Master Thesis 62

Subject

Bearing-only Formation Control with Limited View Constraints

Daniel Frank

July 9, 2018

Prüfer: *Prof. Dr.-Ing. Frank Allgöwer*
Betreuer: *Asisstant Prof. Ph.D. Daniel Zelazo*
M.Sc. Philipp Köhler

Institut für Systemtheorie und Regelungstechnik
Universität Stuttgart
Prof. Dr.-Ing. Frank Allgöwer

Abstract

In this work we tackle the problem of limited visual sensors in bearing-only formation control. Existing control strategies in the literature are able to stabilize formations for arbitrary number of agents but require a sensing system that is able to track the entire surroundings. This assumption can not be made in all real-world applications, therefor we look at the bearing-only controller constrained by limited visual sensing. We propose a heading controller, for the two agent case, that adjusts the direction of the visual sensor such that its neighbor stays inside the field-of-view (FOV). The closed loop, including the position and the heading, is then shown to reach the desired bearing if at least one agent can sense the other initially. Analytically this is proven for the two agent case, we provide numerical simulations for more than two agents. This work also includes experimental results on *TurtleBotII* robots, motivated by simulations on a unicycle model.

Zusammenfassung

Diese Arbeit beschäftigt sich mit dem begrenzten Sichtfeld von Kamerasensoren, die dazu dienen Formationen zu stabilisieren. Basierend auf relativen Winkelinformationen zu Nachbaragenten wurde ein Regler entwickelt der es ermöglicht steife Formationen, die als ein Set von Einheitsvektoren beschrieben sind, zu stabilisieren. Um diesen Algorithmus auf einem System zu implementieren, wird jedoch ein Sensor benötigt der sein komplettes Umfeld wahrnehmen kann. Durch diese Annahme wird die Anzahl der möglichen Systeme drastisch reduziert. Die vorliegende Arbeit befasst sich mit dem Problem des begrenzten Sichtfelds, bei gleichbleibendem Positionsregler. Dabei wird im ersten Schritt die Problemstellung erläutert und die Zustandsgleichungen um einen Zustand, der die Blickrichtung repräsentiert, erweitert. Für diesen Zustand wird dann ein Regler entworfen der eine Verbindung zwischen zwei Agenten garantiert. Mithilfe dieses Reglers und dem Positionsregler kann gezeigt werden, dass die gewünschte Formation, im zwei Agenten Fall, erreicht wird. Die Startbedingungen sind dabei so zu wählen, dass ein Agenten den Andern "sehen" kann. Gestützt wird die mathematische Analyse von numerischen Simulationen, die selbst Formationen mit mehr als zwei Agenten stabilisieren können. Des Weiteren wurden Experimente an Boden Robotern durchgeführt (*TurtleBotII*), die unsere Analyse weiter bestärken. Dabei ändert sich das dynamische

Verhalten der Agenten von einem Integrator zu der Dynamik eines Einrads. Die Experimente wurden im *CoNeCt*-Labor an der Fakultät für Luft- und Raumfahrttechnik am Technion durchgeführt.

Contents

1 Introduction

Multi-agent systems have gained a significant amount of research in both control and robotics communities. The application as well as the theoretical challenge in controlling multiple agents at the same time is an interesting, widely open research question.

Multi-agent systems are mainly inspired by swarm animals like birds or fishes. Swarms are moving in certain dynamic shapes, such that they can travel over long distances and through different environments. Applications of multi-agent systems can be found for example in underwater exploration, surveillance, and deployment in space [6,7]. One major advantage of multi-agent systems is the complexity of each vehicle, where one complicated agent can be replaced by a collection of simple agents that together are capable of solving non-trivial tasks. In harsh environments, where humans can not enter multiple robots would be able to handle certain tasks by combining their tools. If one agent provides the tools for digging another robot might support with cargo. Not only different tools can be combined to solve difficult tasks, information from multiple sensors can for example provide a greater view. Drones that fly over certain areas can combine their cameras to track trapped persons after a natural catastrophe. Great interest in the application of multi-agent systems is available in the filed of agriculture. Autonomous vehicles that are able to drive over huge farms would replace one tractor that has to be controlled by a human. The agents could be smaller and be used more flexible but maintain the area that can be processed at the same time.

Formation control is one of the most actively studied topics within the field of multi-agent systems. Developing algorithms for multiple agents to form a specific geometric pattern can be a good starting point for reaching more complex objectives. In general the goal in formation control is to move an arbitrary number of agents, such that they form a predefined shape. Before stating the problem of this work, we want to point out the distinctions that are made in formation control.

Multiple agents in an environment that should work together need to have some kind of information about the other agents or knowledge about the network. Since in general, we look at an arbitrary number of agents it is not practical to share information with every agent in the network. For a great number of agents a network that is capable to send and receive messages to and from all of its nodes will soon be very complex. In this work we look at a case where the agents know who its neighbors are and can sense information about them, if they are inside the limited field-of-view (FOV).

The information we are able to obtain from our neighbors is crucial for developing algorithms in formation control. Information can be obtained by local sensors or provided over a communication link to other agents or a centralized control unit. If an agent provides its state information over a communication network, then the data represents the true values. Sensing information, on the other hand can not guarantee to extract the right state information, since they suffer from limitations. Common sensor information that are used to do formation control are distances and bearings. Range sensors are able to receive relative distance information to objects around them. Often the distance can only be obtained in a certain area around the sensor, once object are further away the range information is not available anymore. Bearing information refers to unit vectors that can be obtained by visual sensors such as cameras. Computer vision algorithms are capable to extract relative angle information to certain objects within a camera frame, the angles are then used to extract unit vectors. Visual sensors often are not able to track all the surroundings, but only a limited area, it can obtain information about object that are inside the field-of-view, but once objects are outside, no information is available. Due to the limitations of the sensor a sensing network is not necessarily static, also the sensing link is not guaranteed to be undirected. Here undirected refers to the case where two agent are able to sense each other, however if we look at limited sensor information this assumption is not guaranteed, since there are setups where one agent can sense its neighbor but not the other way round. In this thesis we look at controller that uses only information provided by a visual sensor (e.g. bearings) and no communication link that provides state information from other agents.

Another distinction we want to point out, is the calculation of the input signal for the actuators. One way of controlling multiple agents is to setup

Controller

Figure 1.1: Centralized controller that sends corresponding signals to each agents (left). Decentralized controller where each agents calculates its own input signal (right).

a centralized controller that communicates with each individual agent and sends the corresponding control signal. In that case the agents do not require powerful hardware to do complex calculations. Another way to do formation control is to decentralize the calculation of the control signal, such that we do not need a centralized, powerful controller with a communication link to each individual. Our approach uses a decentralized controller that does not require any communication, an illustration of the two different approaches is shown in Figure 1.1.

The survey paper on multi-agent formation control, [14], provides a high-level classification of different formation control strategies. They include position-based, displacement-based, distance-based, and bearing-based approaches, each depending on the sensing mediums available to the agents. Distance-based approaches were extensively studied in [2], [10], [13] and [20]. Bearing-based formation control has become more popular in recent years since visual sensing can be used to extract bearing information. Compared to range sensors, cameras are cheaper, lighter, and require less power. Estimating relative position information from bearings has been studied in [21], whereas [24] used bearing information directly to stabilize formations for an arbitrary number of agents. In both articles, it is assumed that the visual sensors can cover the entire surroundings to extract relative angles to its neighbors. However, this assumption is not realistic in applications using ground robots or UAVs. Usually cameras are only able to record a bounded area defined by its field-of-view (FOV). If neighbors stay inside that bounded area the agent is able to sense them, once they are outside no information can be provided. This leads to state-dependent sensing graphs where the neighbors are not static - they depend both on the position and orientation of the sensing agent, and the FOV constraints of the sensor. In a case where a neighbor enters the FOV the sensing graph gains a new edge and once an agent moves outside the FOV an edge gets lost in the graph. This setup makes the analysis more complex. Motivated by this real-world problem, a number of approaches have been considered in the literature. Formation control over directed sensing graphs have been studied in [9] for distance constrained formations, and in [17, 18, 23] for bearing formations. Limited sensor information in a distance based setup has been studied in [20] and [16], and required a distributed estimation of various quantities. The limited FOV problem has been studied in other multi-agent problems. The authors of [3] study consensus on containment with limited FOV. In the work of [5] a limited FOV is introduced but the sensing graph remains static and range information is needed.

In this work, we aim to directly address the formation control problem using bearing sensors with limited FOV constraints. Our starting point is the bearing-only formation control strategy proposed in [24], which we augment by introducing the state-dependent FOV constrained bearing measurement. We assume the sensor is mounted rigidly to the body frame of the robot, and we also propose a control for the heading, corresponding to the pointing direction of the sensor, for the robot. Our main contributions can be stated as follows:

i) We propose a novel controller for the heading direction based only on sensed bearing measurements. This controller guarantees that once an agent enters the FOV of the sensor, it will remain inside for the remainder of the trajectories.

ii) We provide a complete characterization of the different equilibrium configurations attainable by the two agent case and show that if at least one agent is initially sensed the desired formation is a stable equilibrium.

We also demonstrate the results with a number of simulation examples. While this work focuses on the two-agent case, we also provide simulation examples for the three and four agent case indicating the promise of this approach for larger formations. The simulations are then validated by experiments on *TurtleBotII* robots. Here we used the idea presented in [22] to change the dynamics from integrators to a unicycle.

This thesis is organized as follows. In the remainder of this chapter we introduce the bearing-only control strategy and the agent dynamics, we will also give a short overview of bearing rigidity theory, which is crucial for an arbitrary number of agents. In Chapter 2 we present the problem for the two agent case with limited FOV constraint. We then show a complete stability proof for the two agent case; here we also propose a controller for the heading direction. The analysis is supported by some numerical simulations and also extended for a higher number of agents. Chapter 3 gives an overview of the Cooperative Networks and Controls Lab (*CoNeCt*) in the Faculty of Aerospace Engineering at the Technion. Here we explain the tools that we used for our experiments and then provide results on *TurtleBotII*robots. Concluding remarks are given in Chapter 4 where we summarize our results and give an outlook on possible future research topics.

1.1 Bearing-Only Formation Control

Bearing-only formation control was first introduced in [24] where the authors provide an almost global stability proof for an arbitrary number of agents. In this chapter we will discuss the details about that controller, by first introducing the required notation, then presenting the control strategy and conclude with an overview on bearing rigidity theory.

1.1.1 Preliminaries

The objective of a bearing only formation controller can be stated as the following problem statement:

Problem 1.1.1. *Given a desired geometric pattern and an initial starting point of the agents, we want to design a controller for every agent that uses only bearing information and stabilizes the desired formation.*

In this section we introduce the tools that we will use in the controller and give some remarks on the notation used in this thesis.

The position of the agents with respect to a global reference frame follows as

$$p_i(t) = \begin{bmatrix} x_i \\ y_i \end{bmatrix},$$

where x_i and y_i describe the position in the plane and the subscript indicates the agent. In this work we focus on ground robots and therefor have $p_i \in \mathbb{R}^2$.

In general the theory of bearing-only formation control in [24] also works for higher dimensions $d > 2$ (here d indicates the dimension). Bearing information is a unit vector (in world coordinates) between two agents and can formally be stated as

$$g_{ij}(t) := \frac{p_j(t) - p_i(t)}{\|p_j(t) - p_i\|(t)} = \frac{z_{ij}(t)}{d_{ij}(t)} \qquad i,j \in \mathcal{V}. \tag{1.1}$$

In multi-agent systems the network connection can be described as an undirected graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ describe the node and edge set respectively. The number of edges follows as $e = |\mathcal{E}|$ and the number of vertices in a graph as $n = |\mathcal{V}|$. For an undirected graph $\mathcal{G}$, a connecting edge between two agents means that both agents can sense each other. The graph can be described through an incidence matrix $H \in \mathbb{R}^{exn}$, where the elements $[H]_{ij} = \{-1,0,1\}$ depend on the graph. By choosing arbitrary directions of the edges, it follows that $[H]_{ij} = 1$ if vertex j is head of edge i, $[H]_{ij} = -1$ if vertex j is tail of edge i and in all other cases $[H]_{ij} = 0$. We refer to [8] for more details.

The vertices that are connected through an edge to the same vertex are called neighbors $\mathcal{N}_i := \{j \in \mathcal{V} | (i,j) \in \mathcal{E}\}$. In general the neighbors are static and follow from the incidence matrix.

We now introduce a projection matrix that we will use in the controller proposed in the next section,

$$P_{g_{ij}(t)} = I_2 - g_{ij}(t) g_{ij}(t)^T.$$

By multiplying a non-zero vector to $P_{g_{ij}}$ the result will give an projection onto the orthogonal vector of g_{ij}, note that I_2 represents the identity matrix with dimension 2. The projection matrix is idempotent and therefor $P_{g_{ij}} = P_{g_{ij}}^2$ and $P_{g_{ij}}^T = P_{g_{ij}}$ holds. Moreover $P_{g_{ij}}$ is positive semi-definite and g_{ij} is in the nullspace of $P_{g_{ij}}$ (Null $\left(P_{g_{ij}}\right) = \text{span}\left\{\pm g_{ij}\right\}$). For more useful properties we refer to [24]. The introduced notation is now used to state the position controller.

1.1.2 Controller

The objective of the bearing-only controller is to move all agents in such a way that a desired formation is reached using only bearing information.

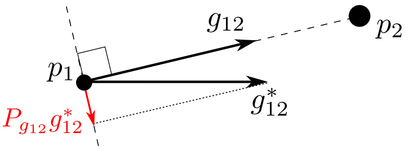


Figure 1.2: The projection matrix $P_{g_{12}}$ projects the desired bearing g_{12}^* onto the orthogonal of g_{12}, the control input then follows as $u_i = -P_{g_{12}} g_{12}^*$.

We define the desired formation as a set of bearing vectors, denoted by $g^* = \left[(g_i^*)^T \quad \cdots \quad (g_e^*)^T \right]^T$. The agents dynamics are modeled as single integrators

$$\dot{p}_i(t) = u_i(t), \tag{1.2}$$

where $u_i(t) \in \mathbb{R}^2$, such that we can control the velocity in $x-$ and $y-$direction directly, without any motion constraints. The objective of the bearing only formation control can formally be stated as:

Problem 1.1.2. *Given a desired formation g^* and an initial configuration $p(0) = \left[p_i(0)^T \quad \cdots \quad p_n(0)^T \right]^T$, design a control input u_i, such that the desired bearing $g_{ij}(t) = g_{ij}^*$ is reached, as $t \to \infty$ $\forall (i,j) \in \mathcal{E}$, using only bearing information.*

The controller u_i, that solves the problem is stated in [24] and follows as

$$u_i(t) = -\sum_{j \in \mathcal{N}_i} P_{g_{ij}(t)} g_{ij}^*, \qquad (i,j) \in \mathcal{E}, i \in \mathcal{V} \tag{1.3}$$

it uses only bearing information $g_{ij}(t)$. Here the projection matrix introduced in Section 1.1.1 is multiplied by the desired formation. The result can be seen in Figure 1.2, where the read vector indicates the projection result. Since the desired formation g_{12}^* refers to a horizontal line it makes sense to use the negative projection result ($-P_{g_{12}} g_{12}^*$) in order to reach the desired bearing. This Illustration shows an geometric interpretation of the bearing only controller for a two agent setup. The controller (1.3) was first introduced in [24] and was shown to be almost globally stable for an arbitrary number of agents (*Theorem 11*). There exists an initial condition from which the desired bearing will not be reached, we call this initial condition the undesired equilibrium point or undesired bearing. Since the nullspace of the projection matrix contains the positive and the negative bearing (Null $\left(P_{g_{ij}}\right) = \left\{\pm g_{ij}\right\}$), the undesired initial condition follows as $g_{ij}(0) = -g^*_{ij}$, $\forall(i,j) \in \mathcal{E}$. While the number of agents for controller (1.3) is arbitrary, the desired formation has to satisfy some geometric conditions, in order to be unique. If for example, the desired formation is a square then it is not enough to have a right angle in every corner since it does not guarantee that all the outer edges have the same length. In order to have a unique square, described by a set of desired bearings, a diagonal edge is needed, more details on the geometric conditions, including examples, are given in the next section. Such unique formation are called rigid formations.

