Master Research Project

Industrial Engineering and Management
Smart Systems in Control and Automation

Distributed Formation Control of Multi-agent Systems Using Bearing Measurements

Author:
Rick Hendriks
s2329921

Supervisors:
prof. dr. Bayu Jayawardhana
prof. dr. ir. Ming Cao

Abstract

There is an increasing interest for formation and cooperative control in robotic systems. Multi-robot systems are able to perform tasks that exceed the capabilities of a single robot. It has been observed that these systems can be very robust and cost effective. Teams of autonomous robots have been used in a wide range of applications, such as search and rescue missions or lifting and transportation of heavy objects. The goal of this research is to develop and analyse the performance and robustness of an angle-constrained distributed formation control algorithm, based on local bearing measurements. Different formation control approaches are reviewed and explained. The formation is simulated in Gazebo and the trajectory and convergence is monitored during various simulations. A Monte Carlo simulation is performed to increase the reliability of the results. Furthermore, the robustness of the control algorithm is tested by disturbing the laser scanner with Gaussian noise. The results from the Monte Carlo simulations show that 89.25% of 800 simulations is successful when an optimal gain and desired inner angle is used. The success rate of the simulations stays approximately the same when noise is added to the laser scanner. This shows that the control law is functional and robust for laser scanner noise. Further research includes the implementation of the control algorithm in a real Nexus robot and optimizing the gain for a desired inner angle.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the last few years, the attention for formation and cooperative control in robotics and unmanned systems has increased enormously. Initially a single autonomous robot was used to perform tasks, but nowadays the interest of researchers shift more to cooperative control of robots to perform various objectives. A multi-robot system is able to perform tasks that exceed the capabilities of a single robot, not only due to workload sharing but also in terms of functionality (Geihs, 2020). Just like a group of humans can accomplish more than a single individual, autonomous robots working together provides opportunities to accomplish tasks that a single robot cannot do alone. Teams of autonomous robotic platforms have been utilized in a wide range of applications, such as cleanup of toxic waste, planetary exploration, search and rescue missions and automated manufacturing (Parker, 1996). It has been observed that multi-robot systems are very robust and cost effective as compared to building a single costly robot with all the capabilities (Gautam and Mohan, 2012).

There has been substantial progress in robotic hardware and software. This makes autonomous robots fit for applications in private, industrial and public environments. Thus, we are observing a shift of research focus from theoretical multi-agent models to practical software and hardware engineering issues (Pěchouček and Mařík, 2008). The ongoing development in all areas makes cooperative control of multi-robot systems an interesting field of research.

The topic of this Master's Thesis is the study and design of distributed multi-robot formation control. The research context of multi-robot systems and formation control is given Section 1.1. The challenges that arise with real-life applications is described in Section 1.2. In Section 1.3 the research goal for this project is formulated. The according research questions are formulated in Section 1.4. Finally, in Section 1.5, the outline of this thesis is presented.

1.1 Research Context

A robot is a programmable machine capable of carrying out a complex series of actions automatically (The Oxford English Dictionary, Oxford University Press). An autonomous robot is capable of perceiving its environment through sensors and operate without human intervention. When a group of robots operates in the same environment, it is called a multi-robot system (Farinelli et al., 2004). A frequently occurring desire when dealing with multi-

robot systems is formation keeping (Zhao and Zelazo, 2019). Generically, the primary goal of formation keeping is to steer and stabilize a group of agents to a desired geometrical formation (Chan et al., 2019). Formation control deals with the problem of controlling the relative position and orientation of robots inside a group to create a desirable group formation.

Early works on the control of multi-agent systems are mainly focused on centralized control methods. Centralized control protocols have been constructed based on the common assumption that the information of all agents is available or the multi-agent system possesses all-to-all communication (Dang et al., 2017). This is a good method for controlling a small group of robots. The centralized architecture has the advantage of controlling and observing all the agents in an integrated manner. The drawbacks of the centralized communication architecture are inflexibility and large computational costs for each agent especially when the number of robots is large. Moreover, the failure of a centralized controller leads to a total failure of the whole control system (Chen et al., 2016).

In order to overcome this type of single point of failure tendency, much research has been focusing on distributed cooperative control strategies where vehicle control laws are coupled and each vehicle makes its own decision according to the states of their neighbors. In contrast to centralized control, a distributed control approach can provide more flexibility, easier implementation, and less computation loads as the controller of each agent only requires the information of its neighbor agents (Dang et al., 2017). Also, distributed control provides increased performance and robustness due to its ability to adjust to agent failures or changing environment. A decentralized system is a subset of a distributed system. Decentralized means that there is no single point where the decision is made. Every node makes a decision for it's own behaviour and the resulting system behaviour is the aggregate response. The requirement that each vehicle have the knowledge of the desired group trajectory may not be realistic for many applications. For example, communication bandwidth and range limitations may prevent each vehicle in the group having access to the group trajectory information (Ren and Sorensen, 2008).A key characteristic of a decentralized system is that typically no single node will have complete system information. A graphical representation of a centralized and distributed network is given in figure 1.1.

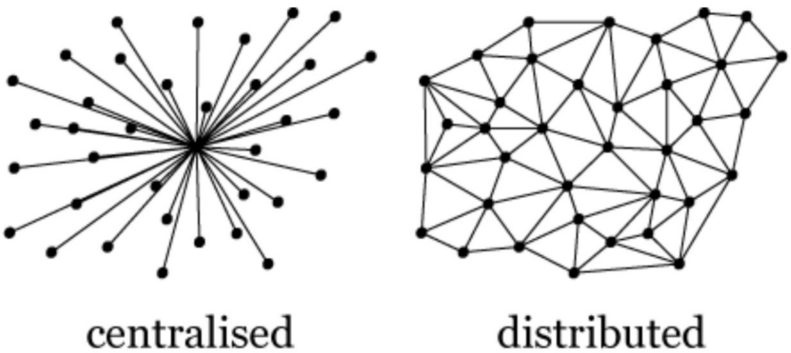


Figure 1.1: Graphical representation of a centralized and distributed network.

There are different approaches to formation control. In (Oh et al., 2015), the existing types of formation shape control strategies are reviewed and compared in terms of sensing capa-

Table 1.1: Distinctions among position-, displacement-, and distance-based formation control.

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

bility and the interaction topology. These are known as the position-based approach, the displacement-based approach and the distance-based approach:

- Position-based control: Agents sense their own positions with respect to a global coordinate system. The robots move based on their own position and the desired position with respect to the global coordinate system.
- Displacement-based control: Agents sense the relative positions of their neighbors with respect to the global reference frame. The control is based on the desired displacements with respect to the global coordinate system. Agents do not require knowledge on their positions with respect to the coordinate system.
- Distance-based control: Agents sense the relative positions of their neighbors with respect to the local reference frame of the agents. The inter-agents distances are controlled to achieve the desired formation.

1.2 Challenges

As explained in section 1.1, there are numerous applications that can utilize formation control and cooperative control of robotic systems. However, there still exist many technological and scientific challenges before a wide application in real life scenario's becomes feasible. One of the challenges that multi-robot systems need to face is how to coordinate their motion without access to an external localization system. While robots can sometimes use GPS signals, in many cases it is desirable for robots to coordinate using only onboard sensing.

In practical applications, multi-robot systems may have to deal with several tasks simultaneously, such as: collision avoidance, tracking the desired trajectory and keeping the desired formation. Real-life application of multi-robot systems is only possible when all challenges are solved. These challenges involve: cooperative control strategies, communication constraints, localization of the robot positions and learning certain behaviours. Within this thesis the focus in mainly on formation control of the robots, without using an external positioning system.

1.3 Research Goal

This research project will focus on the design and validation of a formation controller using bearing measurements. The purpose of this controller is to move robots from random initial positions to a desired formation shape. Thus, the goal of this project can be formulated as follows:

"To develop and to numerically analyse the performance and robustness of an angle-constrained distributed formation control algorithm based on local bearing measurement"

1.4 Research Questions

To achieve the goal, formulated in the previous section, the main research question is formulated.

"How can an angle-constrained distributed formation control algorithm based on local bearing measurements be developed and numerically analyzed to test the performance and robustness?"

To answer this research question, the following sub-questions need to be answered:

1. *Which formation control law is suitable for distributed formation control?*
2. *How is local range sensor data transformed into useful data for distributed formation control?*
3. *How is the formation control algorithm designed?*
4. *How robust is the designed control algorithm?*

1.5 Thesis Outline

This thesis is organized as follows. Chapter 2 provides a literature review where graph theory and different formation control approaches are investigated. Chapter 3 explains the model that will be used in to control the formation. In Chapter 4 the simulation setup is described, followed by the results of the experiments in Chapter 5. In Chapter 6 the results of the simulations are discussed. Finally, conclusions and recommendations for further research are presented in Chapter 7 and Chapter 8.

