

INDUSTRIAL ENGINEERING AND MANAGEMENT

Research Project

## Distributed Formation Control of Multi-Agent Systems Using Feature Measurements

Laurah KUIK S1880780 L.R.Kuik@student.rug.nl Supervisors: Prof. dr. ir. Bayu JAYAWARDHANA

Prof. dr. ir. Ming CAO

March 27, 2019

#### Abstract

A fundamental problem in formation control is the complexity and amount of data involved with the interactions necessary for achieving and maintaining a formation. As a result, costly hardware is required to support the data. To minimize this, cheap sensing methods should be used. Currently, a formation control strategy, called feature-based control, is studied at the University of Groningen. This formation control approach promises to be less expensive and more accessible than most other types of control strategies. The goal of this research is to design and validate a distributed formation controller using feature-based control that can be implemented on the NEXUS robots. In feature-based control, the agent senses two features of its neighbour agent(s) in order to control its linear velocity. In this thesis the features are visual markers placed on the left and right side of the agents. Each agent is equipped with a sensor that is able to detect the markers and measure the relative bearing of the vectors pointing from the sensor to the markers. In this research, stability analysis is provided to determine the performance of a formation controller that uses feature-based measurements.

## List of Symbols

- *B* incidence matrix
- ${\cal E} \qquad {\rm set \ of \ edges}$
- E desired formation
- e error
- F formation constraint
- $\mathcal{G}$  graph
- g relative bearing
- *I* identity matrix
- $k_p$  gain
- N number of agents
- $\mathcal{N}$  set of neighbours
- P orthogonal projection matrix
- p position
- t time
- u input
- v velocity
- $\mathcal{V}$  set of agents
- x state
- ||x|| Euclidean norm of x
- y measurement
- z output
- $z^*$  desired output
- $\hat{z}$  unit vector of z
- $z_{ijL}$  left bearing vector
- $z_{ijR}$  right bearing vector
- $\gamma$  potential function
- $\theta$  angle

## Table of Contents

Li	List of Symbols 2								
1	Introduction         1.1       Research Context         1.2       Research Design         1.2.1       Research Initiation and Stakeholder Analysis         1.2.2       Methodology         1.2.3       Research Goal         1.2.4       Research Questions         1.3       Thesis Outline	<b>5</b> 6 7 7 8 8 9							
2	Literature Review	10							
	<ul> <li>2.1 Graph Theory</li></ul>	$10\\10\\11\\12\\12\\13\\14\\15\\16$							
3	The Proposed Formation Control       1         3.1 The Experiments       1         3.2 The Desired Formation Shape       1	<b>L8</b> 19 19							
4	Simulation Setup24.1 The NEXUS Robot44.2 ROS - Robot Operating System44.3 The Laser44.4 The Camera44.5 AprilTags44.6 The Computation Graph44.7 Gazebo4	21 21 21 22 22 23 25							
5	onte Carlo Simulation       27         Initial Conditions       27         Gains for the Velocity Input       28								
6	Results       5.1       Successful Simulations       5.2         6.2       Convergence Analysis Experiment 1: distance-based and feature-based       5.2         6.2.1       Position and Velocity Convergence       5.2         6.2.2       Distance- and Feature Error Convergence       5.3         Convergence Analysis Experiment 2: distance-based, feature-based including extra unit vector control       5.3	29 29 31 31 33							

	6.3.1Position and Velocity Convergence6.3.2Distance- and Feature Error Convergence	$\frac{35}{37}$
7	Discussion	39
8	Conclusion	39
9	Limitations and Further Research	40

## 1 Introduction

Since the beginning of the 21st century, the production industry has been facing a fundamental change from a vendor's to a customer's market. The increasing surplus of industrial capacity offers the customer greater choice and raises the competition between vendors. Consequently, customers become less loyal to a brand and more demanding. As a result, constant innovation, low-cost customization, and better services are required for the industry to provide the customers what they need. To achieve this, companies must shorten product life cycles, reduce time to market, increase product diversity, reduce investment costs, rapidly satisfy demand, and so on (Bussmann et al., 2013).

Fortunately, the rapid development in information and communication technology enables the manufacturing industry to react to these changes. A key feature that contributes to this rapid development is smart manufacturing. The National Institute of Standards and Technology defines smart manufacturing systems as "fully integrated, collaborative systems that responds in real time to meet the changing demands and conditions in the factory and supply networks and in customer needs" (Kusiak, 2018) (Zheng et al., 2018). Smart manufacturing plays a key role in the shift to the fourth industrial revolution, Industry 4.0, which is marked by automation and data exchange in manufacturing technologies. For this shift to Industry 4.0, it is important that the machines and systems of smart manufacturers become self-aware and self-coordinating. By using sensors and communicating data the smart systems can adjust their behaviour and react to changes in the industry (Almada-Lobo, 2016).

An excellent example of an integrated, collaborative system is a multi-agent system that accomplishes a task through the collaboration of its agents. The agents in such a system can act autonomously and cannot sense a global reference frame. Moreover, the control of the agents is decentralized. One of the important fields in multi-agents systems is formation control, which concerns the coordinated control of the agents. It aims to achieve and maintain a specific desired shape and thereby allows the multi-agent system to perform complicated tasks (Han et al., 2015).

Formation control does not only play an essential role in the shift to Industry 4.0, it can be used for many other applications. For instance, in military and civilian deployment robots are used to perform tasks that would be too dangerous for humans. Moreover, other applications include surveillance, flight control system design, security patrols, mapping and localization, search and rescue, and many more. Additionally, multi-agent systems can save time and labor costs by replacing human laborers. Due to the practical potential in these various applications, formation control of multi-agent systems promises to be a convenient technique for solving complex problems (Guanghua et al., 2013). However, as with every technological innovation, this does not come without challenges. In order for the agents to work together, the sensing capabilities need to be improved. This will enable the agents to self-sense, self-act and communicate with one another. Moreover, the data obtained by the sensors is used for fast and accurate decision making. The main technological challenge, in this case, is that these interactions require an increase in data and complexity. As a result, costly hardware is needed to support the amount of data and complexity in control systems. To minimize this, cheap sensing methods should be used and data usage should be kept low (Zheng et al., 2018).

Recently, a formation control strategy, called bearing based control strategy has become more popular due to the fact that it is often cheaper and more accessible than other types of control strategies. In bearing-based control, a camera can be utilized to sense the inter-agent bearing measurements. This type of sensor is generally more affordable than other appropriate sensors such as a laser scanner. Currently, the University of Groningen is working on a similar technique called feature based formation control.

This research project will focus on this type of strategy. The main goal of this research is to design and validate a formation controller using feature measurements. In the featurebased approach, two features are placed on the agents. Furthermore, agents are equipped with a sensor that can detect the features of their neighbors and measure the relative bearing of the unit vectors from the sensor to the features. The purpose of the controller is to achieve the desired formation shape.

The remaining part of this chapter is arranged as follows. Firstly, the research context of multi-agent formation control will be discussed in Section 1.1. Then, the methodological choices for this thesis are considered in Section 1.2. Finally, the outline of the thesis is provided in Section 1.3.

## 1.1 Research Context

In this section, background information is provided on the research field of formation control.

As stated in the introduction, the rapid development in information and communication technology permits the manufacturing industry to become smarter, meaning that the manufacturing systems become more flexible and self-coordinating. Increasing the number of sensors in the equipment enables the machinery to self-sense, self-act and communicate with one another. Moreover, the sensors are used to obtain and share real-time production data to support fast and accurate decision making.

In a multi-agent system, multiple interacting agents solve problems that cannot be solved with solely the individual capabilities or knowledge of each agent. In Sycara (1998), multiagents systems are defined as systems where; (1) each agent has insufficient information or capabilities to solve the problem, (2) there is no system global control, (3) data is decentralized, and (4) computation is asynchronous. Formation control is one of the most popular topics in the field of multi-agent systems. The aim in formation control is to design a controller for each agent such that it achieves the desired constraints on their states.

Furthermore, there are different approaches to formation control. Table 2 shows the distinctions of the three most widely used approaches, categorized by Oh et al. (2015). The distinctions are based on the sensing capabilities and interaction topology of the agents.

