}, \boldsymbol{x}_{i}, \widehat{\boldsymbol{x}}_{i-}, \boldsymbol{u}_{i}, \boldsymbol{w}_{i}) &= \int_{t_{k}}^{t_{k}+T} L_{i}(\tau | t_{k}, \boldsymbol{x}_{i}, \widehat{\boldsymbol{x}}_{i-}, \boldsymbol{u}_{i}, \boldsymbol{w}_{i}) \mathrm{d}\tau + g_{i}(\boldsymbol{x}_{i}(t_{k} + T | t_{k})) \\ &= \int_{t_{k}}^{t_{k}+T} \lambda_{1i} (x_{i3}^{2}(\tau | t_{k}) + x_{i4}^{2}(\tau | t_{k})) \\ &+ \lambda_{2i} \left(\sum_{j \in \mathbb{N}_{i}} \left(x_{i2}(\tau | t_{k}) - \widehat{x}_{j2}(\tau | t_{k}) \right)^{2} \right) \mathrm{d}\tau \\ &+ \rho_{1i} \left[(x_{i3}(t_{k} + T | t_{k}) + \mu_{i}x_{i4}(t_{k} + T | t_{k}))^{2} \\ &+ x_{i4}^{2}(t_{k} + T | t_{k}) \right] + \rho_{2i}(x_{i2}(t_{k} + T | t_{k}) - x_{ri}(t_{k} + T))^{2}. \end{split}$$

$$(59)$$

3.3. Convergence Analysis of the Multi-UAV System under DMPC. As Problem 14 is constructed following the common practices of MPC, the convergence analysis of the multi-UAV system adopting DPMC cooperative guidance scheme will be carried out in two steps:

- Analysis of recursive feasibility: under auxiliary controller (51) and (52) and with Assumption 5, all the conditions in Definition 10 will be satisfied, and the optimization problem (Problem 14) is feasible at each sampling instant (see Theorem 15)
- (2) Analysis of closed-loop stability: employing compatibility constraint (22) and condition (47), the multi-UAV system using *u*^{*}_i(τ|t_k), τ ∈ [t_k, t_{k+1}) as the real-time guidance command is exponentially stable (see Theorem 16)

The associated theorems and their proofs are given as follows.

Theorem 15. Let the terminal cost and terminal set be given as in (48) and (50); suppose Assumption 12 holds, $\mathbf{x}_i(t|t_k) \in \mathbf{\Omega}_i(t_k)$, and Problem 14 is feasible at t_{k-1} . Then, using auxiliary controller (51), (52), and assumed control inputs (19),

- (a) for any $\tau \ge t$, the requirements in Definition 10 are satisfied
- (b) the optimization problem (Problem 14) is feasible at t_k

Proof of Theorem 15 (for ease of derivation, we omit " (τ) " or " $(\tau|t_k)$ " in (60)–(66)). According to Assumption 12, the auxiliary controller belongs to the admissible set of control inputs, so condition (46) is met.

Under auxiliary controller (51) and (52), according to dynamic equation (9), there are

$$\dot{x}_{i2} = -k_{2i}(x_{i2} - x_{ri}) - 1, \qquad (60)$$

$$\dot{x}_{i4} = -k_{1i}x_{i3} - \frac{1 - k_{1i}(\mu_i^2 + 1)}{\mu_i}x_{i4}.$$
(61)

Applying (54), (60), and (61) yields

$$\begin{aligned} \frac{d}{d\tau} \left((x_{i3} + \mu_i x_{i4})^2 + x_{i4}^2 \right) &= 2(x_{i3} + \mu_i x_{i4})(x_{i4} + \mu_i \dot{x}_{i4}) + 2x_{i4} \dot{x}_{i4} \\ &= 2(x_{i3} x_{i4} + \mu_i x_{i4}^2 + \mu_i x_{i3} \dot{x}_{i4} + \mu_i^2 x_{i4} \dot{x}_{i4} + x_{i4} \dot{x}_{i4}) \\ &= 2\left(-\mu_i k_{1i} x_{i3}^2 + \frac{-1 + k_{1i} (\mu_i^2 + 1)^2}{\mu_i} x_{i4}^2 \right) \\ &< -\frac{\lambda_{1i}}{\rho_{1i}} \left(x_{i3}^2 + x_{i4}^2 \right) < 0, \end{aligned}$$
(62)

$$\frac{d}{d\tau} \left((x_{i2} - x_{ri})^2 \right) = 2(x_{i2} - x_{ri}) (\dot{x}_{i2} - \dot{x}_{ri}) = -2k_{2i}(x_{i2} - x_{ri})^2 < 0.$$
(63)

Thus, for any $\tau \ge t$, condition (45) is met. Applying inequality

$$\begin{split} \sum_{i=1}^{N_{M}} \lambda_{2i} \sum_{j \in \mathbb{N}_{i}} \left(x_{i2} - x_{j2} \right)^{2} &\leq 2 \sum_{i=1}^{N_{M}} \lambda_{2i} \sum_{j \in \mathbb{N}_{i}} \left(x_{i2} - x_{ri} \right)^{2} + \left(x_{j2} - x_{rj} \right)^{2} \\ &= 2 \sum_{i=1}^{N_{M}} \left(\lambda_{2i} |N_{i}| + \sum_{j \in \mathbb{N}_{i}} \lambda_{2j} \right) \left(x_{i2} - x_{ri} \right)^{2}, \end{split}$$

$$(64)$$

the condition (47) holds if

$$2\left(\lambda_{2i}|N_i| + \sum_{j \in \mathbb{N}_i} \lambda_{2j}\right) (x_{i2} - x_{ri})^2 + \lambda_{1i} (x_{i3}^2 + x_{i4}^2) + \dot{g}_i(\boldsymbol{x}_i) < 0.$$
(65)

Substituting (55), (62), and (63) into $\dot{g}_i(\mathbf{x}_i)$ yields

$$\dot{g}_{i}(x_{i}) = \rho_{1i} \frac{d}{d\tau} \left((x_{i3} + \mu_{i} x_{i4})^{2} + x_{i4}^{2} \right) + \rho_{2i} \frac{d}{d\tau} \left((x_{i2} - x_{ri})^{2} \right) < -\lambda_{1i} \left(x_{i3}^{2} + x_{i4}^{2} \right) - 2\rho_{2i} k_{2i} (x_{i2} - x_{ri})^{2} < -\lambda_{1i} \left(x_{i3}^{2} + x_{i4}^{2} \right) - 2 \left(\lambda_{2i} |N_{i}| + \sum_{j \in \mathbb{N}_{i}} \lambda_{2j} \right) (x_{i2} - x_{ri})^{2}.$$

$$(66)$$

According to (66), inequality (65) holds, and condition (47) is satisfied. Thereupon, conclusion (a) holds.

Since the optimization problem (Problem 14) is feasible at t_{k-1} , the optimal control inputs and the optimal terminal states at t_{k-1} are such that $\boldsymbol{u}_i^*(\tau|t_{k-1}) \in \mathbb{U}_i(\tau|t_{k-1})$ and $\boldsymbol{x}_i^*(t_{k-1}+T|t_{k-1}) \in \boldsymbol{\Omega}_i(t_{k-1})$.

Consider the assumed control inputs at t_k , whose corresponding predictive state trajectory satisfies (20). It implies that, for any $\tau \in [t_k, t_{k-1} + T)$, by implementing inputs $u_i(\tau | t_k) = u_i^*(\tau | t_{k-1})$ (following the invariance property of $U_i(\tau | t_k)$, $\Omega_i(t_k)$), there are

$$\boldsymbol{u}_i(\tau|\boldsymbol{t}_k) \in \mathbb{U}_i(\tau|\boldsymbol{t}_k), \tag{67}$$

$$\boldsymbol{x}_i(t_{k-1} + T|t_k) \in \boldsymbol{\Omega}_i(t_k), \tag{68}$$

$$|\varepsilon_i(\tau|t_k)| = 0. \tag{69}$$

Then, according to Definition 10, for any $\tau \in [t_{k-1} + T, t_k + T)$, by implementing auxiliary controller $u_i(\tau | t_k) = u_i^{\kappa}(\boldsymbol{x}_i(\tau | t_k))$, (57), (58), and (69) hold; that is, for the assumed control inputs $\hat{u}_i(\tau | t_k)$, $\tau \in [t_k, t_k + T)$, all the constraints in Problem 14 are satisfied; thus, $\hat{u}_i(\tau | t_k)$ is a feasible solution of Problem 14 at t_k , and conclusion (b) holds.

