Distributed Formation Control of Multi-Agent Systems

Vom Promotionsausschuss der
Technischen Universität Hamburg
zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

von

Marcus Bartels

aus

Pinneberg, Deutschland

2020

1. Gutachter: Prof. Dr. Herbert Werner

2. Gutachter: Prof. Dr.-Ing. Uwe Weltin

3. Gutachter: Prof. Dr. Ya-Jun Pan

Vorsitzender des Promotionsverfahrens: Prof. Dr.-Ing. Robert Seifried

Tag der mündlichen Prüfung:

19. September 2019

ISBN 978-3-8439-4449-6

DOI 10.15480/882.2843

Dedicated to my family

Acknowledgments

This thesis is the result of my research during four years of employment at the Institute of Control Systems, Hamburg University of Technology, from 2012 to 2017. During this period and also afterwards until the final completion of this thesis, my work has been supported in manifold ways by various people such as colleagues, students and researchers. To all of them, I would like to express my gratitude.

First of all, I would like to thank my advisor Prof. Dr. Herbert Werner for accompanying my research work with his broad knowledge and providing valuable hints and guidance when needed. From the beginning on, he put a high amount of trust in me. Besides the technical guidance, he provided valuable assistance in presenting and teaching knowledge, which also made the time at the institute an important experience from this point of view. At this point, I also want to thank the further members of my doctoral exam committee, Prof. Ya-Jun Pan and Prof. Uwe Weltin, for examining my thesis and Prof. Robert Seifried for chairing the examination board.

Very important to mention is my gratitude towards my colleagues, who made the time I spent at the institute a very pleasant time and were willing to give support when needed. In particular, I want to thank Simon Wollnack, who provided an important stimulus to the descriptor-based approach presented in this thesis. With his initial idea and further good hints during the development of this concept, he provided valuable support to this work. Furthermore, I would like to mention Christian Hoffmann, Annika Eichler and Ulf Pilz, who did not only create an important scientific basis for this thesis, but also provided valuable support to my work.

At least the experimental part of this thesis would presumably not exist without the substantial contributions of numerous students, who supported this research with their Bachelor or Master thesis, project work or student job and spent a lot of effort, enthusiasm and creativity in their tasks. Among those, I would like to mention Florian Kunkel, Jonas Farnbacher and Peter Paulsen, who especially advanced the experimental research with their valuable contribution. During the experimental work, I also learned to appreciate the great technical support by Klaus Baumgart and Herwig Meyer, who always provided deliberate and sometimes creative solutions to the encountered technical challenges. Besides this, I also want to express my thanks towards the secretaries of the institute for their helping hands whenever needed.

Finally, I want to express my big thanks to my family and friends, who supported and encouraged me all the time.

Contents

Chapter 1

Introduction

Autonomous mobile robots have attracted the interest of researchers for decades. Their benefit is to efficiently fulfill various tasks, which human workers or manned vehicles could only accomplish in a very inefficient manner or which would require to expose humans to dangerous environments. While unmanned ground transportation systems are already widely used in industry and logistics, the recent development of microelectronics and battery technology even made Unmanned Aerial Vehicles (UAVs) interesting for commercial usage and triggered various projects in many different fields [BMWi, 2019; Shakhatreh et al., 2018]. As such, the systematic application of UAVs to search and rescue missions has been introduced by various organizations [Niemann, 2016; aerzteblatt.de, 2018; Grogan et al., 2018]. Further fields of commercial application are delivery logistics [DHL, 2016], inspection for maintenance [Ascending Technologies, 2017b], aerial photography, agriculture [Mazur, 2016] and traffic monitoring [Kanistras et al., 2013]. An especially emerging technology in this field are autonomous multi-rotor helicopters. Autonomous underwater vehicles (AUVs) are another promising technology with high attention in research [Hackbarth et al., 2015; Klein et al., 2008].

1.1 Multi-Agent Systems

To use the benefits of autonomous mobile robot technology for large and complex tasks exceeding the capabilities of single robots, extensive research is done on multi-agent systems. In such systems, multiple autonomous robots (considered as *agents*) are working together in a coordinated manner to jointly accomplish the given tasks. Inspired by the swarm behavior observed in nature e.g. with flocks of birds, schools of fish or swarms of bees, teams of self-organized robots are an important subject to research. The term *self-organized* in this context means that each agent is equipped with a control unit which enables it to interact with its team mates in a coordinated manner, such that the common goal can be achieved without relying on a central authority. With this approach, several advantages are aimed at:

- **Low vulnerability** As the intelligence required to accomplish the task is distributed, the system does not depend on the operability of a single unit.
- **Distribution of computational load** The computational load is distributed among the agents, such that, especially for large systems, the need for an extremely powerful central computer is avoided.
- **Scalability** The number of agents can be adapted to the complexity of the task. For this purpose, distributed control schemes are desired which can be designed independently of the actual number of agents.
- **Low controller design effort** Instead of a central high-complexity controller governing a large system, only a controller with the complexity of a single agent is designed and deployed to all agents.

The typical system setup is a group of mobile robots or vehicles, which are equipped with local control units and communication devices to exchange information over a wireless network. In most parts of this thesis, aerial vehicles such as quad-rotor helicopters are considered. An important scenario in the research field of multi-agent systems and adopted here is decentralized formation control [Wolfe et al., 1996; Fax and Murray, 2004; Lafferriere et al., 2005]. This scenario considers the goal of robots achieving or maintaining a pre-defined geometrical formation.

For applicability even under harsh conditions such as expected for search and rescue missions, a robust system design is needed. In particular, link failures and desired changes of the communication topology as well as agent failures and the ad-hoc integration of additional agents into the running system are to be covered. Accordingly, scalability and robustness against topology changes are desirable properties of the control system.

1.2 Current State of Research

In recent years several concepts and methods have been proposed for cooperative control of multi-agent systems. In an early work in this field, Tsitsiklis and Athans [1984] examine cooperative decision processes. In the following years, different aspects of multi-agent systems such as flocking [Reynolds, 1987; Olfati-Saber, 2006], coordination [Jadbabaie et al., 2003], consensus processes [Xiao and Boyd, 2004; Olfati-Saber and Murray, 2004; Olfati-Saber et al., 2007] and formation control [Wolfe et al., 1996] gained interest in research. An important basis for distributed formation control has been laid by Fax and Murray [2004]: Describing the interaction topology as a graph, marginal stability of a generic multi-agent system is found equivalent to the stability of *modal subsystems*, which consist of a single augmented agent model parameterized by the eigenvalues of the interconnection matrix. Based on this, a cooperative control scheme is proposed providing robust stability for an arbitrary but known topology. In subsequent publications this framework is re-interpreted as a Linear Fractional Transformation (LFT) of a group of agents as nominal system and a static interaction matrix. In Popov and Werner [2009] the

interaction matrix is diagonalized and its eigenvalues are handled as uncertainty. A similar and more general framework is provided by Massioni and Verhaegen [2008, 2009] as the *decomposable systems* framework, which considers a direct interaction of the controllers. Cooperative control of Linear Parameter-Varying (LPV) systems for state consensus is treated in Seyboth et al. [2012] considering directed topologies.

Another important concept proposed in Fax and Murray [2004] is to use an *Information Flow Filter* (IFF) providing a joint determination of reference positions to be tracked. A simplifying interpretation of this setup, referred to as *consensus-based approach*, is presented in Pilz et al. [2011], decomposing it into a formation consensus algorithm and a local position control scheme. This allows a separate design of the local position controller and the information flow filter handling the interaction of the agents. Based on this approach, in Pilz and Werner [2012b], Popov [2012] and Pilz [2013] a general information flow framework with a combined controller is proposed describing both local control and interaction between agents - respectively their controllers. Within this framework, both cooperative and consensus-based approach can be seen as special cases.

To incorporate robustness against topology changes into controller synthesis and at the same time include performance specifications, two major approaches have been developed: In Popov and Werner [2012] a property of the adjacency matrix is used together with the Small Gain Theorem (SGT) to formulate a condition on the ℓ_1 norm of each subsystem, that implies stability of the whole multi-agent system in case of arbitrary communication delays. To synthesize a robust controller guaranteeing this condition *a-priori* together with optimal performance, a combined $\mathcal{H}_\infty/\ell_1$ procedure is proposed in Pilz and Werner [2012a].

The second approach is based on dissipativity, adopting a common concept from the theory of LPV systems. After a first dissipativity-based approach to distributed control has been proposed in Scorletti and Duc [2001], a decomposed synthesis approach based on [Scherer, 2001] is proposed in Langbort et al. [2004] using a general framework of physically coupled subsystems. In Massioni and Verhaegen [2010] the full-block $\mathcal{S}$-procedure (FBSP, Scherer [2000, 2001]) is applied to the decomposable systems framework. This yields a dissipativity-based controller design technique with $\mathcal{H}_\infty$ performance guarantee and allows arbitrary unknown but undirected and time-invariant topologies with non-scaling problem size. Extensions towards directed and time-varying topologies are given in Hoffmann et al. [2013] considering switching topologies and Eichler et al. [2013a,b] using decompositions by means of complex eigenvalues and Jordan blocks. The same authors provide further extensions regarding $\mathcal{L}_2$-optimal controller design for gain-scheduled control of decomposable Linear Parameter-Varying (LPV) systems [Hoffmann et al., 2013; Eichler et al., 2013b] as well as heterogenous groups with directed links Hoffmann et al. [2014, 2015]. For the latter case, an alternative approach is given by Massioni [2014] using singular value decomposition. An improved method to decompose multi-agent systems is proposed in Hoffmann and Werner [2015, 2017], which allows arbitrary directed topologies. The effects of decomposing the synthesis problem on the $\mathcal{L}_2$ performance bounds are studied in Eichler and Werner [2013].

Recent contributions to this field encompass, among others, a distributed finite-time

global optimal control approach for linear time-varying multi-agent systems with fixed and known topologies [Ebrahimi Dehshalie et al., 2018], where the controller is synthesized on subsystem level. Fathian et al. [2018] propose a semidefinite programming approach to design a distributed state feedback formation controller for multi-integrator agents with fixed and known topology, providing robustness against perturbations. An $\mathcal{H}_\infty$-optimal design method for a simultaneous fault detection and control protocol for multi-agent systems is proposed in Zheng et al. [2017], considering Linear Time-Invariant (LTI) agents with a fixed and undirected topology. One of the most recent publications treating the distributed formation control problem of multiple UAVs is Zou et al. [2018]. It proposes a sub-optimal hierarchical controller design strategy and Lyapunov-based stability criteria for a fixed number of non-linear UAVs connected by a switching topology.

1.3 Scope and Contribution

The control problem addressed in this thesis encompasses systems of physically decoupled agents, which are equipped with a local controller and connected by a data transmission network. For this class of systems, the design of a scalable and topology-independent distributed controller is considered, which provides stability and optimal performance. In this context, synthesis techniques are considered in which a controller is synthesized once at sub-system complexity and deployed to the agents of the multi-agent system. While many publications in this field assume a known and/or fixed communication topology, this thesis aims at allowing arbitrary, undirected and time-varying topologies as well as an arbitrary number of agents (scalability). The considerations of this thesis are mostly limited to LTI agents, while the concept of LPV control is used to model the interaction. As reference scenario, mostly decentralized formation control is considered, i.e. the goal of robots achieving or maintaining a pre-defined geometrical formation.

This thesis is inspired by the works of Fax and Murray [2004], Popov [2012], Pilz [2013], Hoffmann [2015] and Eichler [2016], where important fundamentals of this field are given and promising controller synthesis strategies are provided. For these achievements, the scope of this thesis encompasses an analysis and comparison of the control architectures and synthesis approaches. As second main contribution, in this thesis a new controller design approach is introduced using the framework of descriptor systems to describe multi-agent systems.

The scientific contributions of this thesis are summarized as follows:

1. A global system model is deduced and analyzed for time-invariant multi-agent systems. Using the distributed structure of this model, statements on the global closed loop eigenvalues (Theorems 2.2.1 and 2.2.2) and the final meeting position of a rendez-vous process (Theorem 2.2.3) are derived in terms of agent model and interaction matrix. Furthermore, the response to assigning a desired formation is expressed in terms of the global system dynamics and the relative part of the formation reference (Theorem 2.3.2). Though not covering the full set of systems considered in this thesis, these results are important to the general understanding

of the behavior of multi-agent systems and the roles agent dynamics and interaction topology play to their dynamics.

2. The problem of designing an information flow filter for consensus-based formation control in the sense of Pilz et al. [2011] is examined and design objectives are formulated. An extension of the consensus-based control scheme is introduced, which enables the agents to react on a disturbance acting on a neighbor agent, but maintains the performance advantages with respect to single-loop cooperative architectures. In contrast to other combined design approaches, this scheme still considers separate synthesis of consensus and local control. These findings have been published in Bartels and Werner [2014].

3. A simulation study is conducted to compare two central controller synthesis approaches, one based on the Small Gain Theorem and one adopting an LPV control technique. This study reveals performance benefits of the LPV-based technique, but also uncovers an effect of the underlying synthesis approach which limits the exploitation of the benefits from this technique. In order to achieve a meaningful comparison among different synthesis techniques, an evolutionary tuning technique is developed to optimize controllers across different synthesis techniques with respect to common criteria in a common scenario. Central results of this work have been published in Bartels and Werner [2016].

4. This thesis introduces a representation of multi-agent systems as descriptor systems, combining temporal states modeling the agent dynamics with algebraic relations describing the interaction. This representation provides additional degrees of freedom to the controller design problem. As a central result, for this representation a controller design approach is formulated in the Theorems 4.2.1 and 4.2.2 as LMI problem, based on an existing approach for descriptor LPV systems. Its solution allows the synthesis of a distributed $\mathcal{L}_2$-optimal controller, providing scalability and robustness against topology changes. An additional benefit is given by the ability to cover formation control of agents modeled as descriptor systems.

5. The effectiveness of the descriptor-based synthesis approach is shown by means of a numerical example considering formation control of a group of quad-rotor helicopters. A numeric example considering the coordination of electric circuits with inductive load is used to illustrate the applicability to systems of multiple descriptor agents.

6. Two testbeds for experimental assessment and validation of theoretical concepts on distributed control of swarms of UAVs have been designed and realized. While one testbed realizes a scenario of fully autonomous quad-copters interacting with each other, the second testbed combines small-scale hardware for indoor testing with a centralized platform for emulating distributed control. For challenges faced with respect to the experimental platforms, different solutions have been examined and insight about their applicability have been achieved. This especially applies to the localization of quad-copters. For both testbeds, first experimental results have been generated, which demonstrate the applicability of these testbeds for experimental

research on the considered control aspects. In particular, the extended consensus-based control scheme has been tested experimentally, at which the expected outcome has been confirmed by the actual results.

Chapter 2

Dynamics of Multi-Agent Systems

In this chapter, the dynamic behavior of multi-agent systems is examined with a focus on the roles of the agent dynamics and the interaction topology on the response of the global multi-agent system. As result, the steady state response of an LTI multi-agent system with fixed topology is deduced analytically. Furthermore, the influence of the topology on the closed-loop performance is examined, including weighted links.

2.1 Preliminaries

2.1.1 General framework of Multi-Agent Systems

In the following, a general framework is introduced, which is used to describe multi-agent systems. This general model is shown in Fig. 2.1.1. It is based on a framework proposed in Hoffmann et al. [2015] for analysis of interconnected LPV systems.

The general multi-agent system is considered to consist of N agents, which exchange data

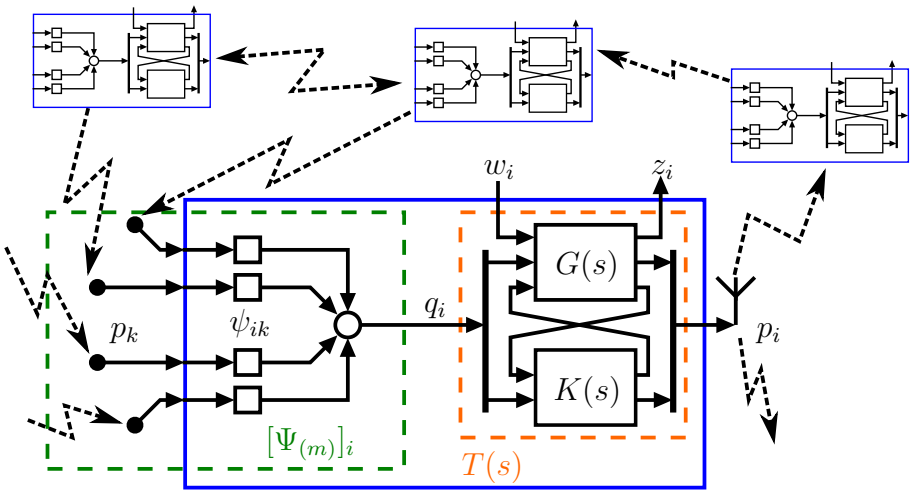


Figure 2.1.1: Interconnection of agents in a multi-agent system

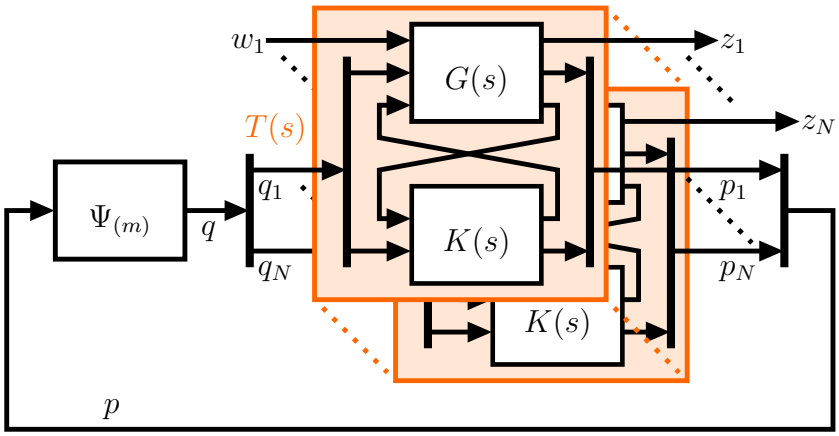


Figure 2.1.2: Block diagram of a multi-agent system

among each other to coordinate their actions (therefore referred to as *coordination data*). Each agent i sends out the data p_i and receives data p_k from some agents k. Technically, the index of an agent can be seen as a unique identifier assigned to the agent and stored on-board. We assume that each data package contains the identifier of the sender agent, such that the receiving agent i knows the identifiers k of the agents it receives data from. These agents (not including agent i itself) are referred to as *neighbors* of agent i, mathematically represented by the node set N_i.

