## Highlights

**Distributed sliding-mode cloud predictive formation control of networked multi-agent systems with application to air-bearing spacecraft simulators**

Chuanguo Chi, Dong Lin, Clara-Mihaela Ionescu, Guo-Ping Liu*

- The proposed SMCPC scheme combined the advantage of SMC and CPC.
- Designed multi-step state predictor for NCS delay compensation based on SMC.
- SMCPC ensures desired performance and feasibility in multi-agent formation control.
- Control scheme verified by practical ABSs formation control experiments.

# Distributed sliding-mode cloud predictive formation control of networked multi-agent systems with application to air-bearing spacecraft simulators ☆

Chuanguo Chi [a,d], Dong Lin [a,c], Clara-Mihaela Ionescu [d,e], Guo-Ping Liu [a,b,*]

[a] Center for Control Theory and Guidance Technology, Harbin Institute of Technology, Harbin 150001, China
[b] Center for Control Science and Technology, Southern University of Science and Technology, Shenzhen 518055, China
[c] State Grid Fujian Electric Power Research Institute, Fuzhou 350007, China
[d] Research Group on Dynamical Systems and Control, Department of Electromechanical, Systems and Metal Engineering, Gent University, 9052 Gent, Belgium
[e] Flanders Make Consortium, core lab MIRO, Tech Lane Science Park 131, Gent, Belgium

ARTICLE INFO

*Keywords:*
Distributed cloud predictive control
Sliding-mode control
Networked multiagent systems
Formation control
Control of air-bearing spacecraft simulator

ABSTRACT

This paper investigates the application of distributed sliding-mode cloud predictive control scheme in networked multi-agent control systems (NMACS). A sliding-mode cloud predictive control (SMCPC) scheme based on the cloud control system is designed. Combining the advantages of variable structure control (VSC) of sliding-mode control(SMC) and the advantages of cloud predictive control (CPC) in handling network delays, the network delay is actively compensated while ensuring the stability and output consistency of the NMACS. Also taking advantage of the small online computation of SMC, reduces the burden of CPC in processing big data and increases the computation rate. In addition, the scheme includes multi-step state prediction and output prediction as well as coordinated optimization among multiple agents. By optimizing the cost function design, the SMCPC scheme is guaranteed to achieve the desired control performance in multi-agent formation control. And the feasibility of the control scheme is verified by stability and consistency analysis. The applicability of the control scheme in practical applications is verified by air-bearing spacecraft simulator (ABS) formation control experiments.

## 1. Introduction

With the rapid development of information technology (IT) and artificial intelligence (AI) in recent years, the research on networked control system (NCS) and multi-agent control system (MACS) has become increasingly more advanced and expanded (Oh, Park, & Ahn, 2015; Qin, Ma, Shi, & Wang, 2016; Zhang, Han, Ge, Ding, Ding, Yue, & Peng, 2019; Zhang, Shi, Wang, & Yu, 2017). The controlled objects and control networks are also becoming more intelligent. This leads to many challenges for traditional local control strategies and centralized control schemes in solving advanced control problems (Yu, Liu, & Hu, 2022b). The development of the Internet of Things (IoT) and cloud computing provides a good solution to this situation (Liu, 2017). Increasing transmission rates of the internet and cloud computing capabilities provide an efficient and stable distributed control platform for (MACS). As the rapid development of today's smart home, driverless, smart grid and telemedicine, etc. Some complex MAC and various different intelligent objects are connected through the network to realize the cloud control (Pang, Zheng, Li, Liu & Han, 2021; Yu, Liu, & Hu, 2022a). With the powerful computing ability of the cloud platform, it greatly reduces the user's requirement for local hardware, while making remote control become possible (Sasaki, Suzuki, Makido, & Nakao, 2016; Shengdong, Zhengxian, & Yixiang, 2019). Cloud control and IoT have been closely connected to our lives and their applications have penetrated into all industries (Sang & Xu, 2017). This not only drives economic development, but also promotes the advancement of technology and industry. There is no doubt that it will bring a new information revolution, and the ability to integrate these new technologies will be of great strategic importance for industry 5.0 (Maxim, Copot, Copot, & Ionescu, 2019).

As networked multi-agent control (NMAC) technologies are increasingly utilized in science and engineering (Knorn, Chen, & Middleton, 2015), the operating environments, communication networks, dynamic characteristics, and task objectives of the agents are also becoming increasingly complex (Ning, Han, & Ding, 2021; Zhao, Hua, & Guan, 2019). In the complex MACSs, such as communication satellite, smart grid, intelligent robot, UAV (Huang, Liu, & Hu, 2022) etc., each agent

☆ This work was supported in part by the National Natural Science Foundation of China under Grant 62173255, 62188101 and Chinese Scholarship Council (CSC) under grant 202106120085.

* Corresponding author at: Center for Control Theory and Guidance Technology, Harbin Institute of Technology, Harbin 150001, China.
*E-mail addresses:* 18B904028@stu.hit.edu.cn (C. Chi), lin_dong2023@163.com (D. Lin), claramihaela.ionescu@ugent.be (C.-M. Ionescu), liugp@sustech.edu.cn (G.-P. Liu).

has its own characteristics, which also leads to the singularity and limitation when they perform tasks (Tan, Yue, Li & Liu, 2020). Networked control realizes the interconnection of multi-agent through common interaction protocol, which makes the communication among multiple agents of the same or different categories work together (Ge, Yang, & Han, 2017; Zhang, Han, & Yu, 2015). It enlarges the application scenario of agents, solves the limitation of single task target of single agent, and improves the stability of agents. In the networked multi-agent control system (NMACS), communication delay is inevitable due to the existence of remote communication (Pang, Luo, Liu, & Han, 2020). Therefore, predictive control method has been extended to various fields involving NMACS to solve the problem of time delays (Tan, Wang, Wu, Huang, & Miao, 2021). In order to solve the huge amount of computation in NMACS and the big data burden of real-time communication, cloud computing is introduced into the control system. By designing a CPC scheme that synchronizes big data processing to the cloud, this reduces the workload of the user's hardware system (Chi, Liu, & Hu, 2022). Many researches on cloud control systems (CCS) are well-known and the applications have also covered the industrial field (Xia, 2015). In the literature (Liu, 2020) professor Liu innovatively applied CPC to NMACS. CPC as a branch of CCS, the existing research mainly focuses on cloud computing based predictive control (Liu, 2022) and data-based predictive control (Tan, Miao, Wang, Wu & Huang, 2020; Xia, Xie, Liu, & Wang, 2013). Through theoretical studies and simulation experiments of CPC has achieved great improvements in dealing with communication constraints, it still presents many challenges in practical engineering applications. Such as large computational effort, slow processing of real-time big data information, and poor synchronization of control protocol parameters.

In order to solve the above problems, this paper combines the advantages of distributed CPC and SMC to study the distributed formation control problem of NMACS. When dealing with practical control problems, it has become a common choice to combine the advantages of different control algorithms to achieve control objectives. Regarding the integration of SMC and predictive control advantages, the SMC+MPC control scheme has been studied. Ferrara, Incremona et al. investigated a model-based control scheme based on a combination of model predictive control (MPC) and integral sliding mode (ISM) control (Ferrara, Incremona, & Magni, 2013, 2014; Incremona, Ferrara, & Magni, 2017). Fesharaki designed a MPC+SMC scheme for the nominal system to recover the nominal performance of the uncertain system while ensuring the system achieves the desired performance (Fesharaki, Kamali, Sheikholeslam, & Talebi, 2020). Xu proposed a novel digital integral terminal sliding mode predictive control scheme based on output (Xu, 2015). Garcia–Gabin combining SMC design techniques with MPC for air-conditioning solar power plants (Garcia-Gabin, Zambrano, & Camacho, 2009). Shi proposed a new terminal unconstrained MPC scheme by embedding an implicit SMC law in the designed terminal cost function (Ji, Ren, Liu, & Shi, 2021). Above works on MPC+SMC, the advantages of MPC and SMC are combined to improve the robustness of the system.

Based on the advantage that SMC requires less online computing (Ionescu & Muresan, 2015), the introduction of SMCPC can further reduce the computational load of the system, improve the computational speed and achieve fast and accurate tracking effects by combining the advantages of two advanced control methods. Furthermore, due to the introduction of computers and microprocessors in networked control, the main goal of designing control systems has focused on the study of discrete-time control. In the design of control methods, there usually exists some discrepancy between the actual device and its corresponding mathematical dynamics model, which leads to the uncertainties of the controlled system. Predictive control has great challenges in dealing with uncertainties and model errors, and how to ensure the robustness of the closed-loop system in this case is a difficult research area (Li, Yang, Sun, & Xia, 2014). As SMC is a variable structure control strategy (Steinberger, Horn, & Fridman, 2020), the system structure is not fixed, so it can be changed according to the current state of the system during the dynamic process to make the system move according to the predefined "sliding mode" state trajectory. SMCPC can obtain the optimal control even when the system model is not very accurate or there are disturbances, and ensure the robustness of the control system.

The principal contributions of this paper are listed below.

(1) A distributed SMCPC control scheme is designed, where SMC ensures the stability and robustness of the system and CPC deals with the time delay existing in the NCS. Combining the advantages of the two control methods realizes the coordinated control of the NMACS and compensates the communication delay between the cloud nodes.
(2) In order to actively compensate the delay existing in the NCS, a multi-step state predictor based on SMC is designed, and based on the state prediction, the output vector, the sliding mode function vector and the ABS velocity vector are predicted, respectively.
(3) Through the optimized design, the desired control performance of SMCPC in multi-agent formation control is ensured. And the feasibility of the control scheme is verified by stability and consistency analysis.
(4) Finally, the validity of the control scheme in practical applications is verified by ABSs formation control experiments.

## 2. Problem description

### 2.1. NMACS via distributed SMCPC

NMACS is composed of multiple agents interacting through a network to achieve one or more common goals. Due to the large number of agents, complex communication network, heterogeneity of agents, unstable dynamic characteristics of the system and other factors, NMACS may be very complex (Cao, Liu, & Zhang, 2023; Pang, Bai, Liu, Han & Zhang, 2021). At the same time, with the increase of equipment in NCS comes a heavy hardware and data processing burden. Distributed CCS can reduce the over-reliance on local hardware devices when dealing with the problem of NCS. By distributing the analysis and processing of data to different clouds, the control of large-scale complex networked systems becomes more convenient. This paper discusses the CCS with different cloud nodes interconnected for distributed control as shown in Fig. 1. In the proposed distributed CCS, each cloud controller controls an agent independently and communicates with other cloud controllers through the network. Different cloud controllers may exist in the same cloud node or different cloud nodes, which inevitably causes a series of problems in the system, such as uncertainty in network communication, packet loss, external interference, time delay, and so on. How to improve the stability and processing efficiency of the system has become a problem that must be solved in the networked multi-agent control system. Based on the above-mentioned problems, this paper introduces SMC into distributed networked multi-agent control system. A SMCPC control scheme is proposed, this scheme is a control method based on SMC and distributed CCS. The proposed SMCPC has the following advantages: (1) It can realize multi-agent cooperative control, coordinate the control among multiple agents through distributed cloud controller, and resist the influence of external interference and system parameter changes while ensuring the control accuracy, so that the whole system has better robustness and coordination. (2) Control tasks can be assigned to the cloud node for computing, making full use of the efficient performance and storage capacity of the cloud platform, and improving the computing efficiency and response speed. (3) The real-time monitoring and remote operation of the system status can be realized through the monitoring platform based on the cloud control system, which makes the maintenance and management of the system more convenient and efficient. (4) SMCPC can be applied to the control of nonlinear systems because both sliding mode control and predictive