1.2 Bearing Rigidity

The theory of bearing rigidity is extensively studied in [24], in this chapter we give an overview of the main results and refer to the corresponding proofs. In general bearing rigidity deals with the problem how a desired formation can be uniquely specified up to a scaling and translation factor. The main goal of this section is to get conditions on the desired formation such that they can be reached by controller (1.3) proposed in Section 1.1.2. In the sequel we call the specific desired formation, described by a set of unit bearing vectors, framework.

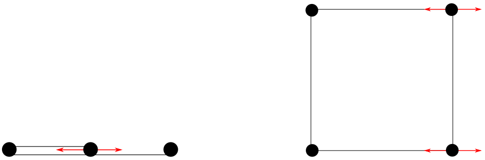
Before we are able to state the conditions for a specific framework the bearing function is defined as

$$F_B(p) := \begin{bmatrix} g_1^T & \cdots & g_e^T \end{bmatrix}^T \in \mathbb{R}^{2e}, \tag{1.4}$$

where 2 is the dimension of the position vector, referring to the plane. $F_B(p)$ describes all the bearings in the framework, the bearing rigidity matrix is then defined as the Jacobian of (1.4)

$$R(p) := \frac{\partial F_B(p)}{\partial p} \in \mathbb{R}^{2ex2n}. \tag{1.5}$$

An infinitesimal bearing motion of $\mathcal{G}(p)$ is a variation δp for which $R(p)\delta p = 0$ holds. Note that $\mathcal{G}(p)$ refers to a setup of the graph given by the position p. In contrast to distance based frameworks, bearing preserving motions are translations and scalings of the entire framework, which in the sequel are called trivial motions. Figure 1.3 shows examples where motions are not trivial, in Figure 1.3a a line with three agents is shown, the agent in



(a) Line with three agents.

(b) Square with four edges.

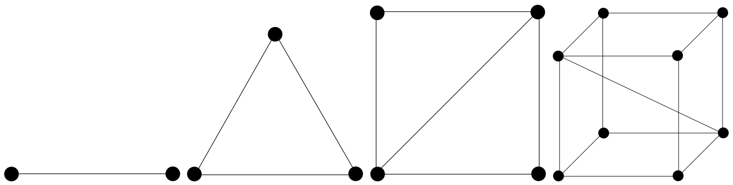
Figure 1.3: The red arrow shows non trivial motions that preserve the bearing but not the formation, frameworks like that are not infinitesimally bearing rigid.

the the middle can move freely on a horizontal line without changing the relative bearing to its left neighbor. The square illustrated in Figure 1.3b shows that if both agents on the right move at the same time on a horizontal line the bearing is preserved but the shape of the formation is not. Following *Definition 5* in [24] a framework is called infinitesimally bearing rigid if all infinitesimally bearing motions are trivial. The conditions therefor are stated in *Theorem 5* and follow as

$$\begin{aligned} \operatorname{Rank}\left(R(p)\right) &= 2n - 2 - 1 \\ \operatorname{Null}\left(R(p)\right) &= \operatorname{span}\left\{\mathbb{1} \otimes I_2, p\right\} = \mathit{textspan}\left\{\mathbb{1} \otimes I_2, p - 1 \otimes \bar{p}\right\}, \end{aligned}$$

where $\bar{p}$ refers to the position of the centroid. The all ones vector is indicated by $\mathbb{1}$ and $\otimes$ refers to the Kronecker product. Frameworks that are infinitesimally bearing rigid do have a unique shape (*Theorem 6* in [24]) and are invariant to space dimensions (*Theorem 7* in [24]). Infinitesimal bearing rigidity implies global beraing rigidity (defined in *Definition 4*) as well as bearing rigidity (defined in *Definition 3*), shown in *Theorem 5* together with *Theorem 3* (all definitions and Theorems are from [24]).

To get a better understanding of what infinitesimally bearing rigid formations are we illustrated some examples in Figure 1.4. All formations that are infinitesimally bearing rigid, with an arbitrary number of agents and for two as well as higher dimensions, can be stabilized by controller (1.3) from almost every initial conditions. In Figure 1.4a it can be seen that trivial motions such as scaling and translations maintain the bearing formation. The two agents can be close together or far away from each other but still



(a) Line. **(b)** Triangle. **(c)** Square. **(d)** Cube.

Figure 1.4: Infinitesimally bearing rigid formations.

preserve a static relative angle to each other. Also the position where the line is shaped does not change the bearing angle. The same properties can be obtained by the formations shown in Figure 1.4b-1.4d. In the next Section we restrict the visual sensor to track only a limited area.

1.3 Limited field-of-view Sensing Model

This work focuses on a bearing-only controller that is capable to stabilize formations in a predefined shape. Since the objective is of great interest in applications, we formulate a realistic sensing framework. Therefor we take a limited FOV into account, often 360°-cameras are expensive, heavy and require more computational hardware power to process images. To bring the controller proposed in [24] closer to be applicable we augment the state space of every agent with a facing direction of the camera. The statespace of agent i then follows as $\chi_i(t) = \begin{bmatrix} p_i(t)^T & \psi_i(t) \end{bmatrix}^T$, where $\psi_i(t) \in \mathcal{S}^1$ refers to the facing direction of the camera.[1] Note that this extension is designed for the plane ($d = 2$) and might have to be defined as vector when considering higher dimension $d > 2$. The dynamics of the agents defined in Section 1.1.1 are then also extended with one more integrator for the facing direction,

$$\dot{\chi}_i(t) = \begin{bmatrix} \dot{p}_i(t) \\ \dot{\psi}_i(t) \end{bmatrix} = \begin{bmatrix} u_i(t) \\ \omega_i(t) \end{bmatrix}, \tag{1.6}$$

where $\omega_i(t) \in \mathbb{R}$ is a controller for the facing direction, directly affecting the rotation rate.

[1] Here, $\mathcal{S}^1$ denotes the 1-dimensional manifold on the unit circle.

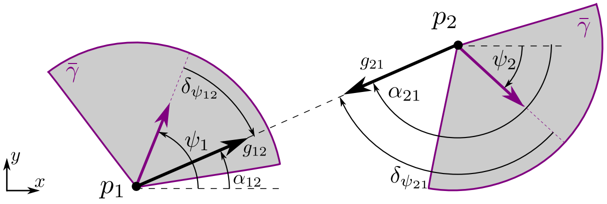


Figure 1.5: Two agent configuration with FOV constrained bearing sensing, note that $|\delta_{\psi_{21}}| > \bar{\gamma}/2$.

In order to deal with limited sensor information we introduce $\bar{\gamma}$ as the maximum area that is covered by the visual sensor, or the maximum limited FOV. If a neighboring agent is inside the limited FOV then the sensor is able to extract a relative angle with respect to the facing direction. We denote by $\delta_{\psi_{ij}}$ the angle between the facing direction ψ_i of agent i and the bearing g_{ij}. That is

$$|\delta_{\psi_{ij}}| = \cos^{-1}\left(\begin{bmatrix}\cos(\psi_i) & \sin(\psi_i)\end{bmatrix} g_{ij}\right).$$

We are now able to state $|\delta_{\psi_{ij}}| < \bar{\gamma}/2$ as a formal condition for being inside the limited FOV, if this condition holds then agent i is able to extract bearing information of its neighboring agent j, described by the bearing g_{ij}. The limited FOV is illustrated in Figure 1.5.

In the next Chapter we propose a controller for ω_i that only requires bearing measurements and state the closed loop dynamics for the two agent case. We then characterize the different equilibrium points and show their stability. Note that the theory of infinitesimally bearing rigidity and the bearing-only formation controller without limited FOV constraints proposed in this introduction can directly be extended to an arbitrary number of agents and for higher dimensions.

2 Bearing-only Formation Control with Limited FOV: Two Agent Case

In this chapter we propose a controller for the facing direction ω_i to guarantee that a neighbor stays inside the FOV. Controlling ψ_i is crucial to reach the desired formation when there is limited visual sensing. If the facing direction is static such that $\psi_i(0) = \psi_i(t)$ for all $t > 0$, then the trajectories of the bearing-only controller (1.3) can evolve such that a neighbor leaves the FOV. Once a neighbor leaves the FOV an agent can not obtain the bearing to that neighbor and therefor does not move anymore. In Figure 2.1 a simulation is shown where the facing direction is not controlled (e.g. $\omega_i(t) = 0$ for $i = 1{,}2$ and $t \geq 0$) for the two agent case. It can be seen that the facing direction does not change during the movement of the agents. The desired bearing $g_{12}^* = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ is not reached by only controlling the positions of the agents by a bearing-only controller (1.3). In this simulations example both agents can sense each other initially but the trajectories evolve such that first agent two looses track of agent one and then also agent one looses its neighbor. Once both agents are unable to track each other, they do not move, since they do not have a sensing input. The final position where no sensing is available does not refer to the desired bearing, which is illustrated by a green dashed line $g_{12}(t)$. In the simulation shown in Figure 2.1 the bearing only controller (1.3) is not able to reach the desired formation. To tackle that problem we propose a control strategy for the facing direction, that guarantees a connection link.

Compared to the bearing-only controller without limited visual sensing the set of neighbors is time dependent. In the simulation example (Figure 2.1) initially $|\mathcal{N}_i(0)| = 1$ for $i = 1{,}2$ but during the movement both sets become empty, since the neighbors leave the FOV. Even for the two agent case this results in non trivial analysis, since the dynamics of the agent change with the number of neighbors it can sense.

In this chapter we first state the notation for the two agent case. Then

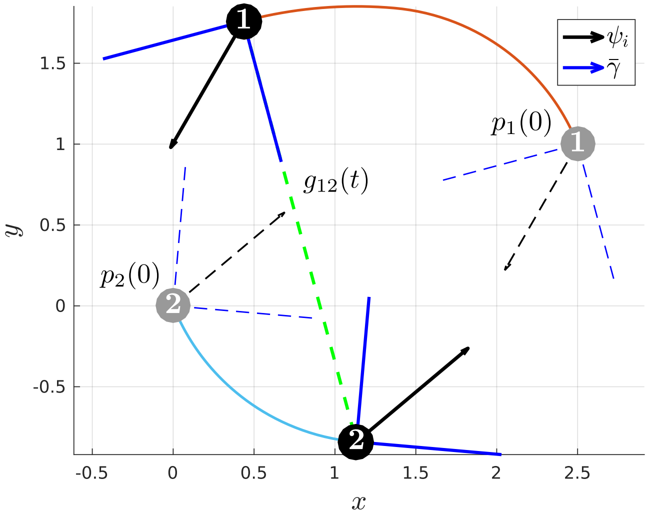


Figure 2.1: Simulation of bearing-only controller (1.3) for the two agent case, the desired bearing is not reached, since the facing direction is not controlled $g_{12}^* \neq g_{12}(t)$.

we propose a controller for the facing of the camera and finally provide a complete stability analysis of the closed loop dynamics. The analysis also includes a characterization of the different initial conditions. In this preliminary work we focus on the two agent case, however we are able to provide simulation results for three and four agents.

Note that in the two agent case rigidity theory described in Section 1.2 is not required, since there is only one edge connecting two agents.

2.1 System and Sensing Model for the two agent case

In this section, we introduce the multi-agent system model with limited FOV bearing sensing for the two agent case. Each agent is described by its position vector and its facing direction ($\chi_i(t)$) defined in Section 1.3. From here on, we will neglect the time dependency of our signals where it is obvious but include it when necessary.

Each agent is equipped with a sensor that is able to measure the relative bearing (in a common reference frame) to the other agent. We assume the sensor is mounted such that it points in the direction of the agent heading, ψ_i. The unit bearing vector between agent 1 and 2 defined in (1.1) follows for the two agent case as

$$g_{12} := \frac{p_2 - p_1}{\|p_2 - p_1\|} = \frac{z_{12}}{d_{12}}. \tag{2.1}$$

The initial distance is denoted as $\bar{d}_{12} = d_{12}(0)$.

As the bearing vector is expressed in a global reference frame, it follows that $g_{12} = -g_{21}$.

The bearing angle of g_{12} with respect to the (world frame) x-axis is defined as

$$\alpha_{12} = \tan^{-1}\left(\frac{[g_{12}]_y}{[g_{12}]_x}\right) = \cos^{-1}([g_{12}]_x) = \sin^{-1}([g_{12}]_y). \tag{2.2}$$

Note that $\alpha_{21} = \alpha_{12} \pm \pi$ defines the angle of g_{21} at the position of agent two. The facing error follows as $\delta_{\psi_i} = \alpha_{ij} - \psi_i$, $i,j = 1,2$ (here we use a single subscript for the bearing error since in the two agent case there is only one bearing that can be obtained). The sign of δ_{ψ_i} indicates if an agent is on the right- or on the left-side with respect to the facing direction. These notations are illustrated in Figure 1.5.

We introduce an indicator function for each agent that indicates if a neighbor can be sensed or not,

$$w_i(t) = \begin{cases} 1 \text{ if } |\delta_{\psi_i}(t)| < \frac{\bar{\gamma}}{2} \\ 0 \text{ else.} \end{cases} \tag{2.3}$$

Here we use $\bar{\gamma}$ introduced in Section 1.3 as the maximum limited FOV. Since the facing direction indicates the middle of a cone shown in Figure 1.5, each agent in the range $[\psi_i - \bar{\gamma}/2, \psi_i + \bar{\gamma}/2]$ can be sensed. The indicator weights for both agents can be compactly written in matrix form as

$$W(t) := \begin{bmatrix} w_1(t) & 0 \\ 0 & w_2(t) \end{bmatrix}. \tag{2.4}$$

In the next section, we will include the weight matrix in the position controller (1.3) for the two agent case, and propose an extension to this law to cope with the limited FOV sensor constraints.

