Formation tracking with multiple mutually coupled quadrotor UAVs on SE(3)

Master's Thesis
DC 2019.002

Author:
M.F.A. van de Westerlo

Supervisor:
prof.dr. H. Nijmeijer
Coach:
dr.ir. A.A.J. Lefeber

Eindhoven University of Technology
Department of Mechanical Engineering
Dynamics & Control Group

Eindhoven, January 18, 2019

Summary

The concept of using several agents that partly rely on their interaction in order to obtain specific combined behavior seems very interesting for the next generation of control systems, while in the mean time progression persists in ever smarter design choices for control. Based on recent stabilizing results for the quadrotor UAV on SE(3) that apply almost globally, we want to synchronize multiple quadrotor UAVs. We feel like the specific combination between both worlds, a relatively complex control strategy based on recent research as well as utilizing a well established coupling technique, opens up a promising direction for progression in multi-agent systems. For synchronized systems of multiple quadrotor UAVs, which can be seen as one new system, we expect cost, robustness and efficiency benefits with respect to single-agent systems, due to potential parallelism and scalability. In fact, since multi-agent systems are able to utilize their combined lifting power and share sensing capabilities, some tasks can only be executed with multiple agents. In order to develop this multi-agent system of quadrotor UAVs, we first try to obtain a very similar model with unicycle robots, so that we first have a system with reduced complexity to work with. We start by designing a mobile robot reference tracking algorithm on SE(2) that is almost globally asymptotically stable, using cascade system theory. After that, we introduce a virtual reference structure that provides a set of feasible reference trajectories for multiple agents. In order to ensure that the mobile robots stay in formation even when one of the robots is perturbed, we include mutual coupling in the system by adding coupling errors in the generalized coordinates of the position tracking subsystem. The cascaded system is proven almost global asymptotically stable based on cascade system theory. Furthermore, simulations show the desired behavior. Subsequently, we follow the same approach for the quadrotor UAV on SE(3), to eventually prove almost global asymptotic stability of the system under mutual coupling. The behavior is analyzed in simulation of specific test scenarios. It is shown that increasing the cost weight on the coupling error can increase coupling at the expense of tracking the individual trajectories.

Acknowledgments

First of all, I would like to express my gratitude to prof. dr. H. Nijmeijer. Your vision was stimulating throughout my entire studies and one of many reasons that I got so enthusiastic about the field of dynamics and control. Thank you for the inspiration, valuable feedback, and overall support during my Master's thesis.

Furthermore, I would like to thank my coach dr. ir. A.A.J. Lefeber. Thank you for your activating and inspiring role in this project. During our weekly meetings you always brought an extreme amount of knowledge and professional but kind guidance to the table. Thank you for all the effort, feedback, and patience that you have put into our project.

A special thanks goes out to friends and family. Friends on the TU/e, thank you for your support during coffee breaks and drinks. Other friends and family, thank you for your sympathy and involvement with me and my studies.

Lastly, to my parents, sister, and girlfriend, thank you for your love, the ever lasting support, and unconditional encouragement during my studies.

Marcel van de Westerlo
Mierlo, December 20, 2018

Contents

Nomenclature

Reference frames

$\mathcal{B}_i$	Right-handed orthonormal body-fixed coordinate frame of agent i
$\mathcal{D}_i$	Right-handed orthonormal frame that describes the desired attitude
$\mathcal{F}$	Right-handed orthonormal virtual center-frame of the formation
$\mathcal{I}$	Right-handed orthonormal inertial frame in North-East-Down (NED) configuration
$\mathcal{R}_i$	Right-handed orthonormal reference frame for agent i, fixed to the virtual body of reference agent i

Number sets

$\mathbb{N}$	The set of nonnegative integers
$\mathbb{R}^n$	The n-dimensional Euclidian space
$SE(n)$	The n-dimensional special Euclidian group
$SO(n)$	The n-dimensional special orthogonal group
$so(n)$	The set of $n \times n$ skew symmetric matrices

Operators

$\mathrm{diag}(\cdot)$	Function that provides a row vector that contains all diagonal elements of its argument
$\sigma(\cdot)$	Smooth vector saturation function for a vector of size $n \times 1$
$\mathrm{Tr}(\cdot)$	Function that provides a scalar equal to the sum of all diagonal elements its argument
$S(\cdot), \bar{S}(\cdot)$	Skew symmetric matrix of size 3×3 and 2×2, respectively
$\dot{x}$	Time-derivative of a system state x
$A^\top$	The transpose of a matrix
$x^\top$	The transpose of a vector

Constants and variables

$\nu_i, \nu_{i,r}$	Body-fixed velocity and reference velocity vector, respectively, expressed with respect to $\mathcal{B}_i$ and $\mathcal{R}_i$ of agent i

Ω_i	Angular velocity of propeller i
$\omega_i, \omega_{i,r}$	Angular velocity and angular reference velocity, respectively, expressed with respect to $\mathcal{B}_i$ and $\mathcal{R}_i$ for agent i
ϕ_i	Roll angle of aerial body i; also used as orientation angle for grounded vehicle
ψ_i	Yaw angle of body i
$\rho_i, \rho_{i,r}$	Position of the origin of body-fixed reference frames $\mathcal{B}_i$ and $\mathcal{R}_i$ of agent i, respectively
$\rho_{i,e}$, $\nu_{i,e}$	Generalized position and velocity coordinates for agent i, respectively; also referred to as the position and velocity tracking errors for agent i
θ_i	Pitch angle of body i
g	Gravitational acceleration constant
I	Identity matrix of size $n \times n$
J_i	Inertia of agent i
$K_{i,x}$	Control parameter matrix (gain matrix of size $n \times n$) for agent i and states x
$k_{i,x}$	Control parameter (scalar gain or cost) for agent i and states x
m_i	Mass of agent i
N_i	The set of adjacent agents for agent i
R_f	Rotation matrix which transforms $\mathcal{F}$ to $\mathcal{I}$
$R_{i,d}$	Rotation matrix which transforms $\mathcal{D}_i$ to $\mathcal{R}$
$R_{i,r}$	Rotation matrix which transforms $\mathcal{R}_i$ to $\mathcal{I}$
R_i	Rotation matrix which transforms $\mathcal{B}_i$ to $\mathcal{I}$
u_i	Virtual input vector of agent i
$v_i, v_{i,r}$	Body-fixed forward velocity and reference velocity of agent i (size 1×1), respectively, expressed with respect to $\mathcal{B}_i$ and $\mathcal{R}_i$

Chapter 1

Introduction

1.1 Background and perspective

The quadrotor type of Unmanned Aerial Vehicle (UAV) is the most popular type of UAV and has four propellers that are used to lift and steer through the 3D space. Other types of UAVs are, e.g., planes, helicopters, bird or insect models and multirotors with more than four propellers. One of the reasons for the quadrotor UAV to be frequently studied is the relatively simple control with respect to UAVs with more propellers, like the hexacopter or octacopter. For the quadrotor UAV, we 'only' have to take four propellers into account and are able to align the body-fixed reference frame with the propellers; of course dependent on the specific configuration of the quadrotor UAV. Another reason for the popularity of the quadrotor is the fact that they are much more agile than aircraft with fixed wings, as they can freely position themselves in space. Winged aircraft, for instance, need to make sure they do not drop out of the sky by controlling their forward velocity and pitch angle. Furthermore, most multirotor systems can take-off vertically, while most aircraft with fixed wings can not. However, quadcopters with respect to fixed wing aircraft are typically less efficient in flying forward when comparing battery consumption, which is why recently even hybrid models are discussed to obtain the best from both worlds [1].

A welcome property of UAVs is that we can capture the behaviour in a model to automate several processes. Many efforts have already been made to derive the dynamics of a quadcopter [2, 3]. Although control of quadrotor UAVs is not particularly simple, researchers and companies have already been able to make the operation for the end users relatively easy and provide stable flight, by using the sensing capabilities to stabilize the drone instead of merely manual joystick controls. Because of this ease of operation for the end users, the use of drones is no longer limited to military activities and large high-tech companies; also smaller companies and individuals use drones as a tool and kids even use them as a toy, since we can now partly rely on the autonomy of the drone during flight.

A key feature of UAVs is their aereal operation, which means that their domain of operation is the space. A downside from operating in the space is that the quadrotor constantly has to generate thrust to overcome the gravitational acceleration, which makes battery power a huge limiting factor for operation. Besides the power consumption, one can imagine that during research and development processes many quadrotors have been destroyed upon impact with the ground, as a result of technical difficulties and flaws. The quadrotor UAV is an underactuated system, which means that the configuration space is larger than the number of inputs. For the quadrotor UAV, this means that in order to move forward, the quadrotor has to slightly tilt forward. Because of this coupling of inputs and states, we are not entirely free to choose the attitude and spatial position simultaneously. Many linear control strategies exist, assuming small angular manouvres [4–6]. By now, there is a vast amount of researches considering trajectory tracking with nonlinear control techniques, like feedback linearization [7], backstepping [8], MPC [9], and sliding mode control [10]. Recently, a controller has been presented that achieves uniform almost global stability of the error tracking dynamics for a quadrotor UAV on SE(3) [11]. A benefit of this approach with respect to other nonlinear controllers for quadrotor UAVs is the fact that singularities of Euler angles and ambiguity of quaternions are avoided, allowing large angular maneuvers and providing an almost global instead of a local result. Since we can now already ensure stable flight of more challenging trajectories, we can accept even more challenging tasks to execute, which basically increases the system agility. However, some tasks can never be executed by a single agent or are simply better suited to be executed by multiple agents. In order to make several agents work on the same task, we somehow have to obtain cooperation in order to make sure that the agents help and not work against each other. Therefore, as a possible improvement to the recently developed controller for a quadrotor UAV on SE(3), we would like to research the synchronization of multiple of these systems.

1.2 Multi-agent cooperation

Many efforts have already been made to synchronize multiple agents in order to utilize their combined capability to complete tasks. In this section, we first investigate the motivation of using multiple agents instead of one. After that, we consider which coupling methods exist and what effects these methods have on the obtained system.

1.2.1 Motivation

Using multiple agents with respect to a single possibly more complex agent provides cost and robustness benefits [12], since the system can consist of multiple fairly simplistic agents and is still able to operate when a single agent is down. In fact, it might be more efficient to use multiple quadrotor UAVs, since the number of agents can easily be scaled to the task complexity and size. These multi-agent systems also facilitate parallelism [12] and procedures like smart charging and maintenance protocols can be applied in order to provide a system that allows for constant 'around-the-clock' operation. Multi-agent systems can be seen as super-additive, since the combined capabilities are more than just the sum of the capabilities of all individuals. This super-additivity property results from the fact that multiple agents can also do specific tasks that can never be done individually [13], like how humans need two persons in order to lift a table. Furthermore, the combined sensing capability can be utilized by the group as a result of mutual communication [14]. In order to fully use the combined value, we can not simply take a number of individual agents, we also need to have some control over their combined behavior and objectives. This raises the desire to have some level of synchronization, based on coupling between agents in the multi-agent system. The exact coupling strategy is of great importance to the resulting behavior of the system.

1.2.2 Coupling methods and approaches

For the class of robots consisting of mobile and (aero)nautical robots, mainly three strategies exist: the hierarchical method of master-slave synchronization, the decentralized behavioral method that works with simple laws for the individuals, and the coordinated virtual structure coupling [15]. The latter, as a result of mutual communication, provides a good ability to obtain a specific formation with multiple agents in a coordinated manner, while master-slave coupling is particularly capable of working with individual differences between the agents, since the slave device then simply tries its best to follow the master [16]. Currently, we do not focus on the behavioral method, since this method does not provide the agile coordinated system that we want; the behavioral method focuses on combined behavior based on simple individual laws like obstacle avoidance and mapping. The main problem for trying to obtain a formation with the master-slave structure is the fact that when a disturbance is presented to a slave, the rest of the agents do not receive information regarding this disruption so they are unable to react cooperatively to maintain the formation [13]. Therefore, master-slave synchronization is often used for teleoperation where the master device is human operated and the slave is located in, e.g., a hazardous location like in deep water [16]. In [15], a virtual structure approach is provided to mutually couple multiple unicycle robots based on cascade system theory. Even though this approach provides enormous inspiration for the quadrotor UAV, we also want to point out that the attitude subsystem provides tracking behavior that we can not allow for a quadrotor UAV; the orientation error of the unicycles is controlled to the absolute origin, even when the initial error is greater than a full rotation, which for a quadrotor UAV would result in full rotations around any axis and is therefore unallowed. Alternatively, considering orientation control of a unicycle robot on SE(2) prevents us from using the virtual structure approach from [15] and therefore we are unlikely to be able to use the same approach for a quadrotor UAV. Thus, in order to provide an answer to the desire of synchronizing multiple quadrotor UAVs, we have to reconsider the exact mutual coupling approach in this thesis. Some of the most widely adopted coupling approaches implement coupling functions to the reference structure [17,18], directly in the control law [15] or in the generalized coordinates [13].

1.3 Problem definition and objectives

Based on previous research that has been carried out at the TU/e [11,19], it is clear that a promising control strategy on SE(3) for a single quadrotor is provided, since the developed controller provides almost global stability and avoids singularities and ambiguity in the orientation control; allowing for large angular maneuvers. However, the operation of an aerial multi-agent system with this controller is still not investigated. Based on the literature results from the previous section, we intend to implement mutual coupling in order to synchronize operation of multiple quadrotor UAVs over master-slave coupling. In this thesis, our aim is to provide this mutual coupling strategy for multiple quadrotor UAVs, so that we are able to operate multiple quadrotors at once as one synchronized system. The idea is that the end user only has to prescribe some simple rules for the multi-agent system as a whole while the agents ensure to stay in formation even when agents are disturbed, by mutually cooperating in order to obtain the desired spatial lay-out. Besides the coupling strategy, we want to provide the actual controllers that stabilize the developed coupled system, in which the individuals are based on [11]. By utilizing the previous results from [11], we directly aim to provide a system that is agile, almost globally stable and synchronized. After the full development including proof of stability and simulation of test-cases, we want to provide some primary steps towards a real world system with two quadrotor UAVs. These steps consist of the development of an external localization and identification algorithm that works for a system of two different quadrotor UAVs, as the previous method used by [19] and [20] only used one quadrotor UAV during experiments. This external localization source is required since the drones are not capable of directly providing an on-board position estimate for themselves without a computationally intensive on-board algorithm like SLAM. Furthermore, we have to provide a network architecture that enables both drones to communicate with the external localization source, the supervisory controller and each other so that we are able to communicate with and coordinate the multi-agent system over the network. Currently, the on-board DHCP server of the quadrotor UAV is used [19–21], but this does not scale to multiple quadrotor UAVs, as each of the quadrotor UAVs hosts a new network. Furthermore, since calculation power is sparse on relatively low-cost micro-controllers that are used on board of quadrotor UAVs, we might want to consider a quaternion implementation. However, due to ambiguity that quaternions naturally cope with [22], we prefer to develop and analyze the controllers by using Euler angles, as in [19]. In order to reach the specified goals, a list of sub-objectives is explicitly provided:

- Follow the design and analysis of the previous work by [19] and [11] in order to find alternative design choices that might be beneficial for the multi-agent system.
- Develop the architecture that defines a virtual structure to track for the desired number of

agents. When developing a control strategy for a scalable number of n agents, there has to be a systematic approach to obtaining a set of n feasible reference trajectories to track, but more importantly together define the wanted formation shape, position and attitude.

- Find a coupling structure that provides a system that we are able to stabilize.
- Find stabilizing control laws for the agents in the mutually coupled system.
- Enable experimentation by providing a localization method that enables to locate and identify multiple quadrotor UAVs for experiments, as well as a network architecture that provides (possibly meshed) communication between all involved parties. Preferably both have to facilitate scalability of the system with respect to the number of agents in the system.
- Implement the system for simulation and and test the behavior by simulating test cases.

1.4 Outline of the thesis

This thesis has the following structure. In Chapter 2, we start off by including some preliminary notions that are used extensively throughout this thesis. Chapter 3 considers the (sub) objectives, but for mobile robots, in order to provide possible answers first for a problem with reduced complexity. Then, Chapter 4 includes similar steps as Chapter 3, but for the quadrotor UAV, which provides a spatial instead of a planar formation-tracking problem. In that chapter, also simulation results are included in order to provide insight in the behavior of the system. Additionally, in Appendix A, some further information is included in order to obtain a system for experimentation.

Chapter 2

Preliminaries

Although many relations and results from previous research are now commonly known and somewhat trivial, some relations and results are not. For this reason, many of the nontrivial preliminary results that are utilized throughout this thesis are recalled, by partly or full adoption herein. In this section, we introduce notation, definitions, theorems and lemmas that are used throughout the remainder of this thesis. We start off by recalling some developed methods for attitude representation, by some more general mathematical notations, relations and definitions. After that, we include stability results and definitions that are utilized for the proof of stability later on in this thesis. At the end we include some definitions from graph theory in order to systematically work with a multi-agent system.

2.1 Attitude representation

As for all aerial control systems, the attitude of a quadrotor UAV is very important. Because of this importance, different attitude representation techniques exist, as a result of the interest of researchers for an extended period of time. Each of the representation techniques has their own strengths, benefits and weaknesses. For example, although quaternions are very efficient in attitude representation [23–26], they typically show sign ambiguity. It is very important to be able to define the attitude of the spatial device unambiguously. Therefore, for controller design we choose to use the special orthogonal group $SO(n)$ of order n. However, we are free to choose a quaternions approach for implementation of the same controllers. Then, we utilize the best from both worlds; providing unambiguous controllers that are efficiently implemented. Therefore, in this section we first introduce the attitude representation with Euler angles, together with some helpful results and theories from previous research. After that, we provide the relation between rotation matrices and quaternions.

2.1.1 Rotation matrices and Euler angles

We define the attitude of a body i, represented by a body-fixed frame $\mathcal{B}_i$, relative to a world-fixed inertial frame $\mathcal{I}$, by a rotation matrix R_i. If we consider a planar example, there is only one rotation angle involved, which we call the orientation angle, defined positive in the rotation direction from the x-axis to the y-axis; we assume all involved frames in this thesis to be right-handed. In order to define the transformation from $\mathcal{B}_i$ to $\mathcal{I}$ for a body i in the planar example, we have the rotation matrix

$$R_i = \begin{bmatrix} \cos\phi_i & -\sin\phi_i \\ \sin\phi_i & \cos\phi_i \end{bmatrix} \in SO(2), \tag{2.1}$$

in which the rotation angle ϕ_i for body i (or body-fixed frame $\mathcal{B}_i$) relative to the inertial frame $\mathcal{I}$ is denoted by ϕ_i and $SO(n) = \{\mathbb{R}_i^{n\times n} \mid \det(R_i) = 1, R_i^\top R_i = 0\}$ denotes the Special Orthogonal group of order n. For any matrix $R_i \in SO(n)$ holds [27]:

- $R_i^\top = R_i^{-1}$, so $R_i R_i^\top = I$, which means that $R_i^\top$ describes the reversed rotation of R_i
- The determinant of a rotation matrix has to equal $\det(R) = 1$ in order to only describe a rotation without any length change as a result of the transformation, i.e., a proper rotation
- Since both R_i and $R_i^\top$ are rotation matrices, each column of R_i and $R_i^\top$ has to be mutually orthogonal
- Since both R_i and $R_i^\top$ are rotation matrices, each column of R_i and $R_i^\top$ has length 1 (has to be a unit vector)

For a rotation from $\mathcal{I}$ to $\mathcal{B}_i$ in a 3D (spatial) example, we consider a rotation matrix $R_i \in SO(3)$. The rotation between two spatial frames $\mathcal{B}_i$ and $\mathcal{I}$ can be defined in several ways, as the rotation order is of importance. As an example, by using the axis-angle convention, we can define a rotation between two spatial frames as an axis with a rotation angle around that axis. However, the most common way to represent this transformation from $\mathcal{I}$ to $\mathcal{B}_i$ and thus define the attitude of the body i, is the use of Euler angles [24]. Intentionally chosen equal to [11, 19], define this rotation matrix $R_i \in SO(3)$ by means of Euler angles, as

$$\begin{aligned} R_i &= R_{z,i}(\psi_i) R_{y,i}(\theta_i) R_{x,i}(\phi_i), && \text{(2.2a)} \\ &= \begin{bmatrix} \cos\psi_i & -\sin\psi_i & 0 \\ \sin\psi_i & \cos\psi_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_i & 0 & -\sin\theta_i \\ \sin\theta_i & 0 & \cos\theta_i \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & \cos\phi_i & -\sin\phi_i \\ 0 & \sin\phi_i & \cos\phi_i \\ 1 & 0 & 0 \end{bmatrix}, && \text{(2.2b)} \end{aligned}$$

which are also called roll, pitch, and yaw angles (RPY), since the rotation $R_{x,i}(\phi_i)$ can be seen as roll, $R_{y,i}(\theta_i)$ as pitch and $R_{z,i}(\psi_i)$ as yaw angle when we align the axes of the body-fixed frame $\mathcal{B}_i$ with the corresponding direction relative to the body i; this is commonly used in the (aero)nautical field of research as it does generally not yield a similar orientation as the reversed order $(R_{x,i}(\phi_i) R_{y,i}(\theta_i) R_{z,i}(\psi_i))$ [19], best explained in [28]. Besides the attitude representation by the rotation order, the most common axis convention for aerial vehicles or projectiles is the North-East-Down (NED) frame, which we adopt in order to overlap with [11]. Another axis convention is for instance the East-North-Up (ENU) convention, but the NED convention is preferred in aviation since we can make the forward and downward direction align with the positive axes of the coordinate frame. Along with rotation matrices come their time-derivatives. However, resulting from the special (orthogonality) properties that are states above for rotation matrices, special results apply for the time derivative of a rotation matrix.

Theorem 2.1.1. *Consider a rotation matrix $R \in SO(3)$. Define three generators for $SO(3)$ that correspond to the three standard axis [29]*

$$G_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \qquad G_y = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \qquad \textit{and} \quad G_z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{2.3}$$

Consider $\omega \in \mathbb{R}^3$ and define

$$\begin{aligned} S(\omega) &:= \omega_x G_x + \omega_y G_y + \omega_z G_z, && \text{(2.4a)} \\ &= \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \in so(3). && \text{(2.4b)} \end{aligned}$$

The time derivative of a rotation matrix $R \in SO(3)$, then equals

$$\dot{R} = RS(\omega), \tag{2.5}$$

which is a very important result that is widely used. In order to prove this result, let us follow [30]. The skew symmetric operator $S(\cdot)$ *has properties that follow from the fact that*

$$S(a)b = a \times b, \tag{2.6}$$

with $a, b \in \mathbb{R}^{3\times 1}$*, which allows to apply operations that hold for the cross-product operator. It holds that*

$$a \times b = -b \times a, \tag{2.7}$$

and for an invertible $M \in \mathbb{R}^{3\times 3}$*, it holds*

$$(Ma) \times (Mb) = \det(M)(M^{-1})^\top (a \times b), \tag{2.8}$$

which under proper rotations, i.e. $\det(M) = 1$ *and* $M^\top M = I$*, meaning that* $M = R$ *is a rotation matrix, reduces to*

$$(Ra) \times (Rb) = R(a \times b). \tag{2.9}$$

From (2.6) *and* (2.9) *it follows [31]*

$$S(Ra)Rb = RS(a)b, \tag{2.10}$$

which holds for any b*, so we find*

$$S(Ra)R = RS(a), \tag{2.11}$$

and

$$S(Ra) = RS(a)R^\top. \tag{2.12}$$

Besides, from (2.4) *it follows*

$$S(a)^\top = -S(a). \tag{2.13}$$

Now, as addition to (2.4)*, let us define the skew symmetric generator for* $so(2)$ *[29]*

$$\bar{G} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \tag{2.14}$$

so that we find the skew symmetric operator

$$\bar{S}(\omega) := \omega \bar{G} = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}, \tag{2.15}$$

with $\omega \in \mathbb{R}$ *and* $\bar{S}(\omega) \in so(2)$*, from which it also follows that* $\bar{S}(\omega)^\top = -\bar{S}(\omega)$*. The time derivative of a rotation matrix*

$$R(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \in SO(2). \tag{2.16}$$

then equals

$$\dot{R} = R\omega\bar{G} = R\bar{S}(\omega), \tag{2.17}$$

and for $R \in SO(2)$*, based on* (2.15) *and* (2.16)*, we find the property*

$$\bar{S}(\omega)R = R\bar{S}(\omega). \tag{2.18}$$

As shown by (2.16), a rotation matrix $R \in SO(2)$ is defined by a single angle ϕ. However, sometimes we want to have some kind of a reversed relation, so that we define the angle based on a specific rotation matrix. By looking at (2.16), we notice that the diagonal only consists of $\cos\phi$, which already provides information about the angle ϕ. This approach of looking at the diagonal is often used as a method to define the rotation angle based on a rotation matrix for the proof of stability [30,32,33].

Definition 2.1.1. *Define, the trace of a matrix $A \in \mathbb{R}^{n\times n}$*

$$\mathrm{Tr}(A) = \sum_{i=1}^{n} a_{ii}, \tag{2.19}$$

with a_{ii} representing the diagonal elements of A and n denoting the dimension of the square matrix. The trace thus provides the sum of all diagonal elements. As a result, from (2.19) *we find*

$$\mathrm{Tr}(M) = \mathrm{Tr}(M^\top). \tag{2.20}$$

For $M = AB$, provided from the definition of matrix products [34, Chapter 2.1]

$$(AB)_{ii} = \sum_{j=1}^{m} a_{ij} b_{ji}, \tag{2.21}$$

which together with (2.19) *leads to*

$$\mathrm{Tr}(AB) = \sum_{i=1}^{n} (AB)_{ii} = \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij} b_{ji} = \sum_{j=1}^{m}\sum_{i=1}^{n} b_{ji} a_{ij} = \mathrm{Tr}(BA), \tag{2.22}$$

In fact, since both A and B could consist of a product of other matrices, we find that the trace operator is invariant under cyclic operations [34, Chapter 2.1]

$$\mathrm{Tr}(ABC) = \mathrm{Tr}(CBA) = \mathrm{Tr}(BCA), \tag{2.23}$$

and

$$\mathrm{Tr}(ABCD) = \mathrm{Tr}(BCDA) = \mathrm{Tr}(CDAB) = \mathrm{Tr}(DABC), \tag{2.24}$$

while arbitrary operations are not allowed. In general

$$\mathrm{Tr}(ABC) \neq \mathrm{Tr}(ACB). \tag{2.25}$$

For example, define

$$A = \begin{bmatrix} 1 & 3 \\ 5 & 9 \end{bmatrix}, \qquad B = \begin{bmatrix} 3 & 1 \\ 4 & 5 \end{bmatrix}, \qquad \text{and} \qquad C = \begin{bmatrix} 5 & 2 \\ 6 & 4 \end{bmatrix} \tag{2.26}$$

then

$$\mathrm{Tr}(ABC) = \mathrm{Tr}\left(\begin{bmatrix} 171 & 94 \\ 555 & 302 \end{bmatrix}\right) = 473, \tag{2.27}$$

$$\mathrm{Tr}(ACB) = \mathrm{Tr}\left(\begin{bmatrix} 125 & 93 \\ 421 & 309 \end{bmatrix}\right) = 434, \tag{2.28}$$

$$\mathrm{Tr}(CAB) = \mathrm{Tr}\left(\begin{bmatrix} 177 & 180 \\ 294 & 296 \end{bmatrix}\right) = 473. \tag{2.29}$$

Furthermore, from (2.19) *follows*

$$\operatorname{Tr}(X+Y) = \operatorname{Tr}(X) + \operatorname{Tr}(Y), \tag{2.30}$$
$$\operatorname{Tr}(rX) = r\operatorname{Tr}(X). \tag{2.31}$$

Lastly, since the trace operator $\operatorname{Tr}(\cdot)$ *is a linear operator, for the time derivative holds*

$$\frac{\partial}{\partial t}\left[\operatorname{Tr}(X)\right] = \operatorname{Tr}\left(\frac{\partial X}{\partial t}\right). \tag{2.32}$$

For $R \in SO(3)$ we are able do a similar thing, by looking at each individual diagonal of each of the rotations from (2.2). This is done with the trace operator [30, 32], or by selecting specific information using cross-products [19] that are related to the generators defined in (2.3). The latter is chosen in this thesis, since it provides the ability to later adopt convergence results that are also used in [11].

2.1.2 Quaternions

Instead of attitude representation with rotation matrices, quaternions can be used. Compared to rotation matrices, quaternions have no singularities in the involved functions. Both representations, so rotation matrices and quaternions, are well suited to integrating the angular velocity of the body over time. However, quaternions have an ambiguity property, which means that equal positive and negative quaternions both describe the same rotation [24]. By designing the controller with rotation matrices and afterwards implementing quaternions, we can overcome this ambiguity. The ambiguity is namely handled automatically in this approach of designing with rotation matrices and afterwards implementing quaternions, since both the negative and positive version of the same quaternion represent the same rotation matrix. Additionally, we can check that both the positive and negative version of the same quaternion generate the same control action in order to validate the system. An important benefit of quaternions is the fact that computational effort and storage capacity needed is lower for quaternions than with rotation matrices [22]. On-board calculation power is something that is always scarce to some degree in portable devices.

Consider a quaternion $q \in \mathcal{H}$, in which $\mathcal{H} = \mathbb{R}^4$ is the quaternion space. Quaternions can be represented as

$$q = q_w + q_x i + q_y j + q_z k, \tag{2.33a}$$
$$q = (q_w, \tilde{q}), \tag{2.33b}$$
$$q = \begin{bmatrix} q_w & q_x & q_y & q_z \end{bmatrix}^\top, \tag{2.33c}$$

which shows that quaternions can be seen as vector, more specifically as a composition of an imaginary vector and a scalar. The imaginary vector elements fulfill

$$i^2 = j^2 = k^2 = ijk = -1, \tag{2.34}$$

from which other useful relations follow. We find

$$ijk = k^2, \tag{2.35a}$$
$$ij = k, \tag{2.35b}$$

and

$$i^2 jk = -jk = -i, \tag{2.35c}$$
$$-j^2 k = k = -ji. \tag{2.35d}$$

By following (2.35) for the other hyper-complex combinations, we find

$$ij = k = -ji, \qquad jk = i = -kj \quad \text{and} \quad ki = j = -ik. \tag{2.36}$$

A quaternion has a conjugate $\bar{q}$ and norm $||q||$, as

$$\bar{q} = q_w - q_x i - q_y j - q_z k = (q_w, -\tilde{q}), \tag{2.37}$$

$$||q|| = \sqrt{q\bar{q}} = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2} = \sqrt{q_w^2 + \tilde{q}\tilde{q}}, \tag{2.38}$$

Let us introduce the operator $\otimes$, denoting the quaternion product, as

$$p \otimes q = \begin{bmatrix} p_w q_w - p_x q_x - p_y q_y - p_z q_z \\ p_w q_x + p_x q_w + p_y q_z - p_z q_y \\ p_w q_y - p_x q_z + p_y q_w + p_z q_x \\ p_w q_z + p_x q_y - p_y q_x + p_z q_w \end{bmatrix}, \tag{2.39}$$

or by following [25], as

$$p \otimes q = (p_w q_w - \tilde{p}\tilde{q}, p_w \tilde{q} + q_w \tilde{p} + \tilde{p} \times \tilde{q}). \tag{2.40}$$

Quaternion products are non-commutative, just like rotations. By looking at (2.39), we define generators Q and $\bar{Q}$ [23] that handle this multiplication, as

$$p \otimes q = Q(p)q = \begin{bmatrix} q_w & -\tilde{q}^\top \\ \tilde{q} & q_w I_3 + S(\tilde{q}) \end{bmatrix} q = \begin{bmatrix} p_w & -p_x & -p_y & -p_z \\ p_x & p_w & -p_z & p_y \\ p_y & p_z & p_w & -p_x \\ p_z & -p_y & p_x & p_w \end{bmatrix} \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix}, \tag{2.41a}$$

$$= \bar{Q}(q)p = \begin{bmatrix} q_w & -\tilde{q}^\top \\ \tilde{q} & q_w I_3 - S(\tilde{q}) \end{bmatrix} p = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & q_z & -q_y \\ q_y & -q_z & q_w & q_x \\ q_z & q_y & -q_x & q_w \end{bmatrix} \begin{bmatrix} p_w \\ p_x \\ p_y \\ p_z \end{bmatrix}. \tag{2.41b}$$

Let us additionally introduce

$$E(q) = \left[-\tilde{q}, q_w I_3 + S(q)\right], \tag{2.42}$$

$$G(q) = \left[-\tilde{q}, q_w I_3 - S(q)\right], \tag{2.43}$$

with $S(q)$ according (2.4), such that we find

$$Q(q) = \begin{bmatrix} q & G(q)^\top \end{bmatrix}, \tag{2.44}$$

$$\bar{Q}(q) = \begin{bmatrix} q & E(q)^\top \end{bmatrix}. \tag{2.45}$$

Furthermore, if the norm of a quaternion $||q|| = 1$ we call this quaternion a unit quaternion, and if the scalar part $q_w = 0$ we call it a pure quaternion. We denote a vector as the pure quaternion

$$q_v = \begin{bmatrix} 0 \\ v \end{bmatrix} = v_x i + v_y j + v_z k. \tag{2.46}$$

A rotation of a body can be expressed with a clockwise rotation around an axis in 3D space [35]. Notice, that even when multiple rotations are combined, it is still possible to express this resulting rotation by a single rotation around a specific rotation axis; similar as how we are able to write multiple subsequent rotation matrices as one new rotation matrix. The rotation axis and rotation angle θ can be captured with a single quaternion [36]

$$\bar{q}_r = \cos\frac{\theta}{2} - (\bar{e})\sin\frac{\theta}{2}, \tag{2.47}$$

in which $\bar{e} \in \mathbb{R}^3$ is a unit vector that represents the rotation axis and $\theta \in \mathbb{R}$ denotes the amount of rotation. The rotation from one frame to another can be accomplished by the three quaternion product [23]. Consider $||q|| = 1$ and $v, v' \in \mathbb{R}^3$, then

$$\begin{aligned}\begin{bmatrix}0\\ v'\end{bmatrix} &= \bar{q} \otimes \begin{bmatrix}0\\ v\end{bmatrix} \otimes q, && (2.48a)\\ &= Q(\bar{q})\bar{Q}(q)\begin{bmatrix}0\\ v\end{bmatrix}, && (2.48b)\\ &= \begin{bmatrix}1 & 0^\top\\ 0 & E(q)G(q)^\top\end{bmatrix}\begin{bmatrix}0\\ v\end{bmatrix}, && (2.48c)\end{aligned}$$

in which we use the carefully chosen definitions (2.42) and (2.43), such that we can easily rotate vectors. Notice that (2.48) shows that quaternion rotations can be linked to rotation matrices, with mapping

$$(0, v') = (0, R_q(q)v) = Q(\bar{q})\bar{Q}(q)\begin{bmatrix}0\\ v\end{bmatrix}, \tag{2.49}$$

so we are able to explicitly find the rotation matrix, following from (2.42), (2.43) and (2.48c), equal to

$$\begin{aligned}R_q(q) &= (q_w^2 - \tilde{q}\tilde{q})I_3 + 2\tilde{q} \otimes \tilde{q} + 2q_w S(q), && (2.50)\\ &= E(q)G(q)^\top, && (2.51)\\ &= \frac{1}{||q||}\begin{bmatrix}||q||^2 - 2(q_y^2 + q_z^2) & 2(q_xq_y - q_zq_w) & 2(q_xq_z + q_yq_w)\\ 2(q_xq_y + q_zq_w) & ||q||^2 - 2(q_x^2 + q_z^2) & 2(q_yq_z - q_xq_w)\\ 2(q_xq_z - q_yq_w) & 2(q_yq_z + q_xq_w) & ||q||^2 - 2(q_x^2 + q_y^2)\end{bmatrix}. && (2.52)\end{aligned}$$

Furthermore, note that the lengths $||v||$ and $||v'||$ are theoretically the same, although numerical errors can be induced in simulations, changing the length of quaternions that are supposed to be unit quaternions. We are able ensure the pureness of the rotation by normalizing every step, which ensures that the rotation quaternion stays at unit length; often normalizing each set of steps is accurate enough. Apart from numerical errors, the linear operation does not change the length of v, since q is a unit quaternion of equal form as (2.50) with a unit vector denoting the axis $\bar{e}$; proving

$$\begin{aligned}||q|| &= \sqrt{\cos^2\left(\frac{\theta}{2}\right) + ||\bar{e}||\sin^2\left(\frac{\theta}{2}\right)}, && (2.53a)\\ &= \sqrt{\cos^2\left(\frac{\theta}{2}\right) + \sin^2\left(\frac{\theta}{2}\right)} = 1. && (2.53b)\end{aligned}$$

2.2 Stability results

Many efforts have already been made to prove the stability of various time-varying systems. In this section, we recall some of them in order to later on use these results directly.