- Position-based control: Agents sense their own positions relative to a global reference system. The control of the agents is based on their own positions and that of the desired with respect to the global reference frame.
- Displacement-based control: Agents sense the relative positions of their neighbors with respect to the global reference frame. The control of the agents is based on the desired displacement with respect to the global reference frame. For this approach agents need to know the orientation of the global reference frame. However, this does not require the agents to know their positions with respect to the global reference frame.
- Distance-based control: Agents sense the relative positions of their neighbors with respect to the local reference frame of the agents. The distance between the agents is controlled by the desired inter-agent distance to achieve the desired formation.

Tab.	2:	Distinctions	$\operatorname{among}$	position-,	disp	lacement-,	and	distance-	based	formation	control
		(Oh et al., $2$	015).								

	Position-based	Displacement-based	Distance-based
Sensed variables Controlled variables Coordinate systems	Positions of agents Positions of agents A global coordinate system	Relative positions of neighbors Relative positions of neighbors Orientation aligned local coordinate systems	Relative positions of neighbors Inter-agent distances Local coordinate systems
Interaction topology	Usually not required	Connectedness or existence of a spanning tree	Rigidity or persistence

Though the previous three approaches have been most dominant, bearing-based control is a promising approach in formation control. Bearing-based control uses inter-agent bearing measurements to control the formation of a multi-agent system. Furthermore, bearing-based formation is becoming more popular, since it can use visual sensing to obtain bearing information. Using a camera is often more accessible and cheaper than using the type of sensors appropriate for other formation approaches Zhao et al. (2016).

Similarly, feature-based control has the same benefits. However, instead of using one point mass for measuring the bearing between the neighbouring agents, two features can be placed on the agent. From these features a relative bearing of the vectors pointing from the sensor to the features is measured. These bearing measurement are used in the control law, which is further explained in Section 2.3.5. In a real life environment, two visual markers placed on each agent can be detected by a camera in order to control the agents.

## 1.2 Research Design

In order for this research to contribute to the knowledge base of formation control, a clear research design is formulated. In this section, the research initiation, methodology, goal and questions are defined.

## 1.2.1 Research Initiation and Stakeholder Analysis

The research concerning a new approach to formation control is initiated by prof. dr. ir. Bayu Jayawardhana. Chan et al. (2019) provided stability analysis on the feature-based formation control of a triangular formation. Thus far feature-based formation control has not been researched for more than three agents or in a real-life environment. Therefore, the main focus of this research will be on the stability of the feature-based control law for more than three agents in a real-life environment. Furthermore, stability analysis on the combination feature-based control with other possible formation approaches in order to achieve a formation has not been performed. Therefore, this research will focus on combining feature-based control with another formation approach as well. Moreover, Bayu Jayawardhana and Nelson Chan are the main stakeholders of this thesis, since the knowledge on feature-based control is extended by this thesis.

## 1.2.2 Methodology

During the Industrial Engineering and Management (IEM) program two research methods are advised. The first one is the Ontwerpen van Bedrijfskundige Systemen (OBS) method, which is most suitable for when solving a particular research problem in a specific setting such as a problem in one of the production lines of a particular company. The second method is the design science method, which is focused on solving more general problems. Since formation shape control is a general problem that can have many applications, the second method is applied for this thesis. Hevner (2007) established a design science framework focusing on three inherent research cycles: the relevance cycle, the design cycle and the rigor cycle. Figure 1 shows the research cycles and their relations.

The relevance cycle draws on the application context that specifies the requirements for the research as well as the acceptance criteria for the final evaluation of the research results. Therefore, the outcome from the design science research must be returned to the environment. In the evaluation the researcher reviews if the designed artifact improves the environment and how this improvement is quantified. In this thesis, the environment is represented by the possible applications of formation control discussed in Section 1. The rigor cycle provides past knowledge to the research project to assure its innovation. In this cycle the researcher must guarantee that the designed artifact contributes to the knowledge base by thoroughly researching and referencing this knowledge base. When new knowledge is generated in the design research this will be provided to the knowledge base. For instance, the mathematical operations applied for designing a feature-based control law are provided by the knowledge base. Validating such a control law would contribute to this knowledge base. The central design cycle iterates between building and evaluating the design artifacts and process of the design science research. The designed artifacts in this research are the feature-based control law and the hardware and software designed to validate this formation control law.



Fig. 1: The three design science research cycles: relevance cycle, design cycle, and rigor cycle.

## 1.2.3 Research Goal

In order to contribute to the knowledge base of formation control, the following research goal is formulated:

Design and validate a distributed formation controller using feature measurements that can be implemented on the NEXUS robots.

## 1.2.4 Research Questions

In order to design and validate a formation controller using feature measurements the following research questions need to be answered.

- 1) How are the formations defined when using features?
- 2) How can feature measurements be used to design a formation controller?
- 3) What type of formation does the formation controller needs to achieve?

If a formation controller is designed for a specific type of formation, the following questions need to be answered:

- 4) How is the stability of the controller?
- 5) What is the performance of the controller in a real-life simulation?

## 1.3 Thesis Outline

To answer the research questions, this thesis is arranged as follows. In Section 2, a literature review is performed to answer the first two research questions. In Section 3 question 3 is answered by proposing a formation controller based on the theory from the previous section. Section 4 discusses the simulation used to validate the formation controller. Section 5 discusses the Monte Carlo simulations performed in order to answer research questions 4 and 5. Then, the results of the simulations are provided in Section 6. Finally, in Section 7 the conclusion and discussion is provided.

## 2 Literature Review

In this section, the theoretical background of this thesis is provided. First, graph theory is introduced. Then, graph rigidity is discussed. Finally, a review of formation control approaches is provided.

## 2.1 Graph Theory

In a multi-agent system the relationships are often modeled by a graph  $\mathcal{G}$ . The agents are represented as nodes and the interactions such as sensing and communication are represented as edges.  $\mathcal{V} = \{1, 2, ..., N\}$  denotes the set of agents and  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  denotes the set of edges between agents. The set of neighbors of agent *i* is defined by  $\mathcal{N}_i := \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ . Another description of the incidence matrix  $B \in \mathbb{R}^{|\mathcal{V}| \times \mathcal{E}}$ , which is defined by:

$$b_{ik} = \begin{cases} -1 & \text{if } i = \mathcal{E}_k^{\text{tail}} \\ 1 & \text{if } i = \mathcal{E}_k^{\text{head}} \\ 0 & \text{otherwise,} \end{cases}$$
(1)

where  $\mathcal{E}_k^{\text{head}}$  and  $\mathcal{E}_k^{\text{tail}}$  represent the head and tail node of  $\mathcal{E}_k$ , whose direction can be arbitrarily selected under a undirected graph.

## 2.2 Graph Rigidity

Rigid graph theory refers to properties of graphs that ensure that the formation modeled by the graph is rigid. Generally speaking, a formation is rigid if the only way to move is by translation or rotation of the formation as a whole. Therefore, the rigidity of a formation is an important property to take into account when designing a formation controller. There are three types of rigid frameworks: a rigid framework, a minimally rigid framework and infinitesimally rigid framework. The latter is most relevant for formation control, since it concerns the minimal number of edges necessary to keep the formation uniquely determined up to translation, rotation and/or scaling. Furthermore, the possibility for infinitesimal rigid formation to rotate, translate or scale depends on the type of formation control. For instance, in distance-based control, the infinitesimally rigid formation is determined up to translation and rotation. For bearing-based control, the infinitesimally bearing rigid formation is determined up to translation and scaling. Moreover, the number of edges necessary for an infinitesimal rigid formation can be different for each type of formation control. Figure 2 illustrates examples of infinitesimally rigid frameworks for both distance- and bearing-based formation, the only difference is that with bearing-based formation the edges are not restricted to a certain length. For the mathematical definition of graph rigidity, please refer to Oh et al. (2015) and Zhao et al. (2016).



Fig. 2: Examples of infinitesimally bearing rigid frameworks (Zhao et al., 2016).

