JES, Vol. 53, No. 1, Jan. 2025 DOI: 10.21608/JESAUN.2024.334576.1382 Part E: Mechatronics Engineering

Journal of
Engineering Sciences
Faculty of Engineering
Assiut University

مجلة
العلوم الهندسية
كلية الهندسة
جامعة أسيوط

journal homepage: http://jesaun.journals.ekb.eg

Distributed Control Design for Formation Control of Double Integrator Multi-Agent Systems Based on Graph Rigidity

Received 7 November 2024; Revised 20 December 2024; Accepted 21 December 2024

Peter Gaber[1]
Mahmoud Abdelrahim[2]
Khalil Ibrahim[3]

Keywords
Distributed control,
Formation control,
Lyapunov stability,
Numerical simulation,
Graph rigidity approach

Abstract: This paper addresses the challenge of controlling the formation of multi-agent system based only on the relative distances between agents obtained individually by local sensors mounted on each autonomous agent in the system. Based on the graph rigidity approach, the inter-agent sensing and communication topology is presented as a rigid graph, and the control model is designed for each agent as a distributed control scheme. This study shows the capability of utilizing graph rigidity in designing distance-based formation control for multi-agent system. It also shows the applicability of the approach to achieve formation control for more complex formations in both two and three-dimensional spaces. To validate the effectiveness and capability of the proposed formation control strategy, three complex formation scenarios are conducted and simulated using MATLAB. These scenarios involve both formation acquisition and maneuvering problems and consider double-integrator multi-agent systems with 5 and 12 agents. The simulation results show the effectiveness of the distance-based formation control based on the graph rigidity, by demonstrating the exponential stability of the controlled system and the convergence of the agents to the desired formation in less than 3 seconds even for a system of 12 agents. The system stability proof is provided using Lyapunov stability theorem. In addition to ensuring system stability, this study shows that the graph rigidity approach implicitly ensures inter-agent collision avoidance. This study demonstrates the effectiveness of using graph rigidity approach in designing formation control of multi-agent system based only on the relative distances between agents, which ensures system stability and can deal with more complex formations, in both two- and three-dimensional spaces, as long as the inter-agent sensing and communication topology can be presented by a rigid graph.

1. Introduction

Nature has been a good inspiration source for many encountered problems in many fields. The collective behavior of group of insects, birds or fish with its capability of performing complex tasks beyond individual capabilities, have inspired humans to design such multi-

[1] Teaching assistant, Mechatronics Engineering Dept., Assiut University, Egypt. peter.a.gaber@aun.edu.eg
[2] Assoc. professor, Renewable Energy Lab, Prince Sultan University, Riyadh 12435, Saudi Arabia. m.abdelrahim@aun.edu.eg
[3] Assoc. professor, Faculty of Industry and Energy Technology, New Assiut Technological University (NATU) / Mechatronics Engineering Dept., Assiut University, Egypt. khalil.ibrahim@aun.edu.eg

https://doi.org/10.21608/jesaun.2024.334576.1382

agent systems (MAS) with distributed, decentralized, coordinated, and collective behavior. This implies that each agent in such systems operates depending on its own local sensing and control mechanisms without any global information, to perform complex tasks collectively with the other agents in the system acting like one whole organism [1]. Using distributed decentralized control method for MASs reduces information processing load through the system compared with central control methods. Specially, distributed and decentralized control overcome the communication limitations and poor connection accuracy between the central controller and each agent in the system which makes distributed control based multi-agent systems more autonomous [2] and [3]. Furthermore, these multi-agent systems are more efficient to accomplish complex tasks compared with one advanced large agent due to their scalability based on scaling number of agents, versatility in combining different types of agents, robustness as it is not a single point of failure system, lower cost; although a MAS has multiple entities not a single one but each agent in the MAS has simpler required capabilities and design compared to that advanced single agent system to achieve same complex task [4], [5]. However, with more opportunities of a MAS, it has its own challenges: coordinating multiple agents cooperatively, avoiding collision between agents in the system, avoiding obstacle in static or dynamic environment, designing decentralized controllers suitable for the distribution of information independent of any global or central intervention [5].

One of the common challenges and considered as a primary challenge for a multi-agent system, especially for that deployed to physical interactive tasks, is Formation Control of the agent coordination. Here we focus our study on two subproblems of the formation control problem for multi-agent system: ***Formation Acquisition Problem***, at which each agent control input is designed so that all agents of the system form a predefined geometric shape in space keeping this formation throughout the required task, and ***Formation Maneuvering Problem***, at which all agents keep predefined formation and simultaneously each agent follows a task-related predefined trajectory in space so that the whole system moves as one unit in space. Such practical applications that consider formation control problem as an essential part of the whole task are large area searching and exploration [6], cooperative transportation, surveillance, reconnaissance, as in military purposes, building structures via autonomous construction vehicles, agriculture and environmental monitoring via multiple unmanned arial vehicles, and improving traffic safety and flow via platoon formation. It doesn't only accommodate multiple vehicles or multiple mobile robots but also for robotic manipulators as in [7]. It is worthy of mention that formation control can be utilized generally for multi-agent system to maintain common relative states between different entities in the system, which states that it does not have to be position-related state to be controlled via formation control method. For example, controlled system states could be frequency, arrival time, voltage, temperature, or pressure.

According to a review of formation control of multi-agent systems [8], formation control methods can be divided into three categories: position-based control methods, displacement-based control methods, and distance-based control methods, according to sensing capabilities of the controlled system and the type of the variables or agent state being controlled. Distance-based formation control for multi-agent systems considers that inter-agent distances are the controlled variables to achieve the required formation given the wanted inter-agent distances. And, the sensing capabilities of each agent is to sense the relative position of each neighboring agent, according to the communication topology of the system, with respect to its local coordinate system [8].

Based on this characteristic of the distance-based formation control, ***graph theory*** provides a natural tool for modeling the whole system of multiple agents, the formation of the agents, and the sensing and communication topology of the system. Utilizing the property of graph rigidity, the entire formation system of agents can be controlled as a rigid graph, with the desired inter-agent distances being the formation system constraints which ensures achieving the desired formation. In addition to obtaining the desired formation, building the formation control based on graph rigidity ensures implicit collision avoidance between the agents.

The first exploitation of rigid graph theory was in 2002 by [9]. Other early work that provided formation control of multi-agent systems based on graph theory can be found in: [10], [11], [12], [13], [14]. Discussions of rigid graph theory and its application to sensing, communication, and control architectures for formations of autonomous vehicles were presented in [4], [15]. A lot of the early research on formation control treated the agents as masses and examined formation control of the agents based on single integrator model [16], [17], [18]. In real life, controlling movable objects needs to be at an acceleration level which can be implemented in actuator level controlling the motion of the target object. Here, double-integrator model was adopted in formation control of multi-agent systems, where the displacement and velocity of the agents are controlled via controlling their accelerations. By using double-integrator model in multi-agent systems, the agents can be considered as mass points whose accelerations are controlled, hence, they can be simply considered as omnidirectional dynamic models.

In [19] and [18], formation maintenance and target interception are achieved using double integrator model and single integrator model, respectively. It has been shown that formation control of multi-agent system that is based on single integrator model has apparent restrictions in an actual implementation when considering the actuator level control of the agent dynamics and its physical constraints. In studying formation control of multi-agent systems based on graph rigidity, literature [20] introduced an extension of formation control of multi-agent systems designed based on single-integrator agent model to obtain multi-agent formation control law based on double-integrator model. In the literature [21], an undirected graph with infinitesimal and minimum rigidity was used to design decentralized formation control for multi-agent system of double-integrator agents, as opposed to the simpler single-integrator agent model. It showed that designing formation control law based on infinitesimal and minimum rigidity ensures the asymptotic stability of the system formation. Exploiting gradient control systems, literature [22] introduced a proof of the local asymptotical stability of undirected formations of both; single-integrator and double-integrator modelled agents in n-dimensional space.

Considering more complex formation of the multi-agent systems, most of the previous studies used simple formations such as triangles and rectangles [23]. For later literatures, more complex formations were introduced, studying its robustness comprehensively. Tetrahedral formations of multi-agent systems in 3D space were studied in literatures [24] [25], and [26].

After the work introduced in [5] to achieve formation maintenance and target interception using rigid graph formation control in velocity level control, an application of distance-based formation control based on rigid graphs to achieve flocking and target interception of multiple nonholonomic agents was implemented in [27]. In [22], and [12], the distance between the second order integrator models is labeled using the Hamiltonian energy system, and asymptotic stabilization is accomplished under the gradient control law.

In this paper, multi-agent distance-based formation control in two- and three- dimensional spaces using double integrator model is investigated. Three different cases will be investigated combining different achievements of formation acquisition control and formation maneuvering control, both in two- and three-dimensional spaces. The paper is organized as follows: in the second section, we introduce some preliminary and concepts of rigid graph theory and infinitesimal rigidity. In the third section, we give the problem statement. In the fourth section the main result of the paper is demonstrated though the provided simulation results. Finally, a conclusion of the work is introduced in the fifth section.

2. Notations and basic concepts based on Graph Theory

For the aim of clearance, this section starts with declaring the list of notations that will be used throughout of the paper as follows:

- $x \in \mathbb{R}^n$ or $x = [x_1, \dots, x_n]$ denotes an $n \times 1$ vector
- $x = [x_1, \dots, x_n]$ where $x_i \in \mathbb{R}^m$ denotes an $mn \times 1$ vector, in this case, vector x will be denoted as $x \in \mathbb{R}^{mn}$
- $\|x\|$ means the Euclidean norm or the second norm of the vector x
- For any points q, $x \in \mathbb{R}^n$ and set Q, $\text{dist}(q, Q) \coloneqq \inf_{x \in Q} \|q - x\|$, is a function that determines the smallest distance between point q to any point x in the set Q.
- $\mathbf{1}_m$ is the $m \times 1$ vector of ones.

An undirected graph G is a pair (V, E), where $V = \{1,2, \dots, n\}$ is the set of n vertices, and $E = (i, j)$ is the set of undirected edges, for all $i, j \in V$ and $i \neq j$, regardless of the order of the vertices in the vertex pair, which means that (i, j) and (j, i) represent the same edge. For any graph G with n vertices and edge set E, there is l number of edges that is $l \in \{1, \dots, n(n-1)/2\}$. Based on graph approach, the set of neighbors of vertex i in graph $G = (V, E)$, is a set of interest and is denoted by $\mathcal{N}_i(E) = \{j \in V | (i, j) \in E\}$. By assigning coordinates for each vertex in an n-vertices graph G with respect to some fixed coordinate frame, a pair (G, q) is obtained where $q = [q_1, \dots, q_n] \in \mathbb{R}^{mn}$, as $q_i \in \mathbb{R}^m$ is the coordinate of vertex i, and m is the number of dimensions of the space coordinates. This pair (G, q) is called a framework, denoted by $F = (G, q)$, which is a realization of the graph so that it can be used to model a geometric formation of multi-agent system in space.