2.2 Positions and Facing Controller

The position controller from [24] described in Section 1.1.2 controls the linear velocity of each agent and results in the states p_i. The desired formation shape is specified by a constant unit bearing vector g_{12}^* (the angle of the desired bearing follows as $\alpha_{ij}^* = \tan^{-1}\left([g_{ij}^*]_y / [g_{ij}^*]_x\right)$). For $n = 2$, controller (1.3) follows as,

$$\dot{p}_i = -P_{g_{ij}} g_{ij}^*, \qquad i = 1,2,\ j \in \mathcal{N}_i. \tag{2.5}$$

When the FOV constraints of the sensor are considered, the dynamics described in (2.5) become

$$\dot{p} = u = \begin{bmatrix} -w_1(t) P_{g_{12}} g_{12}^* \\ w_2(t) P_{g_{12}} g_{12}^* \end{bmatrix} = (W(t) H \otimes I_2) P_{g_{12}} g_{12}^*, \tag{2.6}$$

where $H = \begin{bmatrix} -1 & 1 \end{bmatrix}^T$ is the incidence matrix and vertex 1 and 2 corresponds to the tail and head of the connecting edge. The indicator matrix is noted as $W(t)$ (e.g. (2.4)). Note that $\dot{p}_i = 0$ when $g_{12} = \pm g_{12}^*$ or when $w_i = 0$. It is thus clear that if the agents do not also control their facing direction, it may be they will not converge to the desired formation (compare simulation shown in Figure 2.1).

In the two agent case, a natural approach for controlling the facing direction is to align the facing direction with the bearing measurement, ensuring it is inside the FOV of the sensor. That is, we would like to design a control for the facing direction that drives δ_{ψ_i} to zero.

With this setup, we state the FOV constrained bearing formation control problem below.

Problem 2.2.1. *Given a desired bearing g_{12}^*, an initial formation $\chi(0)$, and the limited FOV $\bar{\gamma}$, find the control inputs u_i and ω_i, such that $\delta_{\psi_i} \to 0$ and $g_{12} \to g_{12}^*$ as $t \to \infty$.*

To solve this problem, we propose the following controller for the facing direction to augment the bearing-only formation control in (2.6),

$$\omega_i = w_i(\underbrace{\kappa\left([g_{12}]_x[g_{12}^*]_y - [g_{12}]_y[g_{12}^*]_x\right)}_{v(\kappa, g_{12})} + \delta_{\psi_i}), \tag{2.7}$$

for some scalar $\kappa > 0$ and w_i the FOV indicator function (2.3). This controller is decoupled from the position controller (2.6) and uses only local information that can be obtained by a visual sensor.

We begin our analysis by first providing a bound for the term $v(\kappa, g_{12})$, which will be useful to prove our main result.

Proposition 2.2.2. *The control term $v(\kappa, g_{12})$ is bounded by $|\kappa|$, that is, $v(\kappa, g_{12}) \leq \kappa$ for all time t.*

Proof. Since g_{12} and g_{12}^* are unit vectors $[g_{12}]_x^2 + [g_{12}]_y^2 = 1$ holds. Therefore $\left([g_{12}]_x[g_{12}^*]_y - [g_{12}]_y[g_{12}^*]_x\right) \leq 1$, and we conclude $v(\kappa, g_{12}) \leq \kappa$. □

The complete dynamics of the closed loop system can be written as

$$\dot{\chi} = \begin{bmatrix} u \\ \omega \end{bmatrix} = \begin{bmatrix} (WH \otimes I_2)P_{g_{12}}g_{12}^* \\ W\left(v(\kappa, g_{12})\mathbb{1} - \delta_{\psi}\right) \end{bmatrix}, \tag{2.8}$$

where $\delta_{\psi} = \begin{bmatrix}\delta_{\psi_1} & \delta_{\psi_2}\end{bmatrix}^T$ and $\mathbb{1} = \begin{bmatrix}1 & 1\end{bmatrix}^T$.

In the next section we describe the different equilibrium configurations of the system and prove their stability.

2.3 Stability Analsyis: Two Agent Case

The proposed control strategy presented in (2.8) naturally will depend on the initial conditions of the agents. Indeed, if the facing direction of both agents are such that neither are in the sensor's FOV, then both agents will remain stationary, and the objective can not be met. In this direction, we identify four different possible initial conditions, characterized by the value of the indicator function $w_i(0)$, that lead to different behaviors of the system. For each set of initial conditions, we provide a complete stability and convergence analysis of the closed-loop system. Qualitatively, the trajectories of the agents can produce four behaviors of the indicator function: i) both agents never sense eachother, ii)both agents sense eachother, iii) only one agent senses the other, iv) an agent enters or leaves the FOV during the trajectory.

The cases proposed are analyzed in the next Sections.

2.3.1 No Sensing: $w_1(0) = w_2(0) = 0$

Formally, the first case is stated as $|\delta_{\psi_i}(0)| > \bar{\gamma}/2$, $i = 1{,}2$ which means that the visual sensor cannot extract any information of its neighbor, and both weights are zero. Thus, we have that $u = 0$ and $\omega = 0$. It follows that the equilibrium point is the initial condition, and the agents simply do not move. This clearly degenerate case motivates the inclusion of the following assumption on our dynamics, which ensures that at least one agent is inside the FOV of the other.

Assumption 2.3.1. *The initial condition $\chi(0)$ is such that $w_1(0) + w_2(0) \geq 1$.*

2.3.2 Complete Sensing: $w_1(0) = w_2(0) = 1$

In the second case both agents can sense each other and $|\delta_{\psi_i}(0)| < \bar{\gamma}/2$, $i = 1{,}2$. Controller (2.6) becomes the bearing-only control law introduced in [24] and described in Section 1.1.2. This has been shown to be stable when $\bar{\gamma} = 2\pi$. In the limited FOV setup, however, the facing direction has to change to ensure the agents remain inside the FOV. The first result shows that the controller (2.7) guarantees that the facing error, δ_{ψ_i} is bounded by $\bar{d}_{12}^{-1}$.

Proposition 2.3.2. *If the indicator function $w_i(0) = 1$, $i = 1,2$, controller (2.7) guarantees that the facing error stays bounded, such that $|\delta_{\psi_i}(t)| \leq 1/\bar{d}_{12}$ holds for all time t.*

Proof. First, observe that $\cos(\alpha_{12}) = \begin{bmatrix} 1 & 0 \end{bmatrix} g_{12}$. Therefore,

$$\begin{aligned} \frac{d}{dt}\cos(\alpha_{12}) = -\dot{\alpha}_{12}\sin(\alpha_{12}) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \dot{g}_{12} \\ &= \begin{bmatrix} 1 & 0 \end{bmatrix} \frac{P_{g_{12}}}{\bar{d}_{12}} (H \otimes I_2)^T \dot{p} \\ &= \frac{1}{\bar{d}_{12}} \begin{bmatrix} 1 & 0 \end{bmatrix} ((w_1 + w_2) I_2) P_{g_{12}} g_{12}^* \\ &= \frac{(w_1 + w_2)[g_{12}]_y}{\bar{d}_{12}} \left([g_{12}]_x [g_{12}^*]_y - [g_{12}]_y [g_{12}^*]_x \right). \end{aligned}$$

Since $\sin(\alpha_{12}) = [g_{12}]_y$ we obtain

$$\dot{\alpha}_{12} = \dot{\alpha}_{21} = \frac{w_1 + w_2}{\bar{d}_{12}} \left([g_{12}]_x [g_{12}^*]_y - [g_{12}]_y [g_{12}^*]_x \right). \tag{2.9}$$

Note that $\alpha_{12} = \alpha_{21} \pm \pi$, and therefore $\dot{\alpha}_{12} = \dot{\alpha}_{21}$.[1] In the case $w_1(0) = w_2(0) = 1$, the dynamics of $\dot{\delta}_{\psi_i}$, $i = 1,2$ become

$$\begin{aligned} \dot{\delta}_{\psi_i} &= \dot{\alpha}_{ij} - \dot{\psi}_i \\ &= \frac{1}{\bar{d}_{12}} \left([g_{12}]_x [g_{12}^*]_y - [g_{12}]_y [g_{12}^*]_x \right) - \delta_{\psi_i} \\ &= v(1/\bar{d}_{12}, g_{12}) - \delta_{\psi_i}. \end{aligned}$$

Here we choose $\kappa = 1/\bar{d}_{12}$, such that we can subtract the terms. By using Proposition 2.2.2 we conclude that $v(1/\bar{d}_{12}, g_{12}) \in [-1/\bar{d}_{12},\ 1/\bar{d}_{12}]$. In a next step we look at the solution of the differential equation

$$\delta_{\psi_i}(t) = e^{-t}\delta_{\psi_i}(t_0) + \int_{t_0}^{t} e^{-(t-\tau)} v(1/\bar{d}_{12}, g_{12}(\tau)) d\tau$$

[1] For more details on the dynamics on $\dot{g}_{12}$, we refer to [18].

which satisfies the bound

$$\begin{aligned} |\delta_{\psi_i}(t)| &\leq e^{-t}|\delta_{\psi_i}(t_0)| + \frac{1}{\bar{d}_{12}} \int_0^t e^{-(t-\tau)} d\tau \\ &\leq e^{-t} \left[|\delta_{\psi_1}(t_0)| - \frac{1}{\bar{d}_{12}} \right] + \frac{1}{\bar{d}_{12}}. \end{aligned}$$

It follows that the bearing error does not exceed $\pm 1/\bar{d}_{12}$. □

Proposition 2.3.2 can be used to establish a relationship between the initial distance of the agents and the required FOV of the sensor to ensure that if $w_1(0) = w_2(0) = 1$ that the agents will not leave the FOV of their neighbors under the trajectories of the system.

Corollary 2.3.3. *If $\bar{\gamma}/2 > 1/\bar{d}_{12}$ and $w_1(0) = w_2(0) = 1$, then under control law (2.8), $w_1(t) = w_2(t) = 1$ for all $t \geq 0$.*

Proof. In Proposition 2.3.2 we showed that $|\delta_{\psi_i}| \leq 1/\bar{d}_{12}$ holds under control law (2.7). Therefor we conclude that if the maximum FOV satisfies the condition $\bar{\gamma}/2 > 1/\bar{d}_{12}$, the indicator function will be constantly active, if it is active initially. □

Note that if $\bar{d}_{12} \leq 1/\pi$ the sensor has to cover 360 degrees.

Remark 2.3.4. *The closer the agents are initially, the faster the sensor has to turn in order to keep track of their neighbor. This makes intuitive sense since trajectories of the bearing controller are circles defined by the initial positions (see [24]). If the agents are closer together the radius of that circle is smaller and the bearing measurement will change faster.*

Corollary 2.3.3 leads to another Assumption on the system that will guarantee agents will not leave the FOV of their neighbor once sensed.

Assumption 2.3.5. *The sensor FOV satisfies $\bar{\gamma}/2 > 1/\bar{d}_{12}$.*

We now examine the equilibrium point of the closed loop (2.8) under Assumption 2.3.5. The equilibrium point of controller (2.6) is given as $g_{12} = \pm g_{12}^*$, as shown by the following,

$$\begin{aligned} \dot{p}_i = 0 &= (I_2 H \otimes I_2) P_{g_{12}} g_{12}^* \\ &= (g_{12}^*)^T (I_2 H \otimes I_2)^T (I_2 H \otimes I_2) P_{g_{12}} g_{12}^* \\ &= (g_{12}^*)^T P_{g_{12}} g_{12}^*. \end{aligned}$$

Note that from the properties of $P_{g_{12}}$ it follows that $(g_{12}^*)^T P_{g_{12}} g_{12}^* = g_{12}^T P_{g_{12}^*} g_{12}$. Here we used *Lemma 8* from [24] and since the kernel of the projection matrix contains the bearing measurement, it follows that Null $\left(P_{g_{12}^*}\right) = \text{span}\left\{\pm g_{12}^*\right\}$ and therefore we can conclude that $\dot{p} = 0$ only holds if $g_{12} = \pm g_{12}^*$. Note that if $g_{12} = \pm g_{12}^*$ holds then the bearing dynamics become zero, which leads to $v(1/\bar{d}_{12}, g_{12}) = 0$. From Proposition 2.3.2 we then conclude that δ_{ψ_i}, $i = 1,2$ will exponentially converge to zero under control law (2.7).

In order to show the stability of the equilibrium point we introduce $\delta_g = g_{12} - g_{12}^*$ as the bearing error. Then we show that $\delta_g = -2g_{12}^*$ is unstable and $\delta_g = 0$ is stable, which refers to $g_{12} = g_{12}^*$ and $g_{12} = -g_{12}^*$ respectively. First, we note that $|\delta_{\psi_i}(0)| < \bar{\gamma}/2$, $i = 1,2$ holds whenever the facing direction belongs to the intervals below,

$$\mathcal{B}_1 := \left[\alpha_{12}(0) - \frac{\bar{\gamma}}{2}, \alpha_{12}(0) + \frac{\bar{\gamma}}{2}\right], \qquad \mathcal{B}_2 := \left[\alpha_{21}(0) - \frac{\bar{\gamma}}{2}, \alpha_{21}(0) + \frac{\bar{\gamma}}{2}\right].$$

Theorem 2.3.6. *Under Assumption 2.3.5 and initial conditions satisfying $\psi_1(0) \in \mathcal{B}_1$ and $\psi_2(0) \in \mathcal{B}_2$, then $\delta_g \to 0$ and $\delta_\psi \to 0$ for almost all initial configurations $p_i(0) \in \mathbb{R}^2$, $i = 1,2$, except for the point corresponding to $g_{12}(0) = -g_{12}^*$.*

Proof. First we show that $\delta_g = -2g_{12}^*$ is an unstable equilibrium point for controller (2.6). Consider the dynamics of the bearing error, δ_g,

$$\begin{aligned} \dot{\delta}_g = f(\delta_g) &= \dot{g}_{12} \\ &= \frac{P_{g_{12}}}{\bar{d}_{12}} (H \otimes I_2)^T (I_2 H \otimes I_2) P_{g_{12}} g_{12}^* \\ &= \frac{2}{\bar{d}_{12}} P_{g_{12}} g_{12}^*. \end{aligned}$$

then we obtain the Jacobian

$$\begin{aligned} A = \frac{\partial f(\delta_g)}{\partial \delta_g} &= -\frac{2}{\bar{d}_{12}} \left(g_{12}^T g_{12}^* I_2 + g_{12}(g_{12}^*)^T\right) \frac{\partial g_{12}}{\partial \delta_g} \\ &= -\frac{2}{\bar{d}_{12}} \left(g_{12}^T g_{12}^* I_2 + g_{12}(g_{12}^*)^T\right). \end{aligned}$$

After inserting $\delta_g = -2g_{12}^*$ we get

$$\begin{aligned} A|_{\delta_g=-2g_{12}^*} &= \frac{4}{\bar{d}_{12}} \left((g_{12}^*)^T g_{12}^* I_2 + g_{12}^* (g_{12}^*)^T \right) \\ &= \frac{4}{\bar{d}_{12}} (I_2 + g_{12}^* (g_{12}^*)^T) \succeq 0 \end{aligned} \tag{2.10}$$

The last line of (2.10) is true since g_{12}^* is a unit vector with length one. The Jacobian is positive semi-definite and therefore we conclude that $g_{12} = -g_{12}^*$ corresponds to an unstable equilibrium point.

