Topology-Preserving Second-Order Consensus: A Strategic Compensation Approach

Zitong Wang, Yushan Li, Xiaoming Duan, and Jianping He

***Abstract*— The interaction topology plays a significant role in the collaboration of multi-agent systems. How to preserve the topology against inference attacks has become an imperative task for security concerns. In this paper, we propose a distributed topology-preserving algorithm for second-order multi-agent systems by adding noisy inputs. The major novelty is that we develop a strategic compensation approach to overcome the noise accumulation issue in the second-order dynamic process while ensuring the exact second-order consensus. Specifically, we design two distributed compensation strategies that make the topology more invulnerable against inference attacks. Furthermore, we derive the relationship between the inference error and the number of observations by taking the ordinary least squares estimator as a benchmark. Extensive simulations are conducted to verify the topology-preserving performance of the proposed algorithm.**

I. Introduction

Over the past few decades, researchers have focused on designing multi-agent systems (MASs) where agents with limited capacity coordinate with each other in a distributed manner and accomplish specific tasks. Distributed cooperative control has broad applications such as sensor networks [1], distributed computing [2], swarm flocking [3] and cooperative manipulation [4]. The importance of the interaction topology for distributed cooperative control is reflected in its impact on the autonomy, adaptation, scalability, and efficiency of the MAS [5]. As a result, it receives significant attention from researchers and brings up various studies focusing on estimating the interaction topology based on accessible observation data. This topology inference problem can be solved by various methods such as the Ordinary Least Squares (OLS) estimator [6], causality-based estimator [7], identification method [8], reinforcement learning [9], etc.

However, as the behavior of common MASs can be observed externally, the advanced topology inference methods may be leveraged by malicious adversaries to regress the interaction topology of the MAS, causing critical security breaches. For example, with knowledge of the topology, adversaries can predict the future states of vulnerable or critical agents and launch precise interceptions [10], or they can attack the communication links, thus paralyzing the system [11]. Such attacks can severely deteriorate the collaboration performance of the MASs. To address this security risk associated with topology inference attacks, it is of utmost importance to develop topology-preserving collaboration algorithms that can efficiently conceal the actual topology while maintaining the stability and cooperative performance of the MASs.

Technically, countering topology inference attacks requires well-designed modifications to the cooperative algorithms. Researchers have delved into the cooperative algorithm to address specific consensus-based needs, such as protecting the privacy of agents [12], [13], enhancing resilience against false data injection attacks [14], and improving robustness in handling intermittent communications and actuator faults [15]. Among these defense mechanisms, two main methods are commonly applied, namely dynamic topology and noise-adding algorithms. In the former approach, the interaction topology of the agents changes over time, significantly increasing the uncertainty of the states of the agents to enhance protection on the system [16]–[18]. Nevertheless, these methods usually have a strong dependence on the connectivity of topology and often present greater challenges to collaboration. On the other hand, noise-adding algorithms impose additional noisy signals on the states of the agents during the collaboration, thereby hiding the accurate state information from the adversaries [12], [13], [19], [20]. For example, the authors of [12] propose a differential privacy scheme based on Laplace noise to preserve the privacy of the states of the agents, and the authors of [13] introduce a noise-adding algorithm to preserve the privacy of the initial states while achieving exact average consensus in the sense of mean square convergence.

According to the aforementioned works, noise-adding methods are flexible and effective in addressing specific needs without a strong dependence on topology, making them promising for topology-preserving algorithm design. Following this idea, our latest work [21] has made prior efforts to preserve the topology of first-order MASs. However, this algorithm is not directly suitable for second-order systems, which have more practical applications such as flocking and formation control [22], [23]. The infeasibility lies in that the added noisy inputs in second-order systems or higher-order systems will accumulate in the states and deteriorate the system convergence. How to preserve the topology of second-order MASs still remains an open issue.

Motivated by the above observations, this paper focuses on designing a topology-preserving algorithm for second-order MASs by utilizing noise-adding methods. Specifically, we propose a distributed topology-preserving second-order consensus (TPSC) algorithm by designing strategic compensating inputs, which effectively conceals the actual topology

The authors are with the Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, China. E-mail address: {wangzitong, yushan_li, xduan and jphe}@sjtu.edu.cn. This work was supported by the Joint Tackling Plan of the Yangtze River Delta Science and Technology Innovation Community under Grant 2022CSJGG1400.

structure from the inference attack without sacrificing collaboration performance. The main challenge related to designing the algorithm is to retain exact second-order consensus performance in a distributed manner while defending against inference attacks as much as possible. The main contributions of our work are summarized as follows:

- We investigate the topology preservation problem in second-order MASs, and we propose a distributed algorithm that secures the interaction topology while guaranteeing collaboration performance.
- By exploiting the sufficient and necessary conditions on added noisy inputs for exact second-order consensus, we develop two strategic compensating input designs for the state-updating process of each node, without relying on any global system parameters.
- For the self-compensating strategies, we prove the exact second-order consensus of the MAS under the TPSC algorithm. The relationship between the convergence rate of the inference error and the number of observations is derived. Representative simulations demonstrate the effectiveness of the proposed TPSC algorithm.