Chapter 2

Literature Review

Substantial research and development has already been performed in the areas of cooperative and formation control of mobile robots. In this chapter a literature review is performed including graph theory and various formation control approaches.

2.1 Graph Theory

Graph theory is a way to model information exchange in a network of dynamical systems. A graph $\mathcal{G}$ consists of a set of nodes $\mathcal{V} = \{1, \cdots, N\}$ and a set of edges $\mathcal{E} = \mathcal{V} \in \mathcal{V}$. The edge (i, j) in the edge set of a directed graph denotes that agent j can obtain information from agent i, but not necessarily vice versa. The set of neighbors of agent i is defined by $\mathcal{N}_i := \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. In the case of a multi-robot system, the nodes represent the robots and the edges represent the information exchange between them.

The edge set $\mathcal{E}$ can be described by the adjacency matrix $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{nxn}$ with $i, j \in \mathcal{V}$. Another fundamental matrix in graph theory, which can be derived from the adjacency matrix is the Laplacian matrix $\mathcal{L} = [\ell_{ij}] \in \mathbb{R}^{nxn}$, with $\ell_{ii} = \sum_{j=1}^{n} g_{ij}$ and $\ell_{ij} = -g_{ij}, \mathrm{i} \neq j$.

The $n \times m$ incidence matrix D, can be defined as:

$$d_{ij} = \begin{cases} +1, & \text{if} \quad i = \mathcal{E}^{\text{head}} \\ -1, & \text{if} \quad i = \mathcal{E}^{\text{tail}} \\ 0, & \quad\quad \text{otherwise} \end{cases} \tag{2.1}$$

2.2 Formation Control Approaches

2.2.1 The behaviour-based method

In the behaviour-based approach, a behaviour (e.g. obstacle avoidance, goal-seeking, formation keeping) is assigned to each individual robot. The vehicle's final action is then derived by weighing the relative importance of each behaviour. The advantage of behaviour-based approach is that it is natural to derive control strategies when agents have multiple competing objectives. In addition, there is explicit feedback to the formation since each agent reacts according to the position of its neighbors (Ren and Beard, 2002). Another advantage

is that the robots are controlled in a decentralized fashion. Therefore, it is ideal for a large number of robots. However, the mathematical analysis of this approach tends to be difficult, and consequently it is not easy to guarantee the convergence of a formation to the desired configuration. In most cases, the dynamics of the individual robots are therefore simplified as a single integrator $\dot{x} = u$. By doing this, the stability analysis becomes easier, but is still very complex. The dynamics of real robots are even more complex, making such approach less feasible. This is not ideal when creating an experimental environment.

2.2.2 The virtual-structure method

In the virtual structure approach, the entire formation is treated as a single rigid structure (e.g. circle, square). The advantage of this approach is that it adds a certain robustness to the formation, since all robots are mutually coupled to each other. Therefore, the formation can be maintained very well while maneuvering. Another advantage if this approach is that it is fairly easy to prescribe the coordinated behaviour of the whole formation group (Ge et al., 2018). A disadvantage is that it is often more difficult to determine the stability of a formation than other methods. Also, it is necessary for the position and control variables of each individual vehicle or the full state of the virtual structure to be communicated to each individual vehicle in the formation.

2.2.3 The leader-follower method

The third approach to cooperative and formation control is the leader-follower approach. In this method, some robots are designated as leaders that follow predefined trajectories, while the followers will follow according to a relative posture to the leader robot. The main advantage of this approach is that it is easy to understand and implement. Also, it is often less difficult to determine stability with the leader-follower approach than with other methods. However, this method holds the disadvantage that there is no explicit feedback from the followers to the leader. As consequence, if a follower is perturbed, the formation can not be maintained due to the lack of explicit follower feedback. Another disadvantage to this method is that the position of the leader must be communicated to the followers through communication or sensor feedback so that follower robots can use this information as a control input. This makes the leader-follower more ideal for a centralized control strategy, which is less ideal for a large number of robots.

2.2.4 The position-based approach

In the position-based approach, agents measure their own positions with respect to the global coordinate system. The desired formation is prescribed by the desired positions for the agents. The primary objective of formation control in this approach is to drive the agents to their destinations. Interconnections between the agents is beneficial when there is an additional objective to maintain a prescribed formation shape. Consider N single-integrator modeled agents in n-dimensional space. Then $\dot{p}_i = u_i$, where $p_i \in \mathbb{R}^n$ and $u_i \in \mathbb{R}^n$ denote the position and control of agent i, for $i \in \{1, ..., N\}$. Suppose the objective is to move the agents from their initial positions to their desired destination, while keeping their formation. Then the destinations and desired formation shape are specified by, $p^* = [p_1^{*T} ... p_N^{*T}]^T \in \mathbb{R}^{nN}$, where the

objective is to achieve $p \to p^*$, while satisfying $p_i - p_j = p_i^* - p_j^*$ for all $i, j \in \{1, ..., N\}$. Then the following control law can be considered:

$$u_i = k_p(p_i^* - p_i) \tag{2.2}$$

where the gain $k_p > 0$.

In the position based approach, the desired formation can be achieved without any interactions between the agents, under ideal conditions. However, disturbances will occur in a real life situation. Some interactions among agents have therefore been introduced in the literature to consider these issues. This approach might not be cost effective because agents need to carry positioning sensors such as GPS receivers to measure their absolute positions.

2.2.5 The displacement-based approach

In displacement-based approach, each agent measures the relative positions (displacements) of neighboring agents with respect to the global coordinate system. Then the agents achieve the desired formation by actively controlling the displacements of their neighboring agents. Consider again, a single-integrator modeled agents n-dimensional space. $\dot{p}_i = u_i$, where $p_i \in \mathbb{R}^n$ and $u_i \in \mathbb{R}^n$ denote the position and control of agent i, for $i \in \{1, ..., N\}$, which is also used in the position-based approach. Since each agent measures the relative positions of some other agents, agent i has the following relative position measurements: $p_j i := p_j - p_i, j \in \mathcal{N}$. The formation control can be solved based on consensus. Consider the following proposed control law for agent i:

$$u_i = k_p \sum_{j \in \mathcal{N}_\rangle} a_{ij}(p_j - p_i - p_j^* + p_i^*) \tag{2.3}$$

where the gain $k_p > 0$. We have the following first-order consensus dynamics:

$$\dot{e_p} = -k_p(L \otimes I_n)e_p \tag{2.4}$$

2.2.6 The distance-based approach

In distance-based approach, agents measure the relative positions of their neighboring agents with respect to their local coordinate system, assuming agents do not share a common sense of orientation and they actively control the length of the relative positions to achieve the desired formation. Since only the inter-agent distances are controlled in this approach, the sensing graph for the agents requires to be persistent. The main advantage of this approach is that agents do not need to share any knowledge on global coordinate system. This makes it a decentralized approach.

2.2.7 Feature based control law using bearing measurements

This control law uses bearing information to control the robots to a desired formation. Consider a triangular formation of N = 3 agents, moving in 2-D space. The position vector of each agent can be described by $p_i \in \mathbb{R}^2$, i = 1, 2, 3. The dynamics of the agent is

$$\dot{p}_i(t) = v_i(t), \tag{2.5}$$

where $v_i(t) \in \mathbb{R}^2$ is the controlled velocity to be designed. Each agent is equipped with two markers, one is located on the left and the other is placed on the right of the agent. If the center of the agent is denoted by p_i, the relative position of the markers to the center is assumed to satisfy

$$p_{iL} = p_i - \underline{a}, \quad p_{iR} = p_i + \underline{a}, \quad i = 1, 2, 3, \tag{2.6}$$

with p_{iL} denote the position of the left marker and p_{iR} denote the position of the right marker. It is assumed that the agents are equipped with a sensor system that is able to measure the bearing of the markers, with respect to the center point of the measuring agent. The bearing vectors between agent i and the markers of agent j is then given by

$$\widehat{z}_{ij\ell} = \frac{z_{ij\ell}}{\|z_{ij\ell}\|}, \quad \ell \in \{L, R\}. \tag{2.7}$$

Denote the angle between the bearing vectors $\widehat{z}_{ijL}$ and $\widehat{z}_{ijR}$ as θ_{ij} and it satisfies

$$\theta_{ij} = \arccos\left(\langle \widehat{z}_{ijL}, \widehat{z}_{ijR} \rangle\right). \tag{2.8}$$

Due to symmetry in the bearing measurements, we have that

$$\langle \widehat{z}_{ijL}, \widehat{z}_{ijR} \rangle = \langle \widehat{z}_{jiL}, \widehat{z}_{jiR} \rangle \Leftrightarrow \cos\theta_{ij} = \cos\theta_{ji}. \tag{2.9}$$

As described above, $\widehat{z}_{ijL}$ and $\widehat{z}_{ijR}$ are the normal vectors pointing from the center of the measuring agent to their neighbors, consisting of an x and y-coordinate. Accordingly, the following control law is proposed:

$$v_i = K \sum_{j \in \mathcal{N}_i} e_{ij} \left(\widehat{z}_{ijL} + \widehat{z}_{ijR}\right), \tag{2.10}$$

where

$$e_{ij} = \langle \widehat{z}_{ijL}, \widehat{z}_{ijR} \rangle - \left\langle \widehat{z}^*_{ijL}, \widehat{z}^*_{ijR} \right\rangle = \cos\theta_{ij} - \cos\theta^*_{ij} \tag{2.11}$$

$$\begin{aligned} \theta_{ij}(t) < \theta^*_{ij} &=> e(t) > 0 \\ \theta_{ij}(t) > \theta^*_{ij} &=> e(t) < 0. \end{aligned} \tag{2.12}$$

Now, the final control can be described by:

$$u_k = K e_{ij} \left(\widehat{z}_{ijL} + \widehat{z}_{ijR}\right), \tag{2.13}$$

with $K > 0$ be the constant gain of the control law.

