Research Article

Design of Connectivity Preserving Flocking Using Control Lyapunov Function

Bayu Erfianto,^{1,2} Riyanto T. Bambang,² Hilwadi Hindersah,² and Intan Muchtadi-Alamsyah³

¹*School of Computing, Telkom University, Bandung, Indonesia*

²*School of Electrical Engineering & Informatics, Institut Teknologi Bandung, Bandung, Indonesia*

³*Faculty of Mathematical Sciences, Institut Teknologi Bandung, Bandung, Indonesia*

Correspondence should be addressed to Bayu Erfianto; bayu.erfianto@gmail.com

Received 19 June 2016; Revised 28 August 2016; Accepted 14 September 2016

Academic Editor: Shahram Payandeh

Copyright © 2016 Bayu Erfianto et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper investigates cooperative flocking control design with connectivity preserving mechanism. During flocking, interagent distance is measured to determine communication topology of the flocks. Then, cooperative flocking motion is built based on cooperative artificial potential field with connectivity preserving mechanism to achieve the common flocking objective. The flocking control input is then obtained by deriving cooperative artificial potential field using control Lyapunov function. As a result, we prove that our flocking protocol establishes group stabilization and the communication topology of multiagent flocking is always connected.

1. Introduction

Flocking in nature is a coordinated movement of group of living entities. Recently, flocking behavior has been adopted in a multiagent system to resemble collective movement of living system, for example, collective movement of group of multi-robot system and autonomous vehicles. Those multiple agents together form a cooperative system to solve complex problems in distributed manner that is impossible to be carried out by a single agent.

With respect to the literature of multiagent system, many theories denote that flocking model is built using three basic heuristic rules, that is, cohesion, separation, and velocity alignment, which is initiated for the first time by [1]. These three rules describe how an individual agent moves according to position and velocity of the nearest agents in flocks. Flocks always move together with velocity converging to the same value and guaranteeing no collision among agents.

In multiagent robot, common flocking rule can further be described technically as centroid attraction, collision avoidance, and velocity alignment, respectively. Centroid attraction rule tries to pull the agent to stay close to the

nearby agents in the flocks if the agents are moving away. Meanwhile, collision avoidance rule forces the agent to avoid collision with nearby agents in the flock, while velocity synchronization rule will drive the agents to synchronize their velocities while moving in the flock. The most notable flocking controllers are based on the works of Tanner et al. [2, 3] and Olfati-Saber [4]. Olfati-Saber [4] does not use flock centering rule to build flocking formation; instead of that he defines lattice formation to keep interagent distance.

The synchronization of movement in flocking can be carried out through local or global communication, with explicit communication and without communication (stigmergic) as summarized in [5]. Erfianto et al. [6] have extended Reynold flocking protocol with explicit communication using broadcast gossip protocol, where the explicit communication of flocking is simulated in Netlogo multiagent simulator. By equipping mobile robot agent with communication capability, it means that flocking protocol is then restricted by communication capability. Previous research works on flocking algorithm are mostly based on the assumption that the network topology is always connected without any mechanism to achieve this circumstance.

Network communication is mainly used to exchange information state between an agent and the nearby agents. Regarding this matter, the problem is that the initially connected network can not guarantee connectivity over the time during flock movement. Once disconnected, flock movement can be separated and that causes collective flocking goal not being achieved. Thus, it is important to prevent separation of the flock into multiple separate groups when the network is being disconnected. Regarding this issue, many kinds of research have been conducted to design flocking control input (flocking protocol) to ensure that the flocks achieve the collective goal while preserving connectedness during the movement. Connectivity preserving flocking of multiagent mobile robot then becomes a new trending area, and various flocking protocols have been proposed including both centralized and decentralized approaches, for example, presented in [7–9].

Since connectivity of agents in flocking formation depends on its communication range, then the communication topology among agents can be connected or disconnected affected by the movement of an agent in flocks. The changes of signal power may also affect to the actual interagent distance since the distance is correspondingly perceived from measured signal power. Regarding this issue, a connectivity preserving mechanism is required to maintain connectivity during the flock movement. Fang et al. in [10] further divide connectivity preservation into two main categories, that is, conservative connectivity preservation and flexible connectivity preservation. In conservative connectivity preservation, the approach is to preserve all existing links; thus the network topology is strictly fixed, as presented in [7, 11, 12]. In flexible connectivity preservation approaches, the underlying communication topology is allowed to switch into different topologies, which means that edges are allowed to be added or removed from a node as long as the global graph topology is always connected. It is done by maintaining the spanning tree of the underlying network. Regarding these two approaches, therefore, the challenge is to design cooperative control law where it depends on the assumption that the underlying communication topology is connected over the time during the whole flocking evolution.

Formation of flocking is commonly used in the various practical implementation of flocking protocol, such as in coverage control using multiagent mobile robot, a formation of multiple UAV, and cooperative mobile sensor networks, as well as cooperative mobile robot, for a surveillance system. Therefore, the design of formation flocking needs to consider navigation capability along with separation, cohesion, velocity alignment, and connectivity of a multiagent network.

In this paper, our work is to preserve network connectivity by using an artificial potential function, where flocking protocol can also perform collision avoidance and obstacle avoidance function. The artificial potential function is then considered as control Lyapunov function candidate to be derived as continuous flocking control input. We also define the algebraic connectivity gradient controller and blend this gradient controller into flocking control input. This is the major difference of our work from [7, 9, 13, 14]. As a result, we conclude that, given the initially connected network, our

flocking protocol can make all agents converge to a common velocity, collisions between agents, as well as between agent and obstacle, can be avoided, and the connectivity of underlying network can be preserved; thus the desired stable flocking behavior can be achieved.

This paper is organized as follows. Section 1 is the introduction of this paper. Section 2 of this paper is about problem definition to design connectivity preserving flocking. Section 3 discusses the design of artificial potential function that will be involved in control Lyapunov function. Flocking controller design is presented in Section 4, while the simulation to verify the controller is presented in Section 5.