Notation: Let $\mathbf{1}$ be an all-one column vector, and $\mathbf{0}$ be an all-zero column vector with compatible dimensions. Let $\mathbb{N}$ and $\mathbb{N}^+$ be the sets of non-negative integers and positive integers. Let $\mathbb{R}$ be the set of real numbers. Let $\|\cdot\|$ and $\|\cdot\|_F$ represent the spectral norm and Frobenius norm of a matrix, respectively. For two functions $f(x)$ and $g(x)$, $f(x) = \mathcal{O}(g(x)), x \to \infty$ mean that there exists a positive real number M and a real number x_0 such that $\|f(x)\| \leq Mg(x), \forall\ x \geq x_0$.

The rest of the paper is organized as follows: Section II provides some preliminary knowledge and formulates the problem. The proposed algorithm and its performance analysis are in Section III. Section IV shows the simulation results. Finally, Section V concludes the work.

II. PRELIMINARIES

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph that models the topology information within the multi-agent system, where $\mathcal{V} = \{1, \ldots, N\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges. Each node represents an agent, and each weighted edge represents an information transmission channel. The adjacency matrix $A_{\mathcal{G}} = [a_{ij}]_{N\times N}$ of a graph $\mathcal{G}$ with N agents specifies the interconnection topology of the system, where $a_{ij} > 0$ if and only if $(i,j) \in \mathcal{E}$, else $a_{ij} = 0$. Let $\mathcal{N}_i = \{j \,|\, (i,j) \in \mathcal{E}\}$ be the neighbor set of agent i and $d_i = |\mathcal{N}_i|$ be its in-degree. Define Laplacian matrix of $\mathcal{G}$ as $L_{\mathcal{G}} = D_{\mathcal{G}} - A_{\mathcal{G}}$, where $D_{\mathcal{G}}$ is the diagonal matrix of all d_is.

A. Second-order Consensus-based Algorithm

The consensus algorithm is widely applied in multi-agent systems for collaboration tasks, where the group of agents reach a consensus based on limited shared information exchanges [24]–[27]. In this section, we will introduce the second-order consensus-based formation algorithm. Consider a network represented by the graph $\mathcal{G}$ with N nodes. Each agent i follows a double-integrator dynamic given by

$$\dot{p}_i(t) = v_i(t), \dot{v}_i(t) = u_i(t), i = 1, 2, \ldots, N, \tag{1}$$

where $p_i(t) \in \mathbb{R}$ and $v_i(t) \in \mathbb{R}$ are the position and velocity of agent i at time $t \geq 0$, respectively, and $u_i(t) \in \mathbb{R}$ is the corresponding control input. Since in the real world, the control inputs are always applied at discrete sampling times, we discretize the dynamics with sampling period T [28]. The system (1) becomes:

$$\begin{aligned} p_i(k+1) &= p_i(k) + Tv_i(k) + \frac{T^2}{2}u_i(k), \\ v_i(k+1) &= v_i(k) + Tu_i(k), i = 1, \ldots, N, \end{aligned} \tag{2}$$

where $p_i(k), v_i(k), u_i(k)$ are position, velocity, control input for agent i at time $t = kT$, respectively. Define $\Delta_{ij} = \delta_i - \delta_j$ as the desired position deviation between agent i and agent j in a formation. To simplify the notation, we use the relative position $\tilde{p}_i(k) = p_i(k) - \delta_i$ of agent i in the rest of this paper. The following algorithm which considers the relative positions and velocities is adopted:

$$u_i(k) = -\sum_{j\in\mathcal{V}} a_{ij} \left[(\tilde{p}_i(k) - \tilde{p}_j(k)) + \alpha(v_i(k) - v_j(k)) \right], \tag{3}$$

where α is a positive scalar. The objective of the algorithm is second-order consensus in the sense that agents come to the desired formation pattern with the same velocity, i.e.,

$$\begin{cases} \lim_{k\to\infty}[p_i(k) - p_j(k)] = \Delta_{ij}, & \forall\ i,j \in \mathcal{V}, \\ \lim_{k\to\infty}[v_i(k) - v_j(k)] = 0, & \forall\ i,j \in \mathcal{V}. \end{cases} \tag{4}$$

The discrete-time system model (2) under the algorithm (3) can be written in the following matrix form:

$$\begin{bmatrix} \tilde{p}(k+1) \\ v(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} I_N - \frac{T^2}{2}L_{\mathcal{G}} & TI_N - \alpha\frac{T^2}{2}L_{\mathcal{G}} \\ -TL_{\mathcal{G}} & I_N - \alpha TL_{\mathcal{G}} \end{bmatrix}}_{G} \begin{bmatrix} \tilde{p}(k) \\ v(k) \end{bmatrix}, \tag{5}$$

where I_N is an identity matrix, $\tilde{p}(k) = [\tilde{p}_1(k), \cdots, \tilde{p}_N(k)]^\top$ is the concatenated relative position vector, and $v(k) = [v_1(k), \cdots, v_N(k)]^\top$ is the concatenated velocity vector.

