Unmanned Systems, Vol. 14, No. 4 (2026) 973–993
© World Scientific Publishing Company
DOI: 10.1142/S2301385026500536

Unmanned Systems
https://www.worldscientific.com/worldscinet/us


# Predefined-Time Null-Space Behavioral Control for Multi-UAVs Rigid Formation Control

Jie Huang [*,†,‡,‖], Junjie Li [*,‡], Zhengyi Zhang [*,§], Tingting Zhao [*,‡], Simcha Vos [¶]

[*]*College of Electrical Engineering and Automation, Fuzhou University, Fuzhou 350108, P. R. China*

[†]*The National Key Lab of Autonomous Intelligent Unmanned Systems, Beijing Institute of Technology, Beijing 100081, P. R. China*

[‡]*5G+Industrial Internet Institute, Fuzhou University, Fuzhou 350108, P. R. China*

[§]*School of Automation and Electrical Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, P. R. China*

[¶]*Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft 2628CD, NL*

This paper proposes a predefined-time null-space behavioral control algorithm based on a time-base generator (PTNSBC-TBG) for multi-quadrotor unmanned aerial vehicle (UAV) systems, enabling rigid formation control with collision and obstacle avoidance within a predefined time. First, the UAV kinematic model is incorporated into the task design of null-space behavioral control (NSBC) to ensure that task instructions adhere strictly to the UAV's motion constraints, overcoming the limitations of the classical NSBC method, which relies on a single integrator model. Second, a predefined-time null-space behavioral control algorithm is designed to drive UAVs to form a rigid formation within a specified time in advance. The issue of excessive initial control input is addressed by utilizing a time-base generator (TBG). Additionally, a nonlinear adaptive sliding mode control law based on the TBG is designed to ensure the predefined-time convergence of the UAVs' velocity command tracking error. Finally Numerical simulations and experiment indicate that the proposed PTNSBC-TBG can form a rigid formation within a predefined time. Compared to the algorithm without TBG, the initial control input is significantly reduced, by at least 60%.

*Keywords*: Predefined time; multi-UAVs; null-space behavior control; time-base generator.

## 1. Introduction

In the past few years, development of unmanned systems has underlined the important role of UAVs as an efficient, convenient, and safe type of aircraft [1, 2]. UAV cluster systems are characterized by redundancy, robustness, and scalability, as well as more excellent synergy, intelligence, and autonomy. Because of the fact that multi-UAVs can complete tasks that are difficult for a single UAV to accomplish, they have been widely applied in various fields. Therefore, the study of multi-UAV system has recently become one of the focus points in the field of unmanned systems.

Formation technology is essential to multi-UAV systems and is widely used in the fields of cargo handling, collaborative surveying, etc. When performing certain special tasks in a formation, such as handling of large objects [3]; multi-UAVs signal coverage [4, 5]; multi-spacecraft stabilized formation [6,7], all UAVs need to form and maintain a strict rigid

Received 27 October 2024; Revised 26 May 2025; Accepted 27 May 2025; Published 15 July 2025. This paper was recommended for publication in its revised form by editorial board member, Jinwen Hu.
Email Addresses: [‖]jie.huang@fzu.edu.cn
[‖]Corresponding author.

formation. And the execution of these tasks firstly requires each UAV to reach the desired task execution location or form the desired stable formation within a time. Currently, researchers have studied the construction of rigid formation and stabilization control of the formation [3, 8], however, they have not considered the obstacle avoidance and collision avoidance requirements of UAVs in the process of movement, which means their methods are difficult to apply in actual systems.

Because the task execution environment is variable, it is an important problem to avoid obstacles and eliminate task conflicts in formation process. Flores-Resendiz *et al.* [9] proposed a control strategy consisting of bounded attraction and complementary repulsion components to solve the collision-free formation control problem in multi-UAV systems; Lin *et al.* [10] proposed a distributed formation control scheme based on Lie algebra and designed internal and external loop control strategies for multi-UAVs to eliminate control deviation caused by interference; Guo *et al.* [11] proposed a distributed controller with obstacle avoidance and formation reconstruction capabilities based on consistency theory and local neighborhood information. In the research of agent multi-task planning in dynamic environments, Bai *et al.* [12, 13] made a series of progress in addressing multi-objective access problems under drift field constraints. In detail, the paper [12] innovatively integrates navigation strategies and allocation algorithms to establish a time optimal allocation model for heterogeneous tasks in a planar lateral drift field. For complex scenes with obstacles, paper [13] further proposes a dynamic cost evaluation framework driven by path planning, which utilizes a real-time updated obstacle avoidance cost matrix to achieve multi-objective spatiotemporal collaborative optimization. The environment adaptive task allocation methodology formed by these two works provides key technical support for intelligent agent collaboration under specific physical constraints. In addition, in the more specialized field of vehicle multi-task collaboration, Xu *et al.* [14] and Chen *et al.* [15] proposed independent technical paths: the former effectively alleviates multi vehicle path conflicts through layered architecture design, while the latter develops a behavioral heuristic mechanism for congestion perception. Together, they have promoted the application and development of multi-objective optimization methods in vehicle scenarios.

Currently, many obstacle avoidance algorithms have been widely used in multi-UAVs formation problems, but these methods all suffer from task conflicts when executing moving tasks at the same time. The multi task allocation method is more complex and mostly an offline planning method in the process of multi-UAVs formation control, while the NSBC method is an online control framework that solves task conflicts through null space projection. It ensures higher flexibility and versatility for the UAVS in performing primary tasks. For example, Zheng *et al.* [16] used an NSBC architecture to realize time-varying formation control of UAVs while avoiding collision with obstacles; Golluccio *et al.* [17] proposed an improved learning algorithm to extend the NSBC method, which enables the agents to dynamically select appropriate low-priority tasks for execution; Wang *et al.* [18] proposed a hierarchical subtask fusion strategy based on the NSBC approach for conflict-free fusion scheduling of multiple subtasks; Zhang *et al.* [19] proposed an adaptive dynamic planning approach based on the NSBC framework to coordinate control of unmanned ground vehicle-robotic arms. Currently, algorithms based on NSBC frameworks are still facing limitations of single integrator models. This paper aims to extend the applicability of current methods to more complex scenarios.

Additionally, the rigid formation algorithm based on the NSBC framework is also constrained by global information limitations. In past research, Zhou *et al.* [20] designed control laws for rigid formations using global relative position information and based on the Null-Space-based Behavioral Control (NSBC) approach. However, in logistic application scenarios, such as robots and UAVs indoor formation control, it is difficult to obtain global relative position information of all UAVs. Therefore, Liu *et al.* [21] considered weakening the relative position information and propose a multi-UAVs formation control algorithm based on partial orientation rigidity, which satisfies the desired relative orientation constraints among the agents in the rigid formation; Xiao *et al.* [22] proposed a rigid formation algorithm based on orientation information, which enables the system to achieve infinitesimal orientation rigidity; Chen *et al.* [23] utilized the relative position information of the initial three agents and based on the relative angle information to ensure that the formation of the remaining agents adheres to the rigidity requirements. However, even though the aforementioned methods reduce the need for relative position information, it is difficult to calculate their convergence time, making them challenging to apply in actual UAV systems.

In formation task and tracking task, the convergence speed of the system is the priority requirement for task objectives, as it is a crucial performance indicator determining the efficiency of formation and tracking tasks. By improving the convergence speed of the formation, the UAVs are able to rapidly form the desired formation within a predefined time, and follow the desired trajectory to move stably, thus improving the completion efficiency of the task. In order to accelerate formation convergence, Niu *et al.* [24] proposed a finite-time formation control strategy to achieve the desired formation within a finite time. However, the convergence time of finite-time formation control strategies is dependent on the initial state of each UAV mostly. To overcome this limitation, Polyakov *et al.* [25] introduced a fixed-time control strategy, ensuring that the convergence strategy is bounded for all initial system values. Although the fixed-time control strategy offers certain advantages over finite-time control, the actual convergence time is difficult to calculate. Hence, Sónchez-Torres *et al.* [26] proposed a predefined-time control strategy, which allows the upper bound of the system's stabilization time to be predefined offline, independent of the system's initial state or other control gains. Li *et al.* [27] designed a distributed formation controller that uses sliding mode control, which enables all UAVs to quickly achieve the desired relative position to the leader within a predefined time; Chen *et al.* [28] proposed a formation control algorithm based on a predefined-time observer using a new non-singular sliding surface to ensure the formation can be formed within a predefined time; Wu *et al.* [29] constructed an adaptive tracking controller triggered by predefined-time events, the output tracking error can converge close to the origin in the desired time. Nevertheless, because predefined-time control relies on fractional powers, the control inputs are too large to be applied in actual systems. To address this issue, Ning *et al.* [30] developed a new integrator based multi-agent scheduled time consistency control framework using the time-base generator (TBG), which alleviates the problem of excessive control input.

At present, many scholars have studied the pre-defined time convergence of multi-agent systems, but most of them only consider the single task such as formation control, and do not restrict the initial input. Hence it is difficult to combine with the actual system. Ensuring the predefined time convergence of smaller control inputs in multi-task scenarios is an important problem and challenge that needs to be solved urgently.

This paper investigates the conflict resolution problem for rigid formation and obstacle avoidance tasks in a multi-UAV system within a predefined time. Based in [30] and [31] this paper proposes the PNSBC-TBG algorithm for multi-UAV system by combining TBG with the NSBC framework. Compared to predefined-time control methods with fractional powers [27], the algorithm this paper proposes requires smaller initial inputs and possesses conflict resolution capabilities due to the NSBC framework. The primary contributions of this paper are outlined as follows:

(1) A new framework is proposed that integrates the UAV model into the NSBC method, breaking through the limitation that the NSBC framework is only applicable to a single integrator model by incorporating kinematic constraints into the task.
(2) A new predefined-time NSBC framework is constructed via TBG. Predefined-time convergence of two tasks is achieved under the NSBC framework. The proposed algorithm extends the application of predefined-time to multi-task scenarios.
(3) The specified time formation control law is designed to address the issue of excessive initial control input in fractional power predefined-time control, reducing the initial input by at least 60%.

The structure of this paper is as follows. In Sec. 2, the UAV model and basic concepts used in this paper are defined. Section 3 describes the proposed predefined-time behavior-based rigid formation scheme. In Sec. 4, the predefined-time controllers based on TBG designed in this paper are described. In Sec. 5, the effectiveness of the PTNSBC-TBG algorithm for UAVs designed in this paper is verified through simulation under unknown interference conditions. Section 6 gives method verification in experiment. The conclusions can be found in Sec. 7.

## 2. Preliminaries

### 2.1. *Mathematical model*

Consider a set of $N(N > 2)$ multi-UAVs whose kinematic model is considered to be controlled by the position loop as well as the attitude loop, respectively, where the position loop kinematic model is described in the ground coordinate system as [32]:

$$\dot{p}_i = Rv_i, \tag{1}$$

where $p_i \in \mathbb{R}^3$ and $v_i \in \mathbb{R}^3$ denotes the position and the velocity of each UAV. $R$ is the rotation matrix from the airframe coordinate system to the ground coordinate system.

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}, \tag{2}$$

where

$$\begin{aligned}
R_{11} &= \cos\theta_i \cos\varphi_i, \\
R_{12} &= \cos\varphi_i \sin\theta_i \sin\phi_i - \sin\varphi_i \cos\phi_i, \\
R_{13} &= \cos\varphi_i \sin\theta_i \cos\phi_i + \sin\varphi_i \sin\phi_i, \\
R_{21} &= \cos\theta_i \sin\varphi_i, \\
R_{22} &= \sin\varphi_i \sin\theta_i \sin\phi_i + \cos\varphi_i \cos\phi_i, \\
R_{23} &= \sin\varphi_i \sin\theta_i \cos\phi_i + \cos\varphi_i \sin\phi_i, \\
R_{31} &= -\sin\theta_i, \\
R_{32} &= \sin\phi_i \cos\theta_i, \\
R_{33} &= \cos\phi_i \cos\theta_i.
\end{aligned}$$

The kinematic model of the attitude loop based on the transformation of the rate of change of the attitude Euler angle

$\beta_i = [\phi_i, \theta_i, \varphi_i]^T$ in the ground coordinate system to the attitude angle which denotes the pitch angle, roll angle and the yaw angle. $\omega_i$ in the airframe coordinate system is given by

$$\dot{\beta}_i = W\omega_i, \tag{3}$$

where $W$ is

$$W = \begin{bmatrix} 1 & \sin\phi_i \tan\theta_i & \cos\phi_i \tan\theta_i \\ 0 & \cos\phi_i & -\sin\phi_i \\ 0 & \sin\phi_i/\cos\theta_i & \cos\phi_i/\cos\theta_i \end{bmatrix}. \tag{4}$$

Define the position and attitude angles of the $i$th UAV as $p_i = [x_i, y_i, z_i]^T \in \mathbb{R}^3$ and $\beta_i = [\phi_i, \theta_i, \varphi_i]^T \in \mathbb{R}^3$, respectively. Therefore, the kinematics of the $i$th UAV can be modeled as the following equation [33]:

$$\dot{\mathbf{X}}_i = \begin{bmatrix} \dot{p}_i \\ \dot{\beta}_i \end{bmatrix} = \Theta_i \mathbf{V}_i, \tag{5}$$

where $\mathbf{X}_i = [p_i, \beta_i]^T$,$\mathbf{V}_i = [v_i, \omega_i]^T$ denote the generalized position and velocity of the $i$th UAV, respectively, $\Theta_i$ denotes the kinematic constraint matrix, which has the specific mathematical form of

$$\Theta_i = \begin{bmatrix} R & 0 \\ 0 & W \end{bmatrix}. \tag{6}$$

## 2.2. *NSBC framework*

In this paper, an improved null-space based behavioral control method is adopted by designing different behaviors of multi-UAVs as different task functions and prioritizing the tasks based on practical experience. The classical null-space behavioral control algorithm projects the low-priority tasks onto the null-space vector of the high-priority tasks. Consequently, complete execution of high-priority tasks is achieved, while still executing the non-conflicting parts of the low-priority tasks. Through this, conflicts between tasks are resolved and stability of the formation is ensured [34, 35]. Based on practical experience, this paper sets collision avoidance and obstacle avoidance as the highest priority tasks for multi-UAVs, and all other tasks such as formation forming and formation switching are treated as sub-priority tasks.

Based on the above task prioritization design, the basic tasks of the multi-UAVs are further designed [36]. Assume that the generic task function is $\boldsymbol{\sigma} \in \mathbb{R}^m$ where $m$ denotes the dimension of the generic task, then the task function is

$$\boldsymbol{\sigma} = f(p), \tag{7}$$

where $p = [p_1, p_2, \ldots, p_n]^T$ represents a collection of multi-UAVs positions, $n$ denotes the number of multi-UAVs and $f(\cdot)$ denotes the mapping of multi-UAVs positions to the task function. The partial derivation with respect to position $p$ for Eq. (7) can be obtained:

$$\dot{\boldsymbol{\sigma}} = \frac{\partial f(p)}{\partial p}\dot{p} = \mathcal{J}(p)\dot{p}, \tag{8}$$

where $\mathcal{J}(p) \in \mathbb{R}^{m\times n}$ is a Jacobi matrix representing the mapping of the task function to the multi-UAVs position $p$.

During the execution of the task, there exists a corresponding desired position $p_d(t)$ for the desired task function $\boldsymbol{\sigma}_d(t)$. The desired trajectory can be obtained from the time integral of the desired velocity, and the least squares solution for the velocity output of the task function is obtained by performing a pseudo-inverse operation on the Jacobi matrix as follows:

$$v = \mathcal{J}^\dagger \dot{\sigma} = \mathcal{J}^T \left(\mathcal{J}\mathcal{J}^T\right)^{-1} \dot{\sigma}, \tag{9}$$

where $v$ is the velocity vector output for the task, $\mathcal{J}^\dagger = \mathcal{J}^T(\mathcal{J}\mathcal{J}^T)^{-1}$ is the pseudo-inverse of the Jacobian matrix.