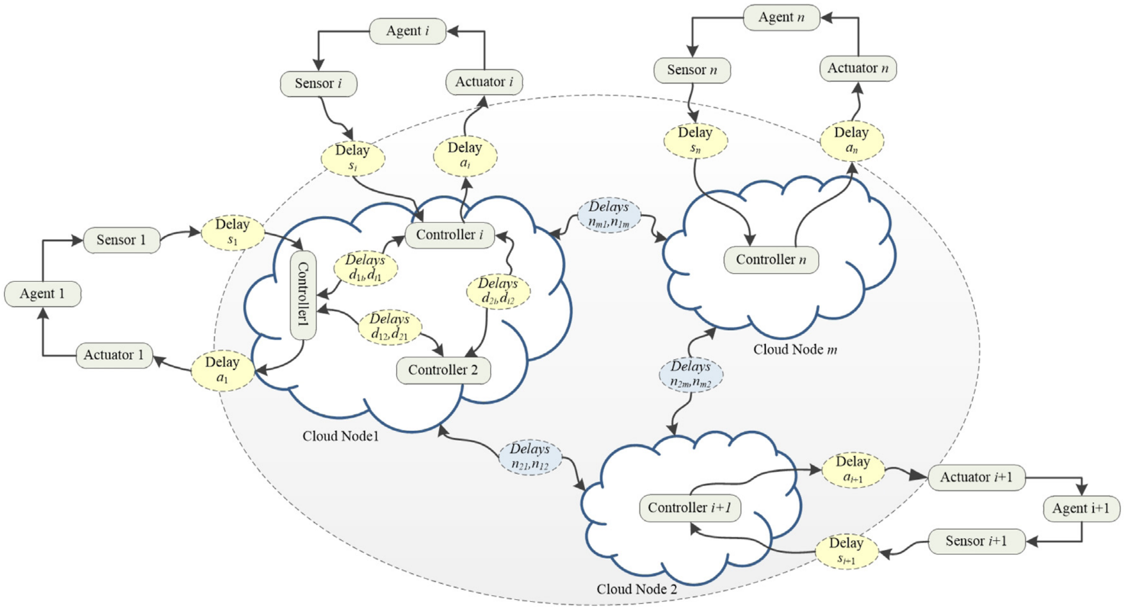


**Fig. 1.** Topology structure of distributed cloud control system.

control can deal with nonlinear systems. At the same time, constraints, such as control input and system status, can be considered to ensure that the system will not exceed the limits during operation. In short, SMCPC has the advantages of multi-agent collaborative control, strong robustness, efficient utilization of computing resources, real-time monitoring and remote operation, applicable to nonlinear systems, handle constraints. It also and can be applied to complex industrial control systems and intelligent manufacturing and other fields. This paper will carry out detailed design and research on SMCPC scheme through formation control.

### 2.2. Formation control

Networked multi-agent formation control is a technology that realizes multi-agent cooperation to complete a task through mutual communication and cooperation. In this technology, each agent can sense the surrounding environment and communicate with other agents to complete the task cooperatively (Zheng, Pang, Wang, Sun, Liu, & Han, 2022). In networked multi-agent formation control, agents can communicate through different protocols, such as distributed algorithms and game theory. Distributed algorithm is an algorithm based on local information and cooperation to achieve global goals, while game theory is a mathematical model used to analyze the interaction between agents (Ning, Han, & Zuo, 2019a, 2019b). The networked multi-agent formation control technology can be applied to many fields, such as UAV formation control, robot formation control, spacecraft formation control, etc. In this paper, the SMCPC control scheme is designed to achieve formation control, which makes the agents more efficient, safe and reliable when they cooperate to complete tasks.

## 3. SMCPC scheme

### 3.1. Control objective

For a general distributed networked cloud control system in Fig. 1. Assuming there are $N$ controllers distributed in $M$ clouds. The $i$th controller controls the corresponding subsystems, and each subsystem composed of sensors, actuators, and $i$th agent $i \in N, N = \{1, 2 \cdots N\}$.

For the convenience of explanation, the following assumptions are made for each variable.

(1) $s_i$ represents the communication delay between the sensor of the $i$th agent and its controller, and $a_i$ represents the communication delay between the actuator of the $i$th agent and its controller;
(2) $d_{ij}$ represents intra-cloud delay, which means the communication delay between the $i$th agent and the $j$th agent located on the same cloud node; $n_{ij}$ means the inter-cloud delay, which means the communication delay between different cloud nodes; $c_{ij}$ represents the actual communication delay between the controller of $i$th agent and the $j$th agent;
(3) On the premise of fixed communication topology, each controller only communicates with its adjacent controller in the same cloud node or the controller in the adjacent cloud node. The communication delay between the $i$th and $j$th controllers on the same cloud node is equal to the intra cloud delay, i.e. $c_{ij} = d_{ij}$. The communication delay between $i$th and $j$th controllers on different cloud nodes is equal to the delay between two cloud nodes, i.e. $c_{ij} = n_{ij}$.
(4) All communication delays $s_i$, $a_i$, $n_{ij}$, $d_{ij}$ are bounded constant integers, which are integer multiples of the sampling period $T$;
(5) All agents are controllable and observable.

In order to simplify the design, analysis and performance description of SMCPC scheme to be introduced, here only considered a linear multi-agent system. However, as described in Section 2, this scheme is also applicable to NMAC with nonlinear dynamics, uncertainty and interference. Therefore, the following system with $N$ agents is considered:

$$\begin{aligned} x_i(k+1) &= A_i x_i(k) + B_i u_i(k) \\ y_i(k) &= C_i x_i(k) \end{aligned} \tag{1}$$

where $x_i \in \mathcal{R}^{n_i}, y_i \in \mathcal{R}^{l_i}$ and $u_i \in \mathcal{R}^{m_i}$ are the state, output and input vectors of the system respectively, $i \in N, N = \{1, 2, \ldots, N\}$. $A_i \in \mathcal{R}^{n_i \times n_i}$, $B_i \in \mathcal{R}^{n_i \times m_i}$ and $C_i \in \mathcal{R}^{l_i \times n_i}$ are the matrices of the system.

According to the state and state estimation information obtained by the $i$th agent at time $k$, the state $\hat{X}_i$ and the desired state $\hat{X}_{dr}$ of the next $N$ cycles are predicted. The cost function $J_i$ is constructed through the prediction information of the obtained agent and adjacent agents, and the predicted value $\hat{V}_i$ of the cooperative term vector is added to the cost function, which can measure the formation effect. The value of the cooperation term vector $\hat{V}_i$ needs to ensure the minimum value of the performance index function $J_i$ by calculating the partial derivative

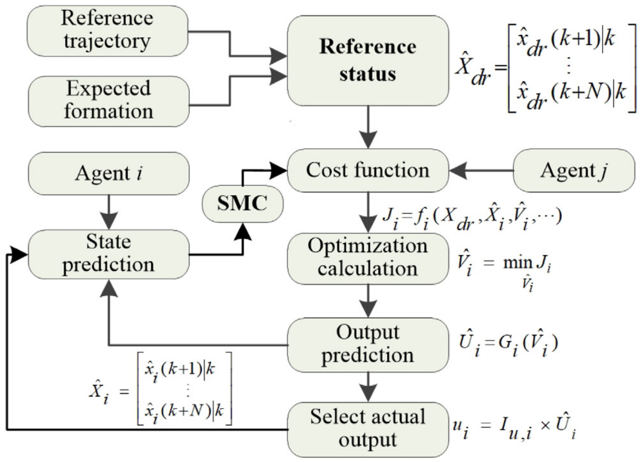


**Fig. 2.** The proposed distributed sliding-mode cloud predictive control scheme.

of $J_i$ with respect to $\hat{V}_i$ and making the partial derivative equal to 0, the value of $\hat{V}_i$ can be obtained. The predictive output vector $\hat{U}_i$ of the controller is a function of $\hat{V}_i$, from which $\hat{U}_i$ can be obtained. $\hat{U}_i$ is used in the state value prediction of the next cycle, and the predicted value of the cycle closest to the current cycle among all predicted values is taken as the actual control quantity of the system, that is, $\hat{u}_i = I_{u,i} \times \hat{U}_i$. The above solution steps are represented by the flowchart of Fig. 2. It should be noted that any shape such as $\hat{s}(k+i|k-j)$, the symbol of $(k+i|k-j)$ indicates that the value of signal $s$ at $k+i$ time is predicted based on $k-j$ time and all information beifore that time. In this article, a practical distributed networked sliding-mode cloud predictive controller will be designed for the mathematical model of ABS, and a detailed control design method will be given.

### 3.2. ABS dynamics

As shown in Fig. 3, three degrees of freedom (3-DOF) ABS was selected the research target, the control system contains three output signals, which are the $X$-axis coordinates, $Y$-axis coordinates, and attitude angle $\phi$, respectively. Fig. 4 shows the geometric architecture of the ABS, and the distribution of the thrusters. Where thrusters 1, 4 provide a force $f_{yb}$ along the $Y_b$ axis, thrusters 2, 3, 5, 6 provide a force $f_{xb}$ along the $X_b$ axis, and a torque $f_T$ for rotation about the $Z_b$ axis. Define the ABS inertial coordinate system (ICS) $O - XYZ$ and body-fixed coordinate system (BCS) $O_b - X_b Y_b Z_b$. In the ICS, the vector $y = (d_x, d_y, \phi)^T$ represents the position, attitude angle state of the ABS, and $v = (\dot{d}_x, \dot{d}_y, \dot{\phi})^T = (v_x, v_y, \omega)^T$ is the velocity and angular velocity. The velocity and angular velocity of the ABS in the ICS $v_b = (v_{xb}, v_{yb}, \omega)^T$. The thrust of the ABS in the ICS and BCS are $f = (f_x, f_y, f_T)^T$ and $f_b = (f_{xb}, f_{yb}, f_T)^T$ respectively. The dynamic equations of the ABS in the ICS are derived from Newton's second law.

$$\dot{v}_x = \frac{1}{m} f_x, \dot{v}_y = \frac{1}{m} f_y, \dot{\omega} = \frac{1}{J} f_T$$

where, $m$ and $J$ are the mass and rotational inertia of the ABS.

To allocate control inputs to the six thrusters, coordinate transformation rules are introduced. The mathematical conversion relations of the physical measurements in the BCS and ICS are as follows:

$$\begin{bmatrix} f_x & f_y & f_T \end{bmatrix}^T = R_b^i \begin{bmatrix} f_{xb} & f_{yb} & f_T \end{bmatrix}^T \tag{2}$$

where $R_i^b$ is the conversion matrix from the ICS to the BCS.

$$R_b^i = \begin{bmatrix} cos(\phi) & -sin(\phi) & 0 \\ sin(\phi) & cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

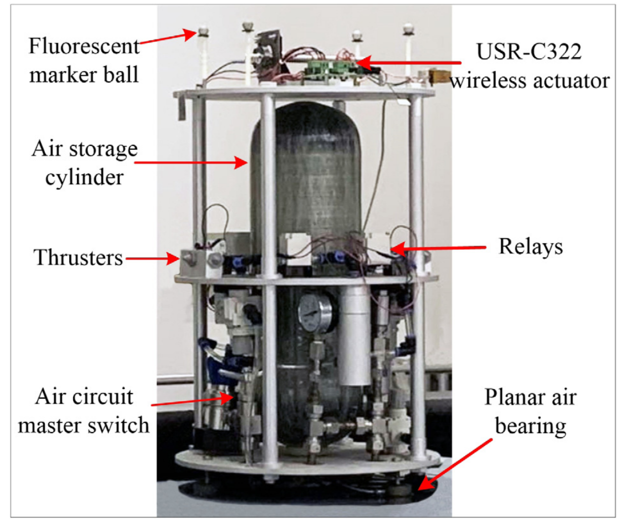


**Fig. 3.** Hardware of ABS.

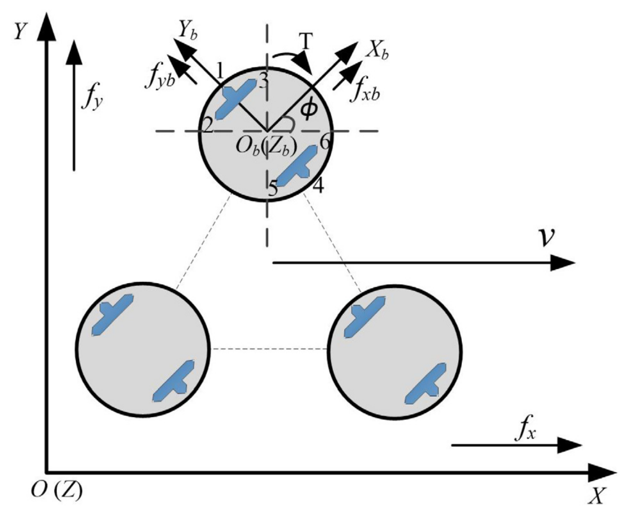


**Fig. 4.** Geometric architecture of ABSs.

### 3.3. SMCPC scheme design of ABS

In order to realize the control algorithm and simplify the control scheme design and performance analysis, the continuous mathematical model of the control object spacecraft simulator needs to be discretized. The discrete mathematical model of multi-agent system with spacecraft simulator as control object is obtained. The state space matrices $A_i, B_i, C_i$ are as follows:

$$A_i = \begin{bmatrix} 1 & 0 & 0 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 & T_s & 0 \\ 0 & 0 & 1 & 0 & 0 & T_s \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, B_i = \begin{bmatrix} \frac{T_s^2}{2m_i} & 0 & 0 \\ 0 & \frac{T_s^2}{2m_i} & 0 \\ 0 & 0 & \frac{T_s^2}{2J_i} \\ \frac{T_s}{m_i} & 0 & 0 \\ 0 & \frac{T_s}{m_i} & 0 \\ 0 & 0 & \frac{T_s}{J_i} \end{bmatrix}$$

$$C_i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, i \in N$$

For the research object spacecraft simulator, $x_i = [d_{xi}, d_{yi}, \phi_i, v_{xi}, v_{yi}, \omega_i]^{\mathrm{T}}$, $y_i = [d_{xi}, d_{yi}, \phi_i]^{\mathrm{T}}$, $u_i = [f_{xi}, f_{yi}, f_{T_i}]^{\mathrm{T}}$.