Our main concern here in using graph theory for modelling multi-agent system formation is utilizing graph rigidity. So that, applying the graph rigidity constraints to our framework model will make the multi-agent system behave as one rigid unit with same properties of rigid body. For that end, the utilized concepts to analyze graph rigidity such as infinitesimal rigidity, minimal rigidity and rigidity matrix of frameworks will be introduced here.

For a framework $F = (G, q)$ where $G = (V, E)$, $q \in \mathbb{R}^{mn}$, and its rigidity matrix $R(q): \mathbb{R}^{mn} \to \mathbb{R}^{l \times mn}$ is defined as

$$R(q) = \frac{1}{2} \frac{\partial \phi(q)}{\partial q} \tag{1}$$

where $\phi: \mathbb{R}^{mn} \to \mathbb{R}^l$ is the edge function and defined as

$$\phi(q) = \left[\dots, \left\|q_i - q_j\right\|^2, \dots\right], \qquad (i,j) \in E. \tag{2}$$

According to this definition, the edge function is a vector of l elements, where each element is the square of the Euclidean distance between agent i and agent$j$. The order of the elements in the edge function is correspond to the order of the edges in the edge set E. As the rigidity matrix is the partial derivative of the edge function with respect to vector $q = [q_1, \dots, q_n] \in \mathbb{R}^{mn}$, it has a row for each element in $\phi(q)$ which means a row for each edge in the edge set E with the rows being ordered in the same order of edges in E. Also, it has a column for each element in q, which means m columns for each q_i in q vector, a total of m times n (the number of vertices) columns.

Note that each edge (i,j) in the E set will have its corresponding row in the rigidity matrix $R(q)$ in the following form, which has n row vectors, with each vector being a $1 \times m$ vector.

$$\begin{bmatrix} 0 & \cdots & 0 & (q_i - q_j)^T & 0 & \cdots & 0 & (q_j - q_i)^T & 0 & \cdots & 0 \end{bmatrix} \tag{3}$$

Where $(q_i - q_j)^T$ is at the columns for i^{th} vertex, $(q_j - q_i)^T$ is at the columns for j^{th} vertex, and any other element is equal to zero.

For two frameworks sharing the same graph $G = (V, E)$ with different vertices coordination in $\mathbb{R}^m$, $F = (G, q)$ and $\hat{F} = (G, \hat{q})$; if only their corresponding edges are equivalent in lengths, $\|q_i - q_j\| = \|\hat{q}_i - \hat{q}_j\|$ for all $(i,j) \in E$, they are said to be equivalent frameworks. It can be easy to note that if F and $\hat{F}$ are equivalent, then they have their edge functions be the same, $\phi(q) = \phi(\hat{q})$. Furthermore, if the two frameworks have their corresponding distances between all vertices are equivalent in lengths, $\|q_i - q_j\| = \|\hat{q}_i - \hat{q}_j\|$ for all$i, j \in V$, they are said to be congruent frameworks [28-29].

Distinguishing between these two concepts is so important here in formation control, as two frameworks being equivalent doesn't guarantee that they are congruent frameworks. Figure (1) can illustrate this concept. Here, two frameworks are illustrated, $F = (G, q)$ and $\bar{F} = (G, \bar{q})$, where $G = (V, E)$, $V = \{1, 2, 3, 4\}$, and $E = \{(1,2), (1,4), (2,3), (2,4), (3,4)\}$. They share the same coordination for vertices 1, 2, and 4 only, as $q_3 \neq \bar{q}_3$. Obviously, these two frameworks are not congruent to each other, although they have the same edge function, $\phi(q) = \phi(\bar{q})$, as $\|q_i - q_j\| = \|\bar{q}_i - \bar{q}_j\|$ for all $(i,j) \in E$ only, and this doesn't include $(i,j) = (1,3)$, where $\|q_1 - q_3\| \neq \|\bar{q}_1 - \bar{q}_3\|$. In this case, F and $\bar{F}$ are equivalent but not congruent, and they are called ambiguous frameworks [4]. In general, this is called flip ambiguity and can be occurred when a set of vertices of a graph $G \in \mathbb{R}^m$ lays in a lower dimensional subspace of the m-dimensional space, which is $(m-1)$-dimensional space it is also called a hyperplane in $\mathbb{R}^m$. As the framework F illustrated in figure (1) with its four vertices laying in 2D space can lead to a flip ambiguity occurrence in 3D space. In this work, $Amb(F)$ will be used as a notation of a set of all ambiguous frameworks to a framework$F$.

As opposed to the ambiguous frameworks that are equivalent to each other but don't have to be congruent, there is the concept of isometric frameworks or isomorphic frameworks. Isomorphic frameworks are all frameworks that are related to each other by an isometry in$\mathbb{R}^m$, which is defined as a bijective map $T: \mathbb{R}^m \to \mathbb{R}^m$ [30], where

$$\|T(q_i) - T(q_j)\| = \|q_i - q_j\|, \quad \forall q_i, q_j \in \mathbb{R}^m. \tag{4}$$

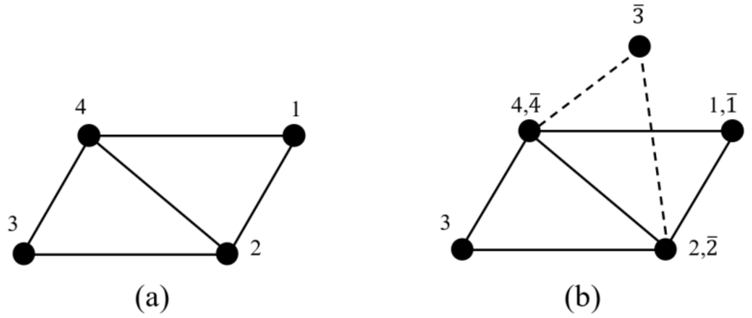


Figure 1 Ambiguous frameworks: (a) Framework F. (b) Framework $\bar{F} \in Amb(F)$ in dashed line with F in solid line.

This means that vector $q_i - q_j$ is translated and/or rotated in the m-dimensional space with preserving the Euclidean distance between points i and$j$. So, when there are two frameworks F and $\hat{F}$ that are said to be isomorphic frameworks, this means that the coordination of one of them can be obtained by translating and/or rotating the other framework. This implies that F and $\hat{F}$ are congruent to each other. Hence, a set of all frameworks that are congruent to framework $F \in \mathbb{R}^m$, are isomorphic to F in $\mathbb{R}^m$, and will be denoted throughout this work by $Iso(F)$.

Practically, the capability of preserving the formation of multi-agent system using distance-based control method is directly related to the rigidity of the framework that represents the sensing and communication network between agents. This is simply the purpose of using graph rigidity. A framework $F = (G, q)$, with n number of vertices that is more than three, and the coordination of the vertices q construct an M dimensional convex hull $\{q_1, \ldots, q_n\}$, is infinitesimally rigid framework in m-dimensional space if and only if $Rank\big(R(q)\big) = mn - \frac{(M+1)(2m-M)}{2}$. For the case of $(M = m)$ which means that the vertices of the framework are distributed to construct a convex hull that occupies m-dimensional space, the rank of rigidity matrix, that implies F is infinitesimally rigid, will be as

$$Rank\big(R(q)\big) = mn - \frac{(m+1)(m)}{2} \text{ [30]}, \qquad \text{[22].}$$

For an infinitesimally rigid framework $F = (G, q)$ with $(n > 3)$ number of vertices, if F is minimally rigid framework then its number of edges must be $l = mn - \frac{(m+1)(m)}{2}$ [4]. Hence, it can be concluded that for an infinitesimally and minimally rigid framework $F = (G, q)$, its rigidity matrix must satisfy that $Rank\big(R(q)\big) = l$ which means it has a full row rank.

3. Problem Statement

In this paper, two formation control problems are being addressed starting with Formation Acquisition, as it is considered the primary objective of formation control. Then, the Formation Maneuvering of multi-agent systems is being demonstrated. Consider a multi-agent system of n agents with u_i being the acceleration-level control input for the i^{th} agent where its relative position to an earth-fixed coordinate frame is $q_i \in \mathbb{R}^m$. Exploiting the graph rigidity, the desired inter-agent distances is denoted by an infinitesimally and slightly rigid framework $F^* = (G^*, q^*)$ where $G^* = (V^*, E^*)$ is the desired formation graph, with n vertices representing agents and l edges representing the only inter-agent distances that need to be controlled, and $q^* = [q_1^*, \dots, q_n^*]$ is the agents desired coordination that achieve the desired formation. The desired distances between agents will be denoted by the vector $d = [\dots, d_{ij}, \dots] \in \mathbb{R}^l$, where d_{ij} is the desired distance between agent i and agent j and is obtained based on the agent desired coordination by

$$d_{ij} = \left\| q_i^* - q_j^* \right\| > 0, \quad i, j \in V^* \tag{5}$$

Let’s build our problem statements based onto the following assumptions:

- **Assumption 1:** Each edge $(i, j) \in E^*$ in the desired formation graph always has its corresponding sensing capability and communication between i^{th} agent and j^{th} agent. In other words, to add or select an edge (i, j) to be in the desired formation graph, there must be an inter-agent sensing capability beteen the corresponding agents i and j.
- **Assumption 2:** The inter-agent sensing, and communication network is always maintained so that agent i is always in the sensing and communication range of its neighbours in the desired formation framework F^*. This means that there is no temporary loss of any sensing or communication connection represented by an edge in the desired framework which in turn guarantees preventing flex ambiguity occurrence.
- **Assumption 3:** The only position information being measured is the proportional position of agent pairs in E^* set, defined in the desired formation. This means that the global position of the agents, $q^* = [q_1^*, \dots, q_n^*]$ is not available for the controllers, only the proportional position of agent pairs, $q_i - q_j$ for $(i, j) \in E^*$.

Based on these assumptions and the background of graph rigidity previewed in the introduction, the formation acquisition problem and the formation maneuvering problem can be detailed as follows.

Formation Acquisition Problem: The agents are required to form and maintain a predefined formation in space. The formation acquisition control objective, which is considered the primary objective of the formation control problems, is designing control input u_i such that

$$F(t) \to Iso(F^*) \text{ as } t \to \infty. \tag{6}$$

In the terms of the inter-agent distances, which is the only position information being measured in the system, the control objective can be represented as

$$\left\| q_i(t) - q_j(t) \right\| \to d_{ij} \ Ast \to \infty, \text{where} i, j \in V^*. \tag{7}$$

And this makes the actual formation to converge to any isometric realization of F^*. In terms of graph theory, the formation will converge to one framework in the set of $Iso(F^*)$ based on the initial position of the formation coordination, $q(0) = [q_1(0), \dots, q_n(0)]$.