Now we will show that $g_{12}(0) = g_{12}^*$ is a stable equilibrium point which will be reached from every initial condition satisfying Assumption 2.3.5, except the unstable equilibrium point ($g_{12} = -g_{12}^*$). We define the Lyapunov function $V_1 := 1/2(\delta_g^T \delta_g + \delta_{\psi_1}^2 + \delta_{\psi_2}^2)$ which is positive semi-definite and zero only at $\delta_g = 0, \delta_{\psi_1} = \delta_{\psi_2} = 0$. Its time derivative follows as

$$\begin{aligned} \dot{V}_1 =& \delta_g^T \dot{\delta}_g + \delta_{\psi_1} \dot{\delta_{\psi_1}} + \delta_{\psi_2} \dot{\delta_{\psi_2}} \\ =& -\frac{2}{d_{12}} (g_{12}^*)^T P_{g_{12}} g_{12}^* \\ & + \delta_{\psi_1} \left(\frac{1}{d_{12}} ([g_{12}]_x [g_{12}^*]_y - [g_{12}]_y [g_{12}^*]_x) - \delta_{\psi_1} \right) \\ & + \delta_{\psi_2} \left(\frac{1}{d_{12}} ([g_{12}]_x [g_{12}^*]_y - [g_{12}]_y [g_{12}^*]_x) - \delta_{\psi_2} \right) \\ =& -\frac{2}{d_{12}} ([g_{12}]_x [g_{12}^*]_y - [g_{12}]_y [g_{12}^*]_x)^2 \\ & + \delta_{\psi_1} \left(\frac{1}{d_{12}} ([g_{12}]_x [g_{12}^*]_y - [g_{12}]_y [g_{12}^*]_x) - \delta_{\psi_1} \right) \\ & + \delta_{\psi_2} \left(\frac{1}{d_{12}} ([g_{12}]_x [g_{12}^*]_y - [g_{12}]_y [g_{12}^*]_x) - \delta_{\psi_2} \right) \\ \dot{V}_1 d_{12} =& -\delta_{\psi_1}^2 + \delta_{\psi_1} ([g_{12}]_x [g_{12}^*]_y - [g_{12}]_y [g_{12}^*]_x) \\ & - ([g_{12}]_x [g_{12}^*]_y - [g_{12}]_y [g_{12}^*]_x)^2 \\ & - \delta_{\psi_2}^2 + \delta_{\psi_2} ([g_{12}]_x [g_{12}^*]_y - [g_{12}]_y [g_{12}^*]_x) \\ & - ([g_{12}]_x [g_{12}^*]_y - [g_{12}]_y [g_{12}^*]_x)^2 \leq 0. \end{aligned}$$

The last line is true since it is an elliptic paraboloid that is smaller then zero everywhere but the origin, concluding the proof. □

We showed that if both neighbors see each other initially and Assumption 2.3.5 is fulfilled, then the desired bearing will be reached and the proposed facing controller (2.7) guarantees that the connection does not get lost during the movement of the agents.

2.3.3 Partial Sensing: $w_1(0) = 1, w_2(t) = 0, t \geq 0$

Without loss of generality, we will consider the case where agent one can sense its neighbor, but is outside the FOV of agent two (i.e., $|\delta_{\psi_1}(0)| < \bar{\gamma}/2$ and $|\delta_{\psi_2}(0)| > \bar{\gamma}/2$, see Figure 1.5). The equilibrium point of (2.8) now clearly depends on the facing direction of agent two. We now focus on the scenario where agent one can achieve the desired formation without ever entering the FOV of agent two. We will define initial conditions for $\psi_2(0)$ that ensures this behavior. Before defining the interval explicitly, we first analyze the movement of agent one assuming $|\delta_{\psi_2}(t)| > \bar{\gamma}/2$ holds (and also $w_2(t) = 0$) for all $t \geq 0$. The dynamics of agent two will be zero and agent one will move on a static circle around agent two.

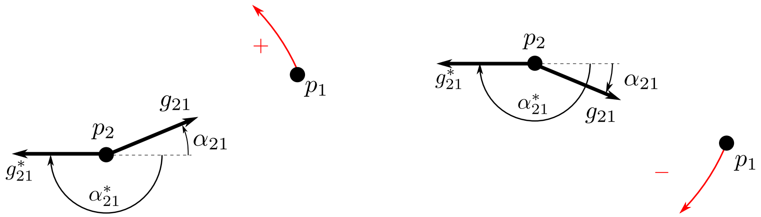
Lemma 2.3.7. *If $w_1(0) = 1$ and $w_2(t) = 0$ for all $t \geq 0$, agent one evolves on a circle with radius $r = \bar{d}_{12}$ and center $c = p_2(0)$.*

Proof. First, observe that the distance between the agents remains invariant along the trajectories of (2.6). Indeed, $d_{12}^2 = z_{12}^T z_{12}$ and

$$\begin{aligned} \frac{d}{dt} d_{12}^2 &= 2 z_{12}^T \dot{z}_{12} \\ &= -2 z_{12}^T P_{g_{12}} g_{12}^* = 0, \end{aligned}$$

since $z_{12} \in \text{Null}(P_{g_{12}})$. Furthermore, p_2 is stationary if $w_2 = 0$ and we conclude that it is the center of a circular movement with radius $r = d_{12}$, for more details we refer to [24](*Lemma 7*). Due to the invariance of the distance, $\bar{d}_{12}(t) = \bar{d}_{12}$ holds for all $t \geq 0$. □

In a next step, we define the direction in which agent one moves on the circle. This is important to formalize the interval for $\psi_2(0)$ that ensures $w_2(t) = 0$ holds. The rotation direction is indicated by the sign of the angle



(a) Agent one moves counter-clockwise $\alpha_{\delta_g} > 0$.

(b) Agent one moves clockwise $\alpha_{\delta_g} < 0$.

Figure 2.2: Direction of agent one if p_2 is stationary.

between g_{12} and g_{12}^* which we define as $\alpha_{\delta_g} = \alpha_{12}^* - \alpha_{12}$. The directions follow as

$$\alpha_{\delta_g} = \begin{cases} > 0 & p_1 \text{ moves clockwise} \\ < 0 & p_1 \text{ moves counter-clockwise.} \end{cases} \tag{2.11}$$

Note that $\alpha_{\delta_g} = 0$ only holds if $g_{12} = g_{12}^*$, which is the desired equilibrium point. An illustration is shown in Figure 2.2. Now we define two intervals for which $|\delta_{\psi_2}(t)| > \bar{\gamma}/2$ holds for all $t \geq 0$, depending on the direction that agent one moves,

$$\mathcal{M}_L := \left[\alpha_{12}(0) + \frac{\bar{\gamma}}{2}, \alpha_{12}^* - \frac{\bar{\gamma}}{2}\right], \qquad \mathcal{M}_U := \left[\alpha_{12}^* + \frac{\bar{\gamma}}{2}, \alpha_{12}(0) - \frac{\bar{\gamma}}{2}\right].$$

Now we will provide a stability proof that fist shows that the facing controller (2.7) guarantees that agent two will always be inside the FOV of agent one. In a second step we analyze the intervals $\mathcal{M}_L$ and $\mathcal{M}_U$ and show that if the facing direction of agent two is in one of the sets (depending on the sign of α_{δ_g}), agent two will never be able to track agent one. Finally we show that agent one is able to reach the objective without agent two moving.

Theorem 2.3.8. *Under Assumption 2.3.1 and initial conditions satisfying $\psi_2(0) \in \mathcal{M}_L$ if $\alpha_{\delta_g} < 0$, or $\psi_2(0) \in \mathcal{M}_U$ if $\alpha_{\delta_g} > 0$, then $\delta_g \to 0$, $\delta_{\psi_1} \to 0$, and $\delta_{\psi_2} = \alpha_{21}^* - \psi_2(0)$ for almost all initial configurations $p_i(0) \in \mathbb{R}^2$, $i = 1,2$, except for the point corresponding to $g_{12}(0) = -g_{12}^*$.*

Proof. First we will show that the sets $\mathcal{M}_L$ and $\mathcal{M}_U$ are invariant under the dynamics of the system (2.8). To do so we look at the upper and lower bound of δ_g. We then define the region in which it is possible for agent two to track agent one and then conclude that every facing direction of agent two outside those bounds will never be able to track agent one. We introduce a Lyapunov function $V_2 = 1/2\delta_g^T \delta_g$ which is positive definite everywhere but zero. Its time derivation follows as

$$\dot{V}_2 = \delta_g^T \dot{\delta}_g = -\left(g_{12}^*\right)^T \frac{P_{g_{12}}}{d_{12}} g_{12}^* \leq 0. \tag{2.12}$$

Now we see that $0 \leq |\delta_g(t)| < |\delta_g(0)|$ holds. From the error dynamics we also see the initial as well as the final bearing (e.g. $g_{12}(0)$ and g_{12}^*). Then we add $\pm\bar{\gamma}/2$ to the upper and lower bound of the bearings, since agent two is able to track its neighbors in the area $\psi_2 \pm \bar{\gamma}/2$. The sign depends on the moving direction of agent one and therefor on α_{δ_g}. Combining the bounds of the bearings and the limited FOV, we are able to state a set in which agent two can track agent one $|\delta_{\psi_2}(T)| < \frac{\bar{\gamma}}{2}$ for some time T, under the dynamics of (2.8). For $\alpha_{\delta_g} > 0$ we get the bounds of

$$\alpha_{21}(t) \pm \frac{\bar{\gamma}}{2} \in \left(\tan^{-1}\left(\frac{[g_{21}(0)]_y}{[g_{21}(0)]_x}\right) - \frac{\bar{\gamma}}{2}, \tan^{-1}\left(\frac{[g_{21}]_y^*}{[g_{21}]_x^*}\right) + \frac{\bar{\gamma}}{2}\right), \tag{2.13}$$

and if $\alpha_{\delta_g} < 0$ we get

$$\alpha_{21}(t) \pm \frac{\bar{\gamma}}{2} \in \left(\tan^{-1}\left(\frac{[g_{21}]_y^*}{[g_{21}]_x^*}\right) - \frac{\bar{\gamma}}{2}, \tan^{-1}\left(\frac{[g_{21}(0)]_y}{[g_{21}(0)]_x}\right) + \frac{\bar{\gamma}}{2}\right). \tag{2.14}$$

If ψ_2 is inside these bounds then it will sense $g_{12}(t)$ before the objective is met, and if not then $|\delta_{\psi_2}(t)| > \bar{\gamma}/2$ holds for all $t \geq 0$. By building the union of intervals $\mathcal{M}_U$ and (2.13) we get an interval covering all angles $[-\pi, \pi]$, the same is true for the union of $\mathcal{M}_L$ and (2.14). We conclude that if $\alpha_{\delta_g} < 0$, $\mathcal{M}_L$ is invariant, and if $\alpha_{\delta_g} > 0$, $\mathcal{M}_U$ is invariant, both under the dynamics of the closed loop (2.8).

An illustration of the angles can be seen in Figure 2.3. Since we assume that ψ_2 is in one of the invariant intervals (e.g $\mathcal{M}_L$, $\mathcal{M}_U$) depending of the sign of α_{δ_g} it follows that $w_2 \equiv 0$ and we get $\psi_2(t) = \psi_2(0)$. From Theorem 2.3.6 we conclude that $\delta_g = 0$ is a stable equilibrium point for controller (2.6) and $\delta_g = -2g_{12}^*$ is unstable. Now we are ready to show that controller (2.8) reaches the desired bearing and that the facing direction of agent one aligns

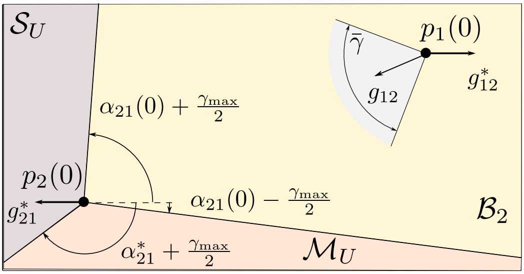

$\mathcal{S}_U$
$\bar{\gamma}$
$p_1(0)$
g_{12}
g^*_{12}
$\alpha_{21}(0) + \frac{\gamma_{\max}}{2}$
$p_2(0)$
g^*_{21}
$\alpha_{21}(0) - \frac{\gamma_{\max}}{2}$
$\mathcal{B}_2$
$\alpha^*_{21} + \frac{\gamma_{\max}}{2}$
$\mathcal{M}_U$

Figure 2.3: When $\psi_1 \in \mathcal{B}_1$ holds, there are three possible trajectories depending on the facing of agent two, marked by three different colors.

with it.

First we show that $w_1(t) = 1$ for all $t \geq 0$ by introducing $V_3 = 1/2\delta^2_{\psi_1}$, which is positive definite everywhere but zero, from

$$\dot{V}_3 = \delta_{\psi_1}\dot{\delta}_{\psi_1} = \delta_{\psi_1}\left(\dot{\alpha}_{12} - \dot{\psi}_1\right) = -\delta^2_{\psi_1}, \tag{2.15}$$

we conclude $|\delta_{\psi_1}(t)| < |\delta_{\psi_1}(0)|$ and $\delta_{\psi_1} \to 0$ as $t \to \infty$. We choose $\kappa = 1/\bar{d}_{12}$, such that we can eliminate $\dot{\alpha}_{12}$ dynamics. In (2.12) we showed that $\dot{V}_2 \leq 0$ and therefor conclude that $\delta_g \to 0$ as $t \to \infty$. □

Note that in this case Assumption 2.3.5 is not required since the dynamics of g_{12} change slower if only one agent moves.

2.3.4 Partial Sensing: $w_1(0) = 1$,$w_2(0) = 0$ and $w_2(t) = 1$ for $t > T$.

Now we look at the case where agent one moves inside the FOV of agent two, such that $|\delta_{\psi_2}(T)| \leq \bar{\gamma}/2$ for some time $T > t$, but $|\delta_{\psi_2}(0)| > \bar{\gamma}/2$ initially. First only agent one moves and once the trajectory enters the FOV of agent two the indicator function w_2 becomes active.

As before we first define an interval for the facing direction of agent two. We further define a switching point s which specifies the point where agent one enters the FOV of agent two. In a last step we show that the desired bearing will be reached after agent one hits the switching point. The intervals for the facing direction depends on α_{δ_g} defined in (2.11),

$$\mathcal{S}_L := \left[\alpha_{21}^* - \frac{\bar{\gamma}}{2}, \alpha_{21}(0) - \frac{\bar{\gamma}}{2}\right], \qquad \mathcal{S}_U := \left[\alpha_{21}(0) + \frac{\bar{\gamma}}{2}, \alpha_{21}^* + \frac{\bar{\gamma}}{2}\right].$$

Theorem 2.3.9. *Under Assumptions 2.3.1 and 2.3.5 and initial conditions satisfying $\psi_2(0) \in \mathcal{S}_L$ if $\alpha_{\delta_g} < 0$ or $\psi_2(0) \in \mathcal{S}_U$ if $\alpha_{\delta_g} > 0$, then $\delta_g \to 0$ and $\delta_{\psi_1} = \delta_{\psi_2} \to 0$ from almost all initial configurations $p_i(0) \in \mathbb{R}^2, i = 1,2$, except for the point corresponding to $g_{12}(0) = -g_{12}^*$.*

Proof. Agent two is only able to track agent one if it satisfies the intervals defined in Theorem 2.3.8, from there we remove the part where $|\delta_{\psi_2}(0)| \leq \bar{\gamma}/2$ holds and get $\mathcal{S}_L$ and $\mathcal{S}_U$. If the facing direction is in that region the trajectory of agent one will come into the FOV of agent two, before the desired bearing is reached.