Theorem 16. For the multi-UAV system, each UAV solves the optimization problem (Problem 14) at sampling instant t_k .

Suppose Theorem 15 holds for any sampling instant after t_k . Then, by implementing the real-time guidance commands u_i $(\tau) = u_i^*(\tau|t_k)$, the guidance error, the angle tracking error, and the consistency error for multi-UAV system will converge to zero.

Proof of Theorem 16. For the multi-UAV system, the Lyapunov function is selected as

$$V(t_k) = \sum_{i=1}^{N_M} J_i(t_k, \boldsymbol{x}_i^*, \hat{\boldsymbol{x}}_{i-}^*, \boldsymbol{u}_i^*, \boldsymbol{w}_i).$$
(70)

If we choose $\hat{u}_i(\tau|t_{k+1})$ as a transitional solution at t_{k+1} , then, following (20), there is

$$V(t_{k+1}) - V(t_k) \leq \sum_{i=1}^{N_M} (J_i(t_{k+1}, \widehat{\mathbf{x}}_i, \widehat{\mathbf{x}}_{i-}, \widehat{\mathbf{u}}_i, \mathbf{w}_i) - J_i(t_k, \mathbf{x}_i^*, \widehat{\mathbf{x}}_{i-}, \mathbf{u}_i^*, \mathbf{w}_i))$$

$$= \sum_{i=1}^{N_M} \begin{pmatrix} \sum_{j \in \mathbb{N}_i} \int_{t_{k+1}}^{t_k + T} \lambda_{2i} \left(\left(\mathbf{x}_{i2}^*(\tau | t_k) - \mathbf{x}_{j2}^*(\tau | t_k) \right)^2 - \left(\mathbf{x}_{i2}^*(\tau | t_k) - \widehat{\mathbf{x}}_{j2}(\tau | t_k) \right)^2 \right) d\tau \\ - \int_{t_k}^{t_{k+1}} L_i(\tau | t_k, \mathbf{x}_i^*, \widehat{\mathbf{x}}_{i-}, \mathbf{u}_i^*, \mathbf{w}_i) d\tau + \int_{t_k + T}^{t_{k+1} + T} L_i(\tau | t_k, \mathbf{x}_i^\kappa, \widehat{\mathbf{x}}_{i-}, \mathbf{u}_i^\kappa, \mathbf{w}_i) d\tau \\ + g_i(\mathbf{x}_i^\kappa(t_{k+1} + T | t_k)) - g_i(\mathbf{x}_i^*(t_k + T | t_k)) \end{pmatrix}.$$
(71)

Applying condition (47) with $\mathbf{x}_i(t|t_k)$ being $\mathbf{x}_i^*(t_k + T|t_k)$, then, (72) holds for any $\tau \ge t_k + T$

$$\sum_{i=1}^{N_M} (\dot{g}_i(\boldsymbol{x}_i^*(\tau|t_k)) + L_i(\tau|t_k, \boldsymbol{x}_i^{\kappa}, \hat{\boldsymbol{x}}_{i-}, \boldsymbol{u}_i^{\kappa}, \boldsymbol{w}_i)) < 0.$$
(72)

Integrating (72) from $t_k + T$ to $t_{k+1} + T$ yields

$$\sum_{i=1}^{N_{M}} \left(\int_{t_{k}+T}^{t_{k+1}+T} L_{i}(\tau|t_{k}, \boldsymbol{x}_{i}^{\kappa}, \hat{\boldsymbol{x}}_{i-}, \boldsymbol{u}_{i}^{\kappa}, \boldsymbol{w}_{i}) d\tau + g_{i}(\boldsymbol{x}_{i}^{\kappa}(t_{k+1}+T|t_{k})) - g_{i}(\boldsymbol{x}_{i}^{\ast}(t_{k}+T|t_{k})) \right) < 0.$$
(73)

Define $\varepsilon_i^*(\tau|t_k) = \widehat{x}_{i2}(\tau|t_k) - x_{i2}^*(\tau|t_k)$. Employing compatibility constraint (22), we can obtain inequality (74) and (75) (" $(\tau|t_k)$ " and " (t_k) " are omitted)

$$\begin{split} &\sum_{i=1}^{N_{M}} \sum_{j \in \mathbb{N}_{i}} \int_{t_{k+1}}^{t_{k}+T} \lambda_{2i} \left(\left(x_{i2}^{*} - x_{j2}^{*} \right)^{2} - \left(x_{i2}^{*} - \widehat{x}_{j2} \right)^{2} \right) d\tau \\ &= \sum_{i=1}^{N_{M}} \sum_{j \in \mathbb{N}_{i}} \int_{t_{k+1}}^{t_{k}+T} \lambda_{2i} \left(2x_{i2}^{*} - x_{j2}^{*} - \widehat{x}_{j2} \right) \varepsilon_{j}^{*} d\tau \\ &= \sum_{i=1}^{N_{M}} \sum_{j \in \mathbb{N}_{i}} \int_{t_{k+1}}^{t_{k}+T} \lambda_{2i} \left(2\left(\widehat{x}_{i2} - \widehat{x}_{j2} \right) - 2\varepsilon_{i}^{*} + \varepsilon_{j}^{*} \right) \varepsilon_{j}^{*} d\tau \end{split}$$

$$\leq \sum_{i=1}^{N_{M}} \sum_{j \in \mathbb{N}_{i}} \int_{t_{k+1}}^{t_{k}+T} \lambda_{2i} \left(2\varphi_{ij} + 2|\varepsilon_{i}^{*}| + |\varepsilon_{j}^{*}| \right) |\varepsilon_{j}^{*}| d\tau$$

$$= \sum_{i=1}^{N_{M}} \sum_{j \in \mathbb{N}_{i}} \int_{t_{k+1}}^{t_{k}+T} \lambda_{2i} \left(2\varphi_{ij} |\varepsilon_{j}^{*}| + 2|\varepsilon_{i}^{*}| |\varepsilon_{j}^{*}| + |\varepsilon_{j}^{*}|^{2} \right) d\tau$$

$$\leq \sum_{i=1}^{N_{M}} \sum_{j \in \mathbb{N}_{i}} \int_{t_{k+1}}^{t_{k}+T} \lambda_{2i} \left(2\varphi_{ij} |\varepsilon_{j}^{*}| + |\varepsilon_{i}^{*}|^{2} + 2|\varepsilon_{j}^{*}|^{2} \right) d\tau$$

$$= \sum_{i=1}^{N_{M}} \sum_{j \in \mathbb{N}_{i}} \int_{t_{k+1}}^{t_{k}+T} 2\lambda_{2j} \varphi_{ij} |\varepsilon_{i}^{*}| + \lambda_{2i} |\varepsilon_{i}^{*}|^{2} + 2\lambda_{2j} |\varepsilon_{i}^{*}|^{2} d\tau$$

$$\leq \sum_{i=1}^{N_{M}} \sum_{j \in \mathbb{N}_{i}} \left(T - \delta \right) \left(2\lambda_{2j} \varphi_{ij} v_{i} + (\lambda_{2i} + 2\lambda_{2j}) v_{i}^{2} \right)$$

$$= \sum_{i=1}^{N_{M}} \left(2(T - \delta) \left(\sum_{j \in \mathbb{N}_{i}} \lambda_{2j} \varphi_{ij} \right) v_{i} + (T - \delta) \left(\lambda_{2i} |N_{i}| + 2\sum_{j \in \mathbb{N}_{i}} \lambda_{2j} \right) v_{i}^{2} \right),$$

$$(74)$$

$$\begin{split} \sum_{i=1}^{N_{M}} \int_{t_{k}}^{t_{k+1}} L_{i}(\tau | t_{k}, \boldsymbol{x}_{i}^{*}, \boldsymbol{\hat{x}}_{i-}, \boldsymbol{u}_{i}^{*}, \boldsymbol{w}_{i}) d\tau &\geq \sum_{i=1}^{N_{M}} \int_{t_{k}}^{t_{k+1}} \left(\sum_{j \in \mathbb{N}_{i}} \lambda_{2i} \left(x_{i2}^{*} - \boldsymbol{\hat{x}}_{j2} \right)^{2} \right) d\tau \\ &= \sum_{i=1}^{N_{M}} \int_{t_{k}}^{t_{k+1}} \left(\sum_{j \in \mathbb{N}_{i}} \lambda_{2i} \left(\left(x_{i2}^{*} - \boldsymbol{\hat{x}}_{j2} \right) - \boldsymbol{\varepsilon}_{i}^{*} \right)^{2} \right) d\tau \geq \sum_{i=1}^{N_{M}} \int_{t_{k}}^{t_{k+1}} \left(\sum_{j \in \mathbb{N}_{i}} \left(\lambda_{2i} \zeta_{ij}^{2} - \lambda_{2i} v_{i}^{2} \right) \right) d\tau \\ &= \sum_{i=1}^{N_{M}} \left(\delta \lambda_{2i} \sum_{j \in \mathbb{N}_{i}} \zeta_{ij}^{2} - \delta \lambda_{2i} |N_{i}| v_{i}^{2} \right). \end{split}$$