Formation Maneuvering Problem: While maintaining a predefined formation, agents are required to maneuver based on a predefined trajectory. So, the formation maneuvering control objective is

$$\dot{q}_i(t) - v_{di}(t) \to 0 \text{ As} t \to \infty, \text{where } i = 1, \dots, n \tag{8}$$

Where $v_{di} \in \mathbb{R}^3$ represent the required rigid body velocity for the swarm of agents based on their mission to be accomplished. So, the formation control has to achieve the formation acquisition simultaneously with the formation maneuvering so that the formation moves in translation motion, rotational motion, or both as a virtual rigid body. According to the problem statements, and the sensing capability and the interaction topology of agents, graph rigidity approach, which is a distance-based control scheme, is adopted. This paper introduces formation control for multi-agent system, based on graph rigidity, using double-integrator model. In the following section the double-integrator model will be previewed in detail and the control design to achieve the formation control objectives stated in (6), (7), and (8).

4. Double Integrator Model

The double-integrator model accounts for the agent acceleration by treating the agent as a point mass. Therefore, it can be considered a very simple dynamic model for omnidirectional robots. Given a system of n agents, the equations of motion for the double-integrator model are

$$\dot{q}_i = v_i \tag{9}$$

$$\dot{v}_i = u_i, \quad i = 1, \dots, n \tag{10}$$

where:

- $v_i \in \mathbb{R}^m$ represents the velocity of the i^{th} agent with respect to an Earth-fixed coordinate frame of m dimensions,
- $u_i \in \mathbb{R}^m$ is the acceleration-level control input of the i^{th} agent,
- and $q_i \in \mathbb{R}^m$ is the position of the i^{th} agent.

Here, the agent velocity is a system state and the formation control laws in this work will be a function of the agent velocities in addition to the positions.

By exploiting the **integrator backstepping methodology**, the double-integrator-based control laws can be obtained as an extension of the single-integrator-based control laws.

As distance-based controller, the inputs $u_i, i = 1, \dots, n$ will control the relative distances $\|q_i - q_j\|$, for all edges $(i,j) \in E^*$, where E^* is the set of edges defined in the desired formation. So, the objective is to ensure that

$$\|q_i(t) - q_j(t)\| \to d_{ij} \text{ As} t \to \infty, (i,j) \in E^*, \tag{11}$$

Where d_{ij} is the required relative distance between agent i and agent j, defined in the desired formation. To simplify the notation in the following derivations, the proportional position of two agents will be defined as

$$\tilde{q}_{ij} = q_i - q_j \tag{12}$$

And $\tilde{q} = [\dots, \ \tilde{q}_{ij}, \dots] \in \mathbb{R}^{ml}, (i,j) \in E^*$, with l is the number of edges of the graph defined in the desired formation. The distance error is given by

$$e_{ij} = \|\tilde{q}_{ij}\| - d_{ij}. \tag{13}$$

The distance error dynamics can be derived from equations (9) and (13) as

$$\dot{e}_{ij} = \frac{d}{dt}\left(\sqrt{\tilde{q}_{ij}^T \tilde{q}_{ij}}\right) = \frac{1}{2}\left(\tilde{q}_{ij}^T \tilde{q}_{ij}\right)^{-\frac{1}{2}}\left(2\, \tilde{q}_{ij}^T \dot{\tilde{q}}_{ij}\right) = \frac{\tilde{q}_{ij}^T (v_i - v_j)}{e_{ij} + d_{ij}} \tag{14}$$

As assumed that the only measured quantities throughout the multi-agent system are the relative distances between agents, specifically those in the E^* set. The agents' velocities $v = [v_1, \dots, v_n] \in \mathbb{R}^{mn}$ are considered only as system states and can not be directly obtained. One solution is by utalizing backstepping technique and introduce the following variable

$$s = v - v_f \tag{15}$$

where $v_f \in \mathbb{R}^{mn}$ denotes the desired velocity input and considered as a fictitious control input, which will be specified later according to the problem need to be solved; formation acquisition problem, formation maneuvering problem, or both. The variable s quantifies the agents velocity error as it is the error between the actual agent velocity and the desired velocity-level input. The desired of v_f will be problem-specific and will be obtained based on the velocity-level control laws of a single-integrator model, where the control input u is the velocity-level input. The block diagrams in Figure 2 illustrates the relationship between the control designs for the single-integrator and double-integrator models. As one can see, the velocity-level, position control algorithms from single-integrator model will be embedded in the acceleration-level, velocity control loop to be designed in the double-integrator model.
Due to the new error variable s introduced in equation (15), an augmented Lyapunov function candidate will be introduced as follows:

$$W_d(e, s) = W_1(e) + W_2(s) \tag{16}$$

Where $W_1(e)$ and $W_2(s)$ are defined as follows,

$$W_1(e) = \frac{1}{4} z^T z = \frac{1}{4} \sum_{(i,j)\in E^*} z_{ij}^2, \tag{17}$$

And

$$W_2(s) = \frac{1}{2} s^T s \tag{18}$$

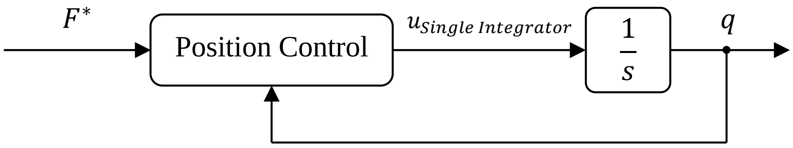
Where $e = [\dots, e_{ij}, \dots] \in \mathbb{R}^l, (i,j) \in E^*$ and $z = [\dots, z_{ij}, \dots] \in \mathbb{R}^l, (i,j) \in E^*$ is a new introduced variable that is

$$z_{ij} = \|\tilde{q}_{ij}\|^2 - d_{ij}^2, \tag{19}$$

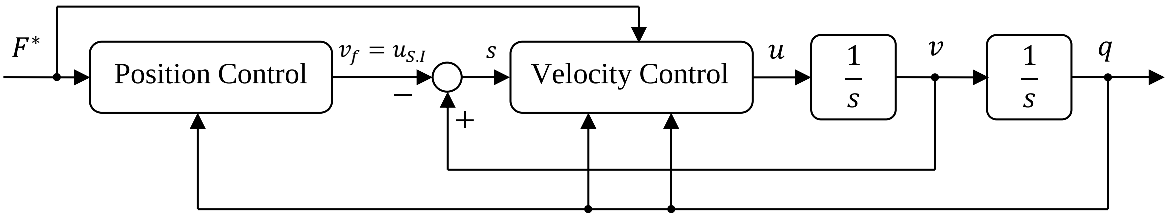
Which can be rewritten in terms of position error from equation (13) as

$$z_{ij} = e_{ij}(e_{ij} + 2d_{ij}) \tag{20}$$

Notice that $W_1(e)$ is a potential energy term since its dependence on position only, while $W_2(s)$ is a kinetic energy term due to its dependence on velocity. Furthermore, $W_d(e, s)$ captures the total energy of the double integrator model formation.



(a) single integrator control model



(b) double integrator control model: Note the implicit inclusion of the $u_{single\ Integrator} = u_{S.I}$ into the double integrator control model.

Figure 2 Illustration of the double integrator control model and its relation to the single integrator control model.

The derivative time of$W_d(e, s)$, it is obtained that

$$\dot{W}_d = \dot{W}_1(e) + \dot{W}_2(s) = \frac{1}{2} z^T \dot{z} + s^T \dot{s} \tag{21}$$

Where $\dot{W}_1$ can be presented using equations (14), (17), (19), and (20) as

$$\dot{W}_1 = \frac{1}{2} z^T \dot{z} = \frac{1}{2} \sum\nolimits_{(i,j)\in E^*} z_{ij} \dot{z}_{ij} = \frac{1}{2} \sum\nolimits_{(i,j)\in E^*} e_{ij}(e_{ij} + 2d_{ij})\, \dot{z}_{ij} \tag{22}$$

Where

$$\dot{z}_{ij} = \frac{d}{dt}\left(\left\|\tilde{q}_{ij}\right\|^2\right) = \frac{d}{dt}\left(\sqrt{\tilde{q}_{ij}^T \tilde{q}_{ij}}^{\,2}\right) = 2\, \tilde{q}_{ij}^T \dot{\tilde{q}}_{ij} = 2\, \tilde{q}_{ij}^T (v_i - v_j) \tag{23}$$

Then, using equation (23) into (22), $\dot{W}_1$ will be in element-wise form as follows

$$\dot{W}_1 = \sum\nolimits_{(i,j)\in E^*} e_{ij}(e_{ij} + 2d_{ij})\, \tilde{q}_{ij}^T (v_i - v_j) \tag{24}$$

Based on the rigidity matrix definition and the introduced variable z_{ij} in equation (20), $\dot{W}_1$ can be conveniently written as

$$\dot{W}_1 = z^T R(\tilde{q}) v \tag{25}$$

Where $v = [v_1, \dots, v_n] \in \mathbb{R}^{mn}$ is the stacked vector of n velocity vectors of each agent, and the rigidity matrix$R(\tilde{q}) \in \mathbb{R}^{l\times mn}$?. Using equations (25) and (15) into (21), the derivative time of the augmented Lyapunov function can be written as

$$\begin{aligned} \dot{W}_d = z^T R(\tilde{q}) v + s^T \dot{s} &= z^T R(\tilde{q})(s + v_f) + s^T(u - \dot{v}_f) \\ &= z^T R(\tilde{q}) v_f + s^T(u + R^T(\tilde{q}) z - \dot{v}_f) \end{aligned} \tag{26}$$

And it will be the initial point for all double-integrator control designs for formation acquisition problems or formation maneuvering problems.

Formation Acquisition

The formation acquisition controller for the double-integrator model will have the general form $u_i = u_i\left(q_i - q_j, v_i - v_j, v_i, d_{ij}\right)$, $i = 1, \dots, n$ and $j \in N_i(E^*)$ where $N_i(\cdot)$ is the set of neighbours of the i^{th} agent. Based on equation (26), the control

$$u = -k_a s + \dot{v}_f - R^T(\tilde{q})z, \tag{27}$$

Where

$$v_f = -k_v R^T(\tilde{q})z, \tag{28}$$

And $k_a > 0$ is a defined control gain, renders $W_d(e, s) = 0$ exponentially stable and ensures that formation converge to the desired construction and equation (11) is satisfied.

A control input u will be selected based on the augmented Lyapunov function in equation (26) so that it ensures the stability of the system. The control input u will be selected so that $\dot{W}_d < 0$ around the$(e, s) = (0,0)$. Using equation (26), each term of $\dot{W}_d$ will be set to be subjected to the following inequalities

$$z^T R(\tilde{q}) v_f \leq -K_v M_1 \tag{29}$$

$$s^T\left(u + R^T(\tilde{q})z - \dot{v}_f\right) \leq -K_a M_2 \tag{30}$$

Where K_v and K_a are sufficiently small positive definite scalar constants, and M_1 and M_2 are positive definite 1-by-1 matrices. Then, selecting v_f and u that undergo these inequalities, will ensure that $\dot{W}_d$ is negative definite. One simple way to select v_f is by selecting its value so that the term $z^T R(\tilde{q}) v_f$ becomes in the form of$-K_v A^T A$, where A is a vector of real values. If it is assumed that$A^T = z^T R(\tilde{q})$, then v_f can be selected to be$v_f = -K_v A$, and that is

$$v_f = -K_v\left(z^T R(\tilde{q})\right)^T = -K_v R^T(\tilde{q})z \tag{31}$$

By substituting the selected value of v_f in $z^T R(\tilde{q}) v_f$ it is obtained that

$$-K_v z^T R(\tilde{q}) R^T(\tilde{q}) z < 0 \quad \forall\, z \in \mathbb{R}^L, R \in \mathbb{R}^{L\times mn}$$

For all $z \in \mathbb{R}^L$ and$R \in \mathbb{R}^{L\times mn}$, where L is the number of edges in the controlled formation and mn is the number of agents times its special dimensions.