2. Problem Formulation and Assumption

Let us consider a group of points of mobile robots that consists of N robot agents moving in 2D space. The dynamic of mobile robot agent is modeled as

$$\begin{aligned}\dot{q}_i &= p_i, \\ \dot{p}_i &= u_i,\end{aligned}\tag{1}$$

where $q = [q_1^T, q_2^T, \dots, q_n^T]^T \in \mathbb{R}^2$ denotes position vector, where $q_{ij} = \|q_i - q_j\|$ is the Euclidean distance between two agents, which is clearly $q_{ij} \geq 0$. Let $p_i \in \mathbb{R}^2$ be the velocity vector and $u_i \in \mathbb{R}^2$ be the control input of agent i , respectively.

Assumption 1. We assume that each agent obtains its position from sensors. Each agent is also equipped with a wireless communication network; therefore information state can be exchanged within the neighboring agents via a communication network.

Regarding Assumption 1, interaction among agents can further be modeled by bidirectional communication network, that is, undirected graph topology.

Definition 2 (flocking topology). Topology of a flock of robots is a dynamic undirected topology, denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, that consists of

- (i) node or vertices $\mathcal{V} = \{1, \dots, n\}$;
- (ii) set of edges or time varying communication links $\mathcal{E} = \{i, j \in \mathcal{V} \times \mathcal{V} \mid q_{ij} \leq R\}$ in which each undirected edge represents the communication between a pair of agents limited by communication range R ;
- (iii) set of neighbors $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$ with symmetric property $i \in \mathcal{N}_j \leftrightarrow j \in \mathcal{N}_i$.

Without loss of generality, each agent is strictly not self-communicating agent $(i, j) \notin \mathcal{E}$. Regarding Definition 2, interagent communication can then be constructed by adjacency matrix $A \in \mathbb{R}^{N \times N}$, with element

$$a_{ij} = \begin{cases} 1 & \text{if } q_{ij} \leq R \\ 0 & \text{otherwise,} \end{cases}\tag{2}$$

where q_{ij} is the Euclidean distance between two agents, and each agent is supposed to have the same communication radius R . Some fundamentals of graph theory used in the paper, including algebraic graph theory, can be found in the main reference [15].

From the adjacency matrix A , the degree matrix is defined as $D = \text{diag}(a_{ij}) = \sum_{j=1}^N a_{ij}$. Laplacian matrix of $\mathcal{G}$ can be further obtained by subtracting adjacency matrix from degree matrix, such that $L = D - A$, where Laplacian matrix L is positive semidefinite. Let the eigenvalues of L be $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$ which is nondecreasing order with corresponding eigenvectors $\{v_1, v_2, \dots, v_n\}$. We denote that the remarkable properties of Laplacian matrix are as follows.

Lemma 3 (see [15]). *Given the undirected graph $\mathcal{G}$*

- (i) $L(\mathcal{G})$ is always symmetric and positive semidefinite and satisfies sum of square property

$$x^T L x = \frac{1}{2} \sum_{i=1}^N a_{ij} (x_i - x_j)^2, \quad (3)$$

where $x_i \in \mathbb{R}^N$;

- (ii) $\lambda_1 = 0$; thus $\lambda_2 > 0$ if graph $\mathcal{G}(t)$ is connected; then λ_2 is referred to algebraic connectivity of $\mathcal{G}(t)$;
- (iii) $L(\mathcal{G})$ has a single zero eigenvalue and the corresponding eigenvector is $\mathbf{1}_N$.

This paper will focus on the design of flocking protocol that can preserve connectivity in terms of algebraic graph representation.

We define the stable flocking protocol as the necessary condition to preserve connectivity of the network all the time.

Definition 4 (stable flocking protocol). Group of robot agents is said to flock when all agents approach the same velocity, with no collisions among them, such that

$$\begin{aligned} \|q_i - q_j\| &\leq R, \\ p_i &= p_j, \\ \lambda_2 &> 0, \end{aligned} \quad (4)$$

where λ_2 is the second smallest eigenvalue of Laplacian matrix L .

The connectivity maintenance of flocking topology is approximately determined by the distance between agents in the flock. The movement of agents that might cause the connectivity can be unavailable during flocking motion. Therefore, our problem is dealing with designing a cooperative flocking protocol using only local information to drive all robot agents to achieve the same velocity, avoiding collision with other agents in flock, avoiding static obstacle, driving the agents to the goal position, and guaranteeing that the network is always connected during the flocking, indicated by $\lambda_2(L) > 0$.

3. Potential Function Design

This section introduces the design of artificial potential functions for the purpose of target attraction, obstacle avoidance, and interagent attraction and repulsion. Since first used in [16], the artificial potential function has been commonly used to solve the problems of motion planning in mobile robotic applications. Theoretically, the artificial potential function generates repulsive fields close to obstacles and attractive fields close target. In artificial potential fields, a noncollided motion is determined by how much the robot is attracted by other agents and repelled by either the obstacle or other agents. We add centroid function and attract-repel function to avoid local minima caused by goal attraction and obstacle function. The potential function method has several advantages, and the most important one is being easy to implement.

3.1. Environmental Potential Field. Consider the multiagent robots move in configuration workspace, which is defined as follows.

Definition 5 (workspace). Suppose a point of robot moves in the workspace $\mathcal{W} \triangleq \{q \in \mathbb{R}^2 \mid q_{ij} \leq s^2\}$; that is, the Euclidean distance between two points does not exceed the area of s^2 .

We define the free space, where there is no obstacle within the area.

Definition 6 (free space). Free space $\mathcal{F}$ is defined as $\mathcal{F} \triangleq \mathcal{W} \setminus \bigcup_{k=1}^m O_k$, where a robot agent could move safely without colliding with any obstacles.