In addition to get the reference velocity, a reference trajectory is also needed. From the above part, it can be seen that the time integration of the reference velocity gives the reference trajectory, but the discrete time integration process of the velocity produces a reference trajectory deviation. It can be offset by a closed-loop inverse kinematics method of bias, then the output of the reference velocity for the $l$th task is

$$v_l = \mathcal{J}_l^\dagger(\dot{\sigma}_{d.l} + \Lambda_l \tilde{\sigma}_l), \tag{10}$$

where $v_l$ is the speed output for the $l$th task, $\mathcal{J}_l^\dagger$ is the pseudo-inverse of the Jacobi matrix $\mathcal{J}_l$, $\sigma_{d.l}$ is the desired task function, $\dot{\sigma}_{d.l}$ is the derivative of the desired task function, $\Lambda_l$ is a positive definite gain matrix and $\tilde{\sigma}_l = \dot{\sigma}_{d.l} - \sigma_l$ is the task error.

Assuming that the global task consists of multiple basic tasks, the low-priority tasks are projected onto the null-space of the high-priority tasks, thus eliminating the conflicting parts between the task speed output instructions, and finally obtaining the speed output of the global task as

$$v_G = v_l + \sum_{l=2}^{M}((I - \mathcal{J}_l^\dagger \mathcal{J}_l)v_{l+1}), \tag{11}$$

where $v_G$ is the speed output for the global task, $I - \mathcal{J}_l^\dagger \mathcal{J}_l$ is the null-space matrix of the $l$th task, $M$ is the max task level.

## 2.3. *Predefined-time control based on time-base generator*

Consider the following nonlinear system:

$$\dot{\boldsymbol{x}}(t) = q(t, \boldsymbol{x}; \delta), \quad t \in [t_0, +\infty), \tag{12}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ denotes the state of the system, $\delta \in \mathbb{R}^m$ denotes the parameters of the system, which is satisfies that

$\dot{\delta} = 0$. When $\boldsymbol{x} = 0$, the system satisfies that $q(t, 0; \delta) = 0$. $t_0$ represents the initial time and $\boldsymbol{x}(t_0)$ is the initial state of the system.

**Definition 1 (26).** Suppose that $t_{fi}$ is a positive constant and can be arbitrarily preset. The origin $x(t) = 0$ of the nonlinear dynamic system (12) is said to be globally predefined-time-stable if and only if

$$\begin{cases} \lim\limits_{t \to t_{fi}-} \boldsymbol{x}(t, \boldsymbol{x}_0) = 0, & t \in [t_0, t_0 + t_{fi}), \\ \boldsymbol{x}(t, \boldsymbol{x}_0) \equiv 0, & t \in [t_0 + t_{fi}, +\infty). \end{cases} \tag{13}$$

If it is holds for any initial state $\boldsymbol{x}_0$ of the system (12), then $t_{fi}$ is called the predefined-time.

Consider the $q(\cdot) \in \mathbf{R}$ is the function based on time-base generator and it satisfied the following:

$$q(t, \boldsymbol{x}; \delta) = -\Xi_i(t) x(t), \tag{14}$$

$$\Xi_i(t) = \frac{\dot{h}_i(t)}{1 - h_i(t) + \varrho_i}, \tag{15}$$

where $\Xi_i(t)$ is the time-base generator gain function (TBGGF), $0 \le \varrho_i \le 1$, $h_i(t)$ is the time-base generator function (TBGF) and it has the following equalities:

(1) $h_i(t)$ is a function which is second-order differentiable in $t \in (0, +\infty)$.
(2) TBGF $h_i(t)$ is a continuously increasing function from $t = 0$ to predefined time constant $t = t_{fi}$, and $h_i(0) = 0, h_i(t_{fi}) = 1$.
(3) The right derivative of $h_i(t)$ at $t = 0$ is $\dot{h}_i(0) = 0$ and $\dot{h}_i(t_{fi}) = 0$ too.
(4) $h(t) = 1, \dot{h}_i(t) = 0$ when $t > t_{fi}$.

**Lemma 1 (37).** *Consider Eq.* (14) *and the TBGGF* $\Xi_i(t)$ *is defined following Eq.* (15), *then system* (14) *will converge to* $\frac{\varrho_i}{1+\varrho_i} x(0)$.

**Definition 2 (30).** Consider a network system consisting of $N$ agent, the $i$th model is shown as

$$\dot{x}_i(t) = u_i(t), \tag{16}$$

where $x_i(t)$ represents the states and $u_i(t)$ represents the input control. if the system satisfied the following equation in any initial state$x_i(t_0)$:

$$\begin{cases} \lim\limits_{t \to T_0} |x_i(t) - x_j(t)| \le c, \\ |x_i(t) - x_j(t)| \le c, \quad \forall\, t > T_0, \\ \lim\limits_{t \to +\infty} |x_i(t) - x_j(t)| = 0. \end{cases} \tag{17}$$

Then the system can achieve predefined-time consistency, where $T_0$ is the predefined time, $c$ is a positive constant.

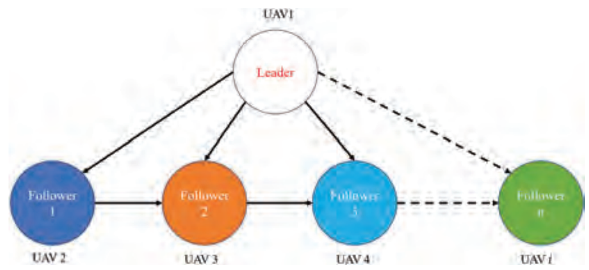



Fig. 1. Topological schematic of $i$UAV.

### 2.4. *Problem statements*

Task goal: The UAVs formation system can track the navigator to form a rigid formation according to the desired distance and relative angle, and can maintain a high formation rigidity in the process of moving the formation as a whole to the target point, with the ability to avoid collision with the intelligent bodies inside the system and the pre-unknown obstacles outside the system, provided that the system satisfies the constraints of its own state and inputs. The following assumptions are made to satisfy the planning objectives:

**Assumption 1.** The UAVs' communication topology satisfies the topology shown in Fig. 1, and there are pathways between any intelligence and its parent and navigator nodes.

**Assumption 2.** The UAVs can obtain real-time position information of obstacles that enter its detection range, and can receive information sent by leader UAV and other UAVs which within its communication range in real time.

**Assumption 3.** Only the effect of unknown external disturbances on the dynamics of the UAVs is considered.

Considering the limited computational resources of the intelligent body and the fact that not all obstacles pose a threat to the intelligent body, and that there is a risk of collision only when the distance between the obstacle and the intelligent body is close, two control tasks, tracking and collision avoidance, are designed based on the ability of the intelligent body to detect obstacles in real time, and according to the risk of collision or not.

## 3. NSBC Rigid Formation Planning Designed with Kinematic Constraints

In this section, a rigid formation planning scheme with kinematic constraints based on an improved null-space behavioral control algorithm is proposed. The algorithm employs a leader-follower strategy and divides the planning scheme design into a formation session and an obstacle avoidance session, thus accomplishing the task of rigid formation of multi-UAVs. The algorithm flowchart of this paper is shown as Fig. 2.

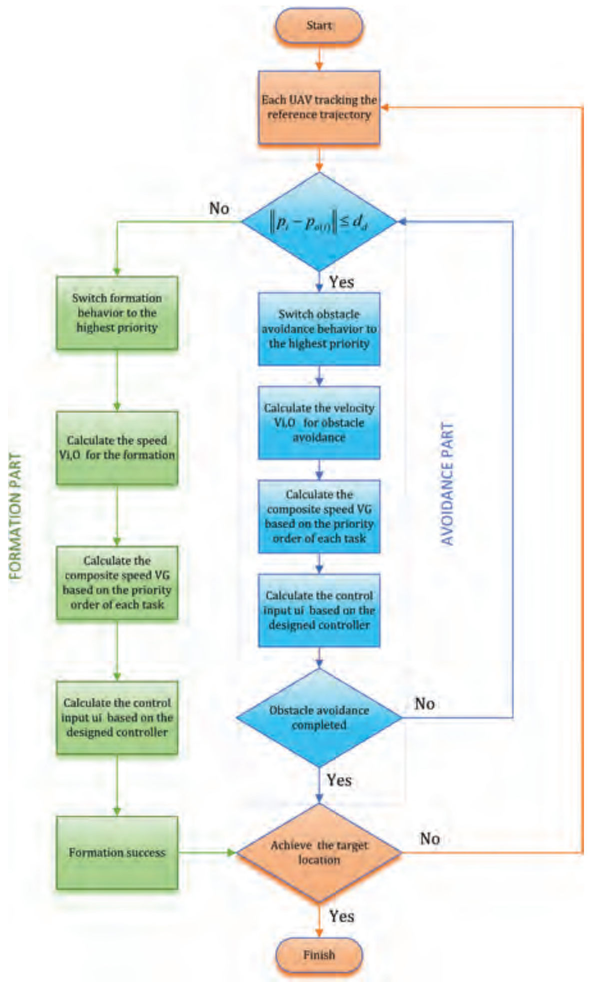



Fig. 2. PTNSBC-TBG algorithm flowchart.

### 3.1. *Basic task design*

It is assumed that each UAV has $M$ basic behaviors, where the $l$th basic behavior of the $i$th UAV can be mathematically modeled using a task variable $\sigma_{i,l} \in \mathbb{R}^{m_i}, l = 1, \dots, M, m_l \leq 3$ as

$$\sigma_{i,l} = g_{i,l}(\mathbf{X}_i), \tag{18}$$

where $g_{i,l}(\mathbf{X}_i)$ is a mathematical functional expression for performing the task.

Since $g_{i,l}(\mathbf{X}_i)$ is limited by the model kinematic constraint $\Theta_i$, the differential form of its task variable $\sigma_{i,l}$ is shown by Eq. (19):

$$\dot{\sigma}_{i,l} = \frac{\partial g_{i,l}(\mathbf{X}_i)}{\partial \mathbf{X}_i}\dot{\mathbf{X}}_i. \tag{19}$$

This can be obtained by associating Eq. (6) with Eq. (19):

$$\dot{\sigma}_{i,l} = \mathcal{J}_{i,l}\dot{\mathbf{X}}_i = \mathcal{J}_{i,l}\begin{bmatrix}\dot{p}_i \\ \dot{\beta}_i\end{bmatrix} = \mathcal{J}_{i,l}\Theta_i\mathbf{V}_i, \tag{20}$$

where $\mathcal{J}_{i,l}$ denotes the Jacobi matrix of the task.

Then, the least squares solution for the speed output of the task function is obtained as follows:

$$v_{i,l} = \Theta_i^{-1}\mathcal{J}_{i,l}^{\dagger}\dot{\sigma}_{i,l} = \Theta_i^{-1}\mathcal{J}_{i,l}^{T}(\mathcal{J}_{i,l}\mathcal{J}_{i,l}^{T})^{-1}\dot{\sigma}_{i,l}, \tag{21}$$

where $v_{i,l}$ is the velocity vector output command for task $\sigma_{i,l}$, $\mathcal{J}_{i,l}^{\dagger} = \mathcal{J}_{i,l}^{T}(\mathcal{J}_{i,l}\mathcal{J}_{i,l}^{T})^{-1}$ is the pseudo-inverse of the Jacobi matrix, $\Theta_i^{-1}$ is the inverse of the kinematic constraint matrix of the UAV model.

After using the closed-loop inverse kinematics method for deviation cancellation, the reference speed output for the $l$th task is

$$v_{i,l} = \Theta_i^{-1}\mathcal{J}_{i,l}^{\dagger}(\dot{\sigma}_{d,i,l} + \Lambda_{i,l}\tilde{\sigma}_{i,l}), \tag{22}$$

where $v_{i,l}$ is the velocity vector output command for task $\sigma_{i,l}$, $\mathcal{J}_{i,l}^{\dagger}$ is the pseudo-inverse of the Jacobi matrix, $\dot{\sigma}_{d,i,l}$ is the derivative of the expected task function, $\Lambda_{i,l}$ is a positive definite gain matrix, $\tilde{\sigma}_{i,l} = \sigma_{d,i,l} - \sigma_{i,l}$ is the task error, $\sigma_{d,i,l}$ is the expected task function.

### 3.2. *Design of rigid formation task*

The multi-UAVs rigid formation task requires multi-UAVs to move to a desired position without changing the overall formation, while maintaining the relative rigidity of the overall formation. Assuming the position information of each agent is $p = \{p_1, p_2, \dots, p_n\}$, the rigid formation task function defined in this section is

$$\sigma_{i,F} = [d_{ij}, \theta_{\measuredangle jik}, \varphi_{\measuredangle jik}]^T, \quad i \in 1, 2, \dots, n, \tag{23}$$

where $d_ij = \|p_i - p_j\|_2$ represents the distance between $p_i$ and $p_j$, $\theta_{\measuredangle jik}$ represents the angle between $p_j - p_i$ and $p_k - p_i$ projected in the XOY plane, $\varphi_{\measuredangle jik}$ represents the angle between $p_j - p_i$ and $p_k - p_i$ projected in the XOZ plane, the detail of $d_{ij}, \theta_{\measuredangle jik}, \varphi_{\measuredangle jik}$ is shown in Fig. 3:

If given the non-overlapping position vectors $q_i$, $q_j$ and $q_k$ denote the projections of $p_i$, $p_j$ and $p_k$ in the XOY plane, then the relative angle $\theta_{\measuredangle jik}$ can be calculated uniquely by the following equation:

$$\theta_{\measuredangle jik} = \begin{cases}\arccos(z_{ij}^{\mathrm{T}}z_{ik}) & z_{ij}\cdot z_{ik} \geq 0, \\ 2\pi - \arccos(z_{ij}^{\mathrm{T}}z_{ik}) & \text{otherwise},\end{cases} \tag{24}$$

where the $z_{ij} = \frac{q_j - q_i}{\|q_j - q_i\|}$, $z_{ik} = \frac{q_k - q_i}{\|q_k - q_i\|}$ and $z_{ij} = Q_0 z_{ij} = \begin{bmatrix}0 & -1 \\ 1 & 0\end{bmatrix} z_{ij}$ is the vector obtained by rotating the vector $z_{ij}$ by $\frac{\pi}{2}$ counterclockwise.

Define the task error as $\tilde{\sigma}_{i,F} = |\sigma_{i,F} - \sigma_{d,i,F}|_{3\times 1}$, where $\sigma_{d,i,F}$ denotes the desired position of the UAVs, i.e., the desired distance versus the desired relative angle.

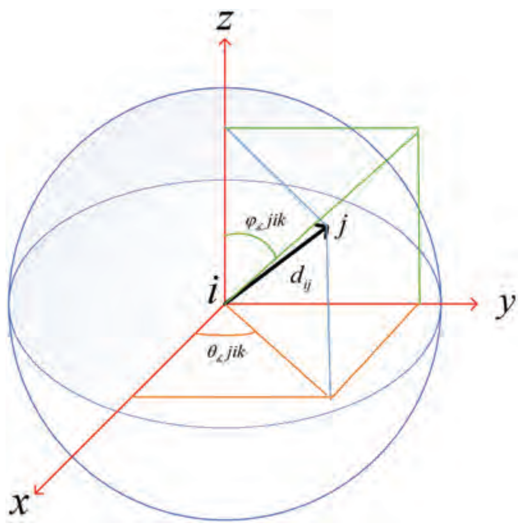



Fig. 3. Relationship Schematic of $d_{ij}, \theta_{\measuredangle jik}, \varphi_{\measuredangle jik}$.