Then, the same considerations can be followed for selecting$u$. The selected control input must ensure that$s^T\left(u + R^T(\tilde{q})z - \dot{v}_f\right) \leq -K_a.M_2$. One simple way to achieve this is by making

$$\left(u + R^T(\tilde{q})z - \dot{v}_f\right) = -K_a s, \tag{32}$$

So that $s^T\left(u + R^T(\tilde{q})z - \dot{v}_f\right)$ will be in the form of$-K_a s^T s$, where s is vector of real values. Hence,

$$u = -K_a s + \dot{v}_f - R^T(\tilde{q})z \tag{33}$$

Where this control input ensures that the term $s^T\left(u + R^T(\tilde{q})z - \dot{v}_f\right) < 0$ and is negative definite. This can be demonstrated by substituting by the selected value of$u$, it gets that

$$s^T\left(u + R^T(\tilde{q})z - \dot{v}_f\right) = s^T\left(-K_a s + \dot{v}_f - R^T(\tilde{q})z + R^T(\tilde{q})z - \dot{v}_f\right) = -K_a s^T s < 0$$

For all $s \in \mathbb{R}^{mn}$ vector of (mn) real elements.
Then these selection of v_f and u will ensure that $\dot{W}_d < 0$ for all $t \geq 0$, hence W_d is nonincreasing for$t \geq 0$. And this ensures the stability of the system at the origin(e, s).
Now $\dot{v}_f$ is given by

$$\dot{v}_f = -k_v \dot{R}^T z - k_v R^T \dot{z} \tag{34}$$

Where

$$\dot{R}(\tilde{q}) = R(\tilde{v}), \tag{35}$$

$\tilde{v} = \left[\dots, v_i - v_j, \dots\right] \in \mathbb{R}^l, (i,j) \in E^*$, And from (17)

$$\dot{z} = 2R(\tilde{q})v. \tag{36}$$

The control law in equations (27) and (28), can be rewritten in element-wise form as

$$u_i = -k_a v_i - \sum_{j \in N_i(E^*)}\left[(k_a k_v + 1)\tilde{q}_{ij} z_{ij} + k_v\left(z_{ij} I_m + 2\tilde{q}_{ij}\tilde{q}_{ij}^T\right)\tilde{v}_{ij}\right] \tag{37}$$

For $i = 1, \dots, n$ and

$$\tilde{v}_{ij} = v_i - v_j,\ (i,j) \in E^*. \tag{38}$$

This control is decentralized, since its operation only needs each agent to measure its own velocity and the proportional position and relative velocity to adjacent agents. The agent's velocity was measured onboard sensors.

Maneuver of Formation

The formation steering control law for the double-integrator model is simply a combination of the designs of formation acquisition control law of a single-integrator model and formation maneuvering of the double-integrator model. The control u is given by (27) with

$$v_f = u_a + v_d \tag{39}$$

Where

$$u_a = -k_v R^T(\tilde{q})z \tag{40}$$

Is the formation acquisition control law of a single-integrator model, and the formation maneuvering velocity is$v_d = [v_{d1}, \dots, v_{dn}] \in \mathbb{R}^{3n}$, where for i^{th} agent

$$v_{di} = v_0 + \omega_0 \times \tilde{q}_{in} \tag{41}$$

$v_0(t) \in \mathbb{R}^3$ Means the desired change velocity for the formation, $\omega_0(t) \in \mathbb{R}^3$ is the desired angular velocity.

The term $\dot{v}_f$ in equation (27) will be as equation (34) with additional terms due to$\dot{v}_d$.
As $\dot{v}_{di}$ is given by

$$\dot{v}_{di} = \dot{v}_0 + \dot{\omega}_0 \times \tilde{q}_{in} + \omega_0 \times \tilde{v}_{in},\ i = 1, \dots, n \tag{42}$$

Where $\dot{v}_0 \in \mathbb{R}^3$ denotes the desire change acceleration and $\dot{\omega}_0 \in \mathbb{R}^3$ is the desired angular acceleration for each agent about the position of the leader, as $\tilde{q}_{in}$ and $\tilde{v}_{in}$ are, respectively, the proportional position and the proportional velocity of the i^{th} agent with respect to the leader.
Then using equations (35) and (36), $\dot{v}_f$ will be obtained as

$$\dot{v}_f = -k_v \dot{R}^T z - k_v R^T \dot{z} + \dot{v}_d \qquad = -k_v R^T(\tilde{v}) z - k_v R^T(\tilde{q})(2R(\tilde{q})v) + \dot{v}_d \tag{43}$$

Substituting bys, v_f, and $\dot{v}_f$ from equations (39), (40), (42), and (43) into the control input u of equation (27) yields

$$\begin{aligned} u &= -k_a(v - v_f) + (-k_v R^T(\tilde{v})z - k_v R^T(\tilde{q})(2R(\tilde{q})v) + \dot{v}_d) - R^T(\tilde{q})z \\ &= -k_a(v - (u_a + v_d)) + (-k_v R^T(\tilde{v})z - k_v R^T(\tilde{q})(2R(\tilde{q})v) + \dot{v}_d) - R^T(\tilde{q})z \\ &= -k_a v + k_a u_a + k_a v_d - k_v(R^T(\tilde{v})z + 2R^T(\tilde{q})(R(\tilde{q})v)) + \dot{v}_d - R^T(\tilde{q})z \\ &= -k_a v - k_a k_v R^T(\tilde{q})z + k_a v_d - k_v(R^T(\tilde{v})z + 2R^T(\tilde{q})(R(\tilde{q})v)) + \dot{v}_d - R^T(\tilde{q})z \end{aligned} \tag{44}$$

By rearranging the terms u will be as follows

$$\begin{aligned} u &= -k_a v - k_a k_v R^T(\tilde{q})z - R^T(\tilde{q})z - k_v(R^T(\tilde{v})z + 2R^T(\tilde{q})(R(\tilde{q})v)) + k_a v_d + \dot{v}_d \\ &= -k_a v - (k_a k_v + 1)R^T(\tilde{q})z - k_v(R^T(\tilde{v})z + 2R^T(\tilde{q})(R(\tilde{q})v)) + [k_a v_d + \dot{v}_d] \end{aligned} \tag{45}$$

Then, the control was written in element-wise form as

$$\begin{aligned} u_i &= -k_a v_i - \sum_{j \in N_i(E^*)} \left[(k_a k_v + 1)\tilde{q}_{ij} z_{ij} + k_v \tilde{v}_{ij} z_{ij} + 2\tilde{q}_{ij}\tilde{q}_{ij}^T \tilde{v}_{ij}\right] \\ &\quad + \sum_{i \in V^*} [k_a v_{di}(\tilde{q}_{in}) + \dot{v}_{di}(\tilde{q}_{in}, \tilde{v}_{in})] \\ &= -k_a v_i - \sum_{j \in N_i(E^*)} \left[(k_a k_v + 1)\tilde{q}_{ij} z_{ij} + k_v(z_{ij} I_m + 2\tilde{q}_{ij}\tilde{q}_{ij}^T)\tilde{v}_{ij}\right] \\ &\quad + \sum_{i \in V^*} [k_a v_{di}(\tilde{q}_{in}) + \dot{v}_{di}(\tilde{q}_{in}, \tilde{v}_{in})] \end{aligned} \tag{46}$$

For the double-integrator model, v_0 and ω_0 essential to be continuously differentiable functions of time with bounded first derivative for the control input to be continuous and bounded. Like v_0 andω_0, the signals $\dot{v}_0$ and $\dot{\omega}_0$ was stored on all agent’s onboard computer since they are usually known a priori.

5. Simulation Results

In this section, three scenarios are used to demonstrate the validation of the proposed formation control laws solving formation achievement and formation maneuvering problems. In the first case, a 2D wedge formation achievement problem is introduced. In the second case, 12 agents deployed to form icosahedral formation in 3D space starting from arbitrary initial positions. Then, the same wedge formation used in the first case is extended in the third case to perform a circular trajectory while maintaining the wedge formation. In the first and second cases, the double integrator formation acquisition control law in equation (27) is deployed, based on v_f in equation (28). And for the formation maneuvering problem in case 3, the control low in equation (45) is deployed. All the cases considered the following equation to randomly select the initial position of each agent.

$$q_i(0)_{m\times 1} = q^*_{i\,m\times 1} + \alpha\,[2\,rand(0,1)_{m\times 1} - \mathbf{1}_{m\times 1}] \tag{47}$$

where initial position vector for each agent is column vector of m elements for the considered m dimensional space, α is a positive constant real value used to adjust the amount of deviation of the agents, $rand(0,1)$ is a randomization function that generates column vector of m elements of real values uniformly distributed on the interval (0,1), and $\mathbf{1}_{m\times 1}$ is column vector of m elements of ones. Designing the randomization function in this way provides uniformly distributed values on the interval $(-\alpha, +\alpha)$ that used to deviate the agents away from the desired formation. The value of α is mainly adjusting the deviation of the whole multi-agent system to be away from an isomorphic framework of F^* (the desired framework) and in the same way closer to $Iso(F^*)$ than$Amb(F^*)$. This ensures that the system is stable around an equilibrium point corresponding to a desired formation.

Case 1: 2D Double Integrator Formation Acquisition

In this case, five agents are controlled to form a wedge in 2D space. The desired framework of the wedge formation is shown in Figure 3. The coordination of the five agents in the wedge formation starting from agent 1 to agent 5 is as follows: $(0,0)$, $(-1,-0.5)$, $(-2,-1)$, $(2,-1)$, and $(1,-0.5)$, where agent 1 is at the head of the wedge and the other agents are order in counterclockwise around the generated convex hull. As this formation control method require that the graph presenting the formation must be infinitesimally and minimally rigid graph to ensure formation stability, the number of edges of the framework must submit with the condition that$l = mn - \frac{m(m+1)}{2}$, where n is the number of agents and m here is the agent coordination dimensions and the dimension of the convex hull generated by the agent coordination. Hence, seven edges ($l = 2 * 5 - 3 = 7$) is sufficient to satisfy the condition of infinitesimal and minimal graph rigidity.

The primary positions of the agents were randomly select based on equation (47) where$\alpha = 0.25$. In this formation acquisition problem, the formation control law of equation (27) is deployed using (28) and$k_v = k_a = 1$. Note that these control gains affect the speed of the system convergence to the desired formation.