Definition 2.2.1 (cf. [11]). *Consider the vector saturation function $\sigma(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ that is twice differentiable and monotone, with $s(\cdot)$ satisfying $s(0) = 0$ and $\lim_{x\to 0} s(x)/x = s'(0) > 0$. Furthermore, let $V_\sigma(e) = \int_0^{e^\top e} s(x)/x\mathrm{d}x$ where the latter function is bounded. Two possible candidates are*

$$\sigma(e) = e, \qquad \sigma(e) = \frac{e}{\sqrt{1 + e^\top e}}, \tag{2.54}$$

where the latter is bounded.

Definition 2.2.2 (cf. [11])**.** *A function σ_i for which $||\sigma_i(e)|| \leq M$ for all e is called a saturation function.*

For example, consider the bounded saturation function

$$\sigma_p(x) = \frac{x}{(1 + (x^\top x)^p)^{\frac{1}{2p}}}, \qquad \text{with} \quad p \in \mathbb{N}, \tag{2.55}$$

in which $x \in \mathbb{R}^3$ is a vector, and $p \in \mathbb{N}$ is a parameter to shape the function, as in [19]. Figure 2.1 shows the saturation function (2.55) for a range of scalar x, for parameter $p = 1$, $p = 2$ and $p = 3$. It can be seen that by increasing p, we increase the order of the saturation function, utilizing a larger portion of the available domain.

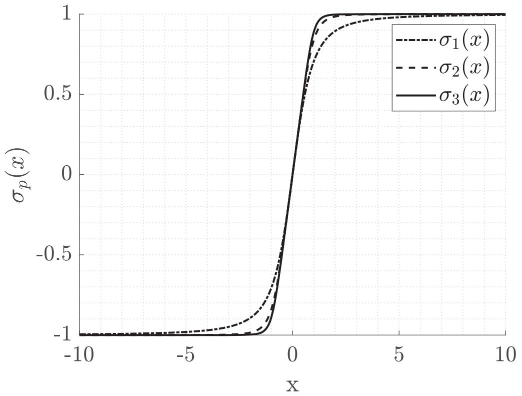


Figure 2.1: Saturation function (2.55) for $p = 1$, $p = 2$ and $p = 3$, with x scalar.

Theorem 2.2.1 (cf. [37, Theorem 1])**.** *Consider the dynamical system*

$$\dot{x} = f(t, x) \qquad\qquad x(t_0) = x_0, \tag{2.56}$$

with $f(t, 0) = 0$, $f \ : \ \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}^n$ locally bounded, continuous and locally uniformly continuous in t.
If there exist j differentiable functions $V_i \ : \ \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}^n$, bounded in t and continuous functions $Y_i \ : \ \mathbb{R}^n \to \mathbb{R}$ for $i \in \{1, 2, \ldots, j\}$ such that

- *V_1 is positive definite ,*
- *$\dot{V}_i(t, x) \leq Y_i(x)$, for all $i \in \{1, 2, \ldots, j\}$,*
- *$Y_i(x) = 0$ for $i \in \{1, 2, \ldots, k-1\}$ implies $Y_k(x) \leq 0$ for all $k \in \{1, 2, \ldots, j\}$,*
- *$Y_i(x) = 0$ for all $i \in \{1, 2, \ldots, j\}$ implies $x = 0$,*

then the origin $x = 0$ of (2.56) is uniformly globally asymptotically stable (UGAS).

Definition 2.2.3 (cf. [11])**.** *The origin of (2.56) is uniformly almost globally asymptotically stable (UaGAS) if it is UGAS, except for initial conditions in a set of measure zero.*

Theorem 2.2.2. *Consider the dynamics*

$$\dot{R}_{i,e} = R_{i,e}\bar{S}(\omega_{i,e}), \tag{2.57a}$$

$$J_i\dot{\omega}_{i,e} = -c_{i,\omega}\omega_{i,e} + \frac{1}{2}c_{i,R}\,\mathrm{Tr}(R_{i,e}\bar{G}), \tag{2.57b}$$

with $R_{i,e} \in SO(2) = \{\mathbb{R}_i^{2\times 2} \mid \det(R_i) = 1, R_i^\top R_i = 0\}$ *and* $\omega_{i,e} \in \mathbb{R}$. *If constants* $c_{i,\omega} > 0$ *and* $c_{i,R} > 0$*, then the equilibrium point* $(I, 0)$ *of* (2.57) *is uniformly locally exponentially stable (ULES) and uniformly almost globally stable (UaGAS). That is, let* $E_{i,c} = \{I, -I\}$. *Then* R_i *converges to* $E_{i,c}$ *and* ω_i *converges to zero. The equilibrium* $(-I, 0)$ *of* (2.57) *is unstable and the set of all initial conditions converging to the equilibrium* $(-I, 0)$ *forms a lower dimensional manifold.*

Proof. Define the candidate Lyapunov function

$$V(R_{i,e}, \omega_{i,e}) = \frac{1}{2}c_{i,R}\operatorname{Tr}(I - R_{i,e}) + \frac{1}{2}J_i\omega_{i,e}^2. \tag{2.58}$$

Differentiation of (2.58) along the error dynamics yields

$$\dot{V}(R_{i,e}, \omega_{i,e}) = \omega_{i,e}\left[\frac{1}{2}c_{i,R}\operatorname{Tr}(R_{i,e}\bar{G}) - c_{i,\omega}\omega_{i,e}\right] - \frac{1}{2}\operatorname{Tr}(R_{i,e}\bar{G}\omega_{i,e}), \tag{2.59a}$$

$$= -c_{i,\omega}\omega_{i,e}^2 \leq 0, \tag{2.59b}$$

in which we have used the property $\operatorname{Tr}(rA) = r\operatorname{Tr}(A)$ with r a scalar and A a matrix (according to (2.31) from Definition 2.1.1) and $\hat{G}\omega_{i,e} = S(\omega_{i,e})$ as in Theorem 2.1.1. Notice that in fact $\dot{V}(R_{i,e}, \omega_{i,e})$ from (2.59) is negative semi-definite. By having $R_i(\phi_i)$ according to (2.1), we can find that not only the intentional equilibrium $(\phi_{i,e}, \omega_{i,e}) = (0, 0)$ provides $\dot{V}_i(0) = 0$, but also the undesired equilibrium $(\phi_{i,e}, \omega_{i,e}) = (\pi, 0)$ provides $\dot{V}_i(\pi) = 0$. The angle $\phi_{i,e} = \pi$ corresponds with the case where the mobile robot is exactly backwards aligned with the desired heading; resulting in $R_{i,e} = -I$. Consider the set

$$Q_i = \{(R_{i,e}, \omega_{i,e}) \in SO(2) \times \mathbb{R} : V_i(R_{i,e}, \omega_{i,e}) \leq V_i(R_{i,e}(t_0), \omega_{i,e}(t_0))\}, \tag{2.60}$$

which is positively invariant. By direct implementation of LaSalle's invariance principle [38], similar to [19], we can conclude that the only solution that can identically stay in the set

$$E_i = \{(R_{i,e}, \omega_{i,e}) \in SO(2) \times \mathbb{R} | \dot{V}_i(R_{i,e}, \omega_{i,e}) = 0\} \subset Q_i, \tag{2.61}$$

is when $\omega_{i,e} = 0$ and $\frac{1}{2}c_{i,R}\operatorname{Tr}(R_{i,e}\bar{G}) = -\sin\phi_{i,e} = 0$. Since we consider rotation matrices in the special orthogonal group $SO(2)$, with $\det(R_{i,e}) = 1$, we know that this only holds for $R_{i,e} \in \mathcal{E}$, with

$$\mathcal{E} = \{I, -I\}, \tag{2.62}$$

of which $R_{i,e} = I$ denotes the preferred equilibrium for robot i. By using the Linearization method of infinitesimal variation, where infinitesimal variation in $R_{i,e}$ is expressed as $\delta R_{i,e} = \bar{S}(\phi_{i,e})R_{i,e}$ [19, 30], we investigate the stability around the equilibria. The linearized dynamics become

$$\begin{bmatrix} \dot{\phi}_{i,e} \\ \delta\dot{\omega}_{i,e} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -c_{i,R}N & -c_{i,\omega} \end{bmatrix} \begin{bmatrix} \phi_{i,e} \\ \delta\omega_{i,e} \end{bmatrix} \qquad \text{whith} \qquad N = \frac{1}{2}\operatorname{Tr}(R_{i,e}) = \cos\phi_{i,e} \tag{2.63}$$

which yields that the preferred equilibrium $R_{i,e} = I$ is stable and the other equilibrium $R_{i,e} = -I$ is unstable, driven by the fact that $-\cos 0 = -1$ causes the linearized dynamics to be Hurwitz and $-\cos\pi = 1$ causes the linearized dynamics to be unstable. Concluding, the equilibrium point $(R_{i,e}, \omega_{i,e}) = (I, 0)$ is uniformly locally exponentially stable (ULES). There is exactly one line corresponding with initial conditions $(R_{i,e}(t_0), \omega_{i,e}(t_0))$ that cause convergence to the unwanted equilibrium, which is a set of measure 0 [39], thus, the system (2.57) is uniform almost-global asymptotic stability (UaGAS). □

Theorem 2.2.2 applies for $SO(2) \times \mathbb{R}$, which can thus be used for the orientation of a unicycle robot. For the attitude control of a quadrotor UAV, however, we need to follow a similar approach, but for $SO(3) \times \mathbb{R}^3$. Let us adopt the following theorem.

Theorem 2.2.3 (adopted from [11], cf. [30,32])**.** *Consider the system*

$$\dot{R}_{i,e} = R_{i,e}S(\omega_{i,e}), \tag{2.64a}$$

$$J_i\dot{\omega}_{i,e} = -K_{i,\omega}\omega_{i,e} + K_{i,R}\sum_{s=1}^{3} k_{i,s}(e_s \times R_{i,e}^\top e_s), \tag{2.64b}$$

where $R_{i,e} \in SO(3) = \{\mathbb{R}_i^{3\times 3} \mid \det(R_i) = 1, R_i^\top R_i = 0\}$, $\omega_i \in \mathbb{R}^3$, $J_i = J_i^\top$ and $S(\cdot)$ according to (2.4). If $K_{i,\omega} = K_{i,\omega}^\top > 0$ and $K_{i,R} = K_{i,R}^\top > 0$ and $k_{i,s} > 0$ and distinct for $s = 1,2,3$ and arbitrary i (e.g., $0 < k_{i,1} < k_{i,2} < k_{i,3}$), then the equilibrium point $(I,0)$ of (2.64) is uniformly locally exponentially stable (ULES) and uniformly almost globally stable (UaGAS). That is, let $E_{i,c} = \{I, \mathrm{diag}(1,-1,-1), \mathrm{diag}(-1,1,-1), \mathrm{diag}(-1,-1,1)\}$. Then R_i converges to $E_{i,c}$ and ω_i converges to zero. The equilibria $(R_i,0)$ of (2.64) where $R \in E_{i,c} \setminus \{I\}$ are unstable and the set of all initial conditions converging to the equilibrium $(R_i,0)$ where $R \in E_{i,c} \setminus \{I\}$ forms a lower dimensional manifold.

Proof. Let us follow [11] and [19]. Choose the candidate Lyapunov function

$$V(R_{i,e},\omega_{i,e}) = \frac{1}{2}\omega_{i,e}^\top J_i\omega_{i,e} + K_{i,R}\underbrace{\sum_{s=1}^{3} k_{i,s}(1 - e_s^\top R_{i,e}e_s)}_{\psi(R_{i,e})}, \tag{2.65}$$

with $k_s > 0$ and distinct for $s = 1,2,3$, e.g., $k_1 > k_2 > k_3 > 0$, and $K_R = K_R^\top > 0$. For $R_{i,e} \neq I$ we have $\psi(R_{i,e}) > 0$ and $\psi(I) = 0$, so $V(R_{i,e},\omega_{i,e})$ is positive definite. Differentiating (2.65) along the dynamics (2.64), yields

$$\dot{V}(R_{i,e},\omega_{i,e}) = \frac{1}{2}(J_i\dot{\omega}_{i,e})^\top\omega_{i,e} + \frac{1}{2}\omega_{i,e}^\top(J_i\dot{\omega}_{i,e}) + K_{i,R}\sum_{s=1}^{3} k_s(e_s^\top R_{i,e}\omega_{i,e} \times e_s), \tag{2.66}$$

$$= \omega_{i,e}^\top\left[-K_{i,\omega}\omega_{i,e} + K_R\sum_{s=1}^{3} k_s\omega_{i,e}^\top(e_s \times R_{i,e}^\top e_s)\right] - K_{i,R}\sum_{s=1}^{3} k_s(e_s \times R_{i,e}^\top e_s), \tag{2.67}$$

$$= -\omega_{i,e}^\top K_{i,\omega}\omega_{i,e} \leq 0, \tag{2.68}$$

assuming that $K_{i,R} = K_{i,R}^\top > 0$, hence, we find $(I,0)$ as stable equilibrium. Notice that the closed-loop dynamics (2.64) are time-invariant. Let us consider the set

$$Q_i = \{(R_{i,e},\omega_{i,e}) \in SO(3)\times\mathbb{R}^3 : V(R_{i,e},\omega_{i,e}) \leq V(R_{i,e}(t_0),\omega_{i,e}(t_0))\}, \tag{2.69}$$

which is positively invariant. Now, let us apply LaSalle as in [19], according to the direct invariant set theorem [38, Theorem 4.4]. The only solutions that can identically stay in the set

$$E_i = \{(R_{i,e},\omega_{i,e}) \in SO(3)\times\mathbb{R}^3 \mid \dot{V}(R_{i,e},\omega_{i,e}) = 0\} \subset Q_i, \tag{2.70}$$

are the solutions of $\sum_{s=1}^{3} k_s\omega_{i,e}^\top(e_s \times R_{i,e}^\top e_s) = 0$ and $\omega_{i,e} = 0$. The latter only holds for $e_s = R_{i,e}^\top e_s = 0$ or $e_s = -R_{i,e}^\top e_s = 0$, which means that we find the following set of equilibria

$$\mathcal{E}_i = \{I, \mathrm{diag}([1,-1,-1]), \mathrm{diag}([-1,1,-1]), \mathrm{diag}([-1,-1,1])\} \subset SO(3). \tag{2.71}$$

From (2.71) we conclude that $R_{i,e}$ converges to one of these equilibria, of which I is the desired equilibrium that corresponds to alignment of the frames; the other three equilibria correspond to partial alignment of the frames with a rotation of $\pm\pi$ radians about the aligned axis. We can find that only the desired equilibrium I is stable and that undesired equilibria $\mathcal{E}\setminus\{I\}$ are unstable, by linearising the system around these equilibria. Linearisation of the dynamics, as in [40], with infinitesimal variation expressed as $\delta R_{i,e} = S(\theta_i)R_{i,e}$, leads to

$$\begin{bmatrix} \dot{\theta}_i \\ \delta\dot{\omega}_{i,e} \end{bmatrix} = \underbrace{\begin{bmatrix} S(\omega_{i,e}) & I \\ K_{i,R}N_i & -K_{i,\omega} \end{bmatrix}}_{\tilde{A}_i} \begin{bmatrix} \theta_i \\ \delta\omega_{i,e} \end{bmatrix}, \quad \text{with} \quad N_i = \sum_{s=1}^{3} k_s \left[(e_s^\top R_e e_s)I - e_s e_s^\top R_{i,e}^\top \right]. \tag{2.72}$$

Only for the desired equilibrium I, we have N_i diagonal and positive definite, leading to $\tilde{A}_i$ being Hurwitz. The other equilibria are unstable (hyperbolic). Hence, the desired equilibrium I, and with that the attitude tracking dynamics, are uniform local exponentially stable (ULES). Define G as the set of attraction to the desired equilibrium. Since the unstable equilibria are hyperbolic, we can conclude that for each unstable equilibrium point there exists a stable and unstable manifold. Then, according to [19, 30, 32] we find that the dimension of the union $\mathcal{M} = (SO(3) \times \mathbb{R}^3) \setminus G$ of unstable manifolds is less than the dimension of SO(3) and therefore has zero measure. Hence, by looking at the entire space of solutions we can conclude that the system is ULES and UaGAS. □

Theorem 2.2.4 (Adopted from [11], cf. [41])**.** *Consider the cascaded system $\dot{x} = f(t, x)$, with $f(t, 0) = 0$ that can be written as*

$$\dot{x}_1 = f_1(t, x_1) + g(t, x_1, x_2)x_2, \tag{2.73a}$$
$$\dot{x}_2 = f_2(t, x_2), \tag{2.73b}$$

where $x_1 \in \mathbb{R}^n$, $x_2 \in \mathbb{R}^m$, $f_1(t, x_1)$ is continuous differentiable in (t, x_1), and $f_2(t, x_2)$, $g(t, x_1, x_2)$ are continuous in their argument and locally Lipschitz in x_2 and (x_1, x_2), respectively. This system is a cascade of the system

$$\dot{x}_1 = f_1(t, x_1), \tag{2.74}$$

and (2.73b)*. If the origins of the systems* (2.74) *and* (2.73b) *are uniform globally asymptotically stable (UGAS) and the solutions of* (2.73) *remain bounded, then the origin of the system* (2.73) *is UGAS. If additionally the systems* (2.74) *and* (2.73b) *are uniform locally exponentially stable (ULES), then* (2.73) *is ULES.*

Theorem 2.2.5 (cf. [42])**.** *If the origin of* (2.74) *is uniform globally exponentially stable (UGES), the origin of* (2.73b) *is ULES and UGAS and*

$$||g(t, x_1, x_2)|| \leq k_1(||x_2||) + k_2(||x_2||)||x_1||, \tag{2.75}$$

then the origin of the system (2.73) *is ULES and UGAS.*

Chapter 3

Formation tracking with mobile robots

In order to first focus on a similar but slightly simpler problem than the tracking control and coupling of quadrotor UAVs, we first consider a similar problem for the mobile robot. This approach is preferred since the positioning of a mobile robot can be considered as a 2D instead of a 3D problem, which also eliminates two of the three rotation angles involved. Besides, recent work on the unicycle robot regarding coupling between agents [15, 43] provides a good starting point to elaborate on and inspired us to at least try to obtain similar results for the quadrotor UAV. Furthermore, unicycle robots are just like quadrotor UAVs underactuated devices, causing that they both rely on their attitude freedom to help obtain specific spatial behavior.

Similar to the main goal that is coupling multiple quadrotor UAVs, we now first want to couple multiple mobile robots. In order to do so, we start by developing a model that is very similar to the used model of a quadrotor UAV from [19], by modeling the unicycle robot according to the Newton-Euler framework. Then, we focus on the tracking problem of a single agent, which provides a good reference point that can be compared with recent results for the quadrotor UAV [19]. After that, we provide two approaches for implementing coupling in the system. First, we investigate altering the control law by directly implementing coupling terms. Then, we alter the generalized coordinates by adding coupling errors in the error definitions.

3.1 Mobile robot modeling and control in the Newton-Euler framework

In this section, we first introduce the model of a mobile robot according to the Newton-Euler framework, in order to stay as close to [19] as possible. After that, we introduce reference dynamics to track and state a tracking problem, so that we eventually can solve this tracking problem in the subsequent sections.

3.1.1 Dynamical modeling of a unicycle robot in the Newton-Euler framework

Let $\rho_i = \begin{bmatrix} x_i, \ y_i \end{bmatrix}^\top \in \mathbb{R}^2$ denote the position of the center of mass of a unicycle robot i relative to the inertial frame $\mathcal{I}$. Consider fixed to the mass at ρ_i a body-fixed coordinate frame $\mathcal{B}_i$ with a relative rotation of $R_i \in SO(2)$ with respect to the inertial frame $\mathcal{I}$. The rotation R_i is defined by a single angle ϕ_i around the axis perpendicular to the plane of $\mathcal{I}$, thus, we can write $R_i := R_z(\phi_i)$

with

$$R_z(\phi) = \begin{bmatrix} \cos\phi_i & -\sin\phi_i \\ \sin\phi_i & \cos\phi_i \end{bmatrix} \in SO(2), \tag{3.1}$$

with properties that are introduced in Section 2.1. Let $v_i \in \mathbb{R}$ and $\omega_i \in \mathbb{R}$, respectively, denote the body-fixed forward velocity of the mass and the angular velocity of the body-fixed frame $\mathcal{B}_i$. A schematic representation of the introduced states is included in Figure 3.1. We assume that the

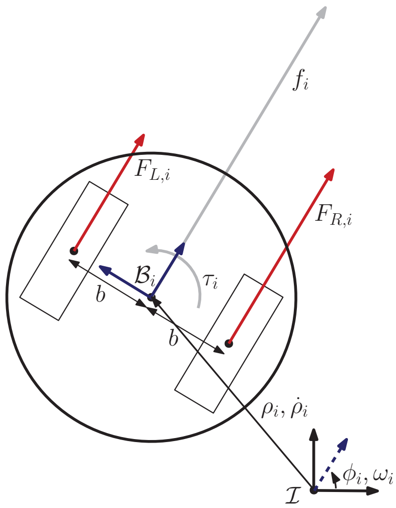


Figure 3.1: Schematic representation of the mobile robot. The body fixed reference frame $\mathcal{B}_i$ and inertial reference frame $\mathcal{I}$ are indicated.

sideways velocity component of the mass in the body-fixed frame is nonexistent; this follows from direct implementation of the commonly used nonholonomic no side-slip constraint (e.g., [15, 28]). This constraint on velocity level can be considered by the kinematic relation

$$\begin{bmatrix} v_i \\ u_i \end{bmatrix} = R_i^\top \dot{\rho}_{i,e} = R_i^\top \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{3.2}$$

in which v_i denotes the longitudinal velocity and u_i denotes the lateral velocity component. By assuming no side-slip, we directly assume $u_i = 0$, which provides

$$\begin{bmatrix} v_i \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\phi_i & \sin\phi_i \\ -\sin\phi_i & \cos\phi_i \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{3.3}$$

which yields the constraint on velocity level

$$\dot{y}_i \cos\phi_i - \dot{x}_i \sin\phi_i = 0. \tag{3.4}$$

The remaining inverse relation from (3.3) yields the kinematic relation between forward velocity in the body-fixed frame $\mathcal{B}_i$ and the velocity in the inertial frame $\mathcal{I}$, as

$$\dot{\rho}_i = \dot{x}_i \cos\phi_i + \dot{y}_i \sin\phi_i. \tag{3.5}$$

In order to obtain a model that is very similar to [11, 19], hopefully allowing us to later on follow a similar approach for the quadrotor UAV, we generalize this kinematic relation to

$$\dot{\rho}_i = R_i v_i e_1, \tag{3.6}$$

with the unit vector $e_1 = [1, 0]^\top$ indicating that v_i coincides with the forward facing unit vector of the body-fixed frame $\mathcal{B}_i$. We want to represent the attitude very similar to [19], also by applying the Newton-Euler method. Notice that generally attitude representation within the Newton-Euler framework leads to

$$J_i \dot{\hat{\omega}}_i = (J_i \hat{\omega}_i) \times \hat{\omega}_i + \tau_i, \tag{3.7}$$

where $\hat{\omega}_i$ a vector that consists of angular velocities around three axis of rotation; this is exactly what is presented in [19] apart from notation differences. However, in this case we only have an angular velocity around the axis perpendicular to the plane, which we can see as the z direction. This single rotational freedom leads to $\hat{\omega}_{i,e} = [0, 0, \omega_{i,e}]$, which also provides $(J_i \hat{\omega}_i) \times \hat{\omega}_i = 0$, so that the expression now equals

$$J_i \dot{\omega}_i = \tau_i. \tag{3.8}$$

We can complete the model, by using the derivative of a rotation matrix $\dot{R}_i = \bar{S}(\omega_i) R_i$ with $R_i \in SO(2)$, as introduced in Theorem 2.1.1, and using Newton's Second law to relate the acceleration $\dot{v}_i$ to the forward force. The dynamical model of a unicycle i, based on the Newton-Euler framework, then equals

$$\dot{\rho}_i = R_i v_i e_1, \tag{3.9a}$$

$$\dot{v}_i = \frac{f_i}{m_i}, \tag{3.9b}$$

$$\dot{R}_i = R_i \bar{S}(\omega_i), \tag{3.9c}$$

$$J_i \dot{\omega}_i = \tau_i, \tag{3.9d}$$

with inertia $J_i > 0$ with respect to the body-fixed frame $\mathcal{B}_i$, mass m_i, forward force magnitude applied by the unicycle wheel $f_i \in \mathbb{R}$ and the torque around the axis of rotation $\tau_i \in \mathbb{R}$ for robot i. Notice that we are able to use the dynamical model of a unicycle for other types of robots, since we can apply an input transformation that relates the specific robot lay-out to a forward facing force f_i and torque τ_i for any type of robot. For example, the forward force and torque of a differential drive robot are composed of two separate forces resulting from wheel contact with the ground for each of the two wheels, as

$$f_i = F_{L,i} + F_{R,i}, \tag{3.10a}$$

$$\tau_i = F_{R,i} b - F_{L,i} b, \tag{3.10b}$$

with the robot lay-out and forces as indicated in Figure 3.1. The inverse relation yields

$$\begin{bmatrix} F_{L,i} \\ F_{R,i} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & -b \\ 1 & b \end{bmatrix} \begin{bmatrix} f_i \\ \tau_i \end{bmatrix}. \tag{3.11}$$

3.1.2 The tracking problem

In addition to the dynamics of the mobile robot (3.9), consider the dynamics of a virtual reference robot

$$\dot{\rho}_{i,r} = R_{i,r} v_{i,r} e_1, \tag{3.12a}$$

$$\dot{v}_{i,r} = \frac{f_{i,r}}{m_i}, \tag{3.12b}$$

$$\dot{R}_{i,r} = R_{i,r} \bar{S}(\omega_{i,r}), \tag{3.12c}$$

$$J_i \dot{\omega}_{i,r} = \tau_{i,r}, \tag{3.12d}$$

where $0 < f_{i,r}^{min} \leq f_r(t)$. Assume that a feasible reference trajectory is given, i.e., a reference trajectory $(\rho_{i,r}, v_{i,r}, R_{i,r}, \omega_{i,r})$ that satisfies (3.12). Given a reference position $\rho_{i,r}(t) = \begin{bmatrix} x_{i,r}(t) & y_{i,r}(t) \end{bmatrix}^\top$ in time, we can conclude feasibility under certain conditions that follow from the dynamics (3.12). Based on (3.12) and (3.1), we can find

$$\frac{\dot{y}_{i,r}}{\dot{x}_{i,r}} = \frac{v_{i,r} \sin \phi_{i,r}}{v_{i,r} \cos \phi_{i,r}}, \tag{3.13a}$$

$$\phi_{i,r} = \operatorname{atan2}\left(\frac{\dot{y}_{i,r}}{v_{i,r}}, \ \frac{\dot{x}_{i,r}}{v_{i,r}}\right), \tag{3.13b}$$

$$\dot{\phi}_{i,r} = \omega_{i,r} = \frac{\ddot{y}_{i,r}\dot{x}_{i,r} + \ddot{x}_{i,r}\dot{y}_{i,r}}{\dot{x}_{i,r}^2 + \dot{y}_{i,r}^2}, \tag{3.13c}$$

which first of all means that we can find a heading angle $\phi_{i,r}$ given the position over time, but only if we can also find $v_{i,r}$ and assume $\rho_{i,r}(t) = [x_{i,r}(t), \ y_{i,r}(t)]^\top$ is twice differentiable. The velocity $v_{i,r}$ is obtained as the size of the velocity state vector

$$v_{i,r} = \sqrt{\dot{x}_{i,r}^2 + \dot{y}_{i,r}^2}, \tag{3.14a}$$

$$\omega_{i,r} = \frac{\ddot{y}_{i,r}\dot{x}_{i,r} + \ddot{x}_{i,r}\dot{y}_{i,r}}{\dot{x}_{i,r}^2 + \dot{y}_{i,r}^2}. \tag{3.14b}$$

The existence of the relations (3.14a) and (3.14b) is well-known in literature regarding robotic systems [44,45] and is a consequence of differential flatness of the dynamics [15]. In order to obtain the forward and angular acceleration of the reference vehicle, differentiate (3.14a) and (3.14b), respectively, as

$$\dot{v}_{i,r} = \frac{\dot{x}_{i,r}\ddot{x}_{i,r} + \dot{y}_{i,r}\ddot{y}_{i,r}}{\sqrt{\dot{x}_{i,r}^2 + \dot{y}_{i,r}^2}}, \tag{3.15a}$$

$$\dot{\omega}_{i,r} = \frac{\dddot{y}_{i,r}\dot{x}_{i,r} + 2\ddot{y}_{i,r}\ddot{x}_{i,r} + \dddot{x}_{i,r}\dot{y}_{i,r}}{\dot{x}_{i,r}^2 + \dot{y}_{i,r}^2} - \frac{2(\ddot{y}_{i,r}\dot{x}_{i,r} + \ddot{x}_{i,r}\dot{y}_{i,r})(\dot{x}_{i,r}\ddot{x}_{i,r} + \dot{y}_{i,r}\ddot{y}_{i,r})}{(\dot{x}_{i,r}^2 + \dot{y}_{i,r}^2)^2}. \tag{3.15b}$$

Concluding, we find a trajectory that fulfills the dynamics (3.12) and thereby feasibility, only when the prescribed position over time $\rho_{i,r}(t) = [x_{i,r}(t), \ y_{i,r}(t)]^\top$ is three times differentiable. Another limitation on feasibility of course is the maximal force f_i and torque τ_i that the actuator of the vehicle is able to deliver, which is for now assumed to be not limiting. Later on when we consider a quadrotor UAV, we do explicitly overcome this problem by means of saturation of the inputs. Define the error coordinates

$$\rho_{i,e} = R_r^\top \left[\rho_{i,r} - \rho_i\right], \tag{3.16a}$$

$$v_{i,e} = v_{i,r}e_1 - R_{i,r}^\top R_i v e_1, \tag{3.16b}$$

$$\bar{R}_{i,e} = R_{i,r}^\top R_i, \tag{3.16c}$$

$$\bar{\omega}_{i,e} = \omega_i - \omega_{i,r}, \tag{3.16d}$$

in which we have used the fact that all rotations are defined around the same axis of rotation that is perpendicular to the plane spanned by the inertial frame $\mathcal{I}$; this is by definition the case with planar dynamics. All error coordinates are expressed in the body-fixed frame of the reference $\mathcal{B}_i$. Now, in order to match the problem from [11] but with a unicycle robot, let us state the tracking problem for a single vehicle.

Problem 3.1.1 (cf. [11])**.** *Given a feasible reference trajectory $(\rho_{i,r}, v_{i,r}, R_{i,r}, \omega_{i,r})$ for robot i, find control laws*

$$f_i = f_i(\rho_i, v_i, R_i, \omega_i, \rho_{i,r}, v_{i,r}, R_{i,r}, \omega_{i,r}), \tag{3.17a}$$

$$\tau_i = \tau_i(\rho_i, v_i, R_i, \omega_i, \rho_{i,r}, v_{i,r}, R_{i,r}, \omega_{i,r}), \tag{3.17b}$$

such that the resulting closed-loop (3.9), (3.12) *and* (3.17) *yields*

$$\lim_{t\to\infty}\left(||\rho_{i,e}|| + ||v_{i,e}|| + ||\log \bar{R}_{i,e}|| + |\bar{\omega}_{i,e}|\right) = 0. \tag{3.18}$$

3.1.3 A virtual structure for reference tracking

Instead of directly assuming the trajectory for a single agent given by the position $\rho_{i,r}(t)$ over time that is three times differentiable, as explained in Section 3.1.2, we assume that the reference trajectory is defined as a result of a defined reference formation shape and pose. In order to do so, we explicitly need to know this location and derive the necessary relations that ensure the feasibility of each of the resulting reference trajectories in the formation. Again, this feasibility is important in order to ensure that the virtual reference trajectories in fact act as virtual robots that fulfill the dynamics, so that the actual robots at least are physically able to track the references. A single resulting reference from this virtual structure approach is then treated similar as the reference dynamics from Section 3.1.2, meaning that when we achieve to answer the tracking problem from Problem 3.1.1, we achieve to track a specific reference trajectory from a virtual reference quadrotor in the formation. If we solve this problem for multiple agents in the same formation simultaneously, we can actually obtain a formation of unicycle robots. Notice that this is by no means coupling or synchronization of multiple agents, since there is no connectivity between the agents, the agents are simply provided with their own tasks that happen to be structurally connected.