Then the behavior-related Jacobian matrix is as follows:

$$\mathcal{J}_{g_G}(p) = [A, \pm B, \pm C], \tag{25}$$

where the specific form of each term in the Jacobi matrix $\mathcal{J}_{gG}(p)$ equation is shown as

$$A_{ij} = -\frac{\operatorname{sgn}(q_i - q_j)\|q_i - q_j\|_2}{\sqrt{\|(q_i - q_j)^2\|_2}}, \tag{26}$$

$$\begin{aligned} B_{jik} = \frac{1}{D_{jik}} \Bigg( & \frac{x_j - 2x_i + x_k}{(\|q_j - q_i\|_2)(\|q_k - q_i\|_2)} \\ & + E_{jik} \Bigg( \frac{|x_i - x_j| \operatorname{sign}(x_i - x_j)}{(\|q_j - q_i\|_2)^3(\|q_k - q_i\|_2)} \\ & + \frac{|x_i - x_k| \operatorname{sign}(x_i - x_k)}{(\|q_j - q_i\|_2)(\|q_k - q_i\|_2)^3} \Bigg) \Bigg), \end{aligned} \tag{27}$$

where

$$D_{jik} = \sqrt{1 - \frac{((x_i - x_j)(x_i - x_k) + (y_i - y_j)(y_i - y_k))^2}{(\|q_j - q_i\|_2)^2(\|q_k - q_i\|_2)^2}}, \tag{28}$$

$$E_{jik} = ((x_i - x_j)(x_i - x_k) + (y_i - y_j)(y_i - y_k)). \tag{29}$$

$C_{jik}$ is similar to $B_{jik}$. This Jacobi matrix implements the rigidity requirement of distributed formation by restricting the distance and relative angle between two intelligences, thus constraining the relative position of each intelligence to its neighbors.

After obtaining the Jacobi matrix of the formation, the output speed of the formation task function can finally be obtained as shown in Eq. (30):

$$v_{i,F} = \Theta_i^{-1} \mathcal{J}_{i,F}^{\dagger} \Lambda_{i,F} (\Xi_1(t) + 1)\tilde{\sigma}_{i,F} - \dot{\hat{X}}_i, \tag{30}$$

where the $\Xi_1(t)$ is the TBG function and shown as the following equation:

$$\Xi_1(t) = \frac{\dot{h}_1(t)}{2\lambda_{\min}(H)(1 - h_1(t) + \varrho_1)}, \tag{31}$$

where the $h_1(t)$ is as follows:

$$h_1(t) = \begin{cases} \frac{20}{tf^7}t^7 - \frac{70}{tf^6}t^6 + \frac{84}{tf^5}t^5 - \frac{35}{tf^4}t^4, & 0 \le t \le tf, \\ 1, & t \ge tf, \end{cases} \tag{32}$$

after adding the TBG function, the UAVs' mission error can converge within a predefined time.

### 3.3. *Design of obstacle avoidance task*

Assuming that obstacle $p_{o(i)}$ is a spherical obstacle, the set of obstacles $O_r$ is defined as follows:

$$O_r = (p_i \in \mathbb{R}^3, p_{o(i)} \in \mathbb{R}^3 : \|p_i - p_{o(i)}\| \le d_d), \tag{33}$$

where $d_d$ is safe distance.

The obstacle avoidance task function defined in this section is

$$\sigma_{i,O} = (\max\{\|p_i - p_{o(i)}\|, d_d\}). \tag{34}$$

Then the Jacobi matrix for the obstacle avoidance task is

$$\mathcal{J}_{i,O} = \frac{(p_i - p_{o(i)})}{\|p_i - p_{o(i)}\|}. \tag{35}$$

The final output obtained is an obstacle avoidance speed which is

$$v_{i,O} = \Theta_i^{-1} \mathcal{J}_{i,O}^{\dagger} \lambda_{i,O} (\Xi_1(t) + 1)\tilde{\sigma}_{i,O}, \tag{36}$$

where $\tilde{\sigma}_{i,O} = (d_d - \|p_i - p_{o(i)}\|)$ is the obstacle avoidance task error, $\lambda_{i,O}$ is the set positive definite gain matrix, $\mathcal{J}_{i,O}^{\dagger}$ is the pseudo-inverse of the obstacle avoidance Jacobi matrix.

### 3.4. *Multi-UAVs composite task design*

A composite behavior is a combination of multiple basic behaviors projected in null space in a certain priority order, defined $\hat{l} \in N$ to denote the behavioral priority order, $N = \{1, \dots, n\}$. The basic behaviors satisfy the following rules:

(1) Basic behaviors with $\hat{l}_1$ priority cannot interfere with basic behaviors with $\hat{l}_2$ priority, which $\hat{l}_1 > \hat{l}_2, \hat{l}_1, \hat{l}_2 \in N$.
(2) The mapping relationship from system speed to task speed is represented by the task Jacobi matrix $\mathcal{J}_{i,l}$.

(3) The dimension $m_M$ of the basic behavior with the lowest priority is likely to be greater than $m_{\text{total}} - \sum_{l=1}^{M-1} m_l$, so the expected dimension $m_{\text{total}}$ is greater than the total dimension of all basic behaviors.

Based on the designed formation and obstacle avoidance task, the final combined velocity solution is $v_{i,G}$, the combined velocity is when the multi-UAVs is not approaching an obstacle and is not at a safe distance which is

$$v_{i,G} = \Theta_i^{-1}(v_{i,F} + (I - \mathcal{J}_{i,F}^{\dagger}\mathcal{J}_{i,F})v_{i,O}). \tag{37}$$

The combined velocity is when the multi-UAVs is approaching an obstacle and is at a safe distance:

$$v_{i,G} = \Theta_i^{-1}(v_{i,O} + (I - \mathcal{J}_{i,O}^{\dagger}\mathcal{J}_{i,O})v_{i,F}). \tag{38}$$

## 4. Predefined-Time Formation Controller

In this section, a predefined-time-convergent sliding mode controller is designed to track the desired velocity obtained from Eqs. (37) and (38). Since the quad-rotor UAV is a typical nonlinear system, in order to achieve a better control effect, the double closed-loop control of the quad-rotor system is established. The system position control loop is set to control the position information $p_i$ of the quad-rotor UAV to realize effective tracking of the desired position trajectory of the quad-rotor UAV, and the system attitude control loop is set to control the attitude information $\beta_i$ of the quad-rotor UAV to realize stable tracking of the desired attitude trajectory of the quad-rotor UAV.

Consider the position loop system.

$$\begin{cases} \dot{\mathbf{X}}_{i,1} = \Theta_i \mathbf{X}_{i,2}, \\ \dot{\mathbf{X}}_{i,2} = -\boldsymbol{\Omega}^{-1}\boldsymbol{C}\dot{\mathbf{X}}_{i,1} + \boldsymbol{\Omega}^{-1}\boldsymbol{\tau}_i + \boldsymbol{\tau}_d, \end{cases} \tag{39}$$

where $\mathbf{X}_{i,1} = \begin{bmatrix} p_i \\ \beta_i \end{bmatrix} \in \mathbb{R}^6$ denotes the position and the velocity of the agent$i$, $\boldsymbol{\tau}_i \in \mathbb{R}^6$ denotes the control input and $\boldsymbol{\tau}_d$ is the bounded disturbances,the matrix $\boldsymbol{C}$ is the Coriolis term. Matrix $\boldsymbol{\Omega} \in \mathbb{R}^{6\times 6}$ is the parameter matrix which is given by

$$\begin{aligned} \boldsymbol{\Omega} &= \begin{bmatrix} 0 & \Omega_{21} & \Omega_{31} & 0 & 0 & 0 \\ \Omega_{11} & 0 & \Omega_{11} & 0 & 0 & 0 \\ \Omega_{11} & \Omega_{11} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Omega_{44} & \Omega_{45} & \Omega_{46} \\ 0 & 0 & 0 & \Omega_{54} & \Omega_{55} & \Omega_{56} \\ 0 & 0 & 0 & \Omega_{64} & \Omega_{65} & \Omega_{66} \end{bmatrix} \\ &= \begin{bmatrix} S(w) & O_{3\times 3} \\ O_{3\times 3} & -Q^{-1}S(w)Q \end{bmatrix}, \end{aligned} \tag{40}$$

where

$$S(w) = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \tag{41}$$

$$Q = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}, \tag{42}$$

where $\omega_i$ denotes the euler angle change rate, $I_x x, I_y y, I_z z$ represent the moments of inertia along the three coordinate axes, respectively.

In order to obtain the position of the navigator under disturbance, a predetermined time estimator was designed to estimate the position of the navigator. First, the variables $\tilde{X}_i$ and $\hat{X}_i$ are defined as

$$\begin{aligned} \tilde{X}_i &= \hat{X}_i - X_1 = [\hat{p}_i,\ \hat{\beta}_i]^T - [p_1,\ \beta_1]^T, \\ \tilde{X}_1 &= 0,\ \ i = 2, 3, \ldots, N, \end{aligned} \tag{43}$$

where $X_1$ denotes the state of the leader, the predetermined time estimator for the $i$th UAV is designed as follows:

$$\dot{\hat{X}}_i(t) = -\Xi_2(t)g_i(t) - k_1 g_i(t) - k_2 SGN(g_i(t)), \tag{44}$$

where the time-base generator gain function $\Xi_2(t)$ and $g_i$ is shown as the following equations:

$$\Xi_2(t) = \frac{\dot{h}_2(t)}{2\lambda_{\min}(H)(1 - h_2(t) + \varrho_2)}, \tag{45}$$

$$g_i(t) = \sum_{j=1}^{N} a_{ij}(\tilde{X}_i(t) - \tilde{X}_j(t)), \tag{46}$$

where $\lambda_{\min}(H)$ denotes the minimum eigenvalue of topologist matrix $H$, $\dot{\hat{X}}_i$, $0 \le \varrho_2 \le 1$ represents the designed constant, $h_2(t)$ represents the TBG function which satisfied that $h_2(t_{f2}) = 1$ and $h_2(t) = 1$ when $t > t_{f2}$, $k_1, k_2 > 0$ and $k_2$ satisfied that $k_2 \ge \sup_{t\ge 0} \|\dot{X}_1(t)\|_\infty$, the SGN$(\cdot)$ denotes the saturation sign function which is as follows:

$$\text{SGN}(x) = \frac{(|x + \varsigma| - |x - \varsigma|)}{2\varsigma}. \tag{47}$$

**Lemma 2 (37).** *Based on Lemma* 1, *the estimator* (44) *can estimate the leader's actual states for the UAVs in the predefined time $t_{f2}$, and it is satisfied by Eq.* (48):

$$\begin{cases} \lim\limits_{t\to t_{f_2}} |\hat{X}_i - X_1| \le N\sqrt{\dfrac{2\varrho_2}{1+\varrho_2}V_X(0)}, \\ |\hat{X}_i - X_1| \le N\sqrt{\dfrac{2\varrho_2}{1+\varrho_2}V_X(0)}, \quad \forall\, t > t_{f_2}, \\ \lim\limits_{t\to t_\infty} |\hat{X}_i - X_1| = 0, \end{cases} \tag{48}$$

*where $t_{f2}$ is the predefined time constant.*

Then the predefined-time controller of $i$th UAV is designed to make sure the velocity of each UAV can track the expected velocity within a predefined time. The controller is designed as follows:

$$\tau_i = \boldsymbol{\Omega}(-(\Xi_3 + 1)\tilde{\mathbf{V}}_i + \Upsilon_i(\mathbf{X}_i, \dot{\mathbf{X}}_i, \mathbf{V}_{r,i}, \dot{\mathbf{V}}_{r,i})\hat{\Psi}_i - \hat{\delta}_i SGN(\tilde{\mathbf{V}}_i)), \tag{49}$$

$$\dot{\hat{\Psi}}_i = -\Gamma_i^{-1}\Upsilon_i(\mathbf{X}_i, \dot{\mathbf{X}}_i, \mathbf{V}_{r,i}, \dot{\mathbf{V}}_{r,i})\tilde{\mathbf{V}}_i, \tag{50}$$

$$\dot{\hat{\delta}}_i = \Upsilon_i^{-1}\|\tilde{\mathbf{V}}_i\|_1, \tag{51}$$

$$\Xi_3(t) = \frac{\dot{h}_3(t)}{2\lambda_{\min}(H)(1 - h_3(t) + \varrho_3)}, \tag{52}$$

where, $\Xi_i$ denotes the TBG gain function similar to Eq. (45), and $h_3$ has the same equality with $h_2$ but the predefined time is $t_{f3}$. $\Gamma_i$ is a positive constant, $\tilde{\mathbf{V}}_i = \mathbf{V}_i - \mathbf{V}_{r,i}$ represents the tracking error, $\tilde{\Upsilon}_i$ and $\tilde{\delta}_i$ represent the estimate error of $\Upsilon_i$ and $\delta_i$.

## 5. Simulation

### 5.1. *Simulation* 1

To illustrate the advantages of the proposed predefined-time controller based on Null space, this paper compares the proposed algorithm with the existing predefined-time cooperative formation control algorithm [27]. Figure 4 shows the communication graph of UAVs. The communication topology can be represented by the adjacency matrix:

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad N = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \tag{53}$$

where the $M$ matrix represents the communication topology between each follower, while the $N$ matrix represents the communication topology between the leader and each follower.

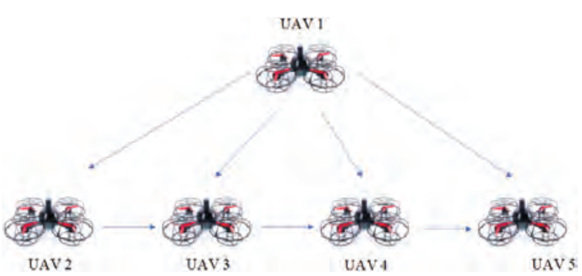



Fig. 4. Communication topology of 5 UAV.