The design of the SMCPC for the ABS focuses on attitude and position control problems. It mainly includes two steps:

(1) Designing a stable sliding-mode surface to ensure that the system state enters the sliding-mode surface and gradually converges to the origin point.
(2) By using the approaching rule design control law to ensure that the system state can reach the sliding-mode surface in finite time, regardless of the initial state.

For the discrete system of Eq. (1), design the sliding mode surface as:

$$s(k) = ce(k) + \ddot{e}(k), c > 0 \tag{4}$$

Where $e(k)$ is the tracking error and $\ddot{e}(k)$ is the second derivative of the tracking error, $c$ represents the sliding surface parameter. Define a switching band surrounding the switching surface.

$$S^{\zeta} = \{x \in R^n | -\zeta < s(x) < +\zeta\} \tag{5}$$

The discrete system starts from any state, reaches the switching surface $s$ in finite steps, and then moves on it to reach the quasi-sliding mode. By designing the sliding surface, the system can achieve fast and stable control of the system state, thus enabling the system to achieve accurate tracking performance.

Considering the rapidity and smoothness of the sliding-mode controller and the output range of the controller, the exponentially approaching rule is used to design the control law. When the sampling time is $T$, the following discrete exponential reaching law can be obtained:

$$s(k+1) = (1 - qT)s(k) - \varepsilon T sat(s(k)) \tag{6}$$

where, $\varepsilon > 0, q > 0, 1 - qT > 0$, $sat(s(k))$ denotes a saturation function. In order to eliminate the chattering effect in the control of sliding-mode variable structures, the commonly used sign functions are replaced by saturation function.

$$sat(s(k)) = \begin{cases} -1, & s < -\zeta \\ ks, & |s| \leq \zeta, k = \frac{1}{\zeta} \\ 1, & s > \zeta \end{cases} \tag{7}$$

where $\zeta > 0$ indicates switching bands. Replacing sign function with saturation function can eliminate chattering to some extent, but at the same time, it sacrifices robustness. Therefore, parameter $\zeta$ should be adjusted reasonably in the experiment to achieve the optimal effect.

Further, we can obtain the sliding surface of the attitude sliding mode controller for ABS:

$$s_{\phi}(k) = c_{\phi} e_{\phi}(k) + e_{\omega}(k), c_{\phi} > 0 \tag{8}$$

where, $e_{\phi} = \phi - \phi_d$ indicates the attitude angle error, $e_{\omega} = \omega - \omega_d$ indicates the angular velocity error, $c_{\phi} > 0$ is the sliding surface parameter, $\phi_d$ and $\omega_d$ are the desired attitude angle and desired angular velocity, respectively. When the system is in the sliding-mode, $s_{\phi} = 0$, the calculation yields:

$$e_{\phi} = e_{\phi}(0) e^{-c_{\phi} t} \tag{9}$$

This makes the attitude angle stable and the attitude error satisfies the exponential convergence.

From (8) the mathematical relationship between $s_{\phi}(k+1)$ and the input torque $f_T$ can be obtained:

$$\begin{aligned} s_{\phi}(k+1) =& c_{\phi} \dot{e}_{\phi}(k+1) + \dot{e}_{\omega}(k+1) \\ =& c_{\phi} e_{\omega}(k+1) + \dot{\omega}(k+1) - \dot{\omega}_d(k+1) \\ =& c_{\phi} e_{\omega}(k+1) + \frac{f_T(k+1)}{J} - \dot{\omega}_d(k+1) \end{aligned} \tag{10}$$

Then, from (6) and (10), the attitude angle control law can be derived as

$$\begin{aligned} f_T(k+1) =& J \Big( \dot{\omega}_d(k+1) - c_{\phi} e_{\omega}(k+1) + (1 - q_{\phi}T) s_{\phi}(k) \\ & - \varepsilon_{\phi} T sat\big(s_{\phi}(k)\big) \Big) \end{aligned} \tag{11}$$

Similarly, the position control rate can be obtained:

$$\begin{aligned} f_x(k+1) =& \ m \Big( \dot{v}_{xd}(k+1) - c_x e_{v_x}(k+1) + (1 - q_x T) s_x(k) \\ & - \varepsilon_x T san\big(s_x(k)\big) \Big) \end{aligned} \tag{12}$$

$$\begin{aligned} f_y(k+1) =& \ m \Big( \dot{v}_{yd}(k+1) - c_y e_{v_y}(k+1) + (1 - q_y T) s_y(y) \\ & - \varepsilon_y T san\big(s_y(k)\big) \Big) \end{aligned} \tag{13}$$

where, $s_x(k)$ and $s_y(k)$ are sliding-mode functions, $c_x(k)$ and $c_y(k)$ are the coefficients of the sliding-mode function.

$$s_x(k) = c_x e_x(k) + e_{v_x}(k), c_x > 0 \tag{14}$$

$$s_y(k) = c_y e_y(k) + e_{v_y}(k), c_y > 0 \tag{15}$$

From (11), (12) and (13), the expression of the overall control input u can be derived as

$$u(k+1) = -I_{mJ} \left( \bar{S}_c e_v(k+1) - \dot{v}_d(k+1) + \bar{S}_{\varepsilon} sat(s_s) + \bar{S}_q s_s \right) \tag{16}$$

where

$$I_{mJ} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J \end{bmatrix}, \bar{S}_c = \begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & c_{\phi} \end{bmatrix}$$

$$\bar{S}_{\varepsilon} = \begin{bmatrix} \varepsilon_x & 0 & 0 \\ 0 & \varepsilon_y & 0 \\ 0 & 0 & \varepsilon_{\phi} \end{bmatrix}, \bar{S}_q = \begin{bmatrix} Q_x & 0 & 0 \\ 0 & Q_y & 0 \\ 0 & 0 & Q_{\phi} \end{bmatrix}$$

$$s_s = \begin{bmatrix} s_x(k) \\ s_y(k) \\ s_{\phi}(k) \end{bmatrix}, sat(s_s) = \begin{bmatrix} sat(s_x(k)) \\ sat(s_y(k)) \\ sat(s_{\phi}(k)) \end{bmatrix}$$

$$e_v(k+1) = \begin{bmatrix} e_{v_x}(k+1) \\ e_{v_y}(k+1) \\ e_{\omega}(k+1) \end{bmatrix}, \dot{v}_d(k+1) = \begin{bmatrix} \dot{v}_{xd}(k+1) \\ \dot{v}_{yd}(k+1) \\ \dot{\omega}_d(k+1) \end{bmatrix},$$

$Q_x = q_x T - 1, Q_y = q_x T - 1, Q_{\phi} = q_{\phi} T - 1.$

Hence, the attitude and position control of the single ABS can be realized by utilizing (16). Further, in order to realize formation control of multiple ABS and eliminate network delay in distributed networked control, multistep state CPC will be introduced into SMC.

The predictive controller $\hat{u}(k+i|k-s_i)$ is designed for the control input $u_i$ of the discrete system (1):

$$\begin{aligned} \hat{u}_i(k+a_i|k-s_i) =& - \tilde{I}_{mJ_i} \Big( \tilde{S}_{c_i} \hat{v}_i(k+a_i|k-s_i) \\ & - \tilde{S}_{c_i} v_{d_i}(k+a_i) - \dot{v}_{d_i}(k+a_i) \\ & + \tilde{S}_{\varepsilon_i} sat(\hat{s}_{s_i})(k+a_i|k-s_i) \\ & + \tilde{S}_{q_i} \hat{s}_{s_i}(k+a_i|k-s_i) \Big) \\ & + \hat{\mathfrak{c}}_i(k+a_i|k-s_i) \end{aligned} \tag{17}$$

In the above equation, the form likes $\hat{f}(k+a_i|k-s_i)$ representation of the prediction vector at time $k+a_i$ based on all the data at time $k-s_i$ and before that time. In this way, the error caused by the information transmission delay in the system is compensated by prediction. Cooperative item $\hat{\mathfrak{c}}_i(k+a_i|k-s_i)$ is added in the controller design. Collaboration item $\hat{\mathfrak{c}}_i(k+a_i|k-s_i)$ is a correlation function based on the interaction between multiple spacecraft simulator, which can help the spacecraft simulators maintain the desired formation. Set the control time domain length of the predictive controller as $N_u$ and the optimized time domain length as $N_y$. Eq. (17) is expressed in vector form as follows:

$$\begin{aligned} \hat{U}_i(k+a_i|k-s_i) =& - I_{mJ_i} \Big( S_{c_i} \hat{V}_i(k+a_i|k-s_i) \\ & - S_{c_i} V_{d_i}(k+a_i) - \dot{V}_{d_i}(k+a_i) \\ & + S_{\varepsilon_i} S_{gn_i} sat(\hat{s}_{s_i})(k+a_i|k-s_i) \\ & + S_{q_i} \hat{S}_i(k+a_i|k-s_i) \Big) \\ & + \hat{\mathscr{C}}_i(k+a_i|k-s_i) \end{aligned} \tag{18}$$

where, each vector is a new vector set obtained by combining the predicted values of the original vector at different times. Their specific expression is:

$$\hat{U}_i(k+a_i|k-s_i) = \begin{bmatrix} \hat{u}_i(k+a_i+N_u|k-s_i) \\ \hat{u}_i(k+a_i+N_u-1|k-s_i) \\ \vdots \\ \hat{u}_i(k+a_i|k-s_i) \end{bmatrix}$$

$$\hat{V}_i(k+a_i|k-s_i) = \begin{bmatrix} \hat{v}_i(k+a_i+N_u|k-s_i) \\ \hat{v}_i(k+a_i+N_u-1|k-s_i) \\ \vdots \\ \hat{v}_i(k+a_i|k-s_i) \end{bmatrix}$$

$$V_{d_i}(k+a_i) = \begin{bmatrix} v_{d_i}(k+a_i+N_u) \\ v_{d_i}(k+a_i+N_u-1) \\ \vdots \\ v_{d_i}(k+a_i) \end{bmatrix}$$

$$S_{gn_i} = \begin{bmatrix} (-1)^{N_u} I_3 \\ (-1)^{N_u-1} I_3 \\ \vdots \\ (-1)^0 I_3 \end{bmatrix}$$

$$\hat{S}_i(k+a_i|k-s_i) = \begin{bmatrix} \hat{s}_{s_i}(k+a_i+N_u|k-s_i) \\ \hat{s}_{s_i}(k+a_i+N_u-1|k-s_i) \\ \vdots \\ \hat{s}_{s_i}(k+a_i|k-s_i) \end{bmatrix}$$

$$\hat{\mathscr{C}}_i(k+a_i|k-s_i) = \begin{bmatrix} \hat{\mathfrak{c}}_i(k+a_i+N_u|k-s_i) \\ \hat{\mathfrak{c}}_i(k+a_i+N_u-1|k-s_i) \\ \vdots \\ \hat{\mathfrak{c}}_i(k+a_i|k-s_i) \end{bmatrix}$$

and matrix $I_{mJ_i}, S_{c_i}, S_{\epsilon_i}$ and $S_{q_i}$ are the Kronecker products of the original matrix and the $N_u+1$ unit matrix $I_{(N_u+1)\times(N_u+1)}$, i.e

$$I_{mJ_i} = \tilde{I}_{mJ_i} \otimes I_{(N_u+1)\times(N_u+1)}, S_{c_i} = \tilde{S}_{c_i} \otimes I_{(N_u+1)\times(N_u+1)}$$
$$S_{\epsilon_i} = \tilde{S}_{\epsilon_i} \otimes I_{(N_u+1)\times(N_u+1)}, S_{q_i} = \tilde{S}_{q_i} \otimes I_{(N_u+1)\times(N_u+1)}$$