Assumption 2.1: Assume graph $\mathcal{G}$ has a spanning tree and $L_{\mathcal{G}}$ has eigenvalues $\lambda_1 = 0$ and $0 < \lambda_2 \leq \cdots \leq \lambda_N$.

Lemma 2.1: (Theorem 4.1 in [28]) If Assumption 2.1 holds, the second-order consensus is achieved if and only if parameters α and T are chosen from the following set:

$$Q_r = \left\{ (\alpha, T) \,|\, \frac{T}{2} < \alpha < \frac{2}{\lambda_N T} \right\}. \tag{6}$$

The velocities and the relative positions of agents will converge as follows:

$$\lim_{k\to\infty} v_i(k) = \bar{v}(0), \qquad \forall\ i \in \mathcal{V}, \tag{7a}$$

$$\lim_{k\to\infty} \tilde{p}_i(k) = \bar{p}(0) + kT\bar{v}(0), \quad \forall\ i \in \mathcal{V}, \tag{7b}$$

where $\bar{p}(0)$ and $\bar{v}(0)$ are the mean values of initial relative positions and velocities of agents in the system, respectively.

Lemma 2.1 provides the necessary and sufficient conditions for convergence to second-order consensus. It is worth noting that (6) is not difficult to satisfy when T is much smaller than 1 and α is chosen properly.

The system model can be written as:

$$\begin{bmatrix}\tilde{p}(k)\\ v(k)\end{bmatrix} = G^k \begin{bmatrix}\tilde{p}(0)\\ v(0)\end{bmatrix}. \tag{8}$$

Agents seek to achieve a global consensus on their relative positions and velocities by leveraging the information shared by their local neighbors. As is shown in (8), this process is tightly coupled with the interaction topology among the neighboring agents. As a result, the changes in the relative positions and velocities can reveal important information about the underlying topology of the MAS.

B. Topology Inference Mechanism

The objective of the adversaries is to obtain the Laplacian matrix $L_{\mathcal{G}}$ of the graph $\mathcal{G}$. From (5), we know that:

$$\begin{aligned} &L_{\mathcal{G}}\left[\frac{T^2}{2}\tilde{p}(k)+\alpha\frac{T^2}{2}v(k)\right]=\tilde{p}(k)+Tv(k)-\tilde{p}(k+1),\\ &L_{\mathcal{G}}\left[T\tilde{p}(k)+\alpha Tv(k)\right]=v(k)-v(k+1).\end{aligned} \tag{9}$$

The above two equations are equivalent. For simplicity, we only use the second equation for topology inference.

Consider the scenario where adversaries collect the data from time 0 to time k and then use widely adopted optimization methods, such as the OLS estimator, to regress the topology. Let $y(k) = T\tilde{p}(k) + \alpha Tv(k)$ and $z(k) = v(k) - v(k+1)$. Stack the vectors and denote $Y(k) = [y(0), \cdots, y(k)]$ and $Z(k) = [z(0), \cdots, z(k)]$. Then, the topology inference problem is formulated as follows:

$$\min_{\hat{L}_{\mathcal{G}}(k)} \|\hat{L}_{\mathcal{G}}(k)Y(k) - Z(k)\|_F^2. \tag{10}$$

In the above equation, $\hat{L}_{\mathcal{G}}(k)$ is the inferred topology based on data from time 0 to time k. If matrix $Y(k)^\top$ has full column rank, which is generally the case, the optimal solution of (10) is given by $\hat{L}_{\mathcal{G}}(k)^\star = Z(k)Y(k)^\top(Y(k)Y(k)^\top)^{-1}$.

C. Problem Formulation

In this paper, we mainly consider how to conceal the actual topology of the MAS by adding inputs to the states of the agents. The regular algorithm (3) is revised to

$$u_i(k)=-\sum_{j\in\mathcal{V}}a_{ij}\left[(\tilde{p}_i(k)-\tilde{p}_j(k))+\alpha(v_i(k)-v_j(k))\right]+\theta_i(k),$$

and the system model (5) can be rewritten as

$$\begin{bmatrix}\tilde{p}(k+1)\\ v(k+1)\end{bmatrix} = G\begin{bmatrix}\tilde{p}(k)\\ v(k)\end{bmatrix} + \begin{bmatrix}\frac{T^2}{2}\theta(k)\\ T\theta(k)\end{bmatrix}, \tag{11}$$

where $\theta(k) \in \mathbb{R}^N$ is the vector of designed inputs at time k. It can be seen that the dynamics of the agents in (11) are affected not only by the topology information of the system but also by the designed inputs.