Within agent i, the incoming data from the neighbor agents are used to compute the *network input* signal q_i as a linear combination

$$q_i = \sum_{k \in N_i} \psi_{ik} p_k. \tag{2.1.1}$$

The weighting factors ψ_{ik} form the pattern matrix $\Psi \in \mathbb{R}^{N \times N}$, which describes the interconnection between the agents and can be interpreted as the weighted adjacency matrix of a graph representing the interconnection. Further details about this representation are given in the following Section 2.1.2. A block diagram of the multi-agent system with matrix representation of the interaction is shown in Fig. 2.1.2.

In addition to the communication signals, each agent has an exogenous input w_i, which can contain both a relative reference position r_i within a formation and a disturbance. For representing performance, an extra output z_i is defined.

As shown in Fig. 2.1.2, each agent $T(s)$ can internally be considered as a feedback connection of a (generalized) plant $G(s)$, which contains the plant dynamics $P(s)$, and a controller $K(s)$. Each agent i is modeled by a state space realization

$$G: \begin{pmatrix} \dot{x}_i \\ z_i \\ p_i^P \\ v_i \end{pmatrix} = \left[\begin{array}{c|ccc} A & B_w & B_q & B_u \\ \hline C_z & D_{zw} & D_{zq} & D_{zu} \\ C_p & D_{pw} & D_{pq} & D_{pu} \\ C_v & D_{vw} & D_{vq} & D_{vu} \end{array}\right] \begin{pmatrix} x_i \\ w_i \\ q_i^P \\ u_i \end{pmatrix}, \tag{2.1.2}$$

$$K: \begin{pmatrix} \dot{x}_i^K \\ u_i \\ p_i^K \end{pmatrix} = \left[\begin{array}{c|cc} A^K & B_v^K & B_q^K \\ \hline C_u^K & D_{uv}^K & D_{uq}^K \\ C_p^K & D_{pv}^K & D_{pq}^K \end{array}\right] \begin{pmatrix} x_i^K \\ v_i \\ q_i^K \end{pmatrix}, \tag{2.1.3}$$

where $x_i, x_i^K \in \mathbb{R}^n; p_i^P, q_i^P \in \mathbb{R}^m, p_i^K, q_i^K \in \mathbb{R}^{m_K}; u_i \in \mathbb{R}^h; v_i \in \mathbb{R}^l; w_i \in \mathbb{R}^p; z_i \in \mathbb{R}^q$.

For the local closed loop subsystem of agent i we obtain the system representation by the Redheffer star product [Timotin, 1995] of G and K:

$$T = G * K: \quad \left(\begin{array}{c} \dot{x}_i \\ \dot{x}_i^K \\ \hdashline p_i^P \\ p_i^K \\ \hdashline z_i \end{array}\right) = \left[\begin{array}{c:c:c} A^{cl} & B_q^{cl} & B_w^{cl} \\ \hdashline C_p^{cl} & D_{pq}^{cl} & D_{pw}^{cl} \\ \hdashline C_z^{cl} & D_{zq}^{cl} & D_{zw}^{cl} \end{array}\right] \left(\begin{array}{c} x_i \\ x_i^K \\ \hdashline q_i^P \\ q_i^K \\ \hdashline w_i \end{array}\right) \tag{2.1.4}$$

In this general framework the transmitted data p_i and the averaged incoming data q_i can be considered to consist of both measured plant data (upper index P) and data generated by the controller (upper index K) or either of them.

Unless stated otherwise, for the signals in this framework we use symbols with lower indices for signals of individual agents and symbols without lower indices for bundles of the respective signals for all agents, i.e. $x = \mathrm{vcat}(x_1, \cdots, x_N)$. Using the kronecker-extended notation $\hat{M} = I_N \otimes M$ for the block-diagonal concatenation of a matrix M, a group of identical and independent systems $G(s)$ is represented by the *aggregated* system $\hat{G}(s) = I_N \otimes G(s)$ by block-diagonal concatenation of the system matrices:

$$\hat{G}(s) = \left[\begin{array}{c|c} I_N \otimes A & I_N \otimes B \\ \hline I_N \otimes C & I_N \otimes D \end{array}\right] = \left[\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & \hat{D} \end{array}\right] \quad \text{for } G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \tag{2.1.5}$$

Conversely, for conveniently describing the interaction between agents each having multiple interaction channels, the notation $M_{(n)} = M \otimes I_n$ is used. These Kronecker extensions have the following properties:

$$M_{(m)}\hat{N} = \hat{N}M_{(n)} \quad \text{for } N \in \mathbb{R}^{m\times n} \tag{2.1.6a}$$

$$\hat{G}(s)M_{(n)} = M_{(m)}\hat{G}(s) \quad \text{for } G(s) \in \mathbb{C}^{m\times n} \tag{2.1.6b}$$

$$M_{(m)}^{-1} = (M^{-1})_{(m)} \tag{2.1.6c}$$

Details about the Kronecker product are given in Appendix A.3. Using the aforementioned notations, a group of agents (2.1.2) can be described by the aggregated model

$$\hat{G}: \left(\begin{array}{c} \dot{x} \\ \hdashline z \\ p^P \\ v \end{array}\right) = \left[\begin{array}{c:ccc} \hat{A} & \hat{B}_w & \hat{B}_q & \hat{B}_u \\ \hdashline \hat{C}_z & \hat{D}_{zw} & \hat{D}_{zq} & \hat{D}_{zu} \\ \hat{C}_p & \hat{D}_{pw} & \hat{D}_{pq} & \hat{D}_{pu} \\ \hat{C}_v & \hat{D}_{vw} & \hat{D}_{vq} & \hat{D}_{vu} \end{array}\right] \left(\begin{array}{c} x \\ \hdashline w \\ q^P \\ u \end{array}\right) \tag{2.1.7a}$$

$$q^P = \Psi_{(m)} p^P. \tag{2.1.7b}$$

The group of controlled agents (2.1.4) is accordingly described by the aggregated closed loop system model

$$\hat{T}: \quad \left(\begin{array}{c} \dot{x} \\ \dot{x}^K \\ \hdashline p \\ z \end{array}\right) = \left[\begin{array}{c:c:c} \hat{A}^{cl} & \hat{B}_q^{cl} & \hat{B}_w^{cl} \\ \hdashline \hat{C}_p^{cl} & \hat{D}_{pq}^{cl} & \hat{D}_{pw}^{cl} \\ \hdashline \hat{C}_z^{cl} & \hat{D}_{zq}^{cl} & \hat{D}_{zw}^{cl} \end{array}\right] \left(\begin{array}{c} x \\ x^K \\ \hdashline q \\ w \end{array}\right) \tag{2.1.8a}$$

$$q = \Psi^{cl} p, \quad \Psi^{cl} = \Pi^T \begin{bmatrix} \Psi_{(m)} & \\ & \Psi^K_{(m)} \end{bmatrix} \Pi \tag{2.1.8b}$$

where Π is a symmetric permutation matrix used to change the order of the coordination signal vector, i.e. $p = \text{vcat}(p_1, p_1^K, \cdots, p_N, p_N^K)$ is mapped to $\Pi p = \text{vcat}(p_1, \cdots, p_N, p_1^K, \cdots, p_N^K)$.

In this model Ψ^K represents the interaction topology of the controllers. For a practical implementation according to Fig. 2.1.1 only the case $\Psi^K = \Psi$ is meaningful, as this case describes the exchange of agent and controller data by the same communication network. However, for some theoretical considerations it is helpful to consider the more general case.

Combining the aggregated closed loop model (2.1.8a) with the interaction topology described by the interaction matrix Ψ^{cl}, the global interconnected closed loop system is then described by

$$\hat{T}_\Psi : \begin{pmatrix} \dot{x}_{cl} \\ z \end{pmatrix} = \begin{bmatrix} \hat{A}^{cl} + \hat{B}_q^{cl} \Phi^{cl} \hat{C}_p^{cl} & \hat{B}_w^{cl} + \hat{B}_q^{cl} \Phi^{cl} \hat{D}_{pw}^{cl} \\ \hat{C}_z^{cl} + \hat{D}_{zq}^{cl} \Phi^{cl} \hat{C}_p^{cl} & \hat{D}_{zw}^{cl} + \hat{D}_{zq}^{cl} \Phi^{cl} \hat{D}_{pw}^{cl} \end{bmatrix} \begin{pmatrix} x_{cl} \\ w \end{pmatrix} \tag{2.1.9}$$

with $\Phi^{cl} = \Psi^{cl}(I - \hat{\mathcal{D}}_{pq} \Psi^{cl})^{-1}$.

2.1.2 Graph-Theoretical Representation of Interaction

For modeling the interaction of the agents in a multi-agent system, a well-established method is using tools from graph theory [Corfmat and Morse, 1976; Fax and Murray, 2004]. Referring to the previously introduced modeling framework, interpreting the communication network as a graph is straight forward.

In general, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined as a combination of a *node* set $\mathcal{V} = \{v_1, ..., v_N\}, N \in \mathbb{N}$ and an *edge* set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. In the interaction graph of a multi-agent system, the nodes of the graph represent the agents and the edges represent the communication links. This means, for each existing edge (v_i, v_k), the *head* node v_i is receiving data transmitted by node v_k. For each node v_i, the set of neighbors is defined as

$$N_i = \{v_k | (v_i, v_k) \in \mathcal{E}\} \subset \mathcal{V} \tag{2.1.10}$$

with the cardinality $|N_i|$ denoting the number of elements in N_i. This number is also referred to as the *in-degree* d_i^{in} of agent i. Accordingly, the *out-degree* d_i^{out} of node v_i is defined as the number of edges (v_j, v_i) in $\mathcal{E}$ and represents the number of agents receiving the data transmitted by agent i.

Definition 2.1.1. *The graph $\mathcal{G}$ is called undirected, if and only if*

$$(v_i, v_k) \in \mathcal{E} \quad \Leftrightarrow \quad (v_k, v_i) \in \mathcal{E} \tag{2.1.11}$$

holds for all edges in $\mathcal{E}$. Otherwise, the graph is called directed.

The graph is called *strongly connected*, if for every vertex pair (v_i, v_k) a path from node k to node i exists, i.e. a sequence of edges $((v_i, v_{j1}), (v_{j1}, v_{j2}), \cdots, (v_{jn}, v_k))$. For each graph $\mathcal{G}$ a family of associated matrices is defined, which are particularly useful for describing the interaction of agents: The *adjacency* matrix is defined both as unscaled version $\mathcal{A}^0$ and the row-wise scaled version $\mathcal{A}$:

$$[\mathcal{A}^0]_{ik} = \begin{cases} 1 & k \neq i, k \in N_i \\ 0 & \text{otherwise} \end{cases}, \tag{2.1.12a}$$

$$[\mathcal{A}]_{ik} = \begin{cases} \dfrac{1}{d_i^{\text{in}}} & k \neq i, k \in N_i \\ 0 & \text{otherwise} \end{cases} \tag{2.1.12b}$$

Furthermore, the *degree matrix* is defined as the diagonal matrix $\mathcal{D} = \text{diag}(d_1^{\text{in}}, \cdots, d_N^{\text{in}})$ containing the in-degree values of all nodes. Based on that, the Laplacian matrix is defined as $\mathcal{L}^0 = \mathcal{D} - \mathcal{A}^0$ and as row-wise scaled version $\mathcal{L} = I - \mathcal{A}$, leading to

$$[\mathcal{L}^0]_{ik} = \begin{cases} d_i^{\text{in}} & i = k \\ -1 & k \neq i, k \in N_i \\ 0 & \text{otherwise} \end{cases}, \tag{2.1.13a}$$

$$[\mathcal{L}]_{ik} = \begin{cases} 1 & i = k \\ \dfrac{1}{d_i^{\text{in}}} & k \neq i, k \in N_i \\ 0 & \text{otherwise} \end{cases}. \tag{2.1.13b}$$

Algebraic Properties It is a well known result that, based on its construction, each row sum of $\mathcal{L}$ is equal to 0. Accordingly, the vector $\mathbf{1} = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^T \in \mathbb{R}^N$ is the right eigenvector of $\mathcal{L}$ associated with the eigenvalue $\lambda_1 = 0$. As stated e.g. in Fax and Murray [2004] and Popov [2012], by using Gershgorin's circle theorem it can be shown that all eigenvalues of $\mathcal{L}$ are contained in the closed *Perron disk* $\bar{\mathbf{P}}$ defined as

$$\bar{\mathbf{P}} = \{z \in \mathbb{C} || z - 1| \leq 1\}. \tag{2.1.14}$$

According to Ren and Beard [2008] the remaining eigenvalues $\lambda_2, \cdots, \lambda_N$ are non-zero, if the graph is strongly connected.

From $\mathcal{A} = I - \mathcal{L}$ can be inferred that the eigenvalues of the adjacency matrix are located in the closed unit disk

$$\bar{\Theta} = \{z \in \mathbb{C} || z| \leq 1\}. \tag{2.1.15}$$

In addition, the zero eigenvalue of $\mathcal{L}$ is mapped to 1, such that $\mathcal{A}$ has the right eigenvector $\mathbf{1}$ associated with the eigenvalue 1.

For the unscaled versions of the adjacency and the Laplacian matrix, by means of Gershgorin's theorem a disk as possible region of eigenvalues can accordingly be obtained. However, in this case the radius of the disk is determined by the maximum in-degree of the nodes. In contrast, for the scaled version the radius can always be determined as 1, even without knowing the number of nodes. This is a very useful feature to avoid conservatism when guaranteeing scalability of the control scheme and robustness against topology changes.

Weighted Graphs In literature such as Fax and Murray [2004]; Massioni and Verhaegen [2009], the scaled Laplacian as defined in (2.1.13b) is used to describe the interaction of the agents, which corresponds to using the scaled adjacency matrix from (2.1.12b) as Ψ in the model (2.1.7). This choice means that all edges pointing towards a particular node are weighted equally, i.e. all neighbor data are weighted equally in (2.1.1). However, according to several publications [Xiao and Boyd, 2004; Carli et al., 2011; Pilz and Werner, 2013; Eichler and Werner, 2014] it is beneficial to choose unequal weights. This is described by a *weighted communication graph* $\mathcal{G}_W = (\mathcal{V}, \mathcal{E}, \Psi)$ defined by vertex set $\mathcal{V}$, edge set $\mathcal{E}$ and weighting matrix Ψ. The weighting matrix inherits the sparsity pattern of the adjacency matrix $\mathcal{A}$, whereas the non-zero entries w_{ik} specify the weight of the edge (v_i, v_k). Thus, Ψ can be seen as a generalization of the adjacency matrix $\mathcal{A}$, if we impose the following constraint:

$$[\Psi]_{ik} = \begin{cases} \psi_{ik} > 0 & k \neq i, k \in N_i \\ 0 & \text{otherwise} \end{cases}, \tag{2.1.16}$$

$$\sum_{k \in N_i} \psi_{ik} = 1 \quad \forall i. \tag{2.1.17}$$

A corresponding weighted version of the Laplacian is given as $\mathcal{L}^\Psi = I - \Psi$ inheriting the algebraic properties of $\mathcal{L}$. The unweighted graph $\mathcal{G}$ is a special case of a weighted graph with all weights chosen as $\psi_{ik} = 1/d_i^{\text{in}}$. In this work we consider the set $\mathbf{\Psi}$ of all weighting matrix fulfilling (2.1.16,2.1.17). Unless stated otherwise, in this thesis arbitrary elements of $\mathbf{\Psi}$ are considered for describing the interaction in (2.1.7).