Figure 2.1 gives a graphical representation of the triangular formation. It shows the agents equipped with markers and indicates the bearing vectors and angles. In the control algorithm, each agents measures both their neighbors.

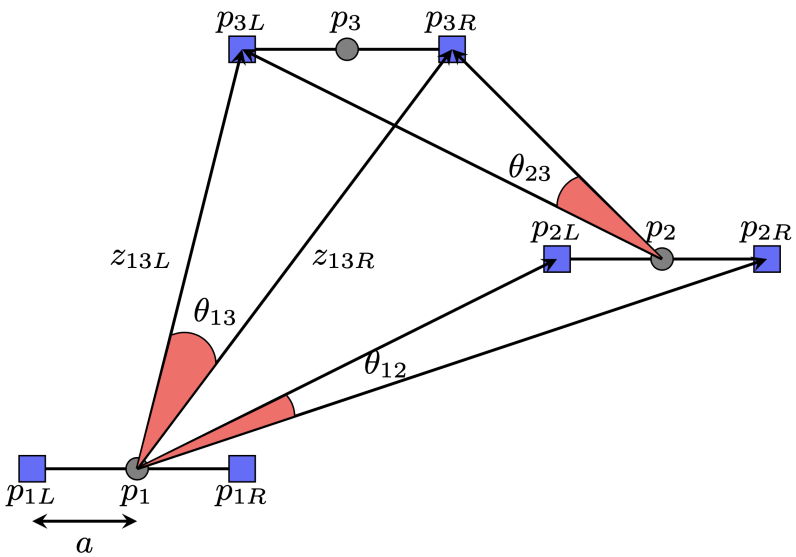


Figure 2.1: Triangular formation with features, defined in terms of angles $\theta_{12}, \theta_{13}, \theta_{23}$.

Chapter 3

Model

The proposed control law introduced in section 2.2.7 defines a desired formation based on inter-agent bearings. It is assumed that the agents are equipped with a sensor system to measure the bearings. In this research, the agent are not equipped with markers and are limited to the use of a laser scanner. This section elaborates on the design of the control algorithm. The proposed control law will be translated into the algorithm.

3.1 Problem setup

Consider a group of n agents moving in the plane. Let $\mathcal{V} = \{1, 2, \ldots, n\}$ be the index set of the robots. Each agent is described by its position vector $p_i \in \mathbb{R}^2, i = 1, 2, 3$. The dynamics of the agent is

$$\dot{p}_i(t) = v_i(t), \tag{3.1}$$

where $v_i(t) \in \mathbb{R}^2$ is the controlled velocity to be designed. Each robot is equipped with a laser scanner at the center p_i of the robot. This laser scanner has a circular shape with a radius specified by $r_i \in \mathbb{R} > 0$. The 'observer' robot is denoted by i and the measured robot is denoted by j. Now, robot i is able to detect two points on the laser scanner of robot j. These points p_{jLi} and p_{jRi} indicate the left and right side of the laser scanner of robot j, detected by robot i.

The measurements from robot j that is available to robot i are the relative bearing measurements $z_{ijL} = \frac{z_{ijL}}{\|z_{ijL}\|}$ and $z_{ijR} = \frac{z_{ijR}}{\|z_{ijR}\|}$, with $z_{ijL} = p_{jL_i} - p_i$ and $z_{ijR} = p_{jR_i} - p_i$. The two bearing factors form an angle θ_{ij}.

3.2 Control algorithm

Since the robot is solely equipped with a laser scanner, this will be the only information source available. Figure 4.2 shows the adjusted triangular formation for this algorithm. The white circles above the nexus robots represent the laser scanners. In this thesis a laser scanner is used as sensor system, but any sensor that is able to measure the distance and bearing to the neighboring agent can be used for this control law.

The laser scanner generates data consisting of 720 distances $r_{ij}(t)$. It scans 360-degree around the agent, and therefore each data value represents $360/720 = 0.5$ degree. With this information, the angle $\theta_{ij}(t)$ to a certain object can be calculated based on the indices in the array of $r_{ij}(t)$. If no objects are detected within range of the laser scanner, it gives the output $r_{ij}(t) = \infty$. This will be altered to 0. Since each robot should detect the two other robots, two separate arrays are generated.

With the distance and bearing information, the bearing vectors $\widehat{z}_{ijL}$ and $\widehat{z}_{ijR}$ can be determined by

$$\begin{bmatrix} z_{ij,x}(t) \\ z_{ij,y}(t) \end{bmatrix} = \begin{bmatrix} \sin\left(\theta_{ij}(t)\right) r_{ij}(t) \\ -\cos\left(\theta_{ij}(t)\right) r_{ij}(t) \end{bmatrix}, \tag{3.2}$$

where $z_{ij,x}$ and $z_{ij,y}$ represent the relative distances from agent i to j in x and y direction.

The normalized bearing vectors are calculated by

$$\widehat{z}_{ij\ell} = \frac{z_{ij\ell}}{\|z_{ij\ell}\|}, \quad \ell \in \{L, R\}, \tag{3.3}$$

with

$$\|z_{ij\ell}\| = \sqrt{z_{ij,x}^2 + z_{ij,y}^2}. \tag{3.4}$$

The signal error e_{ij} can be determined by means of Equation 2.11. The final control is formed by the control inputs for the x-direction and y-direction, and can be described for robot 1 by

$$u_x = K((cos\theta_{12} - cos\theta_{12}^*)(\widehat{z}_{12L,x} + \widehat{z}_{12R,x}) + (cos\theta_{13} - cos\theta_{13}^*)(\widehat{z}_{13L,x} + \widehat{z}_{13R,x})) \tag{3.5}$$

$$u_y = K((cos\theta_{12} - cos\theta_{12}^*)(\widehat{z}_{12L,y} + \widehat{z}_{12R,y}) + (cos\theta_{13} - cos\theta_{13}^*)(\widehat{z}_{13L,y} + \widehat{z}_{13R,y})). \tag{3.6}$$

The resulting input will be adjusted when one of the robots moves, until the system has converged.

The model which is used in this thesis in illustrated in Figure 3.1. Here is shown what the parameters mean when the robots are equipped with a laser scanner. The structure of the control algorithm is given in a pseudo code, which can be seen in Algorithm 1.

3.3 Assumptions

During the design of the control algorithm, several assumption were made. The assumptions are as follows:

- It is assumed that the robots move in an empty world. This means that the laser scanner only detects other agents and not objects.
- It is assumed that the nexus robots all start facing the same direction.
- It is assumed that the robots move in x and y direction, without rotation on the z-axis
- It is assumed that the starting position in are a random generated triangle. Thus, not start from a straight line.

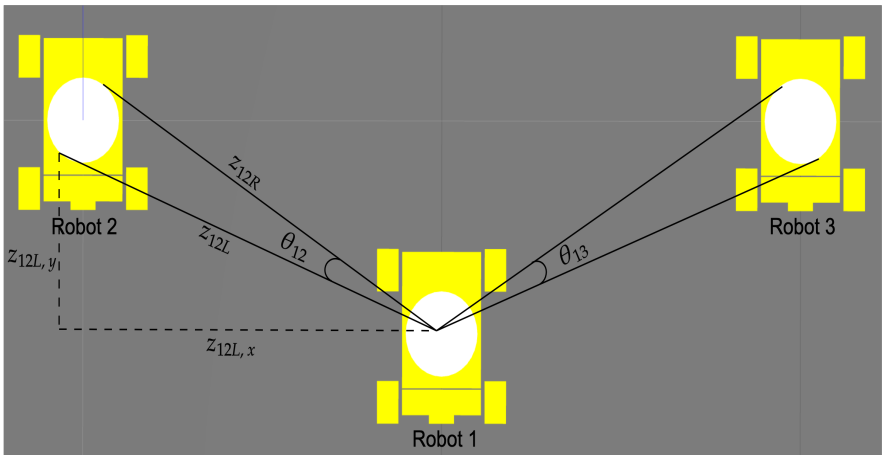


Figure 3.1: Robot with laser scanner setup in Gazebo.