In order to systematically obtain a set of n feasible reference trajectories, we have chosen to follow a similar procedure as in [13, 15], defining the graph structure relative to a virtual formation centered coordinate frame, which also provides a mutually known frame in order to later on relate the individual generalized coordinates for mutual coupling.

Introduce the formation centered frame $\mathcal{F}$ positioned at a fixed but free to choose virtual center (VC) of the formation. Let $\rho_f(t) = [x_f(t), y_f(t)]^\top \in \mathbb{R}^2$ denote the planar position of the VC and let $R_f(t) \in SO(2)$ denote the transformation from $\mathcal{F}$ to $\mathcal{I}$ over time, with

$$R_f(t) = R_z(\phi_f(t)) = \begin{bmatrix} \cos\phi_f(t) & -\sin\phi_f(t) \\ \sin\phi_f(t) & \cos\phi_f(t) \end{bmatrix}. \tag{3.19}$$

Let the position of an agent i over time be given relative to the VC, expressed in $\mathcal{F}$, by $p_i(t) = [p_{i,x}(t), p_{i,y}(t)]^\top \in \mathbb{R}^2$. Notice that $p_i(t)$ with $i = 1, \ldots, n$ is thus the set of shape vectors defining the formation shape (i.e., position vectors relative to the VC). Furthermore, note that although we introduce the shape vectors as explicitly time varying, it is also possible to have a constant formation shape in time.

Subsequently, similar to [15], consider the reference trajectory of vehicle i, as the composition

$$\rho_{i,r} = \rho_f + R_f p_i, \tag{3.20}$$

which is thus a composition between the trajectory of the formation relative to the inertial frame and the position of agent i relative to the virtual formation center, combined and expressed relative to the inertial frame. Let us relate (3.20) to the exact form from [15], by substituting the parameters in (3.20), as

$$\begin{bmatrix} x_{i,r} \\ y_{i,r} \end{bmatrix} = \begin{bmatrix} x_f \\ y_f \end{bmatrix} + \begin{bmatrix} \cos\phi_f & -\sin\phi_f \\ \sin\phi_f & \cos\phi_f \end{bmatrix} \begin{bmatrix} p_{i,x} \\ p_{i,y} \end{bmatrix}, \tag{3.21a}$$

which leads to

$$\begin{cases} x_{i,r} = x_f + p_{i,x}\cos\phi_f - p_{i,y}\sin\phi_f, \\ y_{i,r} = y_f + p_{i,x}\sin\phi_f + p_{i,y}\cos\phi_f, \end{cases} \tag{3.21b}$$

in which (x_f, y_f) describes the movement of a formation center, angle ϕ_f denotes a formation heading and relative position $(p_{i,x},\ p_{i,y})$ gives the possibly time varying formation shape. In order to have (3.20) provide a feasible reference trajectory, we at least need to have existence of (3.14a) and (3.14b) and their derivatives (3.15a) and (3.15b). These relations are defined if $\rho_{i,r}$ is three times differentiable. We differentiate the relation (3.20) that we use to obtain $\rho_{i,r}$ thee times to find the resulting necessary conditions for feasibility of the tracking problem

$$\dot{\rho}_{i,r} = \dot{\rho}_f + \dot{R}_f p_i + R_f \dot{p}_i, \tag{3.22a}$$

$$\ddot{\rho}_{i,r} = \ddot{\rho}_f + \ddot{R}_f p_i + 2\dot{R}_f \dot{p}_i + R_f \ddot{p}_i, \tag{3.22b}$$

$$\dddot{\rho}_{i,r} = \dddot{\rho}_f + \dddot{R}_f p_i + 3\ddot{R}_f \dot{p}_i + 3\dot{R}_f \ddot{p}_i + R_f \dddot{p}_i, \tag{3.22c}$$

Notice that $\dddot{R}_f$ can be related to the orientation angle over time $\ddot{\omega}_f = \dddot{\phi}_f$ by following Theorem 2.1.1. Concluding, in order to provide a feasible reference trajectory, the reference trajectory parameters ρ_f, p_i, and ϕ_f have to be three times differentiable.

3.2 Individual reference tracking with a unicycle robot

Since we have now provided a virtual structure that allows agents to track, we try to develop controllers f_i and τ_i that solve the tracking problem from Problem 3.1.1. In this section, we first look at a single unicycle identified by i, that tracks its own feasible reference trajectory. When we solve the tracking problem for a single unicycle, this means that we also track the set of reference trajectories from the virtual structure with a set of agents in order to track a virtual reference structure that defines a formation.

Although we have already introduced error coordinates in Problem 3.1.1, we are free to choose different error coordinates to solve this problem, as long as this implies that eventually a solution to Problem 3.1.1 is provided. Most intuitive would be to define the error coordinates relative to the inertial frame, but this makes the tracking errors depend on the inertial frame. If we rotate the inertial frame by 90° for instance, the components of the position tracking errors change and the behavior can change. Often, the error coordinates are expressed relative to the reference frame, as also used in [11], but they can also be expressed relative to the body-fixed frame [13, 15]. The generalized coordinates are then independent from the inertial frame choice and can be seen as components in longitudinal and lateral direction to the reference trajectory or actual trajectory, respectively. However, when eventually introducing coupling between the agents, this expression relative to the reference frame $\mathcal{R}_i$ for agent i makes the generalized coordinates from two agents not directly comparable, since they are not expressed relative to the same coordinate frame. Therefore, we try to propose an alternative to the approach from [15], where error coordinates are expressed relative to the body-fixed frame of each robot and also used for coupling. We also divert slightly from [11] where a single quadrotor UAV is considered with generalized coordinates relative to the reference frame.
In this section, we separate the design of the tracking controller into two different parts. First, we consider the subsystem for position tracking, under assumption that we can use accelerations of the unicycle robot as virtual input. Consecutively, we consider the problem of realizing this virtual input by means of the actual inputs.

3.2.1 Position reference tracking with a mobile robot

In the previous section a frame $\mathcal{F}$ is introduced fixed to the virtual center of the reference formation. The frame $\mathcal{F}$ in addition to the frame $\mathcal{I}$ is the second mutually known reference frame in the system. The existence of this frame $\mathcal{F}$ perfectly allows to express all of the involved position tracking errors relative to the same and mutually known reference frame. A benefit of the frame $\mathcal{F}$ with respect to the frame $\mathcal{I}$ is that it does not cause different behavior in the different directions of the inertial frame, and therefore does not change when we choose a different inertial frame. In order to solve Problem 3.1.1 and have generalized error coordinates for all quadrotor UAVs expressed in the same mutually known frame $\mathcal{F}$, define the reference tracking error coordinates

$$\rho_{i,e} = R_f^\top \left[\rho_{i,r} - \rho_i\right], \tag{3.23a}$$
$$v_{i,e} = R_f^\top R_{i,r} v_{i,r} e_1 - R_f^\top R_i v_i e_1. \tag{3.23b}$$

Notice that we are free to choose the formation frame $\mathcal{F}$ as it represents a virtual center. Therefore, in a purely single agent system, we can choose $\mathcal{F}$ equal to one of the other involved frames to obtain more classical error definitions [15, 46]. Differentiating (3.23) along the dynamics (3.9) and (3.12), yields

$$\dot{\rho}_{i,e} = -\bar{S}(\omega_f)\rho_{i,e} + v_{i,e}, \tag{3.24a}$$
$$\dot{v}_{i,e} = -\bar{S}(\omega_f)v_{i,e} + \underbrace{R_f^\top R_{i,r}\bar{S}(\omega_{i,r})v_{i,r}e_1 - R_f^\top R_i \bar{S}(\omega_i)v_i e_1 + R_f^\top R_{i,r}\frac{f_{i,r}}{m_i}e_1 - R_f^\top R_i \frac{f_i}{m_i}e_1}_{u_i}, \tag{3.24b}$$

with skew symmetrix matrix $\bar{S}(\omega_{i,r})$ as defined in Theorem 2.1.1 and we assume u_i to be a virtual input that is defined for control purposes and achieved by controlling the thrust force magnitude f_i and the orientation R_i. Notice that we have full control over u_i since we have full control over the term $R_f^\top R_i \frac{f_i}{m_i} e_1$. We can achieve the input size of $u_i \in \mathbb{R}^2$ by $f_i \in \mathbb{R}$ and the orientation by the direction $R_i \in SO(2)$, meaning that we can obtain any input vector $u_i \in \mathbb{R}^2$ [11]. In order to provide the first part of a solution to Problem 3.1.1, we propose the following virtual input.

Proposition 3.2.1. *Consider the position reference tracking dynamics* (3.24) *with the error coordinates given by* (3.23)*. Choosing the control law*

$$u_i = -k_{i,\rho}\rho_{i,e} - K_{i,v}v_{i,e}, \tag{3.25}$$

yields the closed-loop subsystem for position tracking

$$\dot{\rho}_{i,e} = -\bar{S}(\omega_f)\rho_{i,e} + v_{i,e}, \tag{3.26a}$$
$$\dot{v}_{i,e} = -\bar{S}(\omega_f)v_{i,e} - k_{i,\rho}\rho_{i,e} - K_{i,v}v_{i,e}, \tag{3.26b}$$

in which $k_{i,\rho} > 0$ and $K_{i,v} = K_{i,v}^\top > 0$ are control parameters that act on position and velocity level, respectively. The closed-loop system (3.26) *is uniformly globally asymptotically stable (UGAS).*

Proof. In order to prove the stability of the closed-loop system (3.26), consider the candidate Lyapunov function

$$V_1(\rho_{i,e}, v_{i,e}) = \frac{1}{2}\rho_{i,e}^\top k_{i,\rho}\rho_{i,e} + \frac{1}{2}v_{i,e}^\top v_{i,e} > 0, \tag{3.27}$$

again with scalar $k_{i,\rho} > 0$. Differentiating (3.27) along the error dynamics (3.24) provides

$$\dot{V}_1(\rho_{i,e}, v_{i,e}) = \rho_{i,e}^\top k_{i,\rho}\dot{\rho}_{i,e} + v_{i,e}^\top \dot{v}_{i,e}, \tag{3.28a}$$
$$= \rho_{i,e}^\top k_{i,\rho}\left[-\bar{S}(\omega_{i,r})\rho_{i,e} + v_{i,e}\right] + v_{i,e}^\top\left[-\bar{S}(\omega_{i,r})v_{i,e} + u_i\right], \tag{3.28b}$$
$$= v_{i,e}^\top k_{i,\rho}\rho_{i,e} + v_{i,e}^\top u_i, \tag{3.28c}$$

in which we have used the property of skew symmetric matrices, that $b^\top \bar{S}(a) b = 0$. Choosing the virtual input (3.25) then provides

$$\dot{V}_1(\rho_{i,e}, v_{i,e}) = -v_{i,e}^\top K_{i,v} v_{i,e} := Y_1(v_{i,e}) \leq 0. \tag{3.29}$$

Subsequently, since (3.29) does not yield any information on $\rho_{i,e}$ yet, let us choose the additional function

$$V_2(\rho_{i,e}, v_{i,e}) = \rho_{i,e}^\top v_{i,e}, \tag{3.30}$$

with time derivative along the dynamics

$$\dot{V}_2(\rho_{i,e}, v_{i,e}) = v_{i,e}^\top v_{i,e} - \rho_{i,e}^\top k_{i,\rho}\rho_{i,e} - \rho_{i,e}^\top K_{i,v} v_{i,e} := Y_2(\rho_{i,e}, v_{i,e}). \tag{3.31}$$

We can now use Matrosov's Theorem, included as Theorem 2.2.1, in order to conclude asymptotic stability, similar to [11]. We have $V_1(\rho_{i,e}, v_{i,e}) \geq 0$ provided that $k_{i,\rho} > 0$ and $K_{i,v} = K_{i,v}^\top > 0$, satisfying the first condition. Furthermore, by choosing $\dot{V}_1(\rho_{i,e}, v_{i,e}) = Y_1(v_{i,e})$ and $\dot{V}_2(\rho_{i,e}, v_{i,e}) = Y_2(v_{i,e})$ follows that the second condition is automatically satisfied. Then, we can find $Y_1(v_{i,e}) = 0$ for $v_{i,e} = 0$, which implies $Y_2(\rho_{i,e}, 0) = -\rho_{i,e}^\top k_{i,\rho}\rho_{i,e} \leq 0$ provided $k_{i,\rho} > 0$, which means that the third assumption holds. Lastly, the equalities $Y_1(v_{i,e}) = 0$ and $Y_2(\rho_{i,e}, v_{i,e}) = 0$ only hold at the absolute origin. Concluding, since all conditions from Theorem 2.2.1 are satisfied, the closed-loop system (3.26) is UGAS. □

3.2.2 Orientation control on SO(2)

In the previous part, we showed that having $R_f^\top R_{i,r}\bar{S}(\omega_{i,r})v_{i,r}e_1 - R_f^\top R_i\bar{S}(\omega_i)v_i e_1 + R_f^\top R_{i,r}\frac{f_{i,r}}{m_i}e_1 - R_f^\top R_i\frac{f_i}{m_i}e_1$ equal to the desired virtual input u_i from (3.25) provides that the position tracking subsystem is UGAS. Next, we want to use the actual inputs f_i and τ_i to achieve this desired virtual input u_i.

Let us start by using (3.24b), to find

$$R_i\left[\bar{S}(\omega_i)m_i v_i e_1 + f_i e_1\right] = R_{i,r}\left[\bar{S}(\omega_{i,r})m_i v_{i,r}e_1 + f_{i,r}e_1 - R_{i,r}^\top R_f m_i u_i\right], \tag{3.32a}$$

$$R_i\left[f_i e_1 + m_i\omega_i v_i e_2\right] = R_{i,r}\left[f_{i,r}e_1 + m_i\omega_{i,r}v_{i,r}e_2 - R_{i,r}^\top R_f m_i u_i\right]. \tag{3.32b}$$

Define

$$F_i = \begin{bmatrix} F_{i,x} \\ F_{i,y} \end{bmatrix} = f_{i,r}e_1 + m_i\omega_{i,r}v_{i,r}e_2 - R_{i,r}^\top R_f m_i u_i, \tag{3.33}$$

so that we can write

$$R_i\begin{bmatrix} f_i \\ m_i\omega_i v_i \end{bmatrix} = R_{i,r}\begin{bmatrix} F_{i,x} \\ F_{i,y} \end{bmatrix}. \tag{3.34}$$

Because of the unit length of rotation matrices, as explained in Section 2.1, we can find the force magnitude input f_i, as

$$f_i^2 = F_{i,x}^2 + F_{i,y}^2 - (m_i\omega_i v_i)^2, \tag{3.35}$$

$$f_i = \pm\sqrt{F_{i,x}^2 + F_{i,y}^2 - (m_i\omega_i v_i)^2}, \tag{3.36}$$

for $(m_i\omega_i v_i)^2 \leq F_{i,x}^2 + F_{i,y}^2$. Notice that this means that we are unable to obtain a solution for the case where $(m_i\omega_i v_i)^2 > F_{i,x}^2 + F_{i,y}^2$, which has the physical interpretation that the force perpendicular to the forward facing force vector $f_i e_1$ (a centripetal force that enables to robot to stay on its curve) can not become larger than the magnitude of the reference force and virtual input combined (so we need $m_i\omega_i v_i \leq ||F_i||$, with F_i from (3.33)). This is a limiting factor since it means that we might not always find an input f_i that provides the wanted virtual input u_i. However, for now we accept this limitation, but we provide a solution for a similar problem with the quadrotor UAV. The smallest rotation between the left and right side of (3.34) is obtained if f_i equals

$$f_i = \operatorname{sgn}(F_{i,x})\sqrt{F_{i,x}^2 + F_{i,y}^2 - (m_i\omega_i v_i)^2}. \tag{3.37}$$

In order to achieve the virtual input u_i, besides the right magnitude f_i with (3.37), we have to make the quadrotor UAV attain the correct attitude R_i. Since we can not simply choose the orientation R_i, we try to achieve the desired orientation over time asymptotically. The relation (3.34) shows us that the misalignment between both sides equals $R_{i,r}^\top R_i$. Let us define the desired heading $R_{i,d}$, such that

$$R_{i,d}\begin{bmatrix} f_i \\ m_i\omega_i v_i \end{bmatrix} = \begin{bmatrix} F_{i,x} \\ F_{i,y} \end{bmatrix}, \tag{3.38}$$

provides the desired virtual input u_i (incorporated in $F_{i,x}$ and $F_{i,y}$). The rotation angle between two arbitrary vectors $\vec{u}$ and $\vec{v}$ can be defined by [47]

$$\cos\theta = \frac{\vec{u} \bullet \vec{v}}{||\vec{u}||\ ||\vec{v}||}, \tag{3.39}$$

which by using the vectors (3.38), leads to

$$\cos\phi_{i,d} = \frac{(f_i e_1 + m_i v_i\omega_i e_2) \bullet (F_{i,x}e_1 + F_{i,y}e_2)}{||f_i e_1 + m_i v_i \omega_i e_2||\ ||F_i||}. \tag{3.40}$$

Notice that $||f_i e_1 + m_i v_i\omega_i e_2|| = \sqrt{f_i^2 + (m_i v_i\omega_i)^2} = \sqrt{F_{i,x}^2 + F_{i,y}^2 - (m_i v_i\omega_i)^2 + (m_i v_i\omega_i)^2} = \sqrt{F_{i,x}^2 + F_{i,y}^2}$, so (3.40) equals

$$\begin{aligned}
\cos\phi_{i,d} &= \frac{\overbrace{\operatorname{sgn}(F_{i,x})F_{i,x}}^{|F_{i,x}|}\sqrt{F_{i,x}^2 + F_{i,y}^2 - (m_i v_i\omega_i)^2} + F_{i,y}m_i v_i\omega_i}{F_{i,x}^2 + F_{i,y}^2}, && (3.41a)\\
&= \frac{F_{i,x}f_i + F_{i,y}m_i v_i\omega_i}{\sqrt{f_i^2 + (m_i v_i\omega_i)^2}\sqrt{F_{i,x}^2 + F_{i,y}^2}}. && (3.41b)
\end{aligned}$$

Similarly, we use $\sin\theta = \frac{||\vec{u}\times\vec{v}||}{||\vec{u}||\ ||\vec{v}||}$ [47], leading to

$$\begin{aligned}
\sin\phi_{i,d} &= \frac{\operatorname{sgn}(F_{i,x})F_{i,y}\sqrt{F_{i,x}^2 + F_{i,y}^2 - (m_i v_i\omega_i)^2} - F_{i,x}m_i v_i\omega_i}{F_{i,x}^2 + F_{i,y}^2}, && (3.42a)\\
&= \frac{F_{i,y}f_i - F_{i,x}m_i v_i\omega_i}{F_{i,x}^2 + F_{i,y}^2}. && (3.42b)
\end{aligned}$$

Altogether, the obtained heading information can be used to obtain the rotation matrix

$$R_{i,d} = \begin{bmatrix} \frac{F_{i,x}f_i + F_{i,y}m_i v_i\omega_i}{F_{i,x}^2 + F_{i,y}^2} & -\frac{F_{i,y}f_i - F_{i,x}m_i v_i\omega_i}{F_{i,x}^2 + F_{i,y}^2} \\ \frac{F_{i,y}f_i - F_{i,x}m_i v_i\omega_i}{F_{i,x}^2 + F_{i,y}^2} & \frac{F_{i,x}f_i + F_{i,y}m_i v_i\omega_i}{F_{i,x}^2 + F_{i,y}^2} \end{bmatrix} \in SO(2). \tag{3.43}$$

Following from 3.43, the goal now is to let $R_{i,r}^\top R_i$ converge to $R_{i,d}$, so that we obtain the size of the desired input with f_i and asymptotically converge to the desired heading by finding an appropriate control law τ_i. In order to find this input τ_i, we define the orientation error in the body-fixed frame of the mobile robot

$$R_{i,e} = R_{i,d}^\top(R_{i,r}^\top R_i), \tag{3.44a}$$

with associated angular velocity error

$$\omega_{i,e} = \omega_i - \omega_{i,r} - \omega_{i,d}. \tag{3.44b}$$

Differentiating (3.44) along the solutions yields the dynamics

$$\dot{R}_{i,e} = R_{i,e}\bar{S}(\omega_{i,e}), \tag{3.45a}$$

$$J_i\dot{\omega}_{i,e} = \tau_i - \tau_{i,r} - J_i\dot{\omega}_{i,d}. \tag{3.45b}$$

In order to stabilize the attitude tracking dynamics (3.45), we formulate an orientation tracking problem.

Problem 3.2.1. (Orientation tracking problem) *Consider the dynamics* (3.45) *following from the corresponding error definitions* (3.44). *Assume that we desire the equilibrium* $(I, 0) \in SO(2)\times\mathbb{R}$. *Find an appropriate control law*

$$\tau_i = \tau_i(t, R_{i,e}, \omega_{i,e}) \in \mathbb{R}), \tag{3.46}$$

such that for the resulting closed-loop system (3.45) *and* (3.46)

$$\lim_{t\to\infty} R_{i,e} = I, \qquad \textit{and} \qquad \lim_{t\to\infty} \omega_{i,e} = 0.$$

In order to stabilize the dynamics (3.45) with the input τ_i, we have to know $\dot{\omega}_{i,d}$, so that we can cancel this term in order to achieve the known stable result from Theorem 2.2.3. Note that we can also base the orientation error coordinates and controller on [46], which in fact has already been used to couple multiple unicycle robots in [15, 43]. However, a huge downside from that approach is that is has a problem with the wrapping of the orientation error; in [15] the orientation error is controlled to the absolute origin. This is undesirable since this can cause steering of multiple full rotations in order to achieve the absolute origin; e.g., when the robot initially starts with an angular heading error greater than 2π. The behavior is only worsened when multiple agents are mutually coupled, since we then have to control both the error relative to the reference to the absolute origin as well as the coupling errors, each potentially with their own offset of aa number of full rotations. In order to overcome this effect, we try to stay close to [11] and focus on the results from Theorem 2.2.3, but on $SO(2) \times \mathbb{R}$.

We first provide the desired angular velocity $\omega_{i,d}$, defined by the relation $\dot{R}_{i,d} = R_{i,d}\bar{S}(\omega_{i,d})$. Following from (3.43), the time-derivative of the rotation matrix equals

$$\dot{R}_{i,d} = \begin{bmatrix} \gamma_{\cos} & -\gamma_{\sin} \\ \gamma_{\sin} & \gamma_{\cos} \end{bmatrix}. \tag{3.47}$$

For notation efficiency, define

$$l_o = F_{i,x}f_i + F_{i,y}m_iv_i\omega_i, \tag{3.48a}$$

$$l_a = F_{i,y}f_i - F_{i,x}m_iv_i\omega_i, \tag{3.48b}$$

$$l_h = F_{i,x}^2 + F_{i,y}^2, \tag{3.48c}$$

with the opposite side to the angle $\phi_{i,d}$ denoted by l_o, the adjacent side denoted by l_a, and the hypotenuse side denoted by l_h. As a direct result, we are able to write

$$\cos\phi_{i,d} = \frac{l_o}{l_h}, \sin\phi_{i,d} = \frac{l_a}{l_h}. \tag{3.49}$$

By utilizing (3.49), define the time derivatives $\gamma_{\cos} = \frac{\mathrm{d}}{\mathrm{d}t}\cos\phi_{i,d}$ and $\gamma_{\sin} = \frac{\mathrm{d}}{\mathrm{d}t}\sin\phi_{i,d}$, as

$$\gamma_{\cos} = \frac{\dot{l}_o}{l_h} - \frac{l_o\dot{l}_h}{l_h^2}, \tag{3.50a}$$

$$\gamma_{\sin} = \frac{\dot{l}_a}{l_h} - \frac{l_a\dot{l}_h}{l_h^2}, \tag{3.50b}$$

with

$$\dot{l}_o = \dot{F}_x f_i + F_{i,x}\dot{f}_{i,d} + \dot{F}_y m_i v_i \omega_i + F_{i,y} m_i \dot{v}_i \omega_i + F_{i,y} m_i v_i \dot{\omega}_i, \tag{3.51a}$$

$$\dot{l}_a = \dot{F}_y f_i + F_{i,y}\dot{f}_{i,d} - \dot{F}_x m_i v_i \omega_i - F_{i,x} m_i \dot{v}_i \omega_i - F_{i,x} m_i v_i \dot{\omega}_i, \tag{3.51b}$$

$$\dot{l}_h = 2(\dot{F}_x F_{i,x} + \dot{F}_y F_{i,y}), \tag{3.51c}$$

in which

$$\begin{bmatrix}\dot{F}_x \\ \dot{F}_y\end{bmatrix} = \dot{f}_{i,r} e_1 + m_i(\dot{\omega}_{i,r}) v_{i,r} + \omega_{i,r}\dot{v}_{i,r}) e_2 \ldots \tag{3.52a}$$

$$+ S(\omega_{i,r} R_{i,r}^\top R_f m_i u_i - R_{i,r}^\top R_f S(\omega_f) m_i u_i - R_{i,r}^\top R_f m_i \dot{u}_i,$$

$$\dot{u}_i = -k_{i,\rho}\dot{\rho}_{i,e} - K_{i,v}\dot{v}_{i,e}, \tag{3.52b}$$

$$\dot{f}_{i,r} = m_i \dddot{\rho}_{i,r}, \tag{3.52c}$$

$$\dot{f}_i = \frac{(F_{i,x}\dot{F}_x + F_{i,y}\dot{F}_y - (m_i\omega_i v_i)(m_i\dot{\omega}_i v_i + m_i\omega_i\dot{v}_i)}{\sqrt{F_{i,x}^2 + F_{i,y}^2 - (m_i\omega_i v_i)^2}}. \tag{3.52d}$$

Notice that $f_{i,r}$ can be obtained by using Newton's Second Law, so to find $\dot{f}_{i,r}$, we differentiate both sides with respect to time. Using $\dot{R}_{i,d} = R_{i,d}\bar{S}(\omega_{i,d})$ to find $\omega_{i,d}$, we obtain

$$\bar{S}(\omega_{i,d}) = R_{i,d}^\top \dot{R}_{i,d}, \tag{3.53a}$$

$$\omega_{i,d} = \cos\phi_{i,d}\gamma_{\sin} - \sin\phi_{i,d}\gamma_{\cos}. \tag{3.53b}$$

Subsequently, the desired angular acceleration term $\dot{\omega}_{i,d}$ is obtained by differentiating (3.53b), providing

$$\dot{\omega}_{i,d} = -\omega_{i,d}\sin\phi_{i,d}\gamma_{\sin} + \cos\phi_{i,d}\dot{\gamma}_{\sin} - \omega_{i,d}\cos\phi_{i,d}\gamma_{\cos} - \sin\phi_{i,d}\dot{\gamma}_{\cos}. \tag{3.54}$$

What remains is to obtain $\dot{\gamma}_{\cos}$ and $\dot{\gamma}_{\sin}$, which we obtain as the time derivatives of (3.50), equal to

$$\dot{\gamma}_{\cos} = \frac{\ddot{l}_o}{l_h} - \frac{\dot{l}_o\dot{l}_h}{l_h^2} - \frac{\dot{l}_o\dot{l}_h + l_o\ddot{l}_h}{l_h^2} + \frac{2l_o\dot{l}_h l_h\dot{l}_h}{l_h^4}, \tag{3.55a}$$

$$\dot{\gamma}_{\sin} = \frac{\ddot{l}_a}{l_h} - \frac{\dot{l}_a\dot{l}_h}{l_h^2} - \frac{\dot{l}_a\dot{l}_h + l_a\ddot{l}_h}{l_h^2} + \frac{2l_a\dot{l}_h l_h\dot{l}_h}{l_h^4}. \tag{3.55b}$$

From (3.51) follows

$$\ddot{l}_o = \ddot{F}_x f_i + 2\dot{F}_x\dot{f}_{i,d} + F_{i,x}\ddot{f}_{i,d} + \ddot{F}_y m_i v_i\omega_i + 2\dot{F}_y m_i\dot{v}_i\omega_i + 2\dot{F}_y m_i v_i\dot{\omega}_i \ldots \tag{3.56a}$$

$$+ F_{i,y} m_i\ddot{v}_i\omega_i + 2F_{i,y} m_i\dot{v}_i\dot{\omega}_i + F_{i,y} m_i v_i\ddot{\omega}_i,$$

$$\ddot{l}_a = \ddot{F}_y f_i + 2\dot{F}_y\dot{f}_{i,d} + F_{i,y}\ddot{f}_{i,d} - \ddot{F}_x m_i v_i\omega_i - 2\dot{F}_x m_i\dot{v}_i\omega_i - 2\dot{F}_x m_i v_i\dot{\omega}_i \ldots \tag{3.56b}$$

$$- F_{i,x} m_i\ddot{v}_i\omega_i - 2F_{i,x} m_i\dot{v}_i\dot{\omega}_i - F_{i,x} m_i v_i\ddot{\omega}_i,$$

$$\ddot{l}_h = 2(\ddot{F}_x F_{i,x} + \ddot{F}_y F_{i,y} + \dot{F}_x^2 + \dot{F}_y^2), \tag{3.56c}$$

in which

$$\begin{bmatrix}\ddot{F}_x \\ \ddot{F}_y\end{bmatrix} = m_i\left[\frac{\ddot{f}_{i,r}}{m_i} e_1 + (\ddot{\omega}_{i,r} v_{i,r} + \dot{\omega}_{i,r}\dot{v}_{i,r} + (\dot{\omega}_{i,r}\dot{v}_{i,r}) + \omega_{i,r}\ddot{v}_{i,r})) e_2 - \ddot{u}_i\right], \tag{3.57a}$$

$$\ddot{u}_i = -k_{i,\rho}\ddot{\rho}_{i,e} - K_{i,v}\ddot{v}_{i,e}, \tag{3.57b}$$

$$\ddot{f}_{i,r} = m_i\rho_{i,r}^{(4)}, \tag{3.57c}$$

$$\ddot{f}_{i,d} = \frac{(F_{i,x}\dot{F}_x + F_{i,y}\dot{F}_y - (m_i\omega_i v_i)(m_i\dot{\omega}_i v_i + m_i\omega_i\dot{v}_i))^2}{\sqrt{F_{i,x}^2 + F_{i,y}^2 - (m_i\omega_i v_i)^2}^3}, \tag{3.57d}$$

Notice that in order for $\dddot{f}_{i,r}$ to exist, $\rho_{i,r}^{(4)}$ has to exist. So in addition to the required three times differentiability for feasibility of the reference trajectory, we need need the virtual structure from (3.20) to provide

$$\rho_{i,r}^{(4)} = \rho_f^{(4)} + R_f^{(4)} p_i + 4\dddot{R}_f \dot{p}_i + 6\ddot{R}_f \ddot{p}_i + 4\dot{R}_f \dddot{p}_i + R_f p_i^{(4)}, \tag{3.58}$$

meaning that ρ_f, R_f and p_i need to be four times differentiable in order to find existence of $\rho_{i,r}^{(4)}$ and being able to provide the required expression for $\dot{\omega}_{i,d}$.

Since we now have the explicit expression for $\dot{\omega}_{i,d}$, what remains is to define the control law for τ_i to stabilize the dynamics (3.45). We present our proposition for a solution to Problem 3.2.1 as follows.

Proposition 3.2.2. *Consider the attitude tracking error dynamics* (3.45)*, error definitions* (3.44)*, desired orientation* (3.43)*, desired angular velocity* (3.53b) *and desired angular acceleration* (3.54)*. Choose the control law*

$$\tau_i = \tau_{r,i} + J_i \dot{\omega}_{i,d} + \frac{1}{2} c_{i,R} \operatorname{Tr}(R_{i,e} \bar{G}) - c_{i,\omega} \omega_{i,e}, \tag{3.59}$$

which yields the closed-loop dynamics

$$\dot{R}_{i,e} = R_{i,e} \bar{S}(\omega_{i,e}), \tag{3.60a}$$

$$J_i \dot{\omega}_{i,e} = -c_{i,\omega}\omega_{i,e} + \frac{1}{2} c_{i,R} \operatorname{Tr}(R_{i,e} \bar{G}), \tag{3.60b}$$

with $c_{i,\omega} > 0$ and $c_{i,R} > 0$. The control law (3.59) *uniformly asymptotically stabilized the solutions of the closed-loop system* (3.60) *in the almost-global region of attraction $(R_{i,e}, \omega_{i,e}) \in G \subset SO(2) \times \mathbb{R}$. All trajectories starting in this region converge to preferred equilibrium $R_{i,e} = I$ and $\omega_{i,e} = 0$. That means, except for a set $\mathcal{M} = SO(2) \times \mathbb{R} \setminus G$ with zero Lebesgue measure, all trajectories converge to the preferred equilibrium. The system is uniformly almost-globally asymptotically stable (UaGAS).*

In Theorem 2.2.2 the closed-loop system (3.60) is presented and proven ULES and UaGAS. Notice that a direct consequence of the asymptotic convergence of $R_{i,e}$ and $\omega_{i,e}$ to the desired equilibrium is the asymptotic convergence of the force vector

$$R_{i,r}^\top R_i \begin{bmatrix} f_i \\ m_i \omega_i v_i \end{bmatrix}$$

to the desired force vector (3.38). However, although the virtual input u_i is thus asymptotically achieved in time, it is not perfectly achieved at all times, following from possible remaining misalignment between the actual heading $R_{i,r}^\top R_i$ and desired heading $R_{i,d}$; this means that the error $R_{i,e} = R_{i,d}^\top R_{i,r}^\top R_i$ is not yet in equilibrium state. Therefore, the position tracking subsystem (3.26) is perturbed by the solution of the orientation tracking subsystem (3.60), so we evaluate the combined system using cascade system theory.

3.2.3 Cascade system analysis

In the previous sections, we determined a desired control action by means of a control law for the position tracking sub-dynamics, as well as a controller for f_i and τ_i in order to achieve the desired control action. The desired orientation and longitudinal acceleration, successively, are obtained by the inputs τ_i and f_i. In fact, we identify a cascaded structure [41, 42], in the closed-loop dynamics

((3.9), (3.12), (3.25), (3.37), (3.59)), equal to

$$\dot{\rho}_{i,e} = -\bar{S}(\omega_{i,r})\rho_{i,e} + v_{i,e}, \tag{3.61a}$$

$$\dot{v}_{i,e} = -\bar{S}(\omega_{i,r})v_{i,e} - k_{i,\rho}\rho_{i,e} - K_{i,v}v_{i,e} + \frac{f_i}{m_i}R_{i,r}^\top R_i(I - R_{i,e}^\top)e_1, \tag{3.61b}$$

$$\dot{R}_{i,e} = R_{i,e}\bar{S}(\omega_{i,e}), \tag{3.61c}$$

$$\dot{\omega}_{i,e} = -c_{i,\omega}\omega_{i,e} + \frac{1}{2}c_{i,R}\operatorname{Tr}(R_{i,e}\bar{G}). \tag{3.61d}$$

The cascaded term

$$g(t, \rho_{i,e}, v_{i,e}, R_{i,e}, \omega_{i,e})R_{i,e} = \frac{f_i}{m_i}R_{i,r}^\top R_i(I - R_{i,e}^\top)e_1 \tag{3.62}$$

results from the fact that the position tracking subsystem is perturbed by the control action f_i pointing not exactly in the desired direction. We identify the two systems in cascade, being ((3.61a),(3.61b)) and ((3.61c),(3.61d)). If we are able to conclude

$$\lim_{t\to\infty}\rho_{i,e} = 0, \qquad \lim_{t\to\infty}v_{i,e} = 0, \qquad \lim_{t\to\infty}R_{i,e} = I, \text{ and } \lim_{t\to\infty}\omega_{i,e} = 0, \tag{3.63}$$

the tracking control problem is solved, since if (3.63) holds, the conditions from Problem 3.1.1 hold automatically. The result is presented as follows.

