International Journal Of Industrial Electronics Control and Optimization
Print ISSN: 3517-2645
Online ISSN: 3568-2645
Homepage: https://ieco.usb.ac.ir/

Connectivity Maintenance of Nonlinear Multi-Agent Robotic System Subject to Minimizing Channel Path Attenuation

Sorush Akhlaghi [1] | Naser Pariz [1] | Mohammad Bagher Naghibi Sistani [1]

Engineering Faculty, Electrical Department, Ferdowsi University of Mashhad[1]
Corresponding author's email: n-pariz@um.ac.ir

Article Info

Article type:
Research Article

Article history:
Received: 07-February-2024
Received in revised form: 23-Jun-2024
Accepted: 20-July-2024
Published online: 21-Dec-2024

Keywords:
Multi-agent,
Connectivity maintenance,
Channel fading,
Sliding mode control.

ABSTRACT

In this paper, we address the problem of connectivity maintenance for a group of robots connected via wireless communication. We consider fading in the wireless channel between agents, where the exchange of information is possible at permissible distances. A distributed controller is designed to maintain the global connectivity of the network communication graph. On the other hand, obtaining the best communication quality between agents is the definition of good network connectivity. Therefore, the Laplacian matrix is defined as a weighted graph according to the communication parameters. Initially, the controller uses the supergradient algorithm to remake the network Laplacian matrix for having a greater second eigenvalue. While each agent in the network only has access to the neighbor's information. The control algorithm uses a multi-agent estimator to find the eigenvector corresponding to the second small eigenvalue. The formation problem has been considered in the second step after the reference topology has been obtained. Because of the nonlinear dynamics of the agents, the sliding mode controller has been used for this purpose. This robust controller could be a suitable choice due to the modeling and sensor measurement uncertainties. Finally, an example of multi-agent robots is provided to evaluate the algorithm.

I. Introduction

With the advancement of communication networks, multi-agent systems have been considered in many types of research and applications [1],[2],[3]. In these systems, independent units are interconnected and work together for various purposes. They consist of autonomous units, related to each other and working together to meet a common goal. e.g., the participation of a series of exploration robots or mobile sensor networks in an unknown area [4],[5]. Usually, in multi-agent systems, there are some limitations, and information is not always fully available to all agents. In a group of robots, it is assumed that each agent has access to data from its neighbors at certain intervals only depending on its place and conditions. Therefore, achieving network goals such as average consensus requires the permanent connection of the robot's set in a network.

Because of different plans that can be considered for agents and the existence of disturbances in the workplace, changes in network topology are always likely. In these systems, communication is limited only in the range of specific distances due to hardware limitations, and the movement of agents can cause changes in the communication graph [6],[7]. This problem makes it necessary to set a controller to keep the distances of the agents in an acceptable range. In this case, the controller is designed by assuming the initial connection of the communication graph to prevent the separation or collision of agents over time [8]. In some cases, arranging robots at certain intervals in a rigid network is the primary purpose [9],[10].

Keeping agents at a constant distance is not possible in many applications. For instance, in some cases, secondary goals in the group or obstacle avoidance purposes can make disturbing and stubborn agents [11],[12]. So, the continued maintenance of initial distances imposes many limitations on the agents. However, achieving global connectivity is usually a higher priority than local connectivity. Therefore, in many papers, the connectivity of agents has been studied and controlled according to global characteristics. The second small eigenvalue of the Laplacian matrix is used in the network connection graph to assess global communication [13]. In [14], connectivity in the graph is described as a Semidefinite Programming (SDP) problem. The optimal topology is determined for this optimization problem by considering some constraints.

Connectivity maintenance in distance-based networks has been reviewed in [15] which the states of agents are set to form the best possible communication graph. Due to the lack of access to information for each node in the distributed multi-agent systems, estimating the eigenvalues or eigenvectors is necessary for the global connectivity maintenance process. In [16], the distributed PI estimator and in [17], the decentralized Orthogonal Iteration Algorithm are used for this purpose. The dynamics of the agents in the multi-agent system can be different in modeling (linear or nonlinear). Accordingly, there are used lots of methods like [18],[19],[20] to maintain the connectivity of graph communications.

Multi-agent networks also have a great variety in terms of the type of agent communication. Numerous studies have been carried out on major problems such as packet loss and time delay [21]. In [22], the consensus problem in a multi-agent system is addressed, and the permissible delay boundaries are determined. Also, in [23], the formation control for a second-order multi-agent network, despite the delay in the channel, is designed. In [24], communication links are considered with the possibility of disruption, and the consensus problem is analyzed.

The use of wireless communication networks to exchange information in multi-agent systems is widely used [25],[26]. In this case, communication signals are always at risk of being lost, and part of it is usually converted to heat or absorbed by the environment. So, The received signal power at the receiver will differ from the transmitted power depending on the environment. One of the conventional models in wireless communication is the fading channel. The fading channel is usually modeled as a random time process. Many factors, such as geographic area and radio frequency, affect the fading channel.

Shadow and multipath propagation are two phenomena observed in the fading channel. Rayleigh model [27] is one of the models used in fading channels. In this model, changes and attenuation of the signal (communication quality) are modeled over time by changing the path loss power. So the fading phenomenon is related to the path losses that are increased by distance. In [28], fading channels are used in multi-agent systems, and sufficient condition for consensusability is given. Multipath propagation may cause some information to be lost, and the packet loss rate increases in the network.

In this paper, the connectivity problem for nonlinear multi-agent systems under fading channels is investigated. Due to the fading, the exchange of information is associated with errors, and it causes data loss. In other words, changing the distances between agents will change the channel's specification. So, we can define the connectivity maintenance problem based on achieving the best formation of agents. In this case, connectivity means placing the agents at desired distances to have the lowest information loss rate. With the distributed estimator, the controller in each agent determines the ideal distances to reach the best conditions over time. A robust sliding mode controller was used due to the nonlinear dynamics of the agents and the presence of uncertainties in modeling.

Additionally, this paper considers the fading effect in the communication between agents. In such a communication channel, the signal appears with varying degrees of attenuation, which can generally lead to non-convergence of the multi-agent system. The multi-agent system must be capable of establishing or strengthening effective communications between agents when necessary to maintain connectivity, thereby avoiding non-convergence or slower, weaker convergence; For this purpose, a sliding mode controller is used in this paper.

The remainder of the paper is organized as follows. Section 2 presents the Problem Formulation and introduces signal fading in the wireless communication network. It also describes the optimal topology to achieve the best connectivity conditions and provides the optimization solution with the supergradient algorithm. In Section 3, the design of the distributed controller is considered. A sliding mode controller is used to achieve the reference topology over the undirected graph, despite the presence of the fading effect.