The cost function with cooperation term of the predictive controller based on spacecraft simulator is designed as follows:

$$\begin{aligned} J_i(k) = \sum_{j\in\hbar_i} \Big\| W_{ij} \Big( \big( \hat{Y}_j(k+a_i+1|k-s_j-c_{ji}) \\ - Y_{d_j}(k+a_i+1) \big) - \big( \hat{Y}_i(k+a_i+1|k-s_i) \\ - Y_{d_i}(k+a_i+1) \big) \Big) \Big\|^2 \\ + \Big\| W_i \hat{\mathscr{C}}_i(k+a_i-1|k-s_i) \Big\|^2 \end{aligned} \tag{19}$$

where

$$\hat{Y}_i(k+a_i+1 \mid k-s_i) = \begin{bmatrix} \hat{y}_i(k+a_i+N_y|k-s_i) \\ \hat{y}_i(k+a_i+N_y-1|k-s_i) \\ \vdots \\ \hat{y}_i(k+a_i+1|k-s_i) \end{bmatrix}$$

$$\hat{Y}_j(k+a_i+1|k-s_j-c_{ji}) = \begin{bmatrix} \hat{y}_i(k+a_i+N_y|k-s_j-c_{ji}) \\ \hat{y}_i(k+a_i+N_y-1|k-s_j-c_{ji}) \\ \vdots \\ \hat{y}_i(k+a_i+1|k-s_j-c_{ji}) \end{bmatrix}$$

$$Y_{d_i}(k+a_i+1) = \begin{bmatrix} y_{di}(k+a_i+N_y) \\ y_{di}(k+a_i+N_y-1) \\ \vdots \\ y_{di}(k+a_i+1) \end{bmatrix}$$

and $\hbar_i, i \in N$, represents the collection of agents in the topology of networked control system that directly communicate with the $i$th agent. $W_{ij}$ and $W_i$ are the weight matrices to be designed. $\|\cdot\|$ denotes the Euler norm.

The cost function $J_i(k)$ is used to determine the collaboration term $\hat{\mathscr{C}}_i(k+a_i|k-s_i)$ of $i$th agent. $J_i(k)$ is composed of two Euler norms. The first norm considers the output error of the $i$th agent and its adjacent agents in topology $\hbar_i$ with $N_y+1$ step, which can be used to measure the formation state. The second norm is related to the amplitude of the cooperative term $\hat{\mathscr{C}}_i(k+a_i|k-s_i)$, by adjusting the weight matrices $W_{ij}$ and $W_i$, $\hat{\mathscr{C}}_i(k+a_i|k-s_i)$ is small enough to make the actual output of the controller within the constraint range. $J_i(k)$ takes the minimum value and assigns it to $\hat{\mathscr{C}}_i(k+a_i|k-s_i)$, that is

$$\hat{\mathscr{C}}_i(k+a_i|k-s_i) = \min_{\hat{\mathscr{C}}_i} J_i(k) \tag{20}$$

Next, each prediction vector in (18) is solved in turn to complete the design of distributed cloud prediction controller.

## 4. Design of SMCPC controller

### 4.1. Multistep state predictor

The multistep state predictor is used to predict the system state vector in each stage. In step $k$, for the control cloud node, the following information is known: the system output $y_i$ of step $k-s_i$ and all previous steps; input signal $u_i$ of $k+a_i-1$ step and all previous steps. The $i$th multistep predictor of agent state vector, which can be designed in three steps.

*Step*1 : Predict the state vector $\hat{x}_i$ from step $k-s_i$ to step $k+a_i$.

For the cloud control node of the $i$th spacecraft simulator, all output values $y_i(k-s_i)$ at and before time $k-s_i$ are known at time $k$. In addition, all control inputs $u_i(k+a_i-1)$ at time $k+a_i-1$ and before are also known. Thus, the prediction of $k-s_i$ to $k+a_i$ can be realized.

A state observer is designed to realize the first step of prediction:

$$\begin{aligned} \hat{x}_i(k-s_i+1|k-s_i) = \; & A_i\hat{x}_i(k-s_i|k-s_i-1) \\ & + B_iu_i(k-s_i) + L_i\big( (y_i(k-s_i)) \\ & - C_i\hat{x}_i(k-s_i|k-s_i-1) \big) \end{aligned} \tag{21}$$

Subsequent states are predicted using state space expression (1),

$$\begin{aligned} \hat{x}_i(k+p|k-s_i) = \; & A_i\hat{x}_i(k+p-1|k-s_i) \\ & + B_iu_i(k+p-1) \\ & p = -s_i+2, -s_i+3, \dots, a_i. \end{aligned} \tag{22}$$

where, $\hat{x}_i(k+p|k-s_i)$ denotes the prediction of system state at time $k+p$ based on all known information at time $k-s_i$. Iterate Eq. (22) repeatedly to obtain the general expression of state from $k-s_i$ to $k+a_i$:

$$\begin{aligned} \hat{x}_i(k+p|k-s_i) = \; & A_i^{p+s_i-1}(A_i-L_iC_i)\hat{x}_i(k-s_i|k-s_i-1) \\ & + A_i^{p+s_i-1}L_iy_i(k-s_i) \\ & + \sum_{l=1}^{p+s_i} A_i^{p+s_i-l}B_iu_i(k+l-s_i-1), \\ & p = -s_i+2, -s_i+3, \dots, a_i. \end{aligned} \tag{23}$$

*Step*2 : Predict the state vector $\hat{x}_i$ from step $k+a_i+1$ to step $k+a_i+N_u+1$

Since the real value of the required control input $u_i$ is unknown, the predicted value needs to be used instead. The predicted value of control input $u_i$ can be expressed as:

$$\hat{u}_i(k+a_i+p|k-s_i), p = 0, 1, \dots, N_u$$

The expression of state prediction value obtained from Eq. (1) is

$$\begin{aligned} \hat{x}_i(k+a_i+p|k-s_i) = \; & A_i\hat{x}_i(k+a_i+p-1|k-s_i) \\ & + B_i\hat{u}_i(k+a_i+p-1) \\ & p = 1, 2, \dots, N_u+1 \end{aligned} \tag{24}$$

By iterating the above formula repeatedly, the predicted value expression of the state vector in steps $k + a_i + 1$ to $k + a_i + N_u + 1$ can be obtained:

$$\begin{aligned}\hat{x}_i(k + a_i + p|k - s_i) = {}& A_i^{p+\tau_i-1}(A_i - L_iC_i)\hat{x}_i(k-s_i|k-s_i-1)\\ &+ A_i^{p+\tau_i-1}L_iy_i(k - s_i)\\ &+ \sum_{l=1}^{\tau_i} A_i^{p+\tau_i-1}B_iu_i(k + l - s_i - 1)\\ &+ \sum_{l=1}^{p} A_i^{p-l}B_i\hat{u}_i(k + l + a_i - 1|k - s_i)\end{aligned} \tag{25}$$

*Step*3 : Predict the state vector $\hat{x}_i$ from step $k + a_i + N_u + 2$ to step $k + \bar{a}_i + N_y + c_i$

$c_i = max\left\{c_{i1}, c_{i2}, \dots, c_{iN}\right\}$, $\bar{a}_i$ is the maximum actuators delay $a_j$ of all agent directly interacting with the $i$th agent. Since the control input exceeds the control prediction horizon length, it is advisable to set the control input $u_i$ at this stage to remain unchanged. i.e

$$\begin{aligned}\hat{u}_i(k + a_i + p|k - s_i) &= \hat{u}_i(k + a_i + N_u|k - s_i)\\ p &= N_u + 1, N_u + 2, \dots, N_y + \bar{c}_i - 1\end{aligned} \tag{26}$$

where, $\bar{c}_i = \bar{a}_i - a_i + c_i$, the status value can be predicted by the following formula:

$$\begin{aligned}\hat{x}_i(k+a_i+N_u+p|k - s_i) = {}& A_i\hat{x}_i(k + a_i + N_u + p - 1|k - s_i)\\ &+ B_i\hat{u}_i(k + a_i + N_u|k - s_i)\\ p = {}& 2, 3, \dots, N_y - N_u + \bar{c}_i\end{aligned} \tag{27}$$

Iterate the above formula to obtain (28):

$$\begin{aligned}&\hat{x}_i(k + a_i + N_u + p + 1|k - s_i)\\ &= A_i^{p+N_u+\tau_i}(A_i - L_iC_i)\hat{x}_i(k - s_i|k - s_i - 1)\\ &+ A_i^{p+N_u+\tau_i}L_iy_i(k - s_i)\\ &+ \sum_{l=1}^{\tau_i} A_i^{p+N_u+\tau_i-l+1}B_iu_i(k + l - s_i - 1)\\ &+ \sum_{l=1}^{N_u+1} A_i^{p+N_u-l+1}B_i\hat{u}_i(k + l + a_i - 1|k - s_i)\\ &+ \sum_{l=1}^{p} A_i^{p-l}B_i\hat{u}_i(k + a_i + N_u|k - s_i)\end{aligned} \tag{28}$$

At this moment, the state prediction of all agents has been completed.

### 4.2. Output prediction of ABS

The output vector is predicted based on the prediction of the state vector, and the system output is predicted using the state space expression (1), that is

$$\hat{y}_i(k + p|k - s_i) = C_i\hat{x}_i(k + p|k - s_i) \tag{29}$$

where, $p = -s_i + 1, -s_i + 2, \dots, \tau_i + \bar{c}_i + N_y$. Next, the prediction vectors $\hat{Y}_j(k + a_i + 1|k - s_i)$ and $\hat{Y}_i(k + a_i + 1|k - s_i - c_{ij})$ need to completed.

