Received: July 17, 2022; **Accepted:** July 03, 2023.
DOI. 10.30473/coam.2023.64817.1209
Winter-Spring (2024) Vol. 9, No. 1, (1-34)
Research Article

Open Access

Control and Optimization in Applied Mathematics - COAM

A Fuzzy Sliding Mode Control for Nonlinear Leader-Follower Multi-Agent Systems

Negar Izadi ✉, **Mohammad Taghi Dastjerdi**

Department of Mathematics, Faculty of Sciences, University of Zanjan, P.O. Box 45371-38791, Zanjan, Iran.

✉ **Correspondence:**
Negar Izadi
E-mail:
negarizadi@znu.ac.ir

How to Cite
Izadi, N., Dastjerdi, M.T. (2024). "A fuzzy sliding mode control for nonlinear leader-follower multi-agent systems", Control and Optimization in Applied Mathematics, 9(1): 1-34.

Abstract. In this paper, we present a new approach for achieving leader-follower consensus in a network of nonlinear dynamic agents with an undirected graph topology, using a fuzzy sliding mode controller (FSMC) for Multi-Agent Systems (MASs). Our proposed sliding mode controller is based on a separating hyperplane that effectively addresses the consensus problem in MASs. Additionally, we design a fuzzy controller to eliminate the chattering phenomenon. According to the communication graph topology and the Lyapunov stability condition, the proposed FSMC satisfies the consensus condition. One significant advantage of our approach is that the system states converge to the sliding surface quickly and remain on the surface, thereby ensuring better tracking performance. We validate the effectiveness of our proposed approach through simulation results.

Keywords. Consensus, Fuzzy controller, Multi-agent system, Sliding mode control.

MSC. 93A16; 93C10; 93C42; 93D50.

1 Introduction

In recent years, the number of complex problems that engineers have to deal with has progressively increased. One strategy for solving such problems is to transfer them to a distributed network of smaller and simpler subsystems, resulting in a Multi-Agent System (MAS) instead of a single, complicated system. In a MAS, agents efficiently cooperate to achieve group behavior through local interaction. The structure of the MASs has been inspired by the group life of animals, where each agent (subsystem) communicates with other adjacent agents while performing its independent tasks until the system achieves its overall and final goal. Further explanations of system types and their control are provided in the next section.

In this context, the consensus problem is a crucial research area, which involves achieving the convergence of the outputs or states of all agents to a common value with minimal computational costs and communication requirements. Due to its importance, multiple scientific and research communities have pointed to MAS control design as a challenging and applicable new area of research. MASs have numerous applications in military [7], autonomous vehicles [10], and robotics [5, 18].

The consensus problem involves achieving the convergence of the outputs or states of all agents to a common value [26]. A theoretical description of consensus controls was proposed for the Vicsek model [6, 32] while in [22] a general structure of the consensus problem for networks of integrators was developed. For more details on the consensus problem and its solutions, see [1, 3, 21, 27]. In [25], a sufficient condition was derived to achieve consensus for first-order integrator MASs with jointly connected communication graphs. The consensus problem for networks of second-order and high-order integrators was investigated in [8, 15, 23, 24, 28, 40]. Additionally, the consensus problem of MASs with general linear dynamics has been studied in several works, including [11, 14, 16, 17, 29, 31, 33, 34, 35, 37]. In [30, 36], conditions were established for achieving consensus of MASs with Lipschitz-type nonlinearity. Additionally, a consensus algorithm for MASs with quantized communication links was proposed in [4, 12]. Consensus algorithms can be classified into two types: consensus without any leader and consensus with a leader. The former is called leaderless consensus, while the latter is referred to as leader-follower consensus. In the problems with only one leader, distributed tracking control is observed. However, in some practical applications, it is necessary to deal with more than one leader, which leads to the containment control problem.

In recent years, SMC techniques have been applied to a wide range of MASs. In [39], SMC techniques were employed for a class of leader-follower tracking error problems of a general linear MAS. Moreover, a robust consensus protocol for a linear MAS was developed using the SMC in [13].

In this paper, we propose a separating plane-based sliding mode controller to solve the second-order dynamic consensus problem of multi-agent leader-follower systems. The proposed controller is asymptotically stable in the Lyapunov sense. Moreover, the proposed sliding mode controller has a faster convergence speed than other classical methods, implying that follower agents reach the leader state vector much faster than traditional methods. Additionally, the control presented in this paper causes the state vectors of the system to reach a sliding surface in a finite time. It is worth noting that the changes in the value of control do not depend on the sign of sliding surface, but rather on the placement of S within or outside of the cone generated by the rows of a matrix. This feature reduces the probability of the appearance of chattering.

This paper is structured as follows: In Section 2, we present some preliminaries related to graph theory, the hyperplane separation theorem, and the leader-follower MASs, as well as the Lyapunov stability theorem. Section 3 addresses the problem of designing a fuzzy sliding mode controller based on a separating hyperplane for the leader-follower MAS with second-order dynamics. In Section 4, simulation results are provided to illustrate the effectiveness of the proposed controller. Finally, the last section concludes this paper.

2 Preliminaries

In this section, we give provide some preliminaries related to graph theory, the particle swarm optimization algorithm, the hyperplane separation theorem, the leader-follower MAS, and the Lyapunov stability theorem. Firstly, we introduce some notions used throughout this paper. We use $x(t)$ to denote a function of t, $\dot{x}(t)$ to denote the derivative of $x(t)$ with respect to t, and $\ddot{x}(t)$ to denote the second derivative of $x(t)$. The transpose of a matrix or vector A, denoted by A^T. We denote the non-negative part of $\mathbb{R}^n$ by $\mathbb{R}^n_+$, which is defined as $\mathbb{R}^n_+ = \{(x_1, \ldots, x_n) \,|x_i \in \mathbb{R}^+ \cup \{0\},\ i = 1, \ldots, n\}$. The Kronecker product of a matrix $A \in \mathbb{R}^p \times \mathbb{R}^q$ and a matrix $B \in \mathbb{R}^r \times \mathbb{R}^s$ is denoted by $A \otimes B$ and defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & \ldots & a_{1q}B \\ \vdots & \ddots & \vdots \\ a_{p1}B & \ldots & a_{pq}B \end{bmatrix}.$$

Let $G = (V, E)$ as be an undirected graph, where $V = \{v_1, v_2, \ldots, v_N\}$ is the set of all vertices and $E = \{(i, j)|i, j \in V\}$ is the set of edges consisting of unordered pairs (i, j), which are called edges of G. Vertices i and j are said to be adjacent if the edge (i, j) exists in E. A path between vertices i_1, i_l in G is a sequence of edges $(i_1, i_2), \ldots, (i_{l-1}, i_l)$ such that (i_k, i_{k+1}) for $k = 1, \ldots, l-1$ are in E. The graph G is said to be connected if there is at least one path between two vertices of G. The adjacency matrix of G is a symmetric matrix $A = [a_{ij}] \in \mathbb{R}^N \times \mathbb{R}^N$ where $a_{ij} = 1$ if $(i, j) \in E$ and $a_{ij} = 0$ otherwise. The degree of G is a diagonal matrix $D = diag(d_1, \ldots, d_N)$, where $d_i = \sum_{j=1}^{N} a_{ij}$ for $i = 1, \ldots, N$. The Laplacian matrix of G is defined as $L = D - A$, which is also symmetric. In this paper, we consider simple graphs that contain spanning trees with a fixed topology.

2.1 Particle swarm optimization algorithm

Consider a non-empty set $C \subseteq \mathbb{R}^n$ and a function $f : C \longrightarrow \mathbb{R}$, where the minimization problem for f at C is to find $x^* \in C$ such that

$$f(x^*) = \min \Big\{ f(X) \ : \ X \in C \Big\}. \tag{1}$$

Particle Swarm Optimization (PSO) is an algorithm that can efficiently find good solutions for problems like (1). The PSO was introduced by Jame Kennedy and Russel Ebhart [9] and is inspired by observing the group behavior of animals in their natural habitat, such as bird flocking or fish schooling. In the PSO algorithm, all particles share information, and each particle has a position and velocity. The position and velocity of each particle are updated iteratively as follows:

$$x_i(k+1) = x_i(k) + v_i(k+1), \tag{2}$$

$$v_i(k+1) = v_i(k) + C_1 \mathrm{Rand}(0,1) \left(\mathrm{pbest}_i(k) - x_i(k)\right) + C_2 \mathrm{Rand}(0,1) \left(\mathrm{gbest}_i(k) - x_i(k)\right), \tag{3}$$

where i is the particle index, k is the iteration number, $x_i(k)$ and $v_i(k)$ are the position and velocity of the particle respectively, C_1 and C_2 are the acceleration constants for the cognitive component and the social component respectively, $\mathrm{pbest}_i(k)$ is the location with the best fitness of all the visited locations of particle i, $\mathrm{gbest}(k)$ is the location with the best fitness among all the visited locations of all the particles, and $\mathrm{Rand}(0,1)$ is a random value between 0 and 1. The pseudo-code of the basic PSO algorithm is presented in Algorithm 1.

The constants C_1 and C_2 are also known as trust parameters. C_1 expresses how much confidence a particle has in itself, while C_2 expresses how much confidence a particle has in its neighbors. If $C_1 = C_2$, then the particles are attracted towards the average of the $\mathrm{pbest}_i(k)$ and the $\mathrm{gbest}(k)$.

It is worth mentioning that the PSO algorithm can be combined with deterministic methods to increase the chance of finding the most likely global optimal point of the function. In addition, in [20] Hybrid formulation for optimization problems that use PSO has been studied. As follow we give some definitions related to cones.

Definition 1. A subset C of a vector space V is a cone (also called a linear cone) if for every $x \in C$ and all positive scalars α, we have $\alpha x \in C$.

Algorithm 1 Particle swarm optimization

```
procedure PSO f(x), S
    for Each particle do
        Initialize x_i, v_i randomly
        Evaluate the fitness f(x_i)
    end for
    repeat
        for each particle i in S do
            if f(x_i) < f(pbest_i) then
                pbest_i = x_i
            end if
            if f(pbest_i) < f(gbest) then
                gbest = pbest_i
            end if
        end for
        for each particle i in S do
            v_i = v_i + C_1 Rnd(0,1) (pbest_i - x_i) + C_2 Rnd(0,1) (gbest - x_i)
            x_i = x_i + v_i
        end for
        k = k + 1
    until k < Iterations
end procedure
```

Definition 2. A cone C is a convex cone if $\alpha x + \beta y \in C$ for any positive scalars α, β, and any $x, y \in C$.

Definition 3. A polyhedral cone can be represented in two different ways; as an intersection of inequalities or as the conical hull of vectors. In the description by the inequality, the polyhedral cone can be given by a matrix $A \in \mathbb{R}^m \times \mathbb{R}^n$ such that $C = \{x \in \mathbb{R}^n | Ax \in \mathbb{R}^m_+\}$. In the conical combination description, it can be represented by a finite set of vectors $v_1, \ldots, v_k$ such that $C = \{\alpha_1 v_1 + \ldots + \alpha_k v_k | \alpha_i \in \mathbb{R}^+ \cup \{0\},\ i = 1, \ldots, k\}$.

In the following, we recall two hyperplane separation theorems that will be needed later [2].

Theorem 1. Suppose that there are two disjoint convex sets C and D. Then, there always exists a hyperplane $a^T x - b = 0$ that separates them. Note that, here, the separation is not strict.

Theorem 2. If C is a convex set and p is a point outside of C, then there always exists a hyperplane $a^T x - b = 0$ that strictly separates C and p, therefore $a^T p - b > 0$ and $a^T x - b < 0$ for all x in C.