Proposition 3.2.3. *Consider the closed-loop cascaded system 3.61. The solutions of the closed-loop cascaded system* (3.61) *are uniformly almost-globally asymptotically stable. The solutions of* ((3.61a),(3.61b)) *asymptotically converge to the origin for all $\rho_{i,e} \in \mathbb{R}^2$, $v_{i,e} \in \mathbb{R}$ and $(R_{i,e}, \omega_{i,e}) \in G$ with $G \subset SO(2) \times \mathbb{R}$ and $\mathcal{M} = (SO(2) \times \mathbb{R}) \setminus G$ a set with measure zero.*

Proof. The orientation tracking subsystem ((3.61c),(3.61d)) is UaGAS, which proves the first assumption from Theorem 2.2.4. We consider the attitude tracking subsystem in the region $G \subset SO(2) \times \mathbb{R}$, which is the almost global region of attraction in which our system converges to the desired equilibrium. The stability analysis of the cascaded structure is thus considered on $\mathbb{R}^4 \times G$. Furthermore, we have (3.26) UGAS, providing the second assumption from Theorem 2.2.4. What remains is to prove that the cascaded term from (3.61b) is bounded, providing that stability is remained for the two subsystems in cascade. In order to prove this boundedness, similar to [11], consider

$$V = \frac{1}{2}\rho_{i,e}^\top k_{i,\rho}\rho_{i,e} + \frac{1}{2}v_{i,e}^\top v_{i,e} \geq 0, \tag{3.64}$$

with time derivative along the solutions of (3.61)

$$\dot{V}(\rho_{i,e}, v_{i,e}) = \frac{1}{2}\dot{\rho}_{i,e}^\top k_{i,\rho}\rho_{i,e} + \frac{1}{2}\rho_{i,e}^\top k_{i,\rho}\dot{\rho}_{i,e} + \frac{1}{2}\dot{v}_{i,e}^\top v_{i,e} + \frac{1}{2}v_{i,e}^\top \dot{v}_{i,e}, \tag{3.65a}$$

$$\begin{aligned}
&= \frac{1}{2}\left[-\bar{S}(\omega_{i,r})\rho_{i,e} + v_{i,e}\right]^\top k_{i,\rho}\rho_{i,e} + \frac{1}{2}\rho_{i,e}^\top k_{i,\rho}\left[-\bar{S}(\omega_{i,r})\rho_{i,e} + v_{i,e}\right]\dots \\
&\quad + \frac{1}{2}\left[-\bar{S}(\omega_{i,r})v_{i,e} - k_{i,\rho}\rho_{i,e} - K_{i,v}v_{i,e} + \frac{f_i}{m_i}R_{i,r}^\top R_i(I - R_{i,e}^\top)e_1\right]^\top v_{i,e} + \dots \\
&\quad \frac{1}{2}v_{i,e}^\top\left[-\bar{S}(\omega_{i,r})v_{i,e} - k_{i,\rho}\rho_{i,e} - K_{i,v}v_{i,e} + \frac{f_i}{m_i}R_{i,r}^\top R_i(I - R_{i,e}^\top)e_1\right],
\end{aligned} \tag{3.65b}$$

$$= v_{i,e}^\top\left[\frac{f_i}{m_i}R_{i,r}^\top R_i(I - R_{i,e}^\top)e_1\right] - v_{i,e}^\top K_{i,v}v_{i,e}, \tag{3.65c}$$

$$\leq v_{i,e}^\top\left[\frac{f_i}{m_i}R_{i,r}^\top R_i(I - R_{i,e}^\top)e_1\right] \leq c_{i,1}\sqrt{V}||I - R_{i,e}^\top||, \tag{3.65d}$$

in which we have used that $||R_{i,r}^{\top} R_i|| = 1$. Since ((3.61c),(3.61d)) is ULES, we have

$$\sqrt{V(t)} - \sqrt{V(t_0)} \leq c_{i,2}(t_0). \tag{3.66}$$

So V is bounded and therefore the solutions of ((3.61a),(3.61b)) are bounded, which ensures that the coupling term is bounded. Thus, the last assumption from Theorem 2.2.4 holds, proving that the cascaded system is UaGAS. □

3.3 Mutual coupling of multiple unicycle robots

Until now, we provided a virtual structure approach that generates a set of feasible reference trajectories and provided a controller that allows us to track a specific feasible reference with a specific unicycle robot. Based on these results, we are able to track the entire virtual structure of n virtual robots, with n unicycle robots, allowing us to drive in formation. However, due to the absence of connectivity between the robots, the robots are unable to take action on disturbances that happen to the others, so we can not guarantee any performance for the combined formation tracking behavior. In order to overcome this problem, we want to implement coupling between the robots, so that the overall tracking behavior of the entire formation is somewhat a consideration between tracking the individual reference trajectories and staying in formation. With different relative cost between reference tracking and coupling errors, we are able to shift the weight to a specific preference, so that obtaining the formation shape or converging to the individual reference state can be made more important. Notice that these two tasks do not always have to be conflicting, since in the ideal situation the robots are at their own reference position and in perfect formation. Directly after initiation of the system and as a result of disturbances, the formation is not yet fully obtained and the specific tracking behavior to overcome this imperfection can be altered by altering the relative cost between getting in formation and tracking the individual reference.

Based on the cascaded structure of the developed system for a single agent, we choose to only implement coupling in the position tracking subdynamics. We are able then still operate the orientation tracking subsystem entirely locally, so that the original subsystem remains in tact. This choice allows the orientation to work individually and remain fully in service of the position tracking performance. Additionally coupling the orientation worsens the behavior, as the desired heading then consist of a consideration between the actual desired heading to obtain the virtual input and reducing the angular difference between two robots.

3.3.1 Coupling with position reference tracking

Since we now know which dynamics we want to synchronize, we have to decide on how we want to achieve this goal. Coupling with the position tracking sybsystem can be obtained in several ways, namely, implemented on the reference structure [17, 18], directly in the control law [15] or by considering coupling terms in the generalized coordinates [13]. Furthermore, we can choose a master-slave approach, or consider mutually coupling. The latter is preferred for the unicycle robots, since we consider a more or less homogeneous fleet of unicycle robots and try to obtain a formation that is scalable to any number of agents. Notice that the previously proven stability of the closed-loop position tracking subsystem (3.26) is depending on the gain $k_{i,\rho}$ on position level being scalar. By including coupling directly in the control law, as in [15,43], we are unable to prove stability in a similar manner, as we would have a coupling matrix including cross-coupling between the position errors of the mutually coupled agents. In order to overcome this exact problem, let us choose our error definitions differently from the uncoupled scenario, so that we include coupling errors.

Definition 3.3.1 (cf. [48, 49]). *$N_i \subset \mathcal{V}$ is the set of neighbors of vertex $i \in \mathcal{V}$ defined by*

$$N_i = \{j \in \mathcal{V} | j \neq i \text{ and } a_{ij} \neq 0\}. \tag{3.67}$$

Subsequently, inspired by [18], we define generalized coordinates including coupling errors, as

$$\rho_{i,e} = R_f^\top \left[(\rho_{i,r} - \rho_i) + \sum_{j \in N_i} \tilde{k}_{ij,\rho}((\rho_{j,r} - \rho_j) - (\rho_{i,r} - \rho_i)) \right], \tag{3.68a}$$

$$\nu_{i,e} = R_f^\top \left[(R_{i,r} v_{i,r} e_1 - R_i v_i e_1) + \sum_{j \in N_i} \tilde{k}_{ij,v}((R_{j,r} v_{j,r} e_1 - R_j v_j e_1) - (R_{i,r} v_{i,r} e_1 - R_i v_i e_1)) \right], \tag{3.68b}$$

with $\tilde{k}_{ij,\rho} = \tilde{k}_{ji,\rho}$ and $\tilde{k}_{ij,v} = \tilde{k}_{ji,v}$ denoting the coupling strength between agent i and j on position and velocity level [50], respectively. We choose the coupling strength in both directions equal to underline the homogeneity in mutual coupling. Differentiating (3.68) along the dynamics (3.9) and (3.12), yields

$$\dot{\rho}_{i,e} = -S(\omega_f)\rho_{i,e} + v_{i,e}, \tag{3.69a}$$

$$\begin{aligned} \dot{v}_{i,e} = -S(\omega_f)v_{i,e} + R_f^\top \Bigg[\Bigg(& R_{i,r}\bar{S}(\omega_{i,r})v_{i,r}e_1 e_1 + R_{i,r}\frac{f_{i,r}}{m_i}e_1 - R_i\bar{S}(\omega_i)v_i e_1 - R_i\frac{f_i}{m_i}e_1 \Bigg) \dots \\ + \sum_{j \in N_i} \tilde{k}_{ij,v} \Bigg[\Bigg(& R_{j,r}\bar{S}(\omega_{j,r})v_{j,r}e_1 + R_{j,r}\frac{f_{j,r}}{m_j}e_1 - R_j\bar{S}(\omega_j)v_j e_1 - R_j\frac{f_i}{m_j}e_1 \Bigg) \dots \\ & - \Bigg(R_{i,r}\bar{S}(\omega_{i,r})v_{i,r}e_1 + R_{i,r}\frac{f_{i,r}}{m_i}e_1 - R_i\bar{S}(\omega_i)v_i e_1 - R_i\frac{f_i}{m_i}e_1 \Bigg) \Bigg] \Bigg]. \end{aligned} \tag{3.69b}$$

Define the virtual input

$$\begin{aligned} \tilde{u}_i = R_f^\top \Bigg[\Bigg(& R_{i,r}\bar{S}(\omega_{i,r})v_{i,r}e_1 e_1 + R_{i,r}\frac{f_{i,r}}{m_i}e_1 - R_i\bar{S}(\omega_i)v_i e_1 - R_i\frac{f_i}{m_i}e_1 \Bigg) \dots \\ + \sum_{j \in N_i} \tilde{k}_{ij,v} \Bigg[\Bigg(& R_{j,r}\bar{S}(\omega_{j,r})v_{j,r}e_1 + R_{j,r}\frac{f_{j,r}}{m_j}e_1 - R_j\bar{S}(\omega_j)v_j e_1 - R_j\frac{f_i}{m_j}e_1 \Bigg) \dots \\ & - \Bigg(R_{i,r}\bar{S}(\omega_{i,r})v_{i,r}e_1 + R_{i,r}\frac{f_{i,r}}{m_i}e_1 - R_i\bar{S}(\omega_i)v_i e_1 - R_i\frac{f_i}{m_i}e_1 \Bigg) \Bigg] \Bigg], \end{aligned} \tag{3.70}$$

so that we obtain

$$\dot{\rho}_{i,e} = -S(\omega_f)\rho_{i,e} + v_{i,e}, \tag{3.71a}$$

$$\dot{v}_{i,e} = -S(\omega_f)v_{i,e} + \tilde{u}_i. \tag{3.71b}$$

By using Proposition 3.2.1, we know that choosing the virtual input

$$\tilde{u}_i = -k_{i,\rho}\rho_{i,e} - K_{i,v}v_{i,e}, \tag{3.72}$$

under the assumption that we are able to achieve this virtual input, provides the closed-loop system

$$\dot{\rho}_{i,e} = -S(\omega_f)\rho_{i,e} + v_{i,e}, \tag{3.73a}$$

$$\dot{v}_{i,e} = -S(\omega_f)v_{i,e} - k_{i,\rho}\rho_{i,e} - K_{i,v}v_{i,e}, \tag{3.73b}$$

which is uniformly globally asymptotically stable (UGAS). Since the system (3.73) is equal to the system (3.26) that is considered in Section 3.2.1, the stability proof is provided in Section 3.2.1. What remains is to find the actual inputs f_i and τ_i in order to obtain the virtual input $\tilde{u}_i$.

3.3.2 Achieving the virtual input

In order to achieve the virtual input $\tilde{u}_i$, we again try to obtain the force magnitude input f_i and the desired heading $R_{i,d}$ separately. Following from (3.70), we also find the defined virtual input for the entire system:

$$\tilde{U} = \tilde{K}_v \begin{bmatrix} R_f^\top \left(R_{1,r}\bar{S}(\omega_{1,r})v_{1,r}e_1 + R_{1,r}\frac{f_{1,r}}{m_1}e_1 - R_1\bar{S}(\omega_1)v_1e_1 - R_1\frac{f_1}{m_1}e_1\right) \\ \vdots \\ R_f^\top \left(R_{n,r}\bar{S}(\omega_{n,r})v_{n,r}e_1 + R_{n,r}\frac{f_{n,r}}{m_n}e_1 - R_n\bar{S}(\omega_n)v_ne_1 - R_n\frac{f_n}{m_n}e_1\right) \end{bmatrix} \tag{3.74a}$$

in which

$$\tilde{U} = \begin{bmatrix}(\tilde{u}_1)^\top & \cdots & (\tilde{u}_n)^\top\end{bmatrix}^\top, \tag{3.74b}$$

$$\tilde{K}_v = \begin{bmatrix} I - \sum_{j\in N_1}\tilde{K}_{1j,v} & \tilde{K}_{12,v} & \cdots & \tilde{K}_{1n,v} \\ & I - \sum_{j\in N_2}\tilde{K}_{2j,v} & & \vdots \\ \vdots & & \ddots & \\ \tilde{K}_{n1,v} & \cdots & \tilde{K}_{n(n-1),v} & I - \sum_{j\in N_n}\tilde{K}_{n(n-1),v} \end{bmatrix}, \tag{3.74c}$$

$$K_{ij,v} = I\tilde{k}_{ij,v}. \tag{3.74d}$$

Rewriting (3.74) provides

$$\tilde{U} = \tilde{K}_v G_{R_f} \begin{bmatrix} R_{1,r}m_1\omega_{1,r}v_{1,r}e_2 + R_{1,r}f_{1,r}e_1 - R_1m_1\omega_1v_1e_2 - R_1f_1e_1 \\ \vdots \\ R_{n,r}m_n\omega_{n,r}v_{n,r}e_2 + R_{n,r}f_{n,r}e_1 - R_nm_n\omega_nv_ne_2 - R_nf_ne_1 \end{bmatrix} \tag{3.75a}$$

$$G_R \begin{bmatrix} f_1e_1 + m_1\omega_1v_1e_2 \\ \vdots \\ f_ne_1 + m_n\omega_nv_ne_2 \end{bmatrix} = G_{R_r} \underbrace{\begin{bmatrix} \begin{bmatrix} f_{1,r}e_1 + m_1\omega_{1,r}v_{1,r}e_2 \\ \vdots \\ f_{n,r}e_1 + m_n\omega_{n,r}v_{n,r}e_2 \end{bmatrix} - G_{R_r}^\top G_{R_f}\tilde{K}_v^{-1}\tilde{U} \end{bmatrix}}_{F_d}, \tag{3.75b}$$

in which we use a generator

$$G(R) = G_R = \mathrm{diag}(R_1, \ldots, R_r), \tag{3.76}$$

with $G_R^\top = G(R^\top) = G(R^{-1}) = G_R^{-1}$, following from the special properties of the rotation matrix, as included in Section 2.1. Furthermore, we defined the stack of desired force vectors F_d, which has components

$$F_d = \begin{bmatrix} F_{1,d} \\ \vdots \\ F_{n,d} \end{bmatrix} \tag{3.77}$$

where $F_{i,d} = [\tilde{F}_{i,x}, \tilde{F}_{i,y}]^\top$ and the tilde indicates that we consider mutual coupling, which makes these defined variables different from $F_{i,x}$ and $F_{i,y}$ from Section 3.2.2. However, equal to Section 3.2.2, we can now obtain f_i and $R_{i,d}$ from (3.75b) and (3.77). We obtain

$$f_i = \mathrm{sgn}(\tilde{F}_x)\sqrt{\tilde{F}_x^2 + \tilde{F}_{i,y}^2 - (m_i\omega_iv_i)^2}, \tag{3.78}$$

and

$$R_{i,d} = \begin{bmatrix} \frac{\tilde{F}_x f_i + \tilde{F}_{i,y} m_i v_i \omega_i}{\tilde{F}_x^2 + \tilde{F}_{i,y}^2} & -\frac{\tilde{F}_{i,y} f_i - \tilde{F}_x m_i v_i \omega_i}{\tilde{F}_x^2 + \tilde{F}_{i,y}^2} \\ \frac{\tilde{F}_{i,y} f_i - \tilde{F}_{i,x} m_i v_i \omega_i}{\tilde{F}_x^2 + \tilde{F}_{i,y}^2} & \frac{\tilde{F}_x f_i + \tilde{F}_{i,y} m_i v_i \omega_i}{\tilde{F}_x^2 + \tilde{F}_{i,y}^2} \end{bmatrix} \in SO(2), \tag{3.79}$$

which are exactly the same expressions as for the single-agent system, but with $\tilde{F}_{i,x}$ and $\tilde{F}_{i,y}$ instead of $F_{i,x}$ and $F_{i,y}$. Notice that these two differ as a result of the additional coupling matrix. Since we only implement coupling on position tracking, the orientation controller further remains in tact. Again this means that in the specific situation where $\tilde{F}_x^2 + \tilde{F}_{i,y}^2 < (m_i \omega_i v_i)^2$ we are unable to obtain a suitable f_i to achieve the magnitude of u_i; a situation that corresponds with the scenario of a centripetal force that is larger than the magnitude of the reference vehicle and virtual input combined, as can be seen in (3.75b). For now we accept this outcome in order to later on focus on a solution for the quadrotor UAV. We are now able to use exactly the same orientation dynamics and control law τ_i as in Proposition 3.2.2, in order to asymptotically achieve the desired heading $R_{i,d}$. In order to fully provide the required parameters that are used in the control law, we explicitly provide the required time derivatives of $F_{i,d}$. From Section 3.2.2 follows that the first and second order time derivatives of F_d are required to compute $\dot{\omega}_{i,d}$ for $i = 1, \ldots, n$. We explicitly obtain these time derivatives of F_d, as

$$\dot{F}_d = \begin{bmatrix} \dot{f}_{1,r} e_1 + m_1 \dot{\omega}_{1,r} v_{1,r} + m_1 \omega_{1,r} \dot{v}_{1,r} e_2 \\ \vdots \\ \dot{f}_{n,r} e_1 + m_n \dot{\omega}_{n,r} v_{n,r} e_2 + m_n \omega_{n,r} \dot{v}_{n,r} e_2 \end{bmatrix} \dots \tag{3.80a}$$
$$- G_{\dot{R}_r}^\top G_{R_f} \tilde{K}_v^{-1} \tilde{U} - G_{R_r}^\top G_{\dot{R}_f} \tilde{K}_v^{-1} \tilde{U} - G_{R_r}^\top G_{R_f} \tilde{K}_v^{-1} \dot{\tilde{U}},$$

$$\ddot{F}_d = \begin{bmatrix} \ddot{f}_{1,r} e_1 + 2 m_1 \dot{\omega}_{1,r} \dot{v}_{1,r} + m_1 \ddot{\omega}_{1,r} v_{1,r} + m_1 \omega_{1,r} \ddot{v}_{1,r} e_2 \\ \vdots \\ \ddot{f}_{n,r} e_1 + +2 m_n \dot{\omega}_{n,r} \dot{v}_{n,r} + m_1 \ddot{\omega}_{n,r} v_{n,r} + m_n \omega_{n,r} \ddot{v}_{n,r} e_2 \end{bmatrix} \dots \tag{3.80b}$$
$$- 2 G_{\dot{R}_r}^\top G_{\dot{R}_f} \tilde{K}_v^{-1} \tilde{U} - 2 G_{R_r}^\top G_{\dot{R}_f} \tilde{K}_v^{-1} \dot{\tilde{U}} - 2 G_{\dot{R}_r}^\top G_{R_f} \tilde{K}_v^{-1} \dot{\tilde{U}} \dots$$
$$- G_{\ddot{R}_r}^\top G_{R_f} \tilde{K}_v^{-1} \tilde{U} - G_{R_r}^\top G_{\ddot{R}_f} \tilde{K}_v^{-1} \tilde{U} - G_{R_r}^\top G_{R_f} \tilde{K}_v^{-1} \ddot{\tilde{U}}.$$

The current status is that we are able achieve the virtual input (3.72), by the actual force magnitude input f_i if we asymptotically converge to the desired heading $R_{i,d}$. In order to asymptotically converge to $R_{i,d}$, we follow the same approach as for a single unicycle and define the orientation tracking errors

$$R_{i,e} = R_{i,d}^\top (R_{i,r}^\top R_i), \tag{3.81a}$$
$$\omega_{i,e} = \omega_i - \omega_{i,r} - \omega_{i,d}. \tag{3.81b}$$

Differentiating (3.81) along the solutions yields the dynamics

$$\dot{R}_{i,e} = R_{i,e} \bar{S}(\omega_{i,e}), \tag{3.82a}$$
$$J_i \dot{\omega}_{i,e} = \tau_i - \tau_{i,r} - J_i \dot{\omega}_{i,d}. \tag{3.82b}$$

which is exactly the same as we previously obtained for a single agent. Therefore, we are able to consider the same tracking problem as Problem 3.2.1 with corresponding solution from Proposition 3.2.2; for the proof we refer to the proof from Theorem 2.2.2. Thus, the dynamics (3.82) in closed-loop with (3.59) is UaGas, as proven in Section 3.2.2. Thus, based on these results we conclude that we asymptotically achieve the virtual input $\tilde{u}_i$ by the actual inputs f_i and τ_i. However, since we asymptotically achieve $\tilde{u}_i$ in time rather than perfectly at any time, the position tracking subsystem is perturbed by the solution of the orientation tracking subsystem. We identify a cascaded structure.

3.3.3 Cascade system analysis

In the previous sections, we determined virtual control action by means of a control law for the position tracking sub-dynamics, as well as a controller for f_i and τ_i in order to achieve the virtual control action. The desired orientation and longitudinal acceleration, successively, are obtained by the inputs τ_i and f_i. In fact, we identify a cascaded structure [41, 42]. The closed-loop dynamics (3.9), (3.12), (3.72), (3.78), and (3.59), equal

$$\dot{\rho}_{i,e} = -\bar{S}(\omega_{i,r})\rho_{i,e} + v_{i,e}, \tag{3.83a}$$

$$\dot{v}_{i,e} = -\bar{S}(\omega_{i,r})v_{i,e} - k_{i,\rho}\rho_{i,e} - K_{i,v}v_{i,e} + \frac{f_i}{m_i}R_{i,r}^\top R_i(I - R_{i,e}^\top)e_1, \tag{3.83b}$$

$$\dot{R}_{i,e} = R_{i,e}\bar{S}(\omega_{i,e}), \tag{3.83c}$$

$$\dot{\omega}_{i,e} = -c_{i,\omega}\omega_{i,e} + \frac{1}{2}c_{i,R}\operatorname{Tr}(R_{i,e}\bar{G}). \tag{3.83d}$$

The cascaded term

$$g(t, \rho_{i,e}, v_{i,e}, R_{i,e}, \omega_{i,e})R_{i,e} = \frac{f_i}{m_i}R_{i,r}^\top R_i(I - R_{i,e}^\top)e_1 \tag{3.84}$$

again results from the fact that the position tracking subsystem is perturbed by the control action f_i pointing not exactly in the desired direction. If we are able to conclude

$$\lim_{t\to\infty}\rho_{i,e} = 0, \qquad \lim_{t\to\infty}v_{i,e} = 0, \qquad \lim_{t\to\infty}R_{i,e} = I, \text{ and } \lim_{t\to\infty}\omega_{i,e} = 0, \tag{3.85}$$

the tracking control problem from Problem 3.1.1 is solved, since if (3.63) holds, the conditions from Problem 3.1.1 hold automatically. The dynamics (3.9), (3.12) in closed loop with (3.72), (3.78), and (3.59) provide the closed-loop cascaded system (3.83) that now includes coupling in the generalized position and velocity errors. By directly using Proposition 3.2.3, we can conclude that the cascaded system is UaGAS.

3.4 Simulation study

Although the stability of the system is already guaranteed by the provided proof of stability for the involved generalized dynamics, we still want to at least get some feeling for the effect that the included coupling structure has on the formation tracking behavior. In order to test the effect of the included coupling from the obtained system Section 3.3, we implement the system for simulations. We make the mobile robots initiate relatively far from their reference trajectories in order to show if the coupling provides that we can make the mobile robots drive in formation before they reach the reference trajectories. If that is the case, we thus have control over the trade-off between tracking the formation shape or the individual reference trajectories and can accept the results in order to follow a similar approach for the quadrotor UAVs. The formation trajectory is chosen circular, as

$$\rho_f(t) = \begin{bmatrix}\cos(t) & \sin(t)\end{bmatrix}^\top, \tag{3.86}$$

and the formation shape is defined as

$$p_i(t) = r\begin{bmatrix}\cos(i\frac{(4-n)\pi}{2n}) & \sin(i\frac{(4-n)\pi}{2n})\end{bmatrix}^\top, \tag{3.87}$$

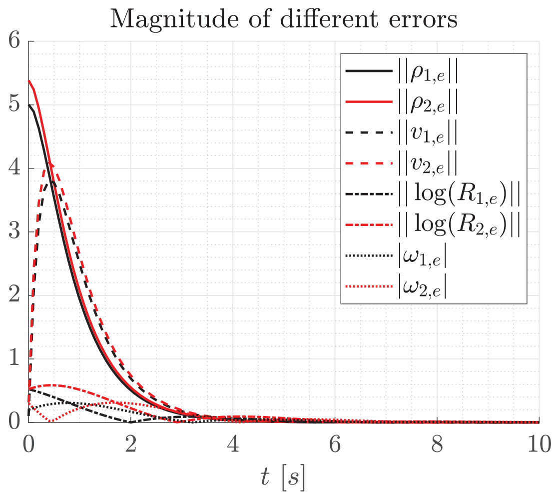
with the agent defined by i, the total number of agents n and the radius $r = 0.2$; this shape function positions a number of n agents equally on a circle circumference with radius r and is actually time invariant. The initial conditions in generalized coordinates are chosen as

$$
\begin{aligned}
\rho_{1,e} &= \begin{bmatrix} -5 & 0.2 \end{bmatrix}^\top, & \rho_{2,e} &= \begin{bmatrix} -5 & -2 \end{bmatrix}^\top, \\
v_{1,e} &= \begin{bmatrix} 0.01 & 0.1 \end{bmatrix}^\top, & v_{2,e} &= \begin{bmatrix} 0.3 & 0.1 \end{bmatrix}^\top, \\
\phi_{1,e} &= -\frac{\pi}{6}, & \phi_{2,e} &= \frac{\pi}{6}, \\
\omega_{1,e} &= 0.2, & \omega_{2,e} &= 0.3.
\end{aligned}
\tag{3.88}
$$

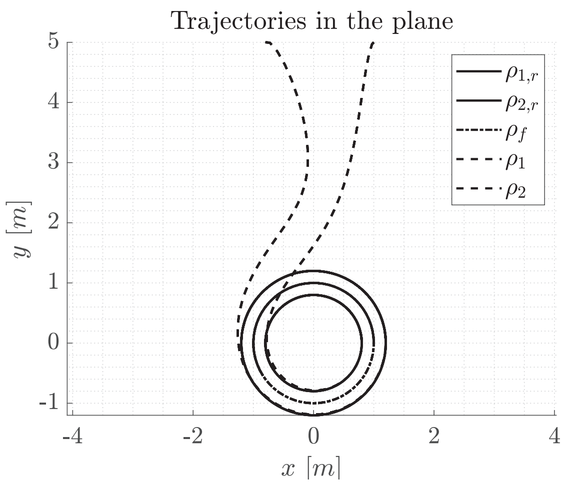
The system parameters are set at

$$
\begin{aligned}
K_{1,\rho} = K_{2,\rho} &= \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}, & \tilde{K}_{12,\rho} = \tilde{K}_{21,\rho} &= \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}, \\
K_{1,v} = K_{2,v} &= \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}, & \tilde{K}_{12,v} = \tilde{K}_{21,v} &= \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}, \\
c_{i,R} &= 1, & m_i = m_i &= 1, \\
c_{i,\omega} &= 1, & J_i = J_{r,i} &= 1.
\end{aligned}
\tag{3.89}
$$

A simulation is executed and the results are implemented in Figure 3.2. First of all, we see that the error coordinates of the position tracking subsystem converge to the absolute origin, which we have also proven mathematically in the previous section. The trajectories in the right graph from Figure 3.2 shows that the robots start to drive in the virtual formation shape even before they reach the actual formation as a result of the coupling terms. This means that the implemented coupling functions provide the ability to control the consideration between tracking the individual reference trajectories and tracking the formation (shape). Therefore, the system of multiple mobile robots shows the wanted behavior, so we follow a similar approach later on for the system of multiple quadrotor UAVs.



(a)



(b)

Figure 3.2: Simulation results in which two mobile robots start tracking a reference formation that moves over a circular path from relatively far away. In (a) the magnitude of the involved error coordinated for both unicycle robots over time and in (b) the trajectories in the plane.

3.5 Concluding remarks

In this chapter, we provided a dynamical model for the unicycle robot, defined error coordinates, and defined a tracking problem. The tracking problem for a unicycle robot is deliberately stated very similar to the problem that we later consider for a quadrotor UAV, in order to use this case study as an introductory problem to the rest of this research. A virtual reference structure is provided by defining the pose and shape of this structure, so that we obtain a set of reference trajectories for the set of unicycle robots. We showed that all reference trajectories are feasible and we are able to obtain the virtual input when the formation trajectory over time ($\rho_{i,r}$), formation heading (R_f) and shape vectors (p_i) all are four times differentiable. To facilitate mutual coupling, we defined the position tracking errors relative to the mutually known formation centered frame, so that individual errors can directly be compared and are independent from the choice of inertial reference frame; which is another (alternative) mutually known frame in the system.

We have shown that if we consider a specific individual robot with its reference trajectory, we are able to stabilize the position tracking dynamics based on a defined virtual input, providing uniform global asymptotic stability (UGAS). After that, we have showed that except for a specific situation where the centripetal force of the unicycle is greater than the magnitude of the virtual input and reference forces combined, we are able to provide the actual force magnitude input and desired heading, based on this virtual input. We also assumed no physical limitations on the achievable force magnitude input and torque inputs, which is why we did not use saturation techniques on the designed virtual input. For now we accept these results in order to later on overcome them for the main focus of this research that is the quadrotor UAV. We then used the orientation tracking subsystem to achieve the desired heading, by defining a control law that renders the orientation tracking dynamics uniformly almost-globally asymptotically stable (UaGAS). Since the orientation of the unicycle robot can be different from the desired orientation during convergence towards this desired orientation, the position tracking error dynamics are perturbed by the solution of the orientation tracking error dynamics. Therefore, the two subsystems in cascade were analyzed using cascade system theory. Based on cascade system theory, we showed that the cascaded system is UaGAS. This stability result of the cascaded system combined with the provided reference structure already allowed us to track a formation with a specific number of robots. However, since there was still no coupling between the robots, this still did not ensure synchronized positioning, since the robots are unable to react to disturbances acting on the other.

Subsequently, in order to have control over the consideration (or trade-off) between tracking the individual trajectories and staying in formation, we included coupling functions in the error definition of the position tracking subsystem. The alternative approach of implementing coupling functions directly in the virtual input was also considered, but we were unable to prove stability because of the resulting coupling matrix in the closed-loop system. We choose to only consider coupling in the position tracking subsystem, since the orientation tracking subsystem is designed fully in service of the position tracking controller. Trying to synchronize orientation then only acts as an additional disturbance to the desired heading angle. Based on our literature research, we did not consider master-slave coupling, because mutual coupling provided better cooperative formation tracking ability. The coupled position tracking subsystem is proven UGAS and we were able to provide the desired heading and force magnitude input. Based on these results, we were able to work with the same orientation error coordinates and apply the same orientation controller as for the uncoupled system, leaving the entire orientation tracking subsystem in tact. We again have the limitations that for some specific cases we are unable to provide a suitable force magnitude input in order to obtain the desired input and we did not consider physical limitations of the motors and therefore saturation. For now we allow these limitations in order to later on overcome these for the main focus that is the quadrotor UAV. Based on the combined results, we have again analyzed the stability of the cascaded structure and proved that the cascaded term is UaGAS.

A brief simulation study showed the behavior in a simple test case and it is shown that it is possible to obtain the formation shape even before the reference states are reached, resulting from the included coupling functions. In the next chapter, we want to follow a very similar approach for a system of quadrotor UAVs, to eventually obtain similar formation tracking (behavior) with quadrotor UAVs.

Chapter 4

Formation tracking with quadrotor UAVs

In the previous chapter, we developed a formation controller that induced mutual coupling between individual agents, in which the agents are represented by unicycle robots. The consecutive intermediate steps deliberately match the objectives of this research, so that we can now follow these same steps but for the quadrotor UAV.

In this chapter, we first introduce the model of a quadrotor UAV, modelled according to the Newton-Euler framework, in order to obtain the same model as used in [19]. After that, we introduce reference dynamics to track and the tracking problems. Subsequently, we first develop a tracking controller for the individual agent, so that we can track provided reference trajectories with a single quadrotor UAV. Then, we provide a virtual structure approach to obtain a set of feasible reference trajectories that together form a reference formation. The set of feasible reference trajectories together with the ability to track a reference trajectory already allows us to track a formation with multiple robots, however, still without any coupling between the agents. Therefore, in order to have some reciprocity between staying in formation and tracking the individual reference, we then induce mutual coupling between the agents, so that we are able to retain the formation even when individual agents are perturbed. At the end, we validate the behavior in some simple simulated case-studies.