$$\tag{75}$$

1: for l = 0 to $N_c - 1$ do Calculate $\sigma_{Mi}(t_{k+l}|t_k)$ and $\tilde{u}_i(t_{k+l}|t_k)$ using (3) and (4) Calculate $a_{Mi}^{OAmin}(t_{k+l}|t_k), a_{Mi}^{OAmax}(t_{k+l}|t_k), \tilde{u}_{i1}^{min}(t_{k+l}|t_k), \tilde{u}_{i1}^{max}(t_{k+l}|t_k),$ 2: 3: $\tilde{u}_{i2}^{\min}(t_{k+l}|t_k)$, $\tilde{u}_{i2}^{\max}(t_{k+l}|t_k)$, according to (40)~(44) and ~(87) Correct $\tilde{u}_i(t_{k+l}|t_k)$ applying $\tilde{u}_i(t_{k+l}|t_k) = \operatorname{sat}(\tilde{u}_i|\tilde{u}_i^{\min}(t_{k+l}|t_k), \tilde{u}_i^{\max}(t_{k+l}|t_k))$, where 4:
$$\begin{split} & \tilde{u}_{i}^{\min}(t_{k+l}|t_{k}) = [\tilde{u}_{i1}^{\min}(t_{k+l}|t_{k}), \tilde{u}_{i2}^{\min}(t_{k+l}|t_{k})]^{\mathrm{T}}, \\ & \tilde{u}_{i}^{\max}(t_{k+l}|t_{k}) = [\tilde{u}_{i1}^{\max}(t_{k+l}|t_{k}), \tilde{u}_{i2}^{\max}(t_{k+l}|t_{k})]^{\mathrm{T}} \end{split}$$
5: Recalculate $u_i(t_{k+l}|t_k)$ according to (4) Correct $\boldsymbol{u}_i(t_{k+l}|t_k)$ applying $\boldsymbol{u}_i(t_{k+l}|t_k) = \operatorname{sat}(\boldsymbol{u}_i|\boldsymbol{u}_i^{\min}(t_{k+l}|t_k), \boldsymbol{u}_i^{\max}(t_{k+l}|t_k))$, where 6: $\boldsymbol{u}_{i}^{\min}(t_{k+l}|t_{k}) = [a_{Mi}^{OAmin}(t_{k+l}|t_{k}), u_{Mi}^{min}]^{T}, \ \boldsymbol{u}_{i}^{\max}(t_{k+l}|t_{k}) = [a_{Mi}^{OAmin}(t_{k+l}|t_{k}), u_{Mi}^{max}]^{T}$ Let $P_h^{(iter)}(j, l+1)$, j = 1, 2 be the corrected $\boldsymbol{u}_i(t_{k+l}|t_k)$ 7: 8: Calculate $\mathbf{x}_i(t_{k+l+1}|t_k)$ applying (78). 9: end for Return the corrected $P_{h}^{(iter)}$ and $X_{i}(t_{k})$ 10:



1: // Initialization

2: Set the parameters of the algorithm: ω , η_1 , η_2 , $V^{\max}(j, l)$, population size *Ps*, the maximum number of iterations *K*, and maximum number of invalid iterations *K'*.

3: Randomly generate the initial positions of particles $P_h^{(0)}$, $h = 1, 2, \dots, Ps$, which include the assumed control input sequence $P_h^{(0)} = [\hat{u}_i(t_k|t_k), \hat{u}_i(t_{k+1}|t_k), \dots, \hat{u}_i(t_{k-1} + T|t_k)].$

Randomly generate the initial velocities of particles $V_h^{(0)}$. 4: 5: Initialize the current iteration as iter = 0 and number of invalid iterations as iter' = 0. 6: // Main loop 7: while *iter* $< K \land iter' < K'$ do for h := 1 to Ps do 8: Correct $V_h^{(iter)}$ using (91) 9: Correct $\vec{P}_{h}^{(iter)}$ and obtain $x_{i}(t_{k+l}|t_{k}), l = 1, \dots, N_{c}$ with algorithm 1 10: Calculate $F_h^{(iter)}$ using (90) 11: Determine the individual best position P_{h}^{pb} 12: end for 13: Determine the global best position P^{gb} 14: Record the global best fitness value at current iteration $\text{Rec}^{(iter)} = F^{gb}$ 15: if *iter* > $1 \land \text{Rec}^{(iter)} = \text{Rec}^{(iter-1)}$ then 16: iter' = iter' + 117: else 18: iter' = 019: 20: **for** *h* **:= 1 to** *Ps* **do** Update the position of particle as $P_h^{(iter+1)}$, using (88) 21: Update the velocity of particle as $V_{h}^{(iter+1)}$, using (89) 22: 23: end for 24: Update the current iteration as iter = iter + 125: end while 26: Determine the optimal control input sequence at t_k as $\boldsymbol{u}_{i}^{*}(\tau|t_{k}) = [\boldsymbol{u}_{i}^{*}(t_{k}|t_{k}), \boldsymbol{u}_{i}^{*}(t_{k+1}|t_{k}), \cdots, \boldsymbol{u}_{i}^{*}(t_{k-1}+T|t_{k})] = \boldsymbol{P}^{gb}$ 27: //Results 28: return $\boldsymbol{u}_{i}^{*}(\tau|t_{k})$

Ulteriorly,

$$\begin{split} &\sum_{i=1}^{N_{M}} \left(\left(\sum_{j \in \mathbb{N}_{i}} \int_{t_{k+1}}^{t_{k+1}} \lambda_{2i} \left(\left(x_{i2}^{*} - x_{j2}^{*} \right)^{2} - \left(x_{i2}^{*} - \hat{x}_{j2} \right)^{2} \right) \mathrm{d}\tau \right) \\ &- \int_{t_{k}}^{t_{k+1}} L_{i} (\tau | t_{k}, x_{i}^{*}, \hat{x}_{i-}, u_{i}^{*}, w_{i}) \mathrm{d}\tau \right) \\ &= \sum_{i=1}^{N_{M}} \left(\left(\sum_{j \in \mathbb{N}_{i}} \int_{t_{k+1}}^{t_{k}+T} \lambda_{2i} \left(\left(x_{i2}^{*} - x_{j2}^{*} \right)^{2} - \left(x_{i2}^{*} - \hat{x}_{j2} \right)^{2} \right) \mathrm{d}\tau \right) \\ &- \gamma_{i} \int_{t_{k}}^{t_{k+1}} L_{i} (\tau | t_{k}, x_{i}^{*}, \hat{x}_{i-}, u_{i}^{*}, w_{i}) \mathrm{d}\tau \right) \\ &+ \sum_{i=1}^{N_{M}} \left((\gamma_{i} - 1) \int_{t_{k}}^{t_{k+1}} L_{i} (\tau | t_{k}, x_{i}^{*}, \hat{x}_{i-}, u_{i}^{*}, w_{i}) \mathrm{d}\tau \right) \\ &\leq \sum_{i=1}^{N_{M}} \left(2(T - \delta) \left(\sum_{j \in \mathbb{N}_{i}} \lambda_{2j} \varphi_{ij} \right) v_{i} + (T - \delta) \left(\lambda_{2i} | N_{i} | + 2 \sum_{j \in \mathbb{N}_{i}} \lambda_{2j} \right) \\ &\cdot v_{i}^{2} + \delta \gamma_{i} \lambda_{2i} | N_{i} | v_{i}^{2} - \delta \gamma_{i} \lambda_{2i} \sum_{j \in \mathbb{N}_{i}} \zeta_{ij}^{2} \right) \\ &+ \sum_{i=1}^{N_{M}} \left((\gamma_{i} - 1) \int_{t_{k}}^{t_{k+1}} L_{i} (\tau | t_{k}, x_{i}^{*}, \hat{x}_{i-}, u_{i}^{*}, w_{i}) \mathrm{d}\tau \right) \\ &= \sum_{i=1}^{N_{M}} \left(b_{i} v_{i} + a_{i} v_{i}^{2} + c_{i} \right) \\ &+ \sum_{i=1}^{N_{M}} \left((\gamma_{i} - 1) \int_{t_{k}}^{t_{k+1}} L_{i} (\tau | t_{k}, x_{i}^{*}, \hat{x}_{i-}, u_{i}^{*}, w_{i}) \mathrm{d}\tau \right) \\ &= \sum_{i=1}^{N_{M}} \left((\gamma_{i} - 1) \int_{t_{k}}^{t_{k+1}} L_{i} (\tau | t_{k}, x_{i}^{*}, \hat{x}_{i-}, u_{i}^{*}, w_{i}) \mathrm{d}\tau \right). \end{split}$$

$$(76)$$

Substituting (73) and (76) into (71) yields

$$V(t_{k+1}) - V(t_k) \le \sum_{i=1}^{N_M} \left((\gamma_i - 1) \int_{t_k}^{t_{k+1}} L_i(\tau | t_k, \boldsymbol{x}_i^*, \widehat{\boldsymbol{x}}_{i-}, \boldsymbol{u}_i^*, \boldsymbol{w}_i) d\tau \right) < 0.$$
(77)