Given the position of robot agents in stack vector $q = [q_1^T, \dots, q_N^T]^T \in \mathbb{R}^2$, we assume all information of the environment in the $\mathcal{W}$ is known by each robot agent. A target or goal position of agent i is defined as $q_t \in \mathcal{F}$, which is in the free area of configuration workspace. To attract the robot agent to reach the desired goal position, the attraction-to-target potential function $\mathcal{T}$ is defined as follows.

Definition 7 (target attraction function). $\mathcal{T}_i : \mathcal{F} \rightarrow [0, \infty)$ is a metric that measures how close a robot agent is to the desired goal position. Target or goal attraction function is defined as

$$\mathcal{T}_i = \frac{1}{2} \|q_i - q_t\|^2. \quad (5)$$

Let $O_k \in \mathbb{R}^2$ be k th obstacle in the workspace and $O_{i,k} \in \mathbb{R}^2$ be the k th obstacle that has to be avoided by agent i .

Definition 8 (obstacle). O_k is a k th disc obstacle that exists in workspace $\mathcal{W}$ with center $q_o \in \mathbb{R}^2$ and radius r_k :

$$O_k = \{q \in \mathbb{R}^2 \mid \|q - q_o\| \leq r_k\} \quad (6)$$

such that the distance of any point to the center of obstacle does not exceed r_k .

Using the definitions above, all robot agents have to avoid obstacles that block the robot route toward the desired

FIGURE 1: Example of environment model used in this paper. Goal position is represented by the lowest potential value; two hills represent different size of obstacles.

position. Therefore, each robot has to execute an obstacle avoidance function that is constructed by measuring the distance between the robot agent and the obstacle center $q_o \in \mathcal{W}$. The obstacle avoidance function computed by i th agent is defined as follows.

Definition 9 (obstacle avoidance function). Robot agent i could avoid k th obstacle with the following potential function:

$$\mathcal{O}_{i,k} = \exp\left(-\frac{(1/2)\|q_i - q_o\|^2}{r_k}\right)r_k, \quad (7)$$

where $q_o \in \mathbb{R}^2$ is the center of obstacle and r_k is radius of obstacle k , such that robot agent i could avoid obstacle O_k if none of them intersect each other within $\|q_i - q_o\| > r_k$.

Using the definitions above, the environmental potential field is composed of goal attractiveness and obstacle repulsiveness, and the result is depicted in Figure 1. This figure illustrates two different sizes of obstacles indicated by two hills with high potential value, while the lowest potential, the darkest valley at the center of the basin, illustrates the desired position.

3.2. Interagent Potential Function. We assume that the sensing capability of an agent equals its communication radius. Suppose there exists pair interaction between robot agents i and j via wireless communication network. The interaction between paired agents is limited by communication range, where the coverage of robot agent i resembles a disk with a certain radius R .

Definition 10 (agent disc radius). Let $\mathcal{B}(q_i, R)$ be a disk with coverage radius R , and $q_i = (x_i, y_i) \in \mathbb{R}^2$ is position of agent i at (x, y) plane:

- (i) Open disk radius is an area covered by $\mathcal{B}(q_i, R)$ if $\{q_j \in \mathbb{R}^2 \mid \|q_i - q_j\| < R\}$.
- (ii) Close disk radius is an area covered by $\mathcal{B}(q_i, R)$ if $\{q_j \in \mathbb{R}^2 \mid \|q_i - q_j\| \leq R\}$.

FIGURE 2: Illustration of an agent disk radius and centroid attraction among the flockmates.

3.3. Attraction to Centroid. The idea of flock centroid is initiated in [1]. Action to attract flock member to the center of a flock, or flock centering, makes the agents move in a direction toward the centroid. By means of flock centering, the flocking algorithm provides a mechanism to make sure that agents are attracted to each other and not moving away. This concept is further known as a part of the well-known one of three heuristic flocking rules initiated by Reynolds [1]. Illustration of agent's disk radius and the center of the flock can be seen in Figure 2. This figure also explains how the interaction among agents in the flock occurs.

The long-range attraction occurs between agents in the flocks. To make sure that robot agents in the flocks are attracted to each other, as well as attracted by the center of the flock, an attractive function is constructed by measuring the distance between individual agents and the center of the flocks. The flock centering attraction function is simply given by

$$\mathcal{C}_i(q_i) = \sum_{i=1}^N \left\| q_i - \frac{1}{N} \sum_{i=1}^N q_i \right\|^2. \quad (8)$$

3.4. Interagent Attraction Repulsion. This paper uses the definition of closed disk coverage, where two interagent distances may occur within $\|q_i - q_j\| \leq R$. This condition is further coded into communication edge $(i, j) \in \mathcal{E}$. Based on this interaction, the neighborhood of an agent i is determined by $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$. Using the definition of $\mathcal{N}_i$, the interagent collision avoidance function is further defined.

Definition 11 (interagent collision avoidance). Collision avoidance function is used to keep $\|q_i - q_j\| \leq R$ such that communication can be preserved and no agents collided during the flock movement.

Interagent collision avoidance is implemented by attract-repel function. The long-range attraction occurs between two

agents in the flocks. It is to make sure that the agents in the flock are attracted by the following function:

$$\Psi_{ij}(q_{ij}) = \sum_{j \in \mathcal{N}_i} \frac{c_a}{2} \|q_i - q_j\|^2, \quad (9)$$

where c_a is attraction constant. Properties of attraction function can be characterized as follows:

- (i) Attractive potential function can be viewed as potential energy between agents in the flock.
- (ii) Let $\Psi_{ij}(q_{ij})$ be the attraction function; then
 - (a) $\Psi_{ij}(q_{ij}) \rightarrow 0$, as agent $\|q_i - q_j\| \rightarrow \infty$; that is, j moves away from agent i .
 - (b) $\Psi_{ij}(q_{ij})$ is differentiable along $q_{ij} > 0$ such that $\nabla_{q_i} \Psi_{ij}(q_{ij}) > 0$.