2.1.3 First Order Consensus

This section briefly introduces the common first order consensus process, as it is found in various sources such as Mesbahi and Egerstedt [2010], Olshevsky and Tsitsiklis [2011] and Xiao and Boyd [2004]. We assume to have N agents which exchange data over a network with a topology described by the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. For the discrete-time version of this process, data exchange takes place at every discrete time step k. Starting with a vector $x(0)$ of initial values at $k = 0$, the agreement process is defined as

$$x_i(k+1) = \sum_{j=1}^{N} \psi_{ij}(k) x_j(k) \tag{2.1.18}$$

$$\Rightarrow x(k+1) = \Psi x(k) \tag{2.1.19}$$

with the nonnegative interaction matrix $\Psi(k) \in \mathbb{R}^{N \times N}$ fulfilling (2.1.16), (2.1.17) and

$$\rho(\Psi - \mathbf{1}u_1^T) < 1 \quad \text{for } u_1^T \Psi = u_1^T \tag{2.1.20}$$

Assumption (2.1.17), saying that every row of Ψ sums up to 1, implies that the vector $\mathbf{1}$ is a right eigenvector to the eigenvalue 1. This leads to the eigenvector decomposition

$$\Psi = X\Lambda X^{-1} = \begin{bmatrix}\mathbf{1} & v_2 & \cdots & v_N\end{bmatrix} \begin{bmatrix}1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N\end{bmatrix} \begin{bmatrix}u_1^T \\ u_2^T \\ \vdots \\ u_N^T\end{bmatrix} \tag{2.1.21}$$

Having the eigenvalues $1 > |\lambda_2| \geq \cdots \geq |\lambda_N| > 0$, we obtain

$$\lim_{k\to\infty} \Psi^k = X\Lambda^k X^{-1} = \mathbf{1}u_1^T. \tag{2.1.22}$$

This yields the final value of the agreement process as

$$\lim_{k\to\infty} x(k) = \lim_{k\to\infty} \Psi^k x(0) = \mathbf{1}u_1^T x(0). \tag{2.1.23}$$

As this is a multiple of the $\mathbf{1}$ vector, all agents converge to the same value $u_1^T x(0)$, which fulfills the goal of reaching a *consensus*. If and only if in addition $\mathbf{1}^T\Psi = \mathbf{1}^T$ holds, i.e. $u_1 = \mathbf{1}$, the agents converge to the average of their initial values.

Continuous-Time Consensus The first order consensus process can as well be formulated as continuous time process, as stated e.g. in Olfati-Saber and Murray [2004]; Olfati-Saber et al. [2007]:

$$\dot{x}_i(t) = -\sum_{j=1}^{N} \psi_{ij}(t)(x_i(t) - x_j(t)) \tag{2.1.24}$$

$$\Rightarrow \dot{x}(t) = -\mathcal{L}^{\Psi} x(t) \tag{2.1.25}$$

The solution of the agreement process (2.1.25) is given as

$$x(t) = e^{-\mathcal{L}t} x(0). \tag{2.1.26}$$

For the Laplacian, the vector $\mathbf{1}$ is the right eigenvector corresponding to the zero eigenvalue. This leads to the eigenvector decomposition

$$\mathcal{L} = X\Lambda X^{-1} = \begin{bmatrix}\mathbf{1} & v_2 & \cdots & v_N\end{bmatrix} \begin{bmatrix}0 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N\end{bmatrix} \begin{bmatrix}u_1^T \\ u_2^T \\ \vdots \\ u_N^T\end{bmatrix} \tag{2.1.27}$$

with eigenvalues $0 = \lambda_1 < \Re(\lambda_2) \leq \cdots \leq \Re(\lambda_N)$. From Ren and Beard [2008] we know that for each $\lambda \in \mathrm{eig}(\mathcal{L})$ we have $e^{-\lambda t} \in \mathrm{eig}(e^{-\mathcal{L}t})$ with the same eigenvectors. For the case of a simple zero eigenvalue of $\mathcal{L}$ this yields

$$x(t) = Xe^{-\Lambda t}X^{-1}x(0) \quad \text{and } \lim_{t\to\infty} x(t) = \mathbf{1}u_1^T x(0), \tag{2.1.28}$$

because of $\Re(\lambda_i) > 0\ \forall i$, which implies that the contributions of all λ_i except for the zero eigenvalue decay to zero.

Leader-Follower Systems A special case of an agreement process occurs if an agent has the role of a leader, i.e. there is a single agent that does not receive any data, but transmits its data to its neighbors. In case agent i is the leader, the i-th row of Ψ is chosen as the i-th unit vector e_i in order to maintain Assumption (2.1.17). As a consequence, the left eigenvector of Ψ for the eigenvalue 1 is the i-th unit vector (i.e. $e_i^T \Psi = e_i^T$). If the properties (2.1.16),(2.1.17) and (2.1.20) are fulfilled, the agents converge to the initial value of the leader:

$$u_1 = e_i \tag{2.1.29}$$

$$\Rightarrow \lim_{k\to\infty} x(k) = \mathbf{1} u_1^T x(0) = \mathbf{1} e_i^T x(0) = \mathbf{1} x_i(0) \tag{2.1.30}$$

2.1.4 Formation Control

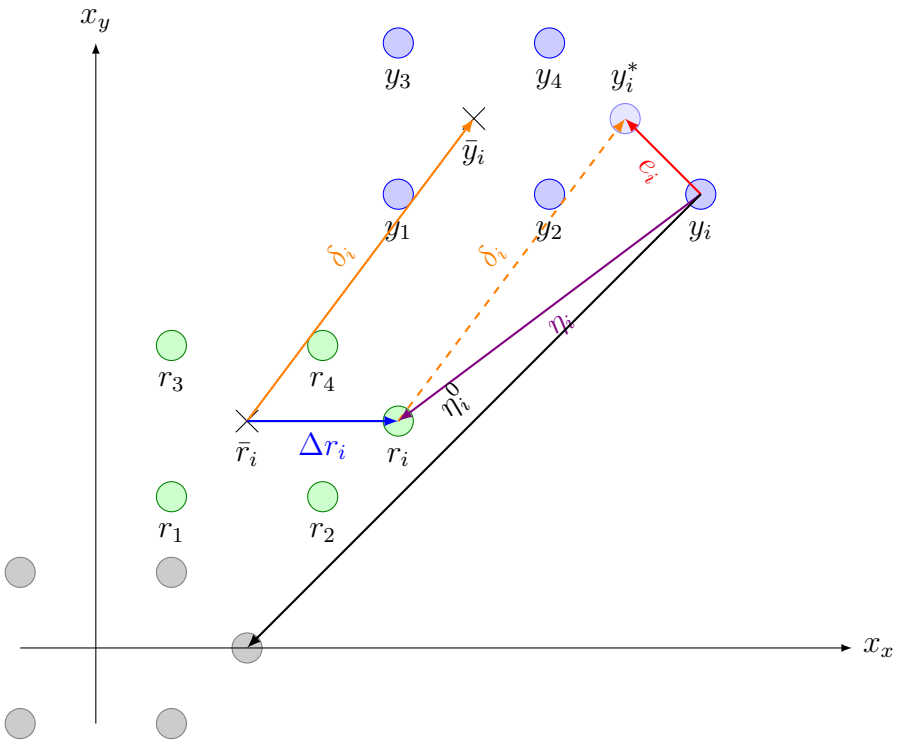


Figure 2.1.3: Displacements and errors in a formation control problem

In a team of mobile agents, the formation control problem considers controlling the relative displacement among the agents in order to achieve or maintain a geometric formation. Here we assume that the desired formation is specified by a reference vector r containing the desired positions of all agents relative to an arbitrary but common reference point. Referring to the geometrical setup shown in Fig. 2.1.3 , we introduce some related quantities (considering $\mathcal{N}_i = \{1, 2, 3, 4\}$):

$$\bar{y}_i = \sum_{k\in\mathcal{N}_i} \psi_{ik} y_k \qquad \text{Center of neighborhood} \tag{2.1.31a}$$

$$\bar{r}_i = \sum_{k \in \mathcal{N}_i} \psi_{ik} r_k \qquad \text{Reference center of neighborhood} \qquad (2.1.31b)$$

$$\Delta r_i = r_i - \bar{r}_i \qquad \text{Relative reference position of agent } i \qquad (2.1.31c)$$

$$\delta_i = \bar{y}_i - \bar{r}_i \qquad \text{Formation displacement} \qquad (2.1.31d)$$

$$y_i^* = \bar{y}_i + \Delta r_i = r_i + \delta_i \qquad \text{Desired position of agent } i \qquad (2.1.31e)$$

$$e_i = r_i + \delta_i - y_i = y_i^* - y_i \qquad \text{Relative error} \qquad (2.1.31f)$$

$$\eta_i = r_i - y_i \qquad \text{Absolute position error} \qquad (2.1.31g)$$

$$\eta_i^0 = \Delta r_i - y_i \qquad \text{Absolute error to shifted reference} \qquad (2.1.31h)$$

According to the definition given in Lafferriere et al. [2005], the agents are said to be *in formation*, if the geometric pattern of the agent positions y_i is equal to that of the corresponding reference positions r_i. This is exactly the case if the position errors η_i and the velocity vectors $\dot{y}_i$ are equal for all agents. Consequently, it is intended to find an agreement on a common vector $\eta_i = \eta_a$. This is expressed by the following formal definition:

Definition 2.1.2 (Formation)**.** *[Lafferriere et al., 2005] Consider a multi-agent system of N agents with output variables $y_i \in \mathbb{R}^m$ denoting their positions and a reference vector $r = \begin{bmatrix} r_1^T & \cdots & r_N^T \end{bmatrix}^T, r \in \mathbb{R}^{mN}$. The agents are in formation at time t if there exist vectors $\eta_a \in \mathbb{R}^m$ and $v_a \in \mathbb{R}^m$ such that*

$$\eta(t) = r - y(t) = \mathbf{1} \otimes \eta_a \quad \textit{for } y = \begin{bmatrix} y_1^T & \cdots & y_N^T \end{bmatrix}^T, \qquad (2.1.32)$$

$$\dot{y}(t) = \mathbf{1} \otimes v_a. \qquad (2.1.33)$$

Definition 2.1.3 (Convergence to Formation)**.** *[Lafferriere et al., 2005] The agents are said to converge to the formation r if there exist $\mathbb{R}^m$-valued functions $\eta_a(t)$ and $v_a(t)$ such that*

$$\lim_{t \to \infty} \eta(t) = \mathbf{1} \otimes \eta_a(t) \qquad (2.1.34)$$

$$\lim_{t \to \infty} \dot{y} = \mathbf{1} \otimes v_a(t). \qquad (2.1.35)$$

For the agents to be in formation, it is however not necessary that η_i is zero for any agent. Thus, bringing η_i to zero is not the goal of formation control. In contrast, the actual location at which the formation is achieved and the common velocity v_a at which the agents are traveling as formation should not be determined by the formation controller. Determining them should be subject to a higher level of mission control not further addressed in this thesis. In this context it is useful to define the *agreement space* with respect to an aggregated vector for N agents each contributing a sub-vector of size p:

$$\mathscr{A} = \{x \in \mathbb{R}^{pN} | \exists\, x_a \in \mathbb{R}^p : x = \mathbf{1} \otimes x_a\} = \text{Span}(\mathbf{1}_N \otimes I_p) \qquad (2.1.36)$$

The complementary space of $\mathscr{A}$ within $\mathbb{R}^{pN}$ can consequently be named *disagreement space*. Every vector $x \in \mathbb{R}^{pN}$ can be decomposed into an agreement part and a disagreement part $\tilde{x}$ (details see Appendix A.4.1):

$$x = \tilde{x} + \mathbf{1}_N \otimes x_a \qquad (2.1.37)$$

$$\text{with } x_a = \frac{1}{N}\sum_{i=1}^{N} x_i.$$

An important property of the extended Laplacian $\mathcal{L}_{(p)}$ is that for fully connected topologies the agreement space is exactly the nullspace of $\mathcal{L}_{(p)}$ as $(\mathcal{L}\otimes I_p)(\mathbf{1}_N \otimes x_a) = \mathcal{L}\mathbf{1}_N \otimes x_a = 0$.

Besides the formulation as an agreement problem on η_i, the formation control problem can also be formulated as a tracking problem with the relative error e_i to be brought to zero. This is equivalent to tracking an individual reference y_i^* defined for each agent i with respect to the positions of its neighbors. For the geometric pattern of the neighbors a center of gravity $\bar{y}_i$ is obtained as weighted sum (2.1.31a) of the neighbor positions, $\bar{r}_i$ is the corresponding center of the reference formation. The individual reference position y_i^* is then easily determined by considering the displacement δ_i between the center of neighborhood and the corresponding reference. Using the interaction matrix Ψ, we obtain

$$\bar{y} = \Psi y, \quad \bar{r} = \Psi r, \tag{2.1.38}$$

$$e = r - y - \Psi(r - y) \tag{2.1.39}$$

2.2 Rendez-vous of LTI Agents

The concept of agreement processes described in Section 2.1.3 for agents with first order dynamics can be extended to the rendez-vous process of a group of general LTI systems. Here the term *rendez-vous* refers to the goal of all agents meeting asymptotically at a common point, which results from team negotiation and is not specified a priori (as examined e.g. in Ren and Beard [2008]). This is a special case of the formation control problem with a zero formation reference $r = 0$, meaning the agents starting with initial state $x(0) = x_0$ are desired to meet at a common point.

In the following, the analysis of consensus processes provided e.g. in Mesbahi and Egerstedt [2010] is used as a starting point to deduce analytic formulation on processes in multi-agent systems with LTI agents and a time-invariant network. Although these results are a straight-forward extension of well-known results, they are helpful to understand important relationships in multi-agent systems.

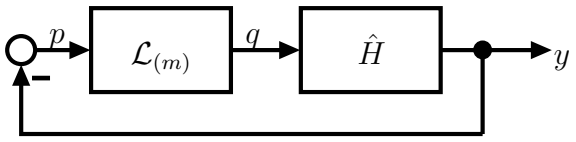


Figure 2.2.1: General Multi-Agent System

We consider the group $\hat{H}(z)$ of N identical agents $H(z)$, which exchange their output data $y_i(k)$, as shown in Fig. 2.2.1. The interconnection is described by the Laplacian matrix $\mathcal{L}$ with property $\mathcal{L}\mathbf{1} = 0$. Here the examination of the closed loop properties is restricted to systems fulfilling the following assumption:

Assumption 2.2.1. *The number of integrator poles of the closed loop system is less than or equal to the number of integrator poles of the open loop agent H.*

Let the agents be described by the discrete-time state space model

$$H : \left\{ \begin{array}{rl} x_i(k+1) & = \Phi x_i(k) + \Gamma u_i(k) \\ y_i(k) & = C x_i(k) \end{array} \right. \tag{2.2.1}$$

$$u(k) = -\mathcal{L}_{(m)} y(k) = -\mathcal{L}_{(m)} \hat{C} x(k) \tag{2.2.2}$$

with $x_i \in \mathbb{R}^n$ and $u_i, y_i \in \mathbb{R}^m$. This yields the closed loop model

$$x(k+1) = (\hat{\Phi} - \hat{\Gamma}\mathcal{L}_{(m)}\hat{C})x(k), \tag{2.2.3}$$

which has the form of the first order consensus process (2.1.19) with closed loop state transition matrix $\Phi_{cl} = \hat{\Phi} - \hat{\Gamma}\mathcal{L}_{(m)}\hat{C}$ at the place of the interconnection matrix. Using the eigenvector decomposition of the closed loop system

$$\Phi_{cl} = \begin{bmatrix} v_1 & \cdots & v_{nN} \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{nN} \end{bmatrix} \begin{bmatrix} u_1^T \\ \vdots \\ u_{nN}^T \end{bmatrix}. \tag{2.2.4}$$

with $|\lambda_1| \geq \ldots \geq |\lambda_{nN}|$, we obtain the rendez-vous state values as

$$\lim_{k\to\infty} x(k) = \lim_{k\to\infty} \Phi_{cl}^k x(0) = \left\{ \begin{array}{cl} \infty & |\lambda_1| > 1 \\ \sum_{i=1}^{\nu} v_i u_i^T x(0) & |\lambda_1| = \ldots = |\lambda_\nu| = 1, |\lambda_{\nu+1}| < 1 \\ 0 & |\lambda_1| < 1 \end{array} \right. \tag{2.2.5}$$

Theorem 2.2.1. *Consider a multi-agent system according to Fig. 2.2.1 consisting of identical agents with discrete-time dynamics $H(z)$ given in (2.2.1) and a strongly connected interaction graph with Laplacian $\mathcal{L}$ fulfilling $\mathcal{L}\mathbf{1} = 0$ and $u_1^{\mathcal{L}T}\mathcal{L} = 0$. The closed loop state transition matrix Φ_{cl} has at least ν eigenvalues $\lambda_i = 1$ with right eigenvectors $v_i = \mathbf{1}_N \otimes v_i^H$ and left eigenvectors $u_i = u_1^{\mathcal{L}} \otimes u_i^H$, if the agent model $H(z)$ has ν integrator poles corresponding to linearly independent eigenvectors v_i^H (and left eigenvectors u_i^H) of the system matrix Φ.*