The necessary and sufficient conditions for stability are also studied. In Section 4, the algorithm is tested by using numeric simulation in MATLAB software. Concluding remarks are drawn in Section 5.

II. Problem Formulation

A network consisting of N mobile robots is assumed with the directional graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that the robots form the graph nodes, and there is an edge between each connected robot. The set of nodes is $\mathcal{V} = \{1.2 \dots . n\}$ and the set of edges $\mathcal{E}$ consists of two subsets of $\mathcal{V}$ from the beginning and end of each edge. The existence of an edge between pairs $(i.j)$ is possible if j is at a certain distance.

$r_{ij} = \|x_i - x_j\|$ is the distance between two agents i and j in two dimensions, which can be connected in a wireless communication channel. Regardless of the disturbing agents in the communication channel, here, the phenomenon of communication path losses will be considered as a criterion for evaluating the quality of network communication. Path loss or path attenuation refers to the reduction in the power density of an electromagnetic wave as it propagates through space. Eq.(1) is used to compute the path loss.

$$P_{ij} = P_{L0} + 10\gamma \log(10) \left(\frac{r_{ij}}{r_0}\right) \tag{1}$$

In this equation, P_{L0} is the initial value of the path loss interference in reference distance r_0 . γ is the path loss exponent, and r_{ij} is the current distance between two agents in each dimension. Fig.1 shows this function relative to the changes in r_{ij}. For each agent within a range of less than r_0, the path loss power is assumed to be the lowest, and for further distances, this value increases slowly.

In this paper, the path losses, which are proportional to the network hardware equipment in the transmitter and receiver of an agent, are considered only in proportion to the distance of the agent to its neighbors. As a result, the adjacency matrix A, which represents the matrix of connections between robots in the graph is: $[\mathcal{A}(\mathcal{G})]_{ij} = P_{L0}/P_{ij}$.

The neighbors of ith agent are a subset of agents located in a circle with radius ρ and the center of ith agent. In other words, the C_i neighborhood is:

$$C_i \triangleq \{ j\epsilon\mathcal{V}, j \neq i : r_{ij} = \|x_i - x_j\| \leq \rho\} \tag{2}$$

Eq.(3) shows the Laplacian matrix $\mathcal{L}(z)$, which represents the network communication graph,

$$\mathcal{L}(x) = \mathcal{D}(x) - \mathcal{A}(x) \tag{3}$$

$\mathcal{D}(x)$ is the diagonal matrix and $\mathcal{D}_i = \sum_{j=1\ldots N} \mathcal{A}_{ij}$ are the main diagonal elements.

The dynamic of each agent in the network where $x_i \epsilon R^2$ and $u_i \epsilon R^2$ is as follows,

$$\dot{x}_i(t) = f(x_i) + u_i(t) \tag{4}$$

The state vector for all network agents is: $x_i = (y_{1i}, y_{2i})$, and $f(x_i) = [f_1(y_{1i}) \quad f_2(y_{2i})]^T$ is a generic nonlinear function of x_i and $\|f(x_i)\| \leq \widetilde{\Upsilon}$, $where\ \widetilde{\Upsilon}\ is\ a\ positive\ scaller\ constant.$

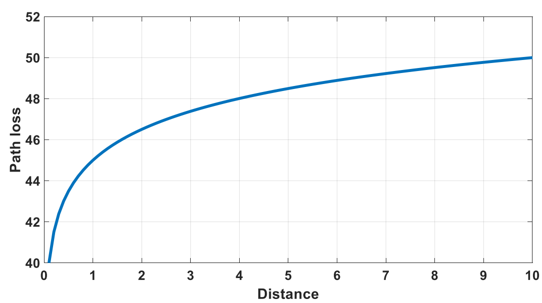

Fig. 1. The function $P(r_{ij})$ for $\gamma = 0.5, P_{L0} = 40, r_0 = 1$.

In an actual multi-agent system, agents often receive only partial or weakened information from their neighboring agents, rather than precise data. This occurrence can be described by the random attenuation of real-state information, meaning the true information is scaled by a random variable ranging between 0 and 1. This phenomenon is known as signal fading . In this study, signal fading is observed during transmission between followers, as described by the following fading model:

$$x_{ij} = \xi_{ij}\, x_j \tag{5}$$

Where x_{ij} represents the signal received by the i-th agent from its neighboring agent j. The fading effect of the transmission channel from agent j to agent i is captured by the random variable ξ_{ij}, which falls within the range (0, 1) and has a mean value of $\bar{\xi}$.

A. *Optimal Laplacian matrix*

Finding an optimal structure for the best second eigenvalue of the Laplacian matrix in the network, $\lambda(\mathcal{L})$, is considered in this Section. According to Eq.(1) and Eq.(2), this problem is expressed as follows,

$$\max_{L} \lambda_2(\mathcal{L}) \tag{6}$$

$\mathcal{L}^*$ is the answer to the above problem and determines the optimal topology During the process, this matrix is considered independent of states. It`s noted that $\lambda_2(\mathcal{L})$ is a concave function [15].

Hence the optimization problem (5) can be solved in the form of an SDP (Semidefinite program) problem. Here, the solution to the problem is in distributed form due to the necessity of using local control commands in agents. In other words, it is not possible to know the location of all agents for each of them, and every agent is able to make decisions using only the information of neighbors. The distributed supergradient optimization algorithm [29] can be a good choice for use in this algorithm.

Definition: Let F be a concave function from R^N to R. If $y \neq x$ and,

$$F(y) \leq F(x) + G^T(y - x) \tag{7}$$

The vector G is the supergradient of F in x. We can use the following updating rule by knowing the value of G in kth step,

$$x_{(k+1)} = x_{(k)} + \alpha_{(k)} G_{(k)} \tag{8}$$

$\alpha_{(k)} > 0$ is a constant coefficient.

According to the above definition, it is shown that the matrix $G = v_2 v_2^T$ is considered as a supergradient function for$\lambda_2(\mathcal{L})$ [15]. By assuming $\tilde{\mathcal{L}} \neq \mathcal{L}$ and $v_2 \epsilon 1^{\perp}$ the following equation holds,

$$\lambda_2(\tilde{\mathcal{L}}) \leq \lambda_2(\mathcal{L}) + < v_2 v_2^T . (\tilde{\mathcal{L}} - \mathcal{L}) > \tag{9}$$

And the L matrix update rule is specified as follows,

$$\mathcal{L}^*_{(k+1)} = \mathcal{L}^*_k + \alpha_{(k)} G_{(k)} \tag{10}$$

While each agent has access only to the neighbors, this rule must be defined in a distributed form. So, for computing the appropriate $\mathcal{L}(x)$ matrix, the agent only needs to update the values of one row of the matrix.