Interagent collision avoidance is implemented by short range repulsive function, which is formulated from the distance between agents. The objective of this function is to repel neighboring agents if they are too close; therefore collision can be avoided. The short range repulsive potential function is defined as

$$\Phi(q_{ij}) = \sum_{j \in \mathcal{N}_i} c_r \exp\left(-\frac{(1/2)\|q_i - q_j\|^2}{R}\right) R, \quad (10)$$

where c_r is repulsion constant. The properties of repulsive function can be characterized as follows:

- (i) Repulsive potential function can be viewed as potential energy between agents in the flock within the range $[0, R]$ such that no collision occurs between agents $j \in \mathcal{N}_i$.
- (ii) Let $\Phi_{ij}(q_{ij})$ be the repulsive function, such that
 - (a) $\Phi_{ij}(q_{ij}) \rightarrow \infty$, as $\|q_i - q_j\| \rightarrow 0$;
 - (b) $\Phi_{ij}(q_{ij})$ is differentiable along $q_{ij} > 0$, such that $\nabla_i \Phi_{ij}(q_{ij}) > 0$.

We combine attraction and repulsion potential function to generate attract-repel function as

$$\begin{aligned} \mathcal{R}_{ij}(q_{ij}) &= \Phi_{ij}(q_{ij}) + \Psi_{ij}(q_{ij}) \\ &= \sum_{j \in \mathcal{N}_i} \frac{c_a}{2} \|q_i - q_j\|^2 \\ &\quad + \sum_{j \in \mathcal{N}_i} c_r \exp\left(-\frac{(1/2)\|q_i - q_j\|^2}{R}\right) R. \end{aligned} \quad (11)$$

The attractive-repulsive potential function may guarantee that agent j never moves away from $\mathcal{N}_i$. The plot of attract-repel function is depicted in Figure 3.

FIGURE 3: Interagent attract-repel potential function with various repulsive constant c_r , where $c_a = 1$ and $R = 3$.

4. Flocking Controller Design

This section describes the design of flocking controller by utilizing control Lyapunov function. We use the direct method of Lyapunov to obtain the flocking control input which is derived from overall artificial potential function that has been defined in the previous section. The reader may refer to [17–19] for the detail of control Lyapunov function for multiagent system.

4.1. Control Lyapunov Function. Regarding [1, 4, 7], flocking control law, which is also known as flocking protocol, is composed of negative gradient of potential force and consensus velocity of flock members. The force is obtained from the gradient of potential fields. Generally, the objective of flocking control is to steer the flock such that velocity between agents i and j is approximately the same.

We extend the control law of formation control problem in [14] into flocking problem. First of all, consider a multiagent system dynamics

$$\begin{aligned} \dot{q}_i &= p_i, \\ \dot{p}_i &= u_i, \end{aligned} \quad (12)$$

where $q_i \in \mathbb{R}^2$ is position of agent in two-dimensional space, p_i is agent velocity, and u_i in control input to be designed. Our intention is to use control Lyapunov function to obtain control input u_i , which is derived from the potential function. We recall the sum of potential functions from (5), (7), (8), and (11),

$$\begin{aligned} \mathcal{P}_i(q_i) &= \sum_{i=1}^N \left(\mathcal{C}_i(q_i) + \sum_{j \in \mathcal{N}_i} \mathcal{R}_{ij}(q_{ij}) + \mathcal{T}_i(q_i) + \mathcal{O}_{i,k}(q_i) \right). \end{aligned} \quad (13)$$

Definition 12. A control Lyapunov function is continuously positive-definite function; that is, $V(x)$ is positive except for $x = 0$, such that

$$\dot{V}(x, u) = \nabla V(x) f(x, u) < 0. \quad (14)$$

Regarding Definition 12, the candidate of Lyapunov function is then defined by combining all potential energy obtained from potential function, with kinetic energy as the difference between velocity of agent i and agent j ,

$$V_i = \mathcal{P}_i(q_i) + \frac{1}{2} \sum_{j \in \mathcal{N}_i} (p_i - p_j)^2. \quad (15)$$

Candidate of Lyapunov function for all agents can then be written as

$$\begin{aligned} V = \sum_{i=1}^N V_i = \sum_{i=1}^N & \left(\mathcal{E}_i(q_i) \right. \\ & + \sum_{j \in \mathcal{N}_i} \mathcal{R}_i(q_i) + \mathcal{T}_i(q_i) + \mathcal{O}_{i,k}(q_i) + \frac{1}{2} \\ & \cdot \sum_{j \in \mathcal{N}_i} (p_i - p_j)^2 \left. \right). \end{aligned} \quad (16)$$

To extract the control input from candidate of Lyapunov function, we differentiate all potential function components in V , such that flocking control input for system (12) is obtained from $\dot{V}(q_i)$. We consider the assumption described in previous section to generate the following theorem.

Theorem 13. *The N holonomic agents use dynamics in system (12). The stable flocking control input u_i can be obtained from the gradient of (16). The flocking protocol is stable such that each agent moves in the flock, the obstacle can be avoided, and there is no interagent collision, and finally agents stop at a defined goal position.*

To prove the above theorem, we consider (16). It is obvious that $V_i(0) = 0$, and V_i is positive-definite function and continuous when

$$\begin{aligned} \sum_{i=1}^N \mathcal{E}_i(q) &> 0, \\ \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \mathcal{R}_{ij}(q) &> 0, \\ q_i &\in \mathbb{R}^2. \end{aligned} \quad (17)$$

Since the force is negative gradient along the trajectories of q_i , then we can write

$$\dot{V} = \sum_{i=1}^N \frac{\partial V_i}{\partial q_i} \frac{\partial q_i}{\partial t} = \sum_{i=1}^N \frac{\partial V_i}{\partial q_i} \cdot \dot{q}_i = \sum_{i=1}^N \frac{\partial V_i}{\partial q_i} p_i. \quad (18)$$