Proof: The interconnection matrix $\mathcal{L}$ and the local system matrix Φ can be decomposed as

$$\mathcal{L} = \begin{bmatrix} \mathbf{1} & v_2^{\mathcal{L}} & \cdots & v_N^{\mathcal{L}} \end{bmatrix} \begin{bmatrix} 0 & & & \\ & \lambda_2^{\mathcal{L}} & & \\ & & \ddots & \\ & & & \lambda_N^{\mathcal{L}} \end{bmatrix} \begin{bmatrix} u_1^{\mathcal{L}T} \\ u_2^{\mathcal{L}T} \\ \vdots \\ u_N^{\mathcal{L}T} \end{bmatrix} \tag{2.2.6}$$

$$\Phi = \left[\begin{array}{ccc:ccc} v_1^H & \cdots & v_\nu^H & \cdots & v_n^H \end{array}\right] \left[\begin{array}{ccc:ccc} I_{\nu\times\nu} & & & & & \\ & & & & & \\ & & & \lambda_{\nu+1}^H & & \\ \hdashline & & & & \ddots & \\ & & & & & \lambda_n^H \end{array}\right] \begin{bmatrix} u_1^{HT} \\ \vdots \\ \vdots \\ u_n^{HT} \end{bmatrix} \tag{2.2.7}$$

with $\Re(\lambda_i^{\mathcal{L}}) > 0 \; \forall i \in [2, N], \quad |\lambda_i^H| < 1 \; \forall i \in [\nu+1, n]$

Using this together with the mixed product property, the vector $v_i = \mathbf{1}_N \otimes v_i^H,\ 1 \leq i \leq \nu$ can be identified as the right eigenvector of the closed loop system matrix for the eigenvalue 1:

$$\begin{aligned} &\hat{\Gamma}\mathcal{L}_{(m)}\hat{C}v_i = \hat{\Gamma}(\mathcal{L} \otimes I_m)(I_N \otimes C)(\mathbf{1}_N \otimes v_i^H) = \hat{\Gamma}(\mathcal{L}\mathbf{1}_N \otimes Cv_i^H) = 0 \\ &\hat{\Phi}v_i = (I_N \otimes \Phi)(\mathbf{1}_N \otimes v_i^H) = \mathbf{1}_N \otimes \Phi v_i^H = \mathbf{1}_N \otimes v_i^H \\ &\Rightarrow (\hat{\Phi} - \hat{\Gamma}\mathcal{L}_{(m)}\hat{C})v_i = v_i \end{aligned} \tag{2.2.8}$$

In the same manner, the vector $u_i = u_1^{\mathcal{L}} \otimes u_i^H,\ 1 \leq i \leq \nu$ is identified as the corresponding left eigenvector:

$$\begin{aligned} &u_i^T\hat{\Gamma}\mathcal{L}_{(m)}\hat{C} = (u_1^{\mathcal{L}} \otimes u_i^H)^T(I_N \otimes \Gamma)(\mathcal{L} \otimes I_m)\hat{C} = (u_1^{\mathcal{L}^T}\mathcal{L} \otimes u_i^{H^T}\Gamma)\hat{C} = 0 \\ &u_1^T\hat{\Phi} = (u_1^{\mathcal{L}} \otimes u_i^H)^T(I_N \otimes \Phi) = u_1^{\mathcal{L}^T} \otimes u_i^{H^T}\Phi = u_1^{\mathcal{L}^T} \otimes u_i^{H^T} \\ &\Rightarrow u_i^T(\hat{\Phi} - \hat{\Gamma}\mathcal{L}_{(m)}\hat{C}) = u_i^T. \end{aligned} \tag{2.2.9}$$

■

The same result is obtained for a continuous-time model of the process: Let the agents be modeled by the state space model

$$H : \left\{ \begin{array}{ll} \dot{x}_i(t) & = Ax_i(t) + Bu_i(t) \\ y_i(t) & = Cx_i(t) \end{array} \right. \tag{2.2.10}$$

$$u(t) = -\mathcal{L}_{(m)}y(t) = -\mathcal{L}_{(m)}\hat{C}x(t) \tag{2.2.11}$$

with the same signal sizes as (2.2.1). Accordingly, the continuous-time closed loop model is obtained as

$$\dot{x}(t) = (\hat{A} - \hat{B}\mathcal{L}_{(m)}\hat{C})x(t), \tag{2.2.12}$$

which is equivalent to the first order consensus process (2.1.25 with $A_{cl} = \hat{A} - \hat{B}\mathcal{L}_{(m)}\hat{C} = -\mathcal{L}_{cl}$. Presuming an eigenvalue decomposition of A_{cl} according to (2.2.4) with $\Re(\lambda_1) \geq \Re(\lambda_2) \geq \cdots \geq \Re(\lambda_{nN})$, we can find an equivalent to the discrete-time result (2.2.5):

$$\lim_{t\to\infty} x(t) = \lim_{t\to\infty} e^{A_{cl}t}x(0) = \left\{ \begin{array}{cl} \infty & \Re(\lambda_1) > 0 \\ \sum_{i=1}^{\nu} v_i u_i^T x(0) & \Re(\lambda_1) = \ldots = \Re(\lambda_\nu) = 0, \Re(\lambda_{\nu+1}) < 0 \\ 0 & \Re(\lambda_1) < 0 \end{array} \right. \tag{2.2.13}$$

Theorem 2.2.2. *Consider a multi-agent system according to Fig. 2.2.1 consisting of identical agents with continuous-time dynamics $H(s)$ given in (2.2.10) and a fully connected interaction graph with Laplacian $\mathcal{L}$ fulfilling $\mathcal{L}\mathbf{1} = 0$ and $u_1^{\mathcal{L}^T}\mathcal{L} = 0$. The closed loop state transition matrix A_{cl} has at least ν eigenvalues $\lambda_i = 0$ with right eigenvectors $v_i = \mathbf{1}_N \otimes v_i^H$ and left eigenvectors $u_i = u_1^{\mathcal{L}} \otimes u_i^H$, if the agent model $H(s)$ has ν integrator poles corresponding to linear independent eigenvectors v_i^H (and left eigenvectors u_i^H) of the system matrix A.*

Proof: The interconnection matrix $\mathcal{L}$ and the local system matrix A can be decomposed as in (2.2.6) and

$$A = \begin{bmatrix} v_1^H & \cdots & v_\nu^H & \cdots & v_n^H \end{bmatrix} \left[\begin{array}{c|ccc} 0_{\nu\times\nu} & & & \\ \hline & \lambda_{\nu+1}^H & & \\ & & \ddots & \\ & & & \lambda_n^H \end{array}\right] \begin{bmatrix} u_1^{H^T} \\ \vdots \\ \vdots \\ u_n^{H^T} \end{bmatrix} \tag{2.2.14}$$

with $\Re(\lambda_i^H) < 0 \; \forall i \in [\nu+1, n]$. Using this, the vector $v_i = \mathbf{1}_N \otimes v_i^H,\ 1 \leq i \leq \nu$ can be identified as the right eigenvector of the closed loop system matrix for the zero eigenvalue:

$$\begin{aligned} &\hat{B}\mathcal{L}_{(m)}\hat{C}v_i = \hat{B}(\mathcal{L}\otimes I_m)(I_N \otimes C)(\mathbf{1}_N \otimes v_i^H) = \hat{B}(\mathcal{L}\mathbf{1}_N \otimes Cv_i^H) = 0 \\ &\hat{A}v_i = (I_N \otimes A)(\mathbf{1}_N \otimes v_i^H) = \mathbf{1}_N \otimes Av_i^H = 0 \\ &\Rightarrow (\hat{A} - \hat{B}\mathcal{L}_{(m)}\hat{C})v_i = 0 \end{aligned} \tag{2.2.15}$$

In the same manner, the vector $u_i = u_1^{\mathcal{L}} \otimes u_i^H,\ 1 \leq i \leq \nu$ is identified as the corresponding left eigenvector:

$$\begin{aligned} &u_i^T\hat{B}\mathcal{L}_{(m)}\hat{C} = (u_1^{\mathcal{L}} \otimes u_i^H)^T(I_N \otimes B)(\mathcal{L}\otimes I_m)\hat{C} = (u_1^{\mathcal{L}^T}\mathcal{L} \otimes u_i^{H^T}B)\hat{C} = 0 \\ &u_i^T\hat{A} = (u_1^{\mathcal{L}} \otimes u_i^H)^T(I_N \otimes A) = u_1^{\mathcal{L}^T} \otimes u_i^{H^T}A = 0 \\ &\Rightarrow u_i^T(\hat{A} - \hat{B}\mathcal{L}_{(m)}\hat{C}) = 0. \end{aligned} \tag{2.2.16}$$

■

The following theorem concludes the deduction of the rendez-vous result and yields the rendez-vous position, at which the agents meet at $t \to \infty$.

Theorem 2.2.3. *Consider a multi-agent system according to Fig. 2.2.1 consisting of identical agents with local dynamics H given by either (2.2.1) or (2.2.10) and a fully connected interaction graph with Laplacian $\mathcal{L}$ fulfilling $\mathcal{L}\mathbf{1} = 0$ and $u_1^{\mathcal{L}^T}\mathcal{L} = 0$. For $r(t) = 0$ and initial state $x(0) = x_0$ the multi-agent system converges to the rendez-vous states*

$$x(t\to\infty) = \begin{cases} (\mathbf{1}u_1^{\mathcal{L}^T} \otimes \sum_{i=1}^{\nu} v_i^H u_i^{H^T})x_0 & \nu > 0 \\ 0 & \nu = 0 \end{cases} \tag{2.2.17}$$

if the multi-agent system fulfills Assumption 2.2.1 and the local agent model has ν integrator poles with corresponding left and right eigenvectors u_i^H and v_i^H being linear independent for $1 \leq i \leq \nu$.

Proof: As the local agent model is assumed to have ν integrator poles with linear independent left and right eigenvectors u_i^H and v_i^H, Theorem 2.2.1 provides ν linear independent eigenvectors of the discrete-time closed loop system. A corresponding statement for the continuous-time representation is provided by Theorem 2.2.2. This implies that the closed loop system has at least ν integrator eigenvalues. As Assumption 2.2.1 gives

ν as upper bound on the number of closed loop integrator poles, in case this assumption holds it is clear that the closed loop system has exactly ν integrator poles. Using Theorem 2.2.2 together with (2.2.13), or for a discrete-time representation Theorem 2.2.1 and (2.2.5), the steady-state values of the rendez-vous process can be obtained:

$$\lim_{t\to\infty} x(t) = \sum_{i=1}^{\nu} v_i u_i^T x_0 = \sum_{i=1}^{\nu} (\mathbf{1}_N \otimes v_i^H)(u_1^{\mathcal{L}} \otimes u_i^H)^T x_0 = \sum_{i=1}^{\nu} (\mathbf{1}_N u_1^{\mathcal{L}^T} \otimes v_i^H u_i^{H^T}) x_0$$

From the definition of the kronecker produckt it is straight forward to obtain (2.2.17), which concludes the proof. ■

In (2.2.17), the expression left of the Kronecker symbol is a dyadic product with the one-vector $\mathbf{1}$ as left factor, thus it is a matrix of N copies of the same row. This fact yields the result $x_1(t \to \infty) = \ldots = x_N(t \to \infty)$ and corresponds to the agents meeting at a single point. The resulting rendez-vous position is obtained by the output equation of H as

$$y_i(t \to \infty) = C(u_1^{\mathcal{L}^T} \otimes \sum_{i=1}^{\nu} v_i^H u_i^{H^T}) x_0. \tag{2.2.18}$$

Remark 2.2.1. *Assuming that the discrete-time model $H(z)$ in (2.2.1) is obtained by zero-order hold discretization of the continuous-time model $H(s)$ given in (2.2.10) at sampling time T, the state transition matrix of $H(z)$ is obtained as $\Phi = e^{AT}$. From this expression one can infer that with a decomposition $A = X\Lambda X^{-1}$ the discrete-time system matrix can be decomposed accordingly as $\Phi = e^{X\Lambda X^{-1}T} = Xe^{\Lambda T}X^{-1}$ (see Ren and Beard [2008]). In this case the eigenvectors of the state transition matrix corresponding to the integrator eigenvalues are equal for the continuous-time and discrete-time representation. Thus, Theorem 2.2.3 provides the same result for both representations.*

Note that the local system matrix Φ having an eigenvalue at 1 (or A having a zero eigenvalue) is equivalent to the local system $H(z)$ (or $H(s)$, respectively) having integral action. If, instead, all eigenvalues of $H(z)$ have absolute values less than one, the agents will converge to the origin. For the case of the existence of integrator poles in H, the eigenvectors v_i^H and u_i^H, $1 \leq i \leq \nu$ describe the steady state value of the free response of the local system H:

$$u(k) = 0 \Rightarrow x(k+1) = \Phi x(k) \tag{2.2.19}$$

$$\Rightarrow \lim_{k\to\infty} x(k) = \Phi^k x(0) = \sum_{i=1}^{\nu} v_i^H u_i^{H^T} x(0). \tag{2.2.20}$$

Consequently, the steady state response of the global multi-agent system is determined both by the properties of the interconnection matrix $\mathcal{L}$ and the free response of the local subsystems H.

2.3 Formation Establishment

Formation establishment covers the process initiated by the assignment of a timewise constant formation reference vector r at time $t = 0$. At this time instance the system is assumed to have the initial state $x(t = 0) = x_0$. The response of the closed loop system can be expressed as a superposition of the free response to the initial states x_0 and the forced response to the assigned reference. While the free response is equal to the rendezvous process described in the previous section, the forced response can be seen as step response to a reference input signal $r(t) = r\sigma(t)$, where $\sigma(t)$ is the unit step function.

Here the forced response is examined, considering the multi-agent system as shown in Fig. 2.2.1 with the agent model $H(s)$ given in (2.2.10), a non-zero reference r and zero initial states $x(0) = 0$. The input vector is given as

$$u(t) = \mathcal{L}_{(m)}e(t) = \mathcal{L}_{(m)}(r - y). \tag{2.3.1}$$

The closed loop system model is accordingly obtained as

$$T : \begin{cases} \dot{x}(t) = A_{cl}x(t) + B_{cl}r \\ y(t) = \hat{C}x(t) \end{cases} \tag{2.3.2}$$

$$\text{with } A_{cl} = \hat{A} - \hat{B}\mathcal{L}_{(m)}\hat{C} \quad B_{cl} = \hat{B}\mathcal{L}_{(m)}. \tag{2.3.3}$$

Using the decomposition

$$A_{cl} = X\Lambda X^{-1} = \begin{bmatrix} X_1 & X_2 \end{bmatrix} \left[\begin{array}{c|c} 0_{\nu\times\nu} & \\ \hline & \Lambda_2 \end{array}\right] \begin{bmatrix} Y_1^T \\ Y_2^T \end{bmatrix}, \quad Y^T = X^{-1} \tag{2.3.4}$$

with $\Lambda_2 = \text{diag}(\lambda_{\nu+1}, \ldots, \lambda_{nN}) < 0$, for the formation establishment process of this system the following can be stated:

Theorem 2.3.1. *Consider a multi-agent system according to Fig. 2.2.1, fulfilling Assumption 2.2.1 and consisting of identical agents with local dynamics $H(s)$ given by (2.2.10) and a fully connected interaction graph with Laplacian $\mathcal{L}$ fulfilling $\mathcal{L}\mathbf{1} = 0$ and $u_1^{\mathcal{L}^T}\mathcal{L} = 0$. For a constant reference r assigned at time $t = 0$ and zero initial states, the closed loop multi-agent system converges to the finite steady state $x(t \to \infty) = -X_2\Lambda_2^{-1}Y_2^T B_{cl}r$, where X_2 and Y_2 contain the right and left eigenvectors of the global closed loop system corresponding to its non-zero poles. The transient response is given by*

$$x(t) = -X_2\Lambda_2^{-1}Y_2^T B_{cl}r + X_2\Lambda_2^{-1}e^{\Lambda_2 t}Y_2^T B_{cl}r. \tag{2.3.5}$$

Proof: It is well known that the forced response of an LTI system can be obtained as the convolution integral of the impulse response of the system and the assigned input signal. For the formation establishment process this yields

$$x(t) = \int_0^t e^{A_{cl}(t-\tau)}B_{cl}r(\tau)d\tau. \tag{2.3.6}$$

Using (2.3.4), for the step reference function $r(t) = r\sigma(t)$ one obtains

$$x(t) = \begin{bmatrix} X_1 & X_2 \end{bmatrix} \begin{bmatrix} \Xi_1(t) & \\ & \Xi_2(t) \end{bmatrix} \begin{bmatrix} Y_1^T \\ Y_2^T \end{bmatrix} B_{cl} r \tag{2.3.7}$$

with

$$\begin{aligned} \Xi_1(t) &= \int_0^t e^{0(t-\tau)} d\tau = t \cdot I_{\nu \times \nu} \\ \Xi_2(t) &= \int_0^t e^{\Lambda_2(t-\tau)} d\tau \\ &= \operatorname*{diag}_{i=\nu+1}^{nN} \left(\int_0^t e^{\lambda_i(t-\tau)} d\tau \right) = \operatorname*{diag}_{i=\nu+1}^{nN} \left(-\frac{1}{\lambda_i}(1 - e^{\lambda_i t}) \right) = -\Lambda_2^{-1} \left(I - e^{\Lambda_2 t} \right). \end{aligned}$$

Accordingly, $x(t)$ is a superposition of three terms:

$$x(t) = X_1 Y_1^T B_{cl} r \cdot t - X_2 \Lambda_2^{-1} Y_2^T B_{cl} r + X_2 \Lambda_2^{-1} e^{\Lambda_2 t} Y_2^T B_{cl} r \tag{2.3.8}$$

For a non-zero vector $X_1 Y_1^T B_{cl} r$ the response $x(t)$ integrates the input signal and thus grows linearly to infinity for a non-zero constant or step reference input. However, if the requirements of Theorem 2.2.2 or 2.2.1 are fulfilled, the respective term disappears, as a row-wise examination yields

$$\left[Y_1^T B_{cl} \right]_{i:} = u_i^T B_{cl} = (u_1^{\mathcal{L}} \otimes u_i^H)^T (I \otimes B)(\mathcal{L} \otimes I) = u_1^{\mathcal{L}^T} \mathcal{L} \otimes u_i^{H^T} B = 0 \quad \forall i \in [1, \nu].$$

In this case, as the third term in (2.3.8) decays to zero for $t \to \infty$, the global closed loop system converges to the second term in (2.3.8). This concludes the proof. ■

In Theorem 2.3.1 the state response of the formation establishment process is given as a linear function of the absolute reference vector r. In contrast, the formation control problem formulated in Section 2.1.4 limits the control task to the relative displacements instead of the absolute positions. However, one can easily show that under a reasonable condition on the closed loop input matrix B_{cl} the closed loop response only depends on the relative reference vector $\tilde{r}$:

Theorem 2.3.2. *Under the conditions of Theorem 2.3.1, the response of the multi-agent system to the assignment of a constant reference r at time $t = 0$ is given by*

$$x(t) = - X_2 \Lambda_2^{-1} Y_2^T B_{cl} \tilde{r} + X_2 \Lambda_2^{-1} e^{\Lambda_2 t} Y_2^T B_{cl} \tilde{r}, \tag{2.3.9}$$

$$x(t \to \infty) = - X_2 \Lambda_2^{-1} Y_2^T B_{cl} \tilde{r} \tag{2.3.10}$$

with $\tilde{r} = r - r_a$ being the relative part of r according to (2.1.37) if the closed loop input matrix B_{cl} fulfills $\mathscr{A} \subseteq ker(B_{cl})$.