2.2 Leader-follower multi-agent system

In the context of MAS, an agent is a dynamical system with a state vector that evolves through time based on its past value and a control input vector. The state of an agent is typically not dependent on any other agent, but the control input is a function of the agent's state vector as well as the state vectors of other agents. A MAS is a set of agents that exchange information and collaborate with each other based on a common control strategy to achieve a goal that cannot be achieved by each agent alone. In a MAS, each agent is characterized by a vertex or node in the graph G, where each edge from node v_i to node v_j represents the information flow of the agent i to agent j. Each edge $(v_j, v_i) \in E$ is associated with a weight $a_{ij} > 0$, which models the strength of the interaction between the nodes. For example, if agent j has higher social standing, then a_{ij} might be selected to be larger so that agent i is more responsive to the behaviors of agent j. A graph is bidirectional if $a_{ij} \neq 0$ and $a_{ji} \neq 0$ so that communication between agents occurs bidirectionally. A graph is said to be undirected if $a_{ij} = a_{ji}$, for all i, j, meaning it is bidirectional and the weights of edges (vi, vj) and (vj, vi) are equal. In a MAS with dynamic graph topology, the communication graph between agents changes with respect to time. However, in this paper, we focus on MAS with undirected fixed graphs, as undirected graphs are very common in practice and many real-world relationships are better modeled with undirected graphs. A MAS can be classified as homogeneous if the dynamics and exchanged information of all the agents are identical, otherwise, it is referred to as a heterogeneous MASs.

A MAS is said to follow a distributed control strategy with topology G if the control input of each agent is a function of its own state (or output) and states (or outputs) of other agents that are in the set of neighbors of that agent in the graph.

The state of node v_i is denoted by $x_i^T \in \mathbb{R}^n$. Then, the state of G is represented as $\mathbf{x} = [x_1^T, \ldots, x_N^T]^T$ in $\mathbb{R}^{n \times N}$. In a consensus problem, all agents must converge to the same value. In a consensus problem with a leader, all nodes of the MAS are coordinated to the state trajectory of the leader node.

2.3 Lyapunov stability

The following is a statement of the Lyapunov theorem, which has been proven in [19].

Theorem 3. Let $V(x,t)$ be a non-negative function with derivative $\dot{V}$ along the trajectories of the system. The following statements hold:

1. If $V(x,t)$ is locally positive definite and $\dot{V} \leq 0$ locally in x and for all t, then the origin of the system is locally stable (in the sense of Lyapunov).

2. If $V(x,t)$ is locally positive definite and decrescent, and $\dot{V} \leq 0$ locally in x and for all t, then the origin of the system is uniformly locally stable (in the sense of Lyapunov).

3. If $V(x,t)$ is locally positive definite and decrescent, and $-\dot{V}$ is locally positive definite, then the origin of the system is uniformly locally asymptotically stable.

4. If $V(x,t)$ is positive definite and decrescent, and $-\dot{V}$ is positive definite, then the origin of the system is globally uniformly asymptotically stable.

Next, we recall a fundamental lemma that we will need in the sequel, which has been proven in [38].

Lemma 1. For any vectors $\mathbf{x}, \mathbf{y}$ of appropriate dimensions and any symmetric positive definite matrix $\mathbf{Z}$ of appropriate dimension, the following inequality holds:

$$\pm 2\mathbf{x}^T\mathbf{y} \leq \mathbf{x}^T\mathbf{Z}\mathbf{x} + \mathbf{y}^T\mathbf{Z}^{-1}\mathbf{y}. \tag{4}$$

Finally, we state the following lemma:

Lemma 2. If at time $t = 0$, the state vectors of the system are not on the sliding surface, i.e., $s(0) \neq 0$, and control u is designed in such a way that the system satisfies the condition $s\dot{s} \leq -\eta\,|s|$, then the state vectors of the system reach the sliding surface in finite time t_r, Such that:

$$t_r \leq \frac{|s(0)|}{\eta}. \tag{5}$$

Proof. In relation $s\dot{s} \leq -\eta\,|s|$, if it is $s \geq 0$, then we have $\dot{s} \leq -\eta$, and hence we have:

$$\int_{s(0)}^{s(t_r)} ds \leq \int_{t=0}^{t=t_r} -\eta\, dt,$$

$$s(t_r) - s(0) \leq -\eta t_r,$$

which gives

$$t_r \leq \frac{s(0)}{\eta}.$$

If $s \leq 0$, then we have $t_r \leq -\frac{s(0)}{\eta}$. Since the right hand side of inequalities $t_r \leq \frac{s(0)}{\eta}$ and $t_r \leq -\frac{s(0)}{\eta}$ is positive, relation (5) is always valid. □

3 Main Results

In this section, we present a study on a fuzzy sliding mode controller based on a separating plane for solving a consensus problem in leader-follower MASs.

Assumption 1. If $f(x(t), \dot{x}(t), t)$ is a real-valued vector function, then there exist two real positive constants W_x and W_v such that f holds in the following inequality

$$\|f(x(t), \dot{x}(t), t) - f(y(t), \dot{y}, t)\| \leq W_x\|x(t) - y(t)\| + W_v\|\dot{x}(t) - \dot{y}(t)\|.$$

Consider a team of N identical agents with one leader. The dynamic equation of the i-th agent is given by :

$$\ddot{x}_i(t) = f(x_i(t), \dot{x}_i(t), t) + g(x_i(t), \dot{x}_i(t), t)u_i, \quad i = 1, \ldots, N. \tag{6}$$

The dynamic equation of the leader is given by:

$$\ddot{x}_0(t) = f(x_0(t), \dot{x}_0(t), t), \tag{7}$$

where $x_i(t), u_i$ are column vectors in $\mathbb{R}^n$ and $f(x_i(t), \dot{x}_i(t), t) \in \mathbb{R}^n$ and $g(x_i(t), \dot{x}_i(t), t) \in \mathbb{R}^n \times \mathbb{R}^n$ are real-valued vector functions for all $i = 1, \ldots, N$; and $g(x(t), \dot{x}(t), t)$ is invertible for all $x(t)$.

We define the global functions F and G as follows:

$$\begin{aligned} F(\mathbf{x}, \dot{\mathbf{x}}, t) &= \left[f(x_1(t), \dot{x}_1(t), t)^T, \ldots, f(x_N(t), \dot{x}_N(t), t)^T\right]^T \in \mathbb{R}^{n\times N}, \\ \mathbf{U} &= [u_1^T, \ldots, u_N^T]^T \in \mathbb{R}^{n\times N}, \\ G(\mathbf{x}, \dot{\mathbf{x}}, t) &= \begin{bmatrix} g(x_1(t), \dot{x}_1(t), t) & \mathbf{O} & \ldots & \mathbf{O} \\ \mathbf{O} & g(x_2(t), \dot{x}_2(t), t) & \ldots & \mathbf{O} \\ \vdots & & \ddots & \vdots \\ \mathbf{O} & \ldots & \mathbf{O} & g(x_N(t), \dot{x}_N(t), t) \end{bmatrix}, \end{aligned} \tag{8}$$

where $\mathbf{O}$ presents a square zero matrix of order n, and $G \in \mathbb{R}^{n\times N} \times \mathbb{R}^{n\times N}$.

Equations (6) and (7) can be written as:

$$\begin{aligned} \dot{x}_i(t) &= v_i(t), \\ \dot{v}_i(t) &= f(x_i(t), v_i(t), t) + g(x_i(t), v_i(t), t)u_i, \quad i = 1, \ldots, N, \end{aligned}$$

and

$$\begin{aligned} \dot{x}_0(t) &= v_0(t), \\ \dot{v}_0(t) &= f(x_0(t), v_0(t), t). \end{aligned}$$

The local neighborhood consensus error for each follower agent is expressed as:

$$\varepsilon_{x_i} = \sum_{j=1}^{N} a_{ij}(x_i(t) - x_j(t)) + b_i(x_i(t) - x_0(t)), \tag{9}$$

$$\varepsilon_{v_i} = \sum_{j=1}^{N} a_{ij}(v_i(t) - v_j(t)) + b_i(v_i(t) - v_0(t)), \tag{10}$$

where $\varepsilon_{x_i}, \varepsilon_{v_i} \in \mathbb{R}^n$ and $b_i \geq 0$.

Note that $b_i > 0$ if and only if the i-th agent is connected to the leader. We have $\varepsilon_{\mathbf{x}} = [\varepsilon_{x_1}, \ldots, \varepsilon_{x_N}]^T \in \mathbb{R}^{n\times N}$ and $\varepsilon_{\mathbf{v}} = [\varepsilon_{v_1}, ..., \varepsilon_{v_N}]^T \in \mathbb{R}^{n\times N}$.

Definition 4. Consider a MAS with N followers and one leader with second-order dynamics. If $x_i(t)$ is the state of the i-th agent and $x_0(t)$ is the state of the leader. We say the MAS will reach a consensus successfully if for all $i = 1, \ldots, N$

$$\lim_{t\to\infty} (\|x_i(t) - x_0(t)\| + \|\dot{x}_i(t) - \dot{x}_0(t)\|) = 0.$$

We define some notations that we will use in the sequel as follows:

$$\begin{aligned} \underline{1} &:= [1, \ldots, 1]^T \in \mathbb{R}^N, \\ \underline{\mathbf{I}} &:= \underline{1} \otimes \mathbf{I}_n, \\ \mathbf{l} &:= (\mathbf{L} + \mathbf{B}) \otimes \mathbf{I}_n. \end{aligned}$$

where **B** is a diagonal matrix $N \times N$ with diagonal elements b_i. Also, **L** is a Laplacian matrix associated with the graph topology of MAS.

Moreover, we define $e_{x_i} := x_i(t) - x_0(t)$, $e_{v_i} := v_i(t) - v_0(t)$, and

$$\begin{aligned} \mathbf{e_x} &:= [e_{x_1}^T, ..., e_{x_N}^T]^T \in \mathbb{R}^{n\times N}, \\ \mathbf{e_v} &:= [e_{v_1}^T, \ldots, e_{v_N}^T]^T \in \mathbb{R}^{n\times N}, \end{aligned}$$

and $\underline{x_0}(t) := \underline{\mathbf{I}} x_0(t)$.

Now, it is easy to see that

$$\varepsilon_{\mathbf{x}} = l\mathbf{e_x}.$$

Let $\underline{f}(x_0(t), v_0(t), t) := \underline{\mathbf{I}} f(x_0(t), v_0(t), t)$, then we define

$$F_e(\mathbf{x}, \dot{\mathbf{x}}, t) := F(\mathbf{x}, \dot{\mathbf{x}}, t) - \underline{f}(x_0(t), v_0(t), t).$$

Therefore, we get

$$\dot{\varepsilon_{\mathbf{x}}} = \varepsilon_{\mathbf{v}}, \tag{11}$$

$$\dot{\varepsilon_{\mathbf{v}}} = l\left(F_e(\mathbf{x}, \mathbf{v}, t) + G(\mathbf{x}, \mathbf{v}, t)U\right). \tag{12}$$

3.1 Sliding mode control

To design a standard sliding mode controller, sliding surfaces $s_i = 0$ must be set for $i = 1, \dots, N$ such that

$$s_i = c_i \mathbf{I}_n \varepsilon_{x_i} + \dot{\varepsilon}_{x_i}, \quad c_i > 0, \tag{13}$$

and

$$\dot{s}_i = c_i \mathbf{I}_n \dot{\varepsilon_{x_i}} + \ddot{\varepsilon}_{x_i}. \tag{14}$$

It is evident that $s_i, \dot{s}_i \in \mathbb{R}^n$ for $i = 1, \dots, N$, and if $s_i = 0$, then $\varepsilon_{v_i} = -c_i \varepsilon_{x_i}$. Hence, $\varepsilon_{x_i} \to 0$ and $\varepsilon_{v_i} \to 0$. Therefore, the control input u_i must be determined for all i such that each ε_{x_i} can be driven to the sliding surface $s_i = 0$. However, finding u_i directly from $s_i = 0$ and $\dot{s}_i = 0$ is not possible.