The velocity is the form of negative force that equals

$$p_i = - \sum_{i=1}^N \frac{\partial V_i}{\partial q_i}. \quad (19)$$

We then check the definiteness of Lyapunov function, such that

$$\dot{V} = - \sum_{i=1}^N \left(\frac{\partial V_i}{\partial q_i} \cdot \frac{\partial V}{\partial q_i} \right) = - \sum_{i=1}^N p_i^T p_i \leq 0. \quad (20)$$

It is clear that since (20) is negative semidefinite, then control input u_i is Lyapunov stable. Therefore, the flocking control input u_i can then be obtained from the gradient of Lyapunov function $\dot{V}(q_i, p_i)$ as follows:

$$\begin{aligned} u_i = -\nabla_q \mathcal{E}_i(q) - \sum_{j \in \mathcal{N}_i} \nabla_q \mathcal{R}_{ij}(q) - \nabla_q \mathcal{T}_i(q) \\ - \nabla_q \mathcal{O}_{i,k}(q) + \sum_{j \in \mathcal{N}_i} (p_i - p_j) \\ = - \left(q_i - \sum_{i=1}^N \bar{q}_i \right) + \exp^{-(q_i - q_o)/r_s} (q_i - q_o) \\ - (q_i - q_i) + (c_a - c_r \exp^{-(q_i - q_j)/R}) (q_i - q_j) \\ + \sum_{j \in \mathcal{N}_i} (p_i - p_j). \end{aligned} \quad (21)$$

The notation $\bar{q} = (1/N) \sum_{i=1}^N q_i$ is the center of the flock. Thus, we can substitute flocking control input in system (12) with (21). It can be observed that the last term of control input is velocity consensus.

4.2. Convergence of Velocity. In this subsection, we present the analysis of velocity convergence which is the necessary condition of stable flocking controller as in Definition 4. We begin with the stationary of the center of the flocks, which is to attract all flock members during the movement. Now let us consider the following lemma.

Lemma 14. *Using system (12) and control input in (21), the centroid of the flocks is stationary $\forall t > 0$.*

Proof. Let $\bar{q} = (1/N) \sum_{i=1}^N q_i$ be the center of the flocks. The time derivative of $\bar{q}$ can further be defined as

$$\dot{\bar{q}} = \frac{1}{N} \sum_{i=1}^N \dot{q}_i. \quad (22)$$

Let us consider the system $u_i = \dot{q}_i$ and let $\dot{q}_i = \nabla_{q_i} \mathcal{R}_i(q) = -(c_a - c_r \exp^{-(q_i - q_j)/R})(q_i - q_j)$ be the attract-repel force. Let $f(-a) = -(c_r \exp^{-(q_i - q_j)/R})(q_i - q_j)$; then $f(-a)$ is odd function such that $f(-a) = -f(a)$. Due to the symmetry of odd function, (22) can be written as

$$\dot{\bar{q}} = \frac{1}{N} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (f(-a) + f(a)) = 0. \quad (23)$$

Let the invariant set of equilibrium (or stationary) points be $\Omega_e = \{q \mid \dot{q} = 0\}$. It implies $\dot{q}_i = \dot{\bar{q}}$, which means that all individual agents' velocity equals the velocity of center of the flocks. $\square$

We begin with following theorem.

Theorem 15. *Robot agent that uses control input (21) eventually can reach the same velocity during flocking and collision can be avoided and achieve the desired goal.*

Proof. To prove this theorem, we rewrite the sum of potential function $\mathcal{P}_i$ defined in (13):

$$\begin{aligned} \mathcal{P}_i(q_i) &= \sum_{i=1}^N \left(\mathcal{E}_i(q_i) + \sum_{j \in \mathcal{N}_i} \mathcal{R}_{ij}(q_i) + \mathcal{T}_i(q_i) + \mathcal{O}_i(q_i) \right). \end{aligned} \quad (24)$$

The candidate of Lyapunov function is then defined, which consists of artificial potential function and velocity of each agent.

$$V_i = \frac{1}{2} \left(\sum_{i=1}^N \mathcal{P}_i(q_i)^2 + p_i^2 \right). \quad (25)$$

The derivative of Lyapunov function along q_i is

$$\begin{aligned} \dot{V}_i &= \sum_{i=1}^N \dot{q}_i \left(\nabla_{q_i} \mathcal{E}_i(q) + \nabla_{q_i} \mathcal{T}_i(q) + \nabla_{q_i} \mathcal{O}_i(q) \right. \\ &\quad \left. + \sum_{j \in \mathcal{N}_i} \nabla_{q_i} \mathcal{R}_{ij}(q) \right) + \sum_{i=1}^N p_i^T \dot{p}_i. \end{aligned} \quad (26)$$

By replacing $\dot{p}$ in (26) with u_i in (21), we obtain

$$\begin{aligned} \dot{V}_i &= \sum_{i=1}^N \dot{q}_i \left(\nabla_{q_i} \mathcal{E}_i(q) + \nabla_{q_i} \mathcal{T}_i(q) + \nabla_{q_i} \mathcal{O}_i(q) \right. \\ &\quad \left. + \sum_{j \in \mathcal{N}_i} \nabla_{q_i} \mathcal{R}_{ij}(q) \right) + \sum_{i=1}^N p_i^T u_i = \sum_{i=1}^N p_i^T \left[\left(\nabla_{q_i} \mathcal{E}_i(q) \right. \right. \\ &\quad \left. \left. + \nabla_{q_i} \mathcal{T}_i(q) + \nabla_{q_i} \mathcal{O}_i(q) + \sum_{j \in \mathcal{N}_i} \nabla_{q_i} \mathcal{R}_{ij}(q) \right) \right. \\ &\quad \left. + \sum_{i=1}^N u_i \right] = \sum_{i=1}^N p_i^T \left[\left(\nabla_{q_i} \mathcal{E}_i(q) + \nabla_{q_i} \mathcal{T}_i(q) \right. \right. \\ &\quad \left. \left. + \nabla_{q_i} \mathcal{O}_i(q) + \sum_{j \in \mathcal{N}_i} \nabla_{q_i} \mathcal{R}_{ij}(q) \right) - \sum_{i=1}^N \nabla_{q_i} \mathcal{E}_i(q) \right. \\ &\quad \left. - \nabla_{q_i} \mathcal{T}_i(q) - \nabla_{q_i} \mathcal{O}_i(q) - \sum_{j \in \mathcal{N}_i} \nabla_{q_i} \mathcal{R}_{ij}(q) \right. \\ &\quad \left. - \sum_{j \in \mathcal{N}_i} (p_i - p_j) \right] = -p_i^T \sum_{j \in \mathcal{N}_i} (p_i - p_j). \end{aligned} \quad (27)$$