Proof: The proof is easily concluded by expressing r in terms of the agreement and disagreement part:

$$\begin{aligned} & r = \tilde{r} + \mathbf{1} \otimes r_a \\ & \Rightarrow B_{cl} r = B_{cl} \tilde{r} + B_{cl}(\mathbf{1} \otimes r_a) = B_{cl} \tilde{r} \quad \text{as } \mathbf{1} \otimes r_a \in \ker(B_{cl}) \end{aligned}$$

■

In the following, a simple example is given to illustrate the previous results.

Example 2.3.1 (Swarm of SISO Vehicles)**.** Consider a swarm of $N = 5$ vehicles with second order dynamics

$$P : \begin{cases} \dot{x}_i(t) = \begin{bmatrix} 0 & 1 \\ 0 & -0.05 \end{bmatrix} x_i(t) + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u_i(t) \\ y_i(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x_i(t) \end{cases}$$

Here the movement is only examined in one spatial dimension, such that the vehicles can be modeled as Single Input Single Output (SISO) systems. Each vehicle is controlled by a local controller $K(s)$, such that the local agent model is $H(s) = P(s)K(s)$. The controller is chosen as

$$K(s) = 20\frac{s + 0.005}{s + 2.4}.$$

The communication topology is given by the adjacency matrix

$$\mathcal{A}^0 = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \Rightarrow \quad \mathcal{L} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ -\frac{1}{3} & 1 & 0 & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{2} & 0 & 1 & -\frac{1}{2} & 0 \\ 0 & -\frac{1}{3} & -\frac{1}{3} & 1 & -\frac{1}{3} \\ 0 & -\frac{1}{2} & 0 & -\frac{1}{2} & 1 \end{bmatrix} \text{ with } \lambda^{\mathcal{L}} = \begin{bmatrix} 0 \\ 0.667 \\ 1 \\ 1.5 \\ 1.83 \end{bmatrix}$$

The local agent model has poles at $\lambda^H = \begin{bmatrix} 0 & -0.05 & -2.4 \end{bmatrix}^T$, thus $\nu = 1$. For the closed loop system the eigenvalues are computed as

$$\lambda = \begin{bmatrix} 0 & -0.05 & -0.05 & -0.05 & -0.05 & -0.05 & -0.87 & \dots & -2.40 \end{bmatrix}^T$$

which confirms that the closed loop system inherits the integrator pole of the local agent model. Computing the rendez-vous position according to 2.2.17 yields $y_i(t \to \infty) = 10/3 \approx 3.3$ and is marked as black asterisk in Fig. 2.3.1.

Figure 2.3.1 shows the position of the vehicles during the rendez-vous process starting at the positions $y_0 = \begin{bmatrix} 1 & 1 & 3 & 3 & 10 \end{bmatrix}^T$. As clearly visible, all agents converge to the expected rendez-vous position.

The formation establishment process is shown in Fig. 2.3.2 for a reference $r = \begin{bmatrix} 1 & 2 & 3 & 5 & 7 \end{bmatrix}^T$ and zero initial states. From (2.1.37), the agreement part is obtained as $r_a = 3.6$. Consequently, the formation is expected to be established at the positions $\tilde{r}_i = r_i - 3.6$. The black asterisks mark the steady state positions computed according to Theorem 2.3.1. It is clearly visible that the agents correctly establish the desired formation at the expected location.

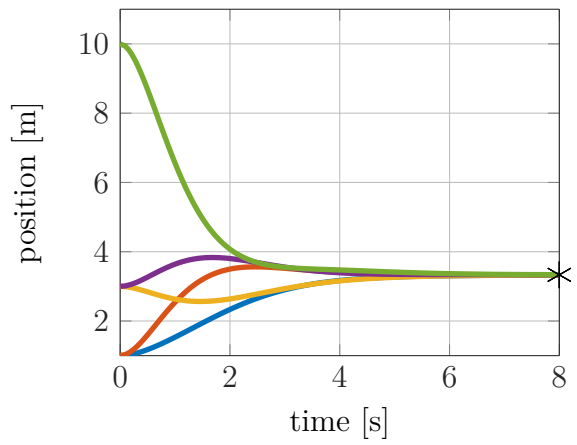


Figure 2.3.1: Agent positions during rendez-vous process

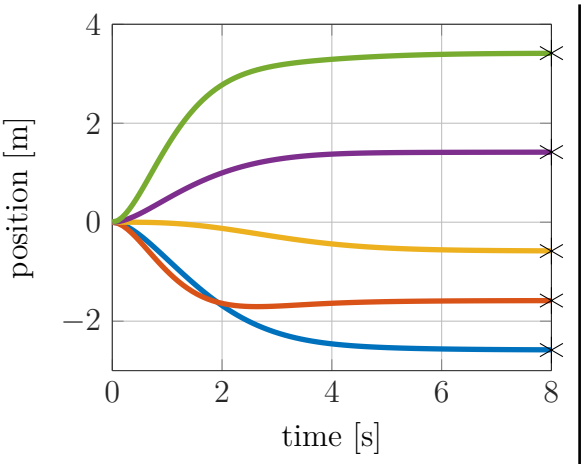


Figure 2.3.2: Agent positions during formation establishment process

2.4 Relationship between Performance and Topology

The previous section already outlined that the closed-loop dynamics of a multi-agent system are not only governed by the local agent dynamics and the distributed controllers, but are as well influenced by the interaction topology. To illustrate this fact and to motivate future research on the topology aspect, in the present section the effect of the interaction topology on the performance of a distributed control scheme in a multi-agent system is examined by means of a numerical example. This comprises both the connectivity of the network as well as different link weighting policies.

2.4.1 Network Properties and Connectivity Measures

Here some terms and definitions are introduced to characterize the interaction topology and the role of particular nodes within the network by means of spectral graph theory. These definitions are adopted from Bertrand and Moonen [2013].

- **Algebraic Connectivity** The term *connectivity* is a property of the graph and describes how densely it is connected. This property can be quantified by means of the second-smallest eigenvalue $\lambda_2^{\mathcal{L}}$ of the unweighted Laplacian $\mathcal{L}$, referred to as *algebraic connectivity* in the following. For connected graphs, $\lambda_2^{\mathcal{L}}$ is strictly positive. Furthermore, it is monotonically increasing with the number of links in the network [Bertrand and Moonen, 2013].

- **Fiedler Vector** The Fiedler vector $\mathbf{f}$ is the eigenvector of $\mathcal{L}$ corresponding to the second-smallest eigenvalue $\lambda_2^{\mathcal{L}}$. Together with $\lambda_2^{\mathcal{L}}$, the Fiedler vector provides important information about the clustering properties of the graph, which is relevant to the properties of distributed algorithms operated over the network [Bertrand and Moonen, 2013].

- **Centrality** The centrality is a property of a node in a network and can be interpreted as the influence or importance of this node in the network. There are multiple ways to quantify this property:
  - **Degree Centrality** The simplest centrality measure is the in-degree d_i^{in}, which is the number of neighbors the node i receives data from. The drawback of this measure is that it only takes into account the local environment of a node rather than the whole network.
  - **Eigenvector Centrality** The importance of a node beyond its local neighborhood can be measured by taking into account not only the number of neighbors, but also their importance, i.e. their centrality. An intuitive definition of the centrality c_i of node i is thus $c_i = 1/\alpha \sum_{k \in N_i} c_k$ (α is a normalization factor). This definition can be rewritten to the eigenvector problem $\alpha c = \Psi c$ with $c = [c_1 \ \cdots \ c_N]^T$ and the adjacency matrix Ψ. It can be shown that for restricting c_i non-negative, a unique solution is given by the principal eigenvector of Ψ corresponding to its largest eigenvalue [Bertrand and Moonen, 2013].
- **Distance** In the context of graph theory, the term *distance* is a property of a node pair and means the number of links traveled along the shortest path between these two nodes.

2.4.2 Influence of Connectivity and Link Weights on Control

In this section a numerical example is presented to study the influence of the interaction topology on the control performance. It considers the performance of a distributed formation control scheme for a swarm of 10 quad-copter agents in cases of different, but time-invariant interaction topologies. In particular, for various instances of the interaction matrix Ψ the global closed loop system model $\hat{T}_\Psi$ was composed according to definition (2.1.9) and the $\mathcal{H}_\infty$ norm of this system was computed numerically. Due to the assumption of a time-invariant topology, the global closed loop system is time-invariant as well and the $\mathcal{H}_\infty$ norm is equal to the induced $\mathcal{L}_2$ gain and is a reasonable performance indicator. With the representation used here, all existing links are assumed ideal.

The agents are modeled by the linearized model from Section B.1, which is the underlying model of the simulations presented in Section 3.3.1. Accordingly, the control scheme was chosen as the cooperative architecture with interconnected controllers as shown in Fig. 3.2.8c. For controller synthesis the interconnection-scheduled approach from Hoffmann and Werner [2017] presented in Chapter 3.1 was used, for which the problem formulation and choice of shaping filters were adopted from the simulation study of Section 3.3.1 (see case (iii) of Table 3.1).

For the present example, 20 leader-follower topologies (one leader, 9 followers) were generated randomly. To sample various different connectivity values, for each topology instance a different link existence probability was taken from the range 0.2 to 0.8. With these topologies, different heuristic weighting schemes were tested. They were chosen with the requirement of being implementable in a distributed manner, such that the distributed

character of the control system including the link weights in preserved. Two different ways of assigning the weights are tested: One method is directly assigning a characteristic value as unscaled weight ψ_{ik}^0. The second method uses a pre-defined increasing sequence s_n known by each agent [Mirali and Werner, 2018]. Locally, the neighbor agents are assigned a priority by a specified criterion (corresponding to the weight assigned for the direct method) and the first d_i^{in} values of s_n are assigned as weights $\psi_{ik}^0 = s_n$ by the order of the neighbors' priority. In both methods the normalized weights ψ_{ik} fulfilling (2.1.17) are obtained as

$$\psi_{ik} = \psi_{ik}^0 / \sum_{j \in N_i} \psi_{ij}^0. \tag{2.4.1}$$

In this work, s_n was chosen as the sequence of integer numbers, i.e. $1, 2, 3, \ldots$. The following weighting schemes were tested, which are mostly inspired by Mirali and Werner [2017]:

1. **Equal weighting** All neighbors are weighted equally, thus $\psi_{ik}^0 = 1 \ \forall k \in N_i$. In this case Ψ equals the adjacency matrix of the communication graph.

2. **Eigenvector centrality, direct** Here the eigenvector centrality is used as defined in Bertrand and Moonen [2013]: $\psi_{ik}^0 = c_k$. Thus, the weight of the data from agent k is the higher the more central node k is in the network. A method to determine the centrality in a distributed manner is given in Bertrand and Moonen [2013].

3. **Eigenvector centrality, sorted sequence** This method uses the eigenvector centrality as well, but assigns the sequence values s_n in the order of the (increasing) centrality.

4. **In-degree, direct** Here each neighbor is weighted according to its in-degree, i.e. $\psi_{ik}^0 = d_i^{\text{in}}$. The idea behind is that a neighbor is assumed more important if it is influenced by more other neighbors. If a neighbor is a leader (who has in-degree zero), its weight is set to $\psi_{ik_L} = \max_{k \in N_i} \psi_{ik}^0 + 1$ in order to assign a high weight to the leader. Without this modification a leader would be disconnected.

5. **In-degree, sorted sequence** Same criterion as (4), but assigning the sequence values as weights in increasing order of the in-degrees.

6. **Leader distance, direct** This is a specific weighting scheme for leader-follower-systems with emphasis on leader tracking. Therefore, the weight of each neighbor is chosen as the shortest path length in the communication graph from the leader to this neighbor. This quantity can be easily obtained using a distributed algorithm.

Figure 2.4.1 shows the resulting values of $\|\hat{T}_\Psi\|_\infty$ for the tested topologies, plotted over the algebraic connectivity $\lambda_2^{\mathcal{L}}$ of the corresponding topology Ψ. As a first observation, a general tendency of a performance increase for increasing connectivity is clearly visible for all weighting schemes. This is the expected result, as a higher connectivity means a faster

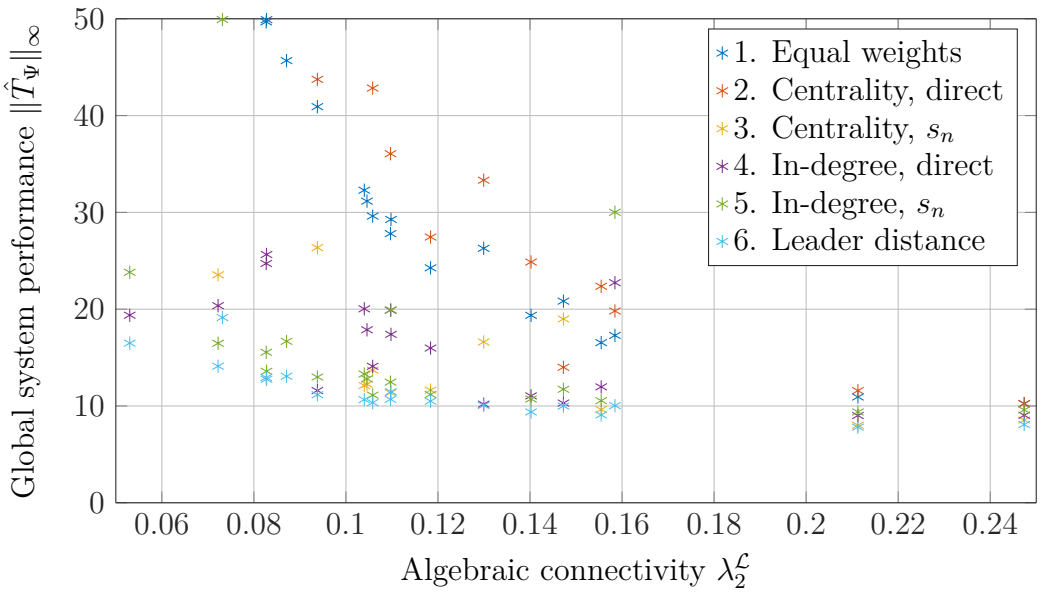


Figure 2.4.1: $\mathcal{H}_\infty$ performance indicator plotted over the algebraic connectivity of the interaction topology

propagation of information through the network. As second observation, especially for sparsely connected networks the performance highly depends on the weighting scheme. In contrast, for rather high connectivities the performance difference between the weighting schemes is marginal. Among the tested weighting schemes, equal weighting achieves the second-lowest performance, which indicates a large benefit of choosing unequal weights. In most cases the best performance is achieved for weighting according to the leader distance, followed by a sorted sequence weighting with respect to the in-degree. For both centrality and in-degree used as weighting strategy, with most topologies the sorted sequence leads to a better performance compared to the direct weighting.

This example emphasizes the high influence of the topology and the choice of link weights on the global system performance. In doing so, it gives a strong motivation to further research on this aspect, which is beyond the scope of this thesis. In particular, the given example does not consider delays or limitations of the channel capacity, which are expected to be relevant in reality under some conditions. While an increase of the connectivity was, as expected, found to be generally beneficial in the present example, this result might be different if non-ideal channels are modeled. In reality, a higher connectivity might come at the price of channel congestions or collisions, which lead to delays or packet dropouts and will most likely decrease the performance. Such aspects are explicitly addressed in recent literature. Among others, a stabilizing controller design approach for the case of packet dropouts is proposed in Pan et al. [2017] and a memory weighted protocol for consensus under packet dropouts is addressed in Datar et al. [2018].