In Theorem 2.3.6 we showed that $g_{12} = -g_{12}^*$ is a unstable equilibrium point of controller (2.6) and therefor undesired.

Now we can define a switching point and show that controller (2.6) will converge first to that point. Furthermore w_2 jumps from zero to one when agent one reaches the point, then both agents can see each other and converge to the desired formation, as shown in Theorem 2.3.6. We define the switching point as

$$s = \begin{bmatrix} \cos\left(\psi_2(0) - \operatorname{sgn}(\alpha_{\delta_g})\frac{\bar{\gamma}}{2}\right) \\ \sin\left(\psi_2(0) - \operatorname{sgn}(\alpha_{\delta_g})\frac{\bar{\gamma}}{2}\right) \end{bmatrix} - g_{12}^*, \tag{2.16}$$

in terms of bearing error. In Theorem 2.3.8 we showed that controller (2.7) guarantees that agent one keeps track of agent two. Since

$$0 \leq s < \delta_g(0) \tag{2.17}$$

holds, we have to show that δ_g decreases constantly. The Lyapunov function V_2 is positive-semidefinite, radially unbounded and zero only at the equilibrium point. The time derivative defined in (2.12) is negative semi-definite and zero only at the equilibrium point. Now we have shown that the bearing error constantly decreases and since (2.17) holds we guaranteed that agent one will hit the switching point s in finite time $t > T$. Once s is reached the stability of the equilibrium point follows as in Theorem 2.3.6. □

We showed now that $g_{12} = g_{12}^*$ will be stabilized if Assumption 2.3.1 and 2.3.5 hold and $g_{12}(0) \neq -g_{12}^*$. The introduced controller for the facing direction (2.7) therefor gives a criteria on the required FOV (Corollary 2.3.3) and uses only information locally available on each agent. In the next section we will provide some simulation results.

2.4 Simulations for $n = 2$ agents

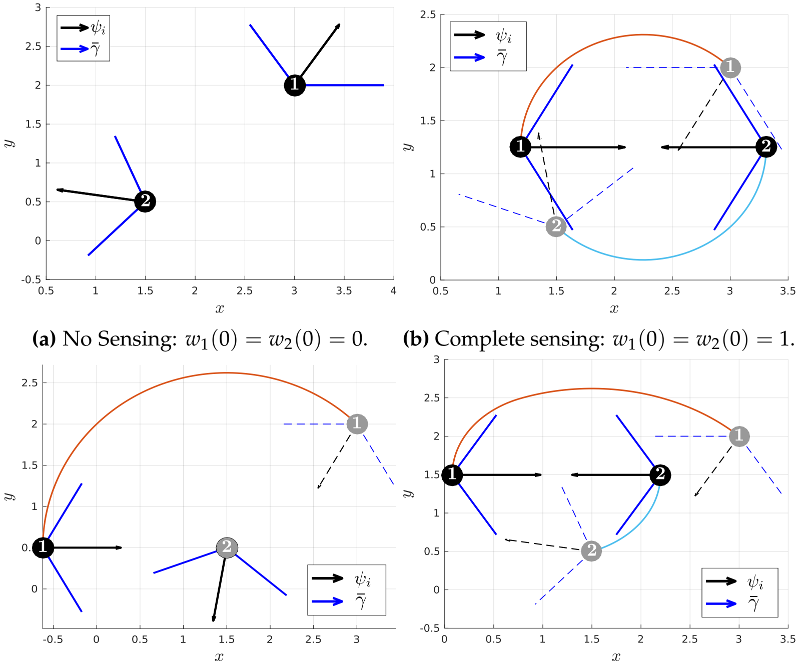
The control law (2.8) is applied for different initial facing directions but static positions in the two agent case. The results are shown in Figure 2.4. The simulation in Figure 2.4d shows the case where agent one moves into the FOV of agent two. The initial position is set to

$$p_1(0) = \begin{bmatrix} 3 & 2 \end{bmatrix}^T \qquad p_2(0) = \begin{bmatrix} 1.5 & 0.5 \end{bmatrix}^T,$$

$\psi(0)$ is different in all four cases, the desired bearing is $g_{12}^* = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$, which corresponds to a horizontal line (e.g. Figure 1.4a). The limited FOV is set to $\bar{\gamma} = \pi/2$. In Figure 2.4a both agents are outside the FOV of their neighbor and therefor are not able to extract the bearing information, its analysis is presented in Section 2.3.1. Figure 2.4b shows an initial conditions where $|\delta_{\psi_i}| < \pi/4$ for $i = 1,2$, which means that both agents are able to sense their neighbor, this refers to the undirected case. However this configuration also requires controller (2.7) to ensure that the neighbor stays inside the FOV, as shown in Section 2.3.2. The partial sensing cases are shown in Figure 2.4c-2.4d, which we have shown to be stable in Section 2.3.3 and 2.3.4 respectively.

2.5 Simulations for $n > 2$ agents

While the analysis in this work focused exclusively on the two agent case, we demonstrate in Figures 2.5a and 2.5b that the proposed strategy may also work for $n > 2$ agents. The main modification relates to the facing direction control. In the three agent case the desired facing direction was chosen to be the closest neighbor, while in the four agent case we choose it to be in the middle of all neighbors that can be sensed (details on the two different strategies are explained in Chapter 3). The stars on the trajectories in Figure 2.5b indicates when the number of sensed agents changes. The FOV is set to $\bar{\gamma} = 90°$ and $\bar{\gamma} = 100°$ for the three and four agent case respectively. In both



(a) No Sensing: $w_1(0) = w_2(0) = 0$.

(b) Complete sensing: $w_1(0) = w_2(0) = 1$.

(c) Partial sensing: $w_1(0) = 1$, $w_2(t) = 0, t \geq 0$.

(d) Partial sensing:$w_1(0) = 1$, $w_2(t) = 1, t \geq T$.

Figure 2.4: Simulation results for $n = 2$ agent cases.

examples, the agents successfully attain the desired formations, even while the number of sensed agents changes along the trajectories. The desired formation was set to

$$g^* = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}^T$$

and

$$g^* = \begin{bmatrix} 0 & 1 & 0 & -1 & \sqrt{2}/2 \\ 1 & 0 & -1 & 0 & \sqrt{2}/2 \end{bmatrix}^T,$$

for the three and four agent case respectively. We only used desired formations that are infinitesimally bearing rigid, where we stated the conditions in Section 1.2. The desired formation in the three agent case shown in Figure 2.5a refers to an equilateral triangle (e.g. Figure 1.4b) and the desired formation from Figure 2.5b to a square (e.g. Figure 1.4c).

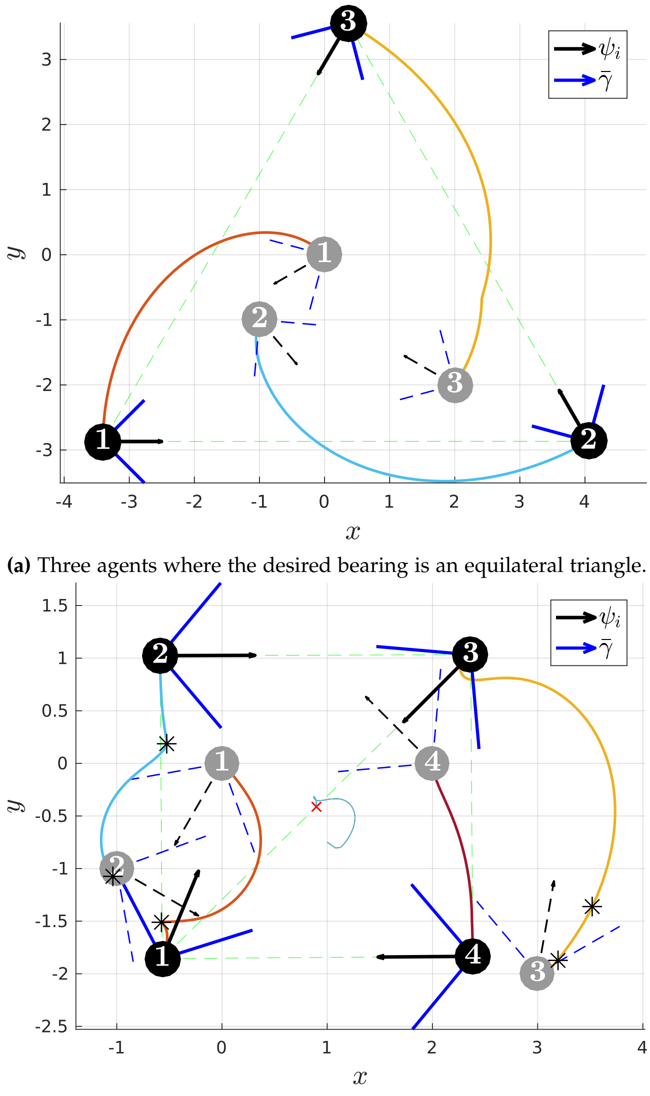
The incidence matrices follows as

$$H = \begin{bmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}^T$$

and

$$H = \begin{bmatrix} -1 & 0 & 0 & 1 & -1 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix}^T$$

for the three and four agent case. The simulations show that the desired bearing is reached in both cases, however this is only a simulation result which was not yet analytically proven. In the next chapter we apply the proposed controller (2.6) and (2.7) for ground robots.



(a) Three agents where the desired bearing is an equilateral triangle.

(b) Four agent case where the desired bearing is a rigid square.

Figure 2.5: Simulation results for $n = 3{,}4$ agent cases.

3 Experiments in the *CoNeCt* Lab

The lab of the *CoNeCt* group at the faculty of Aerospace Engineering is equipped with multiple *TurtleBotII* robots that can be used to validate our control strategy. In this chapter we will explain the setup that is available in the lab and how we used it to implement our controller. We first describe the equipment that we used, then give a short introduction into ROS, followed by a color detection algorithm realised in openCV. We then transform the controller from Chapter 2 to the dynamics of the *TurtleBotII* robots and pay special attention to the facing direction of the camera. The chapter will be concluded by experimental results that support our analysis from Section 2.3.

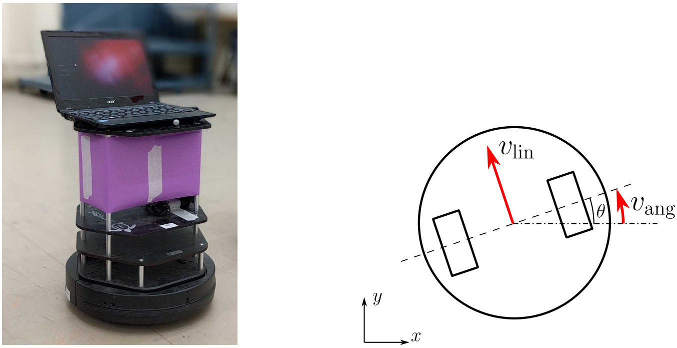
3.1 Setup

Before implementing the controller we will give an overview of the equipment, frameworks and libraries that are available in the lab.

3.1.1 TurtleBotII

TurtleBot is a mobile robotic platform that has range as well as visual sensors [19]. It is a ground robot that has a Kobuki base and a Microsoft Kinect Sensor [4]. The Kinect Sensor is able to create a depth image that uses visual as well as a range information. For our controller we modified the platform, such that we disconnected the Kinect Sensor and use a Logitech webcam [11] that has a wider limited FOV than the Kinect camera.

The dynamics of the Kobuki base can be modeled as unicycle, such that we can control the linear and angular velocity. For agent i the dynamics follow



(a) color coded *TurtleBotII* robot.

(b) States of the unicycle model, top view.

Figure 3.1: Equipment of the *CoNeCt* lab from the Faculty of Aerospace Engineering.

as

$$\begin{aligned}\dot{x}_i &= v_{i_{\text{lin}}} \cos(\theta_i) \\ \dot{y}_i &= v_{i_{\text{lin}}} \sin(\theta_i) \\ \dot{\theta}_i &= v_{i_{\text{ang}}}.\end{aligned} \tag{3.1}$$

A picture of a *TurtleBotII* is shown in Figure 3.1a. The states introduced in (3.1) are shown in Figure 3.1b. The robots are color-coded, such that multiple agents can be distinguished. We will use the color to obtain a relative angle with respect to the facing direction of the camera.

The TurtleBotII robot is controlled by an Acer TravelMate Netbook running a Linux distribution. The code is written is C++ and the Kobuki base is connected via ROS (Robot Operating System). In our setup we create a ROS message that takes the linear- and angular velocity is created and then sent to the motors.

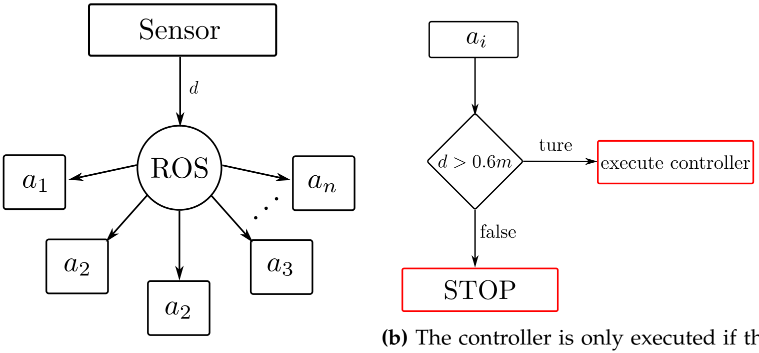
3.1.2 ROS, OptiTrack and Local Sensing

In our experiments ROS is mainly used to send control signals to the motors on the robots [12]. In general it is an extensive framework to control, track and manage robots, therefor a centralized master is collecting all data and distributes it over a set of nodes. Since our approach is decentralized such a master is not necessary. We use the ROS network to start the agents at the same time. Therefor one agent keeps the Kinect sensor to measure range information. This information is distributed over the ROS network to all other agents see Figure 3.2a. The decentralized controller running on each agent individually is executed only if the range of that specific agent is above a predefined threshold, which is illustrated in Figure 3.2b. So we run the program on every agent but cover the range sensor that connects all agents. Once the sensor is uncovered it exceeds the threshold and the agents start to move. That way we are able to have a nearly synchronized starting time. Another advantage of this realization is that we can abort an experiment if agents are not moving as expected, by simply covering the sensor. The flowchart in Figure 3.2 shows how the information of one common sensor is propagated through the network, note that also another common information could be used to generate a synchronized starting point. However this realization was straight forward since the sensors and the network already existed.