The objective of this paper is to develop a topology-preserving algorithm that can effectively prevent adversaries from inferring the topology correctly while guaranteeing the exact second-order average consensus in the MASs. Hence, we formulate the corresponding optimization problem:

$$\begin{aligned} &\max_{\theta}\min_{\hat{L}_{\mathcal{G}}(k)} \|\hat{L}_{\mathcal{G}}(k)Y(k) - Z(k)\|_F^2\\ &\text{s.t. (7a) and (7b) hold.}\end{aligned} \tag{12}$$

Our primary concern is designing optimal inputs that maximize the inference error for adversaries while ensuring accurate convergence. However, the optimal solution to this problem is determined by knowledge of the global topology. Since we aim to use a distributed approach to design inputs for each agent, we can only increase the inference error as much as possible within specific algorithms. Another challenge lies in that achieving second-order consensus is more complicated than first-order consensus. Therefore, the extra inputs must be designed carefully to avoid the accumulation effect of noisy inputs in second-order systems.

III. MAIN RESULTS

In this section, we propose the TPSC algorithm and analyze its performance. First, we demonstrate the key idea and details of the TPSC algorithm to address the challenges mentioned in Section II. Then, we present the performance analysis of the TPSC algorithm, including the convergence analysis and the inference error analysis.

A. Algorithm Design

To fulfill the requirements of the formulated problem, the TPSC algorithm consists of two major parts: inject additive inputs thus enlarging the regression errors, and add compensating inputs to ensure the convergence of the MAS. Instead of adding random inputs with a fixed distribution, we add well-designed inputs to the agents, which can ensure exact convergence.

As mentioned in Section II, the subsequent state of an agent depends on the current states of its neighbors and itself. In this way, the input of a particular agent will spread its influence through the interaction topology, thereby affecting the performance of the entire system. To fulfill the consensus requirement, the conditions proposed by the following lemma must be satisfied.

Lemma 3.1: The second-order average consensus in (11) is achieved when the following conditions are satisfied:

$$\begin{aligned} &\lim_{k\to\infty}\left\{\sum_{l=0}^{k}\sum_{i=1}^{N}\theta_i(l)\right\}=0,\\ &\lim_{k\to\infty}\left\{\sum_{l=0}^{k}\sum_{i=1}^{N}\left[\left(k-l+\frac{1}{2}\right)\theta_i(l)\right]\right\}=0.\end{aligned} \tag{13}$$

This lemma shows the complexity of achieving exact average consensus in second-order systems and intuitively shows that the extra inputs have an accumulation effect on positions and velocities over time.

Denote $a_i(k) \in [-\varphi^k, \varphi^k], 0 < \varphi \leq 1$ as a decision variable whose precise value will be determined later. Denote $b_i(k)$ as the additive input indicator that follows a Bernoulli distribution, given by

$$\Pr\{b_i(k) = 1\} = \epsilon,\ \ \Pr\{b_i(k) = 0\} = 1 - \epsilon,\ \ 0 \leq \epsilon \leq 1.$$

Based on Lemma 3.1, we propose two strategy examples (SE 1 and SE 2) that satisfy (13), thereby guaranteeing consensus of the system. In both examples, the additive input $\omega_i(k \,|\, k)$ is added to the i-th agent at time k if $b_i(k) = 1$. To balance the effect of this additive input on convergence, compensating inputs $\omega_i(k + l \,|\, k), l \in \mathbb{N}^+$ are imposed after several iterations.

SE 1:

$$\begin{cases} \omega_i(k \,|\, k) = a_i(k) \\ \omega_i(k + \tau_m \,|\, k) = -\frac{2}{\tau_e - 1} a_i(k), \;\; \tau_m = 1, 2, \ldots, \tau_e - 1 \\ \omega_i(k + \tau_e \,|\, k) = a_i(k), \;\; \tau_e \in \mathbb{N}^+ \end{cases}$$

SE 2:

$$\begin{cases} \omega_i(k \,|\, k) = a_i(k) \\ \omega_i(k + \tau_m \,|\, k) = -\frac{\tau_e}{\tau_e - \tau_m} a_i(k), \;\; \tau_m \in \mathbb{N}^+ \\ \omega_i(k + \tau_e \,|\, k) = \frac{\tau_m}{\tau_e - \tau_m} a_i(k), \;\; \tau_e \in \mathbb{N}^+ \text{ and } \tau_m < \tau_e \end{cases}$$

In the above examples, τ_m and τ_e stand for the compensating time in the middle of the input sequence and in the end of the sequence, respectively. Note that $\tau_e > 2$ is a variable that can be fixed manually or randomized within a specified range. In SE 1, τ_m and the amplitude of the compensating inputs are determined by $a_i(k)$ and τ_e. In SE 2, τ_m is randomly chosen, and the amplitude of the compensating inputs is also affected by τ_m. In general, the expression of $\theta_i(k)$ is

$$\theta_i(k) = \sum_{l=0}^{k} \omega_i(k \,|\, k - l) b_i(k - l). \tag{14}$$

It is worth noting that based on Lemma 3.1, more strategies for the proposed algorithm can be developed, and this paper only presents two possible designs. The central idea of these designs is similar. In order to avoid redundancy, we mainly focus on the analysis and simulation of SE 1 in this paper. Specifically, the details of the TPSC algorithm with SE 1 are illustrated in Algorithm 1 where k_0 is the terminal time of adding additive inputs.