The neighborhood $j \in \mathcal{N}_i$ is represented by adjacency element a_{ij} ; then (27) can be written as

$$\dot{V} = -\sum_{i=1}^N \left(p_i^T \sum_{j \in \mathcal{N}_i} a_{ij} (p_i - p_j) \right) = -p^T (L \otimes I) p \leq 0, \quad (28)$$

where L is the Laplacian matrix and I is the identity matrix. Therefore it means that when $-p^T (L \otimes I) p = 0$, the agent velocity $p_1 = p_2 = \dots = p_N$, $i = 1, 2, \dots, N$. $\square$

4.3. Algebraic Connectivity Gradient Controller. In this section, we add the term of gradient of algebraic connectivity along q_i into u_i in (21). We redefine the adjacency matrix A as follows:

$$a_{ij} = \begin{cases} \exp^{-(1/2)\|q_i - q_j\|^2/R} R, & \|q_i - q_j\| \leq R \\ 0, & \text{otherwise.} \end{cases} \quad (29)$$

We can also write the second smallest eigenvalue of Laplacian matrix $\lambda_2 = v_2^T L v_2$, where v_2 is corresponding eigenvector of λ_2 . Due to the properties of Laplacian matrix, the connectivity is guaranteed if $\lambda_2(L) > 0$. Suppose we define $(0, 1] \in \epsilon$ to be the lower bound of λ_2 . It is clear that when λ_2 goes to ϵ , it means that the graph connectivity is weak. Regarding this condition, we consider adding the value of algebraic connectivity as the additional term in flocking protocol, which is to make sure that $\lambda_2 \geq \epsilon$.

Let us define the following theorem.

Theorem 16. *Suppose $\varphi(q_i) = \coth^2(\lambda_2(q_i) - \epsilon)$ is the short range function of λ_2 along q_i . Then $\varphi(q_i)$ is differentiable along q_{ij} . Therefore, its gradient exists along q_{ij} , and $\varphi(\lambda_2 - \epsilon)$ approaches a constant value as λ_2 increases.*

Proof. To proof this theorem, we follow [20]. We can also write

$$\inf_{v \in \mathbf{1}^\perp} \lambda_2 = v^T L v = \inf_{v \in \mathbf{1}^\perp} \sum_{i=1}^n \sum_{j \in \mathcal{N}_i} a_{ij} (v_i - v_j)^2, \quad (30)$$

where v_i is corresponding eigenvector of λ_2 . Suppose we want to define the gradient controller with algebraic connectivity function. The control input can be defined by taking the derivative of λ_2 along q_i as follows:

$$u_i^\lambda = \frac{\partial \varphi(\lambda_2(q_i) - \epsilon)}{\partial q_i} = \frac{\partial \varphi(\lambda_2(q_i) - \epsilon)}{\partial \lambda_2} \frac{\partial \lambda_2}{\partial q_i}. \quad (31)$$

The solution of first term is partial derivative of $\varphi(\lambda_2(q_i))$ on λ_2 ,

$$\frac{\partial \varphi(p)}{\partial \lambda_2} = \text{csch}^2(\lambda_2 - \epsilon). \quad (32)$$

The next step is to find the solution of second term of partial derivative of $\partial \lambda_2 / \partial q_i$ in (31). We use [20] to find $\partial \lambda_2 / \partial q_i$.

Lemma 6 (see [20]). *Given any positively weighted graph $\mathcal{G}$, λ_2 is a nondecreasing function of each weight a_{ij} .*

Second smallest eigenvalue can be written in the form of

$$\lambda_2 = \min_{x \perp \mathbf{1}} \frac{x^T L x}{x^T x} = \min_{v \perp \mathbf{1}} \frac{\sum_{(i,j) \in \mathcal{E}} a_{ij} (v_i - v_j)^2}{v^T v}. \quad (33)$$

As a consequence, therefore, the partial derivative of λ_2 can also be written as follows:

$$\frac{\partial \lambda_2}{\partial q_i} = v^T \frac{\partial L}{\partial q_i} v. \quad (34)$$

By replacing $x^T L x$ with $\sum_{(i,j) \in \mathcal{E}} a_{ij} (v_i - v_j)^2$ we obtain

$$v^T \frac{\partial L}{\partial q_i} v = \sum_{(i,j) \in \mathcal{E}} \frac{\partial a_{ij}}{\partial q_i} (v_i - v_j)^2. \quad (35)$$

Therefore, the gradient of $\varphi(\lambda_2 - \epsilon)$ exists along q_i , and this value asymptotically approaches a constant value as λ_2 increases. $\square$

We further define the gradient of algebraic connectivity as control input as follows:

$$u_i^\lambda = -\text{csch}^2(\lambda_2 - \epsilon) \sum_{j \in \mathcal{N}_i} a_{ij} (v_i - v_j)^2 (q_i - q_j). \quad (36)$$