Another information that is available on the Kobuki base is the relative position of each agent with respect to its starting point. This information is obtained by counting cycles of the wheels. However it is not accurate since it has a high drift and therefor will not be used. The lab is also equipped with OptiTrack cameras that can obtain positions of reflecting markers [15]. We used three markers on each agent to get its position as well as its orientation. The position is relative to a fixed coordinate system in the middle of the lab. An illustration of the data that is obtained by OptiTrack is shown in Figure 3.3, the subscript of the robots refers to the specific name of each agent (e.g. Simeon, Gad and Reuben).

Our controller is based on local, visual sensing information obtained by a camera. From that camera image we extract relative angle information to the neighbors that are inside the limited FOV (e.g. δ_{ψ_i}, $i = 1{,}2$ in Figure 1.5). This information is stored in a logfile on each agent individually, we also track the input values and the facing direction of the camera.

This setup allows us to store all relative information to analyze the experiments afterwards. It should be mentioned that even though ROS connects all agents the controller run independent, we just use one sensor to synchronize



(a) We use one common sensor in order to synchronize the robots, this information is propagated through a ROS network.

(b) The controller is only executed if the sensor value exceeds a certain threshold, this is used to start and stop the experiments at the same time.

Figure 3.2: Flowchart of the connecting sensor that allows a synchronized starting time.

the starting point of all robots. In the next Section we will explain how to get relative angle information from a camera frame by using the OpenCV library.

3.1.3 OpenCV

Open Source Computer Vision Library is used for real-time computer vision [1]. In our setup we use it to detect the color-coded agents in a camera frame and then calculate a relative angle with respect to the middle of the camera. In a first step we have to define a color range of the color-coded robots. The colors are represented in HSV-model (hue, saturation, value), which is an alternative representation of the well known RGB color model. HSV is supposed to be more robust to changing light conditions, compared to the classic RGB representation. With a built in function we convert the camera frame to HSV.

```
//convert frame from BGR to HSV colorspace
cvtColor(*CAMERA_FRAME*,*OUTPUT_FRAME*,COLOR_BGR2HSV);
```

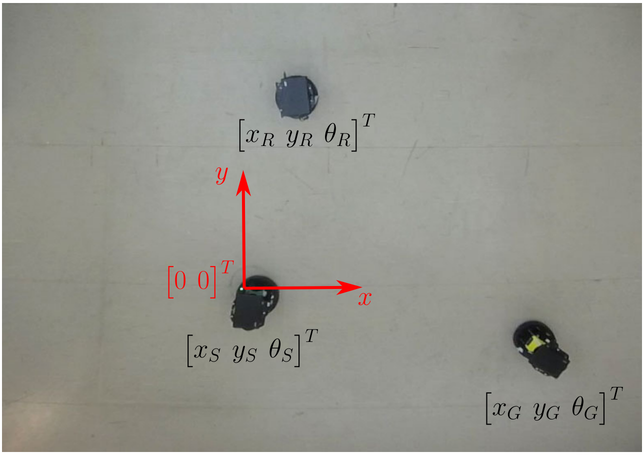


Figure 3.3: Top view of the *CoNeCt* Lab with three *TurtleBotII* robots. OptiTrack senses the marker on top of each robot and Simulink is processing the data.

Now we are able to obtain the neighboring agents by checking if the colors range between `*LOW_HSV_VALUES*` and `*HIGH_HSV_VALUES*` , are inside the HSV image by calling the `inRange()` function.

```
inRange(*INPUT_FRAME*,*LOW_HSV_VALUES*,*HIGH_HSV_VALUES*,*
    OUTPUT_FRAME*);
```

The `*OUTPUT_FRAME*` is then used to calculate moments of the colors that are inside the FOV. With the moments we are able to track the average position of a specified color,

$$\bar{x}_i = \frac{m_{10_i}}{m_{00_i}}$$
$$\bar{y}_i = \frac{m_{01_i}}{m_{00_i}}.$$

The values $\bar{x}_i$ and $\bar{y}_i$ refer to a mass center of the color value, the spatial moments are obtained by `moments(*INPUT_FRAME*)` function. Since the image is rasterized and stored in rows and columns we get the relative angular information by

$$\delta_{\psi_{ij}} = \frac{\bar{x}_j}{|\text{columns}|},$$

where $|\text{columns}|$ refers to the number of columns of a frame. The relative angle $\delta_{\psi_{ij}} \in [-\bar{\gamma}/2, \bar{\gamma}/2]$ represents the angle between agent i and j (e.g. $(i,j) \in \mathcal{E}$) with respect to the facing ψ_i.

This angular information represents bearing information g_{ij}. Since we know the heading of the camera (ψ_i) we can obtain $\alpha_{ij} = \psi_i + \delta_{\psi_{ij}}$ as the bearing angle of g_{ij} with respect to a global x-axis. The described values are shown in Figure 3.4

3.2 Bearing-only formation control with limited FOV: Unicycle Model

In contrast to the controller described in Chapter 2.2 the unicycle model is controlled by the linear and angular velocity shown in Figure 3.1b. In this Section we will transform the bearing-only controller (2.6), such that it suits the dynamics of the unicycle. We follow the idea proposed in [22](Theorem

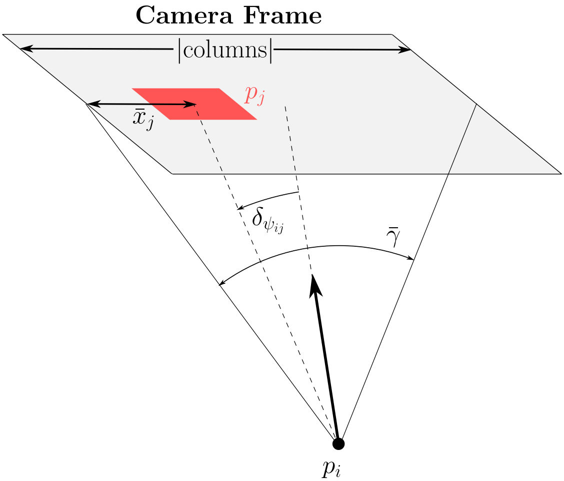


Figure 3.4: Camera frame that is taken from a visual sensor on agent i, the red square indicates the color of neighbor j within the camera frame.

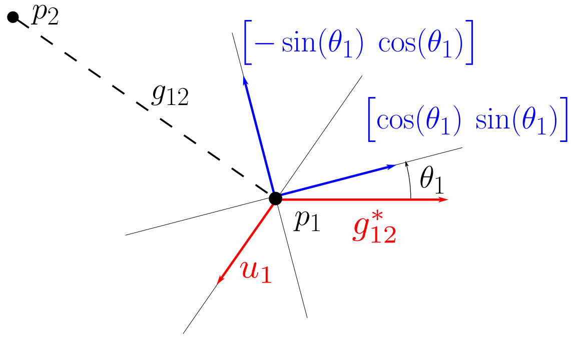

p_2
g_{12}
$\left[-\sin(\theta_1)\ \cos(\theta_1)\right]$
$\left[\cos(\theta_1)\ \sin(\theta_1)\right]$
θ_1
p_1
g^*_{12}
u_1

Figure 3.5: The linear and angular velocity of the unicycle described in (3.2) is the projection of u_1 onto $[\cos(\theta_1)\ \sin(\theta_1)]$ and $[-\sin(\theta_1)\ \cos(\theta_1)]$.

2). Therefor we divide the bearing controller (2.6) into a linear and an angular part

$$\begin{aligned} v_{i_{\text{lin}}} &= \begin{bmatrix}\cos(\theta_i) & \sin(\theta_i)\end{bmatrix} u_i \\ v_{i_{\text{ang}}} &= \begin{bmatrix}-\sin(\theta_i) & \cos(\theta_i)\end{bmatrix} u_i. \end{aligned} \tag{3.2}$$

A geometric interpretation is shown in Figure 3.5, the input u_i is projected on the heading of agent i and its orthogonal vector. In the example setup of Figure 3.5 $v_{1_{\text{lin}}}$ and $v_{1_{\text{ang}}}$ are smaller than zero. This refers to a linear as well as angular movement in negative direction. The closed loop dynamics for arbitrary number of agents and the unicycle model follow as

$$\begin{aligned} \dot{x}_i &= -\begin{bmatrix}\cos(\theta_i) & \sin(\theta_i)\end{bmatrix} \sum_{j\in\mathcal{N}_i} P_{g_{ij}} g^*_{ij} \cos(\theta_i) \\ \dot{y}_i &= -\begin{bmatrix}\cos(\theta_i) & \sin(\theta_i)\end{bmatrix} \sum_{j\in\mathcal{N}_i} P_{g_{ij}} g^*_{ij} \sin(\theta_i) \\ \dot{\theta}_i &= -\begin{bmatrix}-\sin(\theta_i) & \cos(\theta_i)\end{bmatrix} \sum_{j\in\mathcal{N}_i} P_{g_{ij}} g^*_{ij}. \end{aligned} \tag{3.3}$$

Here we assume that our sensor is able to sense $\bar{\gamma} = 2\pi$.

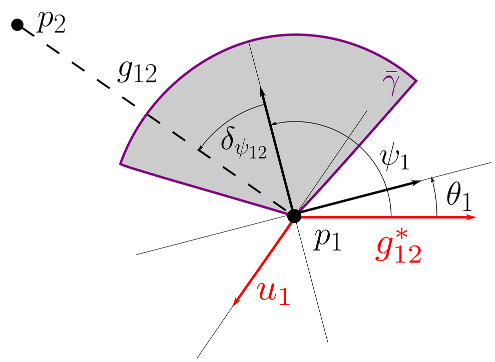


Figure 3.6: The camera does not align with the moving direction of the unicycle but is turned around $+\pi/2$.

3.2.1 Turning the Camera

We now have to decide where to place the camera in order to run the experiments. A natural approach would be to align the facing direction with the moving direction $\psi_i = \theta_i$. However we know that the agents evolve on a circle [24](Lemma 7) and from the projection shown in Figure 3.5 we see that another idea is to place the camera orthogonal to its natural heading (e.g. rotated $+\pi/2$).

In order to run the experiments on unicycle robots we mounted the camera such that it is turned around $+\pi/2$ to the moving direction. It can be seen as a static offset between ψ_i and θ_i. If we take the initial conditions shown in Figure 3.5 then the camera aligns with $\begin{bmatrix}-\sin(\theta_i) & \cos(\theta_i)\end{bmatrix}$, which is illustrated in Figure 3.6.

Turning the camera is motivated by the trajectories of the integrator dynamics for the two agent case.

In a first step we take a closer look at the two agent case to validate the heading of the camera. For the two agent case and the limited FOV sensing

(3.3) changes to

$$\begin{aligned}
\dot{x}_1 &= -w_1 \begin{bmatrix}\cos(\theta_1) & \sin(\theta_1)\end{bmatrix} P_{g_{12}} g_{12}^* \cos(\theta_1) \\
\dot{y}_1 &= -w_1 \begin{bmatrix}\cos(\theta_1) & \sin(\theta_1)\end{bmatrix} P_{g_{12}} g_{12}^* \sin(\theta_1) \\
\dot{\theta}_1 &= -w_1 \begin{bmatrix}-\sin(\theta_1) & \cos(\theta_1)\end{bmatrix} P_{g_{12}} g_{12}^* \\
\dot{x}_2 &= w_2 \begin{bmatrix}\cos(\theta_2) & \sin(\theta_2)\end{bmatrix} P_{g_{12}} g_{12}^* \cos(\theta_2) \\
\dot{y}_2 &= w_2 \begin{bmatrix}\cos(\theta_2) & \sin(\theta_2)\end{bmatrix} P_{g_{12}} g_{12}^* \sin(\theta_2) \\
\dot{\theta}_2 &= w_2 \begin{bmatrix}-\sin(\theta_2) & \cos(\theta_2)\end{bmatrix} P_{g_{12}} g_{12}^*.
\end{aligned}$$

where we extended the indicator function (2.3) such that it is capable to deal with more than one neighbor,

$$w_i(t) = \begin{cases} 1 \text{ if } |\delta_{\psi_{ij}}(t)| < \frac{\bar{\gamma}}{2} \\ 0 \text{ else.} \end{cases} \qquad i,j = 1,2 \tag{3.4}$$

The facing with respect to a bearing is then stated as $\delta_{\psi_{ij}} = \alpha_{ij} - \psi_i$. Note that for the sensing model the heading of the camera ψ_i is taken and for the dynamics of the agents the direction of motion is important θ_i.

From the initial condition shown in Figure 3.6 it can be obtained that

$$\begin{bmatrix}-\sin(\theta_1) & \cos(\theta_1)\end{bmatrix} P_{g_{12}} g_{12}^* < 0$$

and $w_1 = 1$, that means that agent 1 would turn in negative direction. In this case it would increase the facing error $\delta_{\psi_{12}}$.

3.2.2 Turning Direction

In the limited FOV setup the turning direction plays an important role especially when the motion of an agent is constrained. To see that drawback of controller (3.2) we look at four different initial conditions shown in Figure 3.7. They refer to the different sign of $\delta_{\psi_{ij}}$ and the moving direction on a circle described in Section 2.3.3. In Figure 3.7a and 3.7c $\delta_{\psi_{12}} > 0$ and therfor agent 1 turns in positive direction (e.g. $v_{1_{\text{ang}}} > 0$). To obtain the sign of $v_{1_{\text{ang}}}$ we look at the projection of u_1 onto the facing direction of the camera. In Figure 3.7a we see that $v_{1_{\text{ang}}} < 0$ which results in a turning direction that increases the facing error. The same properties follow when we look at $\delta_{\psi_{12}} < 0$, in Figure 3.7b $v_{1_{\text{ang}}} > 0$, such that $\delta_{\psi_{12}}$ increases.

We now take a look at the value $\alpha_{\delta_{g_{12}}}$ introduced in Section 2.3.3, which refers to the moving direction on a the circle (e.g. (2.11)). In Figure 3.7c and 3.7d we obtain that the sign of $v1_{\text{ang}}$ will lead to a decreasing $\delta_{\psi_{12}}$. In both cases the bearing-angle-error $\alpha_{\delta_{g_{12}}}$ is positive.