Algorithm 1 Structure of the control algorithm

1: **while** Laser scanner generates data **do**
2: **if** Data points output $\infty \Rightarrow 0$ **then**
3: Split data points into robot 1 and robot 2 array *scandata_robot_i*
4: Translate array indices to θ_{ij}
5: **if** length *scandata_robot_i* $\neq 0$ **then**
6: Translate data points to $z_{ij}(t)$

$$\begin{bmatrix} z_{ij,x}(t) \\ z_{ij,y}(t) \end{bmatrix} = \begin{bmatrix} \sin\left(\theta_{ij}(t)\right) r_{ij}(t) \\ -\cos\left(\theta_{ij}(t)\right) r_{ij}(t) \end{bmatrix}$$

7: Calculate normalized bearing vector

$$\widehat{z}_{ij\ell} = \frac{z_{ij\ell}}{\|z_{ij\ell}\|}$$

8: Calculate the error

$$e_{ij} = \cos\theta_{ij} - \cos\theta_{ij}^{*}$$

9: Calculate control inputs

$$u_k = K e_{ij}\left(\widehat{z}_{ijL} + \widehat{z}_{ijR}\right)$$

Chapter 4

Simulation setup

In this chapter the setup for the simulations will be discussed. It elaborates on the software and hardware required to validate the control law. The hardware is present in the DTPA-lab of the University of Groningen. Through the software, the simulation can be performed on a digital twin.

4.1 Nexus Robot

The distributed control law will be tested on the Nexus robots, which are used in the DTPA-lab. The robots are equipped with mecanum wheels which allows them to move omnidirectional. This means that the robot can move directly sideways while maintaining its orientation, besides driving forwards and backwards. Figure 4.1 shows the nexus robot with its mecanum wheels and without sensors. In the simulation the Nexus robots are equipped with a laser scanner to gather useful data.

4.2 Laser Scanner

To measure the inter-agent distances needed for executing the control law, a Hokuyo laser scanner is used. This laser scanner estimates a distance by calculating phase differences. It emits a laser beam which moves in a straight line, rebounds from obstacles, and returns to the laser scanner. The range finder compares the wave sent and received and calculates the phase

Figure 4.1: The Nexus robot.

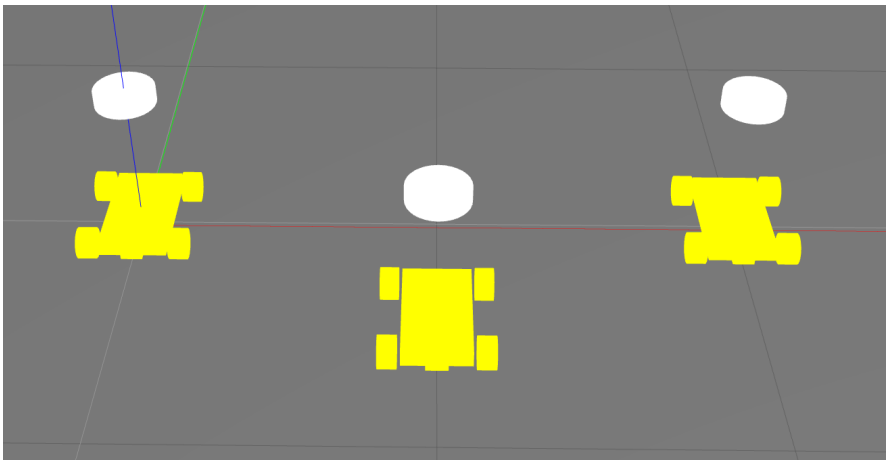


Figure 4.2: Robot with laser scanner setup in Gazebo.

difference between the two. This phase difference is in fact proportional to the time taken by the laser to go from the sensor to the obstacle and back, and this time is itself proportional to the distance travelled.

The laser is placed in the center of the robots 20 cm above the base of the robot. To detect the other robots, the laser is modeled as a cylindrical object. The laser scanner has an angle range of 360°. Within this range, 720 data points are generated 40 times per second. Thus, there is a data point every 0.5°. The distance range of the laser scanner is [0.10 10.0] meter.

4.3 ROS - Robot Operating System

ROS is an open-source software framework that enables communication and controlling of the robots. The main communication mechanism used by ROS is through sending and receiving messages. The messages are organized in specific categories called topics. Nodes may publish messages on a specific topic or subscribe to a topic to receive information.

Nodes are executable files that perform some computations or tasks. The ROS Master node provides naming and registration services to the nodes in the ROS system. It tracks publishers and subscribers to the topics. Communication is established between the nodes by the ROS Master.

4.4 Gazebo

To examine the performance of the the control algorithm, the simulations are executed in the environment of Gazebo. Gazebo is an open-source 3D robotics simulator with real-time graphics. It is compatible with ROS, which makes it easy to communicate with the simulated robot. Figure 4.2 depicts the simulation setup in Gazebo. The three robots are shown with the laser scanners above them.

4.5 Computation Graph

A computation graph can be used to represent mathematical functions in the language of graph theory. Figure 4.3 shows the computation graph for the nexus robots. The eclipses indicate the nodes, while the rectangles indicate the topics. Node /n_1 is subscribed to the /n_1/scan topic and therefore receives laser data as input. Node /n_1 then executes the control algorithm and publishes information to the topics /n_1/cmd_vel and /n_1/processed_data. In the /cmd_vel topic the velocities for each agent are published. The Nexus robots in Gazebo are subscribed to this topic causing them to move in the right direction, according to the control law. In the /n_1/processed_data useful outputs are published. The /store_data node receives this data and writes the results in a separate .csv file.

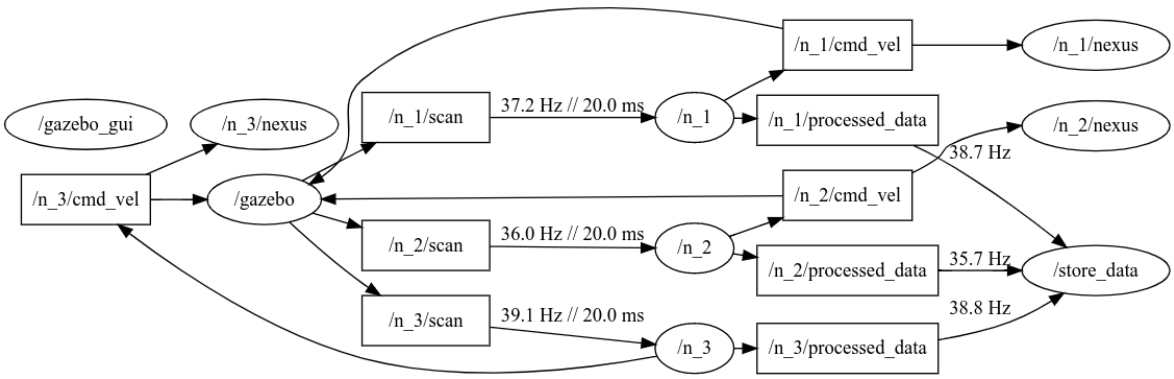


Figure 4.3: Computation graph of ROS/Gazebo.

Chapter 5

Results

This chapter will cover the results of the performed simulations. A Monte Carlo simulation is performed to validate and analyse the performance of the proposed control law. First, a set of starting positions for the nexus robot were generated according to a certain probability distribution function. After that, the success rate and convergence time for various desired theta values and gains are analysed by means of simulating N number of times. The best results are further analysed with a greater number of trials. Finally, the robustness of the algorithm is measured by adding noise to the laser scanner.

5.1 Initial Conditions

5.1.1 Starting positions

The $x-$ and y-coordinates for the starting position of the three nexus robots are generated by means of random uniform distribution. The interval $-2 \leq x, y \leq 2$ is used, since it is assumed that neighboring robots are within a range of four meter. This also makes sure that the robots will be spawned well within the maximum range of the laser scanner, which is 10 meters. To avoid direct collision between the nexus robots, the minimum distance between two robots should be 0.25 meters. This constraint was added when generating the starting positions. Considering the three robots have different starting positions each simulation, θ_{ij}, with $i, j = 1, 2, 3$ and $i \neq j$ also varies each starting position.