We define the following nonnegative energy function as the candidate of Lyapunov function:

$$V = \coth(\lambda_2 - \epsilon). \quad (37)$$

With respect to [20], $\nabla_{q_i} V = \dot{q}_i = u_i$. Thus, the derivative of candidate of Lyapunov function is

$$\dot{V}(q) = \nabla_{q_i} V(q) \dot{q} = \dot{q}^T \dot{q} = -\sum_{i=1}^N p_i^T p_i. \quad (38)$$

Equation (36) can then be added into the previous flocking protocol as a new term in (21),

$$\begin{aligned} u_i = & -\nabla_{q_i} \mathcal{C}_i(q) - \nabla_{q_i} \mathcal{T}_i(q) + \nabla_{q_i} \mathcal{O}_i(q) \\ & + \sum_{j \in \mathcal{N}_i} \nabla_{q_i} \mathcal{R}_{ij}(q) - \sum_{j \in \mathcal{N}_i} (p_i - p_j) \\ & - \text{csch}^2(\lambda_2 - \epsilon) \\ & \cdot \sum_{j \in \mathcal{N}_i} \exp^{-((1/2)\|q_i - q_j\|^2/R)} R (v_i - v_j)^2 (q_i - q_j). \end{aligned} \quad (39)$$

To analyze the stability of flocking controller after u_i^λ addition, we take the derivative in (28) with respect to q_i and add it with the derivative of Lyapunov function in (38) with respect to q_i , such that

$$\begin{aligned} \dot{V}(q) = & -\sum_{i=1}^N \left(p_i^T \sum_{j \in \mathcal{N}_i} a_{ij} (p_i - p_j) \right) - \sum_{i=1}^N p_i^T p_i \\ = & -2 \sum_{i=1}^N (p_i^T (L \otimes I) p_i) = -2 (p^T (L \otimes I) p) \\ \leq & 0. \end{aligned} \quad (40)$$

```

initialization all variables and constants
while  $t < t_{\text{end}}$  do
  capture position  $q$ ;
  for  $i = 1$  to  $N$  do
    compute  $\nabla_{q_i} \mathcal{T}_i, \nabla_{q_i} \mathcal{O}_{i,k}$ ;
    if  $q_{ij} \leq R$  then
      build  $[a_{ij}]$ ;
      compute  $\sum_{i=1}^N (\nabla_{q_i} \mathcal{R}_i + \nabla_{q_i} \mathcal{C}_i)$ ;
    end
  end
  compute  $L, \lambda_2, p_i$ ;
  compute  $u_i$ 
  update  $q_i$ ;
  update  $u_i$ ;
end

```

ALGORITHM 1: Main algorithm.

It means that adding gradient of algebraic connectivity to the previous flocking control input remains guarantees the stability in the sense of Lyapunov, connectivity can be maintained, collision between agents as well as with obstacle can be avoided, and the agent can reach the common desired position.

5. Simulation of Multiagent Connectivity Preserving Flocking

This section demonstrates the simulation results of connectivity preserving flocking in multiagent system using the stable control input obtained from the previous section. The simulation verifies flocking protocol, which is derived using control Lyapunov function. The common algorithm to simulate multiagent robot with connectivity preserving flocking is depicted in Algorithm 1. For simulation purpose, the different flocking scenarios are also presented: workspace without obstacle, with single obstacle, and with multiple obstacles. In all scenarios, the point mass mobile robot agents move from the initial position to the goal position while avoiding each other.

All scenarios in simulation are conducted in Matlab. The initial position of each robot agent is generated randomly with initial connected topology at $X = 1$ to 5, $Y = 1$ to 5. The size of obstacles is varied from $r_k = 1$ to $r_k = 3$. For the purpose of simulation, we use the same communication radius $R = 5$. The simulation scenarios are determined as follows:

- (1) multiagent flocking without obstacle,
- (2) multiagent flocking with single obstacle,
- (3) multiagent flocking with multiple obstacles,
- (4) multiagent flocking with algebraic connectivity controller.

(1) *Multiagent Flocking without Obstacle.* In the first scenario, we present the result of simulation of connectivity preserving

FIGURE 4: Multiagent flocking trajectories without obstacle.

FIGURE 5: Multiagent velocity trajectory in no obstacle scenario.

flocking without obstacle in the configuration workspace. We simulate five points of robots that represent robot agent, randomly starting from initial condition around $XY = (0, 0)$ up to $(5, 5)$ in coordinate plane, and target position is $XY = (35, 35)$. We plot the trajectory of robot during simulation and the result is depicted in Figure 4.

From simulation, it can be seen that the multiagent graph is always connected during flocking evolution. The convergence of agent's velocity is determined by consensus term as described in control input in (21). In Figure 5 it can be seen that the velocity of agents converges to the same value when consensus is used in the connectivity preserving flocking protocol.

(2) *Multiagent Flocking with Single Obstacle.* We plot the trajectory of a robot during simulation as depicted in Figure 6. The multiagent graph is always connected during flocking evolution. In this simulation, we can also see that velocity of agents still converges to the same value, as depicted in Figure 7. A slightly different result is obtained when obstacle is present in the workspace. Agents can reach velocity convergence in the beginning. However, when detecting the obstacles, the velocity is a bit varying before it finally converges.

(3) *Multiagent Flocking with Multiple Obstacles.* In this scenario, multiple sizes of multiple obstacles are present in the

FIGURE 6: Multiagent robot trajectories with single obstacle.

FIGURE 7: Multiagent velocity trajectory in the scenario with obstacle.

workspace. We assume that three multiple sizes of obstacles represent a small, medium, and large size of obstacle. For this purpose, we use two different subscenarios: first is simulation with five robot agents, and the second one is with ten robot agents.

The robot trajectories during simulation of multiagent with multiple size obstacles can be seen in Figure 8. We also plot the velocity convergence of flock during simulation and the results are depicted in Figure 9. In Figure 10, ten robot agents are simulated to perform flocking and reach the goal position. This is to see how the flocking control input works with large number of agents. Ten agents can still move in the flock while avoiding three different sizes of obstacle and finally reach the desired position.