Table 1. The related parameter values.

| Parameter | UAV 1 | UAV 2 | UAV 3 | UAV 4 | UAV 5 |
|---|---|---|---|---|---|
| $m_i(kg)$ | 1.2 | 1.2 | 1.2 | 1.2 | 1.2 |
| $g(-kg/s^2)$ | 9.81 | 9.81 | 9.81 | 9.81 | 9.81 |
| $I_{i_{xx}}(kg \cdot m^2)$ | 0.019 | 0.019 | 0.019 | 0.019 | 0.019 |
| $I_{i_{yy}}(kg \cdot m^2)$ | 0.019 | 0.019 | 0.019 | 0.019 | 0.019 |
| $I_{i_{zz}}(kg \cdot m^2)$ | 0.021 | 0.021 | 0.021 | 0.021 | 0.021 |
| $\Gamma_i$ | $2.5I_6$ | $2.5I_6$ | $2.5I_6$ | $2.5I_6$ | $2.5I_6$ |
| $\Upsilon_i$ | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 |

Table 2. The initial states of each UAV.

| State | UAV 1 | UAV 2 | UAV 3 | UAV 4 | UAV 5 |
|---|---|---|---|---|---|
| $p_i(m)$ | $\begin{bmatrix}0\\1\\1\end{bmatrix}$ | $\begin{bmatrix}-1\\-1\\1\end{bmatrix}$ | $\begin{bmatrix}-1\\0\\1\end{bmatrix}$ | $\begin{bmatrix}1\\0\\1\end{bmatrix}$ | $\begin{bmatrix}2\\-1\\1\end{bmatrix}$ |
| $\beta_i(rad)$ | $\begin{bmatrix}1\\1\\1\end{bmatrix}$ | $\begin{bmatrix}1\\1\\1\end{bmatrix}$ | $\begin{bmatrix}1\\1\\1\end{bmatrix}$ | $\begin{bmatrix}1\\1\\1\end{bmatrix}$ | $\begin{bmatrix}1\\1\\1\end{bmatrix}$ |
| $\boldsymbol{\tau}_{p_i}(m)$ | $\begin{bmatrix}0.2c(t)\\0.1s(t)\\0.1s(t)\end{bmatrix}$ | $\begin{bmatrix}0.3c(t)\\0.2s(t)\\0.2s(t)\end{bmatrix}$ | $\begin{bmatrix}0.3c(t)\\0.1s(t)\\0.4s(t)\end{bmatrix}$ | $\begin{bmatrix}0.2c(t)\\0.3s(t)\\0.1s(t)\end{bmatrix}$ | $\begin{bmatrix}0.4c(t)\\0.3s(t)\\0.1s(t)\end{bmatrix}$ |
| $\boldsymbol{\tau}_{\beta_i}(rad)$ | $\begin{bmatrix}0.1s(t)\\0.2c(t)\\0.1s(t)\end{bmatrix}$ | $\begin{bmatrix}0.1s(t)\\0.2c(t)\\0.2s(t)\end{bmatrix}$ | $\begin{bmatrix}0.1s(t)\\0.1c(t)\\0.1s(t)\end{bmatrix}$ | $\begin{bmatrix}0.1s(t)\\0.1c(t)\\0.1s(t)\end{bmatrix}$ | $\begin{bmatrix}0.1s(t)\\0.1c(t)\\0.1s(t)\end{bmatrix}$ |

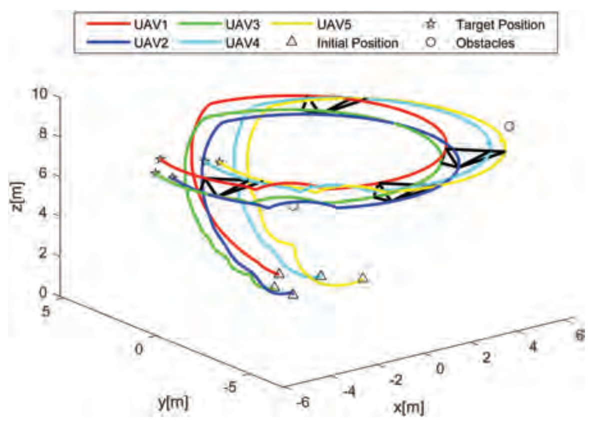



Fig. 5. Trajectories of UAVs with the PTNSBC-TBG algorithm in 3D.

The obstacles are located at $O_1 = [5.5, -4.5, 9]^T m$, $O_2 = [-3, -3.5, 7]^T m$. The desired trajectory of the leader (UAV1) is $X_r, 1(t) = [5s(0.1t); 5c(0.1t); 8; 0; 0; 0]$, where the $s(\cdot) = \sin(\cdot)$ and $c(\cdot) = \cos(\cdot)$. The safety distance $d_d = 1.5$ m. The related parameters of each UAV are shown in Table 1.

The initial states of each UAV are shown in Table 2.

Figures (5) and (6) show the trajectories of the UAVs with the PTNSBC-TBG algorithm. Five UAVs have formed a fixed formation and are moving stably along the desired trajectory, successfully avoiding obstacles near the path and reaching the predetermined target location.

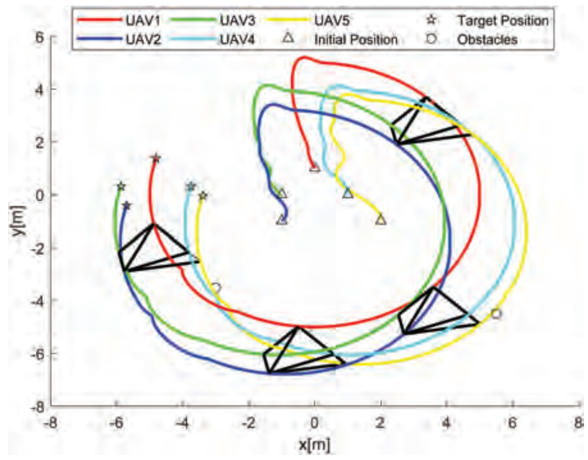



Fig. 6. Trajectories of UAVs with the PTNSBC-TBG algorithm in X-Y plane.

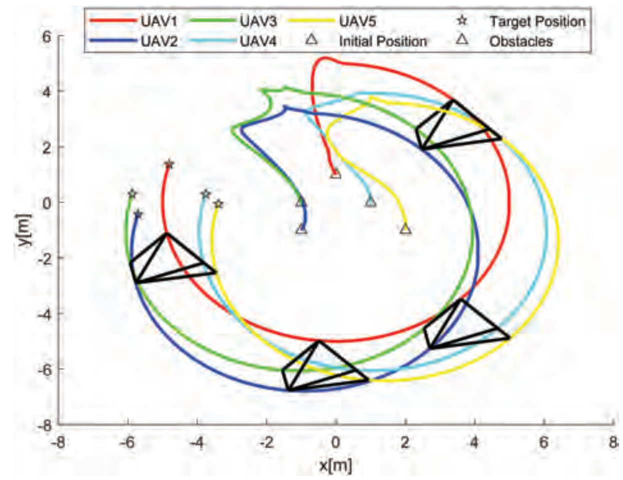



Fig. 8. Trajectories of UAVs with the PTNSBC-TBG algorithm but without obstacle in X-Y plane.

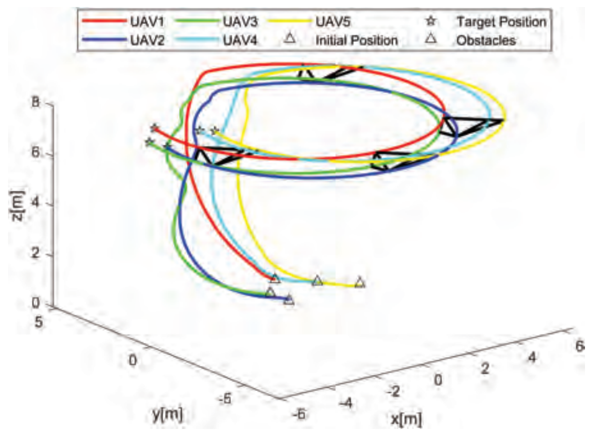



Fig. 7. Trajectories of UAVs with the PTNSBC-TBG algorithm but without obstacle in 3D.

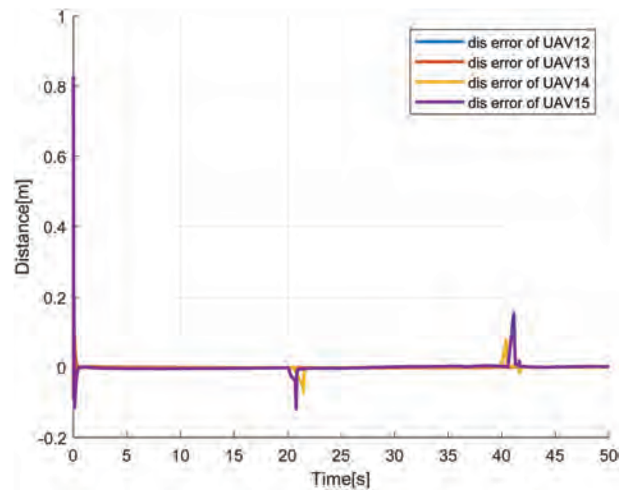



Fig. 9. Distance of each UAV with the PTNSBC-TBG algorithm.

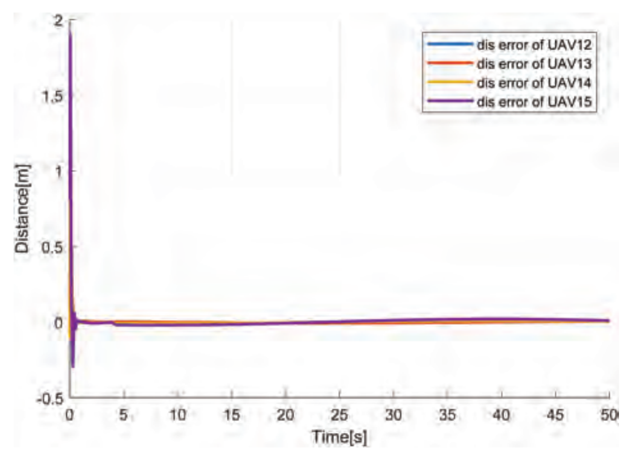



Fig. 10. Distance of each UAV with the PTNSBC-TBG algorithm without obstacles.

Figures (7) and (8) show the trajectories of the UAVs with the PTNSBC-TBG algorithm but without obstacle. Five UAVs form a fixed formation and are moving stably along the desired trajectory, it is capable of reaching the target location stably and accurately.

It is evident that UAVs can achieve formation within the predefined-time, regardless of the presence of obstacles. Figures (9) and (10) show the distance of each UAV, Figs. (11) and (12) show the angle of each UAV, Figs. (13) and (14) show the yaw angle of the lead, Figs. (15)–(18) show the control input of each UAV.

From the above simulation results, it is shown that all UAVs can achieve the desired shape and each one can reach the desired position which can follow from the UAV model in predefined time 5 s.

In addition, we compared the proposed algorithm with predefined-time algorithm [27]. From the input amplitude, it can be seen that the proposed algorithm has significantly

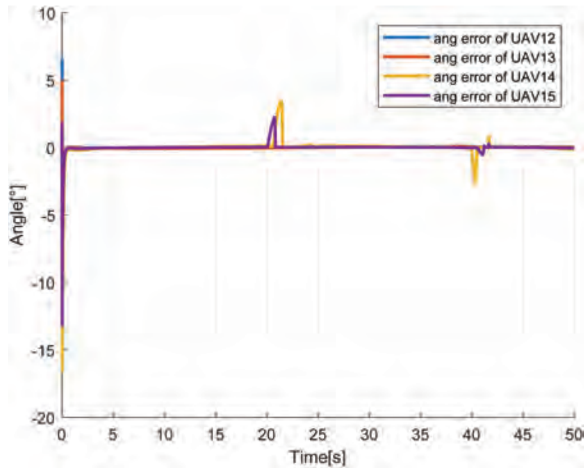



Fig. 11. Angle of each UAV with the PTNSBC-TBG algorithm.

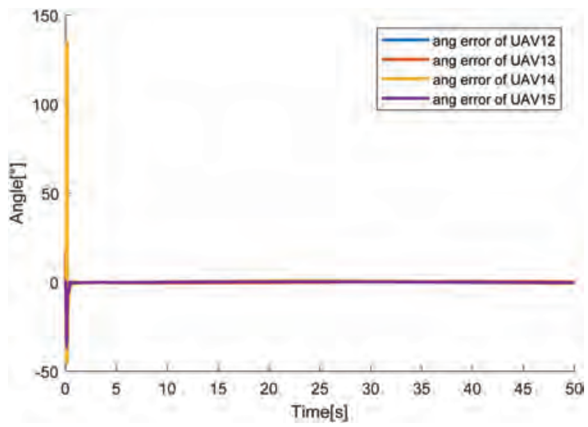



Fig. 12. Angle of each UAV with the PTNSBC-TBG algorithm without obstacles.

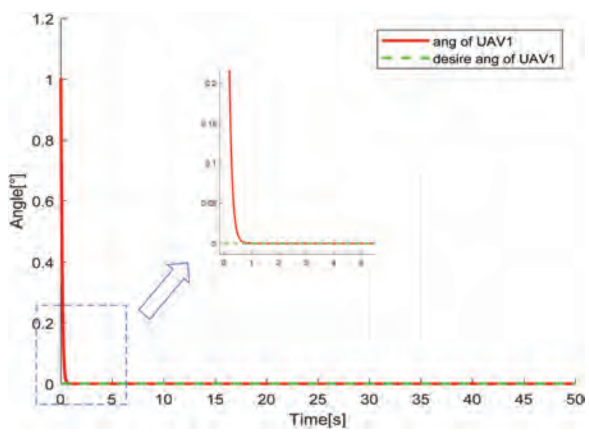



Fig. 13. Yaw angle of leader UAV with the PTNSBC-TBG algorithm.

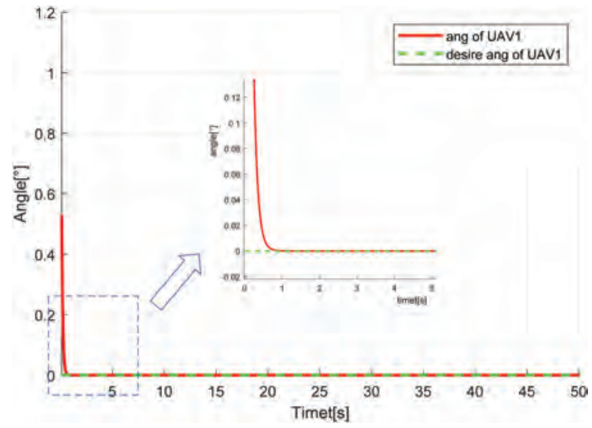



Fig. 14. Yaw angle of leader UAV with the PTNSBC-TBG algorithm without obstacles.

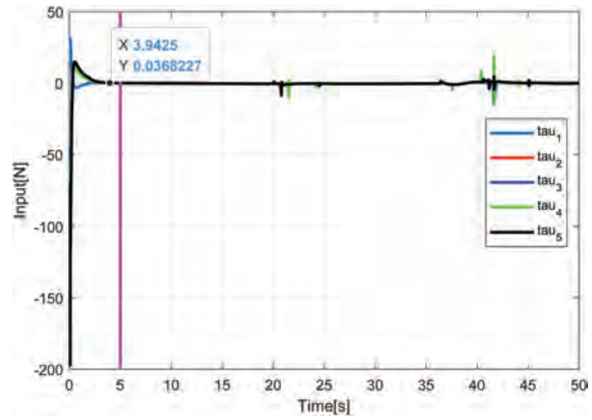



Fig. 15. Input of UAVs with the PTNSBC-TBG algorithm.

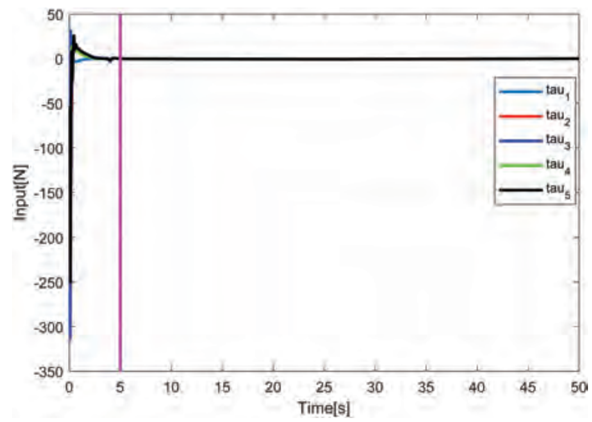



Fig. 16. Input of UAVs with the PTNSBC-TBG algorithm without obstacles.

smaller limitations on control inputs, which is more in line with the needs of actual systems.

Furthermore, we compared the proposed algorithm with predefined-time algorithm [38]. From the input amplitude, it can be seen that the proposed algorithm also has significantly smaller limitations on control inputs.