5.1.2 Successful simulations

The success rate of the simulations in various environments is an important performance indicator for validating the control law. To be able to determine whether a simulation is considered successful, the conditions for a successful simulation must be specified. In this research a simulation is successful if for all three robots the error of theta $e_{\theta,ij} \leq 2$, where $e_{\theta,ij}$ is defined as $|\theta^*_{ij} - \theta_{ij}|$. Thus, the difference between the desired theta θ^*_{ij} and the actual theta θ_{ij} must be smaller than two at the end of a 30 seconds simulation.

An error of 2° is chosen because this was the smallest variation that could be interpreted from the laser scanner. Note that all three nexus robots must converge, i.e if one nexus robot does not (fully) converge, the simulation is considered as a fail. By running several test trials there

was observed that most simulations converge within 20-25 seconds. If the nexus robots did not converge within this time interval, it was highly unlikely that they would converge at all. A simulation time of 30 second was chosen based on these observations. Therefore, it is safe to assume that if the nexus robots do not converge within the simulation time, they would have failed regardless.

5.1.3 Gain

The value of the gain K has an impact on all variables in the control law. Increasing the value K accelerates the speed of convergence. However, such a strategy also leads to the robustness degradation with respect to the measurement noises and an initial overshooting (Luenberger, 1979).

Based on (Chan et al., 2019) and observations during test trials, a K of 5 and 10 is used in the simulations. By using multiple gains, a performance comparison can be made.

5.2 System Behaviour

In this section the results of the performed simulations for will be shown. The performance of the system is analysed based on the following results:

- The number of successful simulations for varying θ^*_{ij} values
- The convergence time for varying θ^*_{ij} values
- The impact of the gain on the simulations.

5.2.1 Triangle to triangle formation

The first example illustrates how the formation converges to an arbitrarily specified equilateral triangle (so long as the triangle is feasible) given a random initial triangle configuration. The desired triangle formation in this case in characterized by $\theta^*_{ij} = 20$. The formation motion is illustrated in Figure 5.1

The initial position of the three agents are randomly distributed and the figure illustrates the trajectories of each agent as the formation converges upon the desired shape. This example shows that the control law can generate arbitrary equilateral triangle formations.

However, the control law is applicable to all types of triangles. Figure 5.2 shows the convergence of the three robots from a random initial triangle to a random non-equilateral triangle. During this experiment random θ^*_{ij} values are generated, ranging from 5-50. This thesis is mostly focused on equilateral triangles because the performance can be analysed better when the desired triangular formation is known.

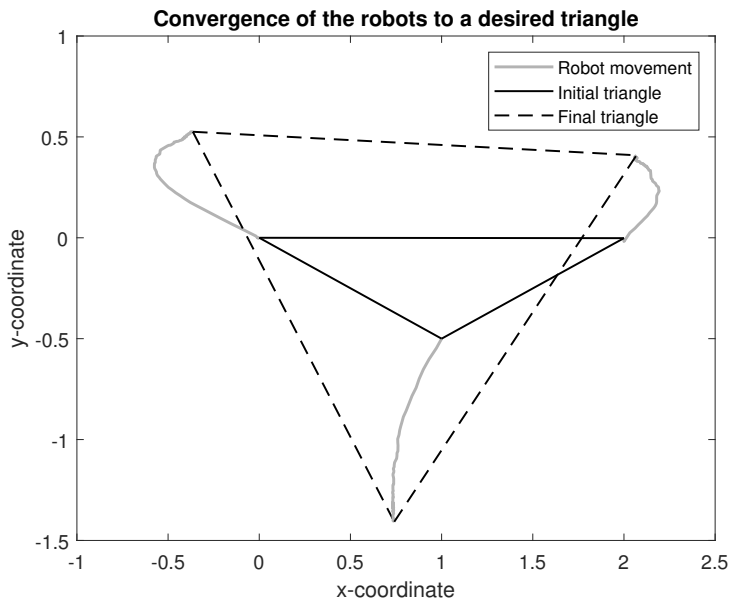


Figure 5.1: The motion of a triangular formation to an equilateral triangle, consisting of three mobile agents starting in a random triangle.

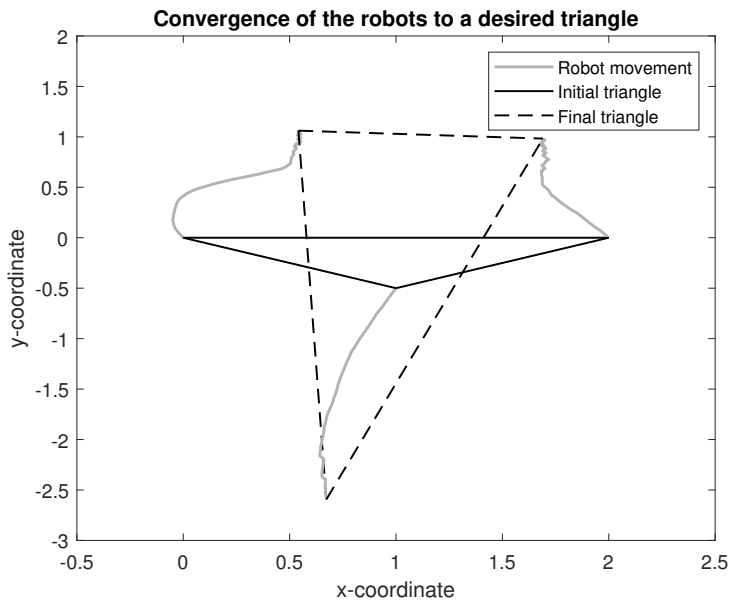
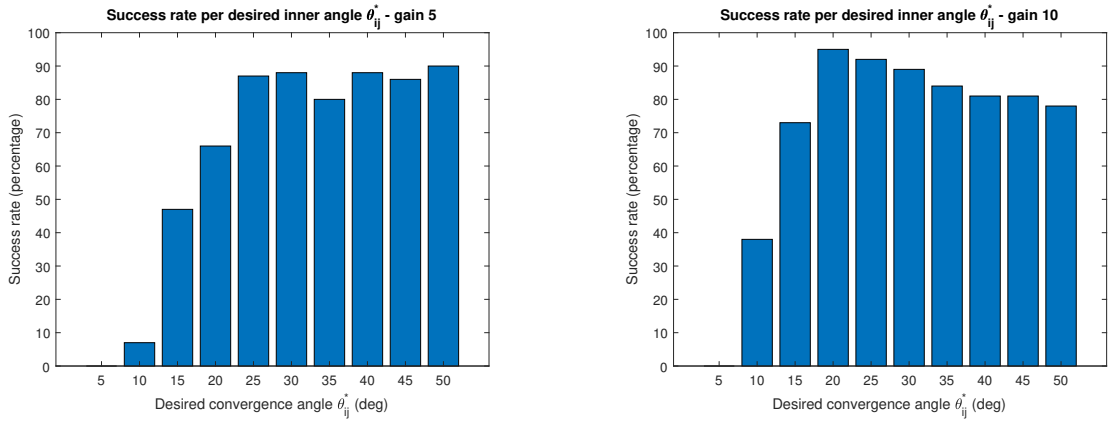


Figure 5.2: The motion of a triangular formation to a non-equilateral triangle, consisting of three mobile agents starting in a random triangle.

5.2.2 Successful simulations

The success rate for random non-equilateral triangles with $K = 10$ has been determined at 79% out of 100 simulations. To be able to conclude more about the performance of the control algorithm, an experiment with known θ^*_{ij} has been performed. The simulations ran

for varying θ^*_{ij} values to determine the success rate per desired inner angle. The results of this experiment are shown in Figure 5.3. Figure 5.3a shows the results for a gain of 5, while



(a) Success rate for varying θ^*_{ij}, with $K = 5$.

(b) Success rate for varying θ^*_{ij}, with $K = 10$.

Figure 5.3: Success rate for varying θ^*_{ij}, with $K = 5$ and $K = 10$.

Figure 5.3b shows the results for a gain of 10. On the x-axis the desired inner angle is given, ranging from $\theta^*_{ij} = 5$ to $\theta^*_{ij} = 50$. This range is chosen because the robots would collide with a lower θ^*_{ij} value, and the maximum θ^*_{ij} value keeps the robots just within the distance range of the laser scanner. For each desired inner angle, the according success rate in percentages is given on the y-axis. The simulation ran 100 times for each variation in θ^*_{ij} and K.