For the prediction of output vector $\hat{Y}_i(k + a_i + 1|k - s_i)$, the output predictor sub vectors $y_i$ at different times are obtained based on the multistep state prediction expressions (23), (25) and (28), and then sorted into matrix form:

$$\begin{aligned}\hat{Y}_i(k + a_i + 1|k - s_i) = {}& I_{C_i}\ \big(\ E_{Y_i}\hat{x}_i(k - s_i|k - s_i - 1)\\ &+F_{Y_i}y_i(k - s_i) + G_{Y_i}U_i(k - s_i)\\ &+H_{Y_i}\hat{U}_i(k + a_i|k - s_i)\ \big)\end{aligned} \tag{30}$$

where $I_{C_i} = \begin{bmatrix}0_{N_y\times 3\bar{c}_i} & I_{3N_y\times 3N_y}\end{bmatrix}$.

$$E_{Y_i} = \begin{bmatrix} C_iA_i^{\tau_i+N_y+\bar{c}_i-1}(A_i-L_iC_i)\\ C_iA_i^{\tau_i+N_y+\bar{c}_i-2}(A_i-L_iC_i)\\ \vdots\\ C_iA_i^{\tau_i}(A_i-L_iC_i)\end{bmatrix}, F_{Y_i} = \begin{bmatrix} C_iA_i^{\tau_i+N_y+\bar{c}_i-1}L_i\\ C_iA_i^{\tau_i+N_y+\bar{c}_i-2}L_i\\ \vdots\\ C_iA_i^{\tau_i}L_i\end{bmatrix}$$

$$G_{Y_i} = \begin{bmatrix} C_iA_i^{N_y+\bar{c}_i}B_i & C_iA_i^{N_y+\bar{c}_i+1}B_i & \cdots & C_iA_i^{N_y+\tau_i+\bar{c}_i-1}B_i\\ C_iA_i^{N_y+\bar{c}_i-1}B_i & C_iA_i^{N_y+\bar{c}_i}B_i & \cdots & C_iA_i^{N_y+\tau_i+\bar{c}_i-2}B_i\\ \vdots & \vdots & \vdots & \vdots\\ C_iA_iB_i & C_iA_i^2B_i & \cdots & C_iA_i^{\tau_i}B_i\end{bmatrix}$$

$$H_{Y_i} = \begin{bmatrix} \sum_{p=0}^{N_y+\bar{c}_i+N_u-1} C_iA_i^pB_i & C_iA_i^{N_y+\bar{c}_i-N_u}B_i & \cdots & C_iA_i^{N_y+\bar{c}_i-1}B_i\\ \sum_{p=0}^{N_y+\bar{c}_i+N_u-2} C_iA_i^pB_i & C_iA_i^{N_y+\bar{c}_i-N_u-1}B_i & \cdots & C_iA_i^{N_y+\bar{c}_i-2}B_i\\ \vdots & \vdots & \ddots & \vdots\\ \sum_{p=0}^{1} C_iA_i^pB_i & C_iA_i^2B_i & \cdots & C_iA_i^{N_u+1}B_i\\ C_iB_i & C_iA_iB_i & \cdots & C_iA_i^{N_u}B_i\\ & C_iB_i & \cdots & C_iA_i^{N_u-1}B_i\\ & & \ddots & \vdots\\ & & & C_iB_i\end{bmatrix}$$

Following the derivation process of Eq. (30), the vector expression of output prediction $\hat{Y}_i(k + a_i + 1|k - s_i - c_{ij})$ can be obtained

$$\begin{aligned}\hat{Y}_i(k+a_i+1|k-s_i-c_{ij}) = {}& I_{C_{ij}}\ \big(\ E_{Y_i}\hat{x}_i(k-s_i-c_{ij}|k-s_i-c_{ij}-1)\\ &+F_{Y_i}y_i(k-s_i-c_{ij})+G_{Y_i}U_i(k-s_i-c_{ij})\\ &+H_{Y_i}\hat{U}_i(k+a_i-c_{ij}|k-s_i-c_{ij})\ \big)\end{aligned} \tag{31}$$

where, $I_{C_{ij}} = \begin{bmatrix}0_{3N_y\times 3(\bar{c}-c_{ij})} & I_{3N_y\times 3N_y} & 0_{3N_y\times 3c_{ij}}\end{bmatrix}$.

### 4.3. Sliding surface function vector prediction

Similar to the prediction of output vector, the prediction of sliding-mode function vector can be obtained from the prediction value of state vector

$$\hat{s}_{s_i}(k + p|k - s_i) = S_{s_i}\hat{x}_i(k + p|k - s_i) \tag{32}$$

where, $S_{s_i}$ is the sliding surface parameter matrix

$$S_{s_i} = \begin{bmatrix}\bar{S}_{c_i} & I_{3\times 3}\end{bmatrix} = \begin{bmatrix} c_{x_i} & 0 & 0 & 1 & 0 & 0\\ 0 & c_{y_i} & 0 & 0 & 1 & 0\\ 0 & 0 & c_{\phi_i} & 0 & 0 & 1\end{bmatrix}$$

Through the prediction expression (25) of the second stage state vector and the relationship between the sliding-mode function and the state vector (32), the prediction value $\hat{S}_i(k + a_i|k - s_i)$ of the $N_u + 1$ step sliding-mode surface function from step $k + a_i$ can be obtained:

$$\begin{aligned}\hat{S}_i(k+a_i|k-s_i) = {}& E_{S_i}\hat{x}_i(k - s_i|k - s_i - 1)\\ &+F_{S_i}y_i(k - s_i) + G_{S_i}U_i(k - s_i)\\ &+H_{S_i}\hat{U}_i(k + a_i|k - s_i)\end{aligned} \tag{33}$$

where

$$E_{S_i} = \begin{bmatrix} S_{s_i}A_i^{\tau_i+N_u-1}(A_i - L_iC_i)\\ S_{s_i}A_i^{\tau_i+N_u-2}(A_i - L_iC_i)\\ \vdots\\ S_{s_i}A_i^{\tau_i-1}(A_i - L_iC_i)\end{bmatrix}, F_{S_i} = \begin{bmatrix} S_{s_i}A_i^{\tau_i+N_u-1}L_i\\ S_{s_i}A_i^{\tau_i+N_u-2}L_i\\ \vdots\\ S_{s_i}A_i^{\tau_i-1}L_i\end{bmatrix}$$

$$G_{S_i} = \begin{bmatrix} S_{s_i}A_i^{N_u}B_i & S_{s_i}A_i^{N_u+1}B_i & \cdots & S_{s_i}A_i^{\tau_i+N_u-1}B_i\\ S_{s_i}A_i^{N_u-1}B_i & S_{s_i}A_i^{N_u}B_i & \cdots & S_{s_i}A_i^{\tau_i+N_u-2}B_i\\ \vdots & \vdots & \vdots & \vdots\\ S_{s_i}B_i & S_{s_i}A_iB_i & \cdots & S_{s_i}A_i^{\tau_i-1}B_i\end{bmatrix}$$

$$H_{S_i} = \begin{bmatrix} 0 & S_{s_i}B_i & \cdots & S_{s_i}A_i^{N_u-1}B_i\\ 0 & 0 & \cdots & S_{s_i}A_i^{N_u-2}B_i\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & S_{s_i}B_i\\ 0 & 0 & \cdots & 0\end{bmatrix}$$

### 4.4. Velocity and angular velocity vector prediction

The velocity and angular velocity vectors are predicted by (34)

$$\hat{v}_i(k+p|k-s_i) = V_i\hat{x}_i(k+p|k-s_i) \tag{34}$$

where $\hat{v}_i = [\hat{v}_{xi}, \hat{v}_{y_i}, \hat{\omega}_i]^{\mathrm{T}}$ are velocity and angular velocity prediction vectors, respectively; $V_i$ is the extraction matrix of velocity and angular velocity, the specific form is

$$V_i = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Through the state vector prediction expression (25) of the second stage and the relationship between the velocity vector and the state vector (34), the velocity and angular velocity prediction vector $\hat{V}_i(k+a_i|k-s_i)$ from step $k+a_i$ to step $N_u+1$ can be deduced:

$$\begin{aligned}\hat{V}_i(k+a_i|k-s_i) =& E_{V_i}\hat{x}_i(k-s_i|k-s_i-1) \\ &+F_{V_i}y_i(k-s_i) + G_{V_i}U_i(k-s_i) \\ &+H_{V_i}\hat{U}_i(k+a_i|k-s_i)\end{aligned} \tag{35}$$

where

$$E_{V_i} = \begin{bmatrix} V_iA_i^{\tau_i+N_u-1}(A_i-L_iC_i) \\ V_iA_i^{\tau_i+N_u-2}(A_i-L_iC_i) \\ \vdots \\ V_iA_i^{\tau_i-1}(A_i-L_iC_i) \end{bmatrix}$$

$$F_{V_i} = \begin{bmatrix} V_iA_i^{\tau_i+N_u-1}L_i \\ V_iA_i^{\tau_i+N_u-2}L_i \\ \vdots \\ V_iA_i^{\tau_i-1}L_i \end{bmatrix}$$

$$G_{V_i} = \begin{bmatrix} V_iA_i^{N_u}B_i & V_iA_i^{N_u+1}B_i & \cdots & V_iA_i^{\tau_i+N_u-1}B_i \\ V_iA_i^{N_u-1}B_i & V_iA_i^{N_u}B_i & \cdots & V_iA_i^{\tau_i+N_u-2}B_i \\ \vdots & \vdots & \vdots & \vdots \\ V_iB_i & V_iA_iB_i & \cdots & V_iA_i^{\tau_i-1}B_i \end{bmatrix}$$

$$H_{V_i} = \begin{bmatrix} 0 & V_iB_i & \cdots & V_iA_i^{N_u-1}B_i \\ 0 & 0 & \cdots & V_iA_i^{N_u-2}B_i \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & V_iB_i \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

### 4.5. Optimal design of controller cooperative terms

Based on (18) and (33), the control quantity prediction vector (35) can be calculated as

$$\begin{aligned}\hat{U}_i(k+a_i|k-s_i) =& -I_{mJ_i}\Big(-S_{c_i}V_{d_i}(k+a_i) - \dot{V}_{d_i}(k+a_i) \\ &+ S_{\varepsilon_i}S_{gn_i}sat(\hat{s}_{s_i})(k+a_i|k-s_i) \\ &+ (S_{q_i}E_{S_i} + S_{c_i}E_{V_i})\hat{x}_i(k-s_i|k-s_i-1) \\ &+ (S_{q_i}F_{S_i} + S_{c_i}F_{V_i})y_i(k-s_i) \\ &+ (S_{q_i}G_{S_i} + S_{c_i}F_{V_i})U_i(k-s_i) \\ &- S_{q_i}S_{d_i}(k+a_i)\Big) \\ &- I_{mJ_i}(S_{q_i}H_{S_i} + S_{c_i}H_{V_i})\hat{U}_i(k+a_i|k-s_i) \\ &+ \hat{\mathscr{C}}_i(k+a_i|k-s_i)\end{aligned} \tag{36}$$

where $S_{d_i}(k+a_i)$ represents the expected value of the sliding surface function,

$$S_{d_i}(k+a_i) = \begin{bmatrix} s_{sd_i}(k+a_i+N_u) \\ s_{sd_i}(k+a_i+N_u-1) \\ \vdots \\ s_{sd_i}(k+a_i) \end{bmatrix}$$

$$s_{sd_i}(k+a_i+p) = \begin{bmatrix} c_{x_i}d_{xd_i}(k+a_i+p) + v_{xd_i}(k+a_i+p) \\ c_{y_i}d_{yd_i}(k+a_i+p) + v_{yd_i}(k+a_i+p) \\ c_{\phi_i}\phi_{d_i}(k+a_i+p) + \omega_{d_i}(k+a_i+p) \end{bmatrix}$$

$p = 0, 1, \ldots, N_u$.

Rewriting the expression of control prediction vector:

$$\hat{U}_i(k+a_i|k-s_i) = \gamma_i(k-s_i) + M_i\hat{\mathscr{C}}(k+a_i|k-s_i) \tag{37}$$

where $\gamma_i(k-s_i)$ represents the sliding-mode predictive control law improved by state prediction, which is used for the motion control of single agent; $M_i\hat{\mathscr{C}}(k+a_i|k-s_i)$ represents the part of the control law related to cooperative control, which is used to maintain the formation.

$$\begin{aligned}\gamma_i(k-s_i) =& -I_{mJ_i}M_i\Big(-S_{c_i}V_{d_i}(k+a_i) - \dot{V}_{d_i}(k+a_i) \\ &+ S_{\varepsilon_i}S_{gn_i}sat(\hat{s}_{s_i}(k+a_i|k-s_i)) \\ &+ (S_{q_i}E_{S_i} + S_{c_i}E_{V_i})\hat{x}_i(k-s_i|k-s_i-1) \\ &+ (S_{q_i}F_{S_i} + S_{c_i}F_{V_i})y_i(k-s_i) \\ &+ (S_{q_i}G_{S_i} + S_{c_i}G_{V_i})U_i(k-s_i) \\ &- S_{q_i}S_{d_i}(k+a_i|k-s_i)\Big)\end{aligned} \tag{38}$$

where, $M_i = (I + I_{mJ_i}S_{q_i}H_{S_i} + I_{mJ_i}S_{c_i}H_{V_i})^{-1}$. Make the performance index function $J_i(k)$ derive from $\hat{\mathscr{C}}(k+a_i|k-s_i)$ to get the minimum value

$$\begin{aligned}&-M_i^{\mathrm{T}}H_{Y_i}^{\mathrm{T}}I_{C_i}^{\mathrm{T}}\sum_{j\in\hbar_i}W_{ij}\Big(\hat{Y}_j(k+a_i+1|k-s_j-c_{ji}) - Y_{d_j}(k+a_i+1) \\ &\quad -I_{C_i}\big(E_{Y_i}\hat{x}_i(k-s_i|k-s_i-1) + F_{Y_i}y_i(k-s_i) \\ &\quad +G_{Y_i}U_i(k-s_i) + H_{Y_i}\hat{U}_i(k+a_i|k-s_i)\big) \\ &\quad +Y_{d_i}(k+a_i+1)\Big) \\ &\quad +W_i\hat{\mathscr{C}}(k+a_i|k-s_i) = 0\end{aligned} \tag{39}$$