4.1 Single-agent reference tracking

First, we want to follow [11,19] in order to control a single quadrotor UAV. This is important since we want to slightly adjust the error definitions, so that the error coordinates of the different agents are mutually comparable. Currently, so in [11] and [19], this comparability is absent, since the error coordinates of the agent are expressed relative to a frame that is only known by the agent itself, following from logical choices that were made without the intention of coupling multiple agents. Therefore, in this section, we first introduce the dynamical model of a quadrotor UAV and state a tracking problem. After that, we follow the controller design procedure from [11] in order to prove the stability of the slightly adjusted controller.

4.1.1 Modeling and Problem definition

Consider a quadrotor UAV i with velocity $\nu_i \in \mathbb{R}^3$ relative to the body fixed reference frame $\mathcal{B}_i$, position $\rho_i \in \mathbb{R}^3$ relative to an inertial frame $\mathcal{I}$, a rotation matrix $R_i \in SO(3)$ that transforms $\mathcal{B}_i$ to $\mathcal{I}$ and angular velocities $\omega_i \in \mathbb{R}^3$ relative to the body-fixed frame $\mathcal{B}_i$. The body-fixed frame $\mathcal{B}_i$ represents the spatial attitude of the quadrotor UAV, which means that R_i equivalently represents the attitude of the body relative to the inertial frame $\mathcal{I}$. A schematic representation of this quadrotor UAV is shown in Figure 4.1. The quadrotor is presented in +-configuration for illustration clarity, meaning that the frame partially aligns with frame $\mathcal{B}_i$. For the rest of this thesis, however, we assume $\times$-configuration, as we intend to develop a system for a quadrotor UAV with a dedicated forward direction according to the $\times$-configuration. The relation between the two configurations is geometrically defined and only shows in the exact conversion from motor inputs $T_{i,j}$ with $j = \{1, 2, 3, 4\}$ to f_i and τ_i. By following the Newton-Euler modelling approach [28], the dynamical

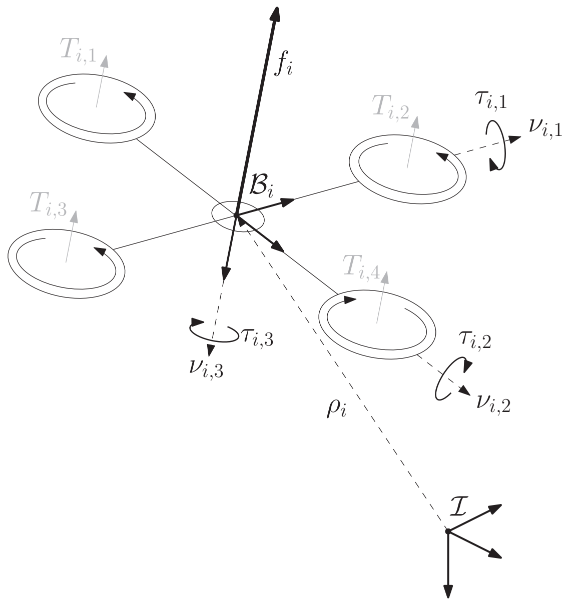


Figure 4.1: Schematic configuration of quadrotor UAV i with body-fixed-frame $\mathcal{B}_i$, relative to Inertial frame $\mathcal{I}$. The rotation matrix R_i rotates $\mathcal{B}_i$ to $\mathcal{I}$. Indicated are the position of the quadrotor ρ_i relative to $\mathcal{I}$, the body fixed accelerations ν_i, and the force f_i and torque τ_i resulting from the individual motors.

model of a quadrotor UAV equals

$$\dot{\rho}_i = R_i \nu_i, \tag{4.1a}$$

$$\dot{\nu}_i = -S(\omega_i)\nu_i + gR_i^\top e_3 - \frac{f_i}{m_i} e_3, \tag{4.1b}$$

$$\dot{R}_i = R_i S(\omega_i), \tag{4.1c}$$

$$J_i \dot{\omega}_i = S(J_i \omega_i)\omega_i + \tau_i, \tag{4.1d}$$

with mass m_i, inertia matrix $J_i = J_i^\top > 0$ with respect to the body-fixed frame $\mathcal{B}_i$, the skew symmetric matrix $S(\cdot)$ from Theorem 2.1.1, the inputs for control $f_i \in \mathbb{R}$ and $\tau_i \in \mathbb{R}^3$. Notice that since the velocity ν_i and angular velocity ω_i are expressed in the body-fixed frame as well as the inertia matrix J_i, force magnitude input f_i and torque input τ_i, the resulting model is relatively simple. The body-fixed frame $\mathcal{B}_i$ is a right-handed reference frame, defined relative to a North-East-Down (NED) inertial frame $\mathcal{I}$. NED coordinate frames are commonly used in the modeling and control of aerial vehicles [51], as they have proven to work particularly well for tasks such as navigation and auto-pilot. In addition to the dynamics of a quadrotor UAV (4.1), assume that we have a feasible reference trajectory, i.e., a trajectory $(\rho_{i,r}, R_{i,r}, \nu_{i,r}, \omega_{i,r}, f_{i,r}, \tau_{i,r})$ that also satisfies the dynamics of a quadrotor

$$\dot{\rho}_{i,r} = R_{i,r}\nu_{i,r}, \tag{4.2a}$$

$$\dot{\nu}_{i,r} = -S(\omega_{i,r})\nu_{i,r} + gR_{i,r}^\top e_3 - \frac{f_{i,r}}{m_i}e_3, \tag{4.2b}$$

$$\dot{R}_{i,r} = R_{i,r}S(\omega_{i,r}), \tag{4.2c}$$

$$J_i\dot{\omega}_{i,r} = S(J_i\omega_{i,r})\omega_{i,r} + \tau_{i,r}, \tag{4.2d}$$

where $0 < f_r^{\min} \leq f_{i,r}(t)$. Similar as for the actual quadrotor UAV, we define a body fixed right-handed reference frame $\mathcal{R}_i$ that is expressed relative to the inertial frame $\mathcal{I}$. As we aim to provide a similar solution to the exact same tracking problem from [11], let us adopt the following error coordinates on $SE(3)$

$$\tilde{\rho}_i = R_{i,r}^\top(\rho_i - \rho_{i,r}), \tag{4.3a}$$

$$\tilde{\nu}_i = \tilde{R}^\top S(\omega_{i,r})\tilde{\rho}_{i,r} + \nu - \tilde{R}_i^\top \nu_{i,r}, \tag{4.3b}$$

$$\tilde{R}_i = R_{i,r}^\top R_i, \tag{4.3c}$$

$$\tilde{\omega}_i = \omega_i - \tilde{R}_i^\top \omega_{i,r}, \tag{4.3d}$$

and adopt the corresponding error measure

$$\varepsilon_i(\tilde{\rho}_i, \tilde{\nu}_i, \tilde{R}_i, \tilde{\omega}_i) = ||\tilde{\rho}_i|| + ||\tilde{\nu}_i|| + ||\log \tilde{R}_i|| + ||\tilde{\omega}_i||, \tag{4.4}$$

for robot i. Then, define the tracking control problem as follows.

Problem 4.1.1 (cf. [11]). *Given a feasible reference trajectory* $(\rho_{i,r}, R_{i,r}, \nu_{i,r}, \omega_{i,r}, f_{i,r}, \tau_{i,r})$ *for robot* i*, find control laws*

$$f_i = f_i(\rho_i, \nu_i, R_i, \omega_i, \rho_{i,r}, \nu_{i,r}, R_{i,r}, \omega_{i,r}), \tag{4.5a}$$

$$\tau_i = \tau_i(\rho_i, \nu_i, R_i, \omega_i, \rho_{i,r}, \nu_{i,r}, R_{i,r}, \omega_{i,r}), \tag{4.5b}$$

such that the resulting closed-loop (4.1), (4.2) *and* (4.5) *yields*

$$\lim_{t\to\infty} \varepsilon_i(\tilde{\rho}_i(t), \tilde{\nu}_i(t), \tilde{R}_i(t), \tilde{\omega}_i(t)) = 0. \tag{4.6}$$

4.1.2 Position reference tracking with a quadrotor UAV

Consider a single quadrotor UAV identified by i from (4.1) and consider the dynamics of a virtual reference quadrotor UAV (4.2). Now, we would like to define generalized coordinates that best facilitate the eventual use for the control of an entire formation. A bad choice, for instance, would be to define generalized coordinates in the reference frame $\mathcal{R}_i$ for robot i, since the error coordinates between different agents would then not be comparable as they are all expressed in their own reference frame; therefore, we slightly diverge from [11] with the objective to enable coupling between agents. A better idea would be to express the error coordinates in the inertial frame $\mathcal{I}$, since this frame is already known by all of the involved agents and the errors are directly comparable since they are expressed in the same mutually known frame. However, by using the inertial frame, we cause different behavior when we make a different choice for the pose of the inertial frame $\mathcal{I}$; e.g., when we rotate the frame by 90 degrees about the z-axis, we change the gains that work on the x-direction and y-direction. Therefore, we imagine that we are able to extend the previously introduced virtual formation structure to the 3D case and let us prematurely introduce a new mutually known frame $\mathcal{F}$ that is also right-handed, denoted as the formation frame. Notice that the frame $\mathcal{F}$ can be translating and rotating in time, which is later on used in order to make a distinction between the formation shape ($\mathcal{R}_i$ relative to $\mathcal{F}$) and the formation pose ($\mathcal{F}$ relative to

$\mathcal{I}$). Let us use the rotation matrix R_f to denote the rotation from $\mathcal{F}$ to $\mathcal{I}$. Subsequently, define the position tracking error coordinates for agent i, expressed in the formation centered reference frame $\mathcal{F}$, as

$$\rho_{i,e} = R_f^\top \left[\rho_{i,r} - \rho_i\right], \tag{4.7a}$$
$$\nu_{i,e} = R_f^\top \left[R_{i,r}\nu_{i,r} - R_i\nu_i\right]. \tag{4.7b}$$

Notice that we are free to choose the formation frame $\mathcal{F}$ as it represents a virtual center. Therefore, in a single agent system, we can choose $\mathcal{F}$ equal to one of the other involved frames to obtain more classical error definitions. Differentiating (4.7) along the dynamics yields

$$\dot{\rho}_{i,e} = -S(\omega_f)\rho_{i,e} + \nu_{i,e}, \tag{4.8a}$$
$$\begin{aligned}\dot{\nu}_{i,e} &= -S(\omega_f)\nu_{i,e} + R_f^\top R_{i,r}S(\omega_{i,r})\nu_{i,r} - R_f^\top R_i S(\omega_i)\nu_i \dots \\ &\quad + R_f^\top R_{i,r}\left[-S(\omega_{i,r})\nu_{i,r} + gR_{i,r}^\top e_3 - \frac{f_{i,r}}{m_i}e_3\right] - R_f^\top R_i\left[-S(\omega_i)\nu_i + gR_i^\top e_3 - \frac{f_i}{m_i}e_3\right],\end{aligned} \tag{4.8b}$$
$$= -S(\omega_f)\nu_{i,e} + R_f^\top R_{i,r}\left[gR_{i,r}^\top e_3 - \frac{f_{i,r}}{m_i}e_3\right] - R_f^\top R_i\left[gR_i^\top e_3 - \frac{f_i}{m_i}e_3\right], \tag{4.8c}$$
$$= -S(\omega_f)\nu_{i,e} + \underbrace{R_f^\top R_i\frac{f_i}{m_i}e_3 - R_f^\top R_{i,r}\frac{f_{i,r}}{m_i}e_3}_{u_i}. \tag{4.8d}$$

Assume that

$$u_i = R_f^\top R_i\frac{f_i}{m_i}e_3 - R_f^\top R_{i,r}\frac{f_{i,r}}{m_i}e_3, \tag{4.9}$$

is a virtual input which later on is achieved by controlling the thrust magnitude f_i and attitude R_i with inputs τ_i. Since the thrust vector in the body-fixed frame f_ie_1 and attitude R_i are both in the same term, we know that we have full control over the defined virtual input, therefore, we consider this virtual input u_i for controller design. The force magnitude f_i is used to obtain the magnitude of the desired virtual input and the attitude R_i is used to point the force vector f_ie_1 in the desired direction. Notice that stabilizing the dynamics (4.8) with error definitions (4.7) also provides a possible solution to Problem 4.1.1. Let us now work towards a possible solution.

In order to stabilize the error dynamics (4.8), we define the feedback control law

$$u_i = -k_{i,\rho}\rho_{i,e} - K_{i,\nu}\nu_{i,e}. \tag{4.10}$$

However, following from (4.9), we see that we can not allow every input size u_i, since there at least is some physical limitation on the maximal thrust magnitude f_i that agent i is able to provide. In order to have u_i bounded, let us use a saturation function $\sigma(\cdot)$, according to Definition 2.2.1, that satisfies $||\sigma(\cdot)|| \leq M$, for some M.
For now, let us step back to the more general definition for a saturation function $\sigma(\cdot)$, from Definition 2.2.1 and Definition 2.2.2. We present our possible solution for stabilizing the dynamics (4.8) as follows.

Proposition 4.1.1. *Consider the generalized dynamics* (4.8)*. Define the feedback control law*

$$u_i = -k_{i,\rho}\sigma(\rho_{i,e}) - K_{i,\nu}\sigma(\nu_{i,e}), \tag{4.11}$$

which provides the closed-loop system

$$\dot{\rho}_{i,e} = -S(\omega_f)\rho_{i,e} + \nu_{i,e}, \tag{4.12a}$$
$$\dot{\nu}_{i,e} = -S(\omega_f)\nu_{i,e} - k_{i,\rho}\sigma(\rho_{i,e}) - K_{i,\nu}\sigma(\nu_{i,e}). \tag{4.12b}$$

The origin of the closed-loop system (4.12) *is uniformly globally asymptotically stable (UGAS).*

Proof. Consider the candidate Lyapunov function

$$V_1(\rho_{i,e}, \nu_{i,e}) = k_{i,\rho} V_\sigma + \frac{1}{2} \nu_{i,e}^\top \nu_{i,e}, \tag{4.13}$$

with V_σ from Definition 2.2.1, scalar gain $k_{i,\rho} > 0$. Note that $V_1(\rho_{i,e}, \nu_{i,e})$ only equals 0 at the absolute origin and in the domain excluding the origin the function is positive definite. Differentiating (4.13) along the solutions (4.12) yields

$$\begin{aligned}
\dot{V}_1(\rho_{i,e}, \nu_{i,e}) &= k_{i,\rho} \dot{\rho}_{i,e}^\top \sigma(\rho_{i,e}) + \nu_{i,e}^\top \dot{\nu}_{i,e}, && (4.14a)\\
&= \left[-S(\omega_f)\rho_{i,e} + \nu_{i,e}\right]^\top k_{i,\rho}\sigma(\rho_{i,e}) + \nu_{i,e}^\top \left[-S(\omega_f)\nu_{i,e} - k_{i,\rho}\sigma(\rho_{i,e}) - K_{i,\nu}\sigma(\nu_{i,e})\right], && (4.14b)\\
&= k_{i,\rho}\rho_{i,e} S(\omega_f)\sigma(\rho_{i,e}) + k_{i,\rho}\nu_{i,e}^\top \sigma(\rho_{i,e}) - \nu_{i,e}^\top \left[S(\omega_f)\nu_{i,e} + k_{i,\rho}\sigma(\rho_{i,e}) + K_{i,\nu}\sigma(\nu_{i,e})\right], && (4.14c)\\
&= -\nu_{i,e}^\top K_{i,\nu}\sigma(\nu_{i,e}) = Y_1(\nu_{i,e}) \leq 0, && (4.14d)
\end{aligned}$$

in which we have used the property of a skew symmetric matrix $b^\top S(a) b = 0$ for $a \in \mathbb{R}^3$ and $b \in \mathbb{R}^3$. Then, since until now we only have $\nu_{i,e}$ included in the time derivative of the Lyapunov function, define

$$\dot{V}_2(\rho_{i,e}, \nu_{i,e}) = \rho_{i,e}^\top \nu_{i,e}, \tag{4.15}$$

with time derivative along the dynamics

$$\begin{aligned}
\dot{V}_2(\rho_{i,e}, \nu_{i,e}) &= \dot{\rho_{i,e}}^\top \nu_{i,e} + \rho_{i,e}^\top \dot{\nu}_{i,e}, && (4.16a)\\
&= \left[-S(\omega_f)\rho_{i,e} + \nu_{i,e}\right]^\top \nu_{i,e} + \rho_{i,e}^\top \left[-S(\omega_f)\nu_{i,e} - k_{i,\rho}\sigma(\rho_{i,e}) - K_{i,\nu}\sigma(\nu_{i,e})\right], && (4.16b)\\
&= \nu_{i,e}^\top \nu_{i,e} - \rho_{i,e}^\top k_{i,\rho}\sigma(\rho_{i,e}) - \rho_{i,e}^\top K_{i,\nu}\sigma(\nu_{i,e}) = Y_2(\rho_{i,e}, \nu_{i,e}). && (4.16c)
\end{aligned}$$

We now use Matrosov's Theorem [37], included as Theorem 2.2.1, in order to conclude asymptotic stability. We have $V_1(\rho_{i,e}, \nu_{i,e}) \geq 0$ provided that $k_{i,\rho} > 0$ and $K_{i,v} = K_{i,v}^\top > 0$, satisfying the first condition from Theorem 2.2.1. Furthermore, by choosing $\dot{V}_1(\rho_{i,e}, \nu_{i,e}) = Y_1(\nu_{i,e})$ and $\dot{V}_2(\rho_{i,e}, \nu_{i,e}) = Y_2(\nu_{i,e}, \rho_{i,e})$ follows that the second condition is automatically satisfied. Then, we find $Y_1(\nu_{i,e}) = 0$ for $\nu_{i,e} = 0$, which implies $Y_2(\rho_{i,e}, 0) = -\rho_{i,e}^\top k_{i,\rho}\sigma(\rho_{i,e}) \leq 0$ provided $k_{i,\rho} > 0$, which means that the third assumption holds. Lastly, the equalities $Y_1(\nu_{i,e}) = 0$ and $Y_2(\rho_{i,e}, \nu_{i,e}) = 0$ only hold at the absolute origin. Concluding, since all conditions from Theorem 2.2.1 are satisfied, the closed-loop system (4.12) is UGAS. □

4.1.3 Attitude control

In the previous part we have shown that if we have $R_f^\top R_i \frac{f_i}{m_i} e_3 - R_f^\top R_{i,r} \frac{f_{i,r}}{m_i} e_3$ equal to the desired value for u_i from (4.10), the resulting closed-loop position tracking error dynamics are UGAS. Therefore, similar to [11], we now aim to use the inputs f_i and τ_i to achieve this virtual input u_i. Let us rewrite (4.9), to

$$R_f^\top R_i \frac{f_i}{m_i} e_3 = R_f^\top R_{i,r} \frac{f_{i,r}}{m_i} e_3 - u_i, \tag{4.17}$$

which means that we can rephrase our aim: we want to find inputs f_i and τ_i in order to obtain the vector $R_f^\top R_{i,r} \frac{f_{i,r}}{m_i} e_3 - u_i$, with the vector $R_f^\top R_i \frac{f_i}{m_i} e_3$. Notice that we are able to control R_i by the input τ_i from (4.1) in order to obtain the desired orientation, while we can obtain the magnitude of the virtual input with f_i. From (4.17), we find

$$R_i f_i e_3 = R_{i,r} \left[f_{i,r} e_3 - R_{i,r}^\top R_f m_i u_i\right]. \tag{4.18}$$

Due to the unit length of rotation matrices, we find the force magnitude, by

$$f_i = ||f_{i,r}e_3 - R_{i,r}^\top R_f m_i u_i||. \tag{4.19}$$

where u_i is given by (4.11), such that $f_i > 0$. What remains is to find the desired attitude to let $R_{i,r}^\top R_i f_i e_3$ equal $(f_{i,r}e_3 - R_{i,r}^\top R_f m_i u_i)$. However, notice that because of the geometrical properties of the propellers, the force vector $f_i e_3$ only has a component in body-fixed e_3 direction. Consequently, based on (4.18), we are free in the rotation around the thrust axis while still able to provide any virtual input u_i. Based on (4.18) and (4.19), we define the desired direction of the thrust vector, as

$$R_{i,d}e_3 = \frac{f_{i,r}e_3 - R_{i,r}^\top R_f m_i u_i}{||f_{i,r}e_3 - R_{i,r}^\top R_f m_i u_i||} := f_{i,d}, \tag{4.20}$$

where

$$f_{i,d} = \begin{bmatrix} f_{i,d1} & f_{i,d2} & f_{i,d3} \end{bmatrix}^\top. \tag{4.21}$$

Notice that (4.19) and (4.20) also indicate the necessity of saturation of the virtual input, since it shows that we need $f_i > 0$ in order to have the desired attitude $R_{i,d}$ well-defined. Since we now have only have a wanted direction for the thrust vector $f_{i,d}$, we still have not fully prescribed the position of the desired frame $\mathcal{D}_i$, since rotation around this thrust vector is still free. Let us choose to let the entire rotation happen in the plane spanned by $f_{i,d}$ and e_3, with the angle that is defined as the angle between those two vectors. This means that the rotation matrix $R_{i,d}$ that rotates the desired frame to the reference frame describes a rotation from the desired thrust vector to the thrust vector of the reference in the spanned plane. The rotation thus happens around an axis perpendicular to this plane, given by

$$n_i = \frac{f_{i,d} \times e_3}{||f_{i,d} \times e_3||}, \tag{4.22}$$

which is normalized to ensure unit length. The amount of rotation is then given by

$$\cos(\theta_{i,d}) = \frac{e_3 \cdot f_{i,d}}{||f_{i,d}||\ ||e_3||} = \begin{bmatrix} -f_{i,d2} & f_{i,d1} & 0 \end{bmatrix}^\top, \tag{4.23}$$

which with Pythagoras in the unit disc and the unit length of any column in a rotation matrix ($||f_{i,d}|| = 1$) also provides

$$\sin(\theta_{i,d}) = \sqrt{1 - \cos^2(\theta_{i,d})} = \sqrt{1 - f_{i,d3}^2}. \tag{4.24}$$

We now use (4.22), (4.23), and (4.1.3) together with Rodrigues' rotation formula, given by

$$R_{i,r} = I + (\sin\alpha_i)S(n_i) + (1 - \cos\alpha_i)S(n_i)^2, \tag{4.25}$$

in order to find $R_{i,d}$, as

$$R_{i,d} = \begin{bmatrix} 1 - \frac{f_{i,d1}^2}{1+f_{i,d3}} & -\frac{f_{i,d1}f_{i,d2}}{1+f_{i,d3}} & f_{i,d1} \\ -\frac{f_{i,d1}f_{i,d2}}{1+f_{i,d3}} & 1 - \frac{f_{i,d2}^2}{1+f_{i,d3}} & f_{i,d2} \\ -f_{i,d1} & -f_{i,d2} & f_{i,d3} \end{bmatrix} \in SO(3). \tag{4.26}$$

Thus, the rotation matrix $R_{i,d}$ rotates $f_{i,d}$ to e_3 in the plane spanned by $f_{i,d}$ to e_3, around the vector perpendicular to those two vectors $f_{i,d}$ and e_3 and with the rotation angle also defined by the angle between the two vectors $f_{i,d}$ and e_3. Since we also know that $\dot{R}_{i,d} = R_{i,d}S(\omega_{i,d})$, we find

$$\omega_{i,d} = \begin{bmatrix} -\dot{f}_{i,d2} + \frac{f_{i,d2}\dot{f}_{i,d3}}{1+f_{i,d3}} \\ \dot{f}_{i,d1} - \frac{f_{i,d1}\dot{f}_{i,d3}}{1+f_{i,d3}} \\ \frac{f_{i,d2}\dot{f}_{i,d1} - \dot{f}_{i,d2}f_{i,d1}}{1+f_{i,d3}} \end{bmatrix}. \tag{4.27}$$

Given (4.18), (4.19) and (4.26), we obtain the equality

$$\left[f_{i,r}e_3 - R_{i,r}^\top R_f m_i u_i\right] = f_i R_{i,d} e_3, \tag{4.28}$$

so in order to obtain the goal of having $R_{i,r}^\top R_i f_i e_3$ equal to $(f_{i,r}e_3 - R_{i,r}^\top R_f m_i u_i)$, what remains is to find inputs τ_i to let $R_{i,d} f_i e_3$ converge to $R_{i,r}^\top R_i f_i e_3$. In order to do so, let us define the attitude tracking error

$$R_{i,e} = R_{i,d}^\top (R_{i,r}^\top R_i), \tag{4.29a}$$

with corresponding angular velocity error

$$\omega_{i,e} = \omega_i - R_i^\top R_{i,r}\omega_{i,r} - (R_i^\top R_{i,r})R_{i,d}\omega_{i,d}. \tag{4.29b}$$

Note that $(R_i^\top R_{i,r})R_{i,d} = (R_{i,d}^\top(R_{i,r}^\top R_i))^\top = R_{i,e}^\top$. The angular velocity error is expressed in the body fixed reference frame $\mathcal{B}_i$ of robot i, as that is convenient for controller design, because that means that the input τ_i, which by definition also aligns with $\mathcal{B}_i$, appears directly in the error dynamics. The terms $R_i^\top R_{i,r}$ and $(R_i^\top R_{i,r})R_{i,d}$ from (4.29b), respectively, are used to transform the angular velocity of the reference frame $\mathcal{R}_i$ and the desired frame $\mathcal{D}_i$ from their own frame to the body-fixed frame $\mathcal{B}_i$. Differentiating (4.29a) and (4.29b) along their solutions, yields

$$\begin{aligned} \dot{R}_{i,e} &= R_{i,e} S(\omega_{i,e}), && (4.30a) \\ J_i\dot{\omega}_{i,e} &= S(J_i\omega_i)\omega_i + \tau_i + S(\omega_i)R_i^\top R_{i,r}\omega_{i,r} - R_i^\top R_{i,r} S(\omega_{i,r})\omega_{i,r} \dots && (4.30b) \\ &\quad - R_i^\top R_{i,r} J^{-1}\left[S(J_i\omega_{i,r})\omega_{i,r} + \tau_{i,r}\right] + S(\omega_{i,e})R_{i,e}^\top \omega_{i,d} - R_{i,e}^\top \dot{\omega}_{i,d}. \end{aligned}$$

Let us now state the attitude tracking control problem.

Problem 4.1.2. (Attitude tracking problem) *Consider the attitude tracking dynamics* (4.30)*, resulting from the attitude tracking errors as defined in* (4.29)*. The desired equilibrium of the system is given as* $(I, 0) \in SO(3) \times \mathbb{R}^3$*. Find an appropriate control law*

$$\tau_i = \tau_i(t, R_{i,e}, \omega_{i,e}), \tag{4.31}$$

such that the resulting closed-loop system (4.30)*,* (4.31)*, yields*

$$\lim_{t\to\infty} R_{i,e} = I, \qquad \textit{and} \qquad \lim_{t\to\infty}\omega_{i,e} = 0.$$

let us immediately propose a possible solution based on the known stability results, included in Theorem 2.2.3. More specifically, let us follow [11,19], based on [30,32,52].

Proposition 4.1.2. *Consider the attitude tracking dynamics* (4.30)*, resulting from the attitude tracking errors as defined in* (4.29) *and* $\omega_{i,d}$ *as defined in* (4.27)*. Choosing the control law*

$$\begin{aligned} \tau_i = -K_{i,\omega}\omega_{i,e} + K_{i,R}\sum_{s=1}^{3} k_s(e_s \times R_{i,e}^\top e_s) - S(J_i\omega_i)\omega_i - S(\omega_i)R_i^\top R_{i,r}\omega_{i,r} \dots \qquad (4.32) \\ + R_i^\top R_{i,r}S(\omega_{i,r})\omega_{i,r} + R_i^\top R_{i,r}J^{-1}\left[S(J_i\omega_{i,r})\omega_{i,r} + \tau_{i,r}\right] - S(\omega_{i,e})R_{i,e}^\top\omega_{i,d} + R_{i,e}^\top\dot{\omega}_{i,d}, \end{aligned}$$

with $K_{i,\omega} = K_{i,\omega}^\top > 0$ *and* $K_{i,R} = K_{i,R}^\top > 0$*, provides a closed-loop format that is equivalent to [11]. The closed-loop system* (4.30) *with* (4.32)*, yields*

$$\dot{R}_{i,e} = R_{i,e}S(\omega_{i,e}), \tag{4.33a}$$

$$J_i\dot{\omega}_{i,e} = -K_{i,\omega}\omega_{i,e} + K_{i,R}\sum_{s=1}^{3} k_{i,s}(e_s \times R_{i,e}^\top e_s), \tag{4.33b}$$

for which $(I, 0)$ *is ULES and UaGAS for distinct* $k_{i,s}$ *with* $s = 1, 2, 3$ *when* $K_{i,\omega} = K_{i,\omega}^\top > 0$ *and* $K_{i,R} = K_{i,R}^\top > 0$*, cf. [30,32]; included in Theorem 2.2.3 including proof.*

In order to provide a full solution to Problem 4.1.1, however, we can not simply look at the position tracking subsystem and attitude tracking subsystem separately. The controlled attitude is namely required to obtain the virtual input for the position tracking subsystem and achieved asymptotically. This means that the virtual input is not achieved at all times, since the desired attitude is asymptotically achieved rather than identically the same as the actual attitude. We can see the two subsystems as one cascaded system, since the solution of the attitude tracking subsystem thus perturbs the position tracking subsystem. In order to analyze the stability of the cascaded system, we execute a cascade system analysis.

4.1.4 Cascade system analysis

In the previous sections, we have designed a virtual control action u_i for the position tracking error dynamics of a robot i and a controller for f_i and τ_i to asymptotically achieve this virtual control action u_i. What remains is to analyze the stability of the attitude and position subsystems in cascade, similar to [11]. Consider the dynamics of the quadrotor UAV (4.1) and its virtual reference quadrotor UAV (4.2) in closed-loop with inputs (4.11), (4.19) and (4.32). The cascaded closed-loop system in generalized coordinates then equals

$$\dot{\rho}_{i,e} = -S(\omega_f)\rho_{i,e} + \nu_{i,e}, \tag{4.34a}$$

$$\dot{\nu}_{i,e} = -S(\omega_f)\nu_{i,e} - k_{i,\rho}\sigma(\rho_{i,e}) - K_{i,\nu}\sigma(\nu_{i,e}) + \frac{f_i}{m_i}R_{i,r}^\top R_i(I - R_{i,e}^\top)e_3, \tag{4.34b}$$

$$\dot{R}_{i,e} = R_{i,e}S(\omega_{i,e}), \tag{4.34c}$$

$$J_i\dot{\omega}_{i,e} = -K_{i,\omega}\omega_{i,e} + K_{i,R}\sum_{s=1}^{3} k_{i,s}(e_s \times R_{i,e}^\top e_s), \tag{4.34d}$$

in which the cascaded term $\frac{f_i}{m_i}R_{i,r}^\top R_i(I - R_{i,e}^\top)e_3 = \frac{f_i}{m_i}(R_{i,r}^\top R_i - R_{i,d})e_3$ is as a disturbance on the position tracking system that results from the fact that the attitude error is not fully converged yet; due to the remaining attitude error, the virtual input u_i is basically not yet entirely obtained by f_i and τ_i.

If we are able to conclude

$$\lim_{t\to\infty} \rho_{i,e} = 0, \qquad \lim_{t\to\infty} \nu_{i,e} = 0, \qquad \lim_{t\to\infty} R_{i,e} = I, \quad \text{and} \quad \lim_{t\to\infty} \omega_{i,e} = 0, \tag{4.35}$$

the tracking control problem from Problem 4.1.1 is solved, since if (4.35) holds, the conditions from Problem 4.1.1 hold automatically. The result is presented as follows.

Proposition 4.1.3. *Consider the closed-loop cascaded system 4.34. The solutions of the closed-loop cascaded system* (4.34) *are uniformly almost-globally asymptotically stable. The solutions of* (4.34) *asymptotically converge to the origin for all* $\rho_{i,e} \in \mathbb{R}^3$, $\nu_{i,e} \in \mathbb{R}^3$ *and* $(R_{i,e}, \omega_{i,e}) \in G$ *with* $G \subset SO(3) \times \mathbb{R}^3$ *and* $\mathcal{M} = (SO(3) \times \mathbb{R}^3) \setminus G$ *a set with measure zero.*

Proof. Identify the cascaded system ((4.34a),(4.34b)) and ((4.34c),(4.34d)). We know that the attitude tracking subsystem ((4.34c),(4.34d)) is UaGAS, so for the cascaded system, we consider the stability analysis on $\mathbb{R}^6 \times G$, where $G \subset SO(3) \times \mathbb{R}^3$ denotes the almost global region of attraction. By only considering this region, we are able to use Theorem 2.2.4 in order to prove the cascaded system stable. Then, from Section 4.1.2 we already know that the position reference tracking subsystem in its unperturbed form (assume $R_{i,e} = I$ in ((4.34a),(4.34b)) to obtain (4.12)) is UGAS. What remains is to show that $\rho_{i,e}$ and $\nu_{i,e}$ remain bounded, with the dynamics from ((4.34a),(4.34b)). Consider

$$V = k_{i,\rho}V_\sigma + \frac{1}{2}\nu_{i,e}^\top\nu_{i,e}, \tag{4.36}$$

so that differentiating yields

$$\dot{V}(\rho_{i,e}, \nu_{i,e}) = k_{i,\rho}\dot{\rho}_{i,e}^{\top}\sigma(\rho_{i,e}) + \nu_{i,e}^{\top}\dot{\nu}_{i,e}, \tag{4.37a}$$

$$= \left[-S(\omega_f)\rho_{i,e} + \nu_{i,e}\right]^{\top} k_{i,\rho}\sigma(\rho_{i,e}) \dots \tag{4.37b}$$

$$+ \nu_{i,e}^{\top}\left[-S(\omega_f)\nu_{i,e} - k_{i,\rho}\sigma(\rho_{i,e}) - K_{i,\nu}\sigma(\nu_{i,e}) + \frac{f_i}{m_i}R_{i,r}^{\top}R_i(I - R_{i,e}^{\top})e_3\right],$$

$$= -\nu_{i,e}^{\top}K_{i,\nu}\sigma(\nu_{i,e}) + \nu_{i,e}^{\top}\frac{f_i}{m_i}R_{i,r}^{\top}R_i(I - R_{i,e}^{\top})e_3, \tag{4.37c}$$

which results in

$$\dot{V}(\rho_{i,e}, \nu_{i,e}) \leq \nu_{i,e}^{\top}\frac{f_i}{m_i}R_{i,r}^{\top}R_i(I - R_{i,e}^{\top})e_3 \leq c_1\sqrt{V(\rho_{i,e}, \nu_{i,e})}||I - R_{i,e}||. \tag{4.38}$$

Resulting from the fact that ((4.34c),(4.34d)) is ULES, as in [11], we have

$$\sqrt{V(\rho_{i,e}(t), \nu_{i,e}(t))} - \sqrt{V(\rho_{i,e}(t_0), \nu_{i,e}(t_0))} \leq c_2(t_0), \tag{4.39}$$

which means that $V(\rho_{i,e}, \nu_{i,e})$ is bounded, so $\rho_{i,e}$ and $\nu_{i,e}$ are bounded. From Theorem 2.2.4 follows: the closed-loop cascaded system (4.34) is ULES and UaGAS. Therefore, the dynamics (4.1) and (4.2) in closed-loop with inputs (4.11), (4.19) and (4.32) are ULES and UaGAS. □

Different from previous results, as [11, 19], the provided solution to the reference tracking problem from Problem 4.1.1 is based on generalized coordinates that are expressed in a mutually known frame. Furthermore, by introducing a notation that allows for identifying a specific agent, by utilizing an index i, we can stack the dynamics of these individuals and scale to any number of agents in the system. In fact, if we are able to provide a set of n feasible reference trajectories that together form a virtual formation in time, we can use the agents with $i = 1, \dots n$ to track this virtual formation. However, without any interconnection, we are unable to provide any guarantee on the formation tracking behavior, since the agents are still unaware of each other, which prohibits them from reacting to disturbances on the others. Therefore, we should implement some coupling between the agents in order to have control over the consideration between tracking of the individual reference trajectory and staying (or going) in formation.