## 2.3 Formation Control Approaches

A general description for N-agents is as follows:

$$\begin{cases} \dot{x}_i = f_i(x_i, u_i), \\ y_i = g_i(x_1, \dots, x_N), \quad i = 1, \dots, N, \\ z_i = h_i(x_i), \end{cases}$$
(2)

where  $x_i \in \mathbb{R}^{n_i}$ ,  $u_i \in \mathbb{R}^{p_i}$ ,  $y_i \in \mathbb{R}^{q_i}$ , and  $z_i \in \mathbb{R}^r$  represent the state, input, measurement, and output of agent *i*. Furthermore,  $f_i : \mathbb{R}^{n_i} \times \mathbb{R}^{p_i} \to \mathbb{R}^{n_i}$ ,  $g_i : \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_N} \to \mathbb{R}^{q_i}$ , and  $h_i : \mathbb{R}^{n_i} \to \mathbb{R}^r$ . A desired output  $z^* \in \mathbb{R}^{r_N}$  is given, which can be a function of time. Let  $F : \mathbb{R}^{n_N} \to \mathbb{R}^M$  be given. Then the desired formation is defined by  $z^*$  and can be described by:

$$F(z^*) = \xi^*,\tag{3}$$

where  $\xi^*$  are the defining desired constraint values. In the following subsections, the state x and output z represent the position of the agents. Furthermore, the formation control problem is as follows:

Design a control law in which only measurements  $y_i$  are used such that the set:

$$E_{\xi^*} = \{x | F(z) = \xi^*\}$$

is asymptotically stable with respect to (2).

In the following approaches, the single-integrator dynamics for agent i in n-dimensional space is as follows:

$$\dot{p}_i(t) = u_i(t), \quad i = 1, ..., N,$$
(4)

where  $p_i \in \mathbb{R}^n$  and  $u_i \in \mathbb{R}^n$  represent the position and control input of agent *i* with respect to a global coordinate system. Furthermore, the dynamics with respect to a local coordinate system can be written as:

$$\dot{p}_{i}^{i}(t) = u_{i}^{i}(t), \quad i = 1, ..., N,$$
(5)

where  $p_i^i \in \mathbb{R}^n$  and  $u_i^i \in \mathbb{R}^n$  represent the position and control input of agent *i* with respect to a local coordinate system. Furthermore, the output  $z_i$  with respect to the global coordinate system is  $p_i$ . The output  $z_i$  with respect to the local system is  $p_i^i$ . Moreover, the measurements  $y_i$  differ per formation control.

## 2.3.1 Position-based Formation Control

In position-based formation control, the measurements for N-agents are as follows:

$$y_i = p_i, \quad i = 1, \dots, N,\tag{6}$$

Furthermore, the constraint (3) is given directly by the desired output  $z^*$ , i.e.:

$$F(z^*) = z^*.$$

The measurements  $y_i$  are the position coordinates that are sensed with respect to a global coordinate frame. Furthermore, agents *i* actively control *z*, which is the position of the agents. The desired formation is defined as follows:

$$E_{p^*} := \{ p | p_i = p_i^*, \quad i \in \mathcal{V} \}.$$
(7)

The general formation control law for the position-based formation control is:

$$u_i = k_p (p_i^* - p_i), \tag{8}$$

where  $k_p > 0$  and  $p_i^*$  is the desired position.

## 2.3.2 Displacement-based Formation Control

In displacement-based formation control, the measurements for N-agents are as follows:

$$y_i = [\dots (p_j - p_i)^T \dots]^T, \quad (i, j) \in \mathcal{E},$$
(9)

Furthermore, the constraint (3) is given as:

$$F(z^*) = [\dots (z_j - z_i)^T \dots]^T, (i, j) \in \mathcal{E}$$

which is invariant to translation applied to z. The measurements  $y_i$  contain relative values that are sensed with respect to a global coordinate frame. Furthermore, agents *i* actively control  $[\ldots (z_j - z_i)^T \ldots]^T$ . Figure 3(a) illustrates the coordinate system in the displacementbased setup. It is assumed that the agents sense the relative positions of their neighbors with respect to the global coordinate system, which is defined as:

$$p_{ij} := p_j - p_i, \quad j \in \mathcal{N}_i. \tag{10}$$

As  $p^* \in \mathbb{R}^{nN}$  is given, the objective of the agents is to satisfy the following constraints:

$$p_i - p_j = p_i^* - p_j^*, \quad (i, j) \in \mathcal{E}.$$
 (11)

In this case  $p_i^*$  and  $p_j^*$  are not the desired positions of agent *i* and *j*,  $p_i^* - p_j^*$  only specifies the desired displacement between the two agents. The desired formation is defined as follows:

$$E_{p^*} := \{ p | p_j - p_i = p_i^* - p_j^*, \quad (i, j) \in \mathcal{E} \}.$$
(12)

A widely used displacement-based formation control law is as follows:

$$u_i = k_p \sum_{j \in \mathcal{N}_i} e_{ij},\tag{13}$$

where  $k_p > 0$  and the error signal is:

$$e_{ij} = p_j - p_i - p_j^* + p_i^*, (14)$$

where  $p_j^* - p_i^*$  is the desired displacement.



Fig. 3: Formation control problem setups for displacement- and distance-based control (Oh et al., 2015).

## 2.3.3 Distance-based Formation Control

In distance-based formation control, the measurements for N-agents are as follows:

$$y_i = [\dots || p_j^i - p_i^i || \dots ]^T, \quad (i, j) \in \mathcal{E},$$

$$(15)$$

The measurements  $y_i$  are relative variables that are sensed with respect to the local coordinates of the agents. Figure 3(b) illustrates the coordinate system in the distance-based setup. The constraint (3) in this approach is given as:

$$F(z^*) = [\dots ||z_j - z_i|| \dots ]^T, \quad (i, j) \in \mathcal{E},$$

and is invariant to translation and rotation applied to z. Agents actively control  $[\ldots, ||z_j - z_i||, \ldots]^T$  in the distance-based formation problem. The agents in this case have their own local coordinate system and this does not have to be aligned with their neighbour agents. The sensed variables of agent *i* are:

$$p_{ji}^i := p_j^i - p_i^i \equiv p_j^i, \quad j \in \mathcal{N}_i, \tag{16}$$

where  $p_j^i$  represents the position of agent j with respect to the local coordinate system of agent i. With  $p^* \in \mathbb{R}^{rN}$  given, the desired formation for the agents is defined as:

$$E_{p^*} := \{ p \in \mathbb{R}^{rN} | \quad ||p_j - p_i|| = ||p_i^* - p_j^*||, \quad (i, j) \in \mathcal{E} \}.$$
(17)

Gradient based methods have been widely used for distance-based formation control, where the potential function is defined as:

$$\phi_i(p_i^i, \dots, p_j^i, \dots) := \frac{k_p}{2} \sum_{j \in \mathcal{N}_i} \gamma_{ij}(||p_j^i - p_i^i||),$$
(18)

where  $k_p > 0$  and  $\gamma_{ij}: \mathbb{R} \to \overline{\mathbb{R}}_+$  is differentiable. A common potential function is:

$$\gamma_{ij}(||p_j - p_i||) := k_p(||p_j - p_i||^2 - ||p_j^* - p_i^*||^2)^2.$$
(19)

This gives the following control input:

$$u_{i} = -k_{p} \sum_{j \in \mathcal{N}_{i}} (||p_{j} - p_{i}||^{2} - ||p_{j}^{*} - p_{i}^{*}||^{2}) \frac{p_{j} - p_{i}}{||p_{j} - p_{i}||},$$
(20)

where  $k_p > 0$  and  $||p_j^* - p_i^*||$  is the desired distance.

## 2.3.4 Bearing-based Formation Control

Though position, displacement, and distance-based formation control have been dominantly used, bearing-based formation control can be used for multi-agent formation control. Zhao et al. (2016) designed a distributed bearing-only control law for infinitesimally bearing rigid formations. Figure 4 shows the geometric interpretation of the bearing-based formation control law.



Fig. 4: The bearing-based setup for agent i sensing agent j (Zhao et al., 2016).