Figure 4 shows the agent trajectories to form the predefined desired formation demonstrated in **Figure 3**. The stability of the formation acquisition control system is proofed to be achievable based on the convergence of the inter-agent distance between all agents in V^* to the desired distances in F^* and the convergence of the introduced variable s to zero, which can be considered as the agent velocity error. It is shown in **Figure 5** that all inter-agent distance errors are converging to zero, and **Figure 6** shows the zero convergence of the x and y components of all agents' velocity errors. In **Figure 7**, the control inputs $u_i(t)$ for $i = 1, \ldots, 5$ are shown in the form of its components the direction of each dimension of the considered two-dimensional space.

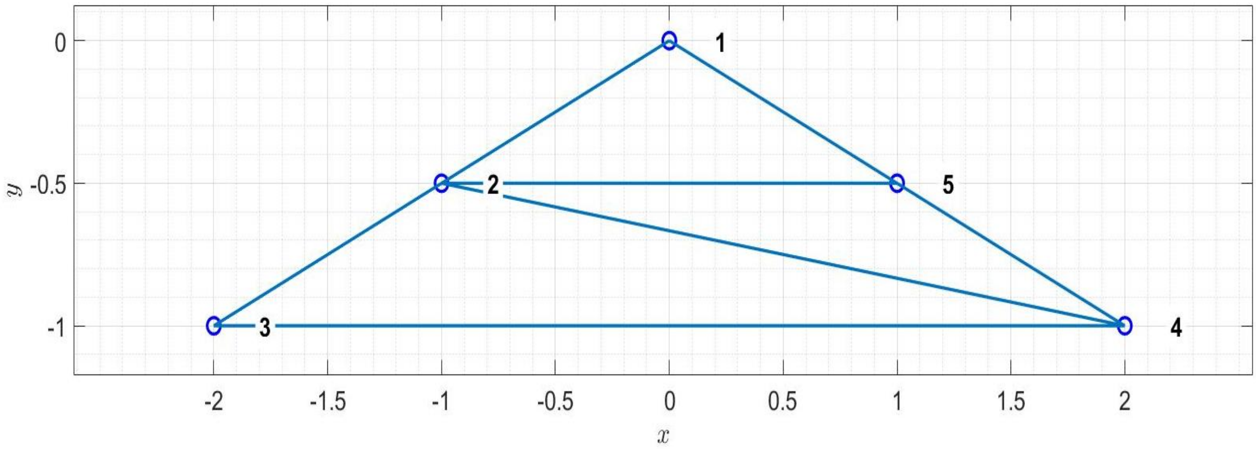


Figure 3 Formation achievement: the desired formationF^*.

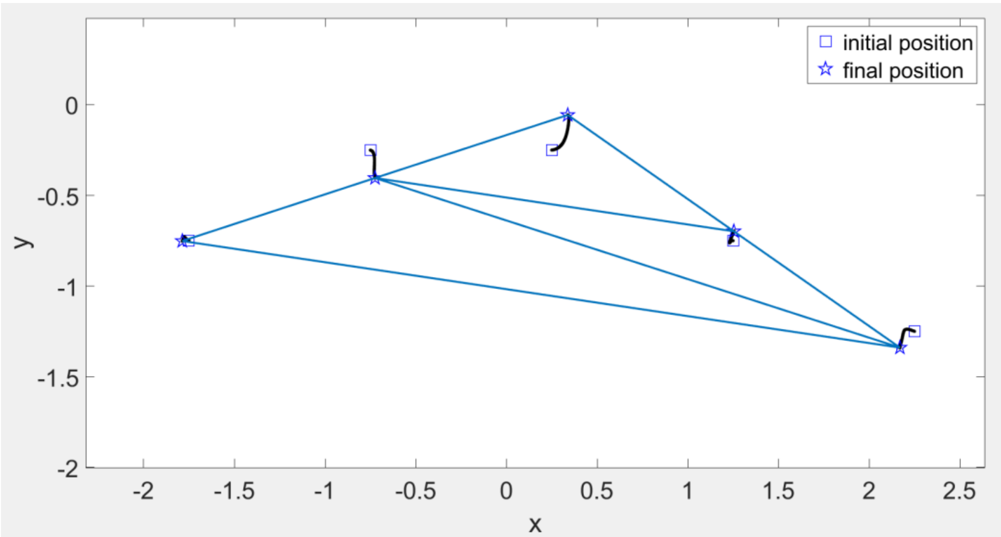


Figure 4 Formation acquisition: Agent trajectories$q_i(t) \forall i \in V^*$, that satisfy formation acquisition control objective.

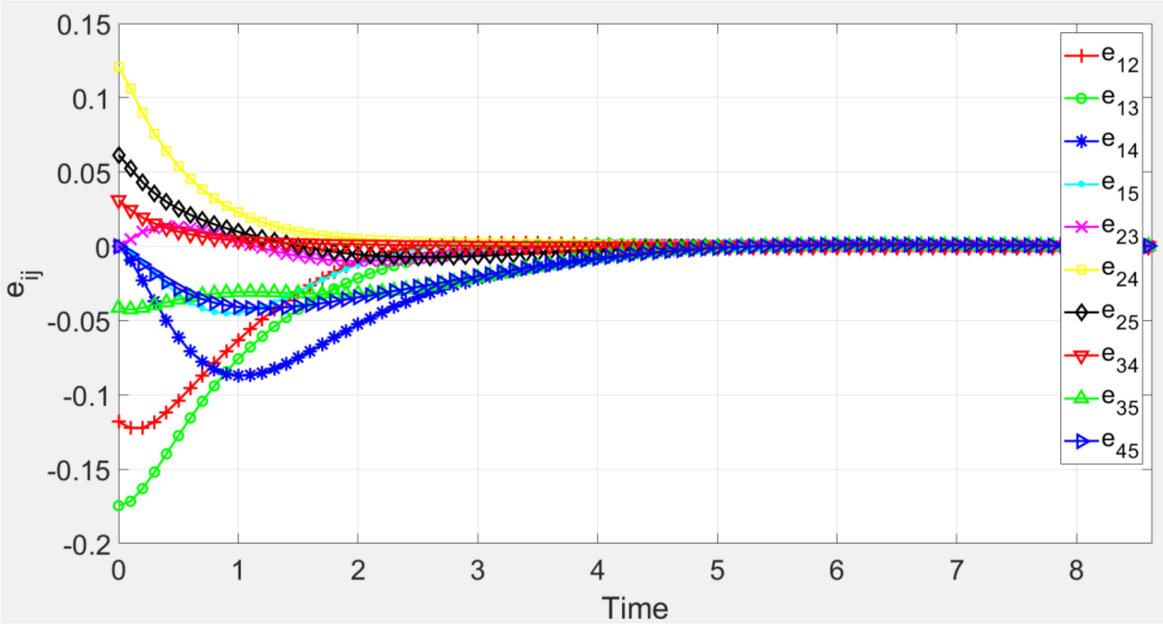
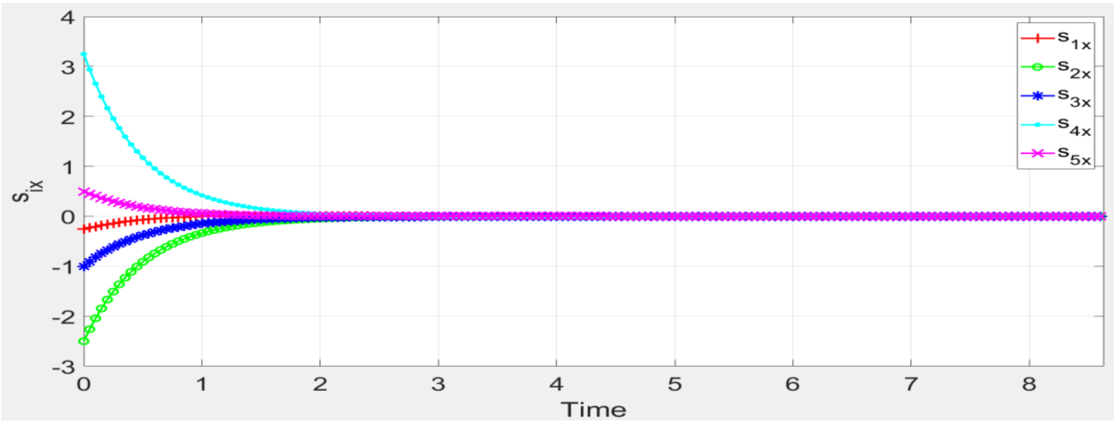
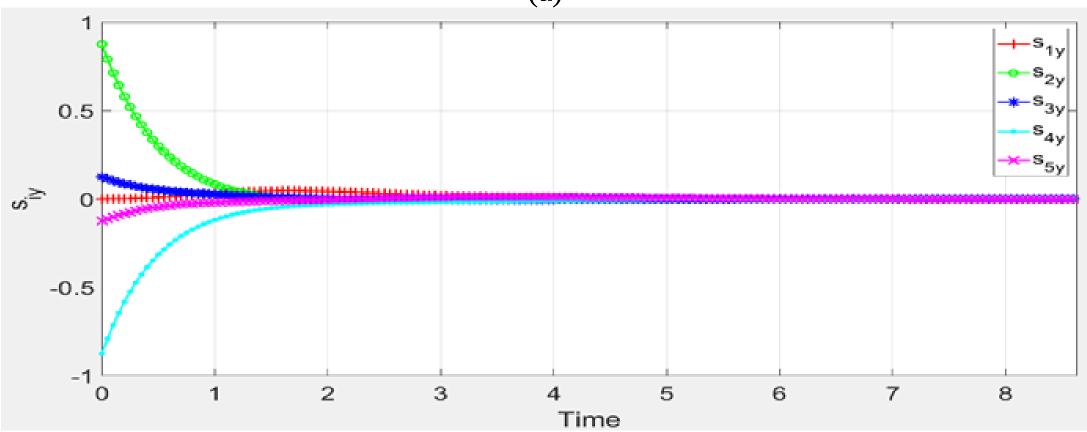


Figure 5 Formation acquisition: distance errors$e_{ij}(t) \forall i, j \in V^*$.