The initial control input in this paper is approximately 300 N, which is more consistent with the torque constraints

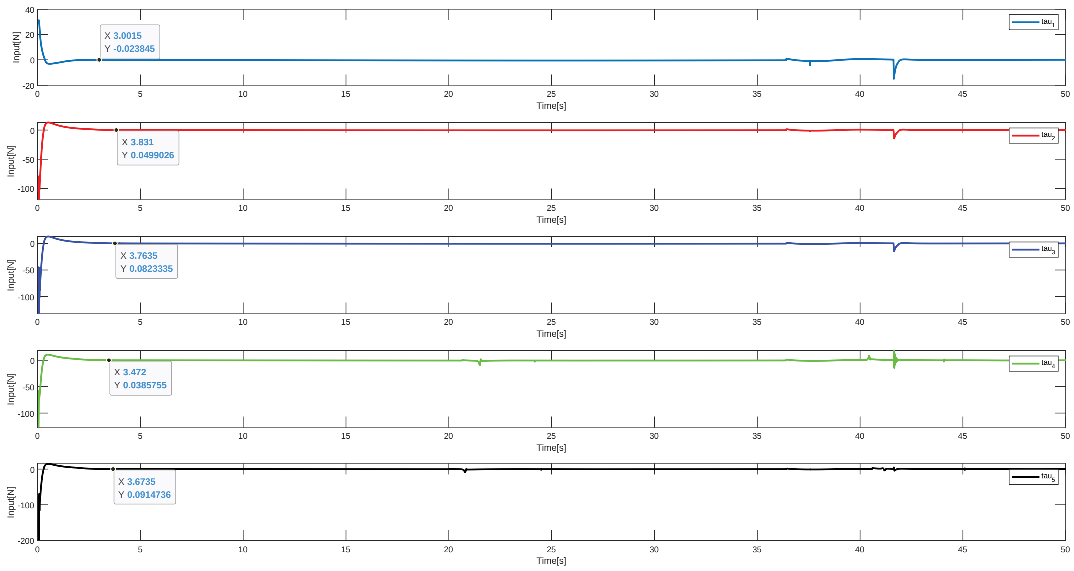



Fig. 17. Input of each UAV with the PTNSBC-TBG algorithm.

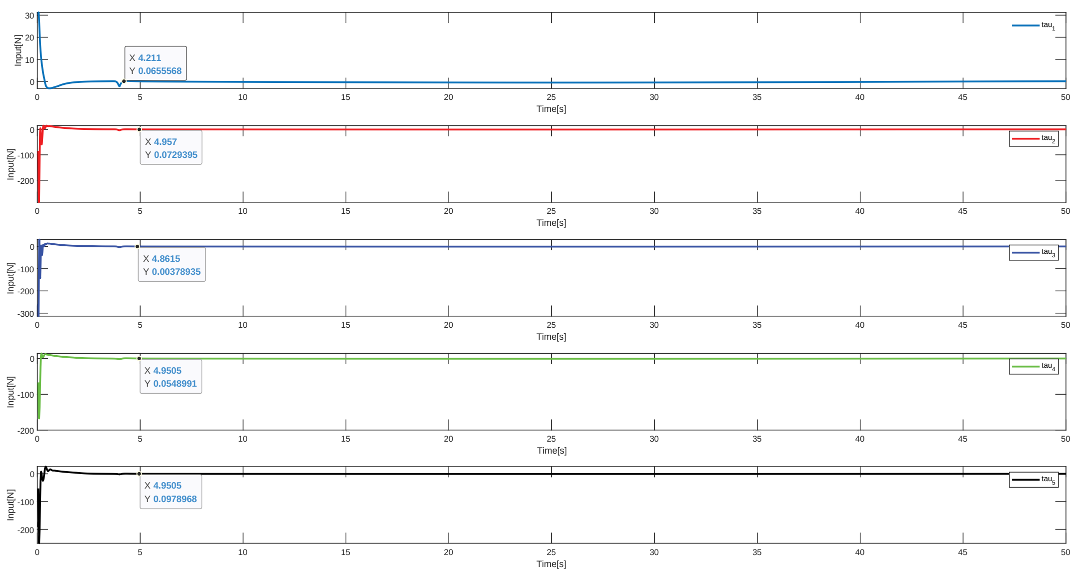



Fig. 18. Input of each UAV with the PTNSBC-TBG algorithm without obstacles.

of practical motors, in contrast to the 4000 N presented in [27] and 1500 N in [38].

**Remark 1.** In addition to the control time requirement for controlling UAVs, energy consumption is another important indicator [39], because even if a UAV is theoretically controllable, it may be uncontrollable in reality if it consumes a large amount of energy. For UAV controllers, the required energy cost is [40]:

$$\mathbf{C} = \int_0^{t_f} \sum_{i=1}^{\aleph} \|\mathbf{u}_i(t)\|^2 dt. \tag{54}$$

The solid curves represent the numerical evaluation of each UAV's energy consumption and are shown in Fig. (23).

### 5.2. *Simulation* 2

To illustrate the Scalability of the proposed predefined-time controller based on Null space, this paper increased the number of UAVs for the proposed algorithm and conducted simulation verification, increasing the number of UAVs to 20. The specific simulation results are as follows:

Figures 24 and 25 show the trajectories of the UAVs with the PTNSBC-TBG algorithm. Figure 26 shows the distance

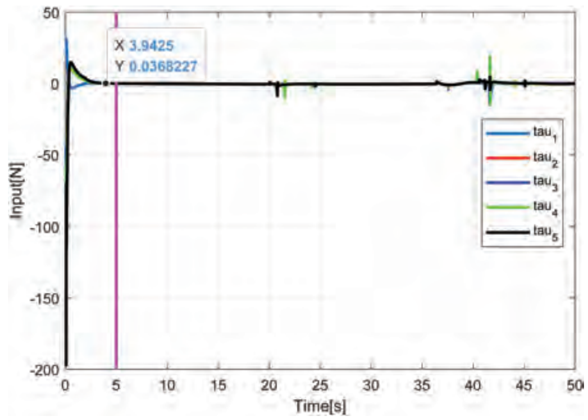



Fig. 19. Input of UAVs with the PTNSBC-TBG algorithm.

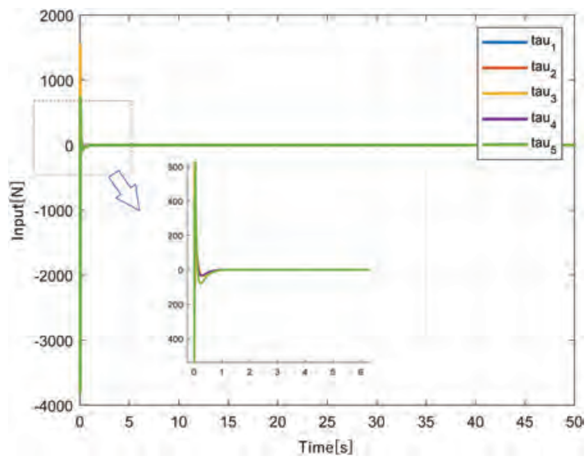



Fig. 20. Input of each UAV with the predefined-time cooperative formation control algorithm [27].

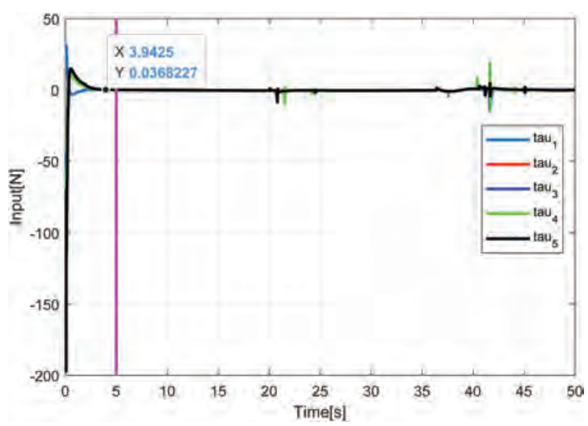



Fig. 21. Input of UAVs with the PTNSBC-TBG algorithm.

error of of each UAV. It is evident that Twenty UAVs have formed a fixed formation and are moving stably along the desired trajectory, successfully avoiding obstacles near the path and reaching the predetermined target location.

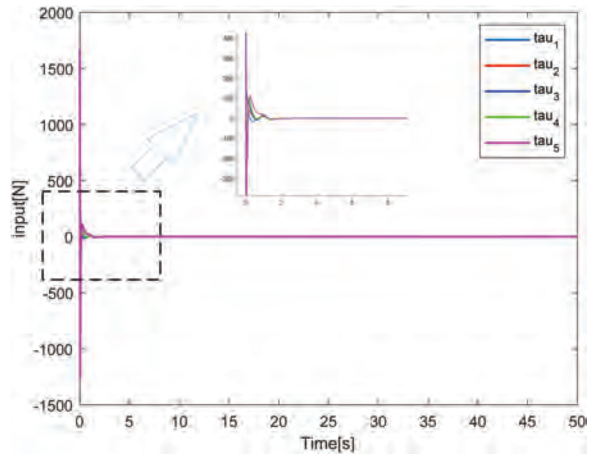



Fig. 22. Input of each UAV with the predefined-time cooperative formation control algorithm [38].

Figures 28–31 show the control input of each UAV. It is clear that each UAV's input is smaller than 500 N, and the initial input is decreased by approximately 66% than [38] and [27].

As the task design process of the proposed algorithm involves deriving a general formula for a formation of $i$ UAVs, and the controller design is also designed for each UAV with the same form but different parameters, theoretically there is no limit to the number of UAVs that the algorithm can adapt to. In terms of computational complexity, due to the planning layer, the Jacobian matrix calculation process for UAVs' formation tasks, and the matrix calculations involved in controller design only involve the UAV itself, its parent node UAVs, and leader information. Therefore, increasing the number of unmanned aerial vehicles alone does not increase the computational complexity of the algorithm. In terms of communication cost, since the designed algorithm only requires the existence of a path between its parent node and the leader to satisfy the topology condition of the minimum spanning tree, increasing the number of UAVs only increases the number of child nodes without affecting the minimum structure required for the topology graph. Therefore, it does not generate greater communication cost.

## 6. Experiment

In this section, we conducted physical experiment using four UAVs. First, introduce the experimental platform and then provide the experimental results. The experimental video can be found on YouTube (https://youtu.be/PIEuXkgB7jA).

The UAV experimental platform consists of the decision subsystem (UAV subsystem), the RTK-GPS positioning subsystem (RTK-GPS and laptop) and the wireless communication subsystem, as shown in Fig. 32.

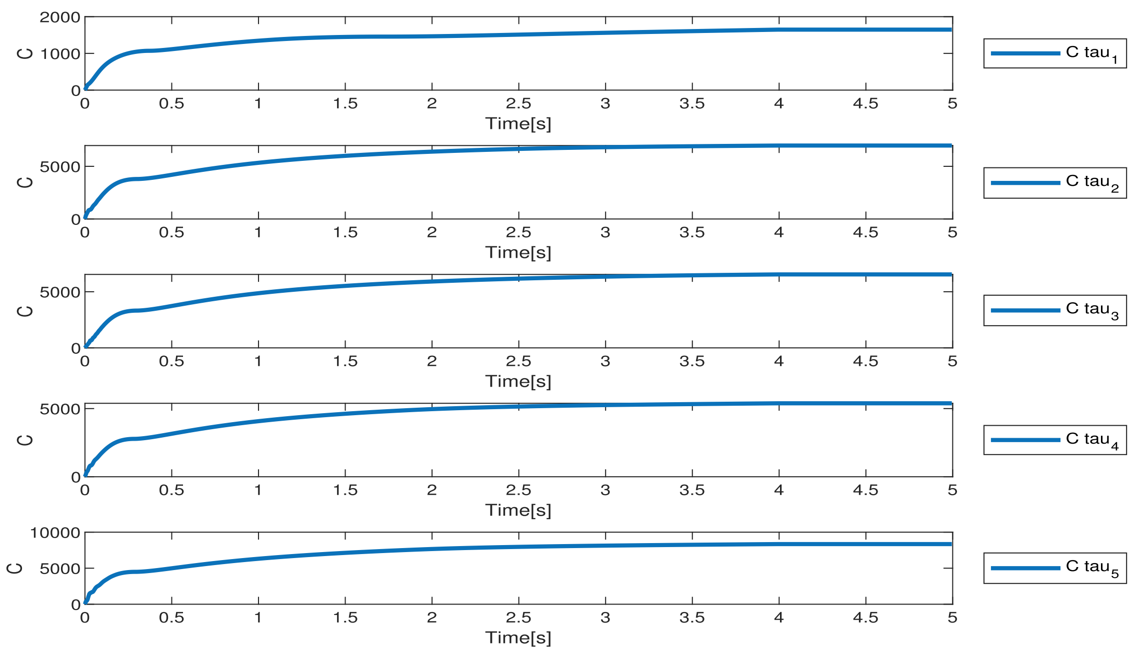



Fig. 23. Energy consumption of each UAV with PTNSBC-TBG algorithm.

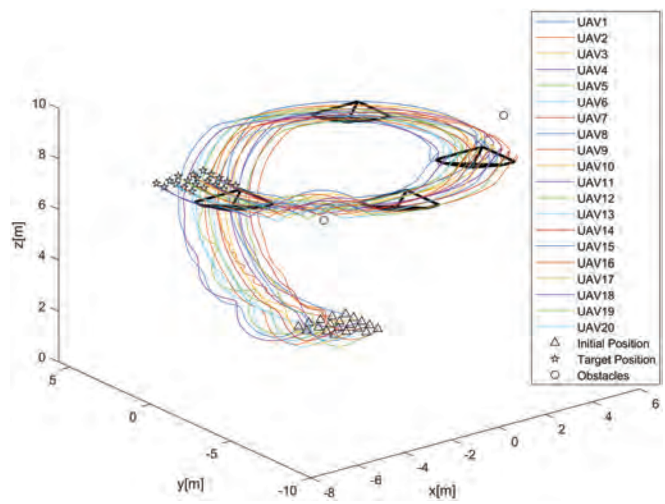



Fig. 24. Trajectories of 20 UAVs with the PTNSBC-TBG algorithm.

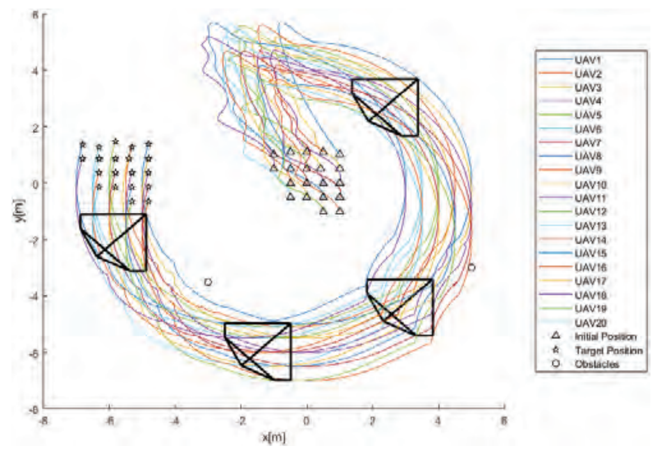



Fig. 25. Trajectories of 20 UAVs with the PTNSBC-TBG algorithm in X-Y plane.