In order to correct the turning direction, such that the facing error decreases, we take the sign of $\alpha_{\delta_{g_{12}}}$ into the closed loop (3.3) such that follows

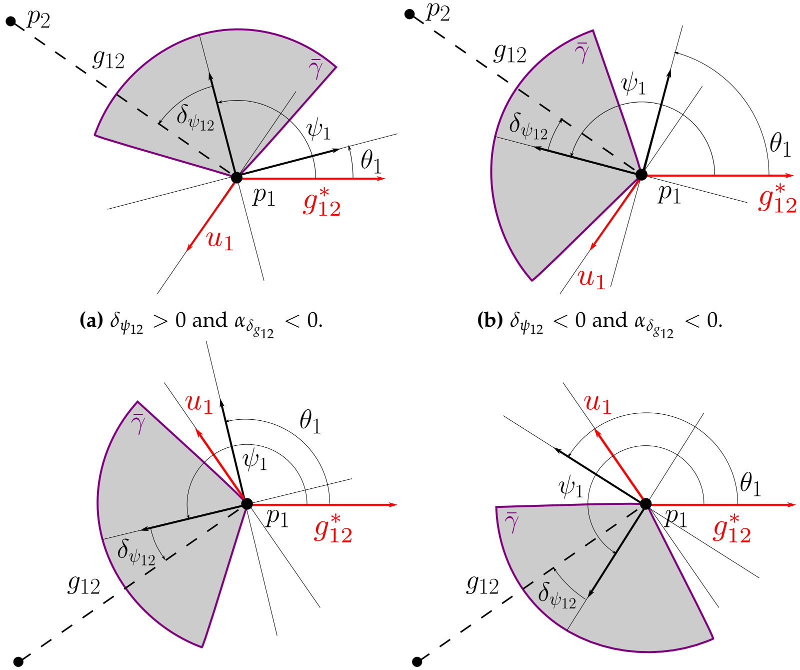
$$
\begin{aligned}
\dot{x}_1 &= -w_1 \begin{bmatrix}\cos(\theta_1) & \sin(\theta_1)\end{bmatrix} P_{g_{12}} g_{12}^* \cos(\theta_1) \\
\dot{y}_1 &= -w_1 \begin{bmatrix}\cos(\theta_1) & \sin(\theta_1)\end{bmatrix} P_{g_{12}} g_{12}^* \sin(\theta_1) \\
\dot{\theta}_1 &= -w_1 \begin{bmatrix}-\sin(\theta_1) & \cos(\theta_1)\end{bmatrix} P_{g_{12}} g_{12}^* \text{sgn}(\alpha_{\delta_{g_{12}}}) k_{\text{ang}} \\
\dot{x}_2 &= w_2 \begin{bmatrix}\cos(\theta_2) & \sin(\theta_2)\end{bmatrix} P_{g_{12}} g_{12}^* \cos(\theta_2) \\
\dot{y}_2 &= w_2 \begin{bmatrix}\cos(\theta_2) & \sin(\theta_2)\end{bmatrix} P_{g_{12}} g_{12}^* \sin(\theta_2) \\
\dot{\theta}_2 &= w_2 \begin{bmatrix}-\sin(\theta_2) & \cos(\theta_2)\end{bmatrix} P_{g_{12}} g_{12}^* \text{sgn}(\alpha_{\delta_{g_{21}}}) k_{\text{ang}}.
\end{aligned}
$$

here we also introduce a gain k_{ang} that we use as design parameter. In the next Section we will provide some simulation results for the two agent case and then increase the number of agents in the framework. Note that the subscript of $\alpha_{\delta_{g_{12}}}$ is now extended by the particular bearing that should be reached.

Turing in the direction to minimize the facing error is here introduced for the two agent case by the sign of value $\alpha_{\delta_{g_{12}}}$. Such an turning indicator can also be found for a network with more than two agents, which we discuss in Section 3.2.4. In the two agent case the desired facing direction do align naturally with the bearing, for more than two agents these assumptions does not hold. Therefor we introduce another notation for the correction term that is independent of bearings α_{δ_i} $i \in \mathcal{V}$, which refers to a specific value for each agent that indicates which direction the agent should turn (to decrease the facing error). Details on α_{δ_i} will be discussed in Section 3.2.4, where we introduce two different approaches.

3.2.3 Simulation for $n = 2$ agents

After adjusting the turning direction we can simulate the closed loop dynamics for the two agent case. The results are shown in Figure 3.8. The limited



(a) $\delta_{\psi_{12}} > 0$ and $\alpha_{\delta_{g_{12}}} < 0$.

(b) $\delta_{\psi_{12}} < 0$ and $\alpha_{\delta_{g_{12}}} < 0$.

(c) $\delta_{\psi_{12}} > 0$ and $\alpha_{\delta_{g_{12}}} > 0$.

(d) $\delta_{\psi_{12}} < 0$ and $\alpha_{\delta_{g_{12}}} > 0$.

Figure 3.7: Different initial conditions to obtain a correction term for the facing direction in (3.3).

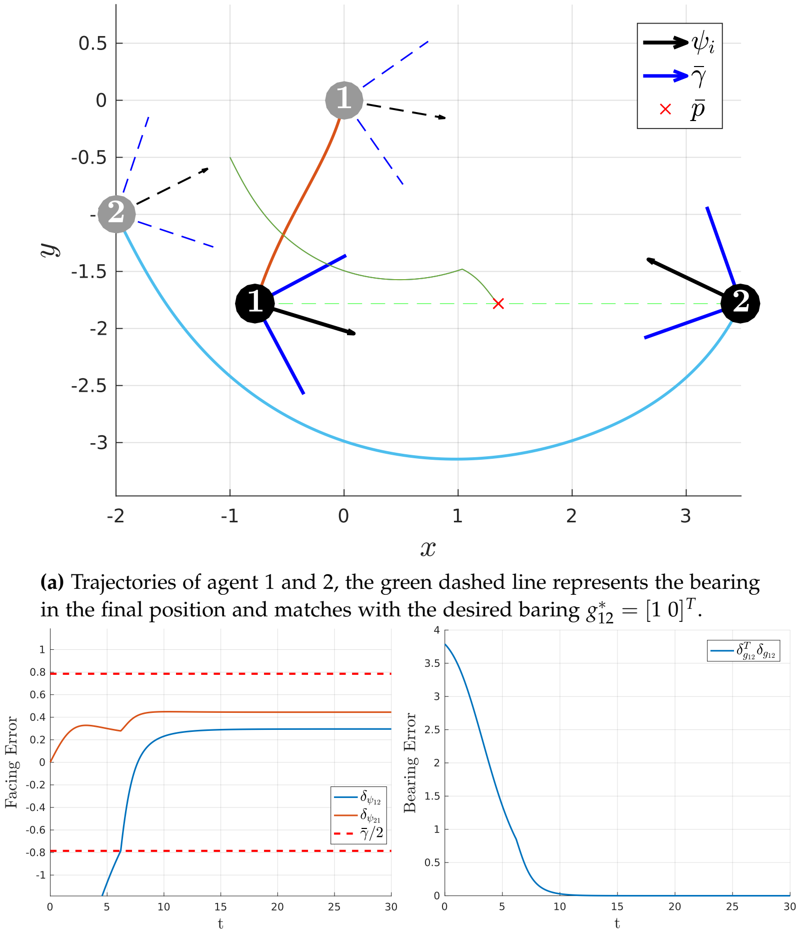
FOV is set to $\bar{\gamma} = \pi/2$ and therefor if $|\delta_{\psi_{ij}}| < \pi/4$,$(i,j) = \{(1,2),(2,1)\}$ holds the neighbor can be sensed. This can be seen in Figure 3.8b where the red dashed line indicates the bound of the FOV. Initially agent two can sense agent one but not the other way round. The initial condition are shown in Figure 3.8a with dashed lines and grey circles. The final position is shown in bold lines and black circles for the agents, the green bold line represents the center of the formation in general, denoted as $\bar{p}$, in the two agent case it refers to the middlepoint between them. It can be seen that in contrast to the single integrator simulations the distance is not invariant, even if both agent see each other. The desired bearing $g^*_{12} = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ is reached in the final positions, which can be seen by the bearing error illustrated in Figure 3.8c.

This however is only a simulation results which has not yet been proven analytically. In this simulation $k_{\text{ang}} = 1$ which does not take into account the dynamics of the bearing $\dot{\alpha}_{12}$. In contrast to the single integrator finding a gain that describes the bearing dynamics, such that the facing error constantly decreases, is not intuitive anymore. On the other hand we can change the static gain k_{ang} and analyze the performance.

Since we turned the facing direction around $+\pi/2$ with respect to the direction of motion we were able to meet the objective with the proposed controller (3.2) by only adding a correction term for turning direction. We did not design a new controller for the facing direction as we did in Section 2.2.

Here we simulated the case where agent two moves inside the FOV of agent one, which was for the integrator dynamics the most interesting case. Simulation for the other cases have the same behaviour as for the single integrator dynamics. From simulation results we assume that if one agent can sense its neighbor initially in the two agent case the desired bearing will be reached. Here we assume that the initial distance is sufficiently large.

Since we were able to show that the single integrator model also works for the $n > 2$ agent case we will provide simulations also for the unicycle model.



(a) Trajectories of agent 1 and 2, the green dashed line represents the bearing in the final position and matches with the desired baring $g^*_{12} = [1\ 0]^T$.

(b) Facing error during the simulation.

(c) Bearing error $\delta_{g_{12}} = g_{12} - g^*_{12}$.

Figure 3.8: Simulation for two agents with unicycle dynamics.

3.2.4 Simulation for $n > 2$ agents

If there are more than two agents each agent can have more than one neighbor. Moreover for the desired facing direction there is more than one possible desired value. We first introduce a desired facing direction $\hat{\psi}_i$. Figure 3.9 shows a simulation with unicycle dynamics (3.3) and $n = 3$ agents. The initial condition is such that agent one can sense agent two, agent two can sense agent three and agent three can sense both its neighbors. The desired facing direction is set to the neighbor that is closest

$$\hat{\psi}_i = \psi_i + \min_{j \in \mathcal{N}_i(t)} |\delta_{\psi_{ij}}|.$$

Closest in this context refers to the agent that is inside the FOV and has the smallest facing error. Here $\mathcal{N}_i(t)$ describes the set of neighbors that can be sensed by agent i at time t. By introducing $\hat{\psi}_i$ we also want to give some remarks on the correction term α_{δ_i} that we added to the closed loop in Section 3.2.2 to ensure that each agent is turning such that the facing error decreases. In the three agent case $\hat{\psi}_i$ refers to one of the bearing angles α_{ij}, the desired bearing angle then follows as α^*_{ij}. The correction term for agent i can then be stated as $\alpha_{\delta_i} = \alpha^*_{ij} - \hat{\psi}_i$, and is added to the closed loop dynamics (3.3) together with a design gain k_{ang}. Note that α^*_{ij} refers to the desired bearing of neighbor j for which $\min_{j \in \mathcal{N}_i(t)} |\delta_{\psi_{ij}}|$ holds. In Figure 3.9b the facing error is illustrated. Note that it only shows the errors that are relevant for the given simulation. Particularly interesting is $\delta_{\psi_{3j}}$, $j = 1{,}2$ where initially both agents can be sensed, due to the policy of facing the closes neighbor agent 3 looses track of agent 2 and only tracks agent 1. An interesting behaviour can be obtained between $10 < t < 15$ where $|\delta_{\psi_{31}}| \approx |\delta_{\psi_{32}}|$ and therefor the desired facing jumps between those two values, which leads to a chattering behaviour in the trajectories. The initial state from the simulation is set to

$$\chi = \begin{bmatrix} 0 & 0 & 2.62 & -1 & -1 & -1.92 & 2 & -2 & 1.4 \end{bmatrix}^T$$

and the limited FOV to $\bar{\gamma} = \pi/2$.

Since we have seen that the center of the formation is not invariant under the dynamics of the unicycle even in the two agent case we did not plot $\bar{p}(t)$ in Figure 3.9a. The desired formation in the three agent case is set to the equilateral triangle as described in Section 2.5. In Figure 3.9c the bearing error is illustrated and converges monotonically to zero. Without having an analytically proof, the simulation motivates to find a general expression for a set of initial conditions for which the bearing error decreases constantly.

Instead of facing the neighbor that is closest, another possibility is to face the middle of all neighbors that are inside the FOV, such that the desired bearing follows as

$$\hat{\psi}_i = \psi_i + \frac{1}{|\mathcal{N}_i(t)|} \sum_{j \in \mathcal{N}_i(t)} \delta_{\psi_{ij}}.$$

A simulation with four agents is shown in Figure 3.10 where the desired bearing is an infinitesimally rigid square, described by five bearings (see Section 2.5 for the desired bearing). Note that for this policy $\hat{\psi}_i$ does not necessarily refer to a bearing angle. Since our aim is to align the facing direction with the middle of all agents that are inside the FOV the desired final facing value ψ_i^*, does not necessarily refer to a desired bearing angle,

$$\psi_i^* = \frac{1}{|\mathcal{N}_i(t)|} \sum_{j \in \mathcal{N}_i(t)} \alpha_{ij}^*.$$

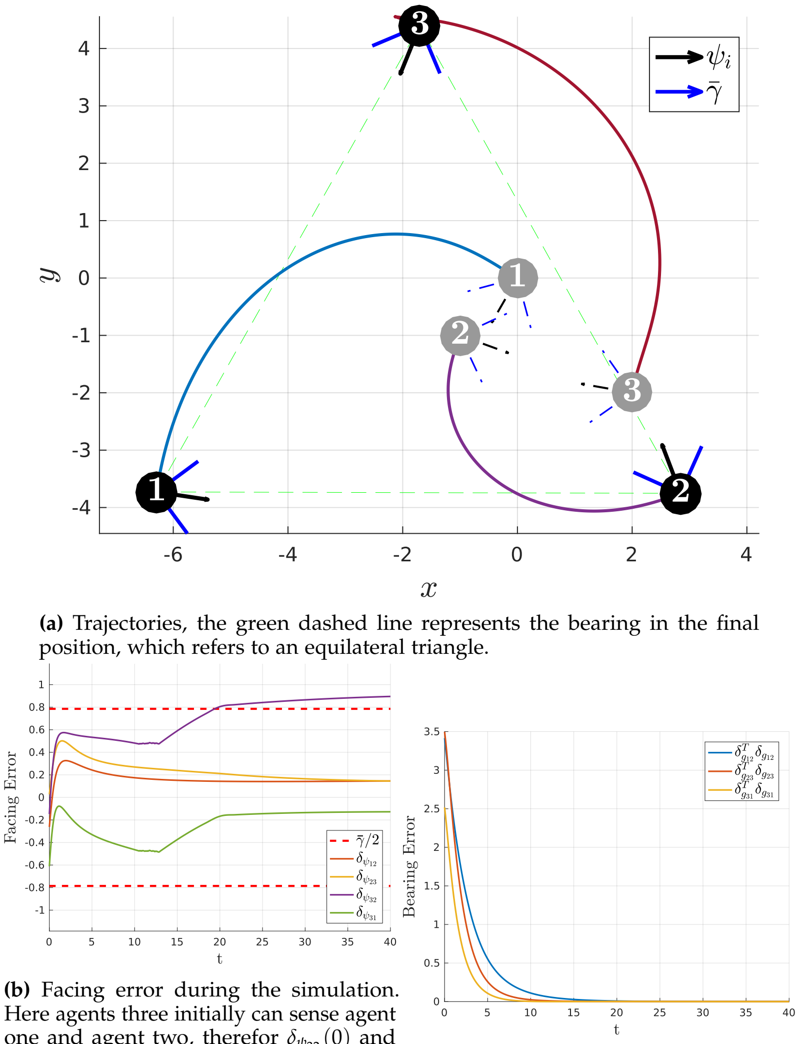
Depending on the set of neighbors, the desired facing for the final bearing, does also lie in the middle of the desired bearings. This behaviour can be seen in Figure 3.10a where agent 1 faces the middle of agent 2 and 3, once the desired baring is reached.