4.2 Formation control

In this section, we first introduce a virtual formation reference structure, similar to Section 3.1.3 (e.g., as in [15]) but now in 3D instead of 2D, in order to systematically obtain a set of n feasible reference trajectories for n quadrotor UAVs to form a formation. Then, we want to implement coupling functions on the individual systems to enable synchronized operation. At the end, we want to analyze the stability and test the behavior of the system.

4.2.1 A spatial virtual reference structure for tracking

In order to systemetically obtain a set of n feasible reference trajectories, we choose to follow a similar procedure as in [13, 15], since there the graph structure relative to a virtual formation centered coordinate frame is defined, which also provides a mutually known frame in order to later on relate the individual generalized coordinates for mutual coupling.

Let us introduce the formation centered frame $\mathcal{F}$, located at a fixed but free to choose virtual center of the formation. Let the position of this frame over time be denoted by the time dependent vector

$$\rho_f(t) = (x_f(t),\ y_f(t),\ z_f(t))^{\top} \in \mathbb{R}^3, \tag{4.40a}$$

relative to the inertial frame $\mathcal{I}$, and the attitude of this frame $\mathcal{F}$ relative to $\mathcal{I}$ by the rotation matrix $R_f(t) \in SO(3)$. Together, the position $\rho_f(t)$ and attitude $R_f(t)$ fully define the frame $\mathcal{F}$. Notice that we could choose $R_f(t) = I$ or $R_f(\rho_f(t))$ in order to always find a suitable attitude, something that we will do for the attitude of the quadrotor UAV, but for now we abstractly assume $R_f(t)$ available. Furthermore, assume a set of possibly time varying formation shape vectors $p_i(i) = (p_{i,x}(t),\ p_{i,y}(t),\ p_{i,z}(t))^\top \in \mathbb{R}^3$ with $i = 1, \ldots, n$ that describes the positions of n quadrotor UAVs relative to the virtual formation center $\mathcal{F}$ available. Subsequently, let us consider the reference position of vehicle i, as

$$\rho_{i,r} = \rho_f + R_f p_i, \tag{4.41}$$

which is thus a composition between the path of the formation relative to the inertial frame and the path of agent i relative to the virtual formation center, combined and expressed in the inertial frame; this is similar to for example [15], but in 3D instead of 2D. Now, we need feasibility of the reference trajectories (4.41), i.e., each reference trajectory i has to fulfill the dynamics (4.2) [22]. Assume that formation trajectory $(\rho_f,\ R_f)$ and the formation shape vectors p_i are provided, as these are the three parameters we need to have or choose in order to obtain the reference i. We can then assume for quadrotor i that $\rho_{i,r}(t)$ is given. Following from Newton's second law, we know $\ddot{\rho}_{i,r}(t) = ge_3 - \frac{f_{i,r}}{m_i} R_{i,r} e_3$, so we can find $f_{i,r}$ by assuming $f_{i,r} > 0$, as

$$f_{i,r} = ||f_{i,r} R_{i,r} e_3|| = m||ge_3 - \ddot{\rho}_{i,r}|| > 0, \tag{4.42}$$

satisfying in which we have used that $||R_{i,r}e_3|| = 1$, following from Section 2.1. The normalized direction of this thrust vector can then be found, which is because of the geometrical lay-out equal to the direction $R_{i,r}e_3$, as

$$R_{i,r} e_3 = \frac{ge_3 - \ddot{\rho}_{i,r}}{||ge_3 - \ddot{\rho}_{i,r}||} = \begin{bmatrix} r_{i,r1} \\ r_{i,r2} \\ r_{i,r3} \end{bmatrix}. \tag{4.43}$$

Notice that this also means that rotation around this thrust vector, i.e., the yaw angle, is still free to choose. This follows from the fact that we are able to obtain any vector $R_{i,r}e_3$ with just the roll and pitch freedom. In order to always obtain a reference attitude $R_{i,r}$ we have to either prescribe a possibly time varying yaw angle $\psi_i(t)$ or choose a relation for the yaw angle as function of other system parameters. Similar to [11], let us choose the yaw angle ψ_i such that the resulting rotation matrix $R_{i,r}$ rotates the thrust vector e_3 to $R_{i,r}e_3$ in the spanned plane. By choosing this rotation plane for the thrust vector, we have a single rotation (namely between the two vectors $R_{i,r}e_3$ and e_3) available to rotate back to hovering mode. In order to rotate the thrust vector in the plane spanned by e_3 and $R_{i,r}e_3$, we define the axis around which we rotate as the vector perpendicular to this plane [28, Theorem 1.2.1] [25], as

$$k_i = e_3 \times R_{i,r} e_3 = \begin{bmatrix} -r_{i,r2} & r_{i,r1} & 0 \end{bmatrix}^\top. \tag{4.44}$$

In order to find the rotation matrix defined by a rotation between two vectors in the spanned plane, consider Rodrigues' rotation formula (4.25), which describes a rotation of α_i counterclockwise around the normalized rotation axis n_i. Since consider two vectors e_3 and $R_{i,r}e_3$ that not only define the rotation axis, but also the rotation angle, we now have enough information to fully define the rotation matrix $R_{i,r}$ from (4.25). The normalized rotation vector can now be found equal to

$$n_i = \frac{k_i}{||k_i||} = \frac{1}{\sqrt{r_{i,r1}^2 + r_{i,r2}^2}} \begin{bmatrix} -r_{i,r2} \\ r_{i,r1} \\ 0 \end{bmatrix} \quad \text{for} \quad ||k_i|| > 0, \tag{4.45}$$

and notice that $\sqrt{r_{i,r1}^2 + r_{i,r2}^2 + r_{i,r3}^2} = 1$. Furthermore, notice that $||k_i|| = 0$ corresponds with the scenario that e_3 and $R_{i,r}e_3$ are aligned, which also means that there is no rotation and thus no rotation axis. The rotation angle α_i between the two vectors is e_3 and $R_{i,r}e_3$ obtained by

$$\cos\alpha_i = \frac{e_3 \bullet R_{i,r}e_3}{||e_3||\ ||R_{i,r}e_3||} = r_{i,r3}, \tag{4.46}$$

and by using Pythagoras in the unit disc, we find

$$\sin\alpha_i = \sqrt{1-\cos^2\alpha_i} = \sqrt{1-r_{i,r3}^2}. \tag{4.47}$$

By using (4.25), (4.45), (4.46), and (4.47), we obtain

$$R_{i,r} = I + \sqrt{1-r_{i,r3}^2}S(n_i) + (1-r_{i,r3})S(n_i)^2 \tag{4.48}$$

which equals

$$R_{i,r} = \begin{bmatrix} 1-\frac{r_{i,r1}^2}{1+r_{i,r3}} & -\frac{r_{i,r1}r_{i,r2}}{1+r_{i,r3}} & r_{i,r1} \\ -\frac{r_{i,r1}r_{i,r2}}{1+r_{i,r3}} & 1-\frac{r_{i,r2}^2}{1+r_{i,r3}} & r_{i,r2} \\ -r_{i,r1} & -r_{i,r2} & r_{i,r3} \end{bmatrix} \in SO(3). \tag{4.49}$$

Figure 4.2 shows an example of the frames of $\mathcal{I}$ and the $\mathcal{R}_i$ with the rotation plane spanned by e_3 and $R_{i,r}e_3$ and rotation axis n_i indicated. Notice that having e_3 and $R_{i,r}e_3$ parallel thus results in $\alpha_i = 0$ and $R_{i,r3} = 1$, which provides $R_{i,r} = I$.

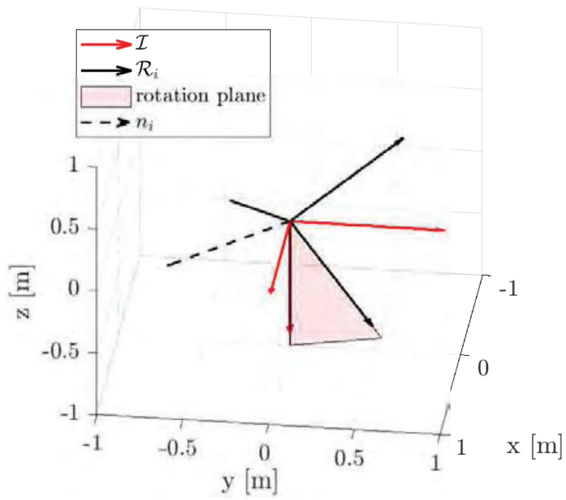


Figure 4.2: Rotation example of the frame $\mathcal{R}_i$ by rotation matrix $R_{i,r}$ from (4.49), relative to the inertial frame $\mathcal{I}$. The plane and normalized axis n_i perpendicular to this plane are indicated. As an example, the direction of the reference thrust was chosen as $R_{i,r}e_3 = \frac{1}{\sqrt{3}}[1\ 1\ 1]^\top$.

Subsequently, using the dynamics (4.2), we find

$$\nu_{i,r} = R_{i,r}^\top \dot{\rho}_{i,r}, \qquad S(\omega_{i,r}) = R_{i,r}^\top \dot{R}_{i,r}, \qquad \tau_{i,r} = J\dot{\omega}_{i,r} - S(J\omega_{i,r})\omega_{i,r}, \tag{4.50}$$

so by using the definition for $S(\omega_{i,r})$ from Theorem 2.1.1, we obtain [22]

$$\omega_{i,r} = \begin{bmatrix} -\dot{r}_{i,r2} + \frac{r_{i,r2}\dot{r}_{i,r3}}{1+r_{i,r3}} \\ \dot{r}_{i,r1} - \frac{r_{i,r1}\dot{r}_{i,r3}}{1+r_{i,r3}} \\ \frac{r_{i,r2}\dot{r}_{i,r1} - \dot{r}_{i,r2}r_{i,r1}}{1+r_{i,r3}} \end{bmatrix}. \tag{4.51}$$

In order to obtain $\tau_{i,r}$, we also need

$$\dot{\omega}_{i,r} = \frac{\mathrm{d}}{\mathrm{d}t}\omega_{i,r} = \begin{bmatrix} -\ddot{r}_{i,r2} + \frac{\dot{r}_{i,r2}\dot{r}_{i,r3}+r_{i,r2}\ddot{r}_{i,r3}}{1+r_{i,r3}} - \frac{\dot{r}_{i,r3}^2 r_{i,r2}}{(1+r_{i,r3})^2} \\ \ddot{r}_{i,r1} - \frac{\dot{r}_{i,r1}\dot{r}_{i,r3}+r_{i,r1}\ddot{r}_{i,r3}}{1+r_{i,r3}} + \frac{\dot{r}_{i,r3}^2 r_{i,r1}}{(1+r_{i,r3})^2} \\ \frac{r_{i,r2}\ddot{r}_{i,r1}-\ddot{r}_{i,r2}r_{i,r1}}{1+r_{i,r3}} - \dot{r}_{i,r3}\frac{r_{i,r2}\dot{r}_{i,r1}-\dot{r}_{i,r2}r_{i,r1}}{(1+r_{i,r3})^2} \end{bmatrix}. \tag{4.52}$$

Notice that in order to compute $\ddot{R}_{i,r}$, we need $\ddot{R}_{i,r}e_3$. For completeness, let us compute $\ddot{R}_{i,r}e_3$, by first defining

$$f_{i,r}^* = m(ge_3 - \ddot{\rho}_{i,r}), \tag{4.53}$$

so that the first and second order time derivatives of $R_{i,r}e_3$, yield

$$\dot{R}_{i,r}e_3 = \begin{bmatrix} \dot{r}_{i,r1} \\ \dot{r}_{i,r2} \\ \dot{r}_{i,r3} \end{bmatrix} = \frac{\dot{f}_{i,r}^*}{f_{i,r}} - \frac{\dot{f}_{i,r}f_{i,r}^*}{f_{i,r}^2}, \tag{4.54a}$$

$$\ddot{R}_{i,r}e_3 = \begin{bmatrix} \ddot{r}_{i,r1} \\ \ddot{r}_{i,r2} \\ \ddot{r}_{i,r3} \end{bmatrix} = \frac{\ddot{f}_{i,r}^*}{f_{i,r}} - \frac{\dot{f}_{i,r}\dot{f}_{i,r}^*}{f_{i,r}} - \frac{\ddot{f}_{i,r}f_{i,r}^* + \dot{f}_{i,r}\dot{f}_{i,r}^*}{f_{i,r}^2} + \frac{2f_{i,r}\dot{f}_{i,r}^2 f_{i,r}^*}{f_{i,r}^3}, \tag{4.54b}$$

in which

$$\dot{f}_{i,r}^* = -m_i \dddot{\rho}_{i,r}, \tag{4.55a}$$

$$\ddot{f}_{i,r}^* = -m_i \rho_{i,r}^{(4)}, \tag{4.55b}$$

$$f_{i,r} = ||f_{i,r}^*|| = \sqrt{(f_{i,r}^*)^\top (f_{i,r}^*)}, \tag{4.55c}$$

$$\dot{f}_{i,r} = \frac{2(f_{i,r}^*)^\top (\dot{f}_{i,r}^*)}{||f_{i,r}^*||}, \tag{4.55d}$$

$$\ddot{f}_{i,r} = \frac{2(\dot{f}_{i,r}^*)^\top (\dot{f}_{i,r}^* + 2(f_{i,r}^*)^\top (\ddot{f}_{i,r}^*)}{||f_{i,r}^*||} - \frac{4((f_{i,r}^*)^\top (\dot{f}_{i,r}^*))^2}{||f_{i,r}^*|| f_{i,r}^*}. \tag{4.55e}$$

Altogether, since we want to find feasible reference trajectories by only prescribing $\rho_{i,r}(t)$ with (4.40), we obtain all required system dynamics if we have existence of $\rho_{i,r}^{(4)}$, since then the entire reference system (4.42) till (4.55) can be expressed in time derivatives of $\rho_{i,r}(t)$. Thus, the time varying reference position $\rho_{i,r}(t)$ needs to be four times differentiable. Differentiating $\rho_{i,r}$ from (4.41) four times, yields

$$\dot{\rho}_{i,r} = \dot{\rho}_f + \dot{R}_f p_i + R_f \dot{p}_i, \tag{4.56a}$$

$$\ddot{\rho}_{i,r} = \ddot{\rho}_f + \ddot{R}_f p_i + 2\dot{R}_f \dot{p}_i + R_f \ddot{p}_i, \tag{4.56b}$$

$$\dddot{\rho}_{i,r} = \dddot{\rho}_f + \dddot{R}_f p_i + 3\ddot{R}_f \dot{p}_i + 3\dot{R}_f \ddot{p}_i + R_f \dddot{p}_i, \tag{4.56c}$$

$$\rho_{i,r}^{(4)} = \rho_f^{(4)} + R_f^{(4)} p_i + 4\dddot{R}_f \dot{p}_i + 6\ddot{R}_f \ddot{p}_i + 4\dot{R}_f \dddot{p}_i + R_f p_i^{(4)}. \tag{4.56d}$$

Concluding, in this section we defined a virtual structure that acts as the reference formation for n quadrotor UAVs to track. In order to have all reference trajectories i with $i = 1, \ldots, n$ feasible, according to (4.42) till (4.55), the prescribed reference position $\rho_{i,r}(t)$ over time needs to be four times differentiable. Since we compose this reference trajectory by means of a virtual structure approach, as defined in (4.41), we need to have ρ_f, p_i, and R_f, four times differentiable, following from (4.56d).

4.2.2 Formation tracking with a fleet of quadrotor UAVs

In the previous sections, we have provided a tracking controller for an individual agent and a method to obtain a set of feasible reference trajectories for a set of agents. However, if we take the previously defined tracking controllers (4.11), (4.19) and (4.32) for all of the individuals, we still have no control over the consideration between tracking the individual reference trajectories and staying in formation. Therefore, we want to find a coupling method to couple the tracking dynamics of the agents in order to enable synchronized behavior [50]. Since the reference structure and current controllers already allow a scalability to a high number of agents, we choose to strive for mutual coupling between the agents. Another method would be master-slave coupling [16, 53], but in our opinion this only provides a partly solution to the problem of synchronizing multiple agents. The solution is considered as partly since the master-slave synchronization can still fail as a result of the master being down or the slave being unable to catch up, both as a result of the hierarchical structure of the system; if the master is down the slave cannot react as it is only provided the task to follow the master and if the slave is down the master never knows. Therefore, we want to provide all agents with their own reference trajectory and mutually implement coupling, so that each of the agents makes a consideration between mutual synchronization and following the own reference trajectory. Several coupling methods exist for mutual coupling, like coupling by altering the reference trajectories [17, 18], directly implementing coupling in the control law [15] or by considering coupling terms in the generalized coordinates [13]. All of these methods require to define some kind of error between any pair of coupled agents, but they differ in where this error is incorporated in order to cause synchronization. Furthermore, since we aim to use a similar cascaded approach as before, we use the attitude tracking subsystem in order to obtain the virtual input. Therefore, we choose to only implement coupling on the position tracking subsystem, to both leave the attitude tracking subsystem entirely in tact, as well as to not compromise the position tracking ability. Adding coupling between agents on the attitude tracking, namely, would act as an additional disturbance to the ability to point the thrust in the desired direction, which compromises the position tracking performance.

The simplest alternative would be to directly add the coupling functions in the control law u_i, as in [15], so that we obtain the control law

$$u_i = -k_{i,\rho}\rho_{i,e} - K_{i,v}v_{i,e} - \sum_{j\in N_i}\tilde{k}_{i,\rho}(\rho_{j,e} - \rho_{i,e}) - \sum_{j\in N_i}\tilde{K}_{i,v}(v_{j,e} - v_{i,e}), \tag{4.57}$$

providing the closed-loop dynamics for the entire system at once, as

$$\dot{\bar{\rho}}_e = -G_{S(\omega_f)}\bar{\rho}_e + \bar{\nu}_e, \tag{4.58a}$$
$$\dot{\bar{\nu}}_e = -G_{S(\omega_f)}\bar{\nu}_e - \bar{K}_\rho\bar{\rho}_e - \bar{K}_\nu\bar{\nu}_e, \tag{4.58b}$$

in which we use the stacked system parameters

$$\bar{\rho}_e = \begin{bmatrix}\rho_{1,e}\\ \vdots \\ \rho_{n,e}\end{bmatrix}, \qquad \bar{v}_e = \begin{bmatrix}v_{1,e}\\ \vdots \\ v_{n,e}\end{bmatrix}, \quad \text{and} \quad G_{S\omega_f} = \mathrm{diag}(S(\omega_f), \ldots, S(\omega_f)), \tag{4.59}$$

and the coupling matrices

$$\bar{K}_\rho = \begin{bmatrix} K_{1,\rho} - \sum_{j\in N_1} \tilde{K}_{1j,\rho} & \tilde{K}_{12,\rho} & \cdots & \tilde{K}_{1n,\rho} \\ & K_{2,\rho} - \sum_{j\in N_2} \tilde{K}_{2j,\rho} & & \vdots \\ \vdots & & \ddots & \\ \tilde{K}_{n1,\rho} & \cdots & \tilde{K}_{n(n-1),\rho} & K_{n,\rho} - \sum_{j\in N_n} \tilde{K}_{n(n-1),\rho} \end{bmatrix}, \tag{4.60}$$

$$\bar{K}_v = \begin{bmatrix} K_{1,v} - \sum_{j\in N_1} \tilde{K}_{1j,v} & \tilde{K}_{12,v} & \cdots & \tilde{K}_{1n,v} \\ & K_{2,v} - \sum_{j\in N_2} \tilde{K}_{2j,v} & & \vdots \\ \vdots & & \ddots & \\ \tilde{K}_{n1,v} & \cdots & \tilde{K}_{n(n-1),v} & K_{n,v} - \sum_{j\in N_n} \tilde{K}_{n(n-1),} \end{bmatrix}, \tag{4.61}$$

with $K_{i,\rho} = k_{i,\rho} I$ and $\tilde{K}_{ij,\rho} = \tilde{k}_{ij,\rho} I$. Notice that as a result of the assumptions $\tilde{k}_{ij,\rho} > 0$, $\tilde{K}_{ij,\nu} = \tilde{K}_{ij,\nu}^\top > 0$, $\tilde{k}_{ij,\rho} = \tilde{k}_{ji,\rho}$, and $\tilde{K}_{ij,\nu} = \tilde{K}_{ji,\nu}$, we have $\bar{K}_\rho = \bar{K}_\rho^\top > 0$ and $\bar{K}_v = \bar{K}_v^\top > 0$. The closed-loop system (4.58) is very similar to the closed-loop system (4.12) for a single quadrotor. However, one very important difference is that the gain on the position error now is a matrix $\bar{K}_\rho$ instead of a scalar $k_{i,\rho}$, which makes it impossible to utilize $k_{i,\rho}\rho_{i,e}^\top S(\omega_f)\rho_{i,e} = 0$, as in (4.14), since we now have $\bar{\rho}_e^\top G_{S(\omega_f)} \bar{K}_{i,\rho}\bar{\rho}_e \neq 0$. This means that we can not follow the Lyapunov proof from [11]. Therefore, instead of directly including the coupling terms in the control law, let us define a new set of errors inspired by [18]

$$\tilde{\rho}_{i,e} = R_f^\top \left[(\rho_{i,r} - \rho_i) + \sum_{j\in N_i} \tilde{k}_{ij,\rho}((\rho_{j,r} - \rho_j) - (\rho_{i,r} - \rho_i)) \right], \tag{4.62a}$$

$$\tilde{\nu}_{i,e} = R_f^\top \left[(R_{i,r}\nu_{i,r} - R_i\nu_i) + \sum_{j\in N_i} \tilde{k}_{ij,\nu}((R_{j,r}\nu_{j,r} - R_j\nu_j) - (R_{i,r}\nu_{i,r} - R_i\nu_i)) \right], \tag{4.62b}$$

in which $\tilde{k}_{ij,\rho} > 0$ and $\tilde{k}_{ij,\nu} > 0$ are coupling gains that can be seen as cost related to the coupling error, expressed in the inertial frame. This choice was made deliberately so that every term has to be multiplied by $R_f^\top$ in order to be expressed in the formation frame eventually, which provides the exact same dynamics as before

$$\dot{\tilde{\rho}}_{i,e} = -S(\omega_f)\tilde{\rho}_{i,e} + \tilde{\nu}_{i,e}, \tag{4.63a}$$

$$\dot{\tilde{\nu}}_{i,e} = -S(\omega_f)\tilde{\nu}_{i,e} + \tilde{u}_i, \tag{4.63b}$$

with $\tilde{u}_i$ a virtual input that equals

$$\begin{aligned} \tilde{u}_i = R_f^\top \Bigg[& (R_{i,r}\{gR_{i,r}^\top e_3 - \frac{f_{i,r}}{m_i} e_3\} - R_i\{gR_i^\top e_3 - \frac{f_i}{m_i} e_3\}) \ldots \\ & + \sum_{j\in N_i} \tilde{k}_{ij,\nu}((R_{j,r}\{gR_{j,r}^\top e_3 - \frac{f_{j,r}}{m_j} e_3\} - R_j\{gR_j^\top e_3 - \frac{f_j}{m_j} e_3\}) \ldots \\ & - (R_{i,r}\{gR_{i,r}^\top e_3 - \frac{f_{i,r}}{m_i} e_3\} - R_i\{gR_i^\top e_3 - \frac{f_i}{m_i} e_3\})) \Bigg]. \end{aligned} \tag{4.64}$$

because of the fact that the gains are expressed in the inertial frame and partly for control purposes, we have chosen the gains $\tilde{k}_{ij,\rho}$ and $\tilde{k}_{ij,\nu}$ scalar so that they work in every direction equally. In our opinion this choice seems reasonable since first of all quadrotor UAVs are usually axisymmetric and we do not necessarily need to have different coupling strengths in different directions. The preferred alternative is to define the error coordinates with all gains expressed in the formation frame, as

$$\tilde{\rho}_{i,e} = \rho_{i,e} + \sum_{j \in N_i} \tilde{k}_{ij,\rho}(\rho_{j,e} - \rho_{i,e}), \tag{4.65a}$$

$$\tilde{\nu}_{i,e} = \nu_{i,e} + \sum_{j \in N_i} \tilde{k}_{ij,\nu}(\nu_{j,e} - \nu_{i,e}), \tag{4.65b}$$

with $\rho_{i,e}$ and $\nu_{i,e}$ as defined in (4.7), so expressed in $\mathcal{F}$. However, these error definitions provide a closed-loop system that we can not directly relate to the original format (4.12) in order to stabilize. Therefore, for now we allow a scalar coupling strength and choose to use the definitions (4.62), which provides similar generalized dynamics similar to the uncoupled generalized dynamics. This known format results from the fact that coupling cost is implemented in the inertial frame and premultiplied with $R_f^\top$, rotating the entire term from the inertial frame to the formation frame. We allow this because this gain only influences the relative cost between the reference tracking error and coupling error and is unidirectional. Since the cost is a scalar it works on all directions equally and we can simply move it to the front in (4.63) and look at it as expressed in any frame equally. Furthermore, we are still free to choose additional tracking gains in $\tilde{u}_i$ to influence the overall convergence of the generalized coordinates (4.62). Notice that stabilizing (4.63) also provides a solution to Problem 4.1.1, since the position tracking error definitions from Problem 4.1.1 are always identically zero if (4.62) is at the origin.

Considering the generalized dynamics (4.63) with generalized coordinates (4.62). By applying Proposition 4.1.1, choosing the virtual input

$$\tilde{u}_i = -k_{i,\rho}\sigma(\tilde{\rho}_{i,e}) - K_{i,\nu}\sigma(\tilde{\nu}_{i,e}), \tag{4.66}$$

yields the closed-loop system

$$\dot{\tilde{\rho}}_{i,e} = -S(\omega_f)\tilde{\rho}_{i,e} + \tilde{\nu}_{i,e}, \tag{4.67a}$$

$$\dot{\tilde{\nu}}_{i,e} = -S(\omega_f)\tilde{\nu}_{i,e} - k_{i,\rho}\sigma(\tilde{\rho}_{i,e}) - K_{i,\nu}\sigma(\tilde{\nu}_{i,e}), \tag{4.67b}$$

which is UGAS, as proven in Section 4.1.2.

4.2.3 Attitude control

Following from the previous section, we know that we are able to stabilize the dynamics with a virtual input $\tilde{u}_i$. However, we need to find a method to achieve the virtual input $\tilde{u}_i$, by means of the actual inputs f_i and τ_i. Rewriting (4.64) provides

$$\begin{aligned}\tilde{u}_i = R_f^\top \Bigg[&(R_{i,r}\{-\frac{f_{i,r}}{m_i}e_3\} - R_i\{-\frac{f_i}{m_i}e_3\}) \dots \\ &+ \sum_{j \in N_i} \tilde{k}_{ij,\nu}((R_{j,r}\{-\frac{f_{j,r}}{m_j}e_3\} - R_j\{-\frac{f_j}{m_j}e_3\}) \dots \\ &\qquad - (R_{i,r}\{-\frac{f_{i,r}}{m_i}e_3\} - R_i\{-\frac{f_i}{m_i}e_3\}))\Bigg],\end{aligned} \tag{4.68}$$

$$\begin{aligned}\tilde{u}_i = R_f^\top \Bigg[&(R_i\frac{f_i}{m_i}e_3 - R_{i,r}\frac{f_{i,r}}{m_i}e_3) \dots \\ &+ \sum_{j \in N_i} \tilde{k}_{ij,\nu}\Big((R_j\frac{f_j}{m_j}e_3 - R_{j,r}\frac{f_{j,r}}{m_j}e_3) - (R_i\frac{f_i}{m_i}e_3 - R_{i,r}\frac{f_{i,r}}{m_i}e_3)\Big)\Bigg].\end{aligned} \tag{4.69}$$

In order to find the input f_i for robot i, we have to solve the entire set of these equations for f_i with $i = 1, \ldots, n$. The entire (stacked) set of relations equals

$$\tilde{U} = \tilde{K}_\nu \begin{bmatrix} R_f^\top (R_1 f_1 - R_{1,r} f_{1,r}) \frac{e_3}{m_1} \\ \vdots \\ R_f^\top (R_n f_n - R_{n,r} f_{n,r}) \frac{e_3}{m_n} \end{bmatrix}, \tag{4.70}$$

in which

$$U^* = \begin{bmatrix} (\tilde{u}_1)^\top & \cdots & (\tilde{u}_n^*)^\top \end{bmatrix}^\top, \tag{4.71a}$$

$$\tilde{K}_\nu = \begin{bmatrix} I - \sum_{j \in N_1} \tilde{K}_{1j,\nu} & \tilde{K}_{12,\nu} & \cdots & \tilde{K}_{1n,\nu} \\ & I - \sum_{j \in N_2} \tilde{K}_{2j,\nu} & & \vdots \\ \vdots & & \ddots & \\ \tilde{K}_{n1,\nu} & \cdots & \tilde{K}_{n(n-1),\nu} & I - \sum_{j \in N_n} \tilde{K}_{n(n-1),\nu} \end{bmatrix}, \tag{4.71b}$$

$$K_{ij,\nu} = I \tilde{k}_{ij,\nu}. \tag{4.71c}$$

We rewrite (4.70), to

$$\tilde{K}_\nu \begin{bmatrix} R_f^\top R_1 f_1 e_3 \\ \vdots \\ R_f^\top R_n f_n e_3 \end{bmatrix} = M\tilde{U} + \tilde{K}_\nu \begin{bmatrix} R_f^\top R_{1,r} f_{1,r} e_3 \\ \vdots \\ R_f^\top R_{n,r} f_{n,r} e_3 \end{bmatrix}, \tag{4.72a}$$

$$\begin{bmatrix} R_f^\top R_1 f_1 e_3 \\ \vdots \\ R_f^\top R_n f_n e_3 \end{bmatrix} = \tilde{K}_\nu^{-1} M\tilde{U} + \begin{bmatrix} R_f^\top R_{1,r} f_{1,r} e_3 \\ \vdots \\ R_f^\top R_{n,r} f_{n,r} e_3 \end{bmatrix}, \tag{4.72b}$$

$$\begin{bmatrix} R_1 f_1 e_3 \\ \vdots \\ R_n f_n e_3 \end{bmatrix} = G_{R_f} \tilde{K}_\nu^{-1} M\tilde{U} + \begin{bmatrix} R_{1,r} f_{1,r} e_3 \\ \vdots \\ R_{n,r} f_{n,r} e_3 \end{bmatrix}, \tag{4.72c}$$

$$\begin{bmatrix} R_1 f_1 e_3 \\ \vdots \\ R_n f_n e_3 \end{bmatrix} = G_{R_r} G_{R_r}^\top G_{R_f} \tilde{K}_\nu^{-1} M\tilde{U} + \begin{bmatrix} R_{1,r} f_{1,r} e_3 \\ \vdots \\ R_{n,r} f_{n,r} e_3 \end{bmatrix}, \tag{4.72d}$$

$$\begin{bmatrix} R_{1,r}^\top R_1 f_1 e_3 \\ \vdots \\ R_{n,r}^\top R_n f_n e_3 \end{bmatrix} = G_{R_r}^\top G_{R_f} \tilde{K}_\nu^{-1} M\tilde{U} + \begin{bmatrix} f_{1,r} e_3 \\ \vdots \\ f_{n,r} e_3 \end{bmatrix} := \begin{bmatrix} F_1 \\ \vdots \\ F_n \end{bmatrix}, \tag{4.72e}$$

in which $M = \text{diag}(m_1, \ldots, m_n)$, $G_{R_f} = \text{diag}(R_f, \ldots, R_f)$, and $G_{R_r} = \text{diag}(R_{1,r}, \ldots, R_{n,r})$ with n equal to the total number of agents in the system. We now find the actual force magnitude input f_i by utilizing the fact that $f_i = ||R_{i,r}^\top R_i f_i e_3||$, resulting in

$$f_i = ||F_i||. \tag{4.73}$$

We consider the misalignment between $R_{i,r}$ and R_i from (4.72e) as the desired relative attitude in order to actually achieve the virtual input $\tilde{u}_i$. We find this desired attitude based on (4.72e), as

$$R_{i,d} e_3 = \frac{F_i}{f_i} := f_{i,d}, \tag{4.74}$$

where $f_i(t) > 0$ because of the included saturation in u_i from (4.66). Similar as before, since this only defines the third unit axis of the rotation matrix which we define as the desired thrust direction

$f_{i,d}$, rotation around this axis is still free. Therefore we still have to choose a desired yaw angle in order to find a suitable rotation matrix, and therefore desired attitude. We choose to make $R_{i,d}$ rotate the direction of the desired thrust vector $R_{i,d}e_3$ to the direction of the thrust vector of the reference in the reference frame e_3, in the spanned plane. We again use Rodrigues' rotation formula from (4.25), since we again find both the rotation angle and axis based on the fact that we rotate $R_{i,d}e_3$ to e_3 in the spanned plane and therefore around the axis perpendicular to these two vectors. By following the same approach as in Section 4.1.3, we obtain

$$R_{i,d} = \begin{bmatrix} 1 - \frac{f_{i,d1}^2}{1+f_{i,d3}} & -\frac{f_{i,d1}f_{i,d2}}{1+f_{i,d3}} & f_{i,d1} \\ -\frac{f_{i,d1}f_{i,d2}}{1+f_{i,d3}} & 1 - \frac{f_{i,d2}^2}{1+f_{i,d3}} & f_{i,d2} \\ -f_{i,d1} & -f_{i,d2} & f_{i,d3} \end{bmatrix} \in SO(3). \tag{4.75}$$

Since we also know that $\dot{R}_{i,d} = R_{i,d}S(\omega_{i,d})$, we find

$$\omega_{i,d} = \begin{bmatrix} -\dot{f}_{i,d2} + \frac{f_{i,d2}\dot{f}_{i,d3}}{1+f_{i,d3}} \\ \dot{f}_{i,d1} - \frac{f_{i,d1}\dot{f}_{i,d3}}{1+f_{i,d3}} \\ \frac{f_{i,d2}\dot{f}_{i,d1} - \dot{f}_{i,d2}f_{i,d1}}{1+f_{i,d3}} \end{bmatrix}. \tag{4.76}$$

What remains is to let $R_{i,r}^\top R_i$ converge to $R_{i,d}$. In order to do so, let us again use the same tracking approach as in Section 4.1.3. Define the attitude tracking error

$$R_{i,e} = R_{i,d}^\top (R_{i,r}^\top R_i), \tag{4.77a}$$

with corresponding angular velocity error expressed relative to the body-fixed frame $\mathcal{B}_i$ of quadrotor i

$$\omega_{i,e} = \omega_i - R_i^\top R_{i,r}\omega_{i,r} - (R_i^\top R_{i,r})R_{i,d}\omega_{i,d}, \tag{4.77b}$$

where $(R_i^\top R_{i,r})R_{i,d} = R_{i,e}^\top$. In order to solve the tracking problem, we follow the remaining steps from Section 4.1.3. In fact, this means that the attitude tracking system remains unchanged and only the relations between $\tilde{u}_i$ and the actual input f_i and desired attitude $R_{i,d}$ is now different as a result of introduced coupling. The method that is used to achieve this desired attitude by tracking remains equal. For completeness, let us adopt some proceeding steps in this section.