The sliding surfaces can be represented in matrix form as

$$\mathbf{S} = \mathbf{C} \varepsilon_{\mathbf{x}} + \varepsilon_{\mathbf{v}}, \tag{15}$$

where

$$\mathbf{S} = \left[s_1^T, \dots, s_N^T \right]^T \in \mathbb{R}^{n \times N},$$

and $\mathbf{C} = \text{diag}(c_1, \dots, c_N) \otimes \mathbf{I}_n$ in $\mathbb{R}^{n \times N} \times \mathbb{R}^{n \times N}$. Then

$$\dot{\mathbf{S}} = \mathbf{C} \varepsilon_{\mathbf{v}} + l F_e(\mathbf{x}, \mathbf{v}, t) + l \mathbf{G} \mathbf{U}. \tag{16}$$

Let $\mathbf{M} := l\mathbf{G}$, and let C be the polyhedral cone generated by the rows of $\mathbf{M}$. We define $C = \left\{ x : \ x = M^T y, \ y \in \mathbb{R}_+^{n \times N} \right\}$. If $\mathbf{S}$ is not in polyhedral cone generated by rows of $\mathbf{M}$, we can use Theorem 2,to propose a separating plane $a^T x - b = 0$ such that

$$a = \mathbf{S} - \hat{c},$$

and

$$b = \frac{\| \mathbf{S} \| - \| \hat{c} \|}{2},$$

where

$$\hat{c} = \mathbf{arg} \min \left\{ \| \mathbf{S} - c \| | \ c \in C \right\}. \tag{17}$$

Here, $\hat{c}$ is obtained by solving the optimization problem

$$\hat{c} = \mathbf{arg} \min \left\{ \ \| \mathbf{S} - M^T y \|, y \in \mathbb{R}_+^{n \times N} \right\}. \tag{18}$$

It is obvious that $\hat{c}$ may not be a solution of the system of the equations $M^T y = \mathbf{S}$. Thus, to obtain $\hat{c}$, we need to solve the minimization problem (18), which can be achieved using Algorithm 1. Hence we define

$$\widetilde{U} = a - \mathbf{M}^{-1}\underline{b} - \mathbf{M}^{-1}\mathbf{K}\mathbf{S}, \tag{19}$$

where $\underline{b} := b\underline{1}$ and

$$\mathbf{K} := \kappa \mathbf{I}_N, \tag{20}$$

such that

$$\kappa := 1 + \frac{\beta^T \mathbf{S}}{\| \mathbf{S} \|^2}, \tag{21}$$
$$\beta := [\beta_1, \ldots, \beta_N]^T,$$

and

$$\beta_i := a^T M_i - b.$$

If $\mathbf{S}$ is already in the cone C, we define $\mathbf{U}_{SW} := -\mathbf{M}^{-1}\mathbf{S}$. In other words, we can say that

$$\mathbf{U}_{SW} = \begin{cases} \widetilde{\mathbf{U}}, & \text{if } \mathbf{S} \notin C, \\ -\mathbf{M}^{-1}\mathbf{S}, & \text{if } \mathbf{S} \in C. \end{cases} \tag{22}$$

We assume that $\mathbf{M}$ is positive definite and by using Theorem 2, we can say that $\widetilde{U}$ defined in (19) satisfies

$$\mathbf{S}^T \mathbf{M} \widetilde{\mathbf{U}} = -\xi \| \mathbf{S} \|, \quad \xi \geq 0.$$

We denote the SMC based on the separating plane by $\mathbf{U_{SM}}$, which is determined as

$$\mathbf{U_{SM}} = \mathbf{M^{-1}} \left(-\mathbf{C}\varepsilon_{\mathbf{v}} - lFe(\mathbf{x}, \mathbf{v}, t) \right) + \rho \mathbf{U}_{SW}, \tag{23}$$

where ρ is an arbitrary positive number.
To stabilize the sliding surface $\mathbf{S}$ and the SMC, proposed in (23), we consider the Lyapunov function

$$V = \frac{1}{2}\mathbf{S}^T\mathbf{S}, \tag{24}$$
$$\dot{V} = \mathbf{S}^T\dot{\mathbf{S}}. \tag{25}$$

Using (16), (15) and (25), we have

$$\dot{V} = \varepsilon_{\mathbf{v}}^T \mathbf{C} \varepsilon_{\mathbf{v}} + \varepsilon_{\mathbf{x}}^T \mathbf{C}^2 \varepsilon_{\mathbf{v}} + \varepsilon_{\mathbf{x}}^T \mathbf{C} l F_e + \varepsilon_{\mathbf{v}}^T l F_e + \mathbf{S}^T l \mathbf{G} \mathbf{U_{SM}}.$$

It is important to note that for both cases of switching control, $\mathbf{U_{SM}}$ satisfies the following condition.

$$\dot{V} \leq -\eta \| \mathbf{S} \|, \qquad \eta \geq 0. \tag{26}$$

Remark 1. In the reaching phase, the state vectors of the system reach the sliding surface, and in the sliding phase, the sliding mode controller in 23 drives the system to the equilibrium point. Using the proposed sliding mode controller (23) in (16), we obtain

$$\begin{aligned}\dot{\mathbf{S}} =& \mathbf{M}a - b - \left(1 + \frac{\beta^T \mathbf{S}}{\| \mathbf{S} \|^2}\right)\mathbf{S},\\ \dot{\mathbf{S}} =& \beta - \left(1 + \frac{\beta^T \mathbf{S}}{\| \mathbf{S} \|^2}\right)\mathbf{S}.\end{aligned}$$

which shows an exponential term for all follower agents. Therefore, the state vector of each agent is forced to reach the sliding surface faster.

Remark 2. The proposed sliding mode controller (23) satisfies (26), and according to Lemma 2, the reaching time is finite when $\mathbf{S}(0) \neq \mathbf{0}$ where $\mathbf{0}$ is the zero vector in $\mathbb{R}^{n\times N}$.

3.2 Fuzzy sliding mode controller

In theory, an ideal sliding mode implies infinite switching frequency, which results in the chattering phenomenon. Chattering is a harmful phenomenon that causes high wear of moving mechanical parts, and high heat losses in power circuits. In the previous subsection, the proposed sliding mode controller may have faced the chattering phenomenon due to the change of $\mathbf{S}$ at each moment. To address this issue, a fuzzy sliding mode controller is used. For the proposed fuzzy sliding mode controller, the fuzzy rules are assumed as follows:

$$\textbf{If} \quad \mathbf{S} \notin \text{Cone}(\mathbf{M}) \quad \textbf{then} \quad \mu = 1. \tag{27}$$

$$\textbf{If} \quad \mathbf{S} \in \text{Cone}(\mathbf{M}) \quad \textbf{then} \quad \mu = 0. \tag{28}$$

The rule (27) specifies that $\mathbf{U_{SM}}$ is obtained using the separating plane when $\mathbf{S}$ does not belong to the cone generated by the rows of $\mathbf{M}$. Specifically, we have

$$\mathbf{U_{SM}} = \mathbf{M^{-1}}\left(-\mathbf{C}\varepsilon_{\mathbf{x}} - lF_e(\mathbf{x}, \mathbf{v}, t)\right) + \theta(a - \mathbf{M}^{-1}\underline{b} - \mathbf{M}^{-1}\mathbf{KS}).$$

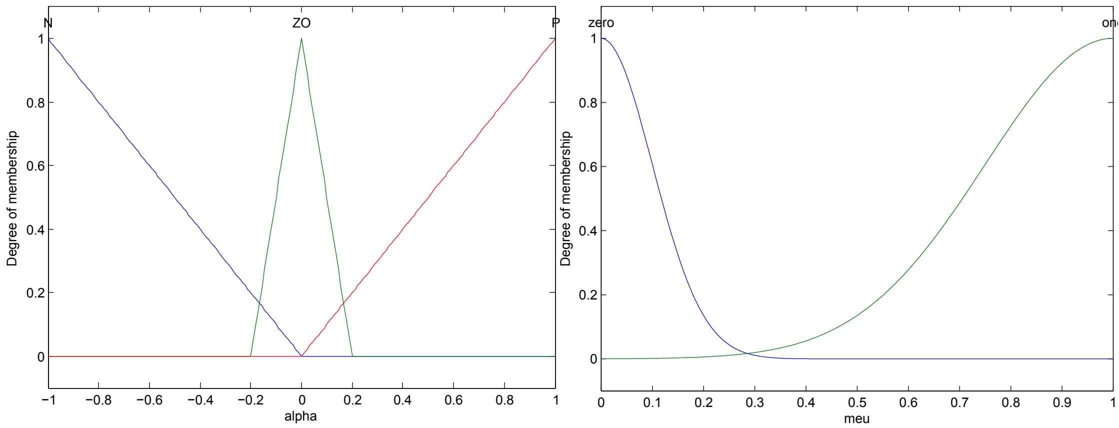
Similarly using (28), $\mathbf{U_{SM}}$ is obtained as:

$$\mathbf{U_{SM}} = \mathbf{M^{-1}}\left(-\mathbf{C}\varepsilon_{\mathbf{x}} - lFe(\mathbf{x}, \mathbf{v}, t)\right) - \theta\mathbf{S},$$

when $\mathbf{S}$ belongs to the cone C. Based on these rules, the fuzzy sliding mode controller is defined as:

$$\begin{aligned}\mathbf{U_{FSM}} = & \mathbf{M^{-1}}\left(-\mathbf{C}\varepsilon_{\mathbf{x}} - lFe(\mathbf{x}, \mathbf{v}, t)\right)\\ & + \mu\theta(a - \mathbf{M}^{-1}\underline{b} - \mathbf{M}^{-1}\mathbf{KS}) - (1-\mu)\theta\mathbf{S}.\end{aligned} \tag{29}$$

The input variable of the fuzzy system is determined as the minimum component of the vector y, which is obtained by solving the equations $\mathbf{S} - M^T y = 0$. If $y \in \mathbb{R}_+^{n\times N}$, then its smallest component is positive, and if $y \notin \mathbb{R}_+^{n\times N}$, then its smallest component is negative. Let positive and negative linguistic variables be transformed into fuzzy values with the input membership function. By adopting the rules established in (28), the Mamdani inference system, and the inverse defuzzification method, the output of the fuzzy system μ is obtained. The input and output fuzzy membership functions are illustrated in Figure 1. The block diagram for the simulation of the separating plane-based sliding mode controller simulation is presented in Figure 2. The blocks shown in the figure are user-defined functions in MATLAB.



(a) Input membership function **(b)** Output membership function

Figure 1: Fuzzy membership functions.

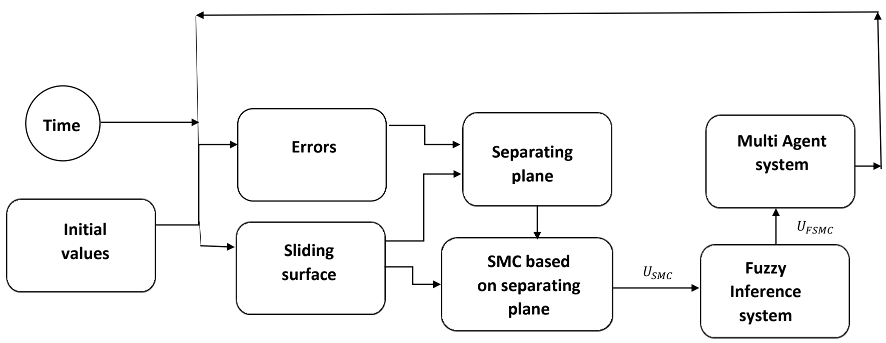


Figure 2: Block Diagram of Simulation Model used.

4 Implementation

This section provides three numerical examples to demonstrate the effectiveness of the theoretical results obtained in Sections 3.

In these examples:

- We aim to solve the minimization problem min$\{\| \mathbf{S} - \mathbf{M}^T y \|,\ y \in \mathbb{R}_+^{n\times N}\}$ using Algorithm 1, where **S** is a sliding surface, $\mathbf{M} = l\mathbf{G}$ and y belongs to $\mathbb{R}_+^{n\times N}$.
- Since **S** is a function of the time t, the objective function changes at each time step.
- For Algorithm 1, we set $C_1 = C_2 = 0.001$, the number of iterations 10, and the number of particles to 50.