The vector $m = (m_{ij})$ is a part of matrix $\mathcal{L}$ is defined as follows,

$$m_{ij} = \mathcal{L}_{ij}.\ \ i = 1.\cdots.N.\quad j > i \tag{11}$$

The matrix $\tilde{\mathcal{L}}_{ij}$ is considered for each pair $(i.j)$ with the elements of $\tilde{\mathcal{L}}_{ij_{ii}} = \tilde{\mathcal{L}}_{ij_{jj}} = -1$, $\tilde{\mathcal{L}}_{ij_{ij}} = \tilde{\mathcal{L}}_{ij_{ji}} = 1$, and other elements equal to zero. The Laplacian $\mathcal{L}$ matrix is:

$$\mathcal{L} = \sum_{j>i} \tilde{\mathcal{L}}_{ij}\, m_{ij}\,,\quad i = 1.\dots.N. \tag{12}$$

And the main optimization problem is expressed as follows,

$$max\, \lambda_2\left(\sum_{j>i} \tilde{\mathcal{L}}_{ij}\, m_{ij}\right),\quad i = 1.\dots.N. \tag{13}$$

$$s.t.\quad -1 \le m_{ij} \le 0\quad \forall (i.j)$$

As mentioned, $\lambda_2(p)$ is a concave function on m[15]. The supergradient vector corresponding to each m_{ij} is defined as follows,

$$G_{ij} = v_2^T \tilde{\mathcal{L}}_{ij} v_2 = -\left(v_{2i} - v_{2j}\right)^2 \tag{14}$$

The v_2 in a network must be estimated by each agent using available information. So, we can write the update rule similar to (7):

$$m^*_{ij,(k+1)} = m^*_{ij,(k)} + \alpha_{(k)} G_{ij,(k)} \tag{15}$$

$\alpha_{(k)}$ is the step-size coefficient, and $G_{ij,(k)}$ is the function of the supergradient in the step k. The $m^*_{ij,(k+1)}$ should be on the feasible set of (12), and if $\alpha_k G_{ij,(k)} \le 0$, the update rule guarantees that the inequality $m^*_{ij,(k+1)} \le m^*_{ij,(k)}$ is true. Therefore, to have a solution on a feasible set, $m^*_{ij,(k+1)} \ge -1$ must be met. It leads to the Eq. (15) [15].

$$p^*_{ij,(k+1)} = max(p^*_{ij,(k+1)}\,, -1) \tag{16}$$

The updated values are determined as follows,

$$\mathcal{L}^*_{ii,(k+1)} = -\sum_{j=1.j\neq i}^{N} \mathcal{L}^*_{ij,(k+1)}\quad .\, i = 1.\cdots.N \tag{17}$$

The above results show that each agent can find the related row values of the updated Laplacian matrix by using the supergradient algorithm. As can be seen, it is important to know the elements of the second eigenvector for ith agent and its neighbors. The following Section presents a method for estimating the second eigenvalue in a connected graph.

In the problem of eigenvalue estimation, the fading effect of the channel in the exchange between agents has been disregarded.

B. Decentralized power iteration

This section introduces an algorithm to estimate the second small eigenvalue (λ_2) for the weighted Laplacian matrix (3). During the process, it is assumed that $\mathcal{L}$ is approximately independent of the change in states. Since all eigenvalues of the Laplacian matrix ($\mathcal{L}$) are non-negative, the Power Iteration Algorithm can be used for $I - \alpha\mathcal{L}$ with small and positive α-values to find the second small eigenvalue of a matrix [16].

The vector $\tilde{v} = (\tilde{v}^1 \dots \tilde{v}^n)^T \in R^n$ is assumed as an estimate of the eigenvector v_2. The algorithm includes the following three parts,

I. $\dot{\tilde{v}} = -Ave(\{\tilde{v}^i\})\mathbb{1}$

II. $\dot{\tilde{v}} = -\alpha\mathcal{L}\tilde{v}$

III. $\dot{\tilde{v}} = -(Ave(\{(\tilde{v}^i)^2\}) - 1)\tilde{v}$

Where $Ave(\{\tilde{v}^i\}) \triangleq (\sum_i \tilde{v}_i)/n$ and $\mathbb{1}$ is the eigenvector corresponding to the largest eigenvalue of the $I - \alpha L$ matrix. The first step is driving $\tilde{v}$ to the null space $\mathbb{1}$ (the space is spanned by eigenvectors $(v_2, \dots, v_n)$. For most of the initial values for $\tilde{v}$, the second step directs $\tilde{v}$ to the $v_1 = \mathbb{1}$, but if $\mathbb{1}^T\tilde{v} = 0$, $\tilde{v}$ converges to the second eigenvector v_2. The third step move $\tilde{v}$ to the sphere of radius $\sqrt{n}$ [16].

The three steps of this algorithm lead to the weighted equations as follows, where the weight parameters k_1, k_2, k_3 are scalar values.

$$\dot{\tilde{v}} = -k_1 Ave(\{\tilde{v}^i\})\mathbb{1} - k_2 L\tilde{v} \\ - k_3(Ave(\{(\tilde{v}^i)^2\}) - 1)\tilde{v} \tag{18}$$

Theorem [16]: For each initial value $\tilde{v}(t_0)$ and the weights $k_1, k_2, k_3 > 0$, as long as $v^2\tilde{v}(t_0) \neq 0$, for system (17), where $\|\tilde{v}_2\| = \sqrt{n(\frac{k_3 - k_2\lambda_2}{k_3})}$, the following conditions for convergence λ_2 to the weight matrix L is necessary and sufficient.

$$k_1 > k_2\lambda_2 \\ k_3 > k_2\lambda_2 \tag{19}$$

The estimate equation (18) requires access to all elements of $\tilde{v}$, but agent i only has access to some of them with local communication. Therefore, a decentralized algorithm should be proposed. In other words, the estimate of λ_2^i will be considered for each agent. The most critical problem in the power iteration algorithm is the Ave operator for $\tilde{v}^i$ and $(\tilde{v}^i)^2$. In [16], the average consensus PI estimator has been used to overcome this problem. Since there are two unknown average variables in (18), two estimators will be required.