Thus, if Theorem 15 holds for any sampling instant after t_k , the multi-UAV system is exponentially stable, and Theorem 16 holds.

Thereupon, we have completed the design and analysis of the cooperative guidance scheme based on DMPC. The next section will mainly focus on the implementation of the proposed cooperative guidance scheme.

4. Implementation of DMPC Cooperative Guidance Scheme

In this section, the particle swarm optimization (PSO) algorithm is introduced to solve the optimization problem (Problem 14), and then, the detailed implementation process of DMPC cooperative guidance scheme is presented.



FIGURE 4: Communication topology among UAVs.

4.1. Solving Procedure Based on Particle Swarm Optimization Algorithm. Unlike traditional control methods with analytical expressions for the control inputs, DMPC is an optimization-based control method which relies on an efficient optimization algorithm to find the optimal control actions at each sampling instant.

However, due to the constraint of nonlinear system dynamic (9), the solving procedure of Problem 14 involves large amounts of integral operations, and therefore, appropriate simplification is needed. To this end, the approximation of the system dynamic is carried out based on explicit Euler's method.

Suppose $N_c = T/\delta$ and $N_c \in \mathbb{Z}^+$, where \mathbb{Z}^+ is the positive integer set. Then, the system within control horizon can be approximated by

$$\mathbf{x}_{i}(t_{k+l+1}|t_{k}) \approx \mathbf{x}_{i}(t_{k+l}|t_{k}) + \delta \dot{\mathbf{x}}_{i}(t_{k+l}|t_{k}), l = 0, 1, \cdots, N_{c} - 1.$$
(78)

Equation (78) implies that the system evolves in the manner of zero-order holding over each sampling interval within the control horizon; thus, the control inputs and the disturbances can be represented in the form of sequences:

$$\boldsymbol{U}_{i}(t_{k}) = [\boldsymbol{u}_{i}(t_{k}|t_{k}), \boldsymbol{u}_{i}(t_{k+1}|t_{k}), \cdots, \boldsymbol{u}_{i}(t_{k-1}+T|t_{k})], \quad (79)$$

$$\boldsymbol{W}_{i}(t_{k}) = [\boldsymbol{w}_{i}(t_{k}|t_{k}), \boldsymbol{w}_{i}(t_{k+1}|t_{k}), \cdots, \boldsymbol{w}_{i}(t_{k-1}+T|t_{k})].$$
(80)

Applying the approximation of the actual system (78), the objective function of Problem 14 is approximately equal to

$$J_i(t_k, \boldsymbol{X}_i, \widehat{\boldsymbol{X}}_{i-}, \boldsymbol{U}_i, \boldsymbol{W}_i) \approx \delta \sum_{l=1}^{N_c} L_i(t_{k+l} | t_k, \boldsymbol{x}_i, \widehat{\boldsymbol{x}}_{i-}, \boldsymbol{u}_i, \boldsymbol{w}_i) + g_i(\boldsymbol{x}_i(t_k + T | t_k)),$$
(81)

where $X_i(t_k) = [x_i(t_{k+1}|t_k), x_i(t_{k+2}|t_k), \cdots, x_i(t_k + T|t_k)].$

All the constraints of Problem 14 can be approximated in a same manner and further transformed into the constraints imposed on $\tilde{u}_i(t_{k+l}|t_k)$. Take the compatibility constraint as an example, which can be transformed into the constraint given by

$$\tilde{u}_{i1}^{\text{CB min}}(t_{k+l}|t_k) \le \tilde{u}_{i1}(t_{k+l}|t_k) \le \tilde{u}_{i1}^{\text{CB max}}(t_{k+l}|t_k), \quad (82)$$

where the upper and lower bounds of $\tilde{u}_{i1}(t_{k+l}|t_k)$, i.e., $\tilde{u}_{i1}^{\text{CB}\max}(t_{k+l}|t_k)$ and $\tilde{u}_{i1}^{\text{CB}\min}(t_{k+l}|t_k)$, can be derived from

| Index | Initial position (m) | Initial velocity (m/s) | Initial flight path angle (°) | Maximum and minimum normal overload (m·s ⁻²) | Maximum and minimum tangential acceleration (m·s ⁻²) |
|-------|----------------------|------------------------|-------------------------------|--|--|
| 1 | (-6000, 0) | 400 | 45 | 200, -200 | 200, -200 |
| 2 | (0, -6000) | 400 | 45 | 200, -200 | 200, -200 |
| 3 | (6000, -6000) | 400 | 67.5 | 200, -200 | 200, -200 |
| 4 | (-6000, 6000) | 400 | 22.5 | 200, -200 | 200, -200 |

TABLE 2: Initial states of UAVs.

Offline stage: For each UAV M_i , give (initial) sampling period δ , control horizon *T*, maximum data size of the observation data *N*, weighting scalars for stage cost: λ_{1i} , λ_{2i} , weighting scalars for terminal cost: ρ_{1i} , ρ_{2i} , μ_i , scalars for terminal set constraint and compatibility constraint: Λ_{1i} , Λ_{2i} , γ_i , and parameters for auxiliary controller: k_{1i} , k_{2i} . Those scalars or parameters are supposed to satisfy assumption 12.

Online stage: At the initial time t_0 , each UAV M_i sets $\hat{u}_i(\tau|t_0)$ as $u_i^{\kappa}(x_i(\tau|t_0))$ and $\hat{x}_{i2}(\tau|t_0)$ as $x_{i2}^{\kappa}(\tau|t_0)$, initializes the observation data for the target as $e(t_0) = \theta_T(t_0)$, sends $\hat{x}_{i2}(\tau|t_0)$ to the neighbors and receives $\hat{x}_{i2}(\tau|t_0)$ from them, $\tau \in [t_0, t_0 + T)$.

- (I) At sampling instant t_k , $k \ge 0$, for each UAV M_i
 - (a) predict $\boldsymbol{w}_i(\tau|t_k), \tau \in [t_k, t_k + T)$ using the method proposed in section 3.1.
 - (b) compute $v_i(t_k)$ according to (23), (24).
 - (c) solve optimization problem 14 using algorithm 2, to obtain $u_i^*(\tau|t_k), \tau \in [t_k, t_k + T)$.
- (II) Over time interval $t \in [t_k, t_{k+1})$, for each UAV M_i
 - (a) implement the real-time guidance command $u_i(t) = u_i^*(t|t_k)$, $t \in [t_k, t_{k+1})$ and sample the system states.
 - (b) update the observation data, by adding $e(t_{k+1}) = \theta_T(t_{k+1})$. If there is $k+1 \ge N$, then delete $e(t_{k+1-N})$.
 - (c) calculate the assumed control inputs $\hat{u}_i(\tau | t_{k+1})$ and the corresponding state trajectories $\hat{x}_{i2}(\tau | t_{k+1}), \tau \in [t_{k+1}, t_{k+1} + T)$, according to (19), (51), (52), and (20).
 - (d) sends $\hat{x}_{i2}(\tau|t_{k+1})$ to M_j and receives $\hat{x}_{j2}(\tau|t_{k+1})$ from M_j , $j \in \mathbb{N}_i$
- (III) Return to (I), update T and δ using (94) and (95). If the updated δ is smaller than the preset one, then stop control and sample system states at $t_k + \delta$.