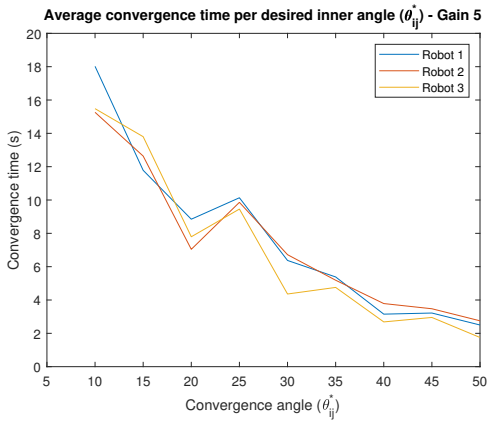
The bar charts indicate that for both gains the system does not converge if θ^*_{ij} is set at 5 degrees. A substantial difference can be observed at $\theta^*_{ij} = 10$. With $K = 5$ only 7 percent of the iterations were successful, while the success rate with $K = 10$ raised to 38 percent. Thus, a slow convergence is not beneficial when the desired inner angle is low. Simulations with a gain of 10 are better performing for the lower θ^*_{ij} values in terms of success rate, with $\theta^*_{ij} = 20$ having the best outcome of 95 percent success. However, there is a decline in success rate when the θ^*_{ij} values grow larger. This already happens when $K > 20$.

In contrary to the high gain, the low gain performs better as θ^*_{ij} increases. Also, the success rate is more stable when $K \geq 25$ in comparison to the higher gain. The highest θ^*_{ij} gives the best success rate, namely 90 percent.

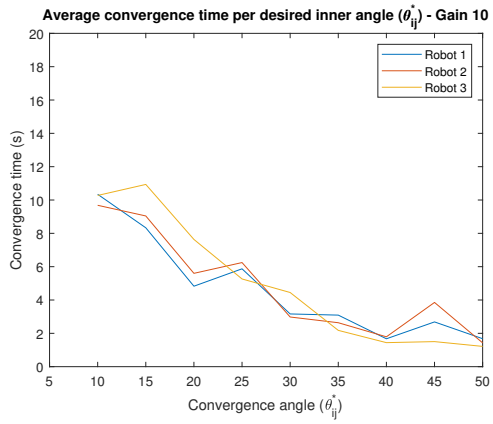
5.2.3 Convergence time

The convergence time measures how fast a system reach the state of convergence. It is defined as the time it takes to detect changes in the network topology and reconfigure the topology correctly. The system is in state of convergence when all three nexus robots have stabilized over time. Convergence time is an important performance indicator, slow convergence is often undesirable. Figure 5.4 shows the average convergence time of the three nexus robots per θ^*_{ij} interval, for both gains.

Figure 5.4 shows that the three robots behave the same over time. The higher gain results in a much a lower convergence time. The difference is especially significant for lower θ^*_{ij} values. Both graph follow the same trend, where the convergence time decreases as θ^*_{ij} grows larger.



(a) Average convergence times per robot with $K = 5$.



(b) Average convergence times per robot with $K = 10$.

Figure 5.4: Average convergence times for varying θ^*_{ij}, with $K = 5$ and $K = 10$.

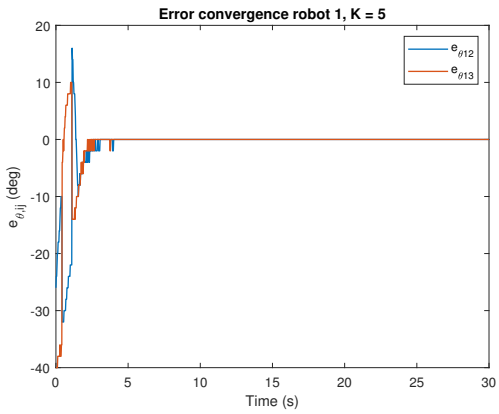
It can be observed that odd θ^*_{ij} values often show peaks above the trend line. Table 5.1 gives an numerical overview from the results in Figures 5.3 and 5.4.

Table 5.1: Comparison of the success rate and average convergence time for varying θ^*_{ij}, with $K = 5$ and $K = 10$.

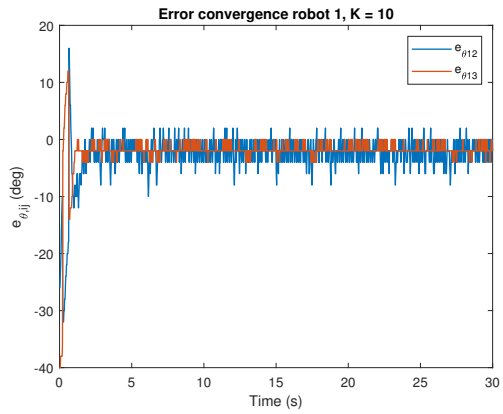
	K = 5		**K = 10**	
	Success rate (%)	Convergence time (s)	Success rate (%)	Convergence time (s)
$\theta^*_{ij} = 5$	0	-	0	-
$\theta^*_{ij} = 10$	7	16.25	38	10.10
$\theta^*_{ij} = 15$	47	12.74	73	9.44
$\theta^*_{ij} = 20$	66	7.90	95	6.02
$\theta^*_{ij} = 25$	87	9.82	92	5.79
$\theta^*_{ij} = 30$	88	5.82	89	3.53
$\theta^*_{ij} = 35$	80	5.11	84	2.64
$\theta^*_{ij} = 40$	88	3.21	81	1.63
$\theta^*_{ij} = 45$	86	3.22	81	2.68
$\theta^*_{ij} = 50$	90	2.33	78	1.44

5.2.4 Gain

In Section 5.2 is illustrated how the gain impacts the success rate and convergence time of the system. When a gain is too high, the system can have difficulties to converge to the desired topography. Especially for high θ^*_{ij} values, the system was affected negatively. An example of this is given in Figure 5.5. Although the θ^*_{ij} and starting positions are identical, a clear dissimilarity can be observed. For $K = 5$, the system reaches a state of convergence after a few seconds. For $K = 10$, the system shows continuous oscillations and ultimately does not converge. This behaviour does not occur consistently at every simulation. However, the chance increases with higher θ^*_{ij} values.



(a) Error convergence in simulation 66, with θ^*_{ij} = 50, K = 5.



(b) Error convergence in simulation 66, with θ^*_{ij} = 50, K = 10.

Figure 5.5: Comparison of the error convergence in simulation 66.

5.3 System analysis for best theta and gain

Since the experiments in Section 5.2 are performed with a relatively small sample size, this number will be increased to get more reliable results. In these experiments, the simulations ran 800 times. This number is chosen based on the maximum number of rows that can be stored in Excel, which is 2^{20} (or 1,048,576). Because every simulation lasts 30 seconds and the frequency of the laser scanner is 40 Hz, each simulation will write 1200 row values to a separate file.

The error convergence of the best performing θ^*_{ij} and K in Section 5.2 will be further analysed.

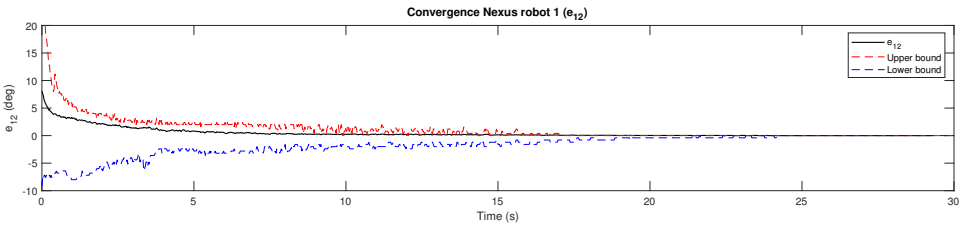
5.3.1 Successful simulations

In Section 5.2.1, the success rate for $\theta^*_{ij} = 20$ and $K = 10$ was determined at 95%. When the same initial conditions were used with a sample size of 800, a success rate of 89.25% was obtained (714/800 simulations).

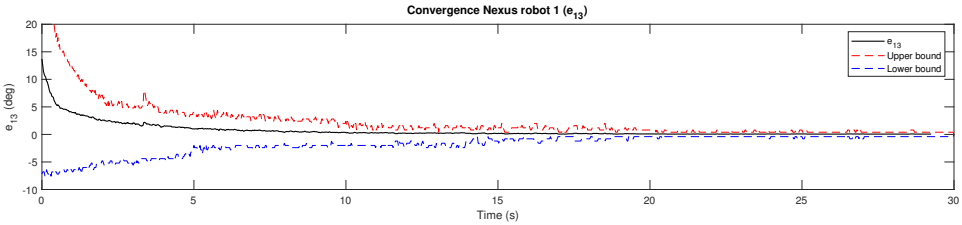
5.3.2 Error convergence

Figures 5.6, 5.7, 5.8 show the mean error per time step $e_{\theta,ij}$, i = 1, 2, 3 and $i \neq j$. The lower and upper bound indicate the maximum positive and negative error per time step. Since $e_{\theta,ij}$ is calculated as $\theta_{ij} - \theta^*_{ij}$, the error is negative when θ^*_{ij} is larger than θ_{ij}. This happens when the triangular formation expands instead of shrinks. However, the mean error is plotted as $|\theta_{ij} - \theta^*_{ij}|$.