In the experiment, four quad-rotor UAVs are utilized, The schematic configuration of the experiment is shown in Fig. 32. Specifically, a real-time kinematic (RTK)-based station obtains the poses of the UAVs and sends carrier calibration data to a personal computer (PC), thereby reducing the computational burden on the UAVs. The processed coordinates of each UAV are transmitted to the respective UAV, which can only access its own pose information through the PC. Neighbor information is obtained based on the communication topology. Once each UAV acquires its local state information, it utilizes the proposed algorithm to compute the control strategy. Finally, the UAVs are directed to the target using target waypoint information by the drive module, which comprises a brushless direct current motor (BLDCM) and three-blade propellers. In this experiment, the UAVs only utilize their own sensors to detect other UAVs. Each UAV obtains neighbor state information through an undirected communication topology. In the experiment, four UAVs are used to perform formation tasks based on the PTNSBC-TBG method, set the initial error to be small and only consider the process of UAV formation task. The initial position of each UAV is $X_1 = [0.1, 1.1, 1, 1, 1, 1]^T$, $X_2 = [-0.8, -0.8, 1, 1, 1, 1]^T$, $X_3 = [-1, -0.1, 1, 1, 1, 1]^T$ and $X_4 = [1.1, -0.2, 1, 1, 1, 1]^T$. The safety pose between UAVs is 0.5, The desired formation

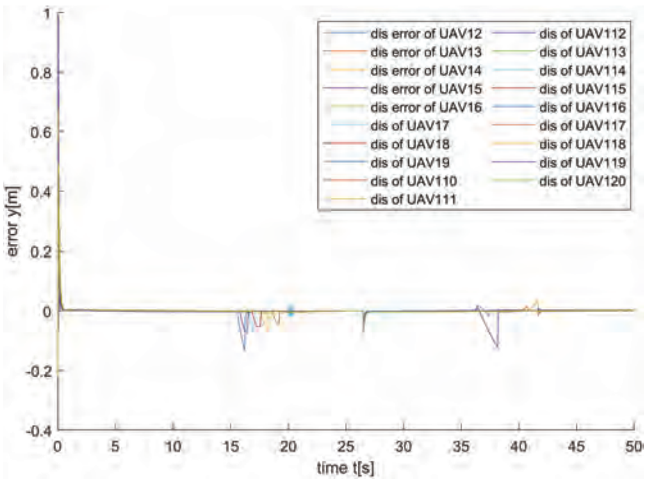



Fig. 26. Distance error of each UAV with the PTNSBC-TBG algorithm.

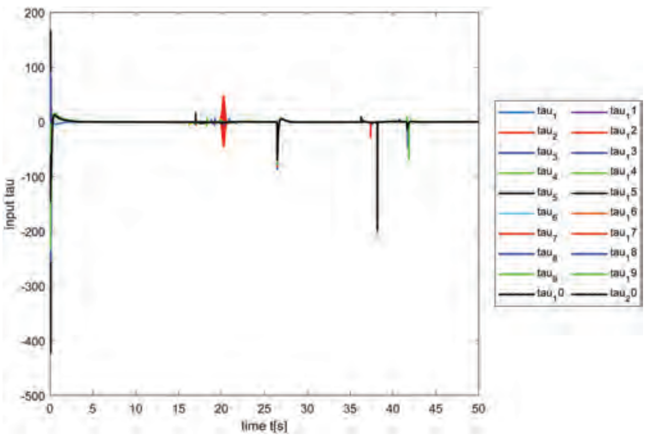



Fig. 27. Input of 20 UAVs with the PTNSBC-TBG algorithm.

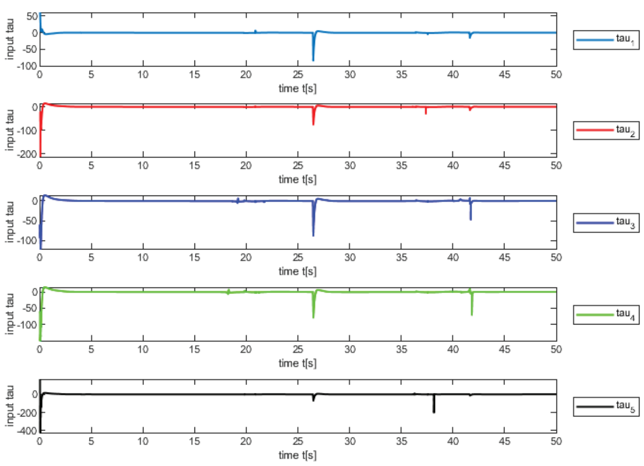



Fig. 28. Input of UAV1 to UAV5 with the PTNSBC-TBG algorithm.

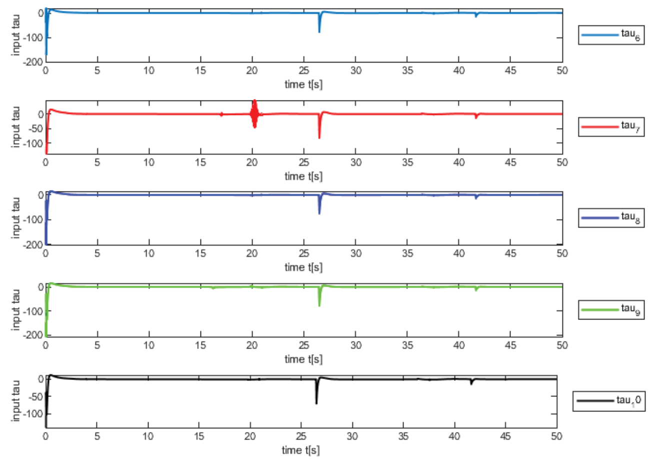



Fig. 29. Input of UAV6 to UAV10 with the PTNSBC-TBG algorithm.

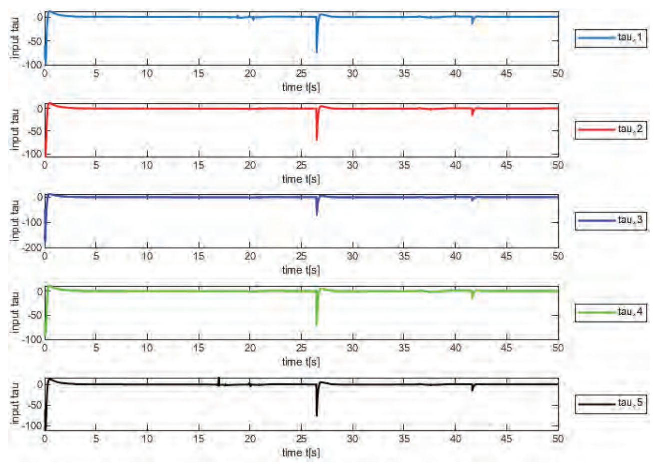



Fig. 30. Input of UAV11 to UAV15 with the PTNSBC-TBG algorithm.

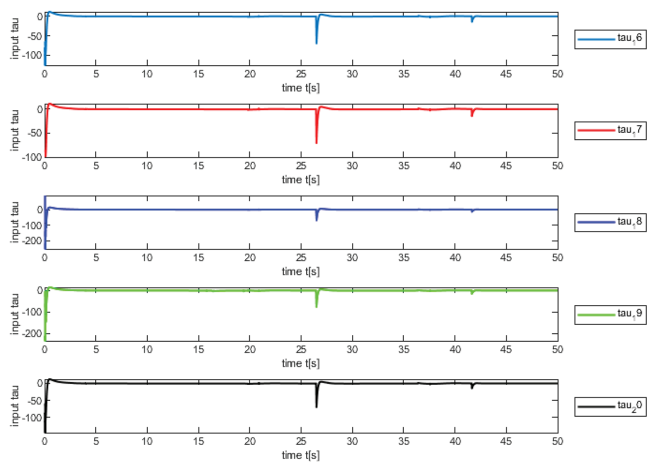



Fig. 31. Input of UAV16 to UAV20 with the PTNSBC-TBG algorithm.

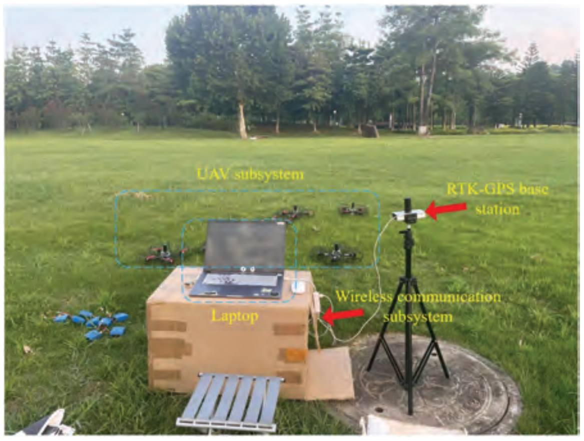



Fig. 32. UAV experimental platform.

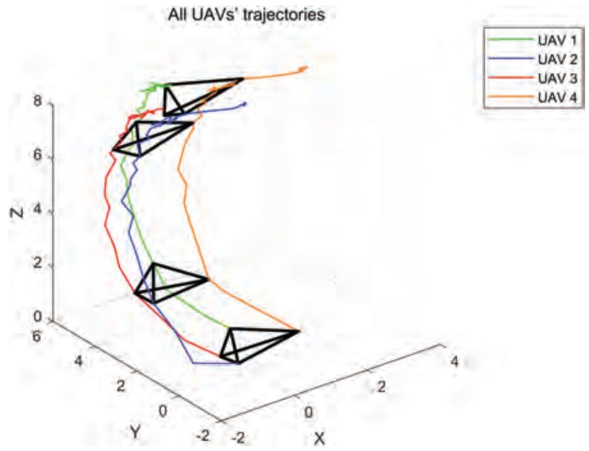
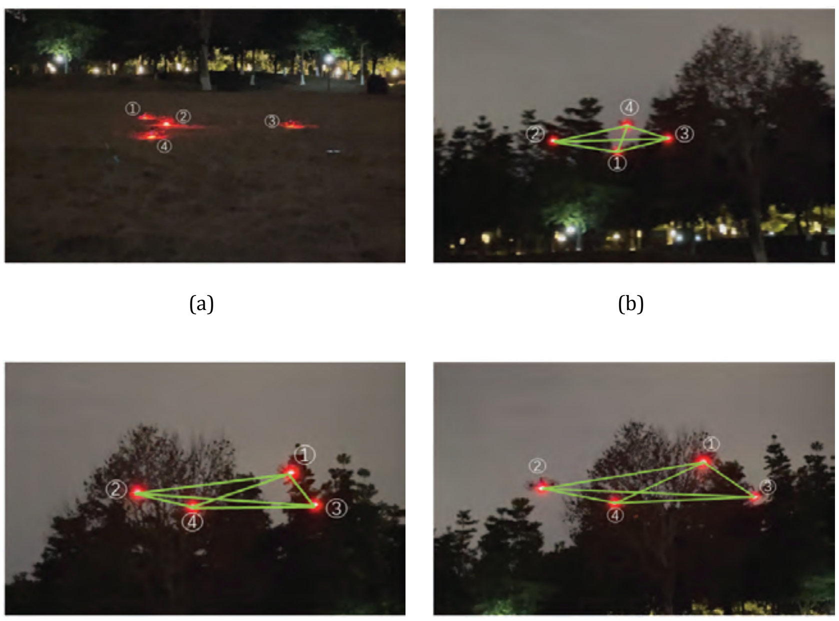



Fig. 33. Experimental formation trajectories of four UAVs.

trajectory is $X_{r,1}(t) = [2.5s(0.1t); 2.5c(0.1t); 8; 0; 0; 0]^T$. In the experiment, the control inputs are given according to Eq. (49) and integrated into the STM32 microcontroller-based flight control system of the UAVs using C++ programming language to calculate the UAV's flight trajectory in real time. Each UAV moves based on the obtained flight trajectory. Therefore, the experimental results are presented in the form of UAV flight trajectories in this experiment. Due to objective factors such as environmental wind and positioning accuracy, the trajectories of UAVs' formations may exhibit oscillatory fluctuations. The flight trajectories of each UAV formation are shown in Fig. 33.

The experimental field views of the UAVs formation at 0 s, 2 s, 23 s, 50 s are shown in Fig. (34). Figure (34) (a) shows all UAVs placed on the ground waiting to take off to the starting point of the algorithm. Figure 34 (b) shows all UAVs arriving at the starting point of the algorithm and waiting for algorithm instructions. Figure 34 (c) shows each UAV forming a formation and moving along the expected trajectory. Figure 34 (d) shows each UAV flying for 50 s before reaching the target point position. It can be concluded that the proposed PTNSBC-TBG algorithm can achieve the desired shape and each one can reach the desired position which can follow from the UAV model.



(a) (b)

(c) (d)

Fig. 34. Snapshots of the experimental formation shape at 0 s (a), 2 s (b), 23 s (c) and 50 s (d), ① ∼ ④ represents the UAV numbers $1 \sim 4$.

## 7. Conclusions

This paper addresses the problem of predefined-time formation control for UAVs in the presence of task conflicts. First, this transformation not only resolves UAV task-level conflicts but also simplifies the UAVs formation control problem into a single UAV tracking control problem. Second, by integrating the TBG into the proposed NSBC framework, predefined-time convergence is achieved at the task level and providing real-time planned tracking trajectories within a predefined time. Finally, a novel predefined-time formation tracking controller and an estimator is proposed which enables UAVs to avoid obstacles and achieve formation control within a predefined time.

The initial control input of the proposed algorithm is significantly smaller than general predefined-time control algorithms [27], decrease the initial input by approximately 66%, thus better aligning with the requirements of practical systems. The algorithm this paper proposed has the potential to be applied to practical scenarios, such as multi-UAV cooperative handling, joint formation inspections, multi-drone collaborative encirclement and multi-UAV light shows.

## Appendix A

In this section, the proof processes for mission stability and control stability of the proposed algorithm are introduced. Due to the unique mission conflict resolution mode of the NSBC method, it is essential to demonstrate the mission stability of the NSBC method [31, 34, 41].

### *A.1. Proof the stability of mission*

In this section, we can prove that the proposed composite behavior (37) can enable each UAV to achieve formation behavior and obstacle avoidance behavior at the predetermined time, the proof process is similar to the literature [31].

First, introduce a Lemma which is similar to Lemma 2, i.e. for the proposed composite behavior speed, there exists a predetermined time $t_{f_1}$ that satisfies the following equation for all UAVs:

$$\begin{cases} \lim\limits_{t\to t_{f_1}} |\sigma_{i,X}| \leq N\sqrt{\dfrac{2\varrho_1}{1+\varrho_1}V_\sigma(0)}, \\ |\sigma_{i,X}| \leq N\sqrt{\dfrac{2\varrho_1}{1+\varrho_1}V_\sigma(0)}, \quad \forall\, t > t_{f_1}, \\ \lim\limits_{t\to t_\infty} |\sigma_{i,X}| = 0, \end{cases} \tag{A.1}$$

where $\sigma_{i,X} = \sigma_{i,F}, \sigma_{i,O}$.

**Proof.** According to the relationship between the distance of each UAV relative to the obstacle and the safe distance $d_d$, task conflicts can be divided into two situations.

(1) When the distance is greater than $d_d$, there is no conflict between the formation task and the obstacle avoidance task, i.e. $\mathcal{J}_{i,F}\mathcal{J}_{i,O}^{\dagger} = 0$.
(2) When the distance is smaller than $d_d$, two task behaviors conflict, with $\mathcal{J}_{i,F}\mathcal{J}_{i,O}^{\dagger} \neq 0$.