Chapter 3

Distributed Control of Multi-Agent Systems

As one of the contributions of this thesis, the present chapter examines different existing control architectures and controller synthesis approaches, establishes a common framework encompassing these approaches and provides a study comparing the achievable performance.

The basis is laid by outlining two existing controller synthesis approaches to be further examined. Different methods of formulating the addressed control problem in consideration of the control objectives are examined and corresponding control architectures are described and related to the common control framework. Particular interest is given to the consensus-based formation control approach, for which an extended version is introduced. Furthermore, a methodology used to compare the achievable performance for different controller synthesis approaches is presented and results of a corresponding comparative simulation study are provided.

3.1 Existing Controller Synthesis Approaches

3.1.1 Robust Control Using the Small Gain Theorem

This approach is proposed in Pilz and Werner [2012a] and is based on the stability criterion introduced in Popov and Werner [2012]. It is used in connection with the interconnected plant architecture. Accordingly, a direct data exchange between the controllers is not considered. The controller guarantees robust stability both for arbitrary and changing communication topologies as well as for communication delays [Pilz and Werner, 2012a].

The key idea of this approach is to exploit knowledge about the adjacency matrix in order to formulate a stability condition in terms of a single subsystem. The row-scaled adjacency matrix $\mathcal{A}$ from (2.1.12b) features an ℓ_1 norm independent of the topology, which permits a controller to be robust against topology changes. As described in

Popov and Werner [2012], from the construction of $\mathcal{A}$ it can be inferred

$$\|\mathcal{A}\|_1 = \sup_{y \neq 0} \frac{\|\mathcal{A}y(t)\|_\infty}{\|y(t)\|_\infty} = 1 \tag{3.1.1}$$

for arbitrarily switching topologies and arbitrary delays. Using $\mathcal{A}_{(m)} = \mathcal{A} \otimes I_m$ for the Multi-Agent System (MAS) shown in Fig. 2.1.2 and applying the small gain theorem w.r.t. the ℓ_1 norm to this loop, we can derive the sufficient stability condition

$$\|\hat{D}\hat{T}(s)\hat{D}^{-1}\|_1 = \|I_N \otimes (DT(s)D^{-1})\|_1 < 1, \tag{3.1.2}$$

with an invertible scaling matrix $D \in \mathbb{R}^{m \times m}$ [Popov and Werner, 2012]. Using the property $\|\hat{T}(s)\|_1 = \|T(s)\|_1$ of the ℓ_1 norm, we can reduce this condition to a single subsystem, which leads to the following statement:

Theorem 3.1.1. *Popov and Werner [2012] A multi-agent system as shown in Fig. 2.1.2 is stable for any number of agents N and arbitrary and switching communication topologies with any time-varying communication delays, if there exists an invertible matrix $D \in \mathbb{R}^{m \times m}$ satisfying $\|DT_{pq}^{cl}(s)D^{-1}\|_1 < 1$, for T_{pq} denoting the transfer function from interconnection input q_i to interconnection output p_i according to Fig. 2.1.2.*

From Theorem 3.1.1 a stabilizing controller synthesis problem can be formulated as finding a controller $K(s)$ for which the closed loop system fulfills the ℓ_1 condition of Theorem 3.1.1. In Pilz and Werner [2012a] this problem is extended by an $\mathcal{H}_\infty$ criterion on the local performance, which is defined as the relation between exogenous input w_i and performance output z_i of the system model (2.1.2). This extension leads to the following mixed $\mathcal{H}_\infty/\ell_1$ synthesis problem:

Problem 3.1.1. *Pilz and Werner [2012a] Find a controller $K(s)$ that satisfies*

$$min \ \|T_{zw}^{cl}\|_\infty \tag{3.1.3}$$

$$subject \ to \ \|DT_{pq}^{cl}D^{-1}\|_1 < 1, \tag{3.1.4}$$

with a suitable scaling matrix D and T_{zw} and T_{pq} denoting the local closed-loop transfer functions from reference/disturbance input w_i to performance output z_i and from interconnection input q_i to interconnection output p_i according to Fig. 2.1.2.

The combined synthesis problem can be solved using a Youla parameter-based approach proposed in Pilz and Werner [2012a]. A less computationally expensive alternative is to instead solve the following Problem 3.1.2, which is the $\mathcal{H}_\infty/\mathcal{H}_\infty$ companion of Problem 3.1.1, and to check the ℓ_1 condition *a posteriori*.

Problem 3.1.2. *Pilz and Werner [2012a] Find a controller $K(s)$ that satisfies*

$$min \ \|T_{zw}^{cl}\|_\infty \tag{3.1.5}$$

$$subject \ to \ \|DT_{pq}^{cl}D^{-1}\|_\infty < 1, \tag{3.1.6}$$

with a suitable scaling matrix D and T_{zw} and T_{pq} as in Problem 3.1.1.

For the special case of a symmetric interaction matrix, i.e. $\mathcal{A} = \mathcal{A}^T$, Popov and Werner [2012] show that solving Problem 3.1.2 is sufficient to guarantee stability even in case of time-varying topology and delays.

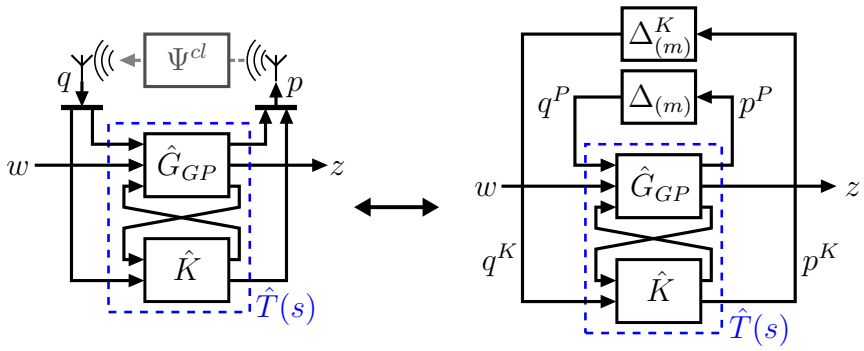


Figure 3.1.1: Analogy of interconnection-scheduled MAS control (left) and gain-scheduled LPV control (right)

3.1.2 Interconnection-Scheduled Approach

The basic idea of this approach is presented in Hoffmann et al. [2014, 2015] and Hoffmann and Werner [2017] and is sketched in Fig. 3.1.1: The interconnected system introduced in Section 2.1.1 is interpreted as LPV system in LFT representation in the sense of Scherer [2001] with the interconnection matrix Ψ handled as parameter block Δ. For this class of systems, Scherer [2001] provides a technique to synthesize a controller which is scheduled by means of an LFT loop connection with a parameter block Δ^K. As here Δ describes the interconnection of the subsystems, the LFT gain scheduling can be realized by connecting the controllers over a network with the corresponding topology Ψ^K. If $\Psi = \Psi^K$ holds, the controllers inherit the topology considered to connect the plants, which equates the framework from Section 2.1.1 with the controller interaction in place. In this interpretation the controller is *scheduled by interconnection* and in this work we refer to this approach as *Interconnection-Scheduled Control (ISC)*.

In this context, the definition of the upper LFT (Def. A.1.2) can be applied to formulate the multi-agent system as parameter-dependent system (assuming $\Psi = \Psi^K$):

$$\mathscr{T}(\Psi): \quad \begin{pmatrix} \dot{x} \\ \dot{x}^K \\ \hline z \end{pmatrix} = \left[\begin{array}{c|c} \mathscr{A}(\Psi) & \mathscr{B}(\Psi) \\ \hline \mathscr{C}(\Psi) & \mathscr{D}(\Psi) \end{array}\right] \begin{pmatrix} x \\ x^K \\ \hline w \end{pmatrix}, \tag{3.1.7}$$

$$\begin{bmatrix} \mathscr{A}(\Psi) & \mathscr{B}(\Psi) \\ \mathscr{C}(\Psi) & \mathscr{D}(\Psi) \end{bmatrix} = \begin{bmatrix} \hat{A}^{cl} & \hat{B}^{cl}_{w} \\ \hat{C}^{cl}_{z} & \hat{D}^{cl}_{zw} \end{bmatrix} + \begin{bmatrix} \hat{B}^{cl}_{q} \\ \hat{D}^{cl}_{zq} \end{bmatrix} \Psi^{cl}(I - \hat{D}^{cl}_{pq}\Psi^{cl})^{-1} \begin{bmatrix} \hat{C}^{cl}_{p} & \hat{D}^{cl}_{pw} \end{bmatrix} \tag{3.1.8}$$

with $\Psi^{cl} = \mathrm{diag}(\Psi_{(m)}, \Psi^K_{(m)})$. Following the lines of Scherer [2000, 2001], for this system a criterion for stability and performance can be formulated using the *Bounded Real Lemma*:

Theorem 3.1.2. *The system $\mathscr{T}(\Psi)$ given in (3.1.7) is asymptotically stable for all interaction matrices Ψ and has a bounded induced $\mathcal{L}_2$ gain $\|\mathscr{T}(\Psi)\|_{\mathcal{L}_2} < \gamma$, if there exists a symmetric Lyapunov matrix $\mathscr{X} > 0$ fulfilling*

$$\left[\begin{array}{c} * \\ \hline * \end{array}\right]^T \left[\begin{array}{cc|cc} 0 & \mathscr{X} & 0 & 0 \\ \mathscr{X} & 0 & 0 & 0 \\ \hline 0 & 0 & \frac{1}{\gamma}I & 0 \\ 0 & 0 & 0 & -\gamma I \end{array}\right] \left[\begin{array}{cc} \mathscr{A}(\Psi) & \mathscr{B}(\Psi) \\ I & 0 \\ \hline \mathscr{C}(\Psi) & \mathscr{D}(\Psi) \\ 0 & I \end{array}\right] < 0 \quad \forall \Psi \tag{3.1.9}$$

A core feature of the method from Scherer [2001] is the *Full-Block $\mathcal{S}$ Procedure*: For a condition (3.1.9) expressing robust quadratic performance of an LPV system with rational parameter dependence, an equivalent condition is established in terms of the LFT representation, at which the nominal system model and the scheduling parameter are separated into different Linear Matrix Inequalities (LMIs). To connect these two LMIs, a new variable $\mathscr{M} \in \mathbb{R}^{2Nm \times 2Nm}$ called *multiplier* is introduced. Applying this method to the aforementioned LPV representation of the multi-agent system leads to an LMI problem with a complexity corresponding to the whole interconnected system. In Hoffmann et al. [2013] the idea is proposed to restrict the multiplier as

$$\mathscr{M} = \begin{bmatrix} I_N \otimes \mathcal{M}_{11} & I_N \otimes \mathcal{M}_{12} \\ I_N \otimes \mathcal{M}_{21} & I_N \otimes \mathcal{M}_{22} \end{bmatrix}. \tag{3.1.10}$$

With this restriction the interconnection matrix is the only non-block-diagonal element in the LMI problem. As it is only contained in one of the LMIs, the second one can already be decomposed into N instances of the same LMI of single subsystem size, which only needs to be considered once. With this method, the following statement is formulated in terms of the closed loop model 2.1.4:

Theorem 3.1.3. *[Hoffmann et al., 2015] The multi-agent system consisting of $\hat{T}$ and $\Psi_{(m)} = \Psi_{(m)}^K$ is stable with induced $\mathcal{L}_2$ gain γ from w to z for all interconnection topologies Ψ, if there exist an $\mathcal{X} \in \mathbb{R}^{2n \times 2n}$ with $\mathcal{X} = \mathcal{X}^T > 0$ and a multiplier $\mathcal{M} \in \mathbb{R}^{4m \times 4m}$ with $\mathcal{M} = \mathcal{M}^T$ fulfilling*

$$\begin{bmatrix} * \\ * \\ * \end{bmatrix}^T \left[\begin{array}{c|c|c} \mathcal{M}_{\mathcal{X}} & & \\ \hline & \mathcal{M} & \\ \hline & & \mathcal{M}_p \end{array}\right] \left[\begin{array}{ccc} A^{cl} & B_q^{cl} & B_w^{cl} \\ \hline I & 0 & 0 \\ \hline C_p^{cl} & D_{pq}^{cl} & D_{pw}^{cl} \\ \hline 0 & I & 0 \\ \hline C_z^{cl} & D_{zq}^{cl} & D_{zw}^{cl} \\ \hline 0 & 0 & I \end{array}\right] < 0, \tag{3.1.11}$$

$$\begin{bmatrix} * \\ * \end{bmatrix}^T \begin{bmatrix} I_N \otimes \mathcal{M}_{11} & I_N \otimes \mathcal{M}_{12} \\ I_N \otimes \mathcal{M}_{21} & I_N \otimes \mathcal{M}_{22} \end{bmatrix} \left[\begin{array}{cc} I_{Nm} & \\ \hline & I_{Nm} \\ \hline \Psi_{(m)} & \\ & \Psi_{(m)} \end{array}\right] > 0 \quad \forall \Psi \tag{3.1.12}$$

with

$$\mathcal{M}_{\mathcal{X}} = \begin{bmatrix} 0 & \mathcal{X} \\ \mathcal{X} & 0 \end{bmatrix}, \quad \mathcal{M}_p = \begin{bmatrix} \frac{1}{\gamma} I & 0 \\ 0 & -\gamma I \end{bmatrix} \tag{3.1.13}$$

If Ψ is normal, a unitary diagonalizing transformation exists [Hoffmann and Werner, 2017], which transforms the multiplier condition (3.1.12) into a family of conditions

$$\begin{bmatrix} * \\ * \end{bmatrix}^T \begin{bmatrix} \mathcal{M}_{11} & \mathcal{M}_{12} \\ \mathcal{M}_{21} & \mathcal{M}_{22} \end{bmatrix} \left[\begin{array}{cc} I_m & \\ \hline & I_m \\ \hline \lambda I_m & \\ & \lambda I_m \end{array}\right] > 0 \quad \forall \lambda \in \mathrm{eig}(\Psi). \tag{3.1.14}$$

With this step, a decomposition of the global synthesis problem into a family of problems on the level of a single subsystem is achieved. By means of the projection lemma and the dualization lemma [Scherer, 2001], a synthesis condition can be derived in terms of the (generalized) plant matrices:

Theorem 3.1.4. *[Hoffmann et al., 2014] There exists a distributed controller $\hat{K}$ such that the multi-agent system consisting of $\hat{T}$ and Ψ is stable with induced $\mathcal{L}_2$ gain γ from w to z for all interconnection topologies Ψ, if there exist $X = X^T > 0$, $Y = Y^T > 0$ and symmetric multipliers $M \in \mathbb{R}^{2m\times 2m}$ and $N \in \mathbb{R}^{2m\times 2m}$ fulfilling*

$$\mathcal{N}_X{}^T \begin{bmatrix} * \\ * \\ * \\ * \end{bmatrix}^T \left[\begin{array}{cc:c:c} 0 & X & & \\ X & 0 & & \\ \hdashline & & M & \\ \hdashline & & & M_p \end{array}\right] \left[\begin{array}{ccc} A & B_q & B_w \\ I & 0 & 0 \\ \hdashline C_p & D_{pq} & D_{pw} \\ 0 & I & 0 \\ \hdashline C_z & D_{zq} & D_{zw} \\ 0 & 0 & I \end{array}\right] \mathcal{N}_X < 0, \tag{3.1.15}$$

$$\mathcal{N}_Y{}^T \begin{bmatrix} * \\ * \\ * \\ * \end{bmatrix}^T \left[\begin{array}{cc:c:c} 0 & Y & & \\ Y & 0 & & \\ \hdashline & & N & \\ \hdashline & & & N_p \end{array}\right] \left[\begin{array}{ccc} -A^T & -C_p{}^T & -C_z{}^T \\ I & 0 & 0 \\ \hdashline -B_q{}^T & -D_{pq}{}^T & -D_{zq}{}^T \\ 0 & I & 0 \\ \hdashline -B_w{}^T & -D_{pw}{}^T & -D_{zw}{}^T \\ 0 & 0 & I \end{array}\right] \mathcal{N}_Y > 0, \tag{3.1.16}$$

$$\begin{bmatrix} X & I \\ I & Y \end{bmatrix} > 0, \tag{3.1.17}$$

$$\begin{bmatrix} * \\ * \end{bmatrix}^T M \begin{bmatrix} I \\ \lambda I \end{bmatrix} > 0, \quad \begin{bmatrix} * \\ * \end{bmatrix}^T N \begin{bmatrix} -\lambda I \\ I \end{bmatrix} < 0 \quad \forall \lambda \in eig(\Psi) \tag{3.1.18}$$

with

$$M = \begin{bmatrix} R & S^T \\ S & Q \end{bmatrix}, \quad N = \begin{bmatrix} \tilde{R} & \tilde{S}^T \\ \tilde{S} & \tilde{Q} \end{bmatrix}, \tag{3.1.19a}$$

$$M_p = \begin{bmatrix} \frac{1}{\gamma} I & 0 \\ 0 & -\gamma I \end{bmatrix}, \quad N_p = \begin{bmatrix} \frac{1}{\gamma} I & 0 \\ 0 & -\gamma I \end{bmatrix}^{-1}, \tag{3.1.19b}$$

$$\mathcal{N}_X = ker\left(\begin{bmatrix} C_v & D_{vq} & D_{vw} \end{bmatrix}\right), \tag{3.1.19c}$$

$$\mathcal{N}_Y = ker\left(\begin{bmatrix} B_u{}^T & D_{pu}{}^T & D_{zu}{}^T \end{bmatrix}\right). \tag{3.1.19d}$$

As this synthesis problem is convex, it can be solved by checking the multiplier condition (3.1.18) only at the boundaries of the interval of the expected values of λ.