This estimator is designed for n separate agents so that each agent can get the value of $a^i(t)$ to compute $\bar{a}(t) = \frac{1}{n}\sum_i a^i(t)$ by measuring using local connections. The PI estimator is considered as follows:

$$\dot{z}^i = \gamma(a^i - z^i) - K_p \sum_{j\in N_i(t)} (z^i - z^j) \\ + K_I \sum_{j\in N_i(t)} (w^i - w^j) \tag{20}$$

$$\dot{w}^i = -K_I \sum_{j\in N_i(t)} (z^i - z^j)$$

$z^i \epsilon R^2$ is the average estimate, $z^i(1)$ for $Ave(\{\tilde{v}^i\})$ and $z^i(2)$ for $Ave(\{(\tilde{v}^i)^2\})$. K_p and K_I are estimator weights, and constant γ determines the effect of past information on the estimate. The estimator error is:

$$e^i(t) = x^i(t) - \frac{1}{n}\sum_{i=1}^{n} a^i(t) \tag{21}$$

In the proposed algorithm, the ith agent, according to the neighborhood radius and the number of adjacent agents, can have access to the variables $\{\tilde{v}^j, z^{j,1}, w^{j,1}, z^{j,2}, w^{j,2}\}$.

The second eigenvector $\tilde{v}$, the variable needed to build the supergradient function, is estimated by each agent. By estimating the unknown variables, the kth step of the supergradient algorithm to determine the optimal structure will be considered as follows,

I. Calculate the second small special value $\tilde{v}_i^{(k)}$ and receive $\tilde{v}_{j,(k)}$ from the neighbors
II. Update $G_{ij,(k)}$ values according to Equ.(13)
III. Update $\mathcal{L}^*_{ij,(k+1)}$ values according to Equ.(14) and Equ.(15).
IV. Update $\mathcal{L}^*_{ii,(k+1)}$ values according to Equ.(16)

III. Design decentralized controller

A decentralized controller will be required to move agents from their current state to the desired status. Defining the topology of the graph in the previous Section is performed without considering that the Laplacian matrix can be different by changing the position of the agents during the estimation process. Therefore, the movement of agents causes a constant error in elements of the Laplacian matrix in each step. So, the controller should be as fast as possible despite the input limitation and have a minimum steady-state error.

According to each $\mathcal{L}^*_{i,(k)}$, based on the definition of the Laplacian matrix Equ.(3) and the adjacency matrix, and the use of the inverse operator, the desired distance between the connected agents can be achieved as follows,

$$d_{ij,(k)} = A_{ij}^{-1}(\mathcal{L}^*_{ij,(k)}) \tag{22}$$

It should be noted that the connections between the agents are affected by the distance between them, or as mentioned in the introduction, the strength of communication and packet loss phenomenon in the network is different in various positions. It is possible that the reference values obtained for the distances Equ.(21), don't be in the detectable range of agents.

The agents in the multi-agent network should present at the appropriate distance from each other in the least possible time according to the desired topology in Equ.(21). We use the answer obtained from the optimization problem as a reference in the decentralized controller. In other words, the problem in this part will be as follows,

$$\min_{x_i} \left\| \mathcal{L}_i(x) - \mathcal{L}^*_{i,(k)} \right\|_2^2 \tag{23}$$

Where $\mathcal{L}_i(x)$ is the ith row of the Laplacian matrix, generally a function of system states. $\mathcal{L}^*_{i,(k)}$ is the row corresponding to the ith agent, which is estimated in the kth iteration from the previous Section optimization problem.

The sliding error $s_i(t)$ for every possible (i,j) is defined as the weighted sum function as follows,

$$s_i(t) = \sum_{j\in C} s_{ij}(t)\,, \qquad i = 1.\,\ldots.\,N \tag{24}$$

Where,

$$s_{ij}(t) = \|r_{ij}\| - d_{ij,(k)}\,, \qquad j \in C \tag{25}$$

The sliding surface for all agents will generally be as follows,

$$S(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_N(t) \end{bmatrix} \tag{26}$$

It should be noted that the sliding surface is defined locally and in a distributed form. It is linear with respect to the $\mathcal{L}$ (Laplacian matrix). The reaching rule is defined as follows for the sliding mode controller,

$$\dot{s}_i(t) = \sum_{j\in C} \left(-\psi\, s_{ij}(t) - \phi\, sgn\left(s_{ij}(t)\right)\right) \tag{27}$$

By substituting Equ. (4) and with the assumption that the position changes of the neighbors are limited in each estimation run, $\dot{x}_j \cong 0, \quad \forall j \in C$, we get,

$$\begin{aligned} \dot{s}_i(t) &= \left(\sum_{j\in C} \nabla_{x_i}\left(s_{ij}(t)\right)\right)\left(f(x_i) + u_i(t)\right) \\ &= \sum_{j\in C} \left(-\psi\, s_{ij}(t) \right.\\ &\quad \left. - \phi\, sgn\left(s_{ij}(t)\right)\right) \end{aligned} \tag{28}$$

By rewriting the above relation, the control signal is written as follows.

$$\begin{aligned} u_i(t) = \Bigg(\sum_{j\in C} -\left(\nabla_{x_i}\left(s_{ij}(t)\right)\right)^{-1} &\left(-\psi\, s_{ij}(t)\right. \\ &\left. - \phi\, sgn\left(s_{ij}(t)\right)\right)\Bigg) - f(x_i) \end{aligned} \tag{29}$$

Theorem: Consider dynamics of the agents as Equ(4). With the distributed controller (29) and by setting parameters $\phi > 0$ and $\psi > 0$, the sliding surface $S(t) = 0$ in a finite time with the boundary $\sum_{j\in C} |s_{ij}(0)|$ is achievable.

Proof: The following Lyapunov function is a candidate for the networked system,

$$V(t) = \frac{1}{2} S(t)^T S(t) \tag{30}$$

By taking the derivative of Lyapunov's function,

$$\dot{V}(t) = S(t)^T . \dot{S}(t) \tag{31}$$
$$= \sum_{i\in N}\left(\sum_{j\in C} s_{ij}(t) . \sum_{j\in C}\left(-\psi\, s_{ij}(t) - \phi\, sgn\left(s_{ij}(t)\right)\right)\right)$$

By replacing the Equ.(23) and Equ.(25), we get,

$$\dot{V}(t) = \left(\sum_{j\in C} s_{ij}(t)\right)^T . \sum_{j\in C}\left(-\psi\, s_{ij}(t) - \phi\, sgn\left(s_{ij}(t)\right)\right) \tag{32}$$
$$\leq -\phi \sum_{j\in C} \frac{s_{ij}(t)^T\, s_{ij}(t)}{\left|s_{ij}\,(t)\right|}$$
$$\leq -\phi \sum_{j\in C} \left|s_{ij}\,(t)\right|$$

With the help of the above Lyapunov function and the appropriate selection the $\phi > 0$, the access time t_r at the boundary $\sum_{j\in C} \left|s_{ij}\,(0)\right|$ is limited. It should be noted that achieving the better structure for the communication graph increases the convergence time in the control Section.