Furthermore, the input and output fuzzy membership functions used in all three examples are presented in Figure 1. To implement these examples, MATLAB Simulink is employed with a step size of 10^{-1}.

Example 1. We present Example 1 to illustrate our approach for a MAS consisting of one leader and three followers. The leader is indexed by 0, and the followers are indexed by 1, 2, and 3. The exchange of information between agents is represented by a fixed, connected and undirected graph topology as shown in Figure 3. The corresponding Laplacian matrix **L** is given by:

$$\mathbf{L} = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}.$$

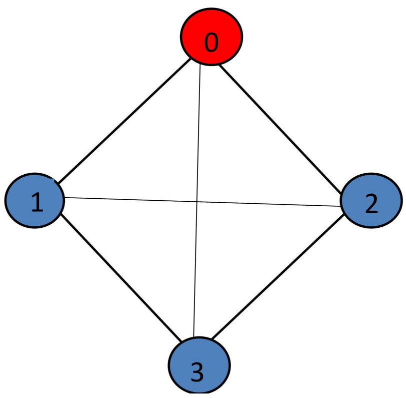


Figure 3: Graph topology of Example 1.

The states of the leader and followers are vectors in $\mathbb{R}^3$. The dynamics of the followers are defined as follows:

$$\begin{aligned}&\dot{x}_i(t) = v_i(t),\\&\dot{v}_i(t) = [0.85x_{i1}(t) - 0.6v_{i1}(t) - 1 + u_{i1}, x_{i2}(t) - v_{i2}(t) - 1 + u_{i2}, x_{i3}(t) - 2v_{i3}(t) - 1 + u_{i3}]^T,\end{aligned} \tag{30}$$

where $i = 1, 2, 3$. The dynamics of the leader are given by:

$$\begin{aligned}&\dot{x}_0(t) = v_0(t),\\&\dot{v}_0(t) = [0.85x_{01}(t) - 0.6v_{01}(t) - 1, x_{02}(t) - v_{02}(t) - 1, x_{03}(t) - 2v_{03}(t) - 1]^T.\end{aligned} \tag{31}$$

It should be noted that the nonlinear term $f_i(x, v, t)$ in (30) satisfies Assumption 1 where $W_x = 1$ and $W_v = 2$.

To simulation Example 1, we set $\rho = \dfrac{1}{1+ \parallel \mathbf{S} \parallel}$ in (23). The initial position and velocity of the leader are vectors in $\mathbb{R}^3$:

$$x_0(0) = [0, 0, 0]^T, \quad v_0(0) = [0.2, 0.25, 0.8]^T.$$

The initial positions of the three followers are:

$$x_1(0) = [4, 3, 1]^T, \quad x_2(0) = [1.5, -2.5, 1.2]^T, \quad x_3(0) = [-2.5, 5, 1]^T.$$

The initial velocity conditions of the followers are:

$$v_1(0) = [-0.8, -0.5, 1.8]^T, \quad v_2(0) = [1.5, 1.6, 1.2]^T, \quad v_3(0) = [-2.5, -1.2, 1]^T.$$

As shown in Figures 4, 5, and 6, the designed controller successfully makes all nodes follow the leader with an error of less than 10^{-8} in 30 seconds.

Figures 7, 8, and 9, depict the error trajectory of all follower agents in $\mathbb{R}^3$, which rapidly and continuously decreases. To provide a clearer view, we zoomed in on the figures from $t = 28$ to $t = 30$. The simulation results demonstrate that the initial error converges to zero with an error of less than 10^{-9}.

Moreover, Figures 10, 11, and 12 display the velocity of all nodes in $\mathbb{R}^3$. It is evident from Figures 4, 5, and 6 that the trajectories of the follower agents converge to that of the leader.

The equations of the leader trajectory are as follows:

$$\begin{aligned}x_0(t) &= 0.1031\,(\exp(0.6695t) - \exp(-1.2695t)) + \frac{100}{85},\\y_0(t) &= -0.1118\,(\exp(-1.6180t) - \exp(0.6180t)) + 1,\\z_0(t) &= 0.8t\exp(-t) + 1,\end{aligned} \tag{32}$$

where $x_0(t)$, $y_0(t)$ and $z_0(t)$ represent the components of the trajectory in the x, y, and z axes, respectively. Regarding the equations in (32), It can be observed that the values of the trajectory

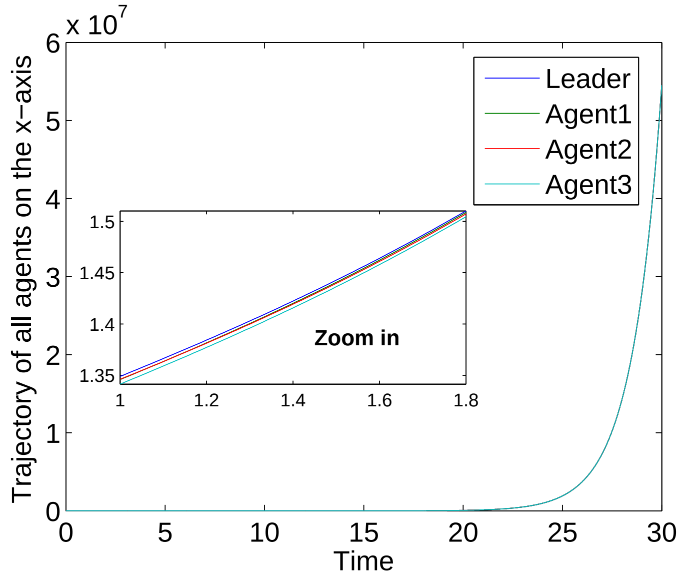


Figure 4: Trajectory of all the agents in x-axis in 30 seconds.

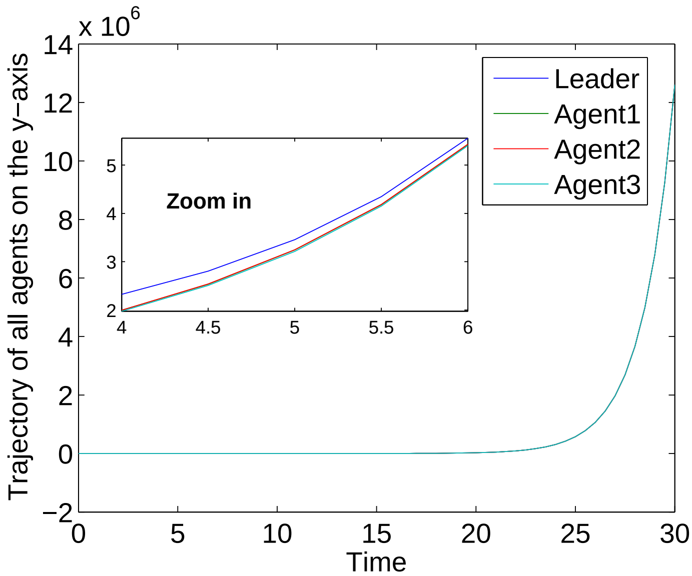


Figure 5: Trajectory of all the agents in y-axis in 30 seconds.

and velocity of the agents on the x and y axes increase significantly to the variable t. Figures 4, 5, and 6 demonstrate that the trajectories of the follower agents converge to the leader trajectory. Additionally, the simulation results in Figures 10, 11, and 12 indicate that the velocity of all follower agents is consistent with the velocity equations of the leader.

We use the fuzzy system to overcome the chattering phenomenon as illustrated in Figures 13, 14, and 15.

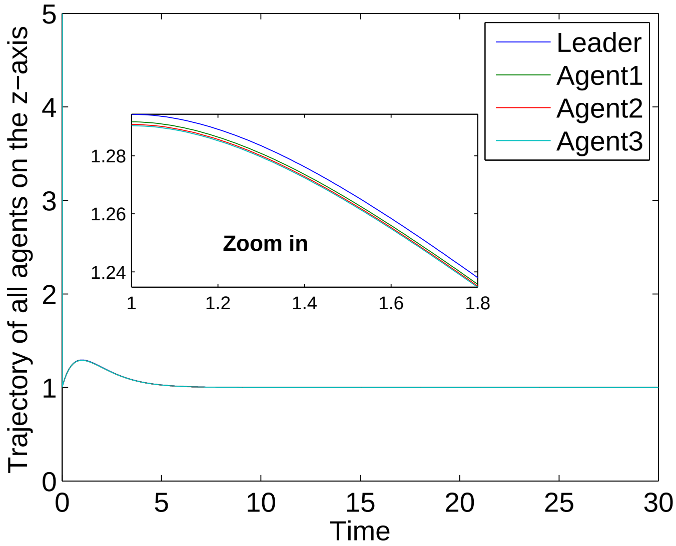


Figure 6: Trajectory of all the agents in z-axis in 30 seconds.

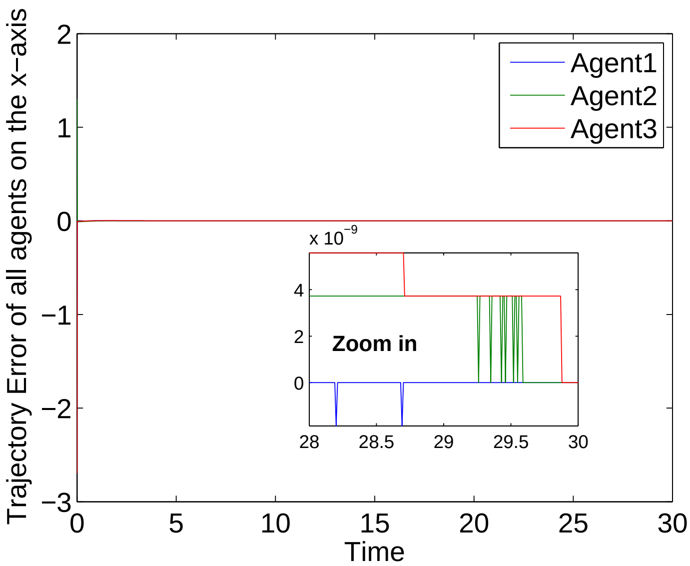


Figure 7: Error trajectory of all the agents in x-axis in 30 seconds.

4.1 SMC based on separating plane without any fuzzy controllers

In practical applications of sliding mode controls, the phenomenon of chattering may occur, which manifests as oscillations with finite frequency and amplitude. Chattering is a detrimental phenomenon that can lead to low control accuracy, high wear of moving mechanical parts, and high heat losses in power circuits.

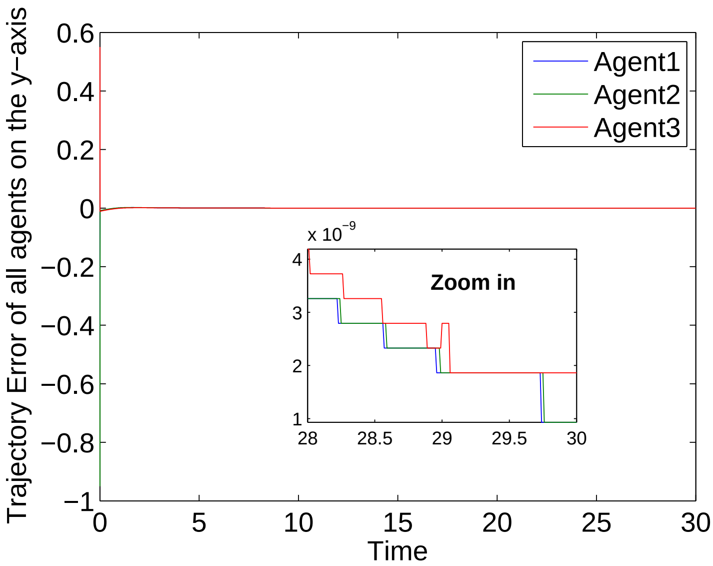


Figure 8: Error trajectory of all the agents in y-axis in 30 seconds.