The correction term for the turning direction follows as $\alpha_{\delta_i} = \psi_i^* - \hat{\psi}_i$. By including the correction term and a gain k_{ang} the closed loop (3.3) can be simulated. The initial condition is set to

$$\chi = \begin{bmatrix} 0 & 0 & 2.62 & -1 & -1 & -1.92 & 2 & -2 & 1.4 & 0 & -3 & -0.7 \end{bmatrix}^T.$$

In contrast to the three agent case we increased the FOV of each agent, such that $\bar{\gamma} = 1.75 \triangleq 100°$. Initially agent 2 can sense only agent 3 since agent 4 is not one of its neighbors defined by the incidence matrix (see Section 2.5), during the movement agent 2 does not see agent 1 and therefor keeps track of agent 3.

The facing error illustrated in Figure 3.10b shows the relative bearing error corresponding to the bearings $\delta_{\psi_{ij}}$ $(i,j) \in \mathcal{E}$. From the given initial condition



(a) Trajectories, the green dashed line represents the bearing in the final position, which refers to an equilateral triangle.

(b) Facing error during the simulation. Here agents three initially can sense agent one and agent two, therefor $\delta_{\psi_{32}}(0)$ and $\delta_{\psi_{31}}(0)$ is between the dashed red lines.

(c) Bearing error $\delta_{g_{ij}} = g_{ij} - g^*_{ij}$ $(i,j) \in \mathcal{E}$.

Figure 3.9: Simulation for three agents with unicycle dynamics.

the desired bearing of a square is reached. However as for the three agent case there is no analytical proof that guarantees the result. There is also not set of initial conditions for which we know that the desired baring is reached. One major problem of finding sufficient analytical results is lacking knowledge of the final position of the agents.

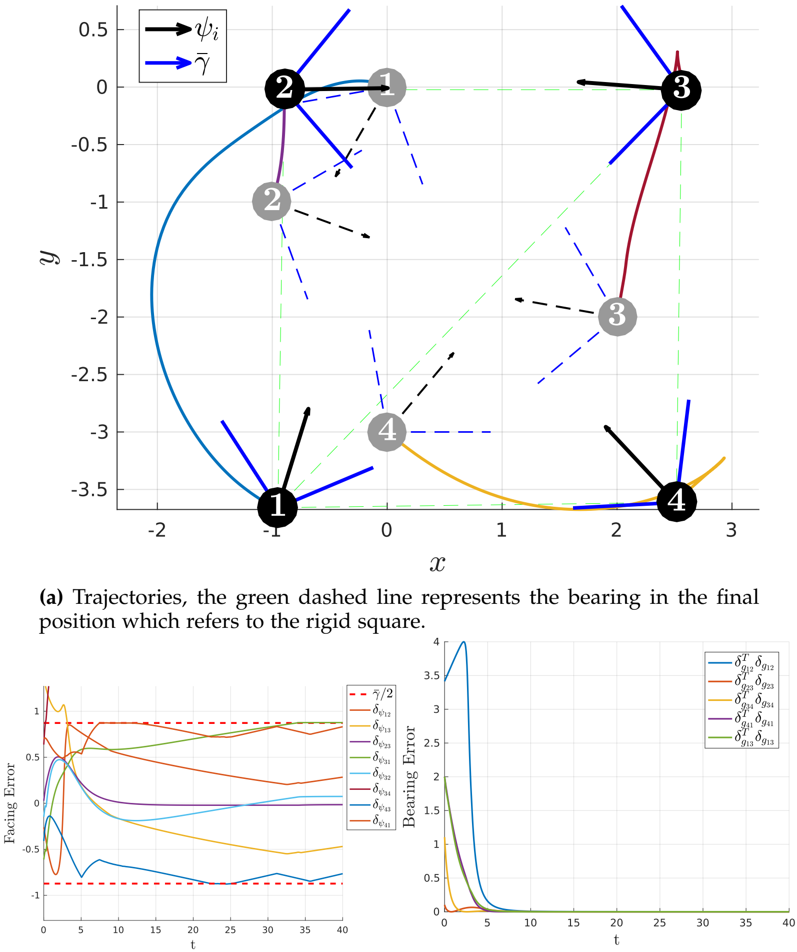
In the next section we implement the proposed controller with the correction term for the turning direction *TurtleBotII*-Robots.

3.3 Experimental results on *TurtleBots*

Motivated by simulation results for the unicycle dynamics we implemented the controller on *TurtleBots*. Each robot thereby runs its own controller, knowing who its neighbors are and what the desired bearing should be relative to them.

One major difference between working on simulations compared to experiments is the dynamic of the system and agents itself. In simulations every possible initial condition can be tested without any danger of destroying equipment. Experiments on the other hand do suffer from space constraints given by the lab, sensing constraints given by the library used, lighting conditions that influence the color recognition and signal delays caused by the network. Before running experiments the goal, initial conditions and the expected behavior should be known in all detail. Another thing that makes implementing algorithms on a real-world system time consuming is debugging if the actual behavior differs from the simulations. In this work we show an experimental result for the four agent case simulated in Section 3.2.4.

One experimental run is shown in Figure 3.11, where we compare the simulations with the data obtained by the OptiTrack cameras. For the simulations the controller gain is set to $k_{\text{ang}} = 5$. The blue, thin solid line refers to the simulation trajectories leading to a final position indicated by a grey circle with the corresponding first character of the robots name. The dashed black and blue line refer to the facing direction and the limited FOV respectively. Data obtained by OptiTrack, during the experiment, is shown in bold, red lines, that result in a final position illustrated by a black circle. Here the facing and limited FOV are marked by solid lines. Even though the



(a) Trajectories, the green dashed line represents the bearing in the final position which refers to the rigid square.

(b) Facing error during the simulation.

(c) Bearing error $\delta_{g_{ij}} = g_{ij} - g_{ij}^*$ $(i,j) \in \mathcal{E}$.

Figure 3.10: Simulation for four agents with unicycle dynamics.

simulation does not perfectly overlay with the experiment we can see that the shape of the trajectories is similar [1]. One issue during the experiment was that the agents did not start moving at the same time, even though they where connected by the range sensor. Another problem was the robustness of the color tracking during the movement, sometimes things in the lab where detected as neighboring agents since they had similar colors. Since the proposed controller relies only on that visual measurement wrongly detected objects lead to jumps in the control signals.

In this Chapter we have shown the process from transforming the bearing-only controller for single integrator dynamics to a unicycle model. Since the simulations for the $n = 2$ agents case looked promising we added more agents to the simulations. Finally we were able to adopt the controller on real-world robots and succeeded in forming a square. Therefor we used a turning correction term such that the facing error decreases and mounted the camera such that the facing does not align with the moving direction but is turned around $\pi/2$. The results of this chapter can be used to implement the bearing controller using only visual sensing. In order to identify if a desired bearing is reached by a set of initial conditions and the limited FOV constraint the analysis has to be studied in more detail.

In the next Chapter we will conclude this thesis and give some remarks on future research problems related to this work.

[1] A video of the agents can be found on YouTube (`https://youtu.be/Ph4sqeh5CiQ`), we filmed the agents by a GoPro camera, such that we can get top view perspective.

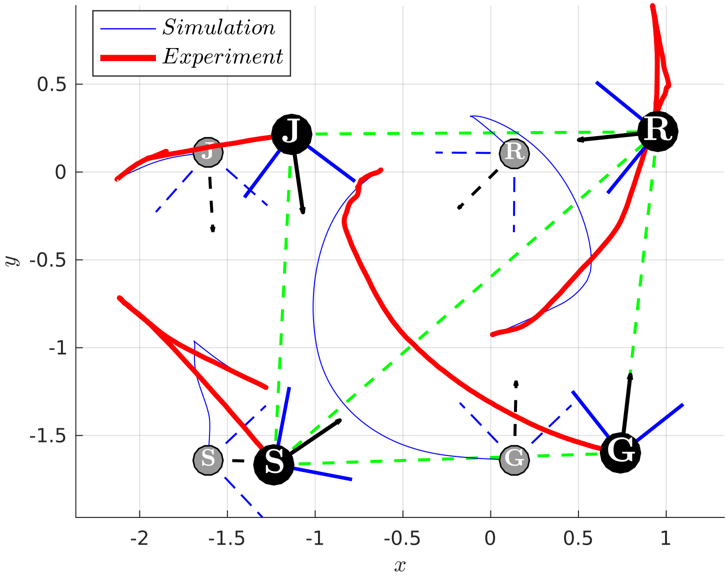


Figure 3.11: Comparing the experimental result with the simulation.

4 Conclusion

In this thesis we solved the bearing-only formation control problem with limited field-of-view sensing for the two agent case. To achieve this, we implemented a bearing-only control for the heading of the agent that ensures agents remain inside the FOV once they enter. We provided a complete analysis of the resulting system showing the approach can stabilize the desired formation from almost all initial conditions of the positions. In numerical simulations we were able to support our analysis, thereby we saw that it also works for networks with $n > 2$ agents.

The formal analysis is the subject of future work. Apart from the stability analysis itself, an interesting problem is to characterize the initial conditions together with the limited FOV. The characterization can give necessary conditions on the minimum FOV, that is required in order to stabilize a formation, or, given a limited FOV, it can be used to see if a desired formation will be reached. One major problem of the presented control strategy for the $n > 2$ agents case, is to obtain the final position of the agents when the desired formation is reached.

Compared to bearing-only formation control algorithms without limited visual sensing, the presented problem deals with directed interaction graphs. For networks with directed edges symmetry properties can not be used. Finding a general formulation for directed graphs that are able to stabilize infinitesimally bearing rigid formations might also be a good lead for solving the problem for an arbitrary number of agents. For the heading controller there is not yet an optimal policy available which neighbor to sense, if there is more than one agent inside the neighboring set. Since the goal is to achieve a controller that does not require communication, tracking not only the relative bearing of a neighbor but also its orientation might be a useful information.

Applying the bearing-only controller on a network with limited FOV is here mainly done on single integrators. However vehicles often do not be-have like integrators, since they can not move in arbitrary directions without motion constraints. We looked at simulations for the unicycle model, in future work more complex dynamics should be analyzed.

Eigenständigkeitserklärung

Ich versichere hiermit, dass ich, Daniel Frank, die vorliegende Arbeit selbstständig angefertigt, keine anderen als die angegebenen Hilfsmittel benutzt und sowohl wörtliche, als auch sinngemäß entlehnte Stellen als solche kenntlich gemacht habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen. Weiterhin bestätige ich, dass das elektronische Exemplar mit den anderen Exemplaren übereinstimmt.

Ort, Datum	Unterschrift

Literaturverzeichnis

[1] OpenCV. https://opencv.org/. [Accessed: May 14, 2018].

[2] Brian D.O. Anderson, Changbin Yu, and Baris¿ Fidan. Rigid Graph Control Architectures for Autonomous Formations. *IEEE Control Systems Magazine*, 28(6):48–63, 2008.

[3] Mohammad Mehdi Asadi, Amir Ajorlou, and Amir G Aghdam. Distributed control of a network of single integrators with limited angular fields of view. *Automatica*, 63:187–197, 2016.

[4] Microsoft Corporation. Kinect Sensor. https://msdn.microsoft.com/en-us/library/hh438998.aspx, 2012. [Accessed: May 14, 2018].

[5] Duarte Dias, Pedro Urbano Lima, and Alcherio Martinoli. Distributed formation control of quadrotors under limited sensor field of view. In *International Conference on Autonomous Agents & Multiagent Systems*, pages 1087–1095, 2016.

[6] Tolga Eren, Brian DO Anderson, A Stephen Morse, Walter Whiteley, Peter N Belhumeur, et al. Operations on rigid formations of autonomous agents. *Communications in Information & Systems*, 3(4):223–258, 2003.

[7] Baris Fidan, C Yu, and BDO Anderson. Acquiring and maintaining persistence of autonomous multi-vehicle formations. *IET Control Theory & Applications*, 1(2):452–460, 2007.

[8] Chris Godsil and Gordon F Royle. *Algebraic graph theory*, volume 207. Springer Science & Business Media, 2013.

[9] Julien M Hendrickx, Brian Anderson, Jean-Charles Delvenne, and Vincent D Blondel. Directed graphs for the analysis of rigidity and persistence in autonomous agent systems. *International journal of robust and nonlinear control*, 17(10-11):960–981, 2007.

[10] Laura Krick, Mireille E. Broucke, and Bruce A. Francis. Stabilization of infinitesimally rigid formations of multi-robot networks. pages 477–482, 2008.

[11] Logitech. Pro Webcam Ultra Wide Angle HD Webcam. https://www.logitech.com/en-us/product/12131?crid=34#specification-tabular. [Accessed: May 14, 2018].

[12] JihoonLee Daniel Stonier Yoonseok Pyo Michael Ferguson, Tully Foote. Turtlebot. https://wiki.ros.org/Robots/TurtleBot. [Accessed: May 14, 2018].

[13] Kwang-Kyo Oh and Hyo-Sung Ahn. Distance-based undirected formations of single-integrator and double-integrator modeled agents in n-dimensional space. *International Journal of Robust and Nonlinear Control*, 24(12):1809–1820, 2014.

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

[15] OpitTrack. Prime 13. https://optitrack.com/products/prime-13/. [Accessed: May 14, 2018].

[16] Fabrizio Schiano, Antonio Franchi, Daniel Zelazo, and Paolo Robuffo Giordano. A rigidity-based decentralized bearing formation controller for groups of quadrotor uavs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5099–5106, 2016.

[17] MH Trinh, S Zhao, Z Sun, D Zelazo, BDO Anderson, and HS Ahn. Bearing-based formation control of a group of agents with leader-first follower structure. *IEEE Transactions on Automatic Control*, 2016.

[18] Minh Hoang Trinh, Dwaipayan Mukherjee, Daniel Zelazo, and Hyo-Sung Ahn. Formations on directed cycles with bearing-only measurements. *International Journal of Robust and Nonlinear Control*, 28(3):1074–1096, 2018.

[19] TurtleBot. TurtleBot2. https://www.turtlebot.com/turtlebot2/, 2012. [Accessed: May 14, 2018].

[20] Daniel Zelazo, Antonio Franchi, Heinrich H Bülthoff, and Paolo Robuffo Giordano. Decentralized rigidity maintenance control with range measurements for multi-robot systems. *The International Journal of Robotics Research*, 34(1):105–128, 2015.

[21] Daniel Zelazo, Antonio Franchi, and Paolo Robuffo Giordano. Rigidity theory in SE(2) for unscaled relative position estimation using only bearing measurements. In *European Control Conference*, pages 2703–2708, Strasgourg, France, jun 2014.

[22] Shiyu Zhao, Dimos V Dimarogonas, Zhiyong Sun, and Dario Bauso. A general approach to coordination control of mobile agents with motion constraints. *IEEE Transactions on Automatic Control*, 63(5):1509–1516, 2018.

[23] Shiyu Zhao and Daniel Zelazo. Bearing-based formation stabilization with directed interaction topologies. In *IEEE Conference on Decision and Control*, pages 6115–6120, 2015.

[24] Shiyu Zhao and Daniel Zelazo. Bearing rigidity and almost global bearing-only formation stabilization. *IEEE Transactions on Automatic Control*, 61(5):1255–1268, 2016.