ALGORITHM 3: DMPC cooperative guidance scheme.

(22) and dynamic equation (9).

$$\tilde{C}_{i}^{\text{CB}}(t_{k+l}|t_{k}) = \left| \hat{x}_{j2}(t_{k+l+1}|t_{k}) - x_{i2}(t_{k+l}|t_{k}) - \delta \dot{x}_{i2}(t_{k+l}|t_{k}) \right| - v_{i}(t_{k}) < 0.$$
(83)

Similarly, the terminal set constraint can be transformed into

$$\tilde{u}_{ij}^{\text{TS min}}(t_{k-1} + T|t_k) \le \tilde{u}_{ij}(t_{k-1} + T|t_k) \le \tilde{u}_{ij}^{\text{TS max}}(t_{k-1} + T|t_k), \quad j = 1, 2,$$
(84)

with $\tilde{u}_{ij}^{\text{TS min}}(t_{k-1} + T|t_k)$ and $\tilde{u}_{ij}^{\text{TS max}}(t_{k-1} + T|t_k)$ being calculated in the same manner as shown in (83) and (84). In summary, for any $l = 0, 1, \dots, N_c - 1$, $\tilde{u}_i(t_{k+l}|t_k)$ is constrained by

$$\tilde{u}_{ij}^{\min}(t_{k+l}|t_k) \le \tilde{u}_{ij}(t_{k+l}|t_k) \le \tilde{u}_{ij}^{\max}(t_{k+l}|t_k), \quad j = 1, 2, \quad (85)$$

where

$$\tilde{u}_{ij}^{\min}(t_{k+l}|t_k) = \begin{cases} \tilde{u}_{ij}^{\text{CBmin}}(t_{k+l}|t_k), & l = 0, \cdots, N_c - 2, \\\\ \max\left\{\tilde{u}_{ij}^{\text{CBmin}}(t_{k+l}|t_k), \tilde{u}_{ij}^{\text{TSmin}}(t_{k+l}|t_k)\right\}, & l = N_c - 1, \end{cases}$$
(86)

$$\tilde{u}_{ij}^{\max}(t_{k+l}|t_k) = \begin{cases} \tilde{u}_{ij}^{\text{CBmax}}(t_{k+l}|t_k), & l = 0, \cdots, N_c - 2, \\\\ \min\left\{\tilde{u}_{ij}^{\text{CBmin}}(t_{k+l}|t_k), \tilde{u}_{ij}^{\text{TSmin}}(t_{k+l}|t_k)\right\}, & l = N_c - 1, \end{cases}$$
(87)

with $\tilde{u}_{i2}^{\text{CBmin}}(t_{k+l}|t_k)$ and $\tilde{u}_{i2}^{\text{CBmax}}(t_{k+l}|t_k)$ being $-\infty$ and $+\infty$, respectively.

Based on the above processing, each UAV solves Problem 14 at sampling instant t_k using particle swarm optimization (PSO) algorithm. PSO algorithm is a heuristic optimization algorithm widely used in UAV decisionmaking problems [40], whose iterative process relies on the searching experience of all the particles in the swarm, with no need for calculating the gradient of each optimization variable.

In PSO, each particle corresponds to a solution of the optimization problem, whose position can be represented as $\boldsymbol{P}_h^{(\mathrm{iter})} = (P_h^{(\mathrm{iter})}(j,l)) \in \mathbb{R}_{2 \times N_c}$ with the same dimension as $\boldsymbol{U}_i(t_k)$. $P_h^{(\mathrm{iter})}(j,l)$ denotes the *j*, *l*-th element of $\boldsymbol{P}_h^{(\mathrm{iter})}$. In the same manner, the velocity of the particle is

In the same manner, the velocity of the particle is denoted as $V_h^{(\text{iter})} \in \mathbb{R}_{2 \times N_c}$, which determines the updating direction and step size of each particle in a single iteration, as shown in

$$\boldsymbol{P}_{h}^{(\text{iter}+1)} = \boldsymbol{P}_{h}^{(\text{iter})} + \boldsymbol{V}_{h}^{(\text{iter})}.$$
(88)







FIGURE 5: Continued.



FIGURE 5: Simulation results for Case 1: (a) flight trajectories; (b) UAV-target distance under FxTCG; (c) UAV-target distance under DMPC-CG; (d) time-to-go under FxTCG; (e) time-to-go under DMPC-CG; (f) LOS angle under FxTCG; (g) LOS angle under DMPC-CG; (h) LOS angle rate under FxTCG; (i) LOS angle rate under DMPC-CG; (j) guidance command a_M under FxTCG; (k) guidance command a_M under DMPC-CG.

TABLE 3: Simulation results for Case 1.

| | | P F C C | | | | |
|-------|-----------------------|------------------------|-----------------------|---------------------|------------------------|-----------------------|
| Index | | FxTCG | | | DMPC-CG | |
| macx | Miss distance (m) | Error of LOS angle (°) | Interception time (s) | Miss distance (m) | Error of LOS angle (°) | Interception time (s) |
| 1 | 7.07×10^{-4} | 1.57×10^{-4} | 37.8654 | 1.17×10^{-6} | 4.94×10^{-5} | 41.3079 |
| 2 | 7.03×10^{-4} | 1.57×10^{-4} | 37.8654 | 1.08×10^{-5} | 7.57×10^{-6} | 41.3079 |
| 3 | 5.32×10^{-5} | 2.62×10^{-4} | 37.8654 | 4.09×10^{-5} | 4.59×10^{-5} | 41.3079 |
| 4 | 6.20×10^{-5} | 2.61×10^{-4} | 37.8654 | 6.44×10^{-6} | 3.16×10^{-5} | 41.3079 |

Commonly, the velocity of the particle is simultaneously affected by its ever-reached best position P_h^{pb} , the global best position P_g^{gb} , and its current velocity, which is formulated as

$$\boldsymbol{V}_{h}^{(\text{iter}+1)} = \boldsymbol{\omega} \cdot \boldsymbol{V}_{h}^{(\text{iter})} + \boldsymbol{\eta}_{1} \cdot \boldsymbol{\xi}_{1} \cdot \left(\boldsymbol{P}_{h}^{\text{pb}} - \boldsymbol{P}_{h}^{(\text{iter})}\right) + \boldsymbol{\eta}_{2} \cdot \boldsymbol{\xi}_{2} \cdot \left(\boldsymbol{P}^{\text{gb}} - \boldsymbol{P}_{h}^{(\text{iter})}\right),$$
(89)

where ω represents the inertia weight, which can be preset or adjusted in iteration; η_1 and η_2 are the acceleration constants given a priori; and ξ_1 and ξ_2 are the random numbers within (0, 1). P_h^{pb} and P^{gb} satisfy $F_h^{\text{pb}} = \min_{\text{iter}} F_h^{(\text{iter})}$ and $F^{\text{gb}} = \min_{\text{iter},h} F_h^{(\text{iter})}$, respectively. $F_h^{(\text{iter})}$ is the fitness function defined as (90), for evaluating the fitness of each particle at its current position.

$$F_{h}^{(\text{iter})} = \delta \sum_{l=1}^{N_{c}} L_{i} \Big(t_{k} + l\delta | t_{k}, \boldsymbol{x}_{i}, \widehat{\boldsymbol{x}}_{i-}, \boldsymbol{P}_{h}^{(\text{iter})}, \boldsymbol{w}_{i} \Big) + g_{i} \big(\boldsymbol{x}_{i} \big(t_{k} + T | t_{k} \big) \big).$$

$$(90)$$

The velocity of each particle is also limited by the maximum velocity, and there is

$$V_{h}^{(\text{iter})}(j,l) = \begin{cases} -V^{\max}(j,l), & V_{h}^{(\text{iter})}(j,l) < -V^{\max}(j,l), \\ V_{h}^{(\text{iter})}(j,l), & -V^{\max}(j,l) < V_{h}^{(\text{iter})}(j,l) < V^{\max}(j,l), \\ V^{\max}(j,l), & V_{h}^{(\text{iter})}(j,l) > V^{\max}(j,l). \end{cases}$$

$$(91)$$

Additionally, in order to guarantee the feasibilities of particles in each iteration, we need to check the positions of the particles to prevent them from violating constraints as given in (57), (84), and (85). The check and correction will be made as follows.