The unit vector that represents the relative bearing is:

$$g_{ij} = \frac{p_j - p_i}{||p_j - p_i||}, \quad j \in \mathcal{N}_i,$$

$$(21)$$

Note that  $g_{ij} = -g_{ji}$ . In bearing-only formation control with a global reference frame, the measurements for N-agents are as follows:

$$y_i = [\dots g_{ij} \dots]^T, \quad (i,j) \in \mathcal{E},$$
(22)

The constraint (3) is given as:

$$F(z^*) = [\dots g_{ij} \dots]^T, \quad (i,j) \in \mathcal{E},$$

which is invariant to translation and scaling applied to z. The measurements  $y_i$  depict the bearings that are sensed with respect to the global frame of the agents. Moreover, agents actively control z. The desired formation is defined as follows:

$$E_{p^*} := \{ p | \quad g_{ij} = g_{ij}^*, \quad i, j \in \mathcal{V} \}.$$
(23)

For the control law, Zhao and Zelazo use an orthogonal projection operator, P. For any nonzero vector  $x \in \mathbb{R}^n (n \ge 2)$ , define the operator  $P : \mathbb{R}^n \to \mathbb{R}^{n \times n}$  as:

$$P(x) = I_n - \frac{x}{||x||} \frac{x^T}{||x||},$$
(24)

which can be denoted at  $P_x$ . This operator is an orthogonal projection matrix which geometrically projects any vector onto the orthogonal compliment of x. The bearing-only control law is defined as:

$$u_i = -\sum_{j \in \mathcal{N}_i} P_{g_{ij}} g_{ij}^*, \quad j \in \mathcal{N}_i,$$
(25)

where  $g_{ij}^*$  is the desired bearing and  $P_{g_{ij}} = I_d - g_{ij}g_{ij}^T$ .

For bearing-only formation control without a global reference frame, please refer to Zhao et al. (2016).

## 2.3.5 Feature-based Formation Control

In feature-based formation control, the desired formation is based on the angle between the feature measurement of the agent. Figure 5 illustrates the formation in the case of two agents with the same orientation. The blue points at the sides of the robot are the features and the green points in the middle of the robots is where the sensors are.



Fig. 5: Feature-based measurements of agent 1 detecting the features of agent 2, both having the same orientation.

The positions of the agents are described by the vector  $p_i^i \in \mathbb{R}^2, i = 1, 2$ . Furthermore, there are two features assigned to each agent, a left and right one. The positions of the features are:

$$p_{iL} = p_i - a;$$
  $p_{iR} = p_i + a, \quad i = 1, 2,$  (26)

in which the constant vector a denotes the relative displacement to the center  $(p_i)$  of the agent and  $p_{iL}$  and  $p_{iR}$  represent the position of the left and right feature. Since vector a is constant,  $\dot{p}_i(t) = \dot{p}_{iL}(t) = \dot{p}_{iR}(t)$ . Each agent is able to detect the feature of the other agent with a sensor that is mounted at the center of the agent. Subsequently, it can measure the relative bearing between the agent and the two features of the other agents. In the case where the Field Of View (FOV) of the sensor is set to 360°, the sensor can detect features and measure relative bearing from all directions. The bearing  $(\theta_{ij})$  is determined by the unit bearing vectors  $(\hat{z}_{ijL} \text{ and } \hat{z}_{ijR})$  between agent i and the features of agent j. The unit bearing vectors are written as follows:

$$\hat{z}_{ijL} := \frac{p_{jL} - p_i}{||p_{jL} - p_i||} = \frac{z_{ijL}}{||z_{ijL}||}; \qquad \hat{z}_{ijR} := \frac{p_{jR} - p_i}{||p_{jR} - p_i||} = \frac{z_{ijR}}{||z_{ijR}||}.$$
(27)

Note that  $z_{ijL}andz_{ijR}$  are the same as the output variable z. Furthermore, the bearing is obtained as:

$$\theta_{ij} = \cos^{-1}(\hat{z}_{ijL} \cdot \hat{z}_{ijR}). \tag{28}$$

In feature-based formation control, the description for N-agents is as follows:

$$y_i = [\dots \theta_{ij} \dots]^T, \quad (i,j) \in \mathcal{E},$$
(29)

The constraint (3) is given as:

$$F(z^*) = [\dots \theta_{ij} \dots]^T, \quad (i,j) \in \mathcal{E},$$

which is invariant to translation and rotation applied to z. The measurements  $y_i$  are the angles between the unit bearing vectors. Furthermore, agents actively control z. The desired formation is defined as follows:

$$E_{p^*} := \{ p | \quad \theta_{ij} = \theta^*_{ij}, \quad (i,j) \in \mathcal{E} \}.$$

$$(30)$$

The feature-based formation control law is as follows:

$$u_i(t) = k_p(\hat{z}_{ijL}(t) + \hat{z}_{ijR}(t))e(t),$$
(31)

where  $k_p > 0$  and in which the error signal is:

$$e(t) = \cos\theta(t) - \cos\theta^*. \tag{32}$$

$$\theta(t) < \theta^* \Longrightarrow e(t) > 0 \tag{33}$$

$$\theta(t) > \theta^* \Longrightarrow e(t) < 0 \tag{34}$$

## 2.4 Performance Properties for Controller Evaluation

In order to evaluate the different control laws, certain properties of the response to the input signal can be used to quantify the performance of the control law. Aström and Murray (2010) illustrate the following key properties of a signal in Figure 6: the rise time  $T_r$ , the overshoot  $M_p$ , the settling time  $T_s$ , and the steady state value  $y_{ss}$ .  $T_r$  is the time it takes for the signal to go from 10 % tot 90 % of its final value.  $M_p$  is the percentage of the final value by which the signal initially shoots over the final value.  $T_s$  is the times it takes for the signal to stay within 2 % of its final value. In the case that the output converges,  $y_{ss}$  is the final level of the output.



Fig. 6: The key performance properties of the response signal: rise time, overshoot, settling time, and steady-state value.

## 3 The Proposed Formation Control

To design a distributed formation controller using feature-based control that can be implemented on the NEXUS robots, first the requirements of the formation need to be defined.

As it would be interesting to test the formation controller on the real NEXUS robots in the future, the number of agents depends on the maximum available robots at the DTPA-lab. Currently, the number of agents available for testing the formation is four. Therefore, the formation will consist of four agents.

Furthermore, the formation will be tested with a camera as the feature-based sensor. Since most cameras have a limited field of view (FOV), all the agents should be oriented in such a way that they are at least within each others FOV. The most convenient formation shape would be a rhombus with every agent oriented towards the centroid of the rhombus.

Moreover, since the formation has to be distributed, the control law used per agent has to be distance-, bearing-, or feature-based.

Since the performance of feature-based control is the main focus of this research, at least one agent should use the feature-based control law.

Furthermore, it is desired that the formation in its stable state can only move translational or rotational. In this way, the distance between the agents should converge to a stable state. Consequently, this formation controller does not use the bearing-only formation control law. This results in two possible formation control approaches: distance- and feature-based.

Considering the aim of analyzing the effect of combining two different approaches, every agent will have at least one neighbour agents that employs a different formation control law.

Another relevant requirement for the formation is that in a real life situation cheap sensing methods can be used. For distance-based approaches sensors are generally more expensive than the sensors required for feature-based approach. Consequently, it is most convenient if the number of agents using distance-based control is minimal. Thus, only one agent will use the distance-based control law. Furthermore, to fulfill the previous requirement, the other three agents should be neighbours of the distance-based agent. For a rhombus shaped formation, as in Figure 7, this means that either agent 1 or agent 3 has to be controlled using the distance-based control law.



Fig. 7: A four agent formation in which the dotted line represent the edges and  $\alpha, \beta$  and  $\gamma$  denote angles.

To sum up, the formation has to fulfill the following requirements:

- The formation shape is a rhombus.
- The formation consists of four agents and five edges.
- One agent that has three neighbours should use distance-based control.
- The other three agents should use feature-based control.
- All agents should be oriented in such a way that they are faced towards the centroid of the rhombus in the desired formation shape.

## 3.1 The Experiments

It is expected that the formation will have trouble converging to the desired shape. In the 3 agent formation control in Chan et al. (2019), the feature-based control law, as in Equation (31), did not reach the exact shape. Adding an extra constraint, however, did result in the 3 agents achieving the desired shape. Therefore, two experiments are proposed. The first experiment satisfies the aforementioned requirements. The second experiment does so too, except there is an extra control for the feature-based agents. The unit vector of the left marker  $(\hat{z}_{ijL})$  will be used as an additional constraint. The control law for agent 2,3 and 4 in this case is:

$$\dot{p}_i(t) = v_i(t) = k_p(\hat{z}_{ijL}(t) + \hat{z}_{ijR}(t))e(t) + (\hat{z}_{ijL} - \hat{z}^*_{ijL}).$$
(35)

## 3.2 The Desired Formation Shape

Figure 8 shows the desired formation that is based on the previously stated requirements. The formation has a rhombus shape with sides of 2 meter, two angles of  $100^{\circ}$ , and two angles of  $80^{\circ}$ .