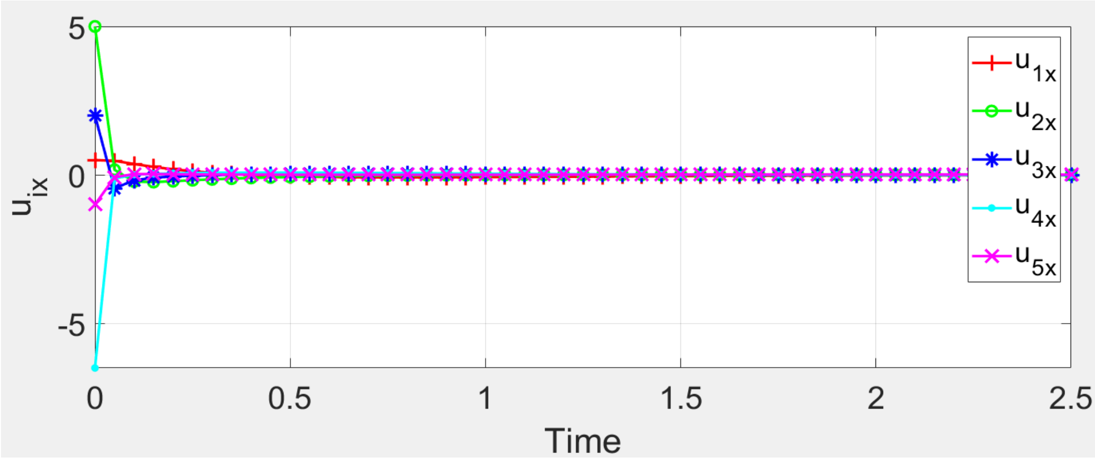


(a)

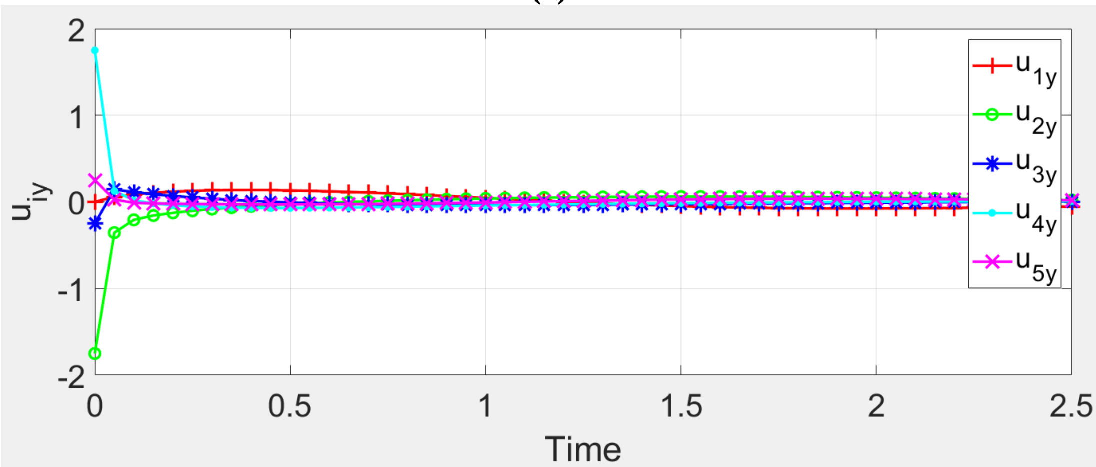


(b)

Figure 6 Formation Acquisition: (a) Velocity errors along x axis (b) velocity errors along y axis.



(a)



(b)

Figure 7 Formation acquisition: Control inputs $u_i(t), i \in V^*$ where (a) is for control inputs along x axis and (b) is for control inputs along y axis.

Case 2: 3D Double Integrator Formation Acquisition

In this case a simulation of formation acquisition of 12 agents forming a regular convex icosahedral geometric shape in 3D space is conducted. The chosen formation of the regular convex icosahedral form is shown in **Figure 8**. The coordination of agents at the vertices of the icosahedron are constructed so that the edge length is equal to 2. This is done by using the Cartesian coordinate system of: $(\pm\varphi, \pm 1, 0), (\pm 1, 0, \pm\varphi), (0, \pm\varphi, \pm 1)$, where $\varphi = (1+\sqrt{5})/2$ denotes the golden ratio. The 30 edges of the convex regular icosahedron are sufficient to satisfy the requirement of the graph being minimally and infinitesimally rigid in$\mathbb{R}^3$, as$3n - 6 = 3 \times 12 - 6 = 30$.

The primary positions of the agents were randomly generated using equation (47) where α were selected to be0.4. The same formation control law of equation (27) is submitted to this formation using equation (28) and setting control gains k_v and k_a both to1.

It is shown in **figure 9** that all the 12 agents have successfully formed the desired regular convex icosahedral. **Figure 10** and **11** shows respectively the inter-agent distance errors between all the 12 agents and the components of the velocity error of each agent in the three dimensions x, y, and z. Both distance and velocity errors are converged to zero demonstrating the stability of the formation acquisition control system used for the 3D formation acquisition problem. The x, y, and z directional components of the control inputs of all the 12 agents are demonstrated in **figure 12**.

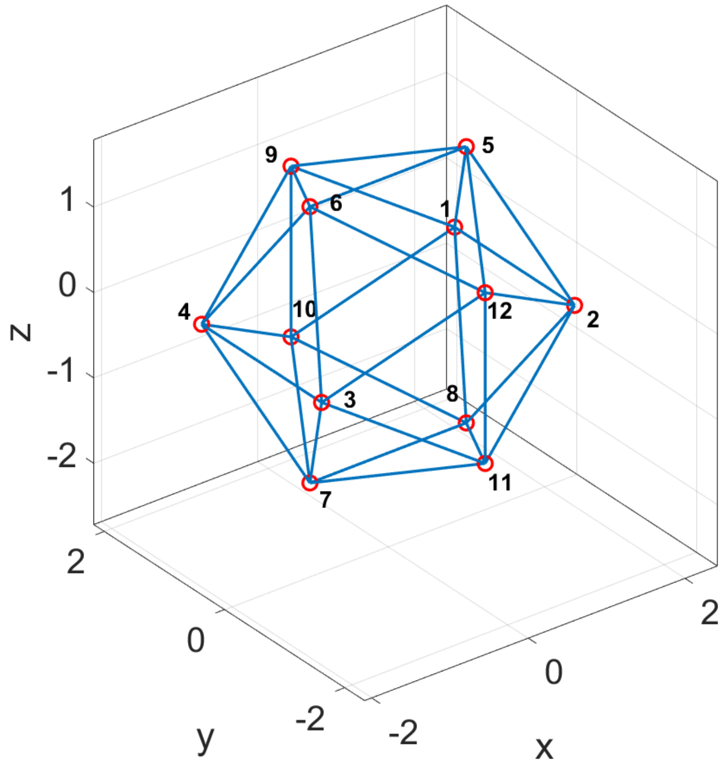


Figure 8 Desired formation of a regular convex icosahedron.

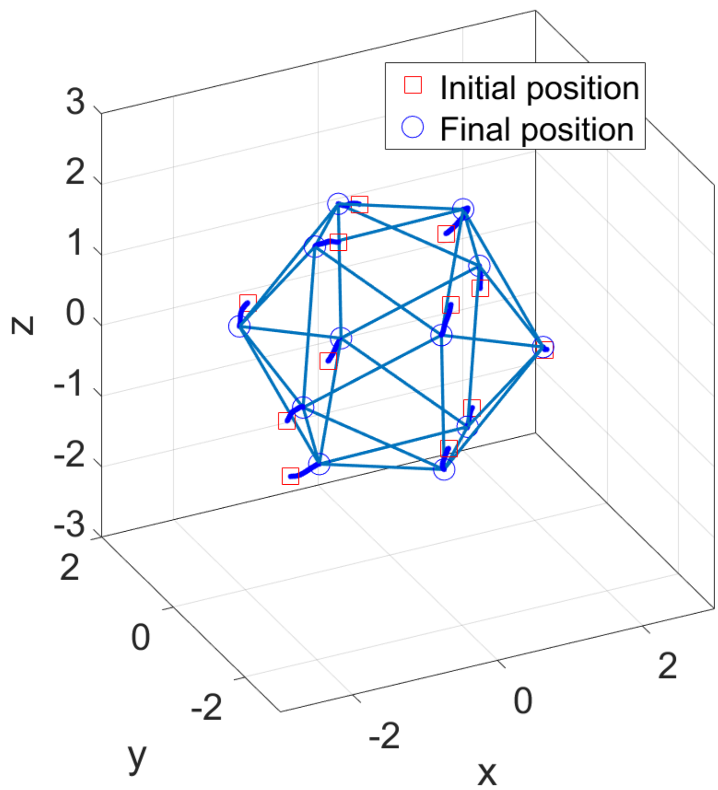


Figure 9 Three-dimensional formation acquisition: all 12 agent trajectories to form the desired regular convex icosahedron.

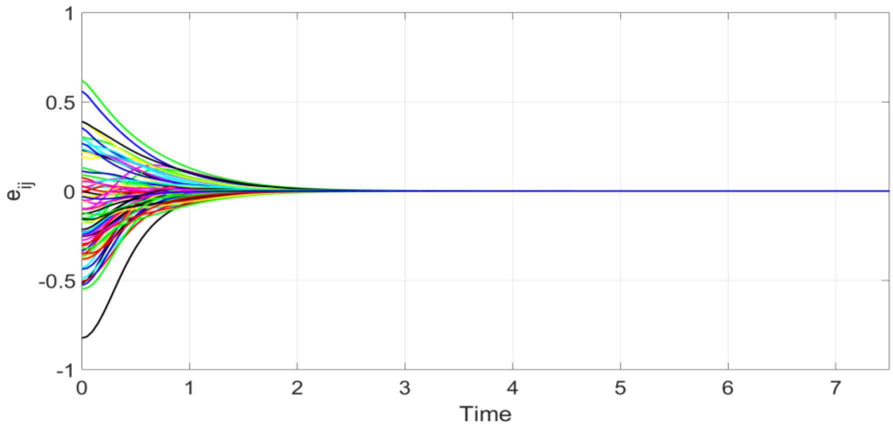
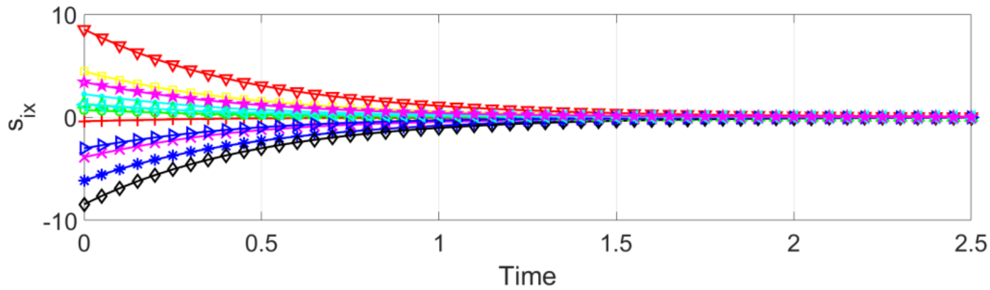
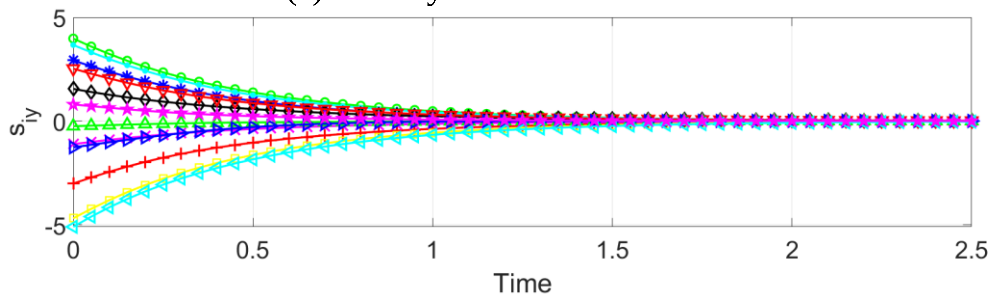


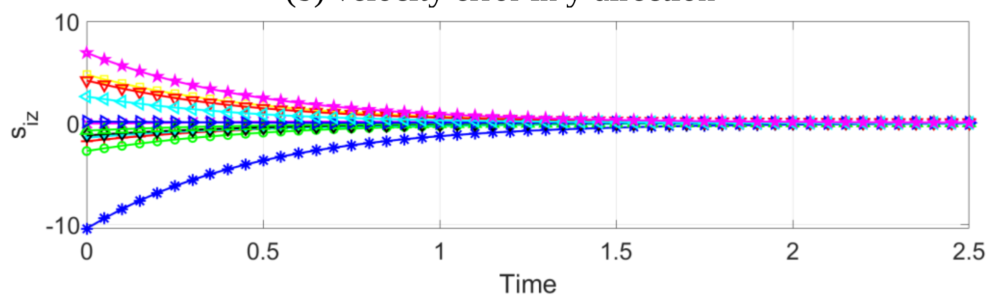
Figure 10 Three-dimensional formation acquisition: distance errors of the distances between all the 12 agents e_{ij} where$i, j \in V^*$.