Further, the prediction vector of the cooperation term can be obtained

$$\begin{aligned}\hat{\mathscr{C}}_i(k+a_i|k-s_i) =& \Gamma_i^{-1}M_i^{\mathrm{T}}H_{Y_i}^{\mathrm{T}}I_{C_i}^{\mathrm{T}}\sum_{j\in\hbar_i}W_{ij}\Big(\hat{Y}_j(k+a_i+1|k-s_j-c_{ji}) \\ &-Y_{d_j}(k+a_i+1) - I_{C_i}\big(E_{Y_i}\hat{x}_i(k-s_i|k-s_i-1) \\ &+F_{Y_i}y_i(k-s_i) + G_{Y_i}U_i(k-s_i) \\ &+H_{Y_i}\gamma_i(k+a_i|k-s_i)\big) + Y_{d_i}(k+a_i+1)\Big)\end{aligned} \tag{40}$$

where

$$\Gamma_i = \sum_{j\in\hbar_i}W_{ij}M_i^{\mathrm{T}}H_{Y_i}^{\mathrm{T}}I_{C_i}^{\mathrm{T}}I_{C_i}H_{Y_i}M_i + W_iI$$

### 4.6. Stability and consensus analysis

For the sliding-mode controller based on exponential reaching law (6), the system reaches the quasi sliding surface under the following condition (Gao, Wang, & Homaifa, 1995).

$$[s(k+1) - s(k)]\, s(k) < 0 \tag{41}$$

which is a necessary but not sufficient condition for quasi sliding-mode motion.

**Theorem 1.** *For discrete systems (1) and sliding-mode functions (8), (14) and (15), controller (16) will make the system reaches the sliding-mode surface in a finite step to ensure the stability of the system.*

**Proof.** Selecting the Lyapunov function.

$$\Theta(k) = \frac{1}{2}s^2(k) \tag{42}$$

When the condition of (43) is satisfied

$$\Delta\Theta(k) = s^2(k+1) - s^2(k) < 0, s(k) \neq 0 \tag{43}$$

According to Lyapunov stability theorem, $s(k) = 0$ is the globally asymptotically stable equilibrium surface, i.e., the state at any initial position will converge to the switching surface $s(k)$. So the arrival condition will be taken as:

$$s^2(k+1) < s^2(k) \tag{44}$$

From the literature (Sarpturk, Istefanopulos, & Kaynak, 1987), when the sampling time $T$ is taken a small value, the discrete sliding-mode existence and reachability conditions are

$$\begin{aligned} &[s(k+1) - s(k)]\, sat(s(k)) < 0, \\ &[s(k+1) + s(k)]\, sat(s(k)) > 0 \end{aligned} \tag{45}$$

By the known conditions gives before, exponential-based discrete reaching law (6) satisfies that

$$\begin{aligned} [s(k+1) - s(k)]\, sat(s(k)) &= [-qTs(k) - \varepsilon T sat(s(k))]\, sat(s(k)) \\ &= -qT|s(k)| - \varepsilon T|s(k)| < 0 \end{aligned} \tag{46}$$

Also, since the sampling time $T$ is small, $2 - qT \gg 0$, then

$$\begin{aligned} [s(k+1)+s(k)]\, sat(s(k)) &= [(2-qT)s(k) - \varepsilon T sat(s(k))]\, sat(s(k)) \\ &= (2-qT)|s(k)| - \varepsilon T|s(k)| > 0 \end{aligned} \tag{47}$$

As the discrete reaching law satisfies the arrival condition, the system can reach the quasi sliding surface in the finite step. Thus, $s_x(k)$, $s_y(k), s_\phi(k)$ are reachable, the control system will converges to the switching surface $s(k)$ at any initial position, the system sliding surface stability.

According to the theory of sliding mode control, when the sliding surface $s(k)$ can be reached, the system state will move along the sliding surface, that is, $\dot{s}(k) = 0$. Since $c > 0$, the signs of $e(k)$ and $s(k)$ are the same. As a result, the absolute value of $s(k)$ will decrease over time, approaching zero. Since $c > 0$, the sign of $\dot{e}(k)$ is opposite to the sign of $\ddot{e}(k)$, meaning that $\ddot{e}(k)$ will also approach zero over time. Using $\dot{s}(k) = 0$ and $s(k)$ in the definition of the sliding surface $s(k) = ce(k) + \ddot{e}(k)$, we have $0 = c\dot{e}(k) + \ddot{e}(k)$, as $\dot{s}(k) = 0$, $\dot{e}(k) = -\frac{c}{\varepsilon}s(k)$. Therefore, $\ddot{e}(k)$ is proportional to $s(k)$, which approaches zero over time. Thus, $\ddot{e}(k)$ also approaches zero over time. Since $\ddot{e}(k)$ approaches zero and $\dot{e}(k)$ approaches zero, we can integrate $\dot{e}(k)$ to obtain $e(k)$ as follows:

$$e(k) = \int_0^k \dot{e}(\tau)d\tau = -\frac{c}{\varepsilon}\int_0^k s(\tau)d\tau \tag{48}$$

Since $s(k)$ approaches zero over time, $e(k)$ will also approach zero over time. Therefore, when $s(k)$ is reachable, $e(k)$ will approach zero over time and eventually be zero, i.e. $\lim_{t\to+\infty}|s(k)| = 0, e(k) = 0$.

**Remark.** Based on (39), the predicted value of the collaborative term $\hat{\mathscr{C}}_i$ is a continuous function about the weight matrix $W_{ij}$, and when the weight matrix $W_{ij}$ is a zero matrix, $\hat{\mathscr{C}}_i$ is a zero vector.

The distributed networked sliding-mode cloud predictive control law designed in this paper is equivalent to the single agent control law and the stability of both is the same. $\forall\sigma > 0$, $\exists\eta > 0$, makes $\|W_{ij}\| < \eta$, $\|\hat{\mathscr{C}}\| < \sigma$. That is, when the two norm of weight matrix $W_{ij}$ and the collaborative term $\hat{\mathscr{C}}_i$ are small enough, the stability of the system will not be affected. So, $y_i(t) \to y_d$ as $t \to \infty$, $\forall i \in \omega$, that is, when the reference output $y_d$ is bounded, the output $y_i$ is also bounded (Liu, 2020). This leads to

$$\lim_{x\to\infty}\|y_j(t) - y_i(t)\| = 0, \forall j, i \in N \tag{49}$$

The above equation ensures the consensus of the output of all agents. The distributed sliding-mode cloud predictive control scheme achieves simultaneous stability and consensus.

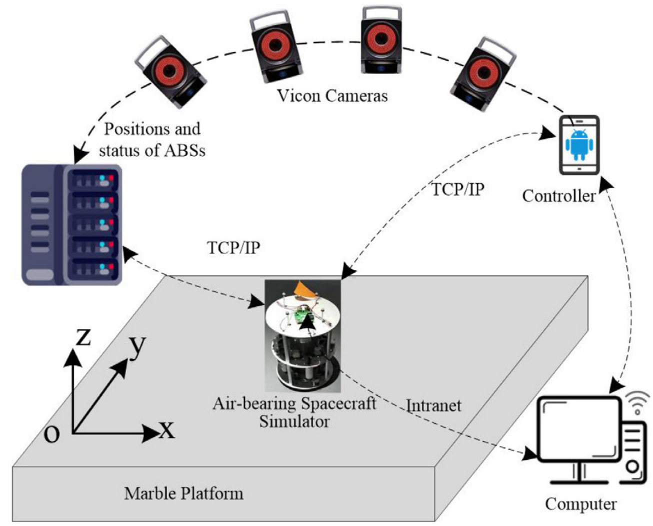


**Fig. 5.** NCS platform for ABS.

**Table 1**
The three ABSs related parameters.

| Parameters | ABS 1 | ABS 2 | ABS 3 |
|---|---|---|---|
| Mass (kg) | 19.5 | 19.4 | 19.0 |
| Rotational inertia (kg m$^2$) | 0.198 | 0.195 | 0.210 |
| Valve jetting force (N) | 0.25 | 0.25 | 0.25 |
| Radius (m) | 0.15 | 0.15 | 0.15 |

## 5. Experiment based on ABS

### 5.1. NCS experimental platform

Experimentation is performed on the NCS platform. In the NCS platform, the developed smartphone is used as the controller and the actuator is the USR-C322 control board on the ABS. The sensor are composed by the VICON motion capture cameras above the experimental platform. In the closed-loop control system, USR-C322 first receives control signals through the network from smartphones and controls the air valve voltage of ABS thrusters to realize air injection control. Then the VICON camera captures the position information of the fluorescent ball on the ABS and transmits the captured information to the VICON server via Ethernet. Lastly, the server transmits the fluorescent ball position information to the smartphone controller, which decodes the fluorescent ball position information to obtain ABS position and attitude information. The global block diagram of NCS experimental platform is shown in Fig. 5. Table 1 shows the relevant parameters of the three ABSs in the experiment.

### 5.2. Simulation test

In the simulation experiments, to demonstrate the path tracking and formation control performance of the designed control scheme, the three ABSs are given unit step signals set to the same reference trajectory. The simulation results, x-coordinates, y-coordinates and attitude angle transformations of the three ABSs are put into three figures respectively for comparison. From the simulation results in Fig. 6, Fig. 7, and Fig. 8, it can be seen that the three ABSs are able to enter the desired formation and maintain very similar attitude angle deviations for cooperative motion after about 15 s of simulation, and the three ABSs reach their desired positions and track on the reference trajectory at 50 s. It can be seen that the designed distributed networked sliding-mode cloud predictive control scheme can realize the formation control of ABSs.

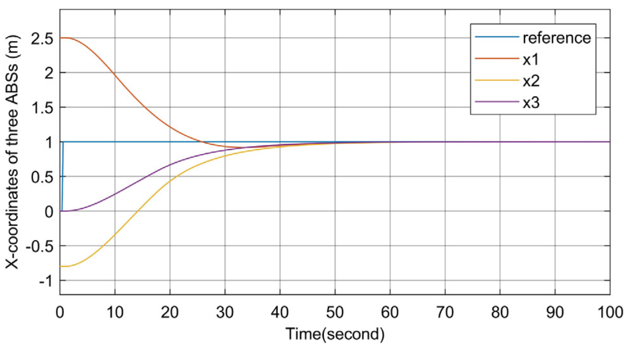


**Fig. 6.** Simulation results of three ABSs X-coordinate transformations.

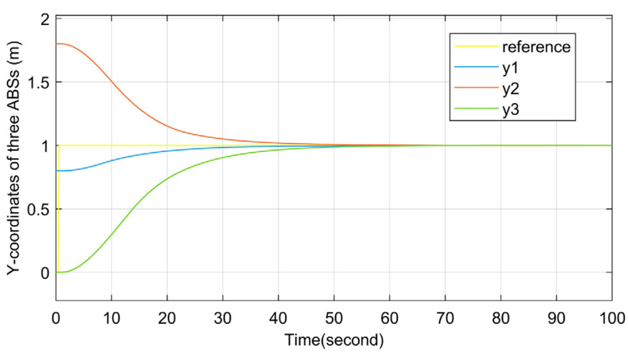


**Fig. 7.** Simulation results of three ABSs Y-coordinate transformations.

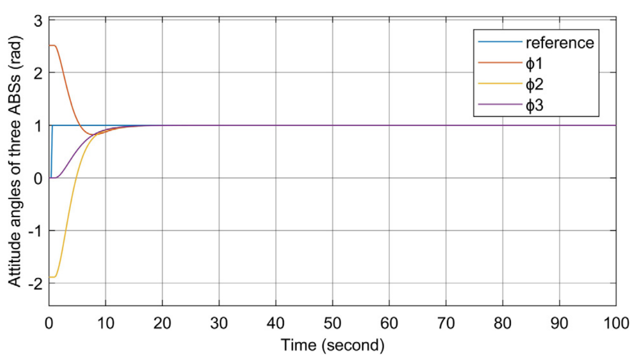


**Fig. 8.** Simulation results of three ABSs attitude angle transformation.

### 5.3. Experimental of ABS formation control

The expected trajectories of the three ABS are shown in Eqs. (50)–(52) respectively. Setting up the three ABSs to run in fixed formation with 1.5 m side length in equilateral triangle. The three ABSs communicate through the UDP module which is set in simulink during the operation to complete the real-time cooperation of position and attitude.