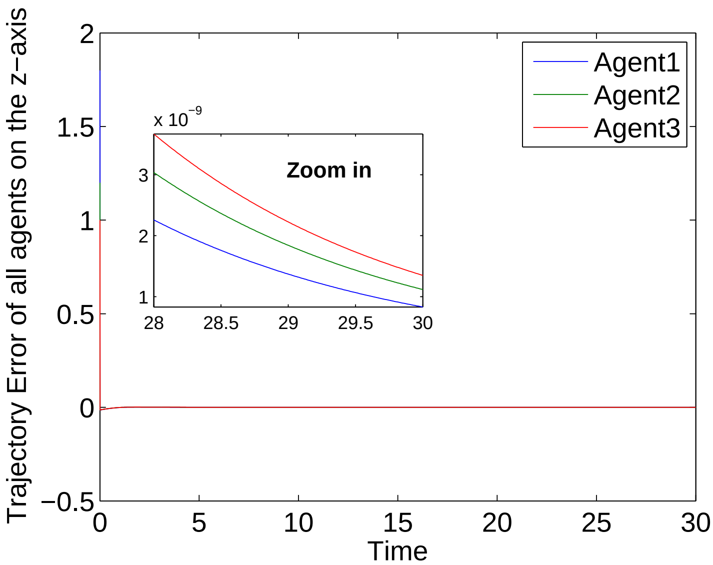


Figure 9: Error trajectory of all the agents in z-axis in 30 seconds.

We implement the proposed controller on the same MAS defined by (30) and (31) without using fuzzy rules. As shown in Figures 16 and 17, the obtained results are satisfactory, and the states of the agents and leader reach a consensus. However, the lack of a fuzzy controller leads to the chattering phenomenon in the control of the MAS as shown in Figure 18.

Now, we give an example to show an application of an SMC based on a separating plane for homogeneous leader-follower MASs with a fixed graph topology.

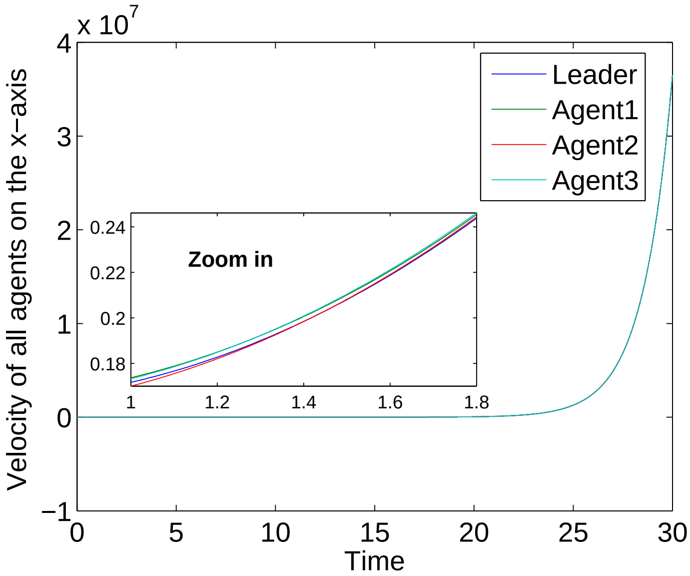


Figure 10: Velocity of all the agents in x-axis in 30 seconds.

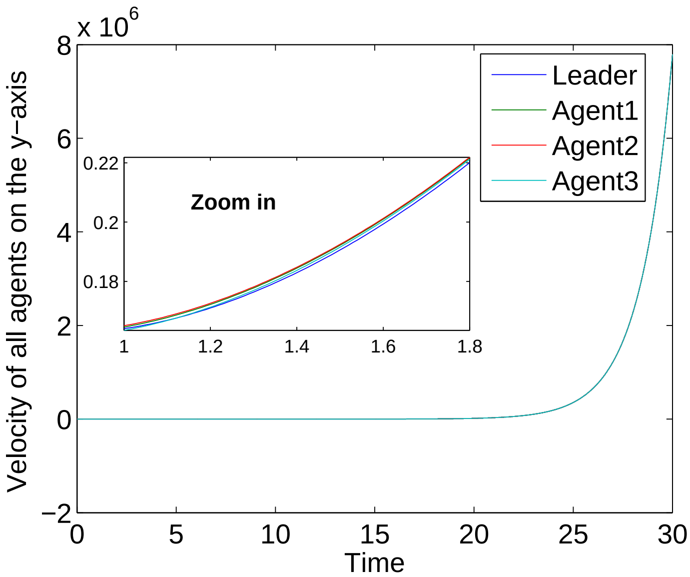


Figure 11: Velocity of all the agents in y-axis in 30 seconds.

Example 2. In this study, we consider a homogeneous multi-agent leader-follower system consisting of six agents to demonstrate the control presented in Section 3. The exchange of information between agents is shown by the graph topology in Figure 19. The dynamic equations of the agents are given by the Van der Pol oscillator equation (33), defined as follows

$$\dot{x}_i(t) = v_i(t),$$

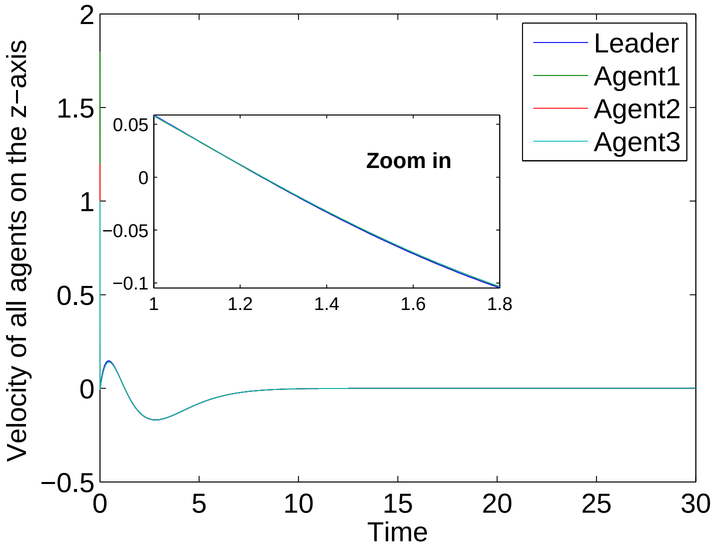


Figure 12: Velocity of all the agents in z-axis in 30 seconds.

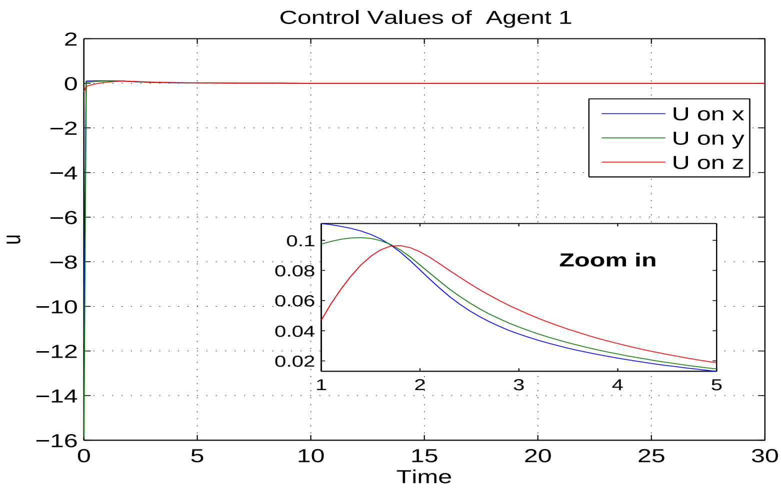


Figure 13: FSMC based on a separating plane of the first agent in three dimensions.

$$\dot{v}_i(t) = -x_i(t) + 3(1 - x_i^2(t))v_i(t) + u_i, \quad i = 1, \dots, 5, \tag{33}$$

where $x_i(t)$ and $v_i(t)$ are real-valued functions, and $i = 1, \dots, 5$. The dynamic equation of the leader is defined as follows

$$\begin{aligned} \dot{x}_0(t) &= v_0(t), \\ \dot{v}_0(t) &= -x_0(t), \end{aligned} \tag{34}$$

where $x_i(t), v_i(t) \in \mathbb{R}$ for all $i = 0, \dots, 5$. The initial values of the followers are given as

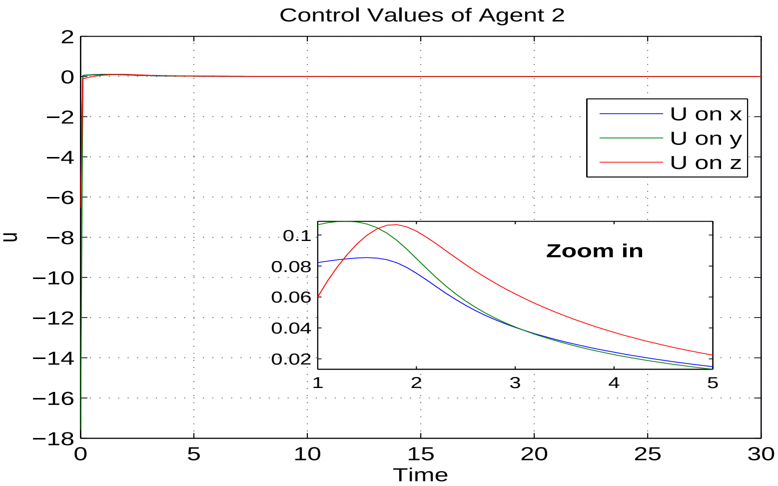


Figure 14: FSMC based on a separating plane of the second agent in three dimensions.

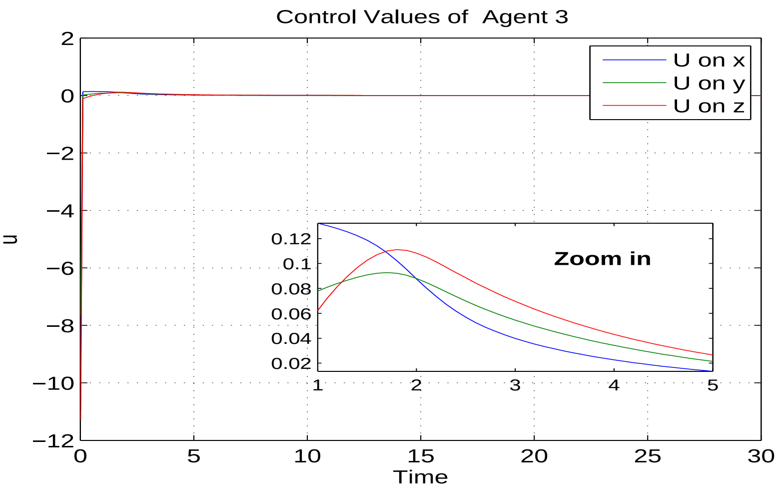
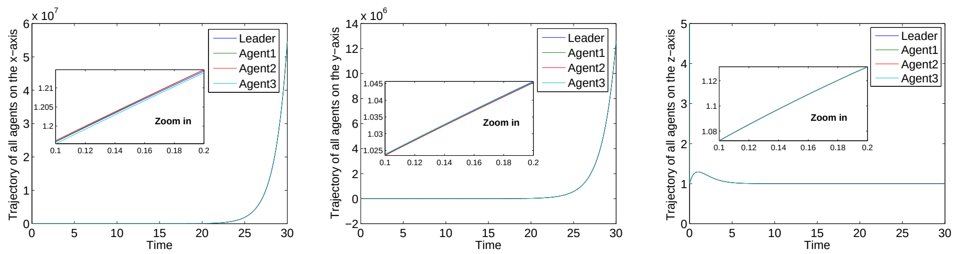
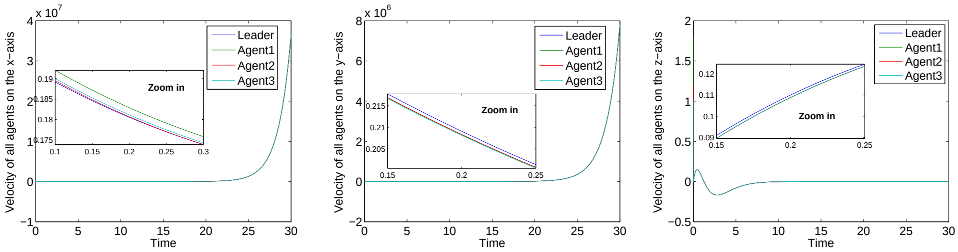


Figure 15: FSMC based on a separating plane of the third agent in three dimensions.