Theorem: Consider dynamics of the agents as Equ(4). In sliding phase, when $S(t) = 0$, the assumed system will converge to $\tilde{\mathcal{L}}^* \subset \mathcal{L}^*$.

Proof: As long as the matrix $\mathcal{L}$ is positive definite and,

$$S(t) = \mathcal{L} \sum_{j\in C} s_{ij}(t) = 0 \tag{33}$$

We can conclude the following equation.

$$\left\|r_{ij}\right\| - d_{ij,(k)} = 0\,, \qquad j \in C \tag{34}$$

This equation shows that the two agents i and j are at the desired distance r_{ij}.

Theorem: Consider dynamics of the agents as Equ(4). In sliding phase, when $S(t) = 0$, the assumed system will converge to $\tilde{\mathcal{L}}^* \subset \mathcal{L}^*$.

Proof: As long as the matrix $\mathcal{L}$ is positive definite and,

$$S(t) = \mathcal{L} \sum_{j\in C} s_{ij}(t) = 0 \tag{35}$$

Considering the fading effect in the channel, the mathematical expression for r_{ij} is modified as follows:

$$\hat{r}_{ij} = \left\|x_i - x_{ij}\right\| \tag{36}$$

As a result,

$$\bar{s}_{ij}(t) = \left\|\hat{r}_{ij}\right\| - d_{ij,(k)} \tag{37}$$
$$= \left\|x_i - \xi_{ij}\, x_j\right\| - d_{ij,(k)}$$

In this case, by substituting the above expression into equation (28), the control signal is defined as follows:

$$u_i(t) = \left(\sum_{j\in C} -\left(\nabla_{x_i}\left(\bar{s}_{ij}(t)\right)\right)^{-1}\left(-\psi\, \bar{s}_{ij}(t) - \phi\, sgn\left(\bar{s}_{ij}(t)\right)\right)\right) - f(x_i) \tag{38}$$

To compensate for the impact of fading on the measured distance values, the new sliding surface is defined as follows using the mean value $\bar{\xi}$:

$$\hat{s}_{ij}(t) = 1/\,\bar{\xi}\ \left\|r_{ij}\right\| - d_{ij,(k)}\,, \qquad j \in C \tag{39}$$

The sliding error $s_i(t)$ for every possible (i, j) is defined as

$$\hat{s}_i(t) = \sum_{j\in C} \hat{s}_{ij}(t)\,, \qquad i = 1. \ldots . N \tag{40}$$

And the sliding surface for all agents will generally be as follows,

$$\hat{S}(t) = \begin{bmatrix} \hat{s}_1(t) \\ \hat{s}_2(t) \\ \vdots \\ \hat{s}_N(t) \end{bmatrix} \tag{41}$$

To analyze the stability of the closed-loop system in the presence of measured values from neighboring nodes affected by fading channels, a control signal in the form of Equ(38) is proposed. By substituting this control signal into the sliding surface (41), the stability of the closed-loop system is guaranteed through the following theorem.

Theorem: Consider dynamics of the agents as Equ(4) and the sliding surface defined by Equ(41), With the distributed controller (38) and by setting parameters $\phi > 0$ and $\psi > 0$, the sliding surface $\hat{S}(t) = 0$ in a finite time with the boundary $\sum_{j\in C} \left|\hat{s}_{ij}(0)\right|$ is achievable.

Proof: The following Lyapunov function is a candidate for the networked system,

$$V(t) = \frac{1}{2}\, \hat{S}(t)^T \hat{S}(t) \tag{42}$$

By taking the derivative of Lyapunov's function,

$$\dot{V}(t) = \hat{S}(t)^T . \dot{\hat{S}}(t) \tag{43}$$
$$= \sum_{i\in N}\left(\frac{1}{\bar{\xi}}\right)^2 \left(s_i(t)\left(-\psi\, \bar{s}_i(t) - \phi\, sgn\left(\bar{s}_i(t)\right)\right)\right)$$
$$= \sum_{i\in N}\left(\frac{1}{\bar{\xi}}\right)^2 \left(\left(-\psi\, s_i(t)\bar{s}_i(t) - \phi\, s_i(t) sgn\left(\bar{s}_i(t)\right)\right)\right)$$

Using Young's inequality for products with $\varepsilon_2, \varepsilon_2 > 0$, we have the following:

$$\dot{V}(t) = \sum_{i\in N}\left(\frac{1}{\bar{\bar{\xi}}}\right)^2\left(\left(-\frac{1}{\varepsilon_1}\,s_i(t)^2 - \frac{\psi^2}{\varepsilon_2}\,\bar{s}_i(t)^2 - \frac{1}{\varepsilon_1}\,s_i(t)^2 - \frac{\phi^2}{\varepsilon_2}\,sgn\big(\bar{s}_i(t)\big)^2\right)\right) < \sum_{i\in N}\left(\frac{1}{\bar{\bar{\xi}}}\right)^2\left(\left(-\frac{\psi}{\varepsilon_2}\,\bar{s}_i(t)^2\right)\right) \tag{44}$$

from (24) and (39), we have $E(s) = E(\hat{s})$. Take the expectation of both sides of the expression (18), we have,

$$E[\dot{V}(t)] \le \sum_{i\in N}\frac{1}{\bar{\bar{\xi}}}\left(-\frac{\psi}{\varepsilon_2}\,E[V(t)]\right) \tag{45}$$

With the help of the above Lyapunov function and the appropriate selection the $\psi > 0$, the access time t_r at the boundary $\sum_{j\in C}\left|\hat{s}_{ij}\,(0)\right|$ is limited. It should be noted that achieving the better structure for the communication graph increases the convergence time of consensus in the multiagent system. Equ(45) shows that if $\hat{S}(t) \neq 0$, the trajectories outside the $\hat{S}(t) = 0$ will reach it in a limited time.