$$\begin{cases} R_{x1}(t) = 0.8 - 0.007t \\ R_{y1}(t) = 1.25 \\ R_{\phi 1}(t) = 0 \end{cases} \tag{50}$$

$$\begin{cases} R_{x2}(t) = 2.1 - 0.007t \\ R_{y2}(t) = 0.5 \\ R_{\phi 2}(t) = 0 \end{cases} \tag{51}$$

$$\begin{cases} R_{x3}(t) = 2.1 - 0.007t \\ R_{y3}(t) = 2 \\ R_{\phi 3}(t) = 0 \end{cases} \tag{52}$$

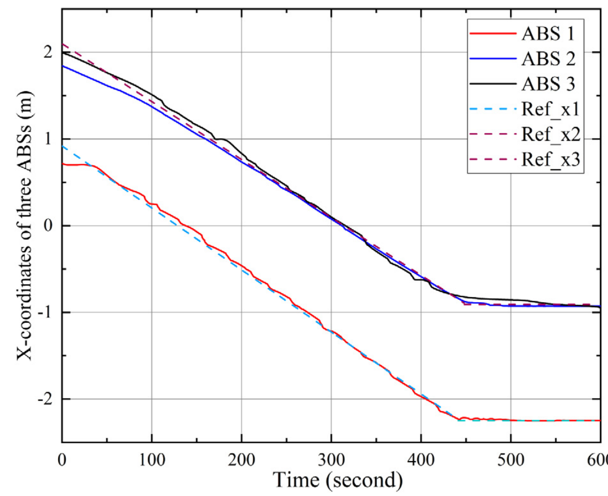


**Fig. 9.** X-coordinate tracking trajectory of ABSs.

**Table 2**
Start and target positions of ABS.

| Coordinate | ABS 1 | ABS 2 | ABS 3 |
|---|---|---|---|
| X (m) | 0.7 to −2.21 | 1.85 to −0.91 | 2 to −0.91 |
| Y (m) | 2 to 1.25 | 1.1 to 0.5 | 2.65 to 2 |
| Angle (rad) | $0.03\pi$ to 0 | $0.03\pi$ to 0 | $0.05\pi$ to 0 |

**Table 3**
Controller parameters.

| Parameter | $q_x$ | $q_y$ | $q_\phi$ | $c_x$ | $c_y$ | $c_\phi$ | $\varepsilon_x$ | $\varepsilon_y$ | $\varepsilon_\phi$ |
|---|---|---|---|---|---|---|---|---|---|
| Value | 0.02 | 0.02 | 0.3 | 0.15 | 0.15 | 0.2 | 0.005 | 0.005 | 0.01 |

Set the starting position and target position of ABS as shown in Table 2. When the target position is reached the ABS starts to hover with the previous formation.

The sampling time of the system is set to 0.2s, time delay parameters $a_i = 1$, $s_i = 1$, $c_{ij} = 3$. As the experiments are performed under the same network, it is considered that the same type of delay in the system takes the identical value. The weight matrices $W_{ij}$ and $W_i$ of the performance indicator function are given as

$$W_{ij} = \begin{bmatrix} 30 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 0.2 \end{bmatrix}, W_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, i, j = 1, 2, 3$$

Due to the small individual differences of the ABSs and in order to test the robustness of the designed controller, each ABS uses completely equal parameters in the experiment. The parameters are given in Table 3.

The desired position signals, actual position signals of the three ABSs are shown in Figs. 9–11, and the $x$-axis, $y$-axis and attitude angle tracking error signal curves are shown in Figs. 12–14. From the position coordinate transformation, it can be seen that the three ABSs start to converge and reach the desired position after a transition time of about 10 s. ABSs coordinate with each other to maintain the fixed formation and moves along the reference trajectory after reaching the desired position. Analysis of the steady-state error of each ABS yields Table 4. During the ABS formation control experiment, there are external disturbances, such as the effect of dust on the marble platform against the planner air bearing, the disturbance of each ABS thruster, and the change of ABS own counterweight, etc. The experimental error will increase compared to the simulation error. From the transition phase of formation, it can be seen that the tracking reference track fluctuates significantly, but because of the collaborative control of ABS, it can quickly adjust to the desired state. The final steady-state error is kept

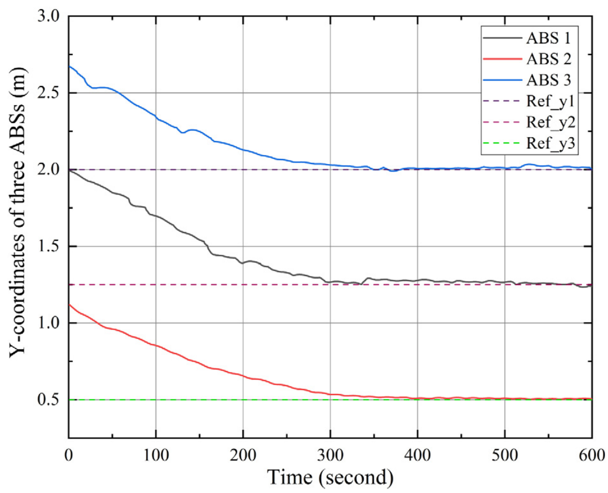


**Fig. 10.** Y-coordinate tracking trajectory of ABSs.

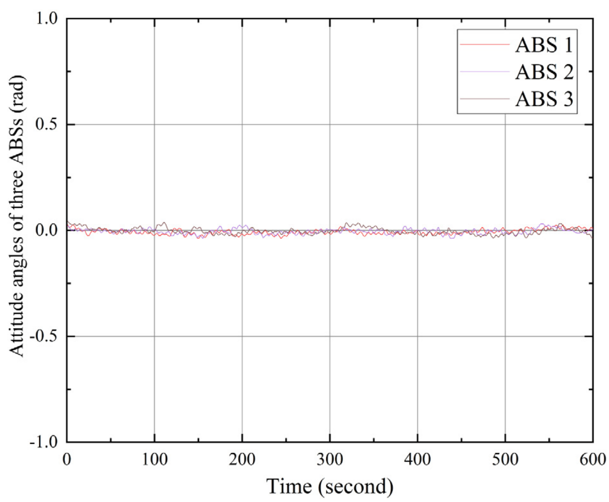


**Fig. 11.** Attitude angle tracking trajectory of ABSs.

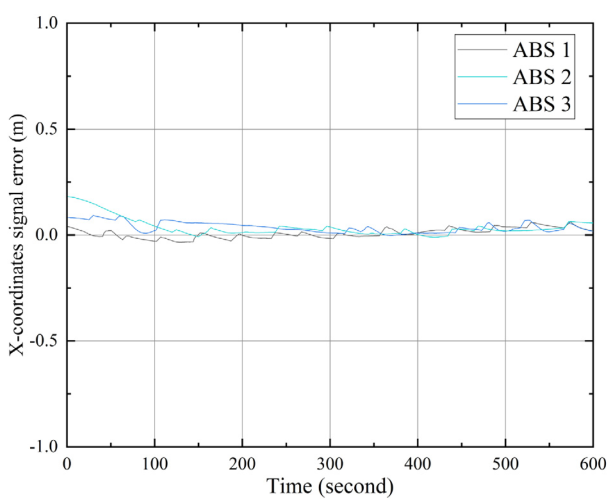


**Fig. 12.** X-coordinate signal error.

within the controllable range. This proved that the designed distributed networked cloud prediction sliding-mode controller can ensure the control accuracy of the controlled agent through the communication

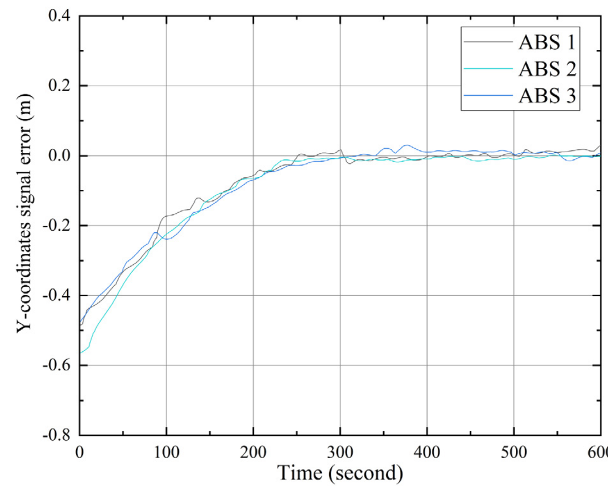


**Fig. 13.** Y-coordinate signal error.

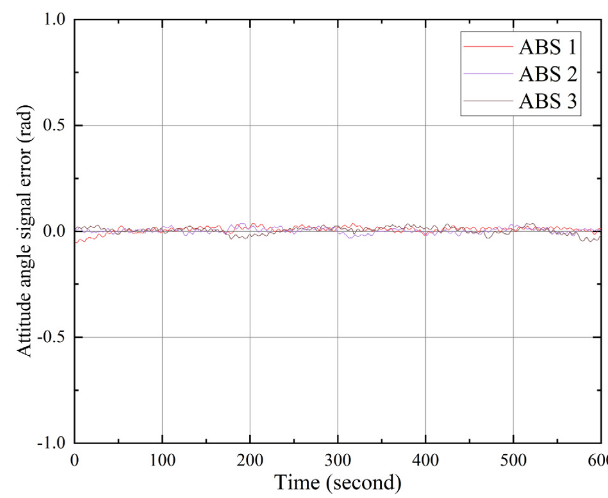


**Fig. 14.** Attitude angle signal error.

**Table 4**
ABSs steady-state errors.

| Coordinate | ABS 1 | ABS 2 | ABS 3 |
|---|---|---|---|
| X (m) | 0.025 | 0.025 | 0.025 |
| Y (m) | 0.015 | 0.02 | 0.02 |
| Angle (rad) | 0.05 | 0.05 | 0.05 |

between controllers. The experimental results show that the SMCPC scheme designed in this paper achieves the following results in the realization of formation control: The goal of reducing individual steady-state errors is achieved through distributed coordinate control; Making the controlled agents form the desired formation faster; The controlled agents form the formation with better cooperation and improved the robustness of the system. These are consistent with our original design intent and design goals.

## 6. Conclusion

In this article, distributed sliding-mode cloud predictive control for NMACS is investigated. A distributed SMCPC control scheme is designed which combines the advantages of both SMC and CPC control methods to realize the formation control of NMACS and compensate the communication delay between cloud nodes. The optimal design of the cost function ensures that the SMCPC achieves the desired control performance in multi-agent formation control. The feasibility of the control scheme is verified by stability and consistency analysis. Finally, the effectiveness of the control scheme in practical applications is verified by ABS formation control experiments. In this article, we only preliminarily study the distributed sliding-mode cloud predictive control of NMACS with fixed delay. However, there are still problems such as network attacks, random perturbations, variable topologies, and variable network delays in NMACS. To resolve these commonly existing problems in NMACS, the proposed distributed sliding-mode cloud predictive control scheme still needs further research.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References


Cao, Lei, Liu, Guo-Ping, & Zhang, Da-Wei (2023). Cloud-based predictive formation control of networked multi-agent system and its application to air bearing spacecraft simulators. *ISA Transactions*, http://dx.doi.org/https://doi.org/10.1016/j.isatra.2023.03.013.

Chi, Chuanguo, Liu, Guo-Ping, & Hu, Wenshan (2022). Design and implementation of a mobile terminal cloud supervisory control platform for networked control systems. *Transactions of the Institute of Measurement and Control*, *44*(5), 1070–1080. http://dx.doi.org/10.1177/01423312211049214.