Fig. 8: The desired formation shape: a rhombus with sides of 2 meter, two angles of 100° and two angles of 80°.

Furthermore, the orientations of angle  $\delta$  with respect to the *x*-axis of the global reference frame are:  $\delta_1 = 0^\circ, \delta_2 = 270^\circ, \delta_3 = 180^\circ$ , and  $\delta_4 = 90^\circ$ . Moreover, the incidence matrix for this formation is:

$$B = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & -1 & 1 \\ 0 & 0 & -1 & 0 & -1 \end{bmatrix}$$
(36)

In this case the desired distances (m) for agent 1 are as follows:

$$||p_2^* - p_1^*|| = 2, ||p_3^* - p_1^*|| = 2.57, ||p_4^* - p_1^*|| = 2.$$

The distance from the markers to the center of the agents is 18 cm. Thus, the desired features (rad) are:

$$\begin{split} \theta_{21} &= 0.11612184, \theta_{23} = 0.11612184, \theta_{31} = 0.13978707, \theta_{32} = 0.13812878, \\ \theta_{34} &= 0.13812878, \theta_{41} = 0.11612184, \theta_{43} = 0.11612184. \end{split}$$

## 4 Simulation Setup

In this section the hardware and software necessary to validate the proposed formation and its control approaches are discussed. The hardware represented is based on the equipment available in the DTPA-lab of the University of Groningen and on the available software packages that can represent the hardware in a simulation environment.

## 4.1 The NEXUS Robot

The formation control setup is tested on four NEXUS robots. Figure 9<sup>1</sup> shows the NEXUS robot without the sensors. In the simulation, the robots are equipped with different sensors. Three robots are equipped with a camera, and one is equipped with a laser scanner. Furthermore, all robots are equipped with mecanum wheels that allow them to move omnidirectional. This means that they can move forwards, backwards, sideways and rotate around their z-axes.



Fig. 9: The NEXUS robot with 4 wheel drive Mecanum wheels.

## 4.2 ROS - Robot Operating System

ROS is an open-source, meta-operating system, that can be used to control a large variety of robotic components from a computer in real world and simulated environments. A ROS system is a peer-to-peer network of individual elements that communicate with each other, creating a distributed environment, called the Computation Graph. This communication is performed by different nodes (ROS nodes) publishing or subscribing to a specific topic (ROS topics). Moreover, these ROS nodes can execute the necessary computations to process the data received by sensors and to determine the control input needed to achieve the desired formation. The data send and received over the different topics are called ROS messages, which are simple data structures such as strings, floats, integers and booleans.

## 4.3 The Laser

To measure the local distances of agent 1 and its neighbours, a Hokuyo laser scanner is used. This laser scan measures the distance of a target by illuminating the object with pulsed laser light and measuring the reflected pulses with a sensor. In ROS, a Hokuyo laser plugin can be easily included in the robot model. Moreover, the plugin is compatible with standard ROS messages and service calls. The laser is placed in the middle on the upper surface of the robot. For the laser to detect the other robots, the neighbours have a cylindrical object placed on the same height. The laser has the same orientation as the NEXUS robot and has an angle

<sup>&</sup>lt;sup>1</sup> source: https://ozrobotics.com/shop/mecanum-wheel-mobile-arduino-robotics-car/

range of  $[-90^\circ, 90^\circ]$ . Within that range 720 samples are taken 40 times per second, thus there is a sample every 0.25°. Furthermore, the distance range of the laser is [0.10, 30.0] meter.

## 4.4 The Camera

Three of the four agents in the simulation will use the feature-based control law. A camera can be used to measure the features of the neighbours. Figure 10 illustrates the projection point  $\mathbf{p} = [x, y, z]^T \in \mathbb{R}^2$  in the Cartesian space, which is projected on the image plane as point feature  $\tilde{\mathbf{p}} = [\tilde{x}, \tilde{y}]^T \in \mathbb{R}^2$ . The camera senses the point features of the left and right markers ( $\tilde{\mathbf{p}}_l$  and  $\tilde{\mathbf{p}}_r$ ) of its neighbours. If the camera is placed in the middle of the robots and if  $\tilde{\mathbf{p}}_l$  and  $\tilde{\mathbf{p}}_r$  are sensed with 100% accuracy, then the angle between the unit bearing vectors of the point features is exactly the same as the angle between the unit bearing vectors of  $\mathbf{p}_l$  and  $\mathbf{p}_r$ . Therefore, by sensing the image coordinates of these markers the feature can be determined and the feature-based control law can be applied.



Fig. 10: Central camera projection model with the projection point and its image coordinate (Ike, 2018).

The minimal required field of view (FOV) for the camera to detect all the markers of its neighbours in the desired formation shape is 110°. A suitable camera in this case is the Genius WideCam F100, a 12 MP webcam with a FOV of 120°, allowing recording 1080p full HD video while maintaining 30 frames per second.

## 4.5 AprilTags

There are two methods for detecting the markers of the robots using a camera. The first is to detect naturally-occurring features in the image plane, such as colours or shapes. The second is to use artificial features (fiducials), which are artificial landmarks designed to be easily recognized and distinguished from each other. Though the use of naturally-occurring features is a central focus of machine perception, visual fiducials are more favorable when simplifying the development of systems where perception is not the central objective. Visual fiducials are designed to have a small information payload and to be automatically detected and localized even with poor conditions. For instance, when the fiducial is oddly rotated, unevenly lit, at very low resolution, or tucked away in the corner of an otherwise cluttered image. Two of the earliest detection systems are the ARToolkit and the ARTag system, which are related to other 2-D barcode systems such as QR codes, however with a reduced information payload, allowing them to be detected from longer ranges and more robustly. Another fiducial marker is the AprilTag. Based on ARToolkit, the University of Michigan developed the Apriltags, a visual fiducial system that outperforms its predecessors in terms of detection rates and accuracy (Olson, 2011). Figure 11 shows the NEXUS robot with the AprilTags and cylindrical object in the simulation world.



Fig. 11: A NEXUS robot with two AprilTags placed at the positions of the left and right feature.

## 4.6 The Computation Graph

Figure 12 and 13 represent the computation graph of the controllers for agent 1 and 4 in Gazebo. The laser node for agent 1 (/laser\_node1) controls its velocity (/n\_1/cmd\_vel) after obtaining information on the distance by the laser scan topic (/n\_1/scan). The camera node for agent 4 (/camera\_node4) controls its velocity (/n\_1/cmd\_vel) after obtaining the image from the camera (/n\_4/camera/image\_raw/compressed) and information of the AprilTag detection node (/n1/scan). The model state node (/gazebo/model\_states) gives the agent information on the absolute position of the agent (for analysis purposes). Moreover, for each feature-based agent, the compressed image and camera info topics are send from the Gazebo node to the AprilTag detection nodes (/apriltags).



Fig. 12: The computation graph of the simulation in Gazebo with the distance-based control of agent 1 activated. The round shaped boxes depict the ROS node and the square boxes represent the ROS topics.



Fig. 13: The computation graph of the simulation in Gazebo with the feature-based control of agent 4 activated. The round shaped boxes depict the ROS node and the square boxes represent the ROS topics.

## 4.7 Gazebo

To examine the performance of the formation in a real life simulation, the 3-D dynamic simulator Gazebo is used. This software package simulates robots, sensors and objects in a 3-D dynamic environment, with realistic sensor feedback and physical interactions between objects. Furthermore, it is compatible with ROS, making is possible for the controller in ROS to directly communicate with the simulated robot. Hence, the control laws used in the simulation can easily be implemented on the real NEXUS robots. Figure 14 depicts the simulation setup in the desired formation in Gazebo. In Figure 15, the camera image of agent 2 is provided, it detects the AprilTags of agent 1 and 3. Agent 4 is not a neighbour of agent 2 and is therefore not recognized.



Fig. 14: The rhombus shaped formation with 4 agents om the Gazebo environment.