Table 5.2 gives an overview of the important values of the figures. Settling time T_s is the time required for an output to reach and remain within the given error margin. y_{ss} is the average time it takes for 95% of the simulations to reach an error of zero, steady state.

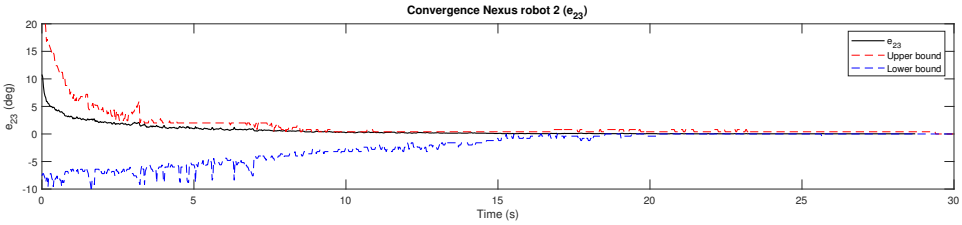


(a) Error convergence over time of robot 1 to robot 2.

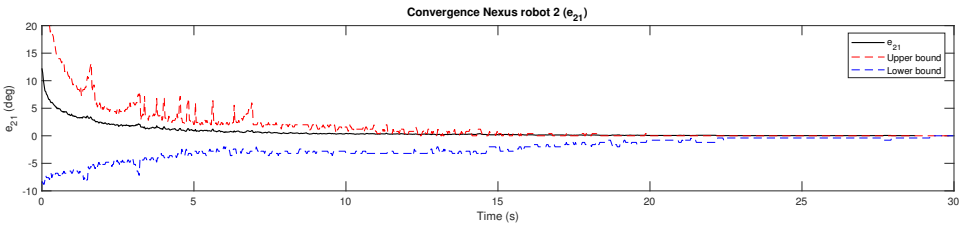


(b) Error convergence over time of robot 1 to robot 3.

Figure 5.6: Error convergence over time of robot 1.



(a) Error convergence over time of robot 2 to robot 3.



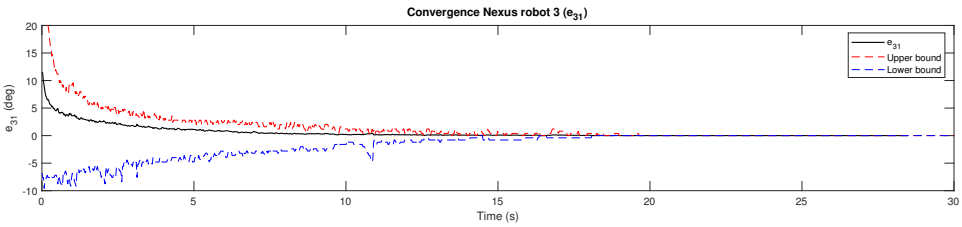
(b) Error convergence over time of robot 2 to robot 1.

Figure 5.7: Error convergence over time of robot 2.

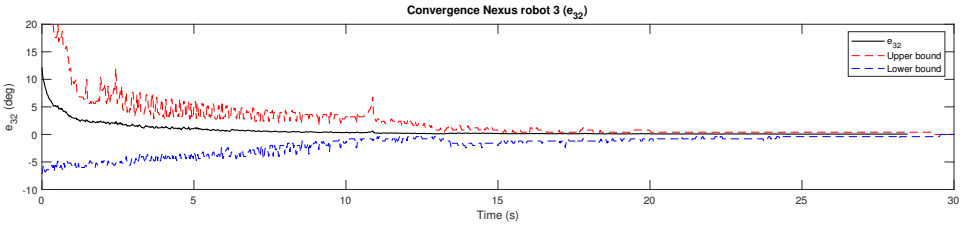
The settling times indicate a fast convergence to the desired triangle. This can also be observed in the given figures. The robots show the same behaviour for each convergence error e_{ij}

5.4 Robustness

To attain a reasonable degree of quality assurance, a robustness test is performed. The purpose of robustness testing is to determine if the addition, removal or substitution of variables changes the outcome of the previously obtained results. Since the simulations in Section 5.2 are performed in a 'perfect' world, the laser data will now be disturbed by Gaussian noise. Gaussian noise is used because it mimics the effect of many random processes that occur in



(a) Error convergence over time of robot 3 to robot 1.



(b) Error convergence over time of robot 3 to robot 2.

Figure 5.8: Error convergence over time of robot 3.

	T_s	y_{ss}
e_{12}	2.155	17.00
e_{13}	2.875	18.80
e_{23}	2.400	19.15
e_{21}	3.275	19.65
e_{31}	2.650	16.98
e_{32}	2.475	18.15

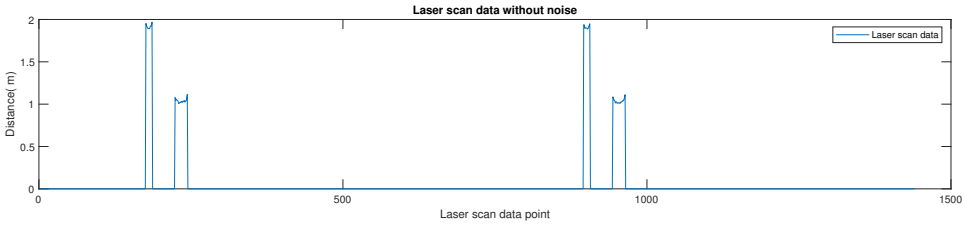
Table 5.2: Mean settling times and steady state times for 714/800 successful simulations.

nature. In this section a Gaussian noise with a mean of $\mu = 0$ and a standard deviation of $\sigma_n = 1$ is added to the incoming laser scan data.

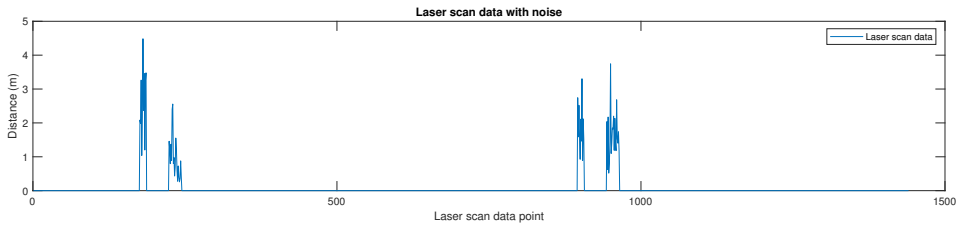
Figure 5.9 shows a comparison between the regular laser scan data and the disturbed data. The peaks indicate the distance to the other robots. As explained in Section 4.2, 720 data points represent an angular range of 360°. Thus, two cycles (or 50 ms) of laser scan data is illustrated in the figure below.

The effect of the disturbance on the error convergence is illustrated in Figure 5.10. It indicates the mean difference in error from robot 1 to robot 2, when noise is added and not. Figure 5.10 shows that the noise does influence the error when the robots have converged, which is after around 13-14 seconds.

The same starting positions have been used with $\theta^*_{ij} = 20$ and $K = 10$. Without added noise, the simulations had a success rate of 94%, which is fairly similar to the success rate obtained in Section 5.3. Since this experiment has a smaller sample size and different starting positions, a small deviation can be expected. When the noise was added, the success rate dropped slightly to 92%.



(a) Undisturbed laser scan data.



(b) Laser scan data disturbed with Gaussian noise.

Figure 5.9: Comparison between undisturbed and disturbed laser scan data.

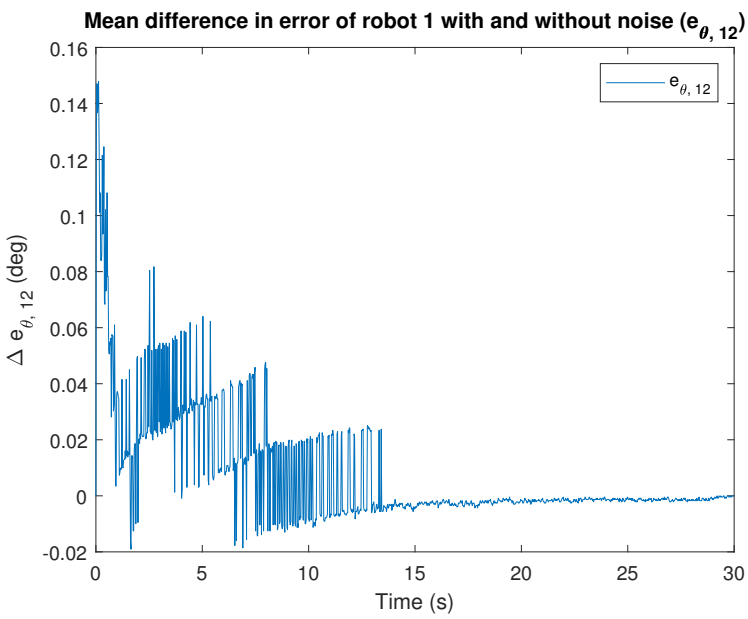


Figure 5.10: Error difference when noise is added, for 100 simulations.