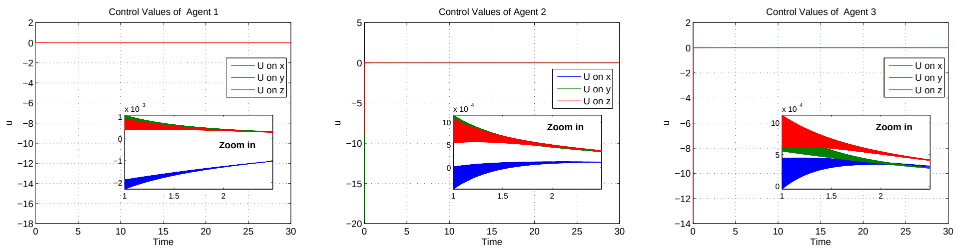
(a) First agent **(b)** Second agent **(c)** Third agent

Figure 16: Trajectory of all the agents using SMC based on a separating plane without using a fuzzy system.



(a) First agent **(b)** Second agent **(c)** Third agent

Figure 17: Velocity of all the agents using SMC based on a separating plane without using a fuzzy system.



(a) First agent **(b)** Second agent **(c)** Third agent

Figure 18: SMC based on a separating plane without using a fuzzy system in 30 seconds.

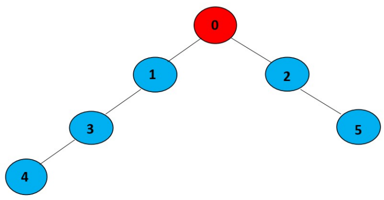


Figure 19: Graph topology of Example 2.

$$
\begin{aligned}
&x_1(0) = 0.1, \quad x_2(0) = 0.4, \quad x_3(0) = 0.2, \quad x_4(0) = 0.3, \quad x_5(0) = 0.5, \\
&v_1(0) = 0.2, \quad v_2(0) = 0.1, \quad v_3(0) = 0.1, \quad v_4(0) = 0.1, \quad v_5(0) = 0.1,
\end{aligned}
$$

and the initial values of the leader are given by

$$
\begin{aligned}
x_0(0) &= 0.2, \\
v_0(0) &= 0.
\end{aligned}
$$

The Laplacian matrix $\mathbf{L}$ of the graph topology of the MAS is given as

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & 2 & -1 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \end{bmatrix}.$$

In this example, we apply two control methods to reach a consensus among the agents. The first control method is FSMC based on a separating plane, and the second one is an equivalent sliding mode control. The simulation results show that the FSMC based on a separating plane achieves consensus faster than the equivalent sliding mode control.

4.2 FSMC based on separating plane

By taking the parameter $\rho = \dfrac{1}{1+ \parallel \mathbf{S} \parallel}$ corresponding to the provided control, the tracking errors of the trajectory and velocity of the follower nodes are shown in Figures 20 and 21 respectively. Note that since the step size is 0.1, it appears from Figures 20 and 21 that the trajectory and velocity errors decrease after 10 iterations. Moreover, the numerical results indicate that the error value is less than 10^{-8}.

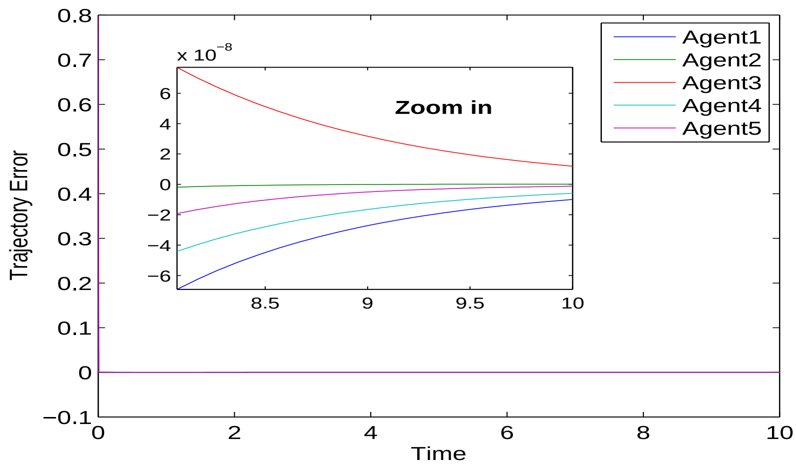


Figure 20: Trajectory error of all the agents in 10 seconds.

Figures 22 and 23 depict the trajectory and velocity curves of the follower agents and the leader, respectively. It is observed that similar to Example 1, the states of the follower agents reach a consensus very quickly, with an error less than 10^{-8}, and coincide with the state of the leader. The simulation results demonstrate that the accuracy of this experiment is less than 10^{-8}. To examine the behavior of all the followers and the leader, that is the velocity and the

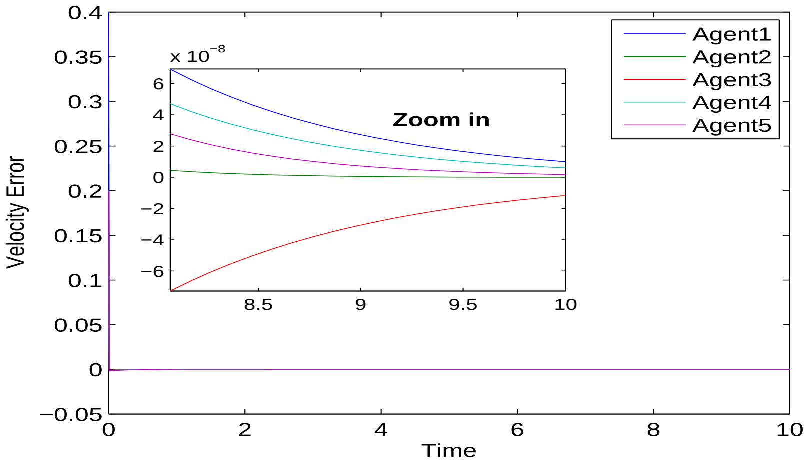


Figure 21: Velocity error of all the agents in 10 seconds.

trajectory of the agents are shown in Figures 22 and 23 using the zoom commands of MATLAB software.

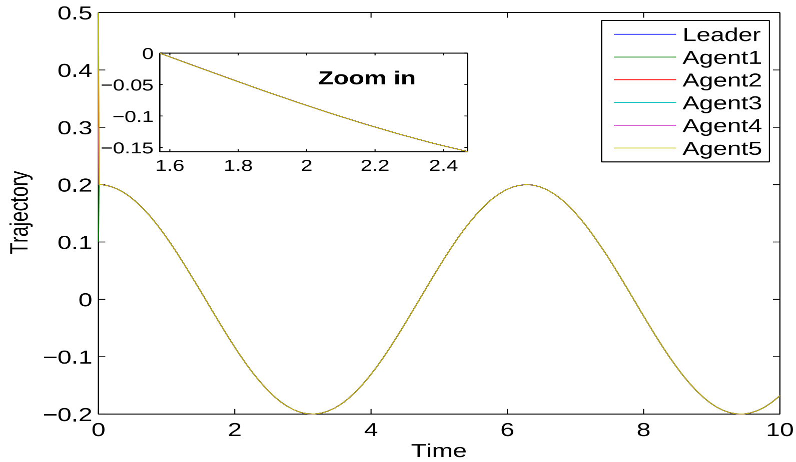


Figure 22: Trajectory of all the agents in 10 seconds.

In the sequel, Figure 24 shows that the proposed fuzzy controller has no chattering phenomenon.

Figure 25 displays the variations of the Lyapunov function to the variable t for the five nodes.

It is important to note that the objective function's behavior changes with respect to the variable t. Therefore, we set $t = 0.1$ and plotted the graph of the objective function corresponding to the number of iterations in Figure 26.

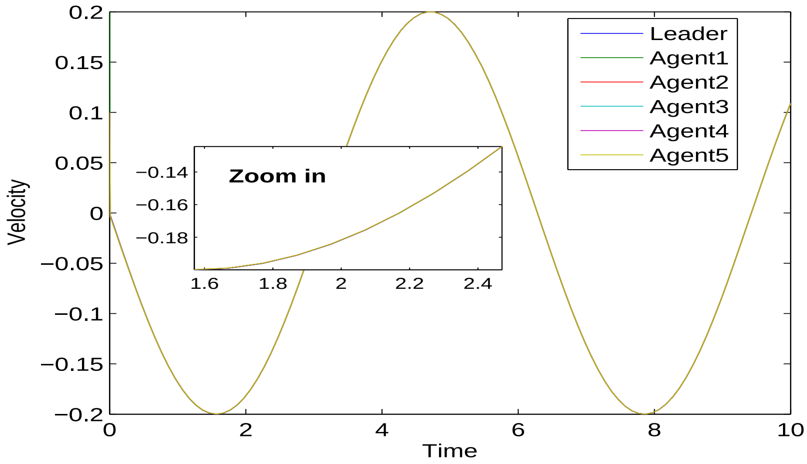


Figure 23: Velocity of all the agents in 10 seconds.

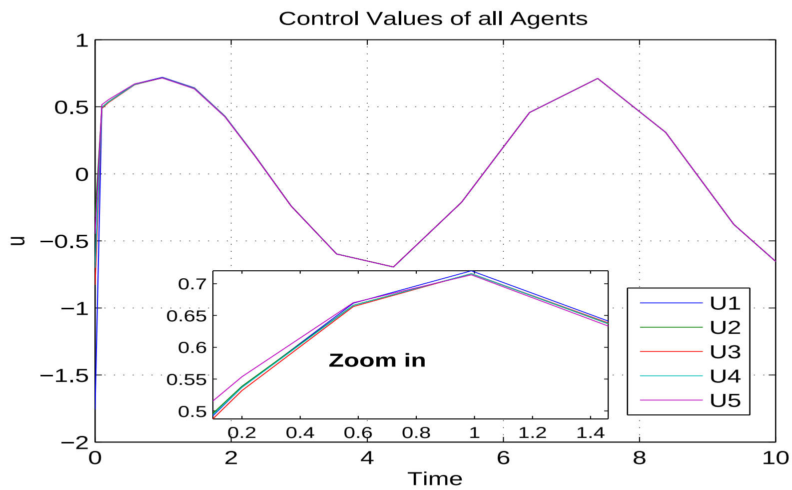


Figure 24: FSMC based on a separating plane in 10 second.

We ran this example under various conditions, and in most cases, the chattering phenomenon was not observed. This occurred because the **S** vector was frequently found outside the cone generated by the rows of **M** during the simulation run, preventing the chattering phenomenon from being observed.

4.3 SMC based on an equivalent control

To achieve a consensus in the van der pol MAS defined by equations (33) and (34), we implement a sliding mode control based on an equivalent control. The results of this implementation

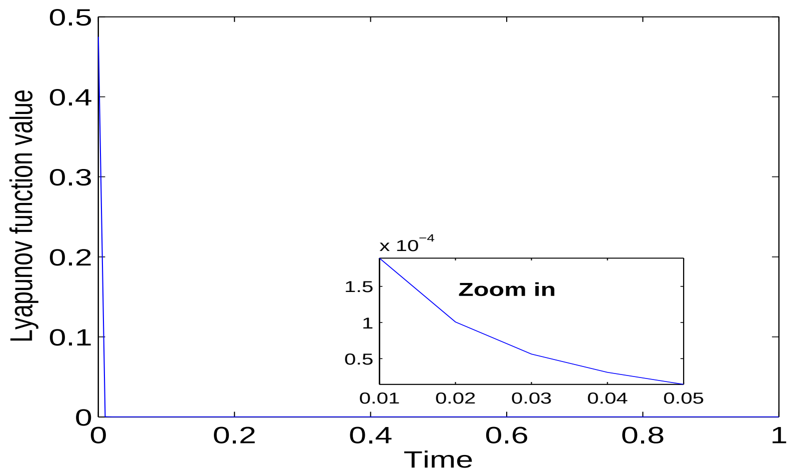


Figure 25: Lyapunov function of the MAS.