For controller reconstruction, the general method provided in Scherer [2000] is not applicable here, as it considers constructing a more general controller scheduling function $\Delta^K(\Delta)$ depending on the multiplier results, usually not fulfilling $\Delta^K = \Delta$. In the corresponding interpretation as interconnected system, this would mean to construct a controller topology $\Psi^K \neq \Psi$, i.e. to consider the controller interaction topology as design parameter. This fundamentally contradicts the objective of a control scheme which provides

robustness with respect to an interaction topology given by physical circumstances. For this reason, an alternative method is required to ensure $\Psi^K = \Psi$. This is provided by the *D/G-scalings* approach proposed in Dettori and Scherer [2001], which restricts the multipliers to

$$M = \begin{bmatrix} R & S^T \\ S & -R \end{bmatrix}, \quad N = \begin{bmatrix} \tilde{R} & \tilde{S}^T \\ \tilde{S} & -\tilde{R} \end{bmatrix} \tag{3.1.20}$$

$$\text{with } R > 0, \quad \tilde{R} < 0, \quad S + S^T = 0, \quad \tilde{S} + \tilde{S}^T = 0. \tag{3.1.21}$$

A beneficial property of this restriction is that it makes the multiplier condition (3.1.18) trivially fulfilled for all interaction matrices with a spectrum inside the closed unit disk. In this case the controller K can be computed by solving the LMI problem (3.1.15) to (3.1.17), reconstructing the closed-loop multipliers [Dettori and Scherer, 2001] and solving for the controller variables. Note that for a robust control problem without providing controller scheduling, like in case of an application to the interconnected plant architecture, this synthesis problem loses convexity and the previously mentioned LMI technique will no longer work. Instead, other solving methods like μ-synthesis would have to be used.

Weighted Links

In the examples of Hoffmann et al. [2014, 2015]; Hoffmann and Werner [2017], Ψ was chosen as scaled adjacency matrix (thus using unweighted links only) and its property of a spectrum within the closed unit disk was used to limit the range of λ. It should be noted that this approach can easily be extended to weighted links: Choosing Ψ according to (2.1.16) and (2.1.17), no matter whether the weights are equal, we obtain the same bound on λ:

Lemma 3.1.1. *Every matrix Ψ chosen according to (2.1.16) and (2.1.17) fulfills*

$$|\lambda| \leq 1 \quad \forall \lambda \in eig(\Psi). \tag{3.1.22}$$

Proof: The proof is easily done by using Gershgorin's circle theorem: As Ψ adopts the sparsity pattern of the adjacency matrix, all diagonal elements are zero. Thus, all Gershgorin circles are centered at the origin. As all ψ_{ik} are chosen positive, from the stochasticity assumption (2.1.17) follows that the radia of the circles, which are the sum of the absolute values of the off-diagonal elements, are one. This limits λ to the closed unit disk. ■

Extension to Non-Symmetric Interaction

In the previously described derivation of a synthesis procedure on subsystem level, the step from the global condition in Theorem 3.1.3 to the decomposed multiplier condition (3.1.14) plays a crucial role. However, this step relies on Ψ being unitarily diagonalizable, which normally is not fulfilled in case of directed topologies such as leader-follower systems or non-symmetric weighting. To allow such a decomposition for arbitrary systems

even in case of non-diagonalizable interaction matrices, in Hoffmann and Werner [2017] methods are proposed using a reformulated problem. The key idea is to express Ψ in terms of a unitarily diagonalizable augmented interaction matrix $\tilde{\Psi}$ as $\Psi = V\tilde{\Psi}W$ using augmentation matrices V and W fulfilling $VW = aI$ (a is some real scalar). With this expression, the original closed loop system model $\hat{T}_\Psi = \Psi * \hat{T}$ from (2.1.8) can be written as

$$\hat{T}_\Psi = \tilde{\Psi} * \left(\begin{bmatrix} W_{(m)} \\ I \end{bmatrix}^T T \begin{bmatrix} V_{(m)} \\ I \end{bmatrix} \right) \tag{3.1.23}$$

by moving the augmentation matrices into the nominal system model. Choosing the augmentation matrices V and W to consist of diagonal blocks of size N, i. e.

$$V = v \otimes I_N, \quad W = w \otimes I_N \quad \text{with } v \in \mathbb{R}^{1\times\beta}, \quad w \in \mathbb{R}^{\beta\times 1} \tag{3.1.24}$$

preserves the decomposability of the nominal system. Thus, the synthesis problem formulated in terms of $\tilde{\Psi}$ can be decomposed into a subsystem-level problem according to Theorem 3.1.4 in terms of the augmented subsystem

$$\tilde{T} = \begin{bmatrix} w_{(m)} \\ I \end{bmatrix}^T T \begin{bmatrix} v_{(m)} \\ I \end{bmatrix} = \left[\begin{array}{c|cc} A^{cl} & B_q^{cl} v_{(m)} & B_w^{cl} \\ \hline w_{(m)} C_p^{cl} & w_{(m)} D_{pq}^{cl} v_{(m)} & w_{(m)} D_{pw}^{cl} \\ C_z^{cl} & D_{zq}^{cl} v_{(m)} & D_{zw}^{cl} \end{array}\right]. \tag{3.1.25}$$

There are multiple possibilities of choosing the augmentation, which result in different performances of the resulting controller. The most promising one found in Hoffmann and Werner [2017] (there titled "Proposition 1") is

$$V = \begin{bmatrix} I_N & 0 \end{bmatrix}, \; W = \begin{bmatrix} I_N \\ 0 \end{bmatrix}, \; \tilde{\Psi} = \begin{bmatrix} \Psi & -\Psi^T \\ \Psi^T & \Psi \end{bmatrix}. \tag{3.1.26}$$

This augmented interaction matrix $\tilde{\Psi}$ is normal, since $\tilde{\Psi}\tilde{\Psi}^T = \tilde{\Psi}^T\tilde{\Psi}$ holds.

Meanwhile, the *symmetrization* method previously proposed in Hoffmann et al. [2014, 2015] has two important drawbacks: Firstly, it imposes constraints on the communication topology. Secondly, it does not allow the use of a row-wise scaled interconnection matrix like (2.1.12b), but requires a both-sided scaling. This has two drawbacks: Firstly, the out-degrees of the neighbors need to be known, which requires additional effort and potentially higher communication load, secondly the resulting weighting of the neighbors' data distorts the consensus.

Remark 3.1.1. *The choices of V and W in Hoffmann and Werner [2017] have the drawback that the augmented interaction matrix $\tilde{\Psi}$ is not symmetric and thus their eigenvalues are complex. In that case, formulating a meaningful problem requires a modification of the LMI condition resulting in a 2-dimensional parameter space to be checked. This can be avoided by choosing the sub-blocks of Proposition 2 [Hoffmann and Werner, 2017] with*

$$V = \begin{bmatrix} I_N & 0 \end{bmatrix}, \quad \tilde{\Psi} = \begin{bmatrix} \Psi_R & \Psi_S \\ \Psi_S^T & \Psi_* \end{bmatrix}, \quad W = \begin{bmatrix} I_N \\ I_N \end{bmatrix}, \tag{3.1.27}$$

$$\Psi_R = 0.5(\Psi+\Psi^T) = \Psi_R^T, \quad \Psi_S = 0.5(\Psi-\Psi^T) = -\Psi_S^T$$

with an arbitrary symmetric real Ψ_**. The advantage of this proposition (3.1.27) is that* $\tilde{\Psi}$ *is symmetric real, such that its eigenvalues* $\tilde{\lambda}$ *are real as well and the LMI condition (3.1.18) only has to be fulfilled on a real interval.*

3.2 Problem Formulation and Control Architectures

In this section, first the performance objectives are stated, which are pursued in distributed controller design for multi-agent systems. They are formulated as requirements on sensitivity functions in terms of the generic multi-agent system model used in Chapter 2.2. Based on this, generalized plants of the form (2.1.2) are given for different control architectures used in the existing approaches reviewed in this thesis.

3.2.1 Control Objectives

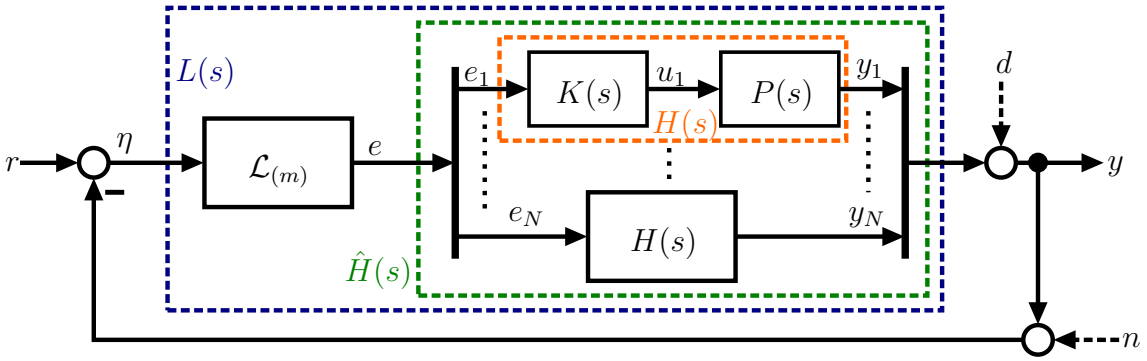


Figure 3.2.1: Global Closed Loop

A general representation of the system under consideration is shown in Fig. 3.2.1, where $H(s)$ is the model of a controlled agent. For an agent system $H(s)$ consisting of a controller and the local agent dynamics, $K(s)$ denotes the transfer function from e_i to the control input u_i assigned internally to control the local dynamics. The reference vector r defines the desired formation, further external inputs are the output disturbance d and the measurement noise n. Corresponding to (2.1.31), the output and error signals are

$$\eta = r - y - n, \tag{3.2.1a}$$
$$e = \mathcal{L}_{(m)}\eta = \mathcal{L}_{(m)}(r - y - n), \tag{3.2.1b}$$
$$y = \hat{H}(s)\mathcal{L}_{(m)}(r - y) + d. \tag{3.2.1c}$$

Regarding the goal of achieving and maintaining the formation defined by r according to Definition 2.1.3 at reasonable control effort, the following objectives can be formulated:

- $\lim_{t\to\infty}(r - y(t)) = \mathbf{1} \otimes \eta_a$

$\Rightarrow$ Tf($r \to e$) small for formation reference tracking

$\Rightarrow$ Tf($d \to e$) small for disturbance rejection

- Tf($n \to y$) small for noise attenuation
- Tf($r \to u$) small for reduction of control effort

Solving (3.2.1) for η and y yields

$$\eta = (I + \hat{H}(s)\mathcal{L}_{(m)})^{-1}(r - d - n), \tag{3.2.2}$$
$$e = \mathcal{L}_{(m)}(I + \hat{H}(s)\mathcal{L}_{(m)})^{-1}(r - d - n), \tag{3.2.3}$$
$$y = \hat{H}(s)\mathcal{L}_{(m)}(I + \hat{H}(s)\mathcal{L}_{(m)})^{-1}(r - d - n). \tag{3.2.4}$$

These expressions contain important transfer functions describing the relation of the external inputs and the outputs of the closed loop system:

$S(s) = (I + \hat{H}(s)\mathcal{L}_{(m)})^{-1}$	(absolute) Sensitivity Function	(3.2.5a)
$S_F(s) = \mathcal{L}_{(m)}(I + \hat{H}(s)\mathcal{L}_{(m)})^{-1}$	Formation Sensitivity Function	(3.2.5b)
$T(s) = \hat{H}(s)\mathcal{L}_{(m)}(I + \hat{H}(s)\mathcal{L}_{(m)})^{-1}$	Complementary Sensitivity Function	(3.2.5c)
$KS_F(s) = \hat{K}(s)S_F(s)$	Control Sensitivity Function	(3.2.5d)

The core idea of $\mathcal{H}_\infty$-optimal mixed-sensitivity controller design, as described e.g. in Skogestad and Postlethwaite [2005], is based on minimizing the $\mathcal{H}_\infty$ norm of shaped versions of $S(s)$ and $KS(s)$. This coincides with the usual design objective of reducing the absolute control error $r - y$. However, in case of the formation control problem we are not interested in reducing η to zero. Instead, the goal is to reach a common (but possibly non-zero) error for all agents. Thus, shaping $S(s)$ is not meaningful in a formation control problem. A reasonable alternative is provided by shaping the *formation sensitivity function* $S_F(s) = \mathcal{L}_{(m)}S(s)$, which yields zero whenever S yields a common error $\eta = \mathbf{1} \otimes \eta_a$.

In terms of S_F the formation control objectives can be formulated as follows:

1. $\bar{\sigma}(S_F) \approx 0$ at $\omega \ll \omega_b$ for reference tracking
2. $\bar{\sigma}(S_F) \approx 0$ at $\omega \ll \omega_b$ for disturbance rejection
3. $\bar{\sigma}(T) \approx 0$ at $\omega \gg \omega_b$ for noise attenuation
4. $\bar{\sigma}(KS_F) \approx 0$ at $\omega \gg \omega_b$ for control effort reduction

A reasonable border between the low frequency range of objectives 1 and 2 and the high frequency range of objectives 3, 4 is the sensitivity bandwidth according to Skogestad and Postlethwaite [2005]. With respect to S_F, it is defined as the frequency ω_b at which $\bar{\sigma}(S_F(j\omega_b)) = 1/\sqrt{2}$ holds.

Although the $\mathcal{H}_\infty$ optimization technique referred to here in context with a frequency domain representation of the sensitivities only exist for time-invariant systems, it is common in LPV control theory to extend this technique to the induced $\mathcal{L}_2$ gain of time-varying sensitivity operators. For this reason it is meaningful to use the $\mathcal{H}_\infty$ argumentation to formulate the synthesis problem even for time-varying systems.

3.2.2 Single-Loop Architectures

In the previous section the control objectives have been introduced for a general control scheme (see Fig. 3.2.1) consisting of the generalized agent model $H(s)$ and the Laplacian $\mathcal{L}$ of the interaction graph. In general, the agent model $H(s)$ contains both a model $P(s)$ describing the local agent dynamics (here also referred to as *plant model*) and a controller $K(s)$. The task of this controller is twofold in a single control loop: On the one hand it has to stabilize and control the local dynamics, on the other hand it has to control the interaction, i.e. to influence the dynamics of an individual agent in a way that the whole team of interacting agents achieves the desired properties. For the particular control architecture, meaning the internal setup of $H(s)$ and the composition of the data being transmitted, different possibilities have been proposed in literature. In the following, some of them are reviewed.

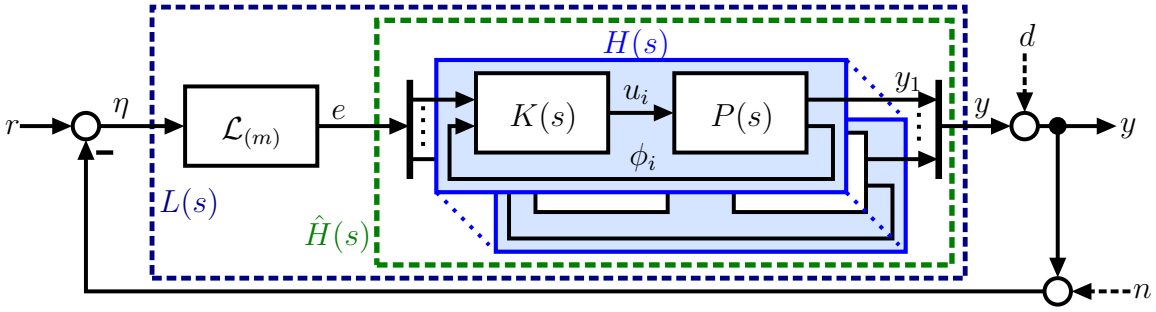


Figure 3.2.2: Global system with control architecture from Pilz and Werner [2012a]

The simplest and most straight forward architecture consists of a series connection of controller $K(s)$ and agent model $P(s)$, resulting in $H(s) = P(s)K(s)$. In this scheme, which is considered in Pilz et al. [2009], the controller inputs are limited to relative data with respect to the neighbor agents and do not provide any information about the absolute state of the agent. For this reason it is only useful if the local agent dynamics P are stable and easy to control or, as in Pilz et al. [2009], if P already contains an existing local controller stabilizing the agent dynamics. If, instead, a controller is to be designed to fulfill both tasks, it is reasonable to feed local data to the controller as an additional input ϕ_i. This leads to the control scheme proposed amongst others in Fax and Murray [2004] and Pilz and Werner [2012a], which is shown in Figure 3.2.2.