First, considering the absence of task conflicts, the following Lyapunov function is chosen:

$$V_\sigma = \frac{1}{2}\sum_{i=1}^{N}\left(\tilde{\sigma}_{i,F}^T\gamma_F\tilde{\sigma}_{i,F} + \tilde{\sigma}_{i,O}^T\gamma_O\tilde{\sigma}_{i,O}\right). \tag{A.2}$$

Then, by taking the derive function $V_\sigma$ with respect to time $t$, we can obtain that:

$$\begin{aligned} \dot{V}_\sigma &= \sum_{i=1}^{N}\left(\gamma_F\tilde{\sigma}_{i,F}^T\dot{\tilde{\sigma}}_{i,F} + \gamma_O\tilde{\sigma}_{i,O}^T\dot{\tilde{\sigma}}_{i,O}\right) \\ &= -\sum_{i=1}^{N}(\gamma_F\tilde{\sigma}_{i,F}^T\mathcal{J}_{i,F}\dot{\mathbf{X}}_{i,r} + \gamma_O\tilde{\sigma}_{i,O}^T\mathcal{J}_{i,O}\dot{\mathbf{X}}_{i,r}) \\ &\leq \sum_{i=1}^{N}((\Theta_i^{-1}\mathcal{J}_{i,F}^{\dagger}\Lambda_F(\Xi_1+1)\tilde{\sigma}_{i,F})(\gamma_F\tilde{\sigma}_{i,F}^T\mathcal{J}_{i,F}(I-\mathcal{J}_{i,O}^{\dagger}\mathcal{J}_{i,O}) \\ &\quad + \gamma_O\tilde{\sigma}_{i,O}^T\mathcal{J}_{i,O}((I-\mathcal{J}_{i,O}^{\dagger}\mathcal{J}_{i,O})) + \mathcal{J}_{i,O}^{\dagger}\Lambda_O(\Xi_1+1)\tilde{\sigma}_{i,O})) \\ &\leq \sum_{i=1}^{N}\left(\gamma_F\tilde{\sigma}_{i,F}^T\Lambda_F\,(\Xi_1+1)\,\tilde{\sigma}_{i,F} + \gamma_O\tilde{\sigma}_{i,O}^T\Lambda_O\,(\Xi_1+1)\,\tilde{\sigma}_{i,O}\right) \\ &\leq -\,(\Xi_1+1)\,\Lambda_{\min}\sum_{i=1}^{N}\left(\gamma_F\tilde{\sigma}_{i,F}^T\tilde{\sigma}_{i,F} + \gamma_O\tilde{\sigma}_{i,O}^T\tilde{\sigma}_{i,O}\right) \\ &\leq -2\,(\Xi_1+1)\,\Lambda_{\min}V_\sigma \leq -2\Xi_1\Lambda_{\min}V_\sigma \\ &= \frac{\dot{h}_1\,(t)}{(1-h_1\,(t)+\varrho_1)}V_\sigma. \end{aligned} \tag{A.3}$$

According to the Lemma 1, it can be obtained that $V_\sigma$ converges to a set $\Omega = \{V_\sigma | V_\sigma \leq \frac{\varrho_1}{1+\varrho_1}\}$ at a predetermined time $t_{f1}$. The convergence time $T_\sigma$ does not depend on the initial state of the system and is a predetermined time constant, and the convergence accuracy of the task error can be increased by reducing the value of parameter $\varrho_1$. What's more, when $t > t_{f1}$, the task error of behavior satisfies $|\tilde{\sigma}_{i,F}|, |\tilde{\sigma}_{i,O}| < N\sqrt{\frac{2\varrho_1}{1+\varrho_1}V_\sigma(0)}$ and $V_\sigma$ is a monotonically decreasing function, then we can obtain that $\lim_{t\to\infty}\sigma_{i,F} = 0$ and $\lim_{t\to\infty}\sigma_{i,O} = 0$.

Second, consider situations where there are task conflicts, taking the derive function $V_\sigma$ with respect to time $t$, we can obtain:

$$\dot{V}_\sigma = -\sum_{i=1}^{N}(\gamma_F\tilde{\sigma}_{i,F}^T J_{i,F}\dot{X}_{r,i} + \gamma_O\tilde{\sigma}_{i,O}^T J_{i,O}\dot{X}_{r,i})$$

$$\le -\sum_{i=1}^{N} (\gamma_F \tilde{\sigma}_{i,F}^T J_{i,F} (J_{i,O}^{\dagger} \Lambda_{i,O} (\Xi_1 + 1) \tilde{\sigma}_{i,O} + (I - J_{i,O}^{\dagger} J_{i,O}) J_{i,F}^{\dagger} (\Lambda_{i,F} (\Xi_1 + 1) \tilde{\sigma}_{i,F})) + \gamma_O \tilde{\sigma}_{i,O}^T J_{i,O} (J_{i,O}^{\dagger} \Lambda_{i,O} (\Xi_1 + 1) \tilde{\sigma}_{i,O} + (I - J_{i,O}^{\dagger} J_{i,O}) J_{i,F}^{\dagger} (\Lambda_{i,F} (\Xi_1 + 1) \tilde{\sigma}_{i,F})$$

$$\le -\sum_{i=1}^{N} (\gamma_F \tilde{\sigma}_{i,F}^T J_{i,F} J_{i,O}^{\dagger} \Lambda_{i,O} (\Xi_1 + 1) \tilde{\sigma}_{i,O} + \gamma_F \tilde{\sigma}_{i,F}^T J_{i,F} (I - J_{i,F} J_{i,O}^{\dagger} J_{i,O} J_{i,F}^{\dagger})) J_{i,F}^{\dagger} \times (\Lambda_{i,F} (\Xi_1 + 1) \tilde{\sigma}_{i,F}) + \gamma_O \tilde{\sigma}_{i,O}^T \Lambda_{i,O} (\Xi_1 + 1) \tilde{\sigma}_{i,O})$$

$$\le -\sum_{i=1}^{N} (\gamma_O \Lambda_{i,O} (\Xi_1 + 1) \left\| \tilde{\sigma}_{i,O} \right\|_2^2) + \sum_{i=1}^{N} (\gamma_F \Lambda_{i,O} (\Xi_1 + 1) \left\| J_{i,F} \right\|_2 \left\| J_{i,O}^{\dagger} \right\|_2 \left\| \tilde{\sigma}_{i,F}^T \right\|_2 \left\| \tilde{\sigma}_{i,O} \right\|_2). \tag{A.4}$$

Next, the Lyapunov function will be proven in the following two cases:

(1) The task error satisfies: $\|\tilde{\sigma}_{i,0}\| \ge \|\tilde{\sigma}_{i,F}\| > O_1$.

First, it is proven that for any initial error $\tilde{\sigma}_{i,F}(0)$ of the formation behavior task, the error $\tilde{\sigma}_{i,O}(0)$ of the obstacle avoidance behavior task is bounded. Obviously, it is easy to prove the Jacobian matrix satisfied $\|J_{i,F}\|_2 \le 1$ for formation tasks, and the 2-norm of the pseudo-inverse of the Jacobian matrix for obstacle avoidance tasks is bounded, i.e. $\|J_{i,O}^{\dagger}\|_2 \le L_{i,O}$. Therefore, at time $t = 0$, Eq. (A.4) can be rewritten as

$$\dot{V}_{\sigma}(0) \le -\sum_{i=1}^{N} (\gamma_O \Lambda_{i,O} (\Xi_1 + 1) \left\| \tilde{\sigma}_{i,O}(0) \right\|_2^2) + \sum_{i=1}^{N} (\gamma_F \Lambda_{i,O} (\Xi_1 + 1) L_{i,O} \left\| \tilde{\sigma}_{i,F}^T(0) \right\|_2 \left\| \tilde{\sigma}_{i,O}(0) \right\|_2)$$
$$\le -\sum_{i=1}^{N} (\gamma_O \Lambda_{i,O} (\Xi_1 + 1) \left\| \tilde{\sigma}_{i,O}(0) \right\|_2^2) + \sum_{i=1}^{N} (\gamma_F \Lambda_{i,O} (\Xi_1 + 1) L_{i,O} \left\| \tilde{\sigma}_{i,O}(0) \right\|_2^2)$$
$$\le -\sum_{i=1}^{N} (\Lambda_{i,O} (\Xi_1 + 1) \left\| \tilde{\sigma}_{i,O}(0) \right\|_2^2) \le 0 \tag{A.5}$$

among them, preset the parameter $\gamma_O \ge \gamma_{O,1} := \gamma_F L_{i,O} + 1$, So when the parameter $\gamma_O \ge \gamma_{O,1}$, we can obtain that $\dot{V}_{\sigma}(0) \le 0$. According to Eq. (A.5), it can be concluded that for any time $\Delta t > 0 \in \mathbb{R}$, the task error $\|\tilde{\sigma}_{i,O}(\Delta t)\| \le \|\tilde{\sigma}_{i,O}(0)\|$ and the derivative of Lyapunov function $\dot{V}_{\sigma}(\Delta t) \le 0$ is satisfied.

Therefore, if there is $\gamma_O \ge \gamma_{O,1}$, then $\dot{V}_{\sigma}(t) \le 0$ holds and $\|\tilde{\sigma}_{i,F}(t)\| \le \|\tilde{\sigma}_{i,F}(0)\|$ can be derived, and since there is $-\|\tilde{\sigma}_{i,O}\| \le -\|\tilde{\sigma}_{i,F}\|$, Eq. (A.4) can be rewritten as

$$\dot{V}_{\sigma} \le -\sum_{i=1}^{N} (\Lambda_{i,O} (\Xi_1 + 1) \left\| \tilde{\sigma}_{i,O} \right\|_2^2)$$
$$\le -\sum_{i=1}^{N} (\frac{1}{2} \Lambda_{i,O} (\Xi_1 + 1) \left\| \tilde{\sigma}_{i,O} \right\|_2^2 + \frac{1}{2} \Lambda_{i,O} (\Xi_1 + 1) \left\| \tilde{\sigma}_{i,F} \right\|_2^2)$$
$$\le -(\Xi_1 + 1) \Lambda_{i,O} V_{\sigma}. \tag{A.6}$$

(2) The task error satisfies: $O_1 < \|\tilde{\sigma}_{i,0}\| < \|\tilde{\sigma}_{i,F}\| \le L_0$.

Similar to the situation (1), if the parameter $\gamma_O \ge \gamma_{O,2} := \gamma_F L_{OA,i} L_0 / O_1 + 1$, we can obtain:

$$\dot{V}_{\sigma}(0) \le -\sum_{i=1}^{N} (\gamma_O \Lambda_{i,O} (\Xi_1 + 1) \left\| \tilde{\sigma}_{i,O}(0) \right\|_2^2 + \sum_{i=1}^{N} (\gamma_F \Lambda_{i,O} (\Xi_1 + 1) L_{OA,i} \left\| \tilde{\sigma}_{i,O}(0) \right\|_2^2)$$
$$\le -\sum_{i=1}^{N} (\Lambda_{i,O} (\Xi_1 + 1) \left\| \tilde{\sigma}_{i,O}(0) \right\|_2^2) \le 0. \tag{A.7}$$

Similarly, for any time $t > 0 \in \mathbb{R}$, if $\gamma_O \ge \gamma_{O,2}$, then $\dot{V}_{\sigma}(t) \le 0$ holds and $\|\tilde{\sigma}_{i,F}(t)\| \le \|\tilde{\sigma}_{i,F}(0)\| \le L_0$ can be derived. Since there is $-\|\tilde{\sigma}_{i,O}\| \le -(\frac{O_1}{L_0})\|\tilde{\sigma}_{i,F}\|$ Eq. (A.4) can be rewritten as

$$\dot{V}_{\sigma} \le -\sum_{i=1}^{N} (\Lambda_{i,O}(\Xi_1 + 1) \|\tilde{\sigma}_{i,O}\|_2^2)$$
$$\le -\sum_{i=1}^{N} \left( \Lambda_{i,O}(\Xi_1 + 1) \left( \frac{1}{2} \|\tilde{\sigma}_{i,O}\|_2^2 + \frac{O_1}{2L_0} \|\tilde{\sigma}_{i,F}\|_2^2 \right) \right)$$
$$\le -(\Xi_1 + 1) \Lambda_{\min} V_{\sigma}, \tag{A.8}$$

where $\Lambda_{\min} = \min\{\Lambda_{i,O}, \frac{O_1}{L_0} \Lambda_{i,O}\}$.

In conclusion, for any initial task error $\tilde{\sigma}_{i,O}(0)$ and $\tilde{\sigma}_{i,F}(0)$, if $\gamma_O \ge \max\{\gamma_{O,1}, \gamma_{O,2}\}$, there exists a predetermined time $T_f = t_{f1} + t_{f2}$ such that when $t > T_f$, the formation task error and obstacle avoidance task error converge to the $O_1$ neighborhood. Then the mission stability certificate is completed. □

### *A.2. Proof the stability of controller*

In this section, we prove the stability of closed-loop control system.

Considering the control law of the unmanned aerial vehicle system as shown in Eq. (49), the velocity tracking error $\tilde{\mathbf{V}}_i$ of UAVs can converge to a neighborhood close to 0 within a predefined time.

**Proof.** The following Lyapunov function is chosen as

$$V_\tau = \sum_{i=1}^{N}\left(\frac{1}{2}\tilde{\mathbf{V}}_i^T\tilde{\mathbf{V}}_i + \frac{1}{2}\tilde{\Psi}_i^T\Gamma_{\eta i}\tilde{\Psi}_i + \frac{1}{2}\Upsilon_i\tilde{\delta}_i^2\right). \tag{A.9}$$

First, prove the boundedness of $\tilde{\mathbf{V}}_i$, $\tilde{\Psi}_i$, and $\tilde{\Upsilon}_i$, the derivative of the Lyapunov function $V_\tau$ with respect to time can be derived as follows:

$$\begin{aligned}
\dot{V}_\tau &= \sum_{i=1}^{N}(\tilde{\mathbf{V}}_i^T\dot{\tilde{\mathbf{V}}}_i + \tilde{\Psi}_i^T\Gamma_{\eta i}\dot{\tilde{\Psi}}_i + \Upsilon_i\tilde{\delta}_i\dot{\tilde{\delta}}_i) \\
&= -\sum_{i=1}^{N}\tilde{\mathbf{V}}_i^T[\mathbf{J}^{-1}\mathbf{C}\tilde{\mathbf{V}}_i + (\Xi_3+1)\tilde{\mathbf{V}}_i \\
&\quad + \Upsilon_i(\mathbf{X}_i,\dot{\mathbf{X}}_i,\mathbf{V}_{r,i},\dot{\mathbf{V}}_{r,i})\tilde{\Psi}_i + \hat{\delta}_i SGN(\tilde{\mathbf{V}}_i) - {}_d] \\
&\quad - \sum_{i=1}^{N}\tilde{\Psi}_i^T\Gamma_{\eta i}\dot{\hat{\Psi}}_i - \sum_{i=1}^{N}\Upsilon_i\tilde{\delta}_i\dot{\hat{\delta}}_i \\
&\le -\sum_{i=1}^{N}\tilde{\mathbf{V}}_i^T[(\Xi_3+1)\tilde{\mathbf{V}}_i + \Upsilon_i(\mathbf{X}_i,\dot{\mathbf{X}}_i,\mathbf{V}_{r,i},\dot{\mathbf{V}}_{r,i})\tilde{\Psi}_i \\
&\quad + \hat{\delta}_i SGN(\tilde{\mathbf{V}}_i) - {}_d] - \sum_{i=1}^{N}\tilde{\delta}_i\|\tilde{\mathbf{V}}_i\|_1 \\
&\quad + \sum_{i=1}^{N}\tilde{\Psi}_i^T\Upsilon_i(\mathbf{X}_i,\dot{\mathbf{X}}_i,\mathbf{V}_{r,i},\dot{\mathbf{V}}_{r,i})\tilde{\mathbf{V}}_i \\
&\le -\sum_{i=1}^{N}\tilde{\mathbf{V}}_i^T(\Xi_3+1)\tilde{\mathbf{V}}_i - \sum_{i=1}^{N}\hat{\delta}_i\|\tilde{\mathbf{V}}_i\|_1 + \sum_{i=1}^{N}\delta_i\|\tilde{\mathbf{V}}_i\|_1 \\
&\quad - \sum_{i=1}^{N}\tilde{\delta}_i\|\tilde{\mathbf{V}}_i\|_1 \le -\sum_{i=1}^{N}\tilde{\mathbf{V}}_i^T(\Xi_3+1)\tilde{\mathbf{V}}_i \le 0.
\end{aligned} \tag{A.10}$$

According to the boundedness theorem in [42], it is easy to obtain $\tilde{\mathbf{V}}_i$, $\tilde{\Psi}_i$, and $\tilde{\Upsilon}_i$ are bounded. Assuming the existence of constant $\varepsilon_i, \zeta_i > 0$ satisfies $\|\tilde{\Psi}_i\| \le \varepsilon_i$, $\|\tilde{\delta}_i\| \le \zeta_i$.