Algorithm 1: Topology-Preserving Second-Order Consensus (TPSC) Algorithm

Input: $G, T, k_0, \tilde{p}(0), v(0), \epsilon, \varphi$;
Output: Observation data set;
Initialization;
for $k = 0, 1, \ldots$ **do**
  if $k < k_0$ **then**
    for $i = 1, \cdots, N$ **do**
      Generate $b_i(k)$ and τ_e;
      if $b_i(k) = 1$ **then**
        Determine $a_i(k) \in [-\varphi^k, \varphi^k]$ by (18);
        $\omega_i(k \,|\, k) = a_i(k)$;
        for $\tau_m = 1, \cdots, \tau_e - 1$ **do**
          $\omega_i(k + \tau_m \,|\, k) =$
            $-2/(\tau_e - 1) \times a_i(k)$;
        end
        $\omega_i(k + \tau_e \,|\, k) = a_i(k)$;
      end
      Calculate $\theta_i(k)$ by (14);
    end
  end
  Update $\tilde{p}(k + 1)$ and $v(k + 1)$ by (11);
end

B. Convergence Analysis

When the TPSC algorithm is applied to the MASs, the added inputs to the agents will confuse not only the adversaries but also the agents in the neighborhood. To ensure the cooperative performance of the system, an exact convergence to the second-order consensus must be guaranteed.

Theorem 3.1: Given any $\tilde{p}(0)$ and $v(0)$, an exact second-order consensus is achieved using the TPSC algorithm, i.e., (7a) and (7b) hold.

Proof: The system under our algorithm can be rewritten in the following way:

$$\begin{bmatrix} \tilde{p}(k) \\ v(k) \end{bmatrix} = G^k \begin{bmatrix} \tilde{p}(0) \\ v(0) \end{bmatrix} + \sum_{l=0}^{k} G^{k-l} \begin{bmatrix} \frac{T^2}{2} \theta(l) \\ T\theta(l) \end{bmatrix}. \tag{15}$$

Generally, we have

$$\lim_{k \to \infty} \sum_{l=0}^{k_0} G^{k-l} \begin{bmatrix} \frac{T^2}{2} \theta(l) \\ T\theta(l) \end{bmatrix} = \lim_{k \to \infty} \sum_{l=0}^{k_0} \begin{bmatrix} \left(k - l + \frac{1}{2}\right) T^2 \bar{\theta}(l) \\ T\bar{\theta}(l) \end{bmatrix},$$

where $\bar{\theta}(l) = \frac{1}{N} \sum_{i=1}^{N} \theta_i(l) \mathbf{1}$. The elements of the above vector can be written as

$$\begin{aligned} &\lim_{k \to \infty} \sum_{l=0}^{k_0} \left[\left(k - l + \frac{1}{2}\right) T^2 \theta_i(l) \right] \\ &= \lim_{k \to \infty} \sum_{l=0}^{k_0} \left[\left(k - l + \frac{1}{2}\right) T^2 \sum_{x=0}^{l} \omega_i(l \,|\, x) b_i(x) \right] \\ &= \lim_{k \to \infty} \sum_{x=0}^{k_0} \sum_{l=x}^{k_0} \left(k - l + \frac{1}{2}\right) T^2 \omega_i(l \,|\, x) b_i(x). \end{aligned}$$

Similarly, it can be derived that

$$\lim_{k \to \infty} \sum_{l=0}^{k_0} T\theta_i(l) = \lim_{k \to \infty} \sum_{x=0}^{k_0} \sum_{l=x}^{k_0} T\omega_i(l \,|\, x) b_i(x).$$

With SE 1, we obtain

$$\begin{cases} \displaystyle\sum_{l=x}^{k_0} \left(k - l + \frac{1}{2}\right) T^2 \omega_i(l \,|\, x) b_i(x) = 0, & \forall x \in \mathbb{N}, \\ \displaystyle\sum_{l=x}^{k_0} T\omega_i(l \,|\, x) b_i(x) = 0, & \forall x \in \mathbb{N}. \end{cases}$$

The above equations lead to

$$\lim_{k \to \infty} \sum_{l=0}^{k_0} G^{k-l} \begin{bmatrix} \frac{T^2}{2} \theta(l) \\ T\theta(l) \end{bmatrix} = \mathbf{0}.$$