Fig. 15: The camera image from agent 2 with the tag detection software detecting AprilTags with IDs 0, 1, 4, and 5.

## 5 Monte Carlo Simulation

To analyze the performance in both experiments, Monte Carlo simulation is performed.<sup>2</sup> In order to perform the Monte Carlo simulation on the formation, the set of random initial conditions needs to be generated first. The initial conditions will be the starting positions of the NEXUS robots. Next, the simulations with all the initial conditions need to be performed. Finally, when all simulations are completed the convergence of the system will be examined.

## 5.1 Initial Conditions

The global positions of the 3 agents with a camera will be randomly distributed over an area of 4 by 3 meter, with the intervals:

$$-2 \le x \le 2 \tag{37}$$

$$0.4 \le y \le 3.4 \tag{38}$$

Agent 1 will have the same starting position every time, which is the coordinate [0.0] in the global coordination frame of the simulation world. To avoid collision with agent 1, the other agents should at least be 0.4 meter at distance in both the x and y direction. Since the laser can not distinguish the objects that it senses, the initial conditions of the other agents should be distributed in such a way that agent 2 is always sensed at the right side, agent 4 at the left side and agent 3 in the middle. In this way, the controller will assign the distance of the left object to agent 2 and so on. Furthermore, agent 2 should be positioned in such a way that it is able to detect the AprilTags of agent 1. For agent 2 this means that the area for its initial conditions becomes:

$$-2 \le x_2 \le -0.4,$$
 (39)

$$0.4 \le y_2 \le 3.4. \tag{40}$$

The interval for the initial conditions of agent 3 is:

$$-2 \le x_3 \le 2,\tag{41}$$

$$0.4 \le y_3 \le 3.4.$$
 (42)

Agent 4 has to be on the right side (positive x-axis) of agent 1 in order to detect the AprilTags of agent 1. Therefore, the interval for the initial conditions of agent 4 is:

$$0.4 \le x_4 \le 2 \tag{43}$$

$$0.4 \le y_4 \le 3.4.$$
 (44)

The sets of all possible x and y coordinates within the interval are generated with steps of 0.1 meter. This results in agents 2 and 4 having 527 possible positions, and agent 3 having 1271 possible positions. Consequently, the total set of position combinations for the agents N= 3,529,935,59. Because this is a rather large data set, a distributed sample of N = 150,000is taken. In this sample of initial conditions there is a high number of cases in which the agents are not in each others view or are colliding. Therefore, the sample set is filtered given the following criteria:

 $<sup>^{2}</sup>$  Monte Carlo simulation is named after the city of Monte Carlo in Monaco, which is known for its gambling establishments. As gambling games involve random behaviour, performing statistical analysis on a large number of simulations is called Monte Carlo simulation.

- Agent 3 should be on the left side of agent 2 (negative x-axis local frame agent 2) and with minimal distance of 0.4 meter (in both x and y direction) from agent 2.
- Agent 4 should be on the left side of agent 3 (negative x-axis local frame agent 3) and with minimal distance of 0.4 meter (in both x and y direction) from agent 3.

With this filtered sample set of N = 15,895, it is assured that the agents are not colliding at the start of the simulation. Furthermore, the changes of the AprilTags being detected by the cameras are increased. Too decrease the total simulation time of the Monte Carlo simulations, only the first 5,000 set of initial conditions are used from the filtered sample set.

## 5.2 Gains for the Velocity Input

The gains for the control laws of the velocity input in the two experiments was determined by checking the velocity of the robots at a small number of simulations. To ensure that the robots are not moving too fast, the norm of the velocity (u) should be between 0 and 0.8 m/s. In the small number of simulations the following gains showed acceptable velocity inputs. For both experiments the gain for the agent using distance-based control (agent 1), the gain is 0.1. For the first experiment, the gain of the agents using feature-based control (agent 2,3, and 4) is 15. In the second experiment, the gain of agents 2,3 and 4 is 1.

## 6 Results

## 6.1 Successful Simulations

In this section, the results from the Monte Carlo simulation are discussed. By executing a small number of test simulations for the first experiment, the simulation time was determined. In most test simulations in which the majority of the AprilTags where detected, the formation showed convergence to a rhombus-like shape, with sides of 2 meter, within the first 10 seconds. In most cases, the formation shape started to rotate after 10 seconds, which resulted in the AprilTags to move out of the FOV of the camera after approximately 16 seconds. Therefore, for every set of initial conditions, the simulation time in Gazebo is around 15.5 seconds. Moreover, it took approximately 29 time steps for the simulations to reach 15.5 seconds.

With the sample set of N = 5,000, it is still possible that, at the start of the simulation, the AprilTags of the NEXUS robots are not detected by the camera of their neighbour agents. When this happens, the simulation is classified as failed. The total number of AprilTags detected by agents 2 and 4 ( $AT_2$ ,  $AT_4$ ) can be four, and the maximum number of AprilTags detected by agent 3 ( $AT_3$ ) is 6. Thus, the simulation is successful if for the first 2 time steps:

$$AT_2^k = 4 \quad \& \quad AT_3^k = 6 \quad \& \quad AT_4^k = 4, \quad k < 3, \tag{45}$$

where k is the time step index. Furthermore, the AprilTag detection software package is often not working. Though the markers are in the FOV of the cameras, it could still happen that some of the AprilTags are not detected. This generally happens for a few deciseconds and could result in the absence of data on the feature positions for one time step. As this is the case in nearly all simulations, (45) is slightly compromised for the rest of the simulation time (k > 3). After a successful start (k < 3), is it allowed that the cameras sometimes fail to detect AprilTags. That is, if the AprilTag is not detected less than three time steps per AprilTag per simulation, the simulation is still classified as successful.

In the first experiment in which the standard feature-based control law was used, there are 629 simulations in which all AprilTags are detected in the first 2 time steps. As the AprilTag detection software is not working perfectly, this number is different in the second experiment. To ensure that the analysis of both experiments is performed using the same set of initial conditions, only the 629 initial conditions are used for the second experiment. The number of simulations in which (45) is satisfied in the second experiment is 382. Thus, there are 382 sets of initial conditions in which both experiments have a successful start. Therefore, all the other simulations are discarded in the analysis. Out of the 382 successful start positions, the first experiment resulted in 25 successful simulations (6.54 %). The second experiments resulted in 257 (67.27 %) successful simulations.

Figure 16 illustrates all initial positions for agent 2,3, and 4 for the 382 simulations with a successful start. Figure 17 shows the initial positions for three successful simulations of the first experiment. Furthermore, the intervals for all initial positions of agent 2, 3, and 4 of the successful simulations are as follows:

$$-2 \le x_2 \le -0.4,$$
 (46)

$$0.4 \le y_2 \le 3.1,\tag{47}$$

$$-1.5 \le x_3 \le 1.5,$$
 (48)

$$1.3 \le y_3 \le 3.4,$$
 (49)

$$0.4 \le x_4 \le 2,\tag{50}$$

$$0.4 \le y_4 \le 3.1. \tag{51}$$



Fig. 16: The initial positions of agent 2,3 and 4 in the 382 simulations with a successful start.



Fig. 17: The initial positions for agent 2,3 and 4 in the simulations 1348, 2746, and 1157.

For the successful simulations, convergence analysis is performed in the next subsections. First, the convergence of the successful simulations using only feature-based and distancebased control is discussed, followed by the convergence analysis of the successful simulations using the extra unit vector control.

## 6.2 Convergence Analysis Experiment 1: distance-based and feature-based

The convergence of the velocity, distance error and feature error is analysed by using the performance properties discussed in Section 2.4.

## 6.2.1 Position and Velocity Convergence

In the first experiment, the agents do not converge to a stable position. In most cases, the formation shows a rotational movement. Figure 18 shows the position for three simulations. Although the positions of the agents do not converge, the formation shape is close to its desired formation shape in the successful simulations. For example, in simulations 1347, 1157, and 2746, the angles of the rhombus at the end of the simulations are:

$$\begin{aligned} \alpha_{1157} &= 102.21^{\circ}, \quad \beta_{1157} = 97.96^{\circ}, \quad \gamma_{1157} = 79.12^{\circ}, \quad \delta_{1157} = 80.71^{\circ}, \\ \alpha_{1347} &= 101.23^{\circ}, \quad \beta_{1347} = 98.65^{\circ}, \quad \gamma_{1347} = 80.67^{\circ}, \quad \delta_{1347} = 79.43^{\circ}, \\ \alpha_{2746} &= 101.38^{\circ}, \quad \beta_{2746} = 98.57^{\circ}, \quad \gamma_{2746} = 80.81^{\circ}, \quad \delta_{2746} = 79.23^{\circ} \end{aligned}$$

where  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $\delta$  are the angles at the positions of agent 1, 3, 2, and 4.