Second prove the predefined-time convergence of $\tilde{\mathbf{V}}_i$. It is easy to get that $\mathbf{X}_i, \dot{\mathbf{X}}_i, \mathbf{V}_{r,i}, \dot{\mathbf{V}}_{r,i}$ is bounded and $\Upsilon_i(\mathbf{X}_i,\dot{\mathbf{X}}_i,\mathbf{V}_{r,i},\dot{\mathbf{V}}_{r,i})$ is bounded too. There exists a positive constant $\xi_i > 0$ satisfying $\|\Upsilon_i(\mathbf{X}_i,\dot{\mathbf{X}}_i,\mathbf{V}_{r,i},\dot{\mathbf{V}}_{r,i})\| \le \xi_i$.

Define a new Lyapunov function $V_{\mathbf{V}} = \sum_{i=1}^{N}\frac{1}{2}\tilde{\mathbf{V}}_i^T\tilde{\mathbf{V}}_i$ and the time derivative is as follows:

$$\begin{aligned}
V_{\mathbf{V}} &= -\sum_{i=1}^{N}\tilde{\mathbf{V}}_i^T[\mathbf{J}^{-1}\mathbf{C}\tilde{\mathbf{V}}_i + (\Xi_3+1)\tilde{\mathbf{V}}_i \\
&\quad + \Upsilon_i(\mathbf{X}_i,\dot{\mathbf{X}}_i,\mathbf{V}_{r,i},\dot{\mathbf{V}}_{r,i})\tilde{\Psi}_i + \hat{\delta}_i SGN(\tilde{\mathbf{V}}_i) - {}_d] \\
&\le -\sum_{i=1}^{N}\tilde{\mathbf{V}}_i^T[(\Xi_3+1)\tilde{\mathbf{V}}_i + \Upsilon_i(\mathbf{X}_i,\dot{\mathbf{X}}_i,\mathbf{V}_{r,i},\dot{\mathbf{V}}_{r,i})\tilde{\Psi}_i \\
&\quad + \hat{\delta}_i SGN(\tilde{\mathbf{V}}_i) - {}_d] \\
&\le -\sum_{i=1}^{N}\tilde{\mathbf{V}}_i^T(\Xi_3+1)\tilde{\mathbf{V}}_i + \sum_{i=1}^{N}\Delta\left\|\tilde{\mathbf{V}}_i\right\|_1 \\
&\le -\Xi_3 V_{\mathbf{V}} + \sum_{i=1}^{N}\Delta\left\|\tilde{\mathbf{V}}_i\right\|_1,
\end{aligned} \tag{A.11}$$

where $\Delta = \max\{\zeta_i + \varepsilon_i\xi_i\}$. Since $\tilde{\mathbf{V}}_i$ is bounded and $\sum_{i=1}^{N}\Delta \times \|\tilde{\mathbf{V}}_i\|_1$ is a positive number with an upper bound. Therefore the velocity tracking error $\tilde{\mathbf{V}}_i$ can converge to the neighborhood of 0 within a predefined time $t_{f3}$. Then the stability of controller certificate is completed. □


## Acknowledgments

This work was partially supported by the Aeronautical Science Foundation of China under grant numbers 20230001144001, the National Key Lab of Autonomous Intelligent Unmanned Systems (Beijing Institute of Technology) under the opening project number ZZKF2024-3-3, the National Key Laboratory of Space Intelligent Control Technology under grant HTKJ2023KL502008 and the Open Project Program of the Key Laboratory for Agricultural Machinery Intelligent Control and Manufacturing of Fujian Education Institutions under grant AMICM202301.



## ORCID

Jie Huang
https://orcid.org/0000-0001-7346-5034
Junjie Li
https://orcid.org/0009-0008-8849-897X
Zhengyi Zhang
https://orcid.org/0000-0001-8581-879X
Tingting Zhao
https://orcid.org/0009-0008-9022-0520
Simcha Vos
https://orcid.org/0000-0003-1672-4110

## References


[1] B. Alzahrani, O. S. Oubbati, A. Barnawi, M. Atiquzzaman and D. Alghazzawi, UAV assistance paradigm: State-of-the-art in applications and challenges, *J. Netw. Comput. Appl.* **166** (2020) 102706–102750.

[2] Y. Zeng, Q. Wu and R. Zhang, Accessing from the sky: A tutorial on UAV communications for 5G and beyond, *Proc. IEEE* **107**(12) (2019) 2327–2375.

[3] L. Chen, M. Cao and C. Li, Angle rigidity and its usage to stabilize multiagent formations in 2-D, *IEEE Trans. Autom. Control* **66**(8) (2020) 3667–3681.

[4] M. B. Sial, Y. Zhang, S. Wang, S. Ali, X. Wang, X. Yang, Z. Liao and Z. Yang, Bearing-based distributed formation control of unmanned aerial vehicle swarm by quaternion-based attitude synchronization in three-dimensional space, *Drones* **6**(9) (2022) 227–246.

[5] P. Li, J. Cao and D. Liang, UAV-BS formation control method based on loose coupling structure, *IEEE Access* **10** (2022) 88330–88339.

[6] H. Wang, J. Zhang, Q. Zhou, H.-J. Sun and Y.-Y. Wu, Optimal attitude consensus control for rigid spacecraft formation based on control lyapunov functions, *IET Control Theory Appl.* **17**(12) (2023) 1720–1731.

[7] H.-J. Sun, Y.-Y. Wu and J. Zhang, A distributed predefined-time attitude coordination control scheme for multiple rigid spacecraft, *Aerosp. Sci. Technol.* **133** (2023) 108134–108150.

[8] L. Chen and M. Cao, Angle rigidity for multiagent formations in 3-d, *IEEE Trans. Autom. Control* **68**(10) (2023) 6130–6145.

[9] J. F. Flores-Resendiz, D. Avilés and E. Aranda-Bricaire, Formation control for second-order multi-agent systems with collision avoidance, *Machines* **11**(2) (2023) 208–226.

[10] J. Lin, Z. Miao, Y. Wang, G. Hu and R. Fierro, Aggressive formation tracking for multiple quadrotors without velocity measurements over directed topologies, *IEEE Trans. Aerosp. Electron. Syst.* **59**(5) (2023) 5541–5553.

[11] J. Guo, J. Qi, M. Wang, C. Wu, Y. Ping, S. Li and J. Jin, Distributed cooperative obstacle avoidance and formation reconfiguration for multiple quadrotors: Theory and experiment, *Aerosp. Sci. Technol.* **136** (2023) 108218–108233.

[12] X. Bai, W. Yan, S. S. Ge and M. Cao, An integrated multi-population genetic algorithm for multi-vehicle task assignment in a drift field, *Inf. Sci.* **453** (2018) 227–238.

[13] X. Bai, W. Yan, M. Cao and D. Xue, Distributed multi-vehicle task assignment in a time-invariant drift field with obstacles, *IET Control Theory Appl.* **13**(17) (2019) 2886–2893.

[14] G. Xu, X. Kang, H. Yang, Y. Wu, W. Liu, J. Cao and Y. Liu, Distributed multi-vehicle task assignment and motion planning in dense environments, *IEEE Trans. Autom. Sci. Eng.* **21**(4) (2023) 7027–7039.

[15] S. Chen, M. Wang and W. Song, Hierarchical learning with heuristic guidance for multi-task assignment and distributed planning in interactive scenarios, *IEEE Trans. Intell. Veh.* (2024) 1–13.

[16] C.-B. Zheng, Z.-H. Pang, J.-X. Wang, J. Sun, G.-P. Liu and Q.-L. Han, Null-space-based time-varying formation control of uncertain nonlinear second-order multi-agent systems with collision avoidance, *IEEE Trans. Ind. Electron.* **70**(10) (2022) 10476–10485.

[17] G. Golluccio, P. Di Lillo, A. Marino and G. Antonelli, When local optimization is bad: Learning what to (not) maximize in the null-space for redundant robot control, in *Proc. 9th Int. Conf. Control, Decision and Information Technologies (CoDIT)* (IEEE, 2023), pp. 506–511.

[18] C. Wang, Z. Wei, W. Jiang, H. Jiang and Z. Feng, Cooperative sensing enhanced UAV path-following and obstacle avoidance with variable formation, *IEEE Trans. Veh. Technol.* **73**(6) (2024) 7501–7516.

[19] Z. Zhang, J. Chen, Z. Mo, Y. Chen and J. Huang, A behavior-based adaptive dynamic programming method for multiple mobile manipulators coordination control, *Int. J. Control Autom. Syst.* **21**(9) (2023) 3022–3035.

[20] N. Zhou and Y. Xia, Coordination control design for formation reconfiguration of multiple spacecraft, *IET Control Theory Appl.* **9**(15) (2015) 2222–2231.

[21] G. Liu, J. Zhang and C. Li, A leader-following formation control bearing-based approach, in *Proc. Chinese Automation Congress (CAC)* (IEEE, 2018), pp. 4075–4080.

[22] Z. Xiao, X. Fan and X. Yang, Bearing measurement based formation maneuver control, in *Proc. 40th Chinese Control Conf. (CCC)* (IEEE, 2021), pp. 4966–4971.

[23] L. Chen, Q. Yang, M. Shi, Y. Li and M. Feroskhan, Stabilizing angle rigid formations with prescribed orientation and scale, *IEEE Trans. Ind. Electron.* **69**(11) (2021) 11654–11664.

[24] Y. Niu, H. Ban, H. Zhang, W. Gong and F. Yu, Nonsingular terminal sliding mode based finite-time dynamic surface control for a quadrotor UAV, *Algorithms* **14**(11) (2021) 315–333.

[25] A. Polyakov, Nonlinear feedback design for fixed-time stabilization of linear control systems, *IEEE Trans. Autom. Control* **57**(8) (2011) 2106–2110.

[26] J. D. Sánchez-Torres, D. Gómez-Gutiérrez, E. López and A. G. Loukianov, A class of predefined-time stable dynamical systems, *IMA J. Math. Control Inf.* **35** (2018) i1–i29.

[27] Q. Li, Y. Chen and K. Liang, Predefined-time formation control of the quadrotor-UAV cluster position system, *Appl. Math. Model.* **116** (2023) 45–64.

[28] J. Chen, Z. Chen, H. Zhang, B. Xiao and L. Cao, Predefined-time observer-based nonsingular sliding mode control for spacecraft attitude stabilization, *IEEE Trans. Circuits Syst. II* **71**(3) (2023) 1291–1295.

[29] X. Wu, S. Ding, B. Niu, N. Xu and X. Zhao, Predefined-time event-triggered adaptive tracking control for strict-feedback nonlinear systems with full-state constraints, *Int. J. Gen. Syst.* **53**(3) (2024) 352–380.

[30] B. Ning, Q.-L. Han and Z. Zuo, Practical fixed-time consensus for integrator-type multi-agent systems: A time base generator approach, *Automatica* **105** (2019) 406–414.

[31] N. Zhou, X. Cheng, Z. Sun and Y. Xia, Fixed-time cooperative behavioral control for networked autonomous agents with second-order nonlinear dynamics, *IEEE Trans. Cybern.* **52**(9) (2021) 9504–9518.

[32] M. C. P. Santos, C. D. Rosales, M. Sarcinelli-Filho and R. Carelli, A novel null-space-based UAV trajectory tracking controller with collision avoidance, *IEEE/ASME Trans. Mechatron.* **22**(6) (2017) 2543–2553.

[33] J. Baek and M. Kang, A synthesized sliding-mode control for attitude trajectory tracking of quadrotor UAV systems, *IEEE/ASME Trans. Mechatron.* **28**(4) (2023) 2189–2199.

[34] J. Huang, N. Zhou and M. Cao, Adaptive fuzzy behavioral control of second-order autonomous agents with prioritized missions: Theory and experiments, *IEEE Trans. Ind. Electron.* **66**(12) (2019) 9612–9622.

[35] J. Chen, M. Gan, J. Huang, L. Dou and H. Fang, Formation control of multiple euler-lagrange systems via null-space-based behavioral control, *Sci. China Inf. Sci.* **59**(1) (2016) 1–11.

[36] G. Antonelli, F. Arrichiello and S. Chiaverini, Experiments of formation control with multirobot systems using the null-space-based behavioral control, *IEEE Trans. Control Syst. Technol.* **17**(5) (2009) 1173–1182.

[37] B. Ning, Q.-L. Han, Z. Zuo, L. Ding, Q. Lu and X. Ge, Fixed-time and prescribed-time consensus control of multiagent systems and its applications: A survey of recent trends and methodologies, *IEEE Trans. Ind. Inf.* **19**(2) (2022) 1121–1135.

[38] Z. Guo, J. Zhang, Y. Shang, Y. Zhang, L. Zhang and W. Chen, Predefined-time global recursive sliding mode control for trajectory tracking of unmanned surface vehicles with disturbances uncertainties, *Ocean Eng.* **313** (2024) 119408–119421.

[39] X. Peng and S. Zhou, Finite-time stochastic control for complex dynamical systems: The estimate for control time and energy consumption, *Syst. Control Lett.* **184** (2024) 105711–105720.
[40] Y.-Z. Sun, S.-Y. Leng, Y.-C. Lai, C. Grebogi and W. Lin, Closed-loop control of complex networks: A trade-off between time and energy, *Phys. Rev. Lett.* **119**(19) (2017) 198301–198308.
[41] C. Ott, A. Dietrich and A. Albu-Schäffer, Prioritized multi-task compliance control of redundant manipulators, *Automatica* **53** (2015) 416–423.
[42] H. K. Khalil, *Control of Nonlinear Systems* (Prentice Hall, 2002).

**Jie Huang** received his B.E. degree in Electrical Engineering and Automation, M.E. degree in Control Engineering from the Fuzhou University in 2005 and 2010, respectively, and Ph.D. degree in Control Science and Engineering from Beijing Institute of Technology in 2015. From 2005 to 2015, he was Lecturer with Fujian Institute of Education. From 2014 to 2016, he was a postdoctoral researcher with the Faculty of Mathematics and Natural Sciences, University of Groningen, the Netherlands. From 2016 to 2018, he was a docent with the Faculty of Science and Engineering, University of Groningen, the Netherlands. He is currently Full Professor of Robotic and Control with the College of Electrical Engineering and Automation, Fuzhou University. He is Vice-president of the Fujian Automation Association, Fujian Province, China. His research interests include autonomous robots, complex network dynamics, multi-agent systems and industrial Internet.

**Junjie Li** received his B.E. degree in the Nanjing University of Information Science and Technology in 2022. He is pursuing M.E. degree in the Fuzhou University. His research interests include path tracking and formation control for UAVs.

**Zhengyi Zhang** ATTAINED his B.E. degree and M.E. degree in Mechanical Engineering from the Zhejiang Sci-Tech University in 2016 and 2019, respectively. He completed his Ph.D. in Electrical Engineering at the Fuzhou University in 2024. Presently, Zhang is Lecturer at the School of Automation and Electrical Engineering, Zhejiang University of Science and Technology. His research interests include intelligent robot ethology and multi-agent learning systems.

**Tingting Zhao** received her M.E. degree in Mathematics from the College of Mathematics and System Sciences, Xinjiang University, Ürümqi, China, in 2022. She is currently pursuing her Ph.D. degree in Electrical Engineering and Automation with Fuzhou University, Fuzhou. Her current research interests include multilayer networks, networked control systems, and distributed control.

**Simcha Vos** received the Bachelor of Computer Science and Engineering from TU Delft in 2022. He is pursuing his M.E. degree in the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, NL.