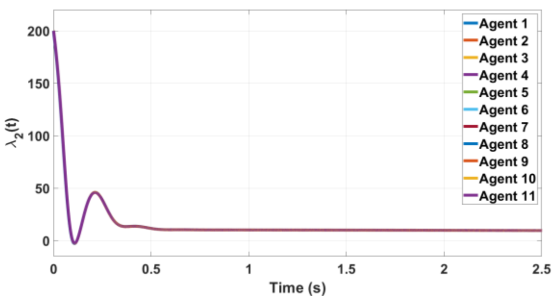

Fig. 2. The graph connectivity λ_2 estimated over time by each agent.

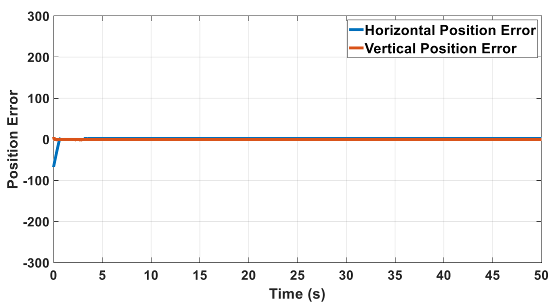

Fig. 3. The closed-loop tracking error for agent#5 under normal conditions, without considering fading in the communication channel.

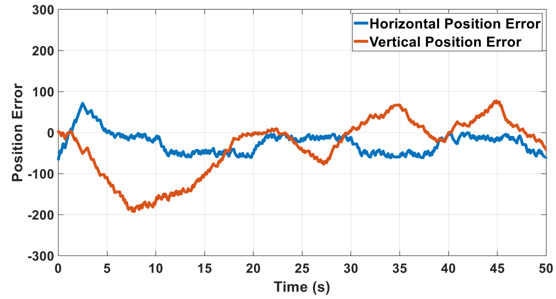

Fig. 4. The closed-loop tracking error for agent #5, considering fading in the communication channel, using conventional control similar to the [15].

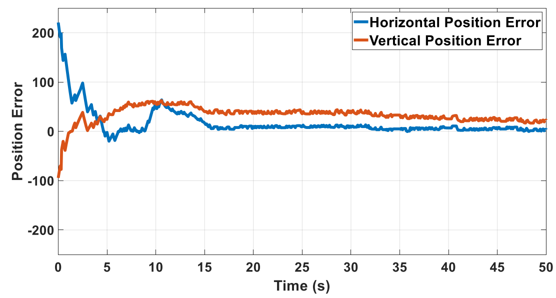

Fig. 5. The closed-loop tracking error for agent #5, considering fading in the communication channel, using the sliding mode controller.

In the above case, in general, there is no guarantee that all neighbors are available and the graph forms a tree, so the Therefore, the convergence study of the supergradient algorithm in the upper layer can be affected by the possible error.

In other words, the difference error between the Laplacian matrix and the desired matrix should not affect the convergence of the algorithm to find the optimal eigenvalue.

Assuming that $m^*_{(k+1)}$ is the updated vector in the matrix $L^*_{(k+1)}$ in the iteration $(k+1)$ of the supergradient calculations because there is no guarantee to find non-neighboring agents, the formation leads to form the agents with the error ε.

In this case, instead of obtaining the vector $m^*_{(k+1)}$, we will reach the vector $\tilde{p}^{(k+1)}$, which is:

$$\widetilde{m}_{(k+1)} = m^*_{(k+1)} + \varepsilon_{(k+1)} \tag{46}$$

$e_{(k+1)} = m^*_{(k+1)} - \mathrm{m}^*$ is the error of the supergradient algorithm relative to the final optimal value of m^*. It is shown that the assumed error will not affect the convergence of the estimator to the optimal value and the error between $\widetilde{\mathrm{m}}_{(k+1)}$ and the optimal value m^* has a finite bound. The error between the optimal eigenvalue and the eigenvalue reached with the supergrdient algorithm can be defined as follows [15],

As a result, the error between the optimal eigenvalue and the eigenvalue reached with the supergrdient algorithm can be defined as:

$$\lambda_2^* - \lambda_2(\tilde{m}_{(k)}) \leq \|\tilde{m}_{(k)} - m^*\|^2 + \frac{\sum_{i=1}^{k}\|\varepsilon_{(i+1)}\|^2}{2\sum_{i=1}^{k}\alpha_{(i)}} + \frac{\sum_{i=1}^{k}\alpha_{(i)}{}^2\, g_{(i)}}{2\sum_{i=1}^{k}\alpha_{(i)}} \quad (47)$$

The difference between the eigenvalue obtained in k^{th} step with the desired optimal value is in a specific band by setting the square and summable values of $\alpha_{(i)}$ and assuming the norm of the supergradient function is bounded $\|\mathrm{G}_{\mathrm{ij},(k)}\| < \bar{G}$ and $\|\varepsilon_{(k)}\|$ is not divergent.

IV. Simulation results

In this Section, the proposed algorithm has been tested using simulation. A multi-agent system is composed of 11 autonomous agents. The maximum permissible range is $R = 10$, and the input signal changes in the range $-1 < U < 1$. The minimum distance is $r_0 = 1$. Assume the graph of the network is initially connected. By choosing $\alpha^{(k)} = \frac{1}{k+1}$ the algorithm is convergent.

The estimated signal of the second small eigenvalue λ for each agent can be seen in Fig.2. It can be seen that the estimates values of all agents converge to the algebraic connectivity of the graph.

Fig.3 shows the closed-loop tracking error for agent#5 under normal conditions, without considering fading in the communication channel. Fig.4 shows the closed-loop tracking error, considering fading in the communication channel, using conventional control similar to the [15] and Fig.5 shows the closed-loop tracking error, considering fading in the communication channel, using the sliding mode controller.

Fig. 6 shows the movement of agents over a period of time. As can be seen from the movement of robots, agents are moving towards more effective communication across the network. This is also shown in Fig. 7, where the second eigenvalue changes are plotted. The small second eigenvalue is $\lambda = 2$ in the first, and its final value is $\lambda = 9$.

In another example, the algorithm was investigated by considering uncontrollable agents in the network. The results are plotted in Fig. 8 and Fig. 9. In some iterations, due to inaccessible agents, the second small eigenvalue may decrease, but the final position of the agents eventually changes to improve the overall connectivity of the graph.

Compared to [15], the obtained results have more robustness to noise in controlling the position of agents. The algorithm of the mentioned article will not guarantee control and reach the optimal formation in every iteration.

Also, comparing the results with [28] and [30] shows that the use of sliding mode control can be effective in maintaining connectivity despite fading in communication channels.

In these papers, the subject of average consensus among agents has been discussed, which does not guarantee connectivity and convergence over time if the structure of the communication network is variable. The use of global preservation connectivity algorithm can be a guarantee of this.