In Algorithm 1, $\tilde{u}_i(t_{k+l}|t_k)$ and $u_i(t_{k+l}|t_k)$ are corrected using the saturation function as follows:

$$\boldsymbol{u} = \operatorname{sat}(\boldsymbol{u} | \boldsymbol{u}^{\min}, \boldsymbol{u}^{\max}) = \begin{bmatrix} \operatorname{sat}(\boldsymbol{u}_1 | \boldsymbol{u}_1^{\min}, \boldsymbol{u}_1^{\max}) \\ \operatorname{sat}(\boldsymbol{u}_2 | \boldsymbol{u}_2^{\min}, \boldsymbol{u}_2^{\max}) \end{bmatrix}, \quad (92)$$

where

$$\operatorname{sat}\left(u_{j} \middle| u_{j}^{\min}, u_{j}^{\max}\right) = \begin{cases} u_{j}^{\min}, & u_{j} < u_{j}^{\min}, \\ u_{j}, & u_{j}^{\min} \le u_{j} \le u_{j}^{\max}, \\ u_{j}^{\max}, & u_{j} > u_{j}^{\max}, \\ j = 1, 2. \end{cases}$$
(93)



FIGURE 6: Continued.



FIGURE 6: Continued.



FIGURE 6: Simulation results for Case 2: (a) flight trajectories; (b) UAV-target distance under FxTCG; (c) UAV-target distance under DMPC-CG; (d) time-to-go under FxTCG; (e) time-to-go under DMPC-CG; (f) LOS angle under FxTCG; (g) LOS angle under DMPC-CG; (h) LOS angle rate under FxTCG; (i) LOS angle rate under DMPC-CG; (j) guidance command a_M under FxTCG; (k) guidance command a_M under DMPC-CG; (l) guidance command u_M under FxTCG; (m) guidance command u_M under DMPC-CG.

| | | 1 | TABLE 4. Simulation TC | suits for Case 2. | | |
|-------|-----------------------|------------------------|------------------------|-----------------------|------------------------|-----------------------|
| Index | FxTCG | | | DMPC-CG | | |
| | Miss distance (m) | Error of LOS angle (°) | Interception time (s) | Miss distance (m) | Error of LOS angle (°) | Interception time (s) |
| 1 | 0.0210 | 0.127 | 26.5123 | 3.97×10^{-3} | 9.49×10^{-3} | 27.7539 |
| 2 | 2.20×10^{-3} | 0.0352 | 26.5122 | 4.12×10^{-4} | 6.27×10^{-3} | 27.7539 |
| 3 | 3.85×10^{-3} | 0.0517 | 26.5123 | 1.76×10^{-3} | 5.64×10^{-3} | 27.7539 |
| 4 | 5.82×10^{-3} | 0.0542 | 26.5124 | 4.23×10^{-3} | 0.0251 | 27.7539 |

TABLE 4: Simulation results for Case 2

Thereupon, the complete pseudocode of the PSO algorithm for Problem 14 is given as follows.

Using Algorithm 2, we can obtain the optimal control inputs and the real-time guidance command of M_i at sampling instant t_k . Based on which, the complete implementation process of the DMPC cooperative guidance scheme will be presented in the next subsection.

4.2. Implementation of DMPC Cooperative Guidance Scheme. In order to improve the terminal guidance effect of each UAV, we adopt two strategies as follows.

(a) Variable control horizon

As the final time approaches, the control horizon of each UAV will be adjusted as

$$T = \begin{cases} T & T \ge x_{i2}(t_k), \\ \delta \cdot \left[\frac{x_{i2}(t_k)}{\delta} \right] & T < x_{i2}(t_k). \end{cases}$$
(94)

(b) Variable sampling period

If the time-to-go is smaller than the current sampling period, then, in order to obtain the actual guidance effect

TABLE 5: Obstacles' positions and radiuses.

| Index of case | Index of obstacle | Center position (m) | Radius (m) |
|---------------|-------------------|---------------------|------------|
| | 1 | (0, 2000) | 2000 |
| 3 | 2 | (6000, 1000) | 1500 |
| | 3 | (-2000, 7000) | 1500 |
| | 1 | (1000, 0) | 1400 |
| | 2 | (5000, 1000) | 1000 |
| | 3 | (7500, -1000) | 1500 |
| 4 | 4 | (-1000, 2000) | 1000 |
| | 5 | (0, 7000) | 800 |
| | 6 | (4000, 5000) | 1200 |
| | 7 | (11000, 1000) | 900 |

at final time, we let sampling period be

$$\delta = \begin{cases} \delta & \delta \ge x_{i2}(t_k), \\ x_{i2}(t_k) & \delta < x_{i2}(t_k). \end{cases}$$
(95)

The DMPC cooperative guidance scheme is implemented in two stages, as follows.

Now the implementation of the DMPC cooperative guidance scheme is completed.



FIGURE 7: Continued.



FIGURE 7: Continued.



FIGURE 7: Simulation results for Case 1: (a) flight trajectories; (b) time-to-go under MPC-DG; (c) time-to-go under DMPC-CG; (d) minimum UAV-obstacle distance under MPC-DG; (e) minimum UAV-obstacle distance under DMPC-CG; (f) guidance command a_M under MPC-DG; (g) guidance command a_M under DMPC-CG; (h) guidance command u_M under MPC-DG; (i) guidance command u_M under DMPC-CG; (j) LOS angle under DMPC-CG; (k) LOS angle rate under DMPC-CG; (l) average computation duration under DMPC-CG.

TABLE 6: Simulation results for Case 1.

| Index | MPC-DG Miss distance (m) Error of LOS angle (°) Interception time (c) | | | DMPC-CG Miss distance (m) Error of LOS angle (°) Intercention time (s) | | |
|-------|--|---|---------|---|-----------------------|---------|
| | | | | | | |
| 1 | 2.2121 | — | 35.5212 | 1.25×10^{-5} | 0.0216 | 35.5241 |
| 2 | 2.4485 | — | 35.5210 | 3.50×10^{-3} | 8.49×10^{-3} | 35.5241 |
| 3 | 2.1738 | — | 35.5199 | 4.19×10^{-4} | 4.13×10^{-3} | 35.5241 |
| 4 | 2.3300 | _ | 35.5215 | 6.71×10^{-4} | 0.0166 | 35.5242 |

5. Numerical Simulation

In order to demonstrate the effectiveness of the DMPC cooperative guidance scheme, the numerical simulations are conducted in this section. The simulation scenario is set as four UAVs with fixed communication topology attacking a single maneuvering target, and in different cases, the target adopts diverse maneuvering strategies. The simulations will be carried out in the cases with obstacles and cases without obstacles, and the methods proposed in recent studies [16, 24] will be used for comparison.

In all of simulations, the communication topology among the UAVs is shown in Figure 4, the initial positions, velocities, and the admissible inputs of the UAVs are given in Table 2, and the initial position and flight path angle of the target are set to $x_T(0) = 15000$ m, $y_T(0) = 15000$ m, and $\theta_T(0) = -135^\circ$; all the scalars and parameters of Algorithm 2 and Algorithm 3 are given as follows.

For Algorithm 2, Ps = 5, K = 20, K' = 5, $\eta_1 = \eta_2 = 0.1$, $\omega_{\text{max}} = 0.8$, $\omega_{\text{min}} = 0.2$, and $\omega = \omega_{\text{max}} - iter^2(\omega_{\text{max}} - \omega_{\text{min}})/(K^2)$.

For Algorithm 3, (initial) $\delta = 0.1$ s, (initial) T = 0.3 s, N = 8, $\lambda_{1i} = 1$, $\lambda_{2i} = 1$, $\rho_{1i} = 10$, $\rho_{2i} = \lambda_{2i} |N_i| + \sum_{j \in \mathbb{N}_i} \lambda_{2j}$, $\mu_i = 2$, $\Lambda_{1i} = 25$, $\Lambda_{2i} = 25$, $\gamma_i = 0.1$, $k_{1i} = 0.36$, and $k_{2i} = 2$.

5.1. Scenario I: Without Obstacles. In this subsection, the comparison simulations are carried out in cases (Cases 1 and 2) without no-fly zones or obstacles. To validate the advantages of the DMPC cooperative guidance scheme (DMPC-CG), it will be compared with the fixed-time cooperative guidance law (FxTCG) proposed in [16]. The velocities and the maneuvering inputs of the target in Case 1 and Case 2 are given as follows.

Case 1. $v_T = 200 \text{ m/s}$, and $a_T(t) = 0 \text{ m/s}^2$.

Case 2. $v_T = 500$ m/s, and $a_T(t) = 5t \cos(0.4t)$ m/s².