Figure 19 depicts the mean, maximum and minimum values of the input velocity norm  $(||u_i||, i = 1, 2, 3, 4)$  of all agents. Table 3 shows the performance properties of the the input velocity norm. The rise time  $(T_r)$  is approximately one-third of the simulation time. Furthermore, agent 1 has the highest rise and settling time  $(T_s)$ . Agent 3 has the second highest rise and settling time, followed by agent 2, and agent 1 has the lowest rise and settling time. Moreover, the mean  $(||u_i||_{ss})$ , maximum  $(||u_i||_{ss})$ , and minimum  $(||u_i||_{ss})$  steady state values, of all agents are quite similar.



Fig. 18: The paths of the agents in simulations 1348, 2746 and 1157, in which the cross represents the start position of each agent and centroid



Velocity convergence agent 1 Velocity convergence agent 2

Fig. 19: The convergence of the velocity norm of agent 1,2,3 and 4.

	$T_r$ (mean)	$T_s$ (mean)	$  u_i  _{ss}$	$  u_i  _{ss}^{max}$	$  u_i  _{ss}^{min}$
Agent 1	5.6721	9.7680	0.0149	0.0281	0.0072
Agent 2	4.2461	6.9276	0.0136	0.0322	0.0068
Agent 3	4.8796	9.0484	0.0162	0.0385	0.0077
Agent 4	4.0406	6.6358	0.0144	0.0386	0.0064

Tab. 3: Performance properties of the input velocity norm of the first experiment.

#### 6.2.2 **Distance- and Feature Error Convergence**

The mean distance errors and mean feature errors (e) are depicted in Figure 20 and 21. Table 4 shows the performance properties of the distance and feature errors. The mean distance errors for edges  $\{1,2\}$  and  $\{2,3\}$  show similar behaviour, while the mean distance error of edge  $\{1,3\}$  start with a higher value. The rise time  $(T_r)$  is lower for the distance errors than for the feature-based errors, expect for edge  $\{4,1\}$ . Furthermore, the settling time  $(T_s)$ for the distance errors is generally higher compared to the errors of the feature-based agent. For the feature-based edges, the error of edge  $\{3,1\}$  has the lowers steady state value  $(e_{ss})$ . Furthermore, edges  $\{2,1\}$  and  $\{4,1\}$  show better convergence than the edges not connected to the distance-based agent. As for the edges connecting only feature-based agents, edges  $\{2,3\}$  and  $\{4,3\}$  show better convergence than edges  $\{3,2\}$  and  $\{3,4\}$ .



Fig. 20: The mean distance error (e) convergence for edges  $\{1,2\}, \{1,3\}, \{1,4\}$ .



 $\mathsf{Fig. 21: Feature error} \ (e) \ \text{for edges} \ \{2,1\}, \{2,3\}, \{3,1\}, \{3,2\}, \{3,4\}, \{4,1\}, \{4,3\}.$ 

	$T_r (\text{mean})$	$T_s$ (mean)	$e_{ss}$	$e_{ss}^{max}$	$e_{ss}^{min}$
$e_{12}$	1.9919	13.2607	9.7552e-03	2.9365e-02	2.7767e-04
$e_{13}$	1.4916	13.5848	9.3997 e-03	2.1447e-02	1.3631e-03
$e_{14}$	2.4168	15.2749	1.0381e-02	2.6309e-02	7.9310e-04
$e_{21}$	3.1544	5.2432	2.6893 e- 04	8.3861e-04	4.9840e-05
$e_{23}$	4.3510	7.3791	3.2034e-04	1.0969e-03	1.7631e-05
$e_{31}$	3.4143	8.1685	4.6769e-05	1.4992e-04	2.8618e-06
$e_{32}$	6.5063	11.0112	4.3315e-04	1.2356e-03	7.0273e-06
$e_{34}$	5.3641	13.0945	5.1914e-04	1.8372e-03	8.4351e-06
$e_{41}$	2.3553	8.2020	2.5434e-04	9.9681e-04	3.0990e-05
$e_{43}$	4.3200	6.9321	3.7890e-04	1.2527e-03	7.4037 e-05

Tab. 4: Performance properties of the distance and feature errors.

# 6.3 Convergence Analysis Experiment 2: distance-based, feature-based including extra unit vector control

#### 6.3.1 Position and Velocity Convergence

The mean, maximum and minimum values of the input velocity norm  $(||u_i||, i = 1, 2, 3, 4)$  of all agents are illustrated in Figure 22. Table 5 shows the performance properties of the input velocity norm. Compared to the results of the formation control based on only the features and distances (Section 6.2.1), the mean, minimum, and maximum velocity norms of all agents are converging closer to zero. Furthermore, the rise  $(T_r)$  and settling time  $(T_s)$  for agent 1 and 3 is lower than in the first experiments. The paths of the agents in simulations 580, 954, and 1821 are illustrated in 23. In simulations 580 and 1821, agents 1 and 3 are not moving in opposite directions in simulations. Furthermore, the formation shape is converging close to the desired formation shape. For example in simulations 580, 954, and 1821, the angles of the rhombus at the end of the simulations are:

$$\begin{aligned} \alpha_{580} &= 99.72^{\circ}, \quad \beta_{580} = 100.12^{\circ}, \quad \gamma_{580} = 79.28^{\circ}, \quad \delta_{580} = 80.87^{\circ}, \\ \alpha_{954} &= 99.81^{\circ}, \quad \beta_{954} = 100.12^{\circ}, \quad \gamma_{954} = 79.24^{\circ}, \quad \delta_{954} = 80.85^{\circ}, \\ \alpha_{1821} &= 99.66^{\circ}, \quad \beta_{1821} = 100.08^{\circ}, \quad \gamma_{1821} = 79.40^{\circ}, \quad \delta_{1821} = 80.89^{\circ}, \end{aligned}$$

where  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $\delta$  are the angles at the positions of agent 1, 3, 2, and 4.



Velocity convergence agent 1 Velocity convergence agent 2

Fig. 22: The convergence of the velocity norm of agent 1,2,3 and 4.



Fig. 23: The paths of the agents in simulations 580, 954, and 1821.

	$T_r$ (mean)	$T_s (\text{mean})$	$  u  _{ss}$	$  u  _{ss}^{max}$	$  u  _{ss}^{min}$
Agent 1	3.1280	5.8301	0.0085	0.0281	0.0007
Agent 2	4.8247	7.8487	0.0126	0.0193	0.0041
Agent 3	1.5594	4.4339	0.0072	0.0189	0.0002
Agent 4	5.4210	9.3523	0.0134	0.0584	0.0046

Tab. 5: Performance properties of the input velocity norm in the second experiment.

## 6.3.2 Distance- and Feature Error Convergence

The mean distance errors and mean feature errors (e) are depicted in Figure 24 and 25. Table 6 shows the performance properties of the distance and feature errors of the formation including the unit vector control. The rise time  $(T_r)$  for edges  $\{1,2\}, \{1,3\}, \{1,4\}, \{2,1\}, \{2,3\}, \{4,1\},$ and  $\{4,3\}$  is higher compared to the first experiment. For edges  $\{3,1\}, \{3,2\}$ , and  $\{3,4\}$ , the rise time is lower than in the first experiment. The settling time  $(T_s)$  of the distance errors are almost three times as small as in the first experiment. Furthermore, the steady state values of all the errors  $(e_{ss})$  are slightly lower compared to the first experiment. Moreover, the steady state value of all the feature errors is lower than in the first experiment. The steady state value of the maximum mean error  $(e_{ss}^{max})$  of edges  $\{1,2\}, \{1,3\}, \{1,4\}, \{3,1\},$ and  $\{3,4\}$  is slightly higher than in the first experiment. For edges  $\{2,1\}, \{2,3\}, \{4,1\},$  and  $\{4,3\}$  the maximum mean error is slightly lower. As for steady state value of the minimal mean error  $(e_{ss}^{max})$ , all edges, except edge  $\{4,1\}$ , have a lower minimal mean error than in the first experiment.