(a) velocity error in x direction

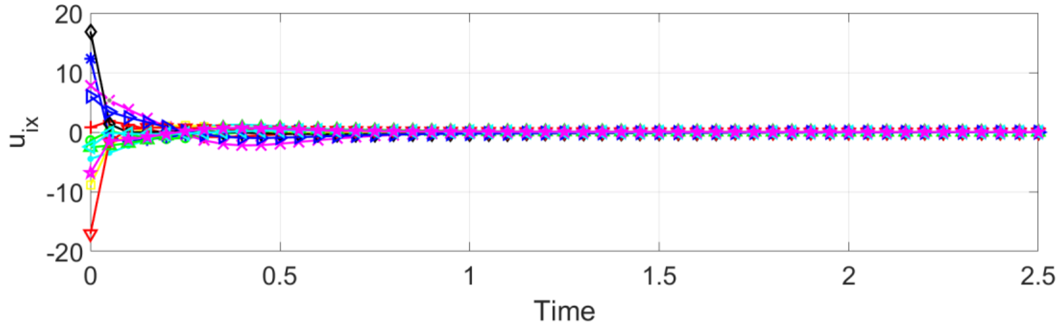


(b) velocity error in y direction

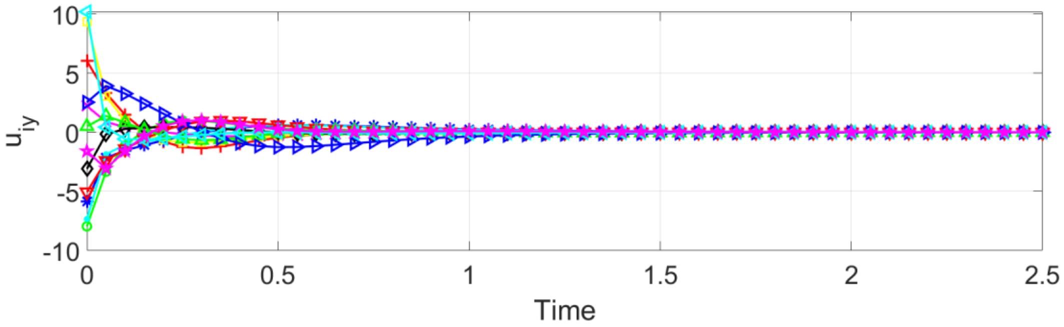


(c) velocity error in z direction

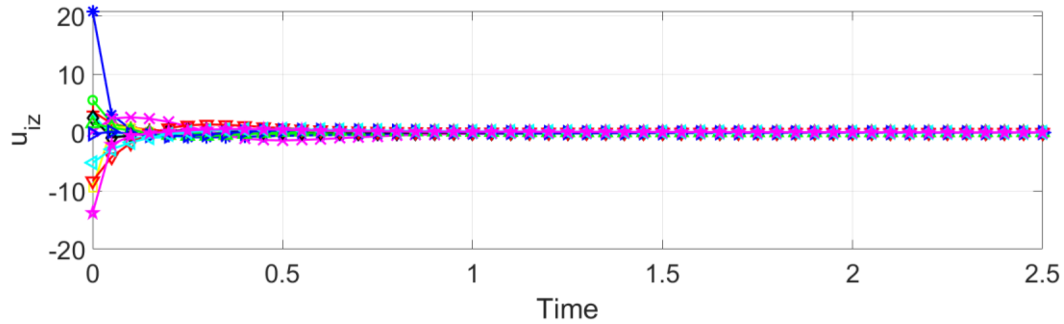
Figure 11 Three-dimensional formation achievement: Velocity error for each agent in the x, y, and z directions for all 12 agents where $\mathrm{i} = 1, \dots, 12$



(a)



(b)



(c)

Figure 12 Three-dimensional formation acquisition: The control inputs for all the 12 agents in x axis (subfigure a), y axis (subfigure b), and in z axis (subfigure c) where $i = 1, \dots, 12$

Case 3: 2D Double Integrator Formation Maneuvering

In the third simulation case, the deployment of the formation control law in (45) is simulated solving formation maneuvering problem using the same wedge formation shape used in the first case. In this maneuvering problem, five agents are desired to maintain a wedge formation according to the same graph used in case 1 (review figure **3**) and move in a circular trajectory of radius 5 where agent 1 being in the head of the formation and considered as the leader of all the other agents that keep rotating around the leader to provide some sort of fixed orientation of the whole formation. The required wedge formation with its vertices' coordination is in the following order starting from agent 1: $(0,0), (-0.5,1), (-1,2), (-1,-2)$, and$(-0.5,-1)$. Here, α is selected to be equal0.2, and simulated the system using control gains k_v and k_a both equals ½.

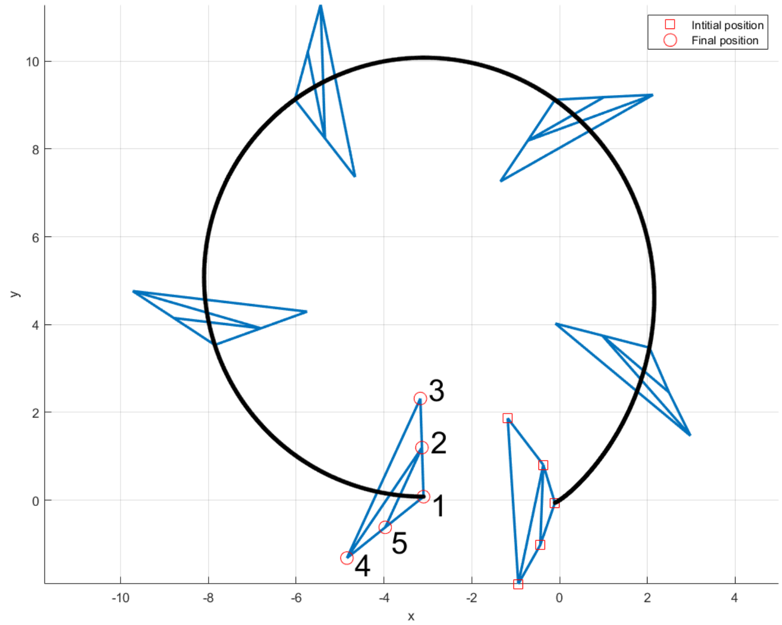
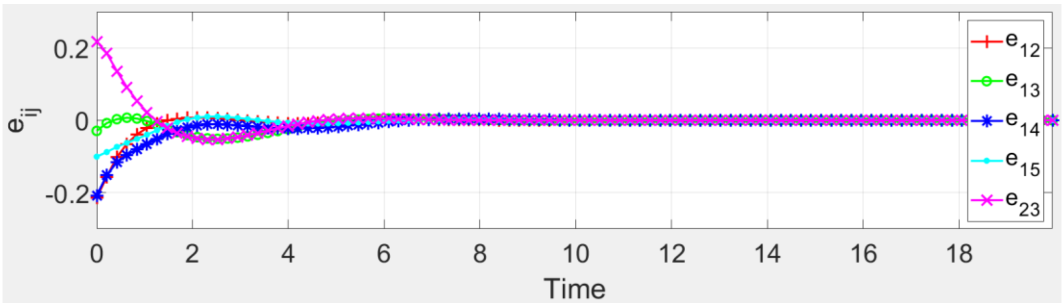
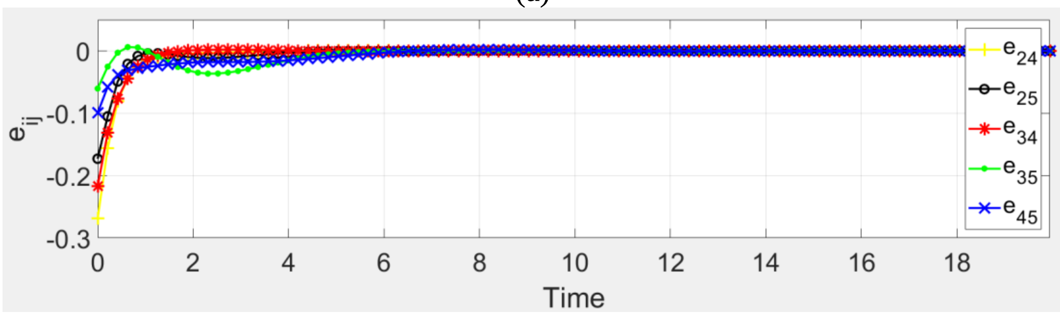


Figure 13 Snapshots of the wedge formation at different instants of time that shows MAS maintaining the desired formation while performing predefined trajectories starting from and arbitrary initial positions.

To show the formation maneuvering trajectory, snapshots are taken in different instants of time along the simulation time as depicted in figure (13). Based on the distance errors between all agents demonstrated in figure (14), it can be shown that the agents successfully achieved the required formation around the 5th second of simulation time with fair amount of distance errors. It is also shown that agents keep the desired inter-agent distances throughout the simulation. Figure (15) shows the control inputs of all agents in the formation.



(a)



(b)

Figure 14 Inter-agent distance error for all agent pairs (i, j) in $V^* \times V^*$.

Based on these three case studies we can demonstrate that the used formation control laws are exponentially stable around the equilibrium point at which the desired formation is achieved. Based on equations (26, 29, 30), the control gains k_v and k_a can control the speed of convergence of the distance errors and velocity errors and hence the convergence of augmented Lyapunov function (16) to zero. It is important to declare that it is a compromising situation, as the larger k_v and k_a is selected, the faster the convergence but on the other hand the larger the control input can go.