Chapter 6

Discussion

This section discusses the results of the experiments performed in Chapter 5. The findings will be compared to the claims in existing literature and explanations are given for unexpected outcomes.

In the first experiment is shown that the average success rate when converging to a random non-equilateral desired triangle is 79%. This is a higher than the average success rate with $K = 10$ for the varying equilateral triangles, which is 71.1%. The discrepancy can be explained by the success rate of zero when $\theta^*_{ij} = 5$. If a random desired triangle is generated it is less likely that both θ^*_{ij} values are low, which would result to more failed simulations.

The results from the varying θ^*_{ij} values indicate that a successful convergence depends on the gain and the desired inner angle. The study demonstrates a correlation between the gain and the convergence time. As expected, a higher gain results generally in a lower convergence time. This corresponds to (Luenberger, 1979), who claims that increasing the gain accelerates the speed of convergence. The article also states that such strategy leads to performance degradation. During some simulations continuous oscillations were observed, which caused the system not to converge. However, a higher gain does not lead to worse results by definition. In the simulations can be observed that for a high gain in combination with a high θ^*_{ij} value, the robots either converge very fast and stable, or overshoot and keep oscillating. This resulted in a slightly lower success rate. However, the lower gain prevents the robots from overshooting, which resulted in a better success rate.

It is important to mention that the value of θ_{ij} and θ^*_{ij} depends in this thesis on the radius of the laser scanner. This has a relatively small diameter in comparison to the features that are used on both sides of the robot in (Chan et al., 2019). In this research a θ_{ij} of 5 corresponds to an inter-agent distance of approximately 9 meters. Accordingly, a θ_{ij} of 50 corresponds to approximately 0.80 meters. The θ_{ij} values are thus dependent on the size of the detected laser scanner and only applicable for this simulation setup.

Remarkable is that the odd θ^*_{ij} values perform generally worse than the even θ^*_{ij} values. This can be explained by the fact that the imported laser scan data is published with intervals of two. This means that the odd θ^* values have effectively an error margin of one, instead of two. As a consequence, the system can never fully converge for odd θ^* values.

After increasing the sample size, the control algorithm performed a little worse. Obviously, the large sample size included a lot of different starting positions. However, with an equal K and θ^*, the starting positions were not the only key factor for success. There was observed that during some simulations the laser scanner gave wrong values, such as high fluctuations in distance measurements while the robot stayed equally far away. This influenced the success rate, but no definite answer can be given by how much.

The control algorithm shows robustness to disturbance in laser scan data. This performance can be explained by the fact that the control input is for a large part based on the error between the current inner angle and the desired angle. This value is calculated with the indices of the laser scan data and do not include distance measurements.

Chapter 7

Conclusion

This thesis presented a distributed formation control algorithm for the movement of robots to a desired triangular formation. This control law is based on the bearing between the agents. With the use of a laser scanner, the distances and bearings were determined.

The first experiment showed that the control algorithm works for both equilateral and non-equilateral triangles. The success rate for converging from random initial positions to non-equilateral triangles was determined at 71.1%. Thereafter, the research focused on the performance of equilateral triangles. There is shown that successful convergence depends on the desired inner angle and the gain. The best result was a 95% success rate with $\theta^*_{ij} = 20$ and $K = 10$. The convergence to non-equilateral triangles had better results than the average success rate of equilateral triangles. However, it performed worse than the most optimal combination of K and θ^*_{ij} tested in this thesis.

In accordance with the literature, the convergence time decreased when the gain was higher. A higher gain is desired when the desired inner angle is small (inter-agent distance is large), and a lower gain is desired when the desired inner angle is high (inter-agent distance is small).

After a new experiment with a larger sample size can be concluded that the success rate of 89.25% might be more reliable. The results indicate that on average the robots converge to the desired triangle within 2-3 seconds and 95% of the simulation converge to a steady state of zero within 20 seconds, given a successful simulation.

From the robustness analysis can be concluded that adding disturbance to the laser scanner does not affect the success rate of the control algorithm. However, the system needs a bit more time to converge to the desired triangle.

In conclusion, the research goal *"To develop and to numerically analyse the performance and robustness of an angle-constrained distributed formation control algorithm based on local bearing measurement"* is achieved. Nonetheless, more research is required to increase the success rate and apply the control law in a real-life scenario.

Chapter 8

Further research

This research focused on the implementation of an angle-constrained distributed formation control algorithm. This chapter gives potential possibilities for further research.

Due to time limitations, the experiments have mostly been performed on equilateral triangles. Although there is shown that the control algorithm can also be used for non-equilateral triangles, the research can be extended by performing more experiments on non-equilateral triangles. In this research the success rate was determined based on the desired inner angle, by using random initial positions. Further research can include determining the success rate based on the starting positions, where the starting θ_{ij} values are known Depending on the application in a real-life scenario, the robots will also need to converge to different non-equilateral formations.

To make this control law feasible in practical, some more solutions for limitations must be found. In this research an empty world is assumed, causing robots only to detect each other. In a real-life scenario the laser scanner will also detect objects in the environment. Therefore, a robot must be able to distinguish a neighboring robot in a dynamical environment. This can be achieved by adding constraints in the control algorithm or with the use of an additional sensor, such as a camera. In addition, the simulations are performed on robots with omni-directional wheels. These robots have a holonomic property, while other vehicles may have not.

The simulations in this thesis are performed in a 2D-world. Since drones are often used in autonomous operations, another interesting topic can be to extend the research from a 2D-world to a 3D-world. This allows drones to use the algorithm as well.

Finally, there is concluded that a successful simulation depends on the gain combined with the desired inner angle. This means that the gain can also be optimized for each θ^*_{ij}. An optimized gain may result in a higher success rate for each formation.

Bibliography

N. Chan, B. Jayawardhana, and H. Garcia de Marina. Controlling a triangular formation using features. 2019.

Wuwei Chen, Hansong Xiao, Qidong Wang, Linfeng Zhao, and Maofei Zhu. *Integrated vehicle dynamics and control*. John Wiley & Sons, 2016.

Anh-Duc Dang, Hung M La, Thang Nguyen, and Joachim Horn. Distributed formation control for autonomous robots in dynamic environments. *arXiv preprint arXiv:1705.02017*, 2017.

Alessandro Farinelli, Luca Iocchi, and Daniele Nardi. Multirobot systems: a classification focused on coordination. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(5):2015–2028, 2004.

Avinash Gautam and Sudeept Mohan. A review of research in multi-robot systems. 08 2012. doi: 10.1109/ICIInfS.2012.6304778.

Hengqing Ge, Guibin Chen, and Guang Xu. Multi-auv cooperative target hunting based on improved potential field in a surface-water environment. *Applied Sciences*, 8(6):973, 2018.

Kurt Geihs. Engineering challenges ahead for robot teamwork in dynamic environments. *Applied Sciences*, 10(4), 2020. ISSN 2076-3417. doi: 10.3390/app10041368. URL `https://www.mdpi.com/2076-3417/10/4/1368`.

David G Luenberger. Introduction to dynamic systems; theory, models, and applications. Technical report, 1979.

Kwang-Kyo Oh, Myoung-Chul Park, and Hyo-Sung Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015.

Lynne E. Parker. Multi-robot team design for real-world applications. In Hajime Asama, Toshio Fukuda, Tamio Arai, and Isao Endo, editors, *Distributed Autonomous Robotic Systems 2*, pages 91–102, Tokyo, 1996. Springer Japan. ISBN 978-4-431-66942-5.

Michal Pěchouček and Vladimír Mařík. Industrial deployment of multi-agent technologies: review and selected case studies. *Autonomous agents and multi-agent systems*, 17(3):397–431, 2008.

Wei Ren and Randal Beard. Virtual structure based spacecraft formation control with formation feedback. In *AIAA Guidance, Navigation, and control conference and exhibit*, page 4963, 2002.

Wei Ren and Nathan Sorensen. Distributed coordination architecture for multi-robot formation control. *Robotics and Autonomous Systems*, 56(4):324–333, 2008.

Shiyu Zhao and Daniel Zelazo. Bearing rigidity theory and its applications for control and estimation of network systems: Life beyond distance rigidity. *IEEE Control Systems Magazine*, 39(2):66–83, 2019.