Fig. 24: The mean distance error (e) convergence for edges  $\{1,2\},\{1,3\},\{1,4\}$ .



Fig. 25: Feature error (e) for edges  $\{2,1\}, \{2,3\}, \{3,1\}, \{3,2\}, \{3,4\}, \{4,1\}, \{4,3\}.$ 

	$T_r$ (mean)	$T_s$ (mean)	$e_{ss}$	$e_{ss}^{max}$	$e_{ss}^{min}$
$e_{12}$	2.8632	4.5831	7.8038e-03	3.0758e-02	3.1426e-06
$e_{13}$	1.6621	3.6089	8.2253e-03	3.6851e-02	6.7172 e- 05
$e_{14}$	2.9305	4.2877	7.9745e-03	3.0396e-02	7.3229e-05
$e_{21}$	4.5531	7.9105	1.7425e-04	3.8078e-04	4.9643 e-07
$e_{23}$	4.5147	7.8286	1.9375e-04	3.3612e-04	7.0479e-06
$e_{31}$	2.2746	5.6298	4.4739e-05	4.2563e-04	5.9109e-07
$e_{32}$	4.4677	7.5509	2.7957e-04	1.1258e-03	1.9979e-06
$e_{34}$	5.2849	9.3239	3.1363e-04	4.8176e-03	1.0427 e- 05
$e_{41}$	5.2104	9.9223	2.0514e-04	7.9611e-04	7.5540e-06
$e_{43}$	4.9191	9.5588	2.3086e-04	6.4695e-04	6.9450 e- 07

Tab. 6: Performance properties of the distance and feature errors.

## 7 Discussion

In section 6, the result of the velocity and error convergence is provided for the successful simulations of the sample data of N = 5,000. Out of the 382 simulations with a successful start, only 25 simulations succeeded to keep the features in the FOV in the first experiment (6.54%), which is extremely low. This can be explained by the rotating movement of the formation that occurs in most simulations. When the formation starts to rotate, the features eventually move out of the FOV of the agents. It is expected that this would also happen after 15.5 seconds for the agents with the initial positions of the successful simulations. The rotational movement is possibly caused by measurement errors of the sensors, these errors could lead to a disagreement of the desired distance/feature angle. Another possible reason could be that the different orientations of the agents result in a disagreement. However, this is not proved in this research. Furthermore, adding the extra unit vector constraint in the second experiment prevented this rotation and resulted in a significant higher success rate of 67.27%. Additionally, the shape of the final formations in both experiments were quite similar to the desired formation. Moreover, the shape of the final formations in the second experiments were closest to the desired formation. Furthermore, the mean velocity norm ||u||converged closer to zero in the second experiment. For the agents with three neighbour agents (agent 1 and 3), the rise time  $T_r$  and settling time  $T_s$  are lower in the second experiment. Agent 2 and 4 perform better in the second experiment. This could possibly be explained by the fact that agent 2 and 4 are most affected by this rotational movement of the first experiment. Agent 2 has three neighbours and is placed parallel to agent 1, this could be the reason that it performs better than agent 2 and 4 in the first experiment. Agent 2 and 4 only have two neighbours with both a different orientation, this could cause difficulties in both converging to the correct feature angles and in staying in the FOV of the camera. Though, this is not proved in this research. As for errors of the feature-based agents, it is noticeable that the error of the edges connected to the distance-based agent converges faster and closer to zero than the ones that are not. Additionally, in 9 out of the 10 edges, the error converged closer to zero in the second experiment. Moreover, the settling time  $T_s$  of the edges with a distance-based agent (6 out of 10) were lower in the second experiment. Another interesting finding is that the settling time  $T_s$  for the distance errors of the second experiment is three times as small as for the first experiment.

## 8 Conclusion

In this thesis, the goal was to design and validate a distributed formation controller using feature measurements that can be implemented on the NEXUS robots. The controlled formation consisted of three agents using feature-based control and one agent using distance-based control. Furthermore, two experiments were performed with each a different control law for the feature-based agents. From the first experiment it can be concluded that in this formation setup using feature measurements to control the feature angle will not perform well when using a camera with a limited FOV. The rotational movement of the agents causes the agents to move out of the FOV of the camera in most cases. The second experiment, in which an extra constraint was added to the feature-based control law, is more applicable in a real-life environment. The extra constraint prevents the agents to move out of the FOV. No extra sensing was required since the left unit bearing vector is also necessary for determining the feature angle. Moreover, since the feature error of the feature agents connected to the dis-

tance agent showed better convergence, combining distance-based control with feature-based control seems to improve the performance of the feature-based agents. To conclude, the designed formation controller using the extra constraint can be implemented on the NEXUS robots. However, it still would result in some failures. Therefore, further research is necessary to improve the performance of formation control using feature-based measurements.

## 9 Limitations and Further Research

Because of the poor performance of the control law in the first experiment, only 25 simulations were analysed. Therefore, the results of the convergence analysis are not reliable. Moreover, it is unclear per simulation what caused the other simulations to fail. The main two causes are presumably the failure of the AprilTag detection software and the performance of the control law. Since the success rate in both experiments differ greatly, it can be assumed that a large number of simulations failed because of the control law. However, the exact number of simulations failed because of the control law is unknown. Moreover, measurement errors caused by the camera and laser are not taken into account in the design of the formation controller. This could lead to a disagreement between the edges and can have effect on the performance of the formation control. Therefore, it would be interesting to research the effect of measurement errors or disagreements in the future. Additionally, it could be that adding an extra edge for agent 2 and 4 would increase their performance. Therefore, this could be considered in further research. Furthermore, it would be interesting to research the rigidity of formations using feature measurements, since this has not been done before. Finally, examining formation movement control using feature-based measurement would be beneficial for practical applications.

## References

- Almada-Lobo, F. (2016). The industry 4.0 revolution and the future of manufacturing execution systems (mes). *Journal of innovation management*, 3(4):16–21.
- Aström, K. J. and Murray, R. M. (2010). Feedback systems: an introduction for scientists and engineers. Princeton university press.
- Bussmann, S., Jennings, N. R., and Wooldridge, M. (2013). Multiagent systems for manufacturing control: a design methodology. Springer Science & Business Media.
- Chan, N., Jayawardhana, B., and Garcia de Marina, H. (2019). Controlling a triangular formation using features. unpublished.
- Guanghua, W., Deyi, L., Wenyan, G., and Peng, J. (2013). Study on formation control of multi-robot systems. In Intelligent System Design and Engineering Applications (ISDEA), 2013 Third International Conference on, pages 1335–1339. IEEE.
- Han, Z.-m., Lin, Z.-y., Fu, M.-y., and Chen, Z.-y. (2015). Distributed coordination in multiagent systems: a graph laplacian perspective. Frontiers of Information Technology & Electronic Engineering, 16(6):429–448.
- Hevner, A. R. (2007). A three cycle view of design science research. Scandinavian journal of information systems, 19(2):4.
- Ike, T. (2018). Vision-Based Distributed Formation Control of Multi-Agent Systems. Master's thesis, University of Groningen, the Netherlands.
- Kusiak, A. (2018). Smart manufacturing. International Journal of Production Research, 56(1-2):508-517.
- Oh, K.-K., Park, M.-C., and Ahn, H.-S. (2015). A survey of multi-agent formation control. Automatica, 53:424–440.
- Olson, E. (2011). AprilTag: A robust and flexible visual fiducial system. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 3400–3407. IEEE.
- Sycara, K. P. (1998). Multiagent systems. AI magazine, 19(2):79.
- Zhao, S., Zelazo, D., et al. (2016). Bearing rigidity and almost global bearing-only formation stabilization. *IEEE Trans. Automat. Contr.*, 61(5):1255–1268.
- Zheng, P., Sang, Z., Zhong, R. Y., Liu, Y., Liu, C., Mubarok, K., Yu, S., Xu, X., et al. (2018). Smart manufacturing systems for industry 4.0: Conceptual framework, scenarios, and future perspectives. *Frontiers of Mechanical Engineering*, pages 1–14.