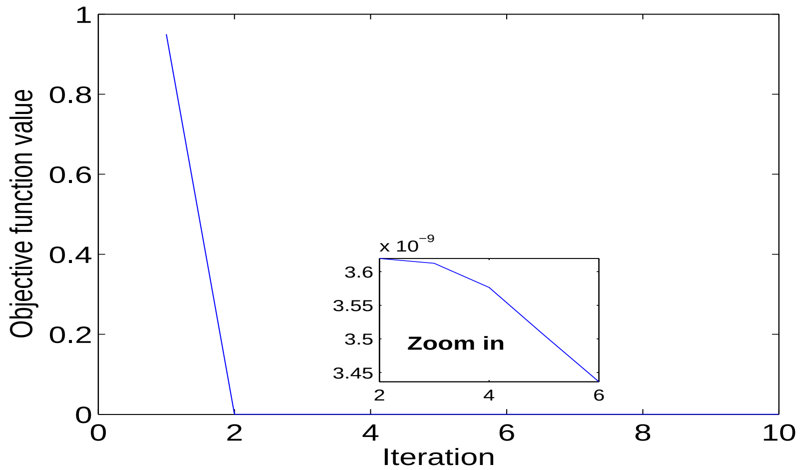


Figure 26: Objective function values with respect to the iterations at $t = 0.1$.

are presented in Figures 27 and 28. It is noteworthy that the proposed SMC in this paper shows a faster convergence rate toward consensus compared to other SMC methods, such as the SMC based on an equivalent control. Additionally, the chattering phenomenon in the sliding mode controller of the MAS can be seen in Figure 29. Comparing the proposed controller with the equivalent sliding mode controller, it can be seen that the former is more accurate, while the latter faces the chattering phenomenon.

Example 3. One of the main reason for using sliding mode or fuzzy control is to mitigate the effects of uncertainty and disturbance. In this Example, we demonstrate the effectiveness of the control presented in Section 3, by considering a homogeneous multi-agent leader-follower system with six agents. The dynamics of the agents are subject to uncertainty and are defined

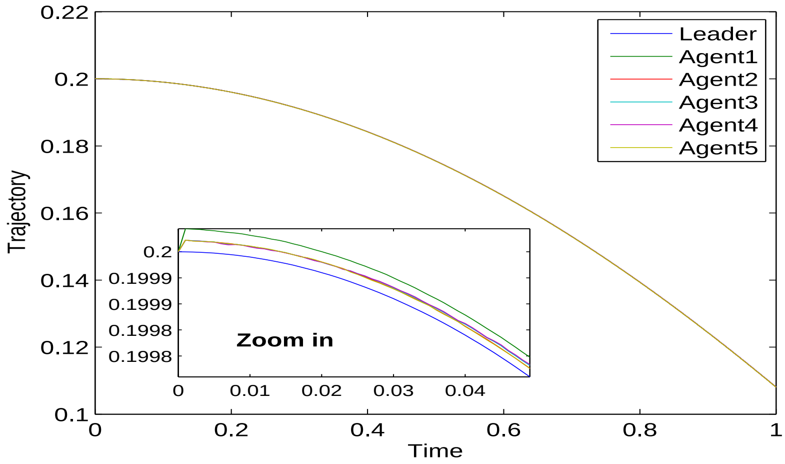


Figure 27: Trajectory of all the agents using SMC based on an equivalent control.

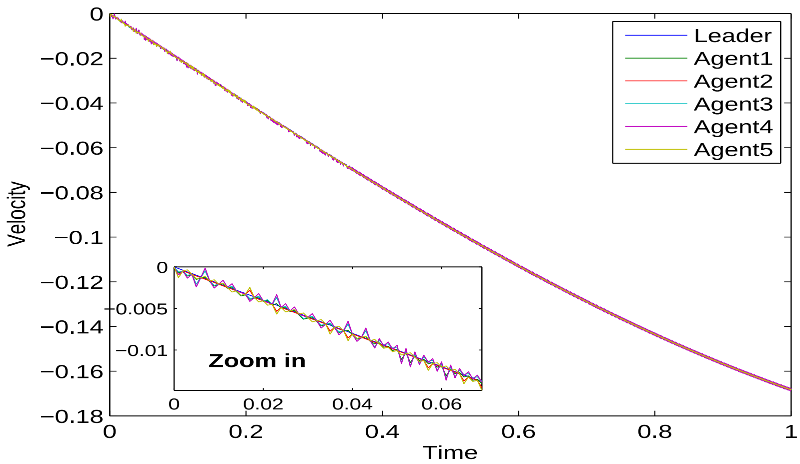


Figure 28: Velocity of all the agents using SMC based on an equivalent control.

as follow:

$$\begin{aligned} &\dot{x}_i(t) = v_i(t), \\ &\dot{v}_i(t) = -0.5v_i(t) + x_i(t) - x_i^3(t) + 13u_i(t) + w(t), \quad i = 1, \dots, 5, \end{aligned} \tag{35}$$

in which $w(t) = \dfrac{1}{2+4t^2}$ is a bounded uncertainty. The exchange of information between agents is shown by a graph topology in Figure 30.

Also, the dynamics of the leader are

$$\begin{aligned} &\dot{x}_0(t) = v_0(t), \\ &\dot{v}_0(t) = 2\pi \cos(2\pi t). \end{aligned} \tag{36}$$

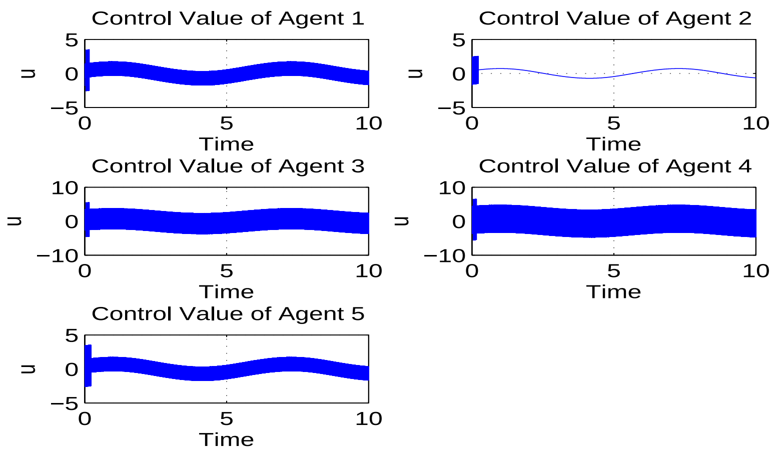


Figure 29: The SMC based on an equivalent control for all the agents in 10 second.

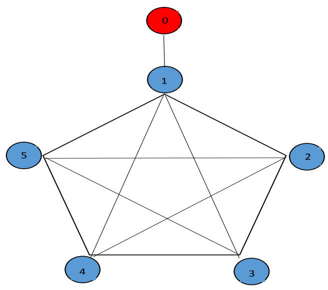


Figure 30: Graph topology of Example 3.

The initial values of the followers are given as

$$x_1(0) = 0.5, \quad x_2(0) = 1, \quad x_3(0) = -0.5, \quad x_4(0) = 0.5, \quad x_5(0) = 0.7,$$
$$v_1(0) = -1, \quad v_2(0) = 1, \quad v_3(0) = -0.5, \quad v_4(0) = 0.5, \quad v_5(0) = 0.5,$$

and the initial values of the leader are given as

$$x_0(0) = 0,$$
$$v_0(0) = 2\pi.$$

The Laplacian matrix $\mathbf{L}$ of the graph topology of the MAS is provided as

$$\mathbf{L} = \begin{bmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{bmatrix}.$$

As shown in Figures 31 and 32, the states of the follower agents quickly reach a consensus with an error less than 10^{-4}, similar to Examples 1 and 2.

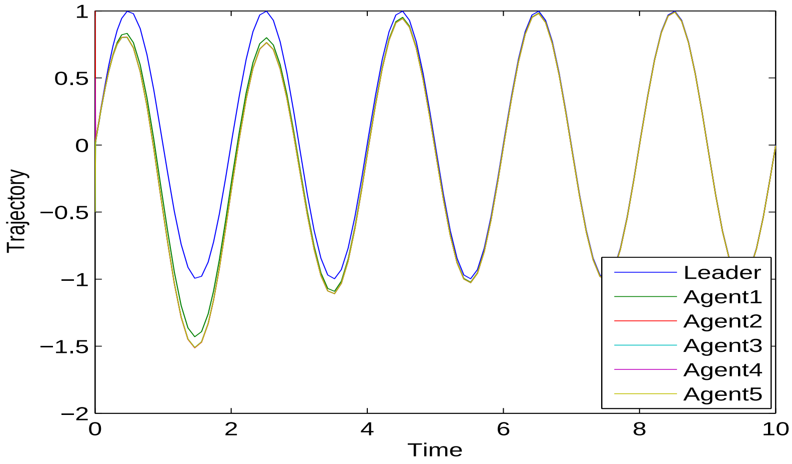


Figure 31: Trajectory of all the agents in 10 seconds.

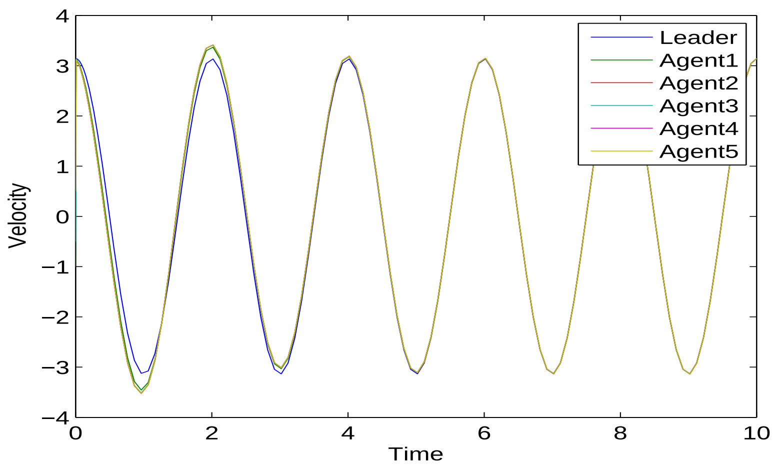


Figure 32: Velocity of all the agents in 10 seconds.

By taking the parameter $\rho = \dfrac{1}{1+ \parallel \mathbf{S} \parallel}$ corresponding to the provided control, the tracking errors of the trajectory and velocity of the follower nodes are shown in Figures 33 and 34 respectively.

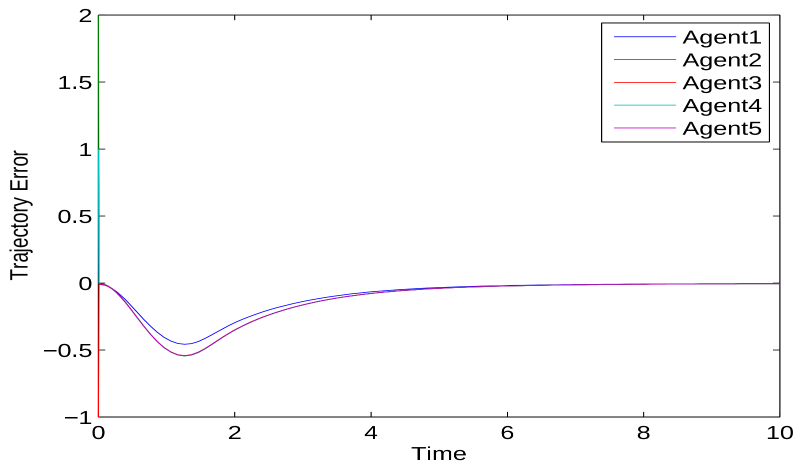


Figure 33: Trajectory error of all the agents in 10 seconds.