Therefore, it can be concluded that

$$\lim_{k \to \infty} \begin{bmatrix} \tilde{p}(k) \\ v(k) \end{bmatrix} = \lim_{k \to \infty} G^k \begin{bmatrix} \tilde{p}(0) \\ v(0) \end{bmatrix}, \tag{16}$$

which completes the proof. ∎

C. Inference Error Analysis

To keep the adversaries from inferring the actual topology, we need to enlarge the regression error:

$$\|\hat{L}_{\mathcal{G}}(k) - L_{\mathcal{G}}\|_F^2. \tag{17}$$

As mentioned in Section II, the optimal solution of the OLS estimator is $\hat{L}_{\mathcal{G}}(k)^\star = Z(k)Y(k)^\top(Y(k)Y(k)^\top)^{-1}$. Then the deviation of the topology inference can be described as $E_{\hat{L}_{\mathcal{G}}(k)} = \hat{L}_{\mathcal{G}}(k) - L_{\mathcal{G}} = \Theta(0;k)Y(k)^\top(Y(k)Y(k)^\top)^{-1}$, where $\Theta(0;k) = [T\theta(0), \cdots, T\theta(k)]$. In terms of the choice of the amplitude of $\omega_i(k\,|\,k)$, the problem can be formulated as a constrained optimization problem where the additive input $\omega_i(k\,|\,k)$ is selected to maximize the inference error:

$$\begin{aligned} \text{P1:}\quad & \max_{\theta(k)} \left\|\Theta(0;k)Y(k)^\top(Y(k)Y(k)^\top)^{-1}\right\|_F \\ & \text{s.t.} \quad -\varphi^k \le \omega_i(k\,|\,k) \le \varphi^k. \end{aligned} \tag{18}$$

Theorem 3.2: (Policy of $\omega_i(k\,|\,k)$ design) The optimal solution of P1 in (18) equals to one of the constrained boundaries, either $-\varphi^k$ or φ^k.

Proof: Define $\Upsilon = Y(k)^\top(Y(k)Y(k)^\top)^{-1}$ and split it into Υ_A whose size is $k \times N$ and Υ_B whose size is $1 \times N$. Split $\Theta(0;k)$ into $\Theta(0;k-1)$ and $\theta(k)$. Thus the optimized objective function of (18) can be broken down as

$$\begin{aligned} & \left\| [\Theta(0;k-1)\,|\,\theta(k)] \left[\frac{\Upsilon_A}{\Upsilon_B}\right] \right\|_F \\ = & \sqrt{\sum_{i=1}^{N} \|(\Theta_i(0;k-1)\Upsilon_A + \theta_i(k)\Upsilon_B)\|_F^2}. \end{aligned}$$

Hence, P1 can be decomposed into N independent sub-optimization problems, i.e., the Frobenius norm optimization problems of each row, which are given by

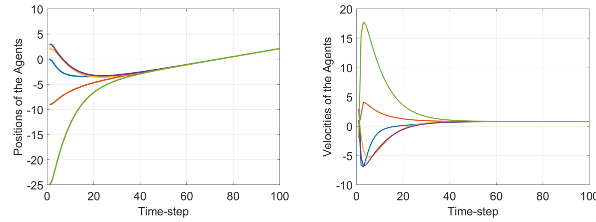
$$\begin{aligned} \text{P2:}\quad & \max_{\theta_i(k)} \|\Theta_i(0;k-1)\Upsilon_A + \theta_i(k)\Upsilon_B\|_F \\ & \text{s.t.} \quad -\varphi^k \le \omega_i(k\,|\,k) \le \varphi^k. \end{aligned} \tag{19}$$

The objective function of P2 in (19) is a convex quadratic function of $\theta_i(k)$. Therefore, it is maximized when the component $\omega_i(k\,|\,k)$ equals one of the restrictions. Furthermore, as the Frobenius norm of each row is independent and always positive, the overall target of (18) is accomplished if and only if each optimization problem in (19) is maximized. ■

This theorem provides the step-by-step optimal choice for each agent when adding additive inputs in a distributed network. In practice, agents cannot access global data $Y(k)$ and determine the ideal boundary. In this way, each agent can randomly select an amplitude for the additive input between the boundaries.

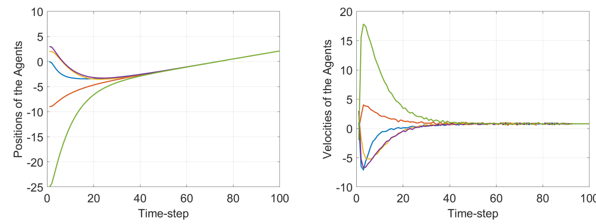
Theorem 3.3: Applying the TPSC algorithm to the system (2), the non-asymptotic error bound of the OLS estimator is characterized by:

$$\begin{cases} \lim\limits_{k_0\to\infty} \mathbb{E}\left[\left\|E_{\hat{L}_{\mathcal{G}}(k_0)}\right\|_F\right] = \mathcal{O}\left(\dfrac{4\epsilon TN}{(\tau_e+1)\sqrt{1-\varphi^2}}\right), & \varphi < 1, \\ \lim\limits_{k_0\to\infty} \mathbb{E}\left[\left\|E_{\hat{L}_{\mathcal{G}}(k_0)}\right\|_F\right] = \mathcal{O}\left(\dfrac{4\epsilon TN}{\tau_e+1}\right), & \varphi = 1. \end{cases}$$