Case 1 simulates the scenario of intercepting a lowspeed, nonmaneuvering target, while Case 2 simulates the interception scenario against a high-speed target that adopts snake maneuvering tactics.

In both cases, the desired LOS angles of the UAVs are set to $q_{d1} = 22.5^{\circ}$, $q_{d2} = 67.5^{\circ}$, $q_{d3} = 90^{\circ}$, and $q_{d4} = 0^{\circ}$, respectively.

Simulation results under Case 1 and Case 2 are shown in Figure 5 and Table 3 and Figure 6 and Table 4, respectively. From Figures 5(a)-5(c) and 6(a)-6(c), it can be seen that using either method, all the UAVs hit the target with smooth flight trajectories. However, significant differences between



FIGURE 8: Continued.



FIGURE 8: Continued.

FIGURE 8: Simulation results for Case 2: (a) flight trajectories; (b) time-to-go under MPC-DG; (c) time-to-go under DMPC-CG; (d) minimum UAV-obstacle distance under MPC-DG; (f) guidance command a_M under MPC-DG; (g) guidance command a_M under MPC-CG; (h) guidance command u_M under DMPC-CG; (j) LOS angle under DMPC-CG; (k) LOS angle rate under DMPC-CG; (l) average computation duration under DMPC-CG.

TABLE 7: Simulation results for Case 2.

| Index | Miss distance (m) | MPC-DG Error of LOS angle (°) | Interception time (s) | Miss distance (m) | DMPC-CG Error of LOS angle (°) | Interception time (s) |
|-------|-------------------|----------------------------------|-----------------------|---------------------|-----------------------------------|-----------------------|
| 1 | 2.6746 | | 35.5210 | 0.0146 | 0.0117 | 35.5243 |
| 2 | 2.3191 | _ | 35.5213 | 2.91×10^{-3} | 0.0316 | 35.5242 |
| 3 | 2.6041 | _ | 35.5219 | 8.86×10^{-3} | 0.0182 | 35.5242 |
| 4 | 1.9049 | — | 35.5184 | 7.34×10^{-4} | 0.0185 | 35.5245 |

the two methods can be observed from Figures 5(d)-5(m)and 6(d)-6(m). Figures 5(d)-5(g) and 6(d)-6(g) show that, with either method, the time-to-go of all the UAVs can reach a consensus within a finite time, and the LOS angles ultimately converge to their desired values. However, under DMPC-CG, the time spent before reaching the consensus is about 2s, which is shorter than that under FxTCG. In Figures 5(j)-5(m) and 6(j)-6(m), the guidance commands exhibit rapid changes in the initial phase of guidance under DMPC-CG, but the duration of acceleration saturation is largely shortened. In contrast, under FxTCG, the saturation of acceleration lasts longer, though the curves of guidance commands are smoother. Besides, according to Tables 3 and 4, the DMPC-CG achieves smaller miss distances and convergence errors in LOS angles than the FxTCG in both cases. Therefore, the proposed DMPC cooperative guidance scheme can achieve cooperative interception with advantageous performance than the fixed-time cooperative guidance law in cases without obstacles.

5.2. Scenario II: With Obstacles. In order to further demonstrate the effectiveness of the proposed DMPC cooperative guidance scheme in the presence of obstacles, simulations are conducted in this subsection (Cases 1 and 2). The proposed DMPC-CG will be compared with another MPCbased distributed guidance method (MPC-DG) [24]. As MPC-DG is designed to achieve time-constrained guidance in consideration of obstacle avoidance, the comparison will mainly focus on the performance of guidance, coordination of impact times, and obstacle avoidance. The control of LOS angles will be analyzed as an additional property of DMPC-CG.

In Cases 1 and 2, the obstacles are randomly dispersed, as shown in Table 5. The desired LOS angles of UAVs are set to $q_{d1} = 45^{\circ}$, $q_{d2} = 67.5^{\circ}$, $q_{d3} = 90^{\circ}$, and $q_{d4} = 22.5^{\circ}$, and the velocities and maneuvering inputs of the target are given as follows.

Case 1. $v_T = 300 \text{ m/s}$

$$a_T(t) = \begin{cases} 10 \text{ m/s}^2 & t < 10 \text{ s} \\ -20 \text{ m/s}^2 & t \ge 10 \text{ s} \end{cases}$$
(96)

Case 2. $v_T = 300 \text{ m/s}$, and $a_T(t) = 100 \cos(0.5t) \text{ m/s}^2$.

For ease of comparison, the objective function designed in ref. [24] is given as follows:

$$J_{i}'(t_{k}, \boldsymbol{x}_{i}, \widehat{\boldsymbol{x}}_{i-}, \boldsymbol{u}_{i}, \boldsymbol{w}_{i}) = \int_{t_{k}}^{t_{k}+T} \lambda_{1i}' \left(\sum_{j \in \mathbb{N}_{i}} \left(x_{i2}(\tau | t_{k}) - \widehat{x}_{j2}(\tau | t_{k}) \right)^{2} \right) \\ + \lambda_{2i}' \left(a_{\mathrm{Mi}}^{2}(\tau | t_{k}) + u_{\mathrm{Mi}}^{2}(\tau | t_{k}) \right) \mathrm{d}\tau \\ + \rho_{1i}' r_{i}^{2}(t_{k} + T | t_{k}).$$
(97)



In (97), λ'_{1i} and λ'_{2i} are the weighting scalars of the stage cost, while ρ'_{1i} is the weighting scalar of the terminal cost. The sampling period, prediction horizon, and other parameters are set to be the same as those in ref. [24].

Simulation results for Case 1 and Case 2 are shown in Figure 7 and Table 6 and Figure 8 and Table 7, respectively. From Figures 7(a), 7(d), and 7(e) and 8(a), 8(d), and 8(e), it can be observed that using these two methods, all the UAVs successfully avoid collisions with obstacles and fly towards the target with smooth trajectories. From Figures 7(b) and 7(c) and 8(b) and 8(c), it can be seen that once a UAV takes a maneuver to avoid obstacles, the consensus on time-to-go is briefly broken, but under either method, the consensus will soon be restored and maintained for most of the time during guidance. Although both methods basically meet the requirements of the time-constrained cooperative guidance with obstacle avoidance based on the above analysis, there are still differences in guidance performance between the two methods. According to Tables 6 and 7, the DMPC-CG achieves smaller miss distance and consistency errors in impact time for both cases, which largely depends on the rational design of the objective function and the related constraints.

In addition, the curves of LOS angles and LOS angle rates are separately depicted for DMPC-CG in Figures 7(j) and 7(k) and 8(j) and 8(k). It can be seen that the LOS angles and angular rates deviate significantly from their desired values at the beginning and exhibit fluctuations during the guidance process. However, by adopting DMPC-CG, all of them eventually converge to the desired values. Figures 7(l) and 8(l) show the average computation duration under DMPC-CG. At each sampling instant, all the UAVs are able to complete calculations within 0.032 s, with most calculations lasting less than 0.01 s. These results show that the proposed DMPC cooperative guidance scheme is capable of achieving time and angle-constrained cooperative guidance in the presence of obstacles or no-fly zones.

6. Conclusion

This study is concerned with the salvo attack problem for controllable thrust UAVs against a maneuvering target. The guidance scheme is presented based on distributed model predictive control to achieve simultaneous interception at the desired impact angle and no-fly zone avoidance. The guidance scheme proposed in this study is advanced as follows:

- By introducing assumed predictive trajectories, all the UAVs are able to synchronously figure out their acceleration commands utilizing the shared information among the neighbors
- (2) By designing the prediction method and associated constraints, both obstacle avoidance and maneuvering target interception are achieved
- (3) By employing compatibility constraints, terminal set constraints, and auxiliary controller, the convergence of the multi-UAV system is guaranteed

(4) Using PSO algorithm, the computation efficiency of the optimal acceleration commands is improved

The effectiveness of the method is verified through the numerical simulation. In the next step of our work, the DMPC will be applied to three-dimensional cooperative interception problem with the constraints of terminal angle and obstacle avoidance, and the disturbance observer-based robust DMPC scheme for cooperative guidance against maneuvering target will also be studied.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the Shandong Natural Science (ZR2020MF090).