Differentiating (4.77) along their solutions, yields

$$\dot{R}_{i,e} = R_{i,e}S(\omega_{i,e}), \tag{4.78a}$$

$$\begin{aligned} J_i\dot{\omega}_{i,e} = {} & S(J_i\omega_i)\omega_i + \tau_i + S(\omega_i)R_i^\top R_{i,r}\omega_{i,r} - R_i^\top R_{i,r}S(\omega_{i,r})\omega_{i,r} \dots \\ & - R_i^\top R_{i,r}J^{-1}\left[S(J_i\omega_{i,r})\omega_{i,r} + \tau_{i,r}\right] + S(\omega_{i,e})R_{i,e}^\top\omega_{i,d} - R_{i,e}^\top\dot{\omega}_{i,d}. \end{aligned} \tag{4.78b}$$

This means that we can consider Problem 4.1.2 as the problem to solve in order to stabilize the dynamics (4.78) too, since apart from some internal parametric relations the dynamics (4.78) are equivalent to (4.30). Thus, since the problem is equal, we adopt the same proposition for a solution, included as Proposition 4.1.2, to solve this problem. Choosing the input

$$\begin{aligned} \tau_i = {} & -K_{i,\omega}\omega_{i,e} + K_{i,R}\sum_{s=1}^{3} k_{i,s}(e_s \times R_{i,e}^\top e_s) - S(J_i\omega_i)\omega_i - S(\omega_i)R_i^\top R_{i,r}\omega_{i,r} \dots \\ & + R_i^\top R_{i,r}S(\omega_{i,r})\omega_{i,r} + R_i^\top R_{i,r}J^{-1}\left[S(J_i\omega_{i,r})\omega_{i,r} + \tau_{i,r}\right] - S(\omega_{i,e})R_{i,e}^\top\omega_{i,d} + R_{i,e}^\top\dot{\omega}_{i,d}, \end{aligned} \tag{4.79}$$

with $K_{i,\omega} = K_{i,\omega}^\top > 0$ and $K_{i,R} = K_{i,R}^\top > 0$, provides a closed-loop format that is equivalent to [11]. The closed-loop system ((4.78), (4.79)), yields

$$\dot{R}_{i,e} = R_{i,e}S(\omega_{i,e}), \tag{4.80a}$$

$$J_i\dot{\omega}_{i,e} = -K_{i,\omega}\omega_{i,e} + K_{i,R}\sum_{s=1}^{3} k_{i,s}(e_s \times R_{i,e}^\top e_s), \tag{4.80b}$$

for which $(I, 0)$ is ULES and UaGAS for distinct $k_{i,s}$ with $s = 1, 2, 3$ when $K_{i,\omega} = K_{i,\omega}^\top > 0$ and $K_{i,R} = K_{i,R}^\top > 0$, conform [30, 32], as included in Theorem 2.2.3. For the full proof, we refer to the proof of Proposition 4.1.2.

4.2.4 Stability of the cascaded structure

In the previous part, we have designed control action $\tilde{u}_i$ for the coupled position tracking error dynamics of a robot i that is coupled to all robots $j \in N_i$ conform Definition 3.3.1, and a controller for f_i and τ_i to asymptotically achieve this virtual control action $\tilde{u}_i$. However, since we asymptotically obtain the virtual input $\tilde{u}_i$, because of the asymptotic convergence of the desired attitude $R_{i,d}$, we do not at all times identically apply the virtual input $\tilde{u}_i$. More specifically, the solution of the attitude tracking subsystem, the residual error $R_{i,e}$, perturbs the position tracking subsystem by sometimes pointing the force magnitude input f_i in not exactly the desired direction. Therefore, we have to analyze the cascaded system. Additionally, if we are able to conclude

$$\lim_{t\to\infty} \tilde{\rho}_{i,e} = 0, \qquad \lim_{t\to\infty} \tilde{\nu}_{i,e} = 0, \qquad \lim_{t\to\infty} R_{i,e} = I, \quad \text{and} \quad \lim_{t\to\infty} \omega_{i,e} = 0, \tag{4.81}$$

the tracking control problem from Problem 4.1.1 is solved, since if (4.81) holds, the conditions from Problem 4.1.1 hold automatically.

Consider the dynamics of a fleet of n quadrotor UAVs (4.1), with $i = 1, \dots, n$, and a virtual structure (4.41) that provides a set of feasible reference trajectories that fulfill the dynamics (4.2), with $i = 1, \dots, n$, in closed-loop with inputs (4.66), (4.73) and (4.79). The cascaded closed-loop system in generalized coordinates then equals

$$\dot{\tilde{\rho}}_{i,e} = -S(\omega_f)\tilde{\rho}_{i,e} + \tilde{\nu}_{i,e}, \tag{4.82a}$$

$$\dot{\tilde{\nu}}_{i,e} = -S(\omega_f)\tilde{\nu}_{i,e} - k_{i,\rho}\sigma(\tilde{\rho}_{i,e}) - K_{i,\nu}\sigma(\tilde{\nu}_{i,e}) + \frac{f_i}{m_i} R_{i,r}^\top R_i (I - R_{i,e}^\top) e_3, \tag{4.82b}$$

$$\dot{R}_{i,e} = R_{i,e} S(\omega_{i,e}), \tag{4.82c}$$

$$J_i \dot{\omega}_{i,e} = -K_{i,\omega}\omega_{i,e} + K_{i,R} \sum_{s=1}^{3} k_{i,s}(e_s \times R_{i,e}^\top e_s), \tag{4.82d}$$

in which the cascaded term equals $\frac{f_i}{m_i} R_{i,r}^\top R_i (I - R_{i,e}^\top) e_3 = \frac{f_i}{m_i}(R_{i,r}^\top R_i - R_{i,d}) e_3$. Notice that the cascaded system (4.82) is very similar to (4.34). By directly applying Proposition 4.1.3, we conclude that the cascaded system (4.82) is UaGAS, meaning that we also provided a solution to Problem 4.1.1 including coupling on the position subsystem.

4.3 Simulation study

In the previous section, the stability of the system is already guaranteed by the provided mathematical proof. However, before using the developed system in a real-world application, we want to test specific behavioral aspects of the system and their relation to system parameters and gains. In this section, we try to find the effect that different coupling gains have on the convergence rate of getting in formation and tracking the individual references. Therefore, we first simulate a system of two quadrotor UAVs with a simple repetitive spatial reference trajectory that we can use in the future to compare to the experimental behavior. We apply several different coupling gains in order to find the effect on both coupling and tracking the individual references. Subsequently, in order to further investigate the formation tracking properties, we created a stylized scenario in which we can clearly see the consideration between tracking the individual references and the formation shape. At the end, we show what happens if one of the quadrotor UAVs is unable to keep up as a result of constant perturbation.

For the first simulations with two quadrotor UAVs, we choose the initial conditions

$$\begin{aligned}
\rho_1 &= \begin{bmatrix}0.5 & 0.5 & 0\end{bmatrix}^\top, & \rho_2 &= \begin{bmatrix}-0.5 & 0.5 & 0\end{bmatrix}^\top, \\
v_1 &= \begin{bmatrix}0.01 & 0.1 & 0.2\end{bmatrix}^\top, & v_2 &= \begin{bmatrix}0.01 & 0.1 & 0.2\end{bmatrix}^\top, \\
\phi_1 &= 0, & \phi_2 &= 0, \\
\theta_1 &= 0, & \theta_2 &= 0, \\
\psi_1 &= 0, & \psi_2 &= 0, \\
\omega_1 &= \begin{bmatrix}0.1 & 0.2 & -0.1\end{bmatrix}, & \omega_2 &= \begin{bmatrix}0.1 & 0.2 & -0.1\end{bmatrix}.
\end{aligned} \tag{4.83}$$

The system parameters are set at

$$\begin{aligned}
k_{1,\rho} = k_{2,\rho} &= 1, \\
K_{1,\nu} = K_{2,\nu} &= I, \\
K_{1,R} = K_{2,R} &= 0.5I, \\
K_{1,\omega} = K_{2,\omega} &= 1.2I, \\
m_i = 1, &\quad J_i = 1,
\end{aligned} \tag{4.84}$$

and the two different sets coupling gains are used for two different simulations

$$\tilde{k}_{12,\rho} = \tilde{k}_{21,\rho} = \tilde{k}_{12,\nu} = \tilde{k}_{21,\nu} = 1, \tag{4.85a}$$

and

$$\tilde{k}_{12,\rho} = \tilde{k}_{21,\rho} = \tilde{k}_{12,\nu} = \tilde{k}_{21,\nu} = 5. \tag{4.86a}$$

We first choose the formation trajectory and shape vectors equal to

$$\rho_f(t) = \begin{bmatrix}\cos(t) & \sin(t) & \sin(t)\end{bmatrix}^\top, \tag{4.87a}$$

$$R_f(t) = R_z(t), \tag{4.87b}$$

$$p_i(t) = r\begin{bmatrix}\sin(i\frac{(4-n)\pi}{2n}) & \cos(i\frac{(4-n)\pi}{2n}) & 0\end{bmatrix}^\top, \tag{4.87c}$$

with radius $r = 0.3$ so that a the agents are equally spaced on a circle circumference in the plane for which $p_{i,z} = 0$. The yaw angle t is chosen to make the forward direction of the formation tangent to the curve, since $\tan^{-1}\left(\frac{\cos(t)}{\sin(t)}\right) = t$; which relates to a constant yaw velocity of 1 rad/s. Thus, we provide a 3D circular formation trajectory with the forward direction of R_f with both reference quadrotor UAVs positioned with an equal offset (one with 0.3 and the other with -0.3) relative to the formation frame $\mathcal{F}$. For clarity, let us explicitly mention that we consider the individual coupling error between two agents i and j, as

$$\rho_{ij,\epsilon} = R_f^\top((\rho_{j,r} - \rho_j) - (\rho_{i,r} - \rho_i)), \tag{4.88a}$$

$$\nu_{ij,\epsilon} = R_f^\top((R_{j,r}\nu_{j,r} - R_j\nu_j) - (R_{i,r}\nu_{i,r} - R_i\nu_i), \tag{4.88b}$$

which provides the full mutual coupling terms, equal to

$$\rho_{i,\epsilon} = R_f^\top \sum_{j\in N_i}((\rho_{j,r} - \rho_j) - (\rho_{i,r} - \rho_i)), \tag{4.89a}$$

$$\nu_{i,\epsilon} = R_f^\top \sum_{j\in N_i}((R_{j,r}\nu_{j,r} - R_j\nu_j) - (R_{i,r}\nu_{i,r} - R_i\nu_i). \tag{4.89b}$$

The reference tracking errors are considered as

$$\rho_{i,e} = R_f^\top(\rho_{i,r} - \rho_i), \tag{4.90a}$$
$$\nu_{i,e} = R_f^\top(R_{i,r}\nu_{i,r} - R_i\nu_i), \tag{4.90b}$$

which together with the weighted sum of all individual coupling terms form the defined error (or generalized) coordinates from the definition (4.62). We like to make this distinction clear so that we are able to mention differences in reference tracking and coupling behavior, both on position and velocity level.

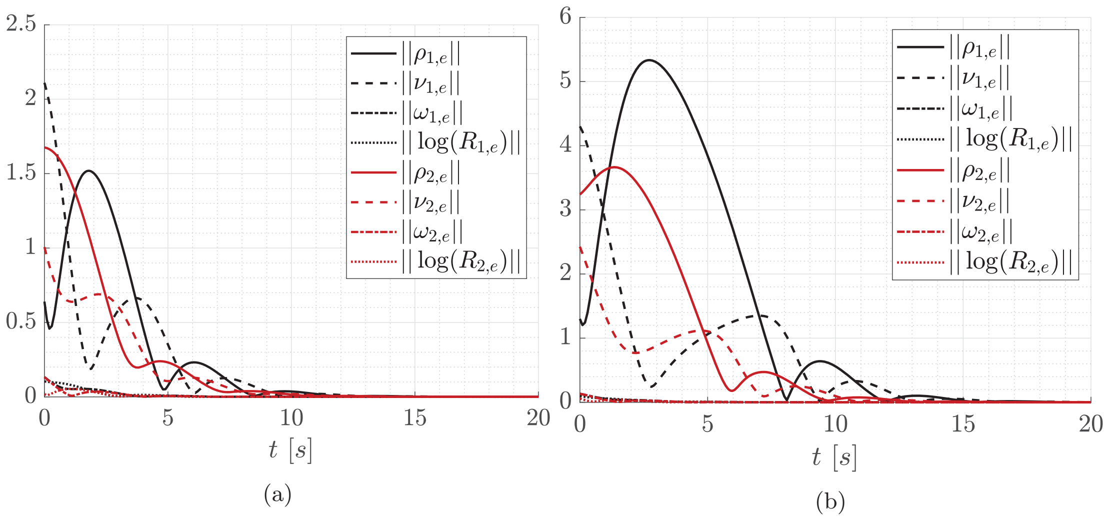


Figure 4.3: Error magnitudes from the simulation scenario (4.87) with system parameters (4.84) and initial conditions (4.83). The coupling gains (4.85a) are used in (a), and (b) shows the results from a simulation with coupling gainss (4.86a).

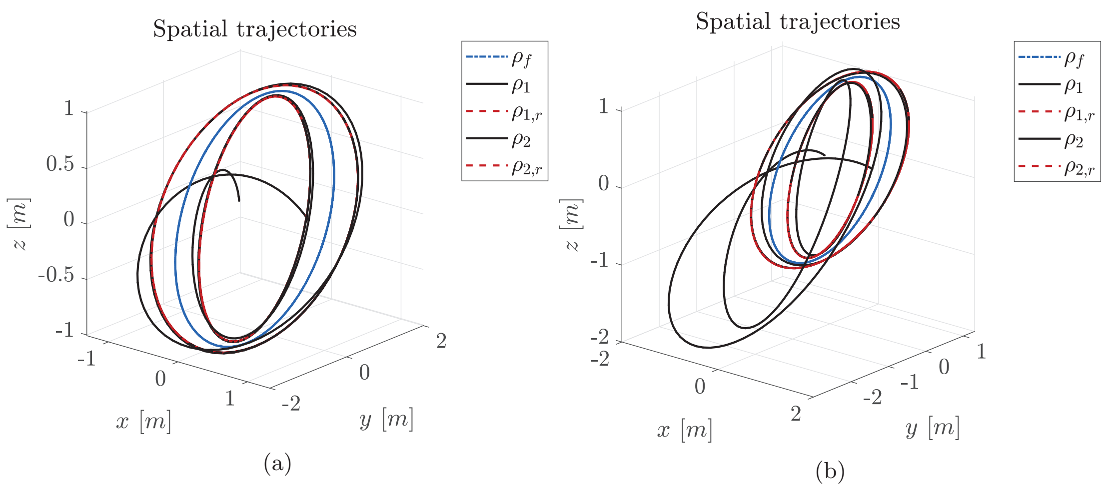


Figure 4.4: Spatial trajectories from the simulated scenario (4.87) with system parameters (4.84) and initial conditions (4.83). The coupling gains (4.85a) are used in (a), and (b) shows the results from a simulation with coupling gainss (4.86a).

The simulation is executed for both sets of coupling gains and the results are implemented in figures 4.3 till 4.5. Figure 4.3 shows that the magnitude of all errors reduces to 0 for both simulations, so the quadrotor UAVs eventually converge to their desired states; as expected from the mathematical proof provided in the previous section. Furthermore, it shows that for a high coupling gain, which basically means a higher cost on the coupling error, the magnitude of the generalized error coordinates is greater. In Figure 4.4, the flight paths of the formation, reference UAVs and quadrotor UAVs are included for both coupling strengths (4.85a) and (4.86a). It shows that for a high coupling gain, the quadrotors take a detour before converging towards the reference trajectory. As a result of a high relative cost for the coupling error with respect to the error with their individual reference position, the quadrotors are allowed to increase their individual tracking errors as long as it provides a reduced coupling error; hence the detour. Figure 4.5 shows what happens to the magnitude of the individual position tracking error vectors and coupling error vector for both coupling strengths (4.85a) and (4.86a). Notice that the overall behavior with the increased coupling gain is worsened, meaning that it takes a longer time before all position errors are converged, because we use the same proportional and derivative gains $k_{i,\rho}$ and $k_{i,\nu}$, while in order to obtain the same spatial tracking behavior we should increase these accordingly. Furthermore, Figure 4.5 shows that for a high cost on coupling error the individual errors increase a lot in order to reduce the coupling error, which is in agreement with the detour that is shown in Figure 4.4.

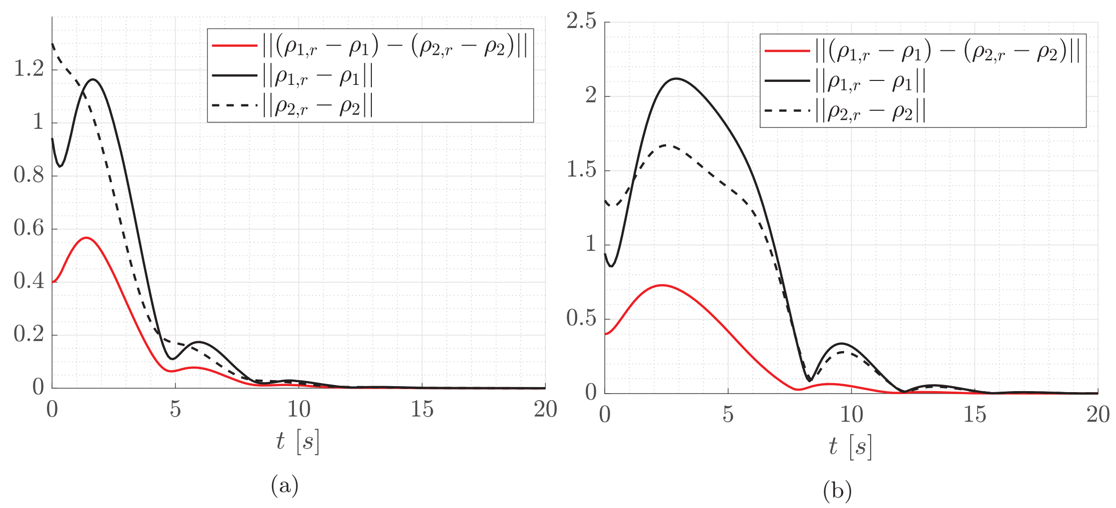
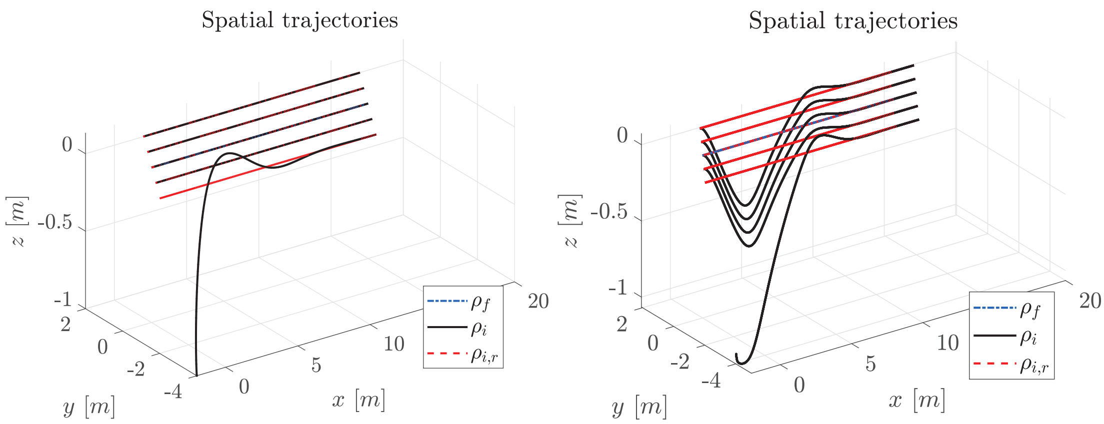


Figure 4.5: Position reference tracking errors and coupling errors from the simulation scenario (4.87) with system parameters (4.84) and initial conditions (4.83). The coupling gains (4.85a) are used in (a), and (b) shows the results from a simulation with coupling gainss (4.86a).

In the previous simulations we have provided a scenario that is reproducible in the real world and showed the effect of different coupling gains on a system that is initiated from specific initial states. However, we especially developed a coupling structure in order to make all agents react to disturbances to a specific agent during operation, so that we have control over the formation (shape) tracking behavior even when one of the agents is perturbed. Therefore, in order to showcase the tracking behavior during these disturbances, we simulate a stylized scenario in which four agents are exactly on their reference states and the 5th and last agent is blown off its reference and thus perturbed. We use the same system parameters as in (4.84) and simulate a linear movement of a diagonal line-formation in positive x-direction. The results for simulations starting directly after the perturbation without coupling ($\tilde{k}_{12,\rho} = \tilde{k}_{21,\rho} = 0$ and $\tilde{k}_{12,\nu} = \tilde{k}_{21,\nu} = 0$) and with coupling ($\tilde{k}_{12,\rho} = \tilde{k}_{21,\rho} = 1$ and $\tilde{k}_{12,\nu} = \tilde{k}_{21,\nu} = 1$) are included in Figure 4.6.



(a) (b)

Figure 4.6: Spatial flight paths with linear reference paths for the situation where all quadrotors are initially on the reference trajectories except for one perturbed quadrotor UAV. In (a) there is no coupling and in (b) there is coupling. The virtual formation centered frame $\mathcal{F}$ is located at the potition of the quadrotor UAV that is in the middle and the formation moved in positive x-direction.

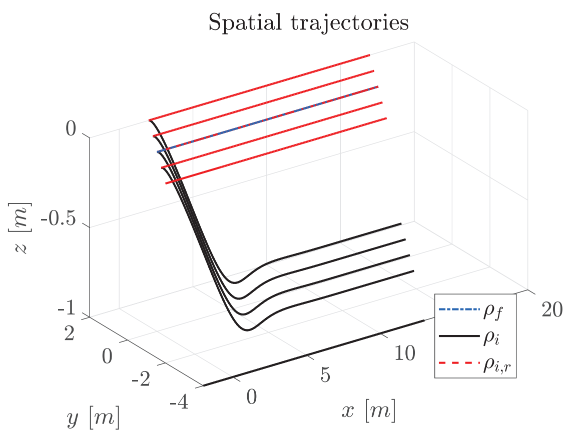


Figure 4.7: Spatial flight paths with linear reference paths for the situation where all quadrotors are initially on the reference trajectories except for one perturbed quadrotor UAV. One agent is constantly perturbed to the extend that it cannot converge to the reference formation and coupling gains are set at $\tilde{k}_{12,\nu} = \tilde{k}_{21,\nu} = 1$. The virtual formation centered frame $\mathcal{F}$ is located at the potition of the quadrotor UAV that is in the middle and the formation moved in positive x-direction.

Figure 4.6 shows that in the simulation without coupling the agents do not react to the disturbance to one of the quadrotors in the system, while in the simulation with coupling the system first converges partly to the desired formation shape before together converging to the desired location. However, as a result of motor failure or disturbances from the surroundings it is possible that the disturbed quadrotor UAV is unable to catch up with the formation. Therefore, in Figure 4.7 we included simulation results of what happens if the perturbed agent is unable to converge back to the reference formation. Figure 4.7 shows that as a result of the coupling term all other agents move towards the perturbed agent until an equilibrium is met, which can be altered by changing the relative cost declared to the coupling errors with respect to the reference tracking errors, as explained before.

Although we showed the desired tracking and coupling behavior and the ability to control the consideration between tracking the individual references and coupling, some necessary steps have to be taken in order to obtain a real-world system of multiple quadrotor UAVs that can be operated. These necessary steps, like how to use an external source to locate and identify multiple quadrotor UAVs and which network architecture to use for communication between quadrotors and with the supervisor, are executed and included in Appendix A. Although the entire system is developed and tested, the system showed some shortcomings as a result of hardware limitations and software architectural decisions. These limitations are also included in Appendix A and possible solutions are included as learning points. The main shortcoming was the large delay and large sample time for the external localization. Furthermore, notice that there is no collision avoidance algorithm, so there is no guarantee that drones will not collide with each other. Therefore, before experimental operation of a real-world system a collision avoidance algorithm should be implemented. Alternatively, the change for a collision can be minimized by defining a reference with all quadrotors relatively far apart and testing in simulations before executing experiments for each scenario.

Concluding, in the first simulations we provided a 3D formation path for a formation of two quadrotor UAVs. We showed that all generalized error coordinates converge to the absolute origin. Solely increasing the cost on the coupling error made the size of these generalized coordinates increase for the same scenario, caused by this increased 'weight'. More important, increasing the cost on the coupling error allowed the tracking errors to increase to reduce the coupling error, which caused the quadrotors to take a detour before converging to the reference paths. In order to further investigate the added ability to control the formation shape with the implemented coupling structure, we did some more simulations that showed what happens if a single agent is momentarily or constantly perturbed. We showed that we are then able to control the consideration between moving towards the perturbed agent in order to keep or get in formation (with a high coupling gain) or moving towards or keeping at the individual reference trajectories (coupling gain very low or zero).

4.4 Concluding remarks

In this chapter, we first introduced a model and notation for the quadrotor UAV that allows for multiple agents in the system, by identifying specific agents with their index number. After that, we adopted a tracking problem from previous research, which we first solved slightly differently for the single agent system. We defined the error coordinates relative to a mutually known formation centered frame and changed the control law accordingly. We showed that the closed-loop position tracking error dynamics are uniform global asymptotically stable (UGAS) based on a defined virtual input. Subsequently, we showed that we are capable of obtaining the virtual input by choosing a specific force magnitude input that is related to the magnitude of the virtual input. Subsequently, to also point the vector that now has the right magnitude in the desired direction, we defined the desired attitude based on the desired direction and a specific choice for yaw angle. Then, we defined an attitude error on SO(3) for the misalignment between the actual attitude and the desired attitude. The desired attitude was then achieved by formulating an attitude tracking control problem and the closed-loop system was rendered uniformly almost-globally asymptotically stable (UaGAS) with the designed control action. Since the attitude converges to the desired attitude rather than that the attitude is obtained at all times, the position tracking subsystem is perturbed by the solution of the attitude tracking subsystem. Thereby, a cascaded structure was identified. Based on cascaded theory, a cascade system analysis was executed showing that the cascaded system is UaGAS.

Proceeding, we first defined a virtual reference structure, used to obtain a set of n feasible reference trajectories for a system with n quadrotor UAVs. In order for the reference to be feasible for tracking, we need the parametric formation flight path ($\rho_f(t)$), formation attitude ($R_f(t)$) and possibly time varying formation shape ($p_i(t)$) to be four times differentiable. The availability of the feasible reference trajectories already allowed us to track a virtual reference formation with a fleet of quadrotor UAVs, but still did not guarantee that the quadrotors stay in formation, since they only try to track their individual reference trajectories and do not communicate and therefore not react to the others. In order to provide some guarantee for the formation tracking abilities of the multi-agent system, we had to synchronize operation based on cooperation of the agents. We provided this wanted level of cooperation by defining a consideration between tracking the individual reference trajectories and getting or staying in formation. This was established by implementing coupling between the agents.

We have chosen to strive for mutual coupling instead of, e.g., master-slave coupling, based on our literature research. This choice was made because we want to obtain a homogeneous system of multiple agents that is easily scalable to different numbers of agents and able to form a structured formation that performs agile maneuvers. Although different alternatives were investigated, we implemented the coupling errors in newly defined generalized coordinates for position tracking and followed the same cascaded approach for developing control laws. Because of this cascaded approach, we chose to leave the attitude subsystems uncoupled, since we already use the attitudes of the UAVs to obtain the virtual inputs that cause the wanted position tracking behavior. We mentioned that additionally taking care of attitude coupling errors besides the reference tracking errors would act as a disturbance to the attitude reference tracking convergence and therefore as a disturbance on the position tracking behavior. This results from the fact that quadrotor UAVs (have to) use the attitude of the quadrotor UAV to move it, since the thrust vector has to be directed by the attitude of the body, because the thrust vector is fixed to the body; quadrotor UAVs are under-actuated devices. Furthermore, by not including coupling on the attitude subsystems, we remained the previously defined attitude subsystems entirely in tact. Similar as before, we defined a virtual input that caused the now coupled position tracking subsystem to be UGAS and used the actual inputs to obtain this virtual input magnitude and direction separately. Since the desired attitude can differ from the actual attitude, the position tracking subsystem is perturbed by the error of the attitude tracking subsystem, which indicated the cascaded structure. Using cascade system theory, the stability of the cascaded structure was analyzed and the system was proven UaGAS.

Subsequently, we tested the behavior of the system by simulating some testcases. It was shown that we are able to alter the coupling strength by altering the cost on the coupling error relative to the cost on reference tracking and therefore have control over the consideration between getting/staying in formation and tracking the individual reference trajectories. For example, a low cost on the coupling errors caused the system to allow the coupling error to increase in order to converge to the reference faster, while a high cost on the coupling error caused the coupling error to converge to 0 faster even when this prohibits the agents to track their individual references perfectly; e.g., if one agent is unable to catch up while the others are on their reference, the agents move off their reference in order to better obtain the formation shape. Thereby, it is shown that the formation is able to react to disturbances to specific agents and that we are able to influence this reaction with the coupling gain; meaning that with a coupling gain of 0 we remove this reaction/coupling entirely and with a higher coupling gain we better obtain the reference formation shape. Furthermore, since there is no collision avoidance implemented in the system, we cannot ensure to avoid collisions. Besides actual collisions, for quadrotor UAVs it is dangerous to enter each others downwash, meaning that is dangerous to cross other UAVs directly above or below. Since both collisions and flying underneath other quadrotors cause problems, in the future, a collision avoidance algorithm that additionally takes into account the hazardous zone underneath (and above) other UAVs has to be implemented.

Furthermore, a program for experimentation is developed including external localization and communication over a wireless network. During some early tests several learning points showed. First of all, a relatively high delay of max about 0.43 s is present in the system. Furthermore, the external localization algorithm has a very long sample time of 0.2 s, which makes it one of the limiting factors. Lastly, it is very important to make sure that the saturation terms and gains in the applied inputs provide inputs that are within the bounds of the motors. If not, the thrusts will be automatically saturated individually to a certain maximum, which makes controlling (the attitude of) the quadrotor UAV impossible.

Concluding, this means that we obtained almost global formation tracking control including coupling of multiple quadrotor UAVs on SE(3), based on some intermediate steps, starting from the results presented in [11] to almost globally control a single quadrotor UAV on SE(3). Simulations showed the ability to synchronize the formation but some future work remains before experimentation.

Chapter 5

Conclusions and recommendations

The concept of using several agents that partly rely on their interaction in order to obtain specific combined behavior seems very interesting for the next generation of control systems, while in the mean time progression persists in ever smarter design choices for control. We feel like the specific combination between both worlds, a relatively complex control strategy based on recent research as well as utilizing a well established coupling technique, opens up a promising direction for progression in multi-agent systems. In this chapter, we first provide conclusions that are drawn from the previous part of this research. After that, we want to point out specific limitations, promises and future work in the form of recommendations.

5.1 Conclusions

In this thesis, we first developed a system of multiple mobile robots under mutual coupling. We deliberately modeled the mobile robots with the Newton-Euler approach, which makes these mobile robots modeled very similar to the model that was later on adopted for the quadrotor UAV. After that, based on the insights that followed from the system of multiple unicycles, we developed the system with multiple quadrotor UAVs under mutual coupling. In this section we separately present our conclusions for the two to some degree similar multi-agent systems.

Formation tracking with mobile robots

Based on the Newton-Euler modeling framework together with the nonholonomic no side slip constraint, a model representation of unicycle robot is derived. Based on this model, we first introduced generalized coordinates with respect to an introduced virtual frame. We showed that we are able to stabilize the position dynamics, by choosing a virtual control law, and we showed that this virtual input was theoretically obtainable since we have full control over the force magnitude input and torque input that are incorporated. We neglected possible saturation necessity for real world systems because of actuator saturation and well definedness, since we only looked at the theoretical example of a unicycle in order to provide insights for the main objective that is the quadrotor UAV. The closed-loop position tracking subsystem is proven uniformly globally asymptotically stable (UGAS) based on the defined virtual input. The force magnitude input allowed us to obtain the virtual input magnitude and we showed that we are able to define a desired heading direction based on the virtual input. Only in some special scenarios, the centripetal force of the unicycle robot exceeds the magnitude of the combined reference and virtual input forces, then we are unable achieve the right magnitude. However, for now we accept these limitations for the unicycle system, as we again purely used this system as introductory case study for the main objective that is coupling the quadrotor UAV. We used the torque freedom as input for control and designed a tracking approach to let the actual heading direction converge to the desired heading direction asymptotically. The orientation tracking control system was proven uniformly almost-globally asymptotically stable (UaGAS). Based on the fact that the desired heading is not at all times obtained, we identified a cascaded structure, since the position tracking subsystem is perturbed by the solution of the orientation tracking subsystem. Subsequently, the cascaded system is proven UaGAS based on cascade system theory.