(a) Positions of agents (b) Velocities of agents

Fig. 1. The positions and the velocities of agents under the algorithm (3)



(a) Positions of agents (b) Velocities of agents

Fig. 2. The positions and the velocities of agents under the TPSC algorithm

Proof: We consider the compact singular value decomposition $Y(k_0) = U\Sigma V^\top$, where $U \in \mathbb{R}^{N\times N}$ is a unitary matrix and $V \in \mathbb{R}^{(k_0+1)\times N}$ is a semi-unitary matrix. Note that we have $\Delta_G(k_0) = \Theta(0;k_0)Y(k_0)^\top(Y(k_0)Y(k_0)^\top)^{-1}$, which implies that

$$\|\Delta_G(k_0)\| \le \sqrt{1/\lambda_{\min}(Y(k_0)Y(k_0)^\top)}\, \|\Theta(0;k_0)V\|.$$

The value of φ affects the variation in the amplitude of the noisy inputs, leading to different convergence rates in $\lambda_{\min}(Y(k_0)Y(k_0)^\top)$. It can be concluded that

$$\sqrt{1/\lambda_{\min}(Y(k_0)Y(k_0)^\top)} = \begin{cases} \mathcal{O}(1), & \varphi < 1, \\ \mathcal{O}\left(1/\sqrt{k_0}\right), & \varphi = 1. \end{cases}$$

Based on SE 1, the expectation of $\|\Theta(0;k_0)\|_F$ can be written as $\mathbb{E}[\|\Theta(0;k_0)\|_F] = 4\epsilon TN\sqrt{\varphi^0 + \cdots + \varphi^{2k_0}}/(\tau_e+1)$, leading to the following equations

$$\begin{cases} \mathbb{E}[\|\Theta(0;k_0)\|_F] = \dfrac{4\epsilon TN}{(\tau_e+1)\sqrt{1-\varphi^2}}, & \varphi < 1, \\ \mathbb{E}[\|\Theta(0;k_0)\|_F] = \dfrac{4\epsilon TN\sqrt{k_0}}{\tau_e+1}, & \varphi = 1, \end{cases} \tag{20}$$

which complete the proof. ■

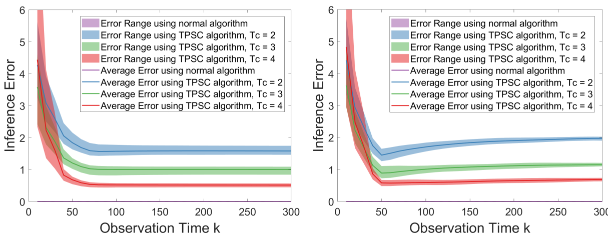
IV. SIMULATION

A. Simulation Setting

In this section, we verify the effectiveness of the TPSC algorithm via simulations. An undirected graph with five agents that represents the interaction topology of the system is randomly constructed. Assign all the agents with specific initial states, and start the iteration as in Algorithm 1.

B. Results and Analysis

Fig. 1 and Fig. 2 depict changes in the positions and velocities of agents in the MAS during the consensus progress under the normal algorithm and the TPSC algorithm, respectively. It can be seen in the figures that the proposed



(a) $\varphi < 1$ (b) $\varphi = 1$

Fig. 3. The error performance of the TPSC algorithm

algorithm ensures exact second-order consensus, although positions and velocities may fluctuate due to the extra inputs.

Fig. 3 illustrates the inference error for the adversaries when the proposed algorithm with SE 1 is adopted. Firstly, the topology of the MAS can be accurately inferred when the normal algorithm is applied (shown by the purple lines on the x-axis). In contrast, the TPSC algorithm effectively enlarges the inference error. Fig. 3(a) and Fig. 3(b) depict inference errors for the adversaries when the proposed TPSC algorithms with $\varphi < 1$ and $\varphi = 1$ are adopted, respectively. In the former case, the curves exhibit a modest decrease and converge to a constant value, while in the latter, the curves display a turning point at around $k = 50$ and also converge to a constant. Before reaching this threshold, limited data is the primary factor that affects the inference accuracy, while after this point, the influence of the inputs becomes dominant.

To conclude, the simulation results demonstrate that the TPSC algorithm performs well in addressing the topology preservation problem for second-order systems.

V. Conclusion

In this work, we focus on the topology preservation problem in second-order MASs. To address this problem, we propose the TPSC algorithm and design extra inputs for agents to prevent the adversaries from performing topology inference attacks while guaranteeing the exact second-order consensus. The convergence analysis and inference error analysis under the proposed algorithm are given. Extensive simulations are conducted to verify the effectiveness of the TPSC algorithm. Future research directions include expanding our study to more general networks, such as higher-order multi-agent systems with switching networks and exploring algorithms that adapt to other topology inference methods.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] A. D. Kshemkalyani and M. Singhal, *Distributed computing: Principles, algorithms, and systems*. Cambridge University Press, 2011.

[3] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, pp. 1–41, 2013.