(4) *Multiagent Flocking with Algebraic Connectivity Controller.* In this scenario, we use (39) to preserve connectivity by using gradient of algebraic connectivity. During simulation, the values of algebraic connectivity, which indicates the measure of connectedness of multiagent network, are plotted and depicted in Figure 13. From the beginning of simulation, it can be seen that the value of algebraic connectivity remains > 0 , meaning that the flocking controller still guarantees

FIGURE 8: Multiagent robot trajectories with multiple size multiple obstacles.

FIGURE 9: Multiagent velocity convergence in multiple size multiple obstacles.

FIGURE 10: Multiagent robot trajectories of 10 agents with multiple obstacles.

FIGURE 11: Multiagent robot trajectories with single and multiple obstacles with algebraic connectivity controller.

connectivity while avoiding obstacle as illustrated by robot trajectories shown in Figure 11.

We also compare the velocity convergence where the gradient of λ_2 is added into the flocking control input, and the plot is depicted in Figure 12. By adding λ_2 gradient controller, velocity alignment in flocking protocol is smooth and faster compared to flocking controller without λ_2 gradient controller.

6. Conclusion

In this paper, the control Lyapunov function is successfully used to construct flocking control input with connectivity preservation in a group of networks of mobile robot agents.

FIGURE 12: Multiagent velocity with multiple obstacles scenario. (a) Without λ_2 gradient. (b) With λ_2 gradient.

FIGURE 13: Algebraic connectivity value during simulation of multi-agent velocity convergence in multiple size multiple obstacles.

We model the agents as a dynamic point mass, and the proposed control strategy involves potential force and velocity consensus. Several potential forces are designed to steer the neighbor agents to predetermined distance while avoiding collision among them such that connectedness of network topologies can be guaranteed. Under the assumption that the initial flocking topology is connected, the overall flocking topology is preserved to be connected while obstacle and collision avoidance are also guaranteed.

Simulation results verify the new proposed control input with gradient of algebraic connectivity. From the multiagent flocking point of view, it implies that the overall controller could maintain connectivity of agents, forming cohesive multiple agents, and matching velocities. In order to guarantee goal achievement of multiagent flocking, connectivity is still preserved by keeping the second smallest eigenvalue of flocking topology $\lambda_2 > 0$.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this article.

References

- [1] C. W. Reynolds, "Flocks, herds and schools: a distributed behavioral model," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)*, vol. 21, pp. 25–34, ACM, Anaheim, Calif, USA, 1987.
- [2] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Stable flocking of mobile agents, part i: fixed topology," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 2, pp. 2010–2015, IEEE, Maui, Hawaii, USA, 2003.
- [3] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Stable flocking of mobile agents part II: dynamic topology," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 2, pp. 2016–2021, IEEE, 2003.
- [4] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.
- [5] J. Penders, "Robot swarming applications," in *Liber Amicorum ter gelegenheid van de 60e verjaardag van Prof.dr. H. Jaap van den Herik*, J. Donkers, Ed., pp. 227–234, Maastricht ICT Competence Center, 2007.
- [6] B. Erfianto, B. R. Trilaksono, and H. Hindersah, "Signal strength-based adjacency matrix and its eigenvalue in mobile robotic networks," in *Proceedings of the International Conference of Information and Communication Technology (ICoICT '13)*, pp. 247–252, Bandung, Indonesia, March 2013.
- [7] M. Zavlanos and G. Pappas, "Flocking while preserving connectivity," in *Proceedings of the IEEE Conference on Decision and Control (CDC '07)*, pp. 2919–2924, New Orleans, La, USA, December 2007.
- [8] M. M. Zavlanos, H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Hybrid control for connectivity preserving flocking," *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2869–2875, 2009.
- [9] Y. Mao, L. Dou, H. Fang, H. Liu, and H. Cao, "Connectivity-preserving flocking of multi-agent systems with application towheeled mobile robots," in *Proceedings of the 29th Chinese Control Conference (CCC '10)*, pp. 4494–4500, IEEE, Beijing, China, July 2010.
- [10] H. Fang, Y. Wei, J. Chen, and B. Xin, "Flocking of second-order multiagent systems with connectivity preservation based on algebraic connectivity estimation," *IEEE Transactions on Cybernetics*, 2016.
- [11] M. Ji and M. Egerstedt, "A graph-theoretic characterization of controllability for multi-agent systems," in *Proceedings of the American Control Conference (ACC '07)*, pp. 4588–4593, IEEE, New York, NY, USA, July 2007.
- [12] M. Ji and M. B. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.
- [13] R. K. Williams and G. S. Sukhatme, "Topology-constrained flocking in locally interacting mobile networks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '13)*, pp. 2002–2007, Karlsruhe, Germany, May 2013.

- [14] B. N. Sharma, J. Vanualailai, and U. Chand, "Flocking of multi-agents in constrained environments," *European Journal of Pure and Applied Mathematics*, vol. 2, no. 3, pp. 401–425, 2009.
- [15] C. Godsil and G. F. Royle, *Algebraic Graph Theory*, vol. 207 of *Graduate Texts in Mathematics*, Springer, New York, NY, USA, 2001.
- [16] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.
- [17] P. Ögren, M. Egerstedt, and X. Hu, "A control Lyapunov function approach to multiagent coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 847–851, 2002.
- [18] S. Ge, "Lyapunov design. Control Systems, Robotics And Automation, XIII," 2007.
- [19] B. Sharma, J. Vanualailai, and J. Raj, "Obstacle and collision avoidance control laws of a swarm of boids," *International Journal of Mathematical, Computational, Statistical, Natural and Physical Engineering*, vol. 8, no. 2, pp. 253–258, 2014.
- [20] 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, no. 2, pp. 390–396, 2010.