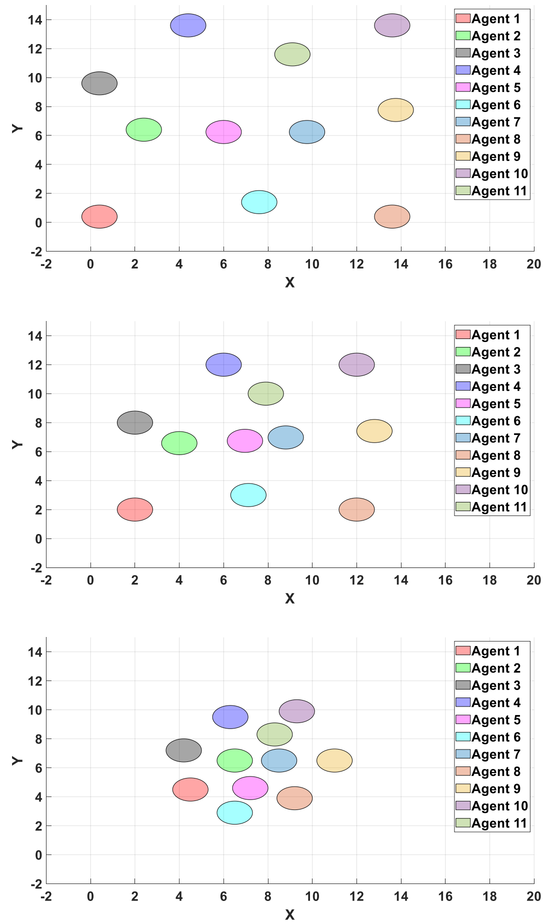

Fig. 6. Changes the position of agents on the plane in order to achieve the maximum second small eigenvalue.

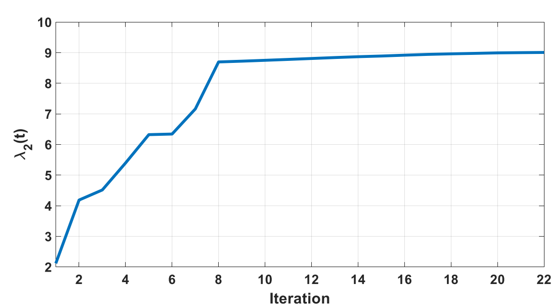

Fig. 7. Value of maximum second small eigenvalue $\lambda_2(t)$ during the first simulation.

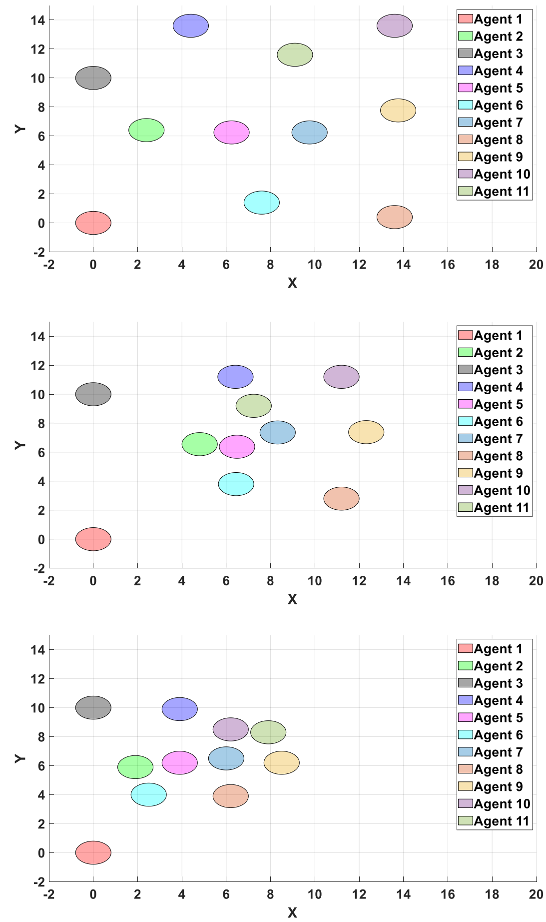

Fig. 8. Changes the position of agents on the plane by considering uncontrollable agents (Agent 1 and Agent 3).

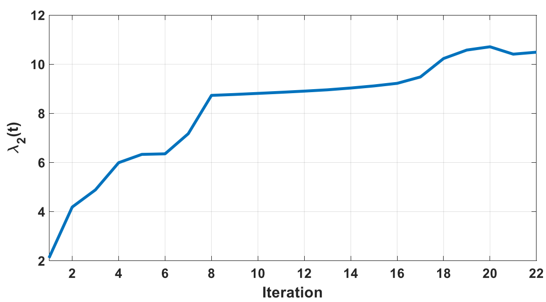

Fig. 9. Value of maximum second small eigenvalue $\lambda_2(t)$ during the second simulation.

I. Conclusion

In this paper, the problem of connectivity maintenance in a robotic multi-agent system with nonlinear dynamics was investigated. Wireless communication is used to exchange information, and according to the fading channel model for a wireless network, the quality of communication is in a specific range in the best condition to have the least packet lost. This goal is achieved by maximizing the second small eigenvalue of the Laplacian matrix $\mathcal{L}(x)$. Distributed sliding mode controller was designed to achieve the best topology for the information exchange. Finally, the results were evaluated through numerical simulations.

References

[1] H. S. Ahn, Formation Control: Approaches for Distributed Agents: Springer International Publishing, 2019.

[2] S. Martínez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," Automatica, vol. 42, pp. 661-668, 2006/04/01/ 2006.

[3] M. Otte, M. J. Kuhlman, and D. Sofge, "Auctions for multi-robot task allocation in communication limited environments," Autonomous Robots, vol. 44, pp. 547-584, 2020/03/01 2020.

[4] E. Olcay, J. Bodeit, and B. Lohmann, "Sensor-based Exploration of an Unknown Area with Multiple Mobile Agents," IFAC-PapersOnLine, vol. 53, pp. 9621-9627, 2020/01/01/ 2020.

[5] S. Rahili, J. Lu, W. Ren, and U. M. Al-Saggaf, "Distributed Coverage Control of Mobile Sensor Networks in Unknown Environment Using Game Theory: Algorithms and Experiments," IEEE Transactions on Mobile Computing, vol. 17, pp. 1303-1313, 2018.

[6] J. Cortes, S. Martínez, and F. Bullo, "Robust Rendezvous for Mobile Autonomous Agents via Proximity Graphs in Arbitrary Dimensions," Automatic Control, IEEE Transactions on, vol. 51, pp. 1289-1298, 09/01 2006.