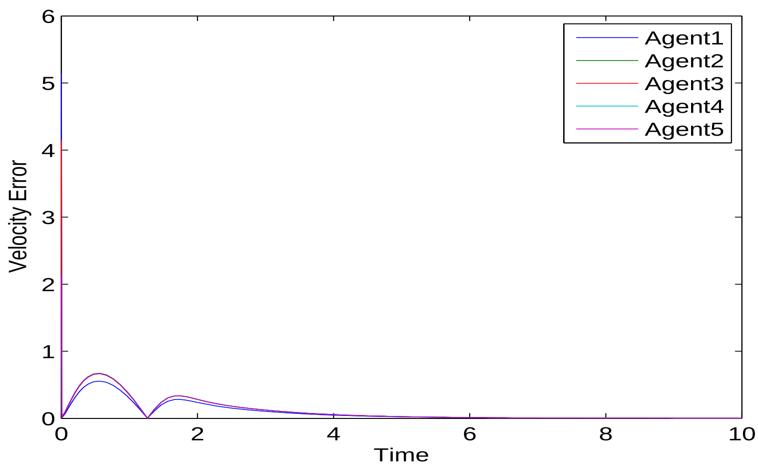
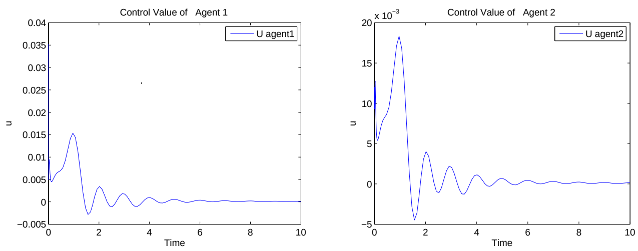


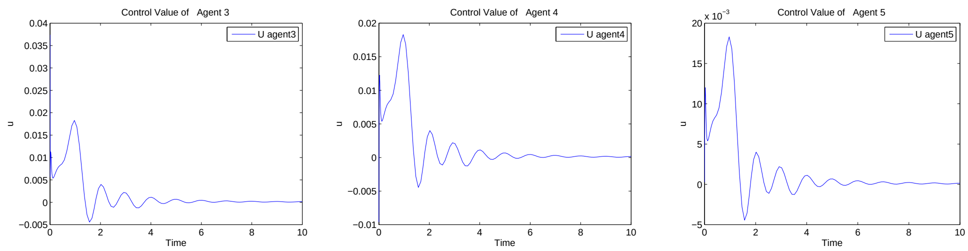
Figure 34: Velocity error of all the agents in 10 seconds.

As shown in Figures 35 and 36 the fuzzy sliding mode controller based on a separating plane , which is proposed in this paper , does not exhibit the chattering phenomenon.



(a) **(b)**

Figure 35: SMC based on a separating plane for agents 1 and 2.



(a) **(b)** **(c)**

Figure 36: SMC based on a separating plane for agents 3,4 and 5.

5 Conclusion

This paper conducted a study on study a fuzzy sliding mode controller (FSMC) that is based on a separating plane for a specific class of nonlinear multi-agent systems (MASs). The purpose of this study was to demonstrate the advantages of the proposed FSMC, and to do so, we presented three numerical examples. Our results showed that the proposed FSMC achieved a faster convergence to reach a consensus compared to other SMCs, such as the one based on an equivalent control. As a potential future direction for this research, it would be interesting to further investigate the application of the proposed FSMC to leaderless consensus problems.

Declarations

Availability of supporting data

All data generated or analyzed during this study are included in this published paper.

Funding

No funds, grants, or other support were received for conducting this study.

Competing interests

The authors have no competing interests to declare that are relevant to the content of this paper.

Authors' contributions

The main manuscript text is written collectively by the authors.

References

[1] Antonelli, G. (2013). "Interconnected dynamic systems: An overview on distributed control", IEEE, Control Systems Magazine, 33(1), 76-88.

[2] Boyd, S., Vandenberghe, L. (2004). "Convex optimization", Cambridge University Press, USA.

[3] Cao, Y., Yu, W., Ren, W., Chen, G. (2013). "An overview of recent progress in the study of distributed multi-agent coordination", IEEE, Transactions on Industrial Informatics, 9(1), 427-438.

[4] Carli, R., Bullo, F., Zampieri, S. (2010). "Quantized average consensus via dynamic coding/decoding schemes", International Journal of Robust and Nonlinear Control, 20(2), 156-175.

[5] Grocholsky, B., Keller, J., Kumar, V., Pappas, G. (2006). "Cooperative air and ground surveillance", IEEE, Robotics Automation Magazine, 13(3), 16-25.

[6] Jadbabaie, A., Lin J., Morse, A.S. (2003). "Coordination of groups of mobile autonomous agents using nearest neighbour rules", IEEE, Transactions on Automatic Control , 48(6), 988-1001.

[7] Jeon, I., Lee, J., Tahk, M. (2010). "Homing guidance law for cooperative attack of multiple missiles", Journal of Guidance Control and Dynamics, 33(1), 275-280.

[8] Jiang, F., Wang, L. (2010). "Consensus seeking of high-order dynamic multi-agent systems with fixed and switching topologies", International Journal of Control, 83(2), 404-420.

[9] Kennedy, J., Eberhart, R. (1995). "Particle swarm optimization", In Proceedings of ICNN'95 International Conference on Neural Networks, 4, 1942-1948.

[10] Larson, J., Kammer, C., Liang, K., Johansson, K.H. (2013). "Coordinated route optimization for heavy-duty vehicle platoons", IEEE, In 16th International IEEE Conference on Intelligent Transportation Systems, 1196-1202.

[11] Li, Z., Duan, Z., Chen, G. (2011). "Dynamic consensus of linear multi-agent systems", IET, Control Theory & Applications, 5, 19-28.

[12] Li, T., Fu, M., Xie, L., Zhang, J. (2011). "Distributed consensus with limited communication data rate", IEEE, Transactions on Automatic Control, 56(2), 279-292.

[13] Li, K., Zheng, B., Park, J.H. (2018). "Event based robust consensus for multi-agent systems via sliding mode control", 2018 Chinese Control And Decision Conference, 4483-4488.

[14] Li, Z., Duan, Z., Chen, G., Huang, L. (2011). "Consensus of multi-agent systems and synchronization of complex networks: A unified viewpoint", IEEE, Transactions on Circuits and Systems I: Regular Papers, 57(1), 213-224.

[15] Lin, P., Jia, Y. (2009). "Consensus of second-order discrete-time multi-agent systems with nonuniform time-delays and dynamically changing topologies", Automatica, 45(9), 2154-2158.

[16] Liu, X., Du, C., Liu, H., Lu, P. (2018). "Distributed event-triggered consensus control with fully continuous communication free for general linear multi-agent systems under directed graph", International Journal of Robust and Nonlinear Control, 28(1), 132-143.

[17] Ma, C., Zhang, J. (2010). "Necessary and sufficient conditions for consensusability of linear multi-agent systems", IEEE, Transactions on Automatic Control, 55(5), 1263-1268.

[18] Michael, N., Fink, J., Kumar, V. (2011). "Cooperative manipulation and transportation with aerial robots", Autonomous Robots, 30(1), 73-86.

[19] Murray, R.M., Sastry, S.S., Li, Z. (1994). "A mathematical introduction to robotic manipulation", 1st edition, RC Press, Inc., USA.

[20] Oh, R., Shi, Y., Woong, J. (2021). "A hybrid newton-raphson and particle swarm optimization method for target motion analysis by batch processing", Sensors, 21(6), 2033 .

[21] OlfatiSaber, R., Fax, J.A., Murray, R.M. (2007). "Consensus and cooperation in networked multi-agent systems", Proceedings of the IEEE, 95(1), 215-233.

[22] OlfatiSaber, R., Murray, R.M. (2004). "Consensus problems in networks of agents with switching topology and time-delays", IEEE, Transactions on Automatic Control, 49(9), 1520-1533.

[23] Ren, W. (2008). "On consensus algorithms for double-integrator dynamics", IEEE, Transactions on Automatic Control, 53(6), 1503-1509.

[24] Ren, W., Atkins, E.M. (2007). "Distributed multi-vehicle coordinated control via local information exchange", Journal of Robust and Nonlinear Control, 17(10-11), 1002-1033.

[25] Ren, W., Beard, R.W. (2005). "Consensus seeking in multi-agent systems under dynamically changing interaction topologies", IEEE, Transactions on Automatic Control, 50(5), 655-661.

[26] Ren, W., Beard, R.W. (2007). "Distributed Consensus in Multi-Vehicle Cooperative Control: Theory and Applications", 1st edition, Springer.

[27] Ren, W., Beard, R.W., Atkins, E.M. (2007). "Information consensus in multi vehicle cooperative control", IEEE, Control Systems Magazine, 27(2), 71-82.

[28] Ren W., Moore K., Chen Y. (2007). "High-order and model reference consensus algorithms in cooperative control of multi-vehicle systems", Journal of Dynamic Systems Measurement and Control-transactions of The Asme, 129(09), 678-688.

[29] Seo J. H., Shim H., Back J. (2009). "Consensus of high-order linear systems using dynamic output feedback compensator: Low gain approach", Elsevier, Automatica, 45(11), 2659-2664.

[30] Song Q., Cao J., Yu W. (2010). “Second-order leader-following consensus of nonlinear multi-agent systems via pinning control”, IEEE, Systems & Control Letters, 59(9), 553-562.

[31] Tuna S. E. (2009). “Conditions for synchronizability in arrays of coupled linear systems”, IEEE, Transactions on Automatic Control, 54(10), 2416-2420.

[32] Vicsek T. S., Czirók A., Eshel B. J., Inon C., Ofer S. (1995). “Novel type of phase transition in a system of self-driven particles”, Physical Review Letters, 75(6), 1226-1229.

[33] Wand D., Zhou Q., Zhu W. (2018). “Adaptive event-based consensus of multi-agent systems with general linear dynamics”, Springer, Journal of Systems Science and Complexity, 31(1), 120-129.

[34] Ye D., Chen M., Yang H. (2019). “Distributed adaptive event-triggered fault-tolerant consensus of multi agent systems with general linear dynamics”, IEEE, Transactions on Cybernetics, 49(3), 757-767.

[35] You K., Xie L. (2011). “Network topology and communication data rate for consensusability of discrete-time multi-agent systems”, IEEE, Transactions on Automatic Control, 56(10), 2262-2275.

[36] Yu W., Chen G., Cao M., Kurths J. (2010). “Second-order consensus for multiagent systems with directed topologies and nonlinear dynamics”, IEEE, Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 40(3), 881-891.

[37] Zhang H., Lewis F. L., Das A. (2011). “Optimal design for synchronization of cooperative systems: State feedback, observer and output feedback”, IEEE, Transactions on Automatic Control, 56(8), 1948-1952.

[38] Zhang Q., Wei X., Xu J. (2005). “Delay-dependent exponential stability of cellular neural networks with time-varying delays”, Chaos, Solitons & Fractals, 23(4), 1363-1369.

[39] Zhang Z., Zhang Z., Li L. (2017). “Dynamical sliding-mode control of leader-following multi-agent systems”, IEEE, 43rd Annual Conference of the IEEE Industrial Electronics Society, 6273-6278.

[40] Zhu W., Cheng D. (2010). “Leader-following consensus of second-order agents with multiple time varying delays”, Elsevier, Automatica, 1994-1999.