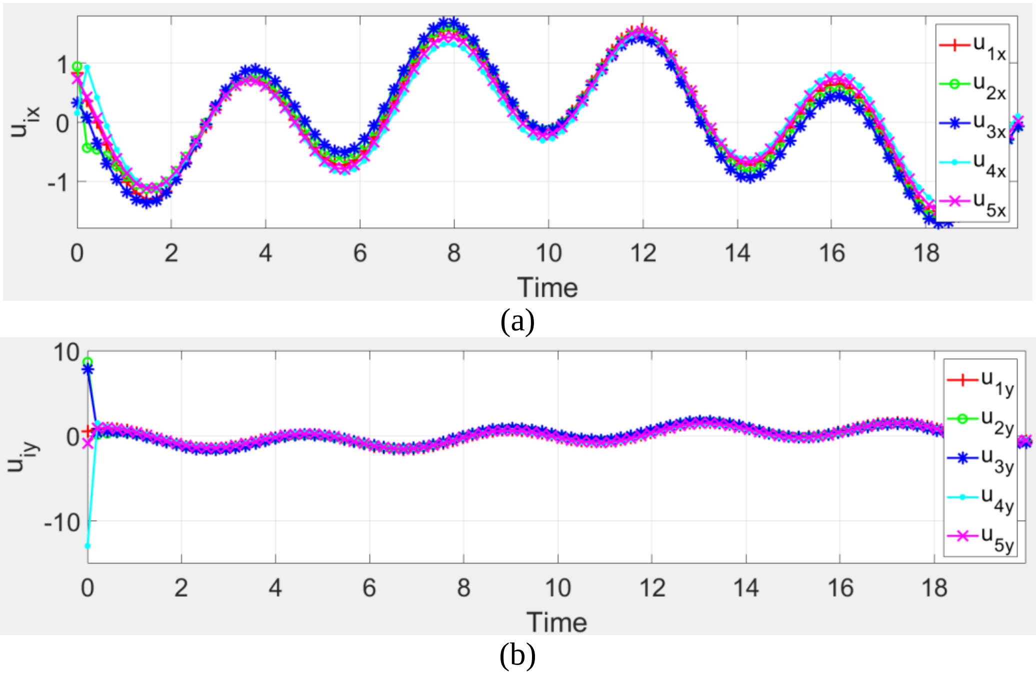


Figure 15 Formation maneuvering control inputs for each agent (a) for x axis (b) for y axis.

6. Conclusions

In this paper, a construction of distance-based formation control law has been shown which is applicable for solving formation acquisition and formation maneuvering problems of multi-agent systems based on double-integrator model. The used control laws were mainly constructed based on graph rigidity, specifically the properties of the infinitesimal and minimal rigidity of the graph that modeled the sensing and communication network topology in the multi-agent system. This is what guarantees that this formation method is asymptotically stable around the desired formation and implicitly ensures collision avoidance between agents in the system. The stability of the system is proved based on Lyapunov theory. For the double-integrator model, backstepping technique is used implicitly to construct the formation control law. This formation method provides a distributed control law for each agent in the system, provided that each agent can obtain the proportional position and velocity of its neighbors in the rigid graph and its own velocity. These required inputs can be obtained by using on board sensors or wireless communication with other agents in the network. One of the important limitations of the graph rigidity approach is that it requires satisfying some conditions to ensure not to converge to any ambiguous formation and only converge to isomorphic framework of the desired framework.

References

[1] V. Gazi and K. M. Passino, *Swarm Stability and Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. Doi: 10.1007/978-3-642-18041-5.

[2] G. Wen and W. X. Zheng, "On Constructing Multiple Lyapunov Functions for Tracking Control of Multiple Agents With Switching Topologies," *IEEE Trans Automat Contr*, vol. 64, no. 9, pp. 3796–3803, Sep. 2019, doi: 10.1109/TAC.2018.2885079.

[3] Y. Hua, X. Dong, Q. Li, and Z. Ren, "Distributed Fault-Tolerant Time-Varying Formation Control for Second-Order Multi-Agent Systems With Actuator Failures and Directed Topologies," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 6, pp. 774–778, Jun. 2018, doi: 10.1109/TCSII.2017.2748967.

[4] Brian D.O. Anderson, Changbin Yu, Baris Fidan, and Julien M. Hendrickx, "Rigid graph control architectures for autonomous formations," *IEEE Control Syst*, vol. 28, no. 6, pp. 48–63, Dec. 2008, doi: 10.1109/MCS.2008.929280.

[5] X. CAI and M. De Queiroz, "Formation maneuvering and target interception for multi-agent systems via rigid graphs," *Asian J Control*, vol. 17, no. 4, pp. 1174–1186, Jul. 2015, doi: 10.1002/asjc.1044.

[6] J. Zheng, M. Ding, L. Sun, and H. Liu, "Distributed Stochastic Algorithm Based on Enhanced Genetic Algorithm for Path Planning of Multi-UAV Cooperative Area Search," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8290–8303, Aug. 2023, doi: 10.1109/TITS.2023.3258482.

[7] K. Sakurama, "Formation Control of Mechanical Multi-agent Systems under Relative Measurements and its Application to Robotic Manipulators," in *2021 60th IEEE Conference on Decision and Control (CDC)*, IEEE, Dec. 2021, pp. 6445–6450. Doi: 10.1109/CDC45484.2021.9682816.

[8] K. K. Oh, M. C. Park, and H. S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, Mar. 2015, doi: 10.1016/j.automatica.2014.10.022.

[9] R. Olfati-Saber and R. M. Murray, "Graph rigidity and distributed formation stabilization of multi-vehicle systems," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.* IEEE, 2002, pp. 2965–2971. Doi: 10.1109/CDC.2002.1184307.

[10] J. P. Desai, J. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, IEEE, 1998, pp. 2864–2869. Doi: 10.1109/ROBOT.1998.680621.

[11] T. Eren, P. N. Belhumeur, and A. S. Morse, "Closing ranks in vehicle formations based on rigidity," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, IEEE, 2002, pp. 2959–2964. Doi: 10.1109/CDC.2002.1184306.

[12] R. Olfati-Saber and R. M. Murray, "Distributed cooperative control of multiple vehicle formations using structural potential functions," in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, IFAC Secretariat, 2002, pp. 495–500. Doi: 10.3182/20020721-6-es-1901.00244.

[13] J. Baillieul and A. Suri, "Information patterns and hedging brockett's theorem in controlling vehicle formations," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, IEEE, 2003, pp. 556–563. Doi: 10.1109/CDC.2003.1272622.

[14] J. A. Fax and R. M. Murray, "Information Flow and Cooperative Control of Vehicle Formations," *IEEE Trans Automat Contr*, vol. 49, no. 9, pp. 1465–1476, Sep. 2004, doi: 10.1109/TAC.2004.834433.

[15] J. M. Hendrickx *et al.*, "RIGIDITY AND PERSISTENCE FOR ENSURING SHAPE MAINTENANCE OF MULTI-AGENT META-FORMATIONS," *Asian J Control*, vol. 10, no. 2, pp. 131–143, 2008, doi: 10.1002/asjc.014.

[16] L. Krick, M. E. Broucke, and B. A. Francis, "Stabilisation of infinitesimally rigid formations of multi-robot networks," *Int J Control*, vol. 82, no. 3, pp. 423–439, Mar. 2009, doi: 10.1080/00207170802108441.

[17] K.-K. Oh and H.-S. Ahn, “Formation control of mobile agents based on inter-agent distance dynamics,” *Automatica*, vol. 47, no. 10, pp. 2306–2312, Oct. 2011, doi: 10.1016/j.automatica.2011.08.019.

[18] Xiaoyu CAI and M. de Queiroz, “Multi-agent formation maintenance and target tracking,” in *2013 American Control Conference*, IEEE, Jun. 2013, pp. 2521–2526. Doi: 10.1109/ACC.2013.6580213.

[19] X. CAI and M. de Queiroz, “Multi-agent formation maneuvering and target interception with double-integrator model,” in *2014 American Control Conference*, IEEE, Jun. 2014, pp. 287–292. Doi: 10.1109/ACC.2014.6858603.

[20] P. Zhang, M. De Queiroz, and X. Cai, “Three-Dimensional Dynamic Formation Control of Multi-Agent Systems Using Rigid Graphs,” *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 137, no. 11, Nov. 2015, doi: 10.1115/1.4030973.

[21] X. CAI and M. de Queiroz, “Rigidity-Based Stabilization of Multi-Agent Formations,” *J Dyn Syst Meas Control*, vol. 136, no. 1, Jan. 2014, doi: 10.1115/1.4025242.

[22] K. K. Oh and H. S. Ahn, “Distance-based undirected formations of single-integrator and double-integrator modeled agents in n-dimensional space,” *International Journal of Robust and Nonlinear Control*, vol. 24, no. 12, pp. 1809–1820, 2014, doi: 10.1002/rnc.2967.

[23] B. D. O. Anderson, M. Cao, S. Dasgupta, A. S. Morse, and C. Yu, “Maintaining a directed, triangular formation of mobile autonomous agents,” *Commun Inf Syst*, vol. 11, no. 1, pp. 1–16, 2011, doi: 10.4310/CIS.2011.v11.n1.a1.

[24] R. Babazadeh and R. Selmic, “Distance-Based Multiagent Formation Control with Energy Constraints Using SDRE,” *IEEE Trans Aerosp Electron Syst*, vol. 56, no. 1, pp. 41–56, Feb. 2020, doi: 10.1109/TAES.2019.2910361.

[25] M.-C. Park, Z. Sun, B. D. O. Anderson, and H.-S. Ahn, “Stability analysis on four agent tetrahedral formations,” in *53rd IEEE Conference on Decision and Control*, IEEE, Dec. 2014, pp. 631–636. Doi: 10.1109/CDC.2014.7039452.

[26] Z. Sun, S. Mou, B. D. O. Anderson, and A. S. Morse, “Non-robustness of gradient control for 3-D undirected formations with distance mismatch,” in *2013 Australian Control Conference*, IEEE, Nov. 2013, pp. 369–374. Doi: 10.1109/AUCC.2013.6697301.

[27] M. Khaledyan, T. Liu, V. Fernandez-Kim, and M. De Queiroz, “Flocking and Target Interception Control for Formations of Nonholonomic Kinematic Agents,” *IEEE Transactions on Control Systems Technology*, vol. 28, no. 4, pp. 1603–1610, Jul. 2020, doi: 10.1109/TCST.2019.2914994.

[28] Khalil Ibrahim, Ahmed A Ramadan, Mohamed Fanni, Yo Kobayashi, Ahmed A Abo Ismail, Masakatus G Fujie "Kinematic analysis and control of limited 4-DOF parallel manipulator based on screw theory" IEEE International Conference on

[29] B. Jackson, “Notes on the Rigidity of Graphs,” Oct. 2007. Accessed: Sep. 29, 2024. [Online]. Available: https://webspace.maths.qmul.ac.uk/b.jackson/levicoFINAL.pdf

[30] Izmestiev, I., 2009, “Infinitesimal Rigidity of Frameworks and Surfaces,” Lectures on Infinitesimal Rigidity, Kyushu University, Japan