Ferrara, Antonella, Incremona, Gian Paolo, & Magni, Lalo (2013). A robust MPC/ISM hierarchical multi-loop control scheme for robot manipulators. In *52nd IEEE Conference on decision and control* (pp. 3560–3565). IEEE, http://dx.doi.org/10.1109/CDC.2013.6760430.

Ferrara, Antonella, Incremona, Gian Paolo, & Magni, Lalo (2014). Model-based event-triggered robust MPC/ISM. In *2014 European control conference* (pp. 2931–2936). IEEE, http://dx.doi.org/10.1109/ECC.2014.6862523.

Fesharaki, Shekoofeh Jafari, Kamali, Marzieh, Sheikholeslam, Farid, & Talebi, Heidar Ali (2020). Robust model predictive control with sliding mode for constrained nonlinear systems. *IET Control Theory & Applications*, *14*(17), 2592–2599. http://dx.doi.org/10.1049/iet-cta.2019.1357.

Gao, Weibing, Wang, Yufu, & Homaifa, Abdollah (1995). Discrete-time variable structure control systems. *IEEE Transactions on Industrial Electronics*, *42*(2), 117–122. http://dx.doi.org/10.1109/41.370376.

Garcia-Gabin, Winston, Zambrano, Darine, & Camacho, Eduardo F. (2009). Sliding mode predictive control of a solar air conditioning plant. *Control Engineering Practice*, *17*(6), 652–663. http://dx.doi.org/10.1016/j.conengprac.2008.10.015.

Ge, Xiaohua, Yang, Fuwen, & Han, Qing-Long (2017). Distributed networked control systems: A brief overview. *Information Sciences*, *380*, 117–131. http://dx.doi.org/10.1016/j.ins.2015.07.047.

Huang, Yi, Liu, Guo-Ping, & Hu, Wenshan (2022). Priori-guided and data-driven hybrid model for wind power forecasting. *ISA Transactions*, http://dx.doi.org/10.1016/j.isatra.2022.07.028.

Incremona, Gian Paolo, Ferrara, Antonella, & Magni, Lalo (2017). Asynchronous networked MPC with ISM for uncertain nonlinear systems. *IEEE Transactions on Automatic Control*, *62*(9), 4305–4317. http://dx.doi.org/10.1109/TAC.2017.2653760.

Ionescu, Clara, & Muresan, Cristina (2015). Sliding mode control for a class of subsystems with fractional order varying trajectory dynamics. *Fractional Calculus & Applied Analysis*, *18*(6), 1441–1451. http://dx.doi.org/10.1515/fca-2015-0083.

Ji, Daxiong, Ren, Jie, Liu, Changxin, & Shi, Yang (2021). Stabilizing terminal constraint-free nonlinear MPC via sliding mode-based terminal cost. *Automatica*, *134*, Article 109898. http://dx.doi.org/10.1016/j.automatica.2021.109898.

Knorn, Steffi, Chen, Zhiyong, & Middleton, Richard H. (2015). Overview: Collective control of multiagent systems. *IEEE Transactions on Control of Network Systems*, *3*(4), 334–347. http://dx.doi.org/10.1109/TCNS.2015.2468991.

Li, Hongbo, Yang, Hongjiu, Sun, Fuchun, & Xia, Yuanqing (2014). Sliding-mode predictive control of networked control systems under a multiple-packet transmission policy. *IEEE Transactions on Industrial Electronics*, *61*(11), 6234–6243. http://dx.doi.org/10.1109/TIE.2014.2311411.

Liu, Guo-Ping (2017). Predictive control of networked multiagent systems via cloud computing. *IEEE Transactions on Cybernetics*, *47*(8), 1852–1859. http://dx.doi.org/10.1109/TCYB.2017.2647820.

Liu, Guo-Ping (2020). Coordinated control of networked multiagent systems via distributed cloud computing using multistep state predictors. *IEEE Transactions on Cybernetics*, http://dx.doi.org/10.1109/TCYB.2020.2985043.

Liu, Guo-Ping (2022). Coordinated control of networked nonlinear multi-agent systems using variable horizon learning predictors via cloud edge computing. *IEEE Transactions on Control of Network Systems*, http://dx.doi.org/10.1109/TCNS.2022.3181549.

Maxim, Anca, Copot, Dana, Copot, Cosmin, & Ionescu, Clara M. (2019). The 5w's for control as part of industry 4.0: Why, what, where, who, and when—a PID and MPC control perspective. *Inventions*, *4*(1), 10. http://dx.doi.org/10.3390/inventions4010010.

Ning, Boda, Han, Qing-Long, & Ding, Lei (2021). Distributed finite-time secondary frequency and voltage control for islanded microgrids with communication delays and switching topologies. *IEEE Transactions on Cybernetics*, *51*(8), 3988–3999. http://dx.doi.org/10.1109/TCYB.2020.3003690.

Ning, Boda, Han, Qing-Long, & Zuo, Zongyu (2019a). Distributed optimization for multiagent systems: An edge-based fixed-time consensus approach. *IEEE Transactions on Cybernetics*, *49*(1), 122–132. http://dx.doi.org/10.1109/TCYB.2017.2766762.

Ning, Boda, Han, Qing-Long, & Zuo, Zongyu (2019b). Practical fixed-time consensus for integrator-type multi-agent systems: A time base generator approach. *Automatica*, *105*, 406–414. http://dx.doi.org/10.1016/j.automatica.2019.04.013.

Oh, Kwang-Kyo, Park, Myoung-Chul, & Ahn, Hyo-Sung (2015). A survey of multi-agent formation control. *Automatica*, *53*, 424–440. http://dx.doi.org/10.1016/j.automatica.2014.10.022.

Pang, Zhonghua, Bai, Chuandong, Liu, Guoping, Han, Qinglong, & Zhang, Xianming (2021). A novel networked predictive control method for systems with random communication constraints. *Journal of Systems Science and Complexity*, *34*(4), 1364–1378. http://dx.doi.org/10.1007/s11424-021-0160-y.

Pang, Zhong-Hua, Luo, Wen-Cheng, Liu, Guo-Ping, & Han, Qing-Long (2020). Observer-based incremental predictive control of networked multi-agent systems with random delays and packet dropouts. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *68*(1), 426–430. http://dx.doi.org/10.1109/TCSII.2020.2999126.

Pang, Zhong-Hua, Zheng, Chang-Bing, Li, Chao, Liu, Guo-Ping, & Han, Qing-Long (2021). Cloud-based time-varying formation predictive control of multi-agent systems with random communication constraints and quantized signals. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *69*(3), 1282–1286. http://dx.doi.org/10.1109/TCSII.2021.3106694.

Qin, Jiahu, Ma, Qichao, Shi, Yang, & Wang, Long (2016). Recent advances in consensus of multi-agent systems: A brief survey. *IEEE Transactions on Industrial Electronics*, *64*(6), 4972–4983. http://dx.doi.org/10.1109/TIE.2016.2636810.

Sang, Zhiqian, & Xu, Xun (2017). The framework of a cloud-based CNC system. *Procedia Cirp*, *63*, 82–88. http://dx.doi.org/10.1016/j.procir.2017.03.152.

Sarpturk, S. Z., Istefanopulos, Yorgo, & Kaynak, Okyay (1987). On the stability of discrete-time sliding mode control systems. *IEEE Transactions on Automatic Control*, *32*(10), 930–932. http://dx.doi.org/10.1109/TAC.1987.1104468.

Sasaki, Kengo, Suzuki, Naoya, Makido, Satoshi, & Nakao, Akihiro (2016). Vehicle control system coordinated between cloud and mobile edge computing. In *2016 55th Annual conference of the society of instrument and control engineers of japan* (pp. 1122–1127). IEEE, http://dx.doi.org/10.1109/SICE.2016.7749210.

Shengdong, Mu, Zhengxian, Xiong, & Yixiang, Tian (2019). Intelligent traffic control system based on cloud computing and big data mining. *IEEE Transactions on Industrial Informatics*, *15*(12), 6583–6592. http://dx.doi.org/10.1109/TII.2019.2929060.

Steinberger, Martin, Horn, Martin, & Fridman, Leonid (2020). *Variable-structure systems and sliding-mode control*. Springer.

Tan, Haoran, Miao, Zhiqiang, Wang, Yaonan, Wu, Min, & Huang, Zhiwu (2020). Data-driven distributed coordinated control for cloud-based model-free multiagent systems with communication constraints. *IEEE Transactions on Circuits and Systems. I. Regular Papers*, *67*(9), 3187–3198. http://dx.doi.org/10.1109/TCSI.2020.2990411.

Tan, Haoran, Wang, Yaonan, Wu, Min, Huang, Zhiwu, & Miao, Zhiqiang (2021). Distributed group coordination of multiagent systems in cloud computing systems using a model-free adaptive predictive control strategy. *IEEE Transactions on Neural Networks and Learning Systems*, http://dx.doi.org/10.1109/TNNLS.2021.3053016.

Tan, Chong, Yue, Liang, Li, Yanjiang, & Liu, Guo-Ping (2020). Group consensus control for discrete-time heterogeneous multi-agent systems with time delays. *Neurocomputing*, *392*, 70–85. http://dx.doi.org/10.1016/j.neucom.2020.01.092.

Xia, Yuanqing (2015). Cloud control systems. *IEEE/CAA Journal of Automatica Sinica*, *2*(2), 134–142. http://dx.doi.org/10.1109/JAS.2015.7081652.

Xia, Yuanqing, Xie, Wen, Liu, Bo, & Wang, Xiaoyun (2013). Data-driven predictive control for networked control systems. *Information Sciences*, *235*, 45–54. http://dx.doi.org/10.1016/j.ins.2012.01.047.

Xu, Qingsong (2015). Digital integral terminal sliding mode predictive control of piezoelectric-driven motion system. *IEEE Transactions on Industrial Electronics*, *63*(6), 3976–3984. http://dx.doi.org/10.1109/TIE.2015.2504343.

Yu, Yi, Liu, Guo-Ping, & Hu, Wenshan (2022a). Coordinated distributed predictive control for voltage regulation of DC microgrids with communication delays and data loss. *IEEE Transactions on Smart Grid*, http://dx.doi.org/10.1109/TSG.2022.3208946.

Yu, Yi, Liu, Guo-Ping, & Hu, Wenshan (2022b). Security tracking control for discrete-time stochastic systems subject to cyber attacks. *ISA Transactions*, http://dx.doi.org/10.1016/j.isatra.2022.02.001.

Zhang, Xian-Ming, Han, Qing-Long, Ge, Xiaohua, Ding, Derui, Ding, Lei, Yue, Dong, et al. (2019). Networked control systems: A survey of trends and techniques. *IEEE/CAA Journal of Automatica Sinica*, *7*(1), 1–17. http://dx.doi.org/10.1109/JAS.2019.1911651.

Zhang, Xian-Ming, Han, Qing-Long, & Yu, Xinghuo (2015). Survey on recent advances in networked control systems. *IEEE Transactions on Industrial Informatics*, *12*(5), 1740–1752. http://dx.doi.org/10.1109/TII.2015.2506545.

Zhang, Dan, Shi, Peng, Wang, Qing-Guo, & Yu, Li (2017). Analysis and synthesis of networked control systems: A survey of recent advances and challenges. *ISA Transactions*, *66*, 376–392. http://dx.doi.org/10.1016/j.isatra.2016.09.026.

Zhao, Guanglei, Hua, Changchun, & Guan, Xinping (2019). Distributed event-triggered consensus of multiagent systems with communication delays: A hybrid system approach. *IEEE Transactions on Cybernetics*, *50*(7), 3169–3181. http://dx.doi.org/10.1109/TCYB.2019.2912403.

Zheng, Chang-Bing, Pang, Zhong-Hua, Wang, Jing-Xu, Sun, Jian, Liu, Guo-Ping, & Han, Qing-Long (2022). Null-space-based time-varying formation control of uncertain nonlinear second-order multi-agent systems with collision avoidance. *IEEE Transactions on Industrial Electronics*, 1–10. http://dx.doi.org/10.1109/TIE.2022.3217585.