Then, we chose the previously introduced virtual frame as virtual center for a reference formation. Based on the formation trajectory, we provided conditions that ensured feasibility of the set reference trajectories, so that they can be used as reference for tracking. This already allowed to track the set of references by a set of unicycle robots, but without coupling we cannot guarantee the formation tracing behavior, since the quadrotor UAVs can not react to disturbances that happen to the other quadrotor UAVs. Therefore, in order to overcome this problem, we implemented mutual coupling between the agents. Since the defined tracking errors are expressed with respect to the mutually known virtual formation center, we directly compared tracking errors in coupling functions to synchronize behavior. We obtained a new set of generalized coordinates by adding coupling functions to the previously defined position tracking errors. The new generalized tracking dynamics provided a very similar structure as the single-agent system, which allowed us to use the same approach as for the single agent. We chose to leave the orientation tracking subsystem uncoupled since it then remains entirely in service of achieving the virtual input by attaining the desired orientation. This resulted in the fact that the orientation tracking subsystem remained in tact and prevented from conflicting orientation objectives; the orientation controller can simply try its best to service the position subsystem by obtaining the desired orientation instead of additionally trying to synchronize. Again, the closed-loop position tracking subsystem is proven UGAS for the scenario in which the orientation tracking subsystem is converged, which corresponds with the scenario in which the defined virtual input is perfectly obtained. The orientation tracking subsystem remained unchanged thus is proven UaGAS. What remained is to analyze the system in cascade, which additionally analyzed the scenario where the virtual input is not at all times obtained. This scenario originates from the fact that the actual orientation can differ from the desired orientation, which perturbs the position tracking subsystem. Based on the stability proofs for both separate subsystems and proven conditions on the cascade term, we proved the cascaded system UaGAS.

In order to test the behavior of the system, we simulated a simple test-case and showed that the mobile robots converged to the reference structure.

Formation tracking with quadrotor UAVs

Subsequently, we adopted an approach for the quadrotor UAV very similar to the approach we used for the unicycle robot. Different from the unicycle robot, we directly adopted a model and controller with only minor differences regarding coupling-readiness, namely, we again defined the error coordinates relative to a mutually known virtual formation centered frame. We followed the same approach of defining a virtual input for the now spatial position tracking subsystem, which rendered this subsystem UGAS. Then, the size of the virtual input is again obtained with the force magnitude input and we defined a desired attitude. Because of the fact that we used saturation in for the virtual input, the desired direction is well defined and the magnitude of the thrust vector is bounded, in compliance with actuator saturation. Together, these force magnitude input and desired attitude provide the wanted virtual input. The actual thrust vector is fixed to the body and aligned with the body-fixed downwards direction. Because on this alignment, we were still free to choose any rotation around the thrust vector while we still able to obtain the virtual input. In order to have the desired attitude fully defined from the desired direction, we chose to prescribe the rotation as a rotation around the axis perpendicular to both the desired thrust vector and the thrust vector of the reference. This choice provided that we that the yaw angle is defined in addition to the roll and pitch angles that follow from the desired direction of the thrust vector, meaning that we have the desired attitude fully defined. In order to asymptotically achieve this desired attitude, we defined attitude tracking errors and a control law that renders this subsystem UaGAS.

A cascaded structure resulted from the fact that quadrotor UAV is not in the desired attitude at all times, so the system was analyzed using cascade system theory. Based on cascade system theory, the cascaded system was proven UaGAS.

Now, we defined virtual formation structure in space to obtain a set of feasible reference trajectories for a set of quadrotor UAVs. This allowed to already use multiple uncoupled quadrotor UAVs to track this structure, but coupling is needed to have some control over the formation tracking abilities as a group; without coupling agents cannot react to disturbances to the other. Therefore, we defined coupling errors for the agents, based on the directly comparable tracking error definitions. These errors are directly comparable since we defined these relative to the mutually known formation centered frame. This frame choice also provided tracking behavior that is characteristic relative to the formation frame. Alternatively, if we chose the inertial frame to express the errors in, the tracking gains are expressed relative to the chosen inertial frame, thus the behavior differed for a different inertial frame choices. We showed that directly implementing the coupling functions in the defined virtual control laws provided a system that was hard to prove stable. Therefore, we implemented the coupling errors together with the tracking errors in new generalized coordinates in order to provide generalized dynamics that are similar to the generalized dynamics for a single agent. We were able to follow the same steps as before in order to prove stability.

We showed that the mutually coupled position tracking subsystem was UGAS, based on a defined virtual input. We also showed that this virtual is obtainable by the actual inputs for control. Then, we obtained the magnitude of the virtual input with the actual input, i.e., the force magnitude or thrust magnitude. After that, we defined a desired attitude. When we achieve this desired attitude, the thrust vector is pointed in the desired direction. In order to asymptotically achieve this desired attitude, we again formulated the same tracking problem which is again rendered stable using the torque input. The attitude system is proven to be UaGAS. Similar as for the unicycle robot, we chose to not implement coupling on the attitude subsystem, since we wanted it to be fully in service of the position tracking subsystem. Since the desired attitude is obtained asymptotically, resulting from the tracking approach that is used to obtain this desired attitude, the desired attitude is not exactly obtained at all times, which means that the position tracking subsystem is perturbed by the resulting error in the attitude tracking subsystem. Therefore, a cascaded structure is again identified. The cascaded structure is shown to be UaGAS based on cascade system theory.

In order to fully test the system, we executed simulations both with two and five agents. For the two-agent system, we verified that a low coupling strength resulted in relatively weak cooperation with respect to tracking the individual reference trajectories and a high coupling strength resulted in relatively strong cooperation with respect to tracking the individual references. With the five-agent system, we showed the resulting behavior when a single agent is perturbed once or perturbed continuously for the coupled system compared to the system with coupling off. We showed that the formation is able to react to disturbances to specific agents and that we are able to influence this reaction with the coupling gain.

Then, the system is implemented for experiments with two Parrot AR.Drone 2.0 quadrotors and some necessary additions are provided. These additions comprise external localization and a network architecture for real-time operation but we did not yet include an algorithm to avoid collisions and entering each others dangerous airflows. Some primary results showed a relatively large delay of 0.43 s and relatively long sampling time of 0.2 s for the external localization source. This prevented us from performing aerial experiments with the current software implementation on the chosen hardware.

5.2 Recommendations

Although the main objective of our research is obtained, the current status is far from a complete solution for multi-agent quadrotor systems. First of all, the quadrotor UAVs are coupled by including coupling errors in the generalized coordinates and then stabilized by a virtual input. However, in order to actually achieve and thus apply this virtual input, all quadrotors have to know almost all states of the other quadrotors. This of course consumes a large amount of the communication bandwidth, but also requires an enormous amount of calculation power. This means that the system tends to be operated in a centralized manner, since all the quadrotor UAVs otherwise have to run very similar and relatively heavy calculations. For future research, two obvious improvements arise. We can embrace the centralized operation and find uses that can work while hosted by a centralized ground unit, or we can reconsider the coupling approach to create a more decentralized formation tracking algorithm. The first approach allows us to focus a bit less on calculation power and more on additional algorithms to create a more advanced system, but also makes it much harder to get the developed system out of the research and demonstration atmosphere. The second alternative provides a system that is potentially very scalable, but provides constraints on calculation power and possibly prevents from a very coordinated approach as the one we developed.

Additionally, currently the program is developed with and implemented in Euler angles, but we are free to use a quaternion representation in the software of the system. This quaternion representation requires less storage and less calculation power for rotation chaining than the rotation matrix alternative. Possible improvement of the program can be made regarding the required amount of calculations (CPU) and required memory capacity (RAM) by using quaternions to represent rotations.

Currently, the controllers are developed under the assumption that we can perfectly access the state of all quadrotor UAVs at any time. The truthful situation is, however, that we have sampled and delayed measurements of some states with a limited accuracy and at specific times. The exchange of state information even happens over WiFi with UDP, allowing for information to never arrive or end up at the wrong device without any of the agents noticing. Therefore, in the future, each of the agents has to be equipped with a sufficient state observer in order to provide some guaranteed accuracy regarding the own state, and therefore for others, even between measurements. In that way, the currently developed system with the relatively low sampling frequency for external localization could be sufficient for real-world operation. Past research has already considered observers that can be used, but improvements can still be made, e.g., by including time delays. Another approach might be to increase the sampling frequency of external localization, e.g., by considering only a part of the image from the overhead camera based on the last known position of the quadrotors.

In the current system, it is necessary to have the thrust magnitude larger than zero in order to have the desired attitude well defined. Currently, this requirement is ensured to be achieved by using saturation of the virtual input, other approaches like a hybrid system approach can be considered as well. This can increase the performance of the position tracking controller and with that the agility of the UAV.

Then, before executing experiments, some kind of collision avoidance algorithm should be implemented in order to have the guarantee that collisions will not happen. This algorithm should take into account possible collisions, but also dangerous airflow resulting from the propellers of other quadrotor UAVs. Without collision avoidance, it is only possible to decrease the chance on a collision by flying with a large distance between the quadrotors and by checking the flight paths beforehand.

Lastly, in previous research it has been shown that integral action is required to overcome offsets that result from a faulty mass in the model, while we currently did not implement this integral action. Especially for large systems of 'homogeneous' quadrotor UAVs it is very hard to get all masses modeled correctly, since the identity of a specific quadrotor can change during or between operations and these systems are never truly homogeneous as a result of production differences. Many of the sensor differences should be handled by the observer or by some kind of automated calibration. For the mass estimation fault, however, we should include the integral action in order to overcome steady state errors as a result of faulty mass determination.

Bibliography

[1] M. Hochstenbach, C. Notteboom, B. Theys, and J. de Schutter, “Design and control of an unmanned aerial vehicle for autonomous parcel delivery with transition from vertical take-off to forward flight – vertikul, a quadcopter tailsitter,” *International Journal of Micro Air Vehicles*, vol. 7, no. 4, pp. 395–405, 2015.

[2] R. Mahony, V. Kumar, and P. Corke, “Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor,” *IEEE Robotics Automation Magazine*, vol. 19, pp. 20–32, 09 2012.

[3] K. Thu and A. Gavrilov, “Designing and modeling of quadcopter control system using l1 adaptive control,” *Procedia Computer Science*, vol. 103, pp. 528 – 535, 2017. XII International Symposium Intelligent Systems 2016, INTELS 2016, 5-7 October 2016, Moscow, Russia.

[4] R. García, F. Rubio, and M. Ortega, “Robust PID control of the quadrotor helicopter,” *IFAC Proceedings Volumes*, vol. 45, no. 3, pp. 229 – 234, 2012. 2nd IFAC Conference on Advances in PID Control.

[5] M. Belkheiri, A. Rabhi, A. Hajjaji, and C. Pegard, “Different linearization control techniques for a quadrotor system,” in *CCCA12*, pp. 1–6, Dec 2012.

[6] X. Heng, D. Cabecinhas, R. Cunha, C. Silvestre, and X. Qingsong, “A trajectory tracking lqr controller for a quadrotor: Design and experimental evaluation,” in *TENCON 2015 - 2015 IEEE Region 10 Conference*, pp. 1–7, Nov 2015.

[7] S. A. Al-Hiddabi, “Quadrotor control using feedback linearization with dynamic extension,” in *2009 6th International Symposium on Mechatronics and its Applications*, pp. 1–3, March 2009.

[8] T. Madani and A. Benallegue, “Backstepping control for a quadrotor helicopter,” in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3255–3260, Oct 2006.

[9] M. Bangura and R. Mahony, “Real-time model predictive control for quadrotors,” *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11773 – 11780, 2014. 19th IFAC World Congress.

[10] E.-H. Zheng, J.-J. Xiong, and J.-L. Luo, “Second order sliding mode control for a quadrotor uav,” *ISA Transactions*, vol. 53, no. 4, pp. 1350 – 1356, 2014. Disturbance Estimation and Mitigation.

[11] E. Lefeber, S. J. A. M. van den Eijnden, and H. Nijmeijer, “Almost global tracking control of a quadrotor uav on SE(3),” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 1175–1180, Dec 2017.

[12] P. Stone and M. Veloso, “Multiagent systems: A survey from a machine learning perspective,” *Autonomous Robots*, 1997. Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213.

[13] H. Gutierrez, A. Morales, and H. Nijmeijer, "Synchronization control for a swarm of unicycle robots: Analysis of different controller topologies," *Asian Journal of Control*, vol. 19, no. 5, pp. 1822–1833, 2017.

[14] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 926–939, Dec 1998.

[15] A. Sadowska, T. van den Broek, H. Huijberts, N. van de Wouw, D. Kostic, and H. Nijmeijer, "A virtual structure approach to formation control of unicycle mobile robots using mutual coupling," in *International Journal of Control*, pp. 1886–1902, 09 2011.

[16] A. Bondhus, K. Pettersen, and H. Nijmeijer, "Master-slave synchronization of robot manipulators," *IFAC Proceedings Volumes*, vol. 37, no. 13, pp. 447 – 452, 2004. 6th IFAC Symposium on Nonlinear Control Systems 2004 (NOLCOS 2004), Stuttgart, Germany, 1-3 September, 2004.

[17] H. Nijmeijer and A. Rodrguez-Ángeles, *Synchronization of Mechanical Systems*. Nonlinear Science, 46, 01 2003. Singapore: World Scientific.

[18] A. Rodriguez-Angeles and H. Nijmeijer, "Mutual synchronization of robots via estimated state feedback: a cooperative approach," *IEEE Transactions on Control Systems Technology*, vol. 12, pp. 542–554, July 2004.

[19] S. van den Eijnden, "Cascade based tracking control of quadrotors," Master's thesis, Eindhoven University of Technology, Dynamics and Control Group, Department of Mechanical Engineering, 2017. Eindhoven, The Netherlands, MSc Thesis, DC 2017.012.

[20] N. Jeurgens, "Identification and control implementation of an ar.drone 2.0," Master's thesis, Eindhoven University of Technology, Dynamics and Control Group, Department of Mechanical Engineering, 2017. Eindhoven, The Netherlands, MSc Thesis, DC 2017.013.

[21] N. Jeurgens, "Implementing a simulink controller in an ar.drone 2.0," 2016. Internship Report, Eindhoven, The Netherlands, DC 2016.005.

[22] E. Lefeber, "Modeling and control of quadrotors." Presentation, 18-01-2018. Eindhoven University of Technology, Dynamics and Control Group, Department of Mechanical Engineering.

[23] E. Fresk and G. Nikolakopoulos, "Full quaternion based attitude control for a quadrotor," in *2013 European Control Conference (ECC)*, pp. 3864–3869, 07 2013.

[24] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," in *Matrix*, vol. 58, pp. 1–35, 10 2006.

[25] P. Betsch and R. Siebert, "Rigid body dynamics in terms of quaternions: Hamiltonian formulation and conserving numerical integration," *International Journal for Numerical Methods in Engineering*, vol. 79, no. 4, pp. 444–473, 2009.

[26] J. B. Kuipers, "Quaternions and rotation sequences," in *Proceedings of the International Conference on Geometry, Integrability and Quantization*, (Sofia, Bulgaria), pp. 127–143, Coral Press Scientific Publishing, 2000.

[27] M. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Wiley select coursepack, ProQuest Information and Learning Company, 2005.

[28] N. van de Wouw, *Multibody and Nonlinear Dynamics: Multibody Dynamics*. Dereumaux Printservice, Eindhoven University of Technology, 2016. Lecture notes, Course ID: 4DM10.

[29] E. Eade, "Lie groups for 2D and 3D transformations." `http://ethaneade.com/lie.pdf`, 2017. Updated on 20-05-2017: added paragraph 2.4.2 by clarifying notational differences.

[30] N. Chaturvedi, A. Sanyal, and N. McClamroch, "Rigid-body attitude control," *IEEE Control Systems Magazine*, vol. 31, pp. 30–51, 2011.

[31] S. Zhao, "Time derivative of rotation matrices: A tutorial," *CoRR*, vol. abs/1609.06088, 2016.

[32] A. Sanyal and N. Chaturvedi, "Almost global robust attitude tracking control of spacecraft in gravity," 2008.

[33] T. Lee, "Exponential stability of an attitude tracking control system on SO(3) for large-angle rotational maneuvers," *Systems and Control Letters*, vol. 61, no. 1, pp. 231 – 237, 2012.

[34] D. Lay, *Linear Algebra and Its Applications*. Pearson Education, 2002.

[35] H. Goldstein, C. Poole, and J. Safko, *Classical Mechanics*. Addison Wesley, 2002.

[36] J. Cariño, H. Abaunza, and P. Castillo, "Quadrotor quaternion control," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 825–831, 06 2015.

[37] A. Loria, E. Panteley, D. Popovic, and A. Teel, "A nested Matrosov theorem and persistency of excitation for uniform convergence in stable nonautonomous systems," *IEEE Transactions on Automatic Control*, vol. 50, pp. 183–198, Feb 2005.

[38] H. Khalil, *Nonlinear Systems*. Pearson Education, Prentice Hall, 2002.

[39] P. Monzon, "On necessary conditions for almost global stability," *IEEE Transactions on Automatic Control*, vol. 48, pp. 631–634, April 2003.

[40] T. Lee, M. Leok, and N. McClamroth, "Stable manifolds of saddle points for pendulum dynamics on S2 and SO(3)," 03 2011. Proceedings of the 50th IEEE Conference on Decision and Control.

[41] E. Panteley and A. Loria, "On global uniform asymptotic stability of nonlinear time-varying systems in cascade," *Systems and Control Letters*, vol. 33, no. 2, pp. 131 – 138, 1998.

[42] A. Lefeber, *Tracking Control of Nonlinear Mechanical Systems*. PhD thesis, University of Twente, Enschede, The Netherlands, 2000.

[43] A. Sadowska, D. Kostic, N. van de Wouw, H. Huijberts, and H. Nijmeijer, "Distributed formation control of unicycle robots," in *2012 IEEE International Conference on Robotics and Automation*, pp. 1564–1569, 05 2012.

[44] J. Laumond, *Robot motion planning and control*, vol. 229 of *Lecture notes in control and information sciences*. Berlin, Heidelberg: Springer-Verlag, 1998.

[45] P. Morin and C. Samson, *Motion Control of Wheeled Mobile Robots*, pp. 799–826. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[46] E. Panteley, E. Lefeber, A. Loria, and H. Nijmeijer, "Exponential tracking control of a mobile car using a cascaded approach," *IFAC Proceedings Volumes*, vol. 31, no. 27, pp. 201 – 206, 1998. IFAC Workshop on Motion Control (MC'98), Grenoble, France, 21-23 September.

[47] L. Smith, *Vectors in the plane and space*, pp. 1–12. New York, NY: Springer US, 1978.

[48] C. Godsil and G. Royle, *Algebraic Graph Theory*, vol. 207. 01 2001.

[49] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, stu - student ed., 2010.

[50] E. Steur, H. Nijmeijer, and J. Ploeg, "Dynamics and control of cooperation: Lecture slides," 2017. Eindhoven University of Technology, Dynamics and Control Group, Department of Mechanical Engineering, Course 4DM50.

[51] G. Cai, B. Chen, and T. Lee, *Unmanned Rotorcraft Systems*. Springer Publishing Company, Incorporated, 1st ed., 2011.

[52] A. Weiss, G. Cruz, K. Agarwal, Y. Rahman, M. Medikeri, A. Xie, M. Camblor, I. V. Kolmanovsky, and D. S. Bernstein, "Inertia-free attitude control laws based on rotation matrices for spacecraft with torquers, thrusters, and wheels," *Symposium on Estimation, Navigation, and Spacecraft Control*, pp. 322–391, 2012.

[53] T. van den Broek, "Formation control of unicycle mobile robots," Master's thesis, Eindhoven University of Technology, Dynamics and Control Group, Department of Mechanical Engineering, 2008. Eindhoven, The Netherlands, MSc Thesis, DC 2008.131.

[54] A. Khairuddin, M. Talib, and H. Haron, "Review on simultaneous localization and mapping (SLAM)," in *2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pp. 85–90, Nov 2015.

[55] T. Li, S. Sun, and J. Duan, "Monte carlo localization for mobile robot using adaptive particle merging and splitting technique," in *The 2010 IEEE International Conference on Information and Automation*, pp. 1913–1918, June 2010.

[56] L. Chen, H. Hu, and K. McDonald-Maier, "EKF based mobile robot localization," in *2012 Third International Conference on Emerging Security Technologies*, pp. 149–154, Sept 2012.

[57] B. van Aert, "Control and coordination algorithms for autonomous multi-agent quadrotor systems," Master's thesis, Eindhoven University of Technology, Control Systemm Technology Group, Department of Mechanical Engineering, 2016. Eindhoven, The Netherlands, MSc Thesis, CST 2016.070.

[58] "Specification of parrot ar.drone 2.0." `https://www.drones.nl/drones/parrot-ar-drone-2-0-power-edition/specs`. Accessed: 28-09-2018.

[59] "Tyhe node copter hacker guide tips and tricks." `http://www.nodecopter.com/hack`. Accessed: 28-09-2018.

[60] J. Malinen, "ARdrone-wpa2 wpa2 support for ar drone 2.0." `https://github.com/daraosn/ardrone-wpa2`. MIT, Accessed: 28-09-2018.

[61] S. Marx, "High performance drone target pursuit," Master's thesis, Eindhoven University of Technology, Control System Technology Group, Department of Mechanical Engineering, Control System Group, Department of Electrical Engineering, 2017. Eindhoven, The Netherlands, CST 2017.17.

Appendix A

Towards experimental validation

In this Appendix, we provide some steps for the development of a system of multiple quadrotor UAVs that operates in the real-world. In order to test the real-world behavior, we want to execute experiments in the future with at least two quadrotor UAVs. Some necessary steps have been carried out and the learning points that followed from the resulting system are included at the end of this Appendix.

A.1 Experimental setup

In this section, we introduce a localization algorithm for multiple agents and design a network architecture that allows us to work with multiple agents at once. After that, we introduce the hardware that is used for the experiments.

A.1.1 Multi-agent localization

In order to control the quadrotor UAVs in a real-world scenario, we need to have some sort of localization of the quadrotors. Several different on-board techniques exist for doing this by using the on-board sensors of the device, like SLAM (simultaneous localization and mapping) [54], particle filters like adaptive Monte Carlo localization (AMCL) [55] or sensor fusion based on the extended Kalman filter (EKF) [56]. However, since our current setup is short on processing power, we aim to use an external localization technique. During the experiments, we want to use a single-camera system, while in the outdoors this can later be done by something like GPS (global positioning system). It is important that the external localization algorithm can make a clear and robust distinction between the different agents, since it can be catastrophic to send the position of a specific quadrotor UAV to a wrong agent.

An answer to multi-agent and robust localization has been found in [57], where experiments with multiple agents have been executed based on specific marker-patterns. The different marker-patterns that are used for localization are included in Figure A.1. When starting with the algorithm from [57], change the number of drones k from 1 to a range that describes the set of agents in the system. Furthermore, make sure that the UDP-send function only sends localization data from agent i to the specific address of quadrotor i. In order to send data to all involved quadrotors, all quadrotors have to be connected to the same network. Although we now have a system that locates multiple quadrotor UAVs, there are many potential situations that provide a mix-up of the ID numbers. For an imaginable system of two quadrotor UAVs with ID 1 and 2 from Figure A.1, any one of the LEDs from quadrotor 2 that is not seen by the algorithm causes a mix-up in the ID numbers. Therefore, additional robustness is created by making sure that when not all LEDs are located the new location of the quadrotor UAVs are provided with the same ID numbers as

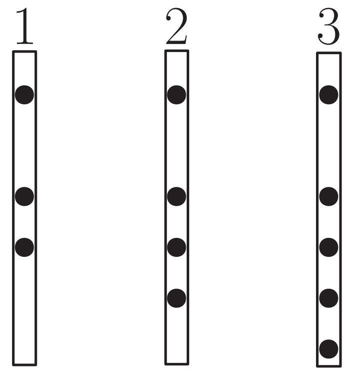


Figure A.1: Schematic representation of the markers used for localization and identification with the algorithm from [57]. The black dots represent LED lights, the bars represent LED-strips and the numbers are the automatic ID numbers.

the closest quadrotor in the previous time step, this is schematically shown in Figure A.2. Notice that as a result of the absolute minimum sample frequency of 5 Hz and absolute maximum flight speed (but usually much lower) of 5 m/s [58] combined, the position vector between the current and previous step for a specific drone is usually smaller than the vector between the current location and the previous location of any of the other quadrotor UAVs. Under normal circumstances the length between two samples of the same quadrotor is thus less than 1 m, while we intent have much more space than 1 m between any two drones. Notice that the sampling frequency can be increased by dedicating more computational power to operating the algorithm, but also by only considering a part of the image based on the last known position of the quadrotor UAVs.

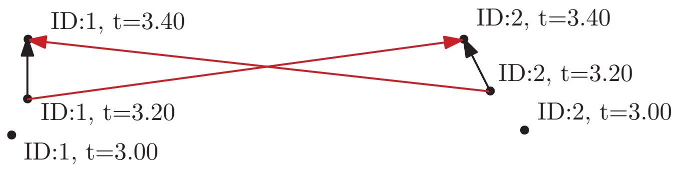


Figure A.2: Schematic 2D projection of the position of the quadrotor UAVs in the xy-plane as provided by the localisation algorithm with the overhead camera. The ID number update procedure is indicated with the black vectors denoting the right ID decision and the red vectors representing the wrong decision for ID numbers.

A.1.2 Communication architecture

The Parrot AR 2.0 quadrotor UAV features an on-board DHCP server. This means that in order to communicate with multiple quadrotors, we have to develop a different network architecture. When we do not change anything, the supervisor has to connect to all quadrotors separately, which means that the number of agents is very limited, since normally every network card can only connect to a single WiFi network. One way to solve this problem is to use an additional DHCP server that features a wireless access point (AP), which are both services that most common routers have included. By aborting the original networking process on the quadrotor and simultaneously making the quadrotor a client to an open network with specific service set identifier (SSID) and internet protocol (IP) address, we can connect the drone to the AP of choice. We use [59]

```
killall udhcpd; iwconfig ath0 mode managed essid SSID; ifconfig ath0
    DRONE_IP netmask 255.255.255.0 up;
```

of which SSID has to equal the SSID of the AP and DRONE_IP has to equal an arbitrary IP address that is free and within the range of the router. Notice that in order to communicate with the drone after booting the device, we have to connect to the network of the quadrotor in order to execute these commands. The removal and installation thus has to happen simultaneously since the used network for communication is terminated by the commands. Furthermore, notice that we are only able to connect to an open network, in order to connect to a WPA secured network additional firmware is required [60]. The resulting Network architecture is schematically shown in Figure A.3. Notice that if the router also has an internal network switch, which most of the commonly used

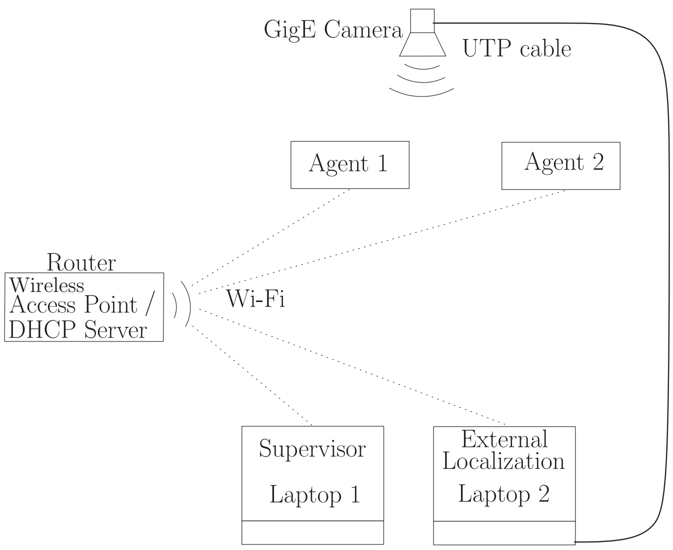


Figure A.3: Schematic representation of the proposed system architecture in which agents are connected to a DHCP server/wireless access point (WAP) combination. The network allows a connection between the connected devices and allows scalability for multiple agents and support hardware.

routers have, we can easily connect the router to the internet and we can easily connect the GigE camera to the router by a wired connection. However, for the GigE camera, we prefer to connect the camera to the laptop directly to prevent from a vast amount of data being transmitted over the wireless network unnecessarily; only the resulting locations are being transmitted over the wireless network by laptop 2.

A.1.3 Hardware

For the experiments we use two identical (apart from manufacturing differences) quadrotor UAVs that are commercially available: the Parrot AR.Drone 2.0. A list of specifications is included in Table A.1.

Table A.1: Specifications of the Parrot AR.Drone 2.0.

Dimensions	$l_x \times l_y \times l_z = 517 \times 451 \times 50$ mm^3 [58]
Mass	456 g including LED-strip [19] 420 g out of the box with hull [58]
Processor	1 GHz 32 bit ARM Cortex A8 Processor [58]
Memory	1 GB DDR2 RAM (200MHz) [58]
WiFi	Own network (DHCP host) max range of 50 m [58]
Motors	4 brushless inrunner moters, max 14.5 W and 28.500 RPM [58]
Battery	1500 mAh 3S (3-cell) Li-Po [58]
Max velocity	5 m/s [58]
Sensors	GPS, front-facing camera (1280×720 @ 30 fps, fov 90° diagonal), down-facing camera (320×240 @ 60 fps, fov 64° diagonal), barometer (precision 10 Pa), 3-axis magnetometer (precision 6°), 3-axis gyroscope (precision $\pm 2000°/s$), 3-axis accelerometer (precision ±50 mg) and ultrasonic distance sensor (range 0.2 m - 6 m) [20]

Additionally, the quadrotors have a cool-white LED strip mounted on top for localization with the specific lay-out as in Figure A.1. The camera that is used for localization is the GigE GE1900 camera [57], operating at 30 fps and full HD. The router that is used for the experiments is the D-link DIR 600 router. The laptop that is used for execution of the control system is the HP EliteBook 8570w Workstation, running Windows 10 and Matlab 2016b. The second laptop is used for localization with Matlab 2015b is the Asus R500V, also running Windows 10.

A.2 System-architecture for experimentation

The required relations for control of the multi-agent system require more calculation power than the single agent system from [19], while even then the central processing unit (CPU) was a limiting factor. Therefore, we are unable to operate the multi-agent quadrotor system entirely on the on-board micro processors. In order to overcome this problem, we develop a system that operates somewhat hybrid, based on the tooling presented in [61], by outsourcing the input calculation to a central ground-unit. This means that each agent gathers sensor data and bundles this data into an ordered bit-package that is sent to the central ground-unit over the wireless network hosted by a router. The ground unit clusters the data of each of the agents with an extra additional localization source from the top camera based on [57], also obtained over the wireless network. The now reconstructed states are used to obtain the required relation to provide control action, by also computing all reference and desired states for that time step. The calculated inputs f_i and τ_i are sent to each of the corresponding drones in the network. On board of the quadrotor UAVs, these inputs are then transformed to PWM signals based on motor constants that are obtained trough calibration, as in [61]. On-board of the quadrotor UAV, the PWM signals are also saturated one more time in order to make sure that the PWM signals do not exceed the maximum allowable value. Notice that it is important to make sure that this saturation bound is not reached, since it will change the control action drastically, since both the thrust magnitude and torques are effected. Because of that, one possible improvement is to change this saturation so that it scales the force magnitude and torques directly in order to make the PWM signals fit in the domain. This provides the benefit that we are still able to control the attitude during saturation so that we have full control over the quadrotor at all times. For now, we accept the saturation of PWM signals directly, since with the incorporated saturation terms and gains in the control laws we can make sure to keep within the saturation bounds of the PWM signals.

The modeling is done with Euler angles, since because of the use of a central ground unit computational power is not scarce. In future on-board implementations, however, a quaternion approach should be considered, as it has certain benefits with respect to Euler angles, as explained in Section 2.1.2; a quaternion implementation requires less storage and less calculation power for rotation chaining than the rotation matrix alternative.

The tooling from [61] is used for calibration, communication and actuation functionality and the controllers from Chapter 4 are implemented for two drones in order to control the drones.

A.3 Experiments and learning points

After calibration, we tested the system that is described in the previous section in experiments. First, we tested the communication by communicating known data between clients of the network. We were able to verify first on bit-level that we are able to transfer binary data. After that, we extended to sending bytes that represented singles and doubles from measurement data. After this verification of the data transfer, we connected the sensor data feed and were able to directly send the data that was required for operation. This means that we were able to access the measurement data to compute the control input. The controller was already tested in simulations and therefore we now have all the building blocks needed to operate the system. However, during real time operation we noticed some shortcomings in the proposed setup which we would like to include here as learning points for future work.

- The sampling frequency of the IMU is 400 Hz, assuming that the wireless connection is not limiting, we can operate the system at up to 400 Hz. However, the only source for the absolute position in x and y is the overhead camera which can currently only run steady at a maximum of 5 Hz. This position update frequency of 5 Hz is a huge limiting factor in the feedback control and therefore either the sampling frequency should be improved or some kind of observer should be included to provide a better estimate of the position between measurement samples. One possible method to increase the sampling frequency of the top camera is to only search in a part of the image for the quadrotor UAV given their previous position, direction of motion and the amount of time that has past. In that way the image search algorithm can operate much quicker which is possible since the camera can run up to 30 fps.

- There is a considerable time delay induced by the specific system configuration between measurements and actuation. For example, whenever the localization algorithm is ready to accept a new image it takes the last available image, which can be as old as 1/5 seconds because of the frame rate which can maximal be 30 fps is decreased to the frequency of the image searching algorithm. Then, the image search algorithm takes place which takes 1/5 seconds. The location data is then transmitted to the laptop that bundles this data with sensor data from the quadrotor UAV that is slightly less delayed since it operates at 100 Hz and transmits at 100 Hz. The laptop then calculates the new inputs. Since the program runs at 100 Hz, the data is read after a maximum of 1/100 s as soon as the new time step starts. During next sample of 1/100 s, the input is calculated and send towards the microchip on the quadrotor UAV. Since the wireless connection with the quadrotor operates at 100 Hz, after a maximum of 1/100 s the new inputs are read and applied for 1/100 s. Thus, in the worst case scenario the quadrotor UAV is provided with inputs that are calculated based on measurements that are taken 0.43 s before, meaning that the delay can be up to 43 samples when operating the quadrotor UAV at 100 Hz.

- The on-board saturation on the microprocessor of the quadrotor UAV simply saturates each if the four motor signals to the chosen maximum whenever the controller provides motor signals that are higher. However, this means that the desired input is not applied at all, since the torque as a result of the difference between the motor signals is canceled because all of the motors can potentially be saturated to the same value. It is beneficial to only scale the overall magnitude an keep the desired torque in tact whenever possible, because then the quadrotor is able to best control its attitude at all times.