[4] F. Caccavale and M. Uchiyama, "Cooperative manipulation," in *Springer Handbook of Robotics*, 2nd ed., B. Siciliano and O. Khatib, Eds. Springer, 2016, ch. 39, pp. 989–1006.

[5] Y. Emam, S. Mayya, G. Notomista, A. Bohannon, and M. Egerstedt, "Adaptive task allocation for heterogeneous multi-robot teams with evolving and unknown robot capabilities," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 7719–7725.

[6] Y. Li, J. He, L. Cai, and X. Guan, "Local topology inference of mobile robotic networks under formation control," *IEEE Transactions on Automatic Control*, 2023.

[7] Y. Li and J. He, "Topology inference for networked dynamical systems: A causality and correlation perspective," in *IEEE Conference on Decision and Control*, 2021, pp. 1218–1223.

[8] H. J. van Waarde, P. Tesi, and M. K. Camlibel, "Topology identification of heterogeneous networks: Identifiability and reconstruction," *Automatica*, vol. 123, p. 109331, 2021.

[9] Y. Du, B. Liu, V. Moens, Z. Liu, Z. Ren, J. Wang, X. Chen, and H. Zhang, "Learning correlated communication topology in multi-agent reinforcement learning," in *International Conference on Autonomous Agents and Multi-Agent Systems*, 2021, pp. 456–464.

[10] J. Li, J. He, Y. Li, and X. Guan, "Unpredictable trajectory design for mobile agents," in *American Control Conference*, 2020, pp. 1471–1476.

[11] M. Taheri, K. Khorasani, I. Shames, and N. Meskin, "Undetectable cyber attacks on communication links in multi-agent cyber-physical systems," in *IEEE Conference on Decision and Control*, 2020, pp. 3764–3771.

[12] T. Dong, X. Bu, and W. Hu, "Distributed differentially private average consensus for multi-agent networks by additive functional laplace noise," *Journal of the Franklin Institute*, vol. 357, no. 6, pp. 3565–3584, 2020.

[13] Y. Mo and R. M. Murray, "Privacy preserving average consensus," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 753–765, 2016.

[14] W. Zheng, C. Fang, J. He, and Y. Peng, "Resilient average consensus of second-order multi-agent systems," in *American Control Conference*, 2022, pp. 1466–1471.

[15] M. Sader, Z. Chen, Z. Liu, and C. Deng, "Distributed robust fault-tolerant consensus control for a class of nonlinear multi-agent systems with intermittent communications," *Applied Mathematics and Computation*, vol. 403, p. 126166, 2021.

[16] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.

[17] W. Ni and D. Cheng, "Leader-following consensus of multi-agent systems under fixed and switching topologies," *Systems & Control Letters*, vol. 59, no. 3-4, pp. 209–217, 2010.

[18] G. Wen, G. Hu, W. Yu, J. Cao, and G. Chen, "Consensus tracking for higher-order multi-agent systems with switching directed topologies and occasionally missing control inputs," *Systems & Control Letters*, vol. 62, no. 12, pp. 1151–1158, 2013.

[19] H. Sun, Z. Wang, J. Xu, and H. Zhang, "Exact consensus error for multi-agent systems with additive noises," *Journal of Systems Science and Complexity*, vol. 33, no. 3, pp. 640–651, 2020.

[20] V. Katewa, A. Chakrabortty, and V. Gupta, "Protecting privacy of topology in consensus networks," in *American Control Conference*, 2015, pp. 2476–2481.

[21] Z. Wang, Y. Li, C. Fang, and J. He, "Distributed topology-preserving collaboration algorithm against inference attack," in *2022 American Control Conference*, 2022, pp. 2166–2171.

[22] X. Dong, Y. Zhou, Z. Ren, and Y. Zhong, "Time-varying formation tracking for second-order multi-agent systems subjected to switching topologies with application to quadrotor formation flying," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 5014–5024, 2016.

[23] S. Yazdani and M. Haeri, "Flocking of multi-agent systems with multiple second-order uncoupled linear dynamics and virtual leader," *IET Control Theory & Applications*, vol. 10, no. 8, pp. 853–860, 2016.

[24] J. Qin, H. Gao, and W. X. Zheng, "Second-order consensus for multi-agent systems with switching topology and communication delay," *Systems & Control Letters*, vol. 60, no. 6, pp. 390–397, 2011.

[25] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[26] W. Yu, G. Chen, M. Cao, and J. Kurths, "Second-order consensus for multiagent systems with directed topologies and nonlinear dynamics," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 3, pp. 881–891, 2009.

[27] W. Ren, K. Moore, and Y. Chen, "High-order consensus algorithms in cooperative vehicle systems," in *IEEE International Conference on Networking, Sensing and Control*, 2006, pp. 457–462.

[28] W. Ren and Y. Cao, "Convergence of sampled-data consensus algorithms for double-integrator dynamics," in *IEEE Conference on Decision and Control*, 2008, pp. 3965–3970.