References

- X. Gao, L. Wang, X. Yu et al., "Conditional probability based multi-objective cooperative task assignment for heterogeneous UAVs," *Engineering Applications of Artificial Intelligence*, vol. 123, article 106404, 2023.
- [2] Y. Chen, J. Wang, J. Shan, and M. Xin, "Cooperative guidance for multiple powered missiles with constrained impact and bounded speed," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 4, pp. 825–841, 2021.
- [3] S. R. Kumar and D. Mukherjee, "Terminal time-constrained nonlinear interception strategies against maneuvering targets," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 1, pp. 200–209, 2021.
- [4] T. Long, Y. Cao, J. Sun, and G. Xu, "Adaptive event-triggered distributed optimal guidance design via adaptive dynamic programming," *Chinese Journal of Aeronautics*, vol. 35, no. 7, pp. 113–127, 2022.
- [5] S. He and D. Lin, "Adaptive nonsingular sliding mode based guidance law with terminal angular constraint," *International Journal of Aeronautical and Space Sciences*, vol. 15, no. 2, pp. 146–152, 2014.
- [6] B. G. Park, H. H. Kwon, Y. H. Kim, and T. H. Kim, "Composite guidance scheme for impact angle control against a nonmaneuvering moving target," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 5, pp. 1132–1139, 2016.
- [7] J. Zhao, S. Shi, Y. Chong, M. Feng, and H. You, "Distributed cooperative guidance law for multiple missiles with impact angle constraint," *Journal of Chinese Inertial Technology*, vol. 26, no. 4, p. 8, 2018.
- [8] I.-S. Jeon, J.-I. Lee, and M.-J. Tahk, "Impact-time-control guidance law for anti-ship missiles," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 2, pp. 260–266, 2006.
- [9] Z. Shiyu and Z. Rui, "Cooperative guidance for multimissile salvo attack," *Chinese Journal of Aeronautics*, vol. 21, no. 6, pp. 533–539, 2008.

- [10] I. S. Jeon, J. I. Lee, and M. J. Tahk, "Homing guidance law for cooperative attack of multiple missiles," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 1, pp. 275–280, 2010.
- [11] J. Zhao, R. Zhou, and Z. Dong, "Three-dimensional cooperative guidance laws against stationary and maneuvering targets," *Chinese Journal of Aeronautics*, vol. 28, no. 4, pp. 1104–1120, 2015.
- [12] A. Sinha, S. R. Kumar, and D. Mukherjee, "Cooperative integrated guidance and control design for simultaneous interception," *Aerospace Science and Technology*, vol. 120, p. 107262, 2022.
- [13] K. Li, J. Wang, C. H. Lee, R. Zhou, and S. Zhao, "Distributed cooperative guidance for multivehicle simultaneous arrival without numerical singularities," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 7, pp. 1365–1373, 2020.
- [14] J. Song, S. Song, and S. Xu, "Three-dimensional cooperative guidance law for multiple missiles with finite-time convergence," *Aerospace Science and Technology*, vol. 67, pp. 193– 205, 2017.
- [15] S. Liu, B. Yan, T. Zhang, P. Dai, R. Liu, and J. Yan, "Threedimensional cooperative guidance law for intercepting hypersonic targets," *Aerospace Science and Technology*, vol. 129, p. 107815, 2022.
- [16] W. Dong, C. Wang, J. Wang, Z. Zuo, and J. Shan, "Fixed-time terminal angle-constrained cooperative guidance law against maneuvering target," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 2, pp. 1352–1366, 2022.
- [17] J. Zhao and R. Zhou, "Unified approach to cooperative guidance laws against stationary and maneuvering targets," *Nonlinear Dynamics*, vol. 81, no. 4, pp. 1635–1647, 2015.
- [18] S. Zhang, Y. Guo, Z. Liu, S. Wang, and X. Hu, "Finite-time cooperative guidance strategy for impact angle and time control," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 2, pp. 806–819, 2021.
- [19] H. Yu, K. Dai, H. Li et al., "Three-dimensional adaptive fixedtime cooperative guidance law with impact time and angle constraints," *Aerospace Science and Technology*, vol. 123, p. 107450, 2022.
- [20] S. Liu, B. Yan, R. Liu, P. Dai, J. Yan, and G. Xin, "Cooperative guidance law for intercepting a hypersonic target with impact angle constraint," *The Aeronautical Journal*, vol. 126, no. 1300, pp. 1026–1044, 2022.
- [21] Y. Zhang, G. Ma, and A. Liu, "Guidance law with impact time and impact angle constraints," *Chinese Journal of Aeronautics*, vol. 26, no. 4, pp. 960–966, 2013.
- [22] X. Wang, B. Li, X. Su et al., "Autonomous dispatch trajectory planning on flight deck: a search-resampling-optimization framework," *Engineering Applications of Artificial Intelligence*, vol. 119, p. 105792, 2023.
- [23] X. Li, L. Wang, H. Wang, L. Tao, and X. Wang, "A warmstarted trajectory planner for fixed-wing unmanned aerial vehicle formation," *Applied Mathematical Modelling*, vol. 122, pp. 200–219, 2023.
- [24] J. Zhao, S. Zhou, and R. Zhou, "Distributed time-constrained guidance using nonlinear model predictive control," *Nonlinear Dynamics*, vol. 84, no. 3, pp. 1399–1416, 2016.
- [25] Z. Jiang, X. Yang, Y. Zhang, C. Wang, and H. Yu, "Threedimensional guidance approach with obstacle avoidance against maneuvering targets," *International Journal of Aerospace Engineering*, vol. 2023, Article ID 8872166, 25 pages, 2023.

- [26] H. Jiang, Z. An, S. Chen, and F. Xiong, "Cooperative guidance with multiple constraints using convex optimization," *Aero-space Science and Technology*, vol. 79, pp. 426–440, 2018.
- [27] J. Zhao and R. Zhou, "Distributed three-dimensional cooperative guidance via receding horizon control," *Chinese Journal of Aeronautics*, vol. 29, no. 4, pp. 972–983, 2016.
- [28] S. Kang, J. Wang, G. Li, J. Shan, and I. R. Petersen, "Optimal cooperative guidance law for salvo attack: an MPC-based consensus perspective," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2397–2410, 2018.
- [29] M. Cong, X. Cheng, Z. Zhao, and Z. Li, "Studies on multiconstraints cooperative guidance method based on distributed MPC for multi-missiles," *Applied Sciences*, vol. 11, no. 22, p. 10857, 2021.
- [30] P. Wang and B. Ding, "Distributed RHC for tracking and formation of nonholonomic multi-vehicle systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1439–1453, 2014.
- [31] L. Dai, Q. Cao, Y. Xia, and Y. Gao, "Distributed MPC for formation of multi-agent systems with collision avoidance and obstacle avoidance," *Journal of the Franklin Institute*, vol. 354, no. 4, pp. 2068–2085, 2017.
- [32] M. Qin, Z. Wang, H. Li, and X. Liu, "DMPC for formation of multi-UAV with collision/obstacle avoidance," *Journal of Beijing University of Aeronautics and Astronautics*, 2022.
- [33] S. Vargas, H. M. Becerra, and J. B. Hayet, "MPC-based distributed formation control of multiple quadcopters with obstacle avoidance and connectivity maintenance," *Control Engineering Practice*, vol. 121, article 105054, 2022.
- [34] H. Kim, H. Shim, and J. H. Seo, "Output consensus of heterogeneous uncertain linear multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 200–206, 2011.
- [35] E. Camponogara and L. B. De Oliveira, "Distributed optimization for model predictive control of linear-dynamic networks," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 39, no. 6, pp. 1331–1338, 2009.
- [36] W. H. Chen, D. J. Ballance, and J. O'Reilly, "Optimisation of attraction domains of nonlinear MPC via LMI methods," in *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, vol. 4, pp. 3067–3072, Arlington, VA, USA, June 2001.
- [37] S. Y. Yu, H. Chen, P. Zhang, and X. J. Li, "An LMI optimization approach for enlarging the terminal region of MPC for nonlinear systems," *Acta Automatica Sinica*, vol. 34, no. 7, pp. 798–804, 2008.
- [38] X. Liu, Y. Shi, and D. Constantinescu, "Robust distributed model predictive control of constrained continuous-time nonlinear systems using two-layer invariant sets," *Journal of Dynamic Systems, Measurement, and Control*, vol. 138, no. 6, article 061004, 2016.
- [39] H. Li and Y. Shi, "Robust distributed model predictive control of constrained continuous-time nonlinear systems: a robustness constraint approach," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1673–1678, 2014.
- [40] X. Wang, Y. Wang, X. Su et al., "Deep reinforcement learningbased air combat maneuver decision-making: literature review, implementation tutorial and future direction," *Artificial Intelligence Review*, vol. 57, no. 1, p. 1, 2024.