[7] S. Ponda, L. Johnson, A. Kopeikin, H.-L. Choi, and J. How, "Distributed Planning Strategies to Ensure Network Connectivity for Dynamic Heterogeneous Teams," Selected Areas in Communications, IEEE Journal on, vol. 30, pp. 861-869, 06/01 2012.

[8] H. Su, X. Wang, and G. Chen, "A connectivity-preserving flocking algorithm for multi-agent systems based only on position measurements," International Journal of Control, vol. 82, pp. 1334-1343, 2009/07/01 2009.

[9] D. Zelazo, A. Franchi, P. Allgöwer, H. Bülthoff, and P. Giordano, "Rigidity Maintenance Control for Multi-Robot Systems," 2012.

[10] D. Zelazo, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano, "Decentralized rigidity maintenance control with range measurements for multi-robot systems," The International Journal of Robotics Research, vol. 34, pp. 105-128, 2015/01/01 2014.

[11] L. Zhang, J. Wang, Z. Lin, L. Lin, Y. Chen, and B. He, "Distributed Cooperative Obstacle Avoidance for Mobile Robots Using Independent Virtual Center Points," Journal of Intelligent & Robotic Systems, vol. 98, pp. 791-805, 2020/06/01 2020.

[12] K. Khateri, M. Pourgholi, M. Montazeri, and L. Sabattini, "Effect of Stubborn Agents on Bounded Confidence Opinion Dynamic systems: Unanimity in Presence of Stubborn Agents," in 2019 27th Iranian Conference on Electrical Engineering (ICEE), 2019, pp. 875-880.

[13] C. D. G. G. Royle, C. Godsil, and G. F. Royle, Algebraic Graph Theory: Springer, 2001.

[14] R. Dai and M. Mesbahi, "Optimal topology design for dynamic networks," in 2011 50th IEEE Conference on Decision and Control and European Control Conference, 2011, pp. 1280-1285.

[15] M. C. D. Gennaro and A. Jadbabaie, "Decentralized Control of Connectivity for Multi-Agent Systems," in Proceedings of the 45th IEEE Conference on Decision and Control, 2006, pp. 3628-3633.

[16] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S.

Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks," Automatica, vol. 46, pp. 390-396, 2010/02/01/ 2010.
[17] D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," Journal of Computer and System Sciences, vol. 74, pp. 70-83, 2008/02/01/ 2008.
[18] X. Xue, X. Yue, and J. Yuan, "Connectivity preservation and collision avoidance control for spacecraft formation flying in the presence of multiple obstacles," Advances in Space Research, 2020/06/06/ 2020.
[19] S. J. Yoo and B. S. Park, "Connectivity-Preserving Approach for Distributed Adaptive Synchronized Tracking of Networked Uncertain Nonholonomic Mobile Robots," IEEE Transactions on Cybernetics, vol. 48, pp. 2598-2608, 2018.
[20] C. P. Bechlioulis and K. J. Kyriakopoulos, "Robust model-free formation control with prescribed performance and connectivity maintenance for nonlinear multi-agent systems," in 53rd IEEE Conference on Decision and Control, 2014, pp. 4509-4514.
[21] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," IEEE Transactions on automatic control, vol. 49, pp. 1520-1533, 2004.
[22] Y. Zhang and Y.-P. Tian, "Allowable delay bound for consensus of linear multi-agent systems with communication delay," International Journal of Systems Science, vol. 45, pp. 2172-2181, 2014/10/03 2014.
[23] A. Nejadvali, R. Esfanjani, A. Farnam, and G. Crevecoeur, "Delay dependent criteria for the consensus of second-order multi-agent systems subject to communication delay," IET Control Theory and Applications, vol. 15, pp. 1-10, 05/07 2021.
[24] R. Mu and A. Wei, "Consensus disturbance rejection for linear multi-agent systems based on output feedback," in 2019 Chinese Control Conference (CCC), 2019, pp. 6094-6099.
[25] R. Carli, G. Como, P. Frasca, and F. Garin, "Distributed averaging on digital erasure networks," Automatica, vol. 47, pp. 115-121, 01/31 2011.
[26] J. Mou, Q. He, Z. Xia, and J. Wang, "Consensus of the Distributed Multiagent System with the Framework of the Small-World Network," Mathematical Problems in Engineering, vol. 2021, p. 6193508, 2021/02/15 2021.
[27] E. Carrillo, S. Yeotikar, S. Nayak, M. K. M. Jaffar, S. Azarm, J. W. Herrmann, et al., "Communication-Aware Multi-Agent Metareasoning for Decentralized Task Allocation," IEEE Access, vol. 9, pp. 98712-98730, 2021.
[28] X. Gu, T. Jia, and Y. Niu, "Consensus tracking for multi-agent systems subject to channel fading: a sliding mode control method," International Journal of Systems Science, vol. 51, pp. 1-9, 07/30 2020.
[29] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: design, analysis and applications," in Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies., 2005, pp. 1653-1664 vol. 3.
[30] J. Li, Y. Niu and Y. Zou, "Sliding Mode Control for Networked Control System Under Fading Channels," 2019 IEEE Conference on Control Technology and Applications (CCTA), Hong Kong, China, 2019, pp. 561-566.

Sorush Akhlaghi received the B.S. degree in Electrical Engineering from Shahroud University, Semnan, Iran, in 2010, the M.S. degree at control engineering from the Shiraz University, Shiraz, Iran, in 2014. He is currently working toward the Ph.D. degree in electrical and computer engineering with Ferdowsi University of Mashhad, Mashhad, Iran. His research interests include control and optimization of multi-agent systems.

Naser Pariz received the B.S. and M.Sc degree in Electrical Engineering from Ferdowsi University of Mashhad, Iran, in 1988 and 1991 respectively. He received his Ph.D. from the Department of Electrical Engineering at Ferdowsi University of Mashhad in 2001. He is a Professor at Ferdowsi University. His research interests are nonlinear and control systems.

Mohammad Bagher Naghibi Sistani received the B.S. degree in electronics from the University of Tehran, Tehran, Iran, in 1991, the M.S. degree at control engineering from the University of Tehran, Tehran, Iran, in 1995, and the Ph.D. degree in control engineering from the Ferdowsi University of Mashhad, Iran, in 2005. He currently is Associate Professor at the Department of Electrical Engineering and Biomedical Engineering, Ferdowsi University of Mashhad. His research interests include reinforcement learning, soft computing, optimal control, multi- agent systems, and machine learning. He has published over 70 journal and conference papers.