For the application of robust controller design techniques to the control architecture shown in Fig. 3.2.2, the global control loop can be redrawn in LFT form as shown in Fig. 3.2.3 (with switch position (ii)). The interaction is expressed by $\mathcal{L}_{(m)} = I - \Psi_{(m)}$, of which Ψ is considered as uncertainty. The exogenous inputs are combined to the generalized

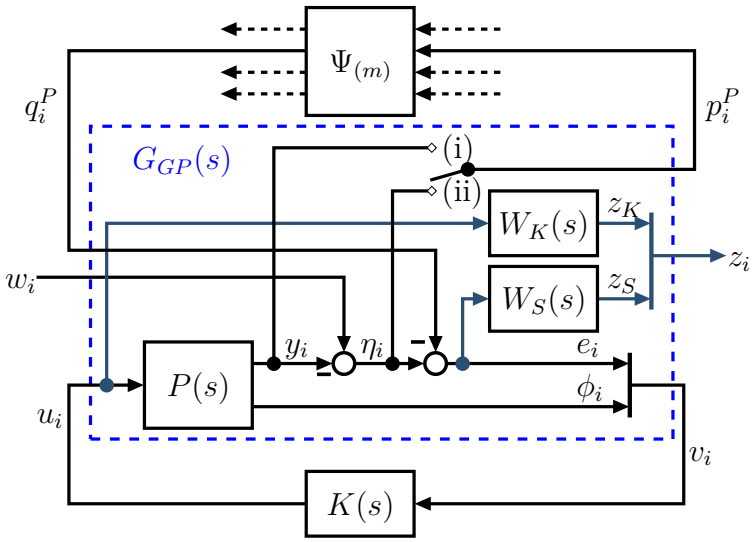


Figure 3.2.3: Generalized plant for an architecture with plant interaction

input $w^{(ii)} = r - d - n$. With this arrangement the distributed control problem has the form of a robust control problem, in which a controller K is to be found such that closed loop stability and performance requirements are robustly fulfilled for all Ψ in a specified set [Pilz and Werner, 2012a]. While the setup shown in Fig. 3.2.2 considers the error η_i to be transmitted to the neighbors (referred to here as *error data exchange*), other formulations such as in Hoffmann et al. [2015] consider the output y_i to be transmitted (*output data exchange*). In Fig. 3.2.3 the latter option is shown as switch position (i), error data exchange as (ii). As the transmission is modeled as a linear operator, $e = \mathcal{L}_{(m)}\eta = \mathcal{L}_{(m)}r - \mathcal{L}_{(m)}y$ holds and both formulations are equivalent. This can be easily seen from Fig. 3.2.2. For the output transmission formulation (i) the generalized input is $w^{(i)} = \mathcal{L}_{(m)}(r - d - n)$.

To allow the usage of loop shaping techniques, additional fictitious output channels are defined as the formation error e_i and the control signal u_i weighted by the shaping filters W_S and W_K [Pilz et al., 2009]. For this choice the transfer functions from w to z_S and z_K are obtained as the weighted formation sensitivity function $\hat{W}_S(s)S_F(s)$ and the weighted control sensitivity function $\hat{W}_K(s)KS_F(s)$. A typical choice of these shaping filters is shown in Fig. 3.2.4: For W_S a low-pass filter is chosen in order to penalize the formation error at low frequencies (corresponding to Objective 1 and 2 in Section 3.2.1), but to allow a transient error at high frequencies ($\omega \gg \omega_b$). The filter W_K penalizes the control effort (Objective 4) and is chosen as a high-pass filter, which imposes a higher penalty for high-frequency control action in order to avoid an amplification of measurement noise (Objective 3).

An extension of the architecture from Fig. 3.2.2 is shown in Fig. 3.2.5 and is provided by the idea of a direct data exchange among the controllers, which is introduced in Massioni and Verhaegen [2009]. The controller K is assumed to generate a separate coordination signal p_i^K, which is transmitted to the controllers of the neighbors using the same data transmission link as the plant coordination data. The composition of the coordination

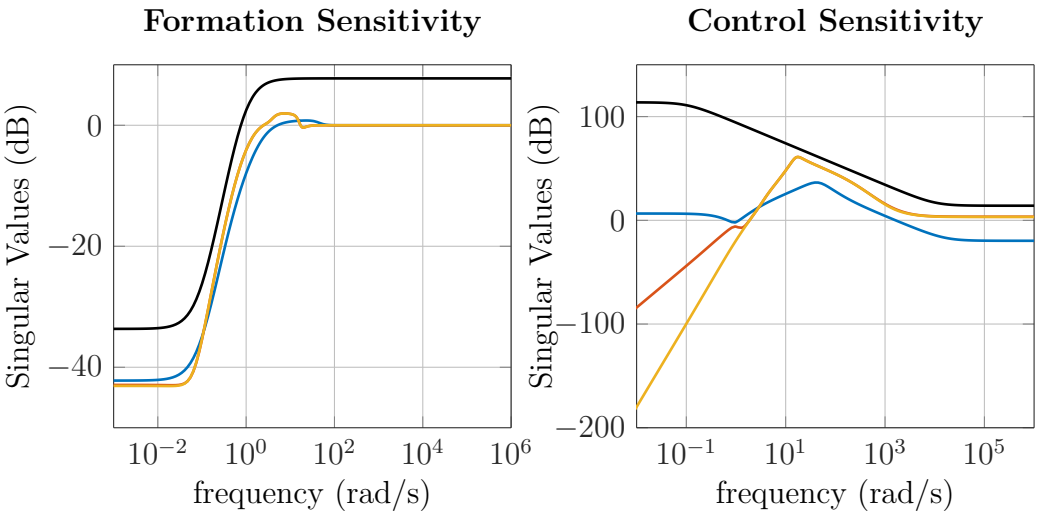


Figure 3.2.4: Typical sensitivity plots for design of a cooperative formation controller; singular values of the inverse shaping filter (black) and the sensitivity function (colored) plotted over frequency

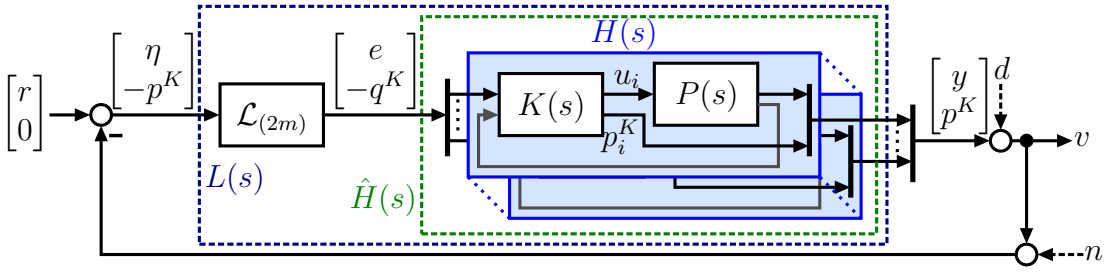


Figure 3.2.5: Global system with plant and controller interconnection

data p_i^K is defined by K and thus is obtained and optimized as part of the controller synthesis procedure.

Considering Ψ as uncertainty, an LFT representation can be derived with a copy of $\Psi_{(m)}$ LFT-wise connected to $K(s)$, as shown in Fig. 3.2.6. This corresponds to the control structure subject to the gain-scheduled control problem in Scherer [2000, 2001] and thus allows the application of techniques developed for this kind of problems.

A drawback of this architecture is that for each agent $2m$ instead of m data transmission channels have to be provided, which means an increased communication load. A modified architecture avoiding this drawback is proposed in Hoffmann et al. [2015], where the absolute error η_i is fed to the controller instead of the formation error e_i. The corresponding generalized plant is shown in Fig. 3.2.7. Within this generalized plant the interaction input q_i^P only influences the virtual output z_S and can thus be considered as virtual itself. For a practical implementation of this architecture, only the controller interaction data would be required to be transmitted.

Figure 3.2.8 shows an overview of the three proposed architectures.

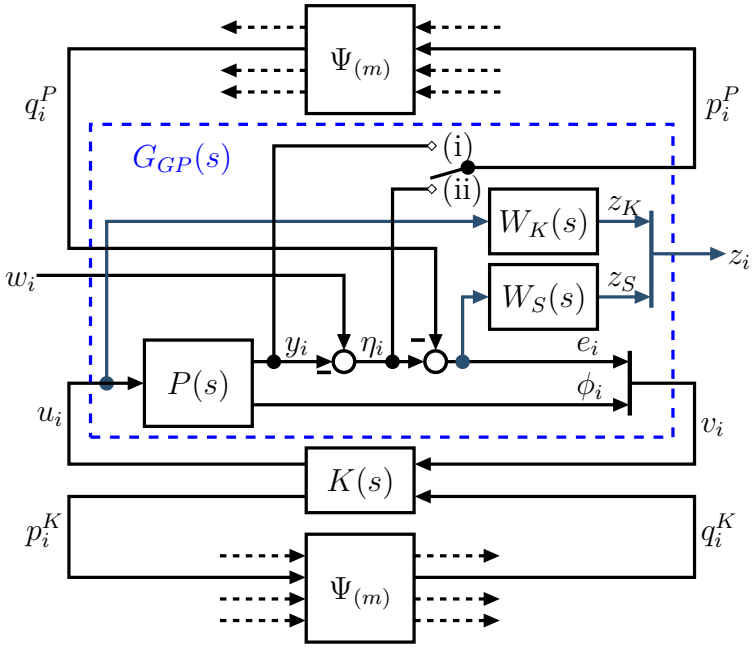


Figure 3.2.6: Generalized plant for an architecture with plant and controller interaction

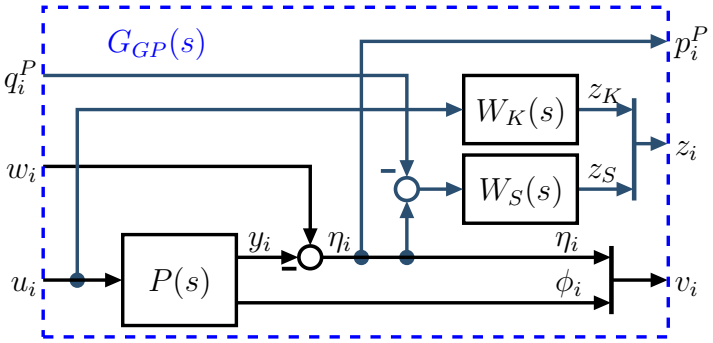
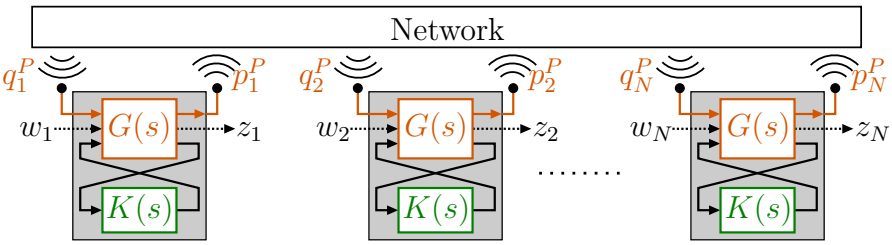


Figure 3.2.7: Generalized plant for an architecture with controller interaction and error data exchange

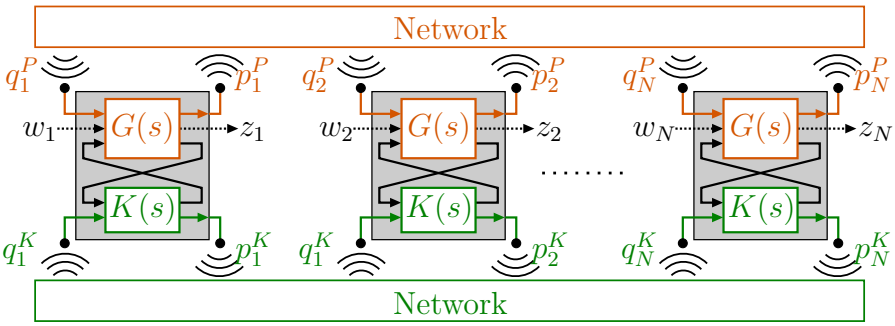
3.2.3 Consensus-Based Formation Control

An important property of the previously presented cooperative control architecture consists of having a single controller which is responsible for controlling both the interaction with neighbors and the local dynamics. As visible in Fig. 3.2.2, in this architecture the agents are coupled by linking the main control loops containing the controller and the local dynamics $P(s)$ of the particular agent.

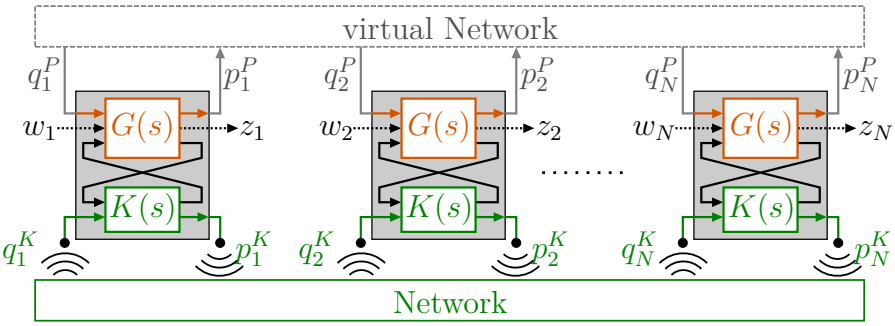
An alternative control architecture is introduced in Fax and Murray [2004]. The basic idea of this approach is to let the agents jointly determine a formation reference position to be tracked locally by each agent. In Pilz et al. [2011] this scheme is simplified and separated into a consensus loop and a local position control loop, as shown in Fig. 3.2.9. The task of the consensus loop depicted on the left is to estimate an absolute position reference $\hat{r}_i$ for each agent, which is fed to the local position control loop to be tracked by y_i. With



(a) Interconnected plants [Fax and Murray, 2004; Pilz et al., 2009; Pilz and Werner, 2012a]



(b) Interconnected plants and controllers [Massioni and Verhaegen, 2009]



(c) Interconnected controllers [Hoffmann et al., 2015]

Figure 3.2.8: Cooperative control architectures

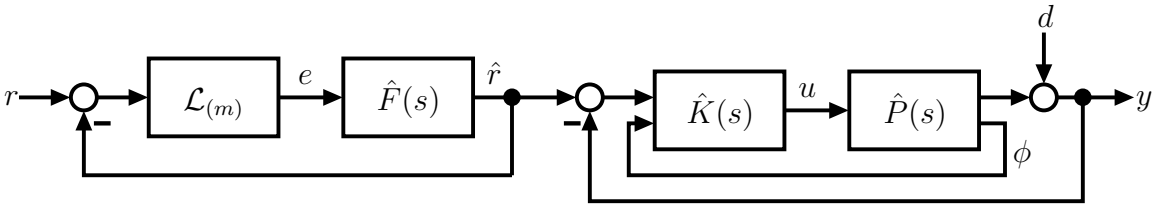


Figure 3.2.9: Consensus-based formation control scheme [Pilz et al., 2011]

$\hat{F}(z) = I \otimes F(z)$, this loop has the same structure as the general global control loop shown in Fig. 3.2.1 with $H(s) = F(s)$. Meanwhile, the position control loop shown on the right does not contain any interaction elements (i.e. all transfer function matrices are block-diagonal). Thus, it consists of N independent instances of a local position control

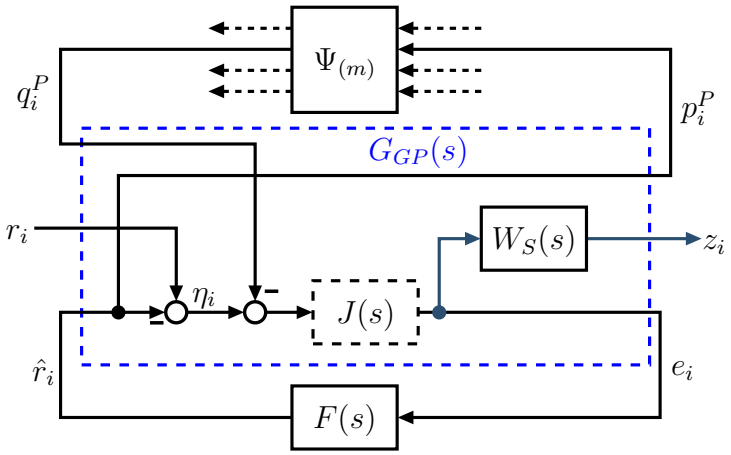


Figure 3.2.10: Generalized plant for information flow filter design [Pilz et al., 2011]

loop consisting of the agent model $P(s)$ and the local controller $K(s)$ enabling the agent to track the estimated reference position $\hat{r}_i$ determined by the information flow filter.

A major advantage of this setup is the separation of interaction dynamics and local dynamics, which allows them to run in separate time scales. This separation is similar to the reduced formulation by time-scale separation in networked systems, which is proposed and theoretically analyzed in Awad et al. [2015]. An important case to consider here is a consensus loop reacting significantly faster than the agent dynamics, for which the overall performance is dominated and limited by the agent dynamics (later referred to as *fast consensus*). As another benefit of the consensus-based approach, $K(s)$ and $F(s)$ can be designed independently, as no feedback connection between position control part and consensus loop exists. For the local position controller, this allows the usage of well-established techniques without considering the interaction with neighbors, also non-linear control schemes such as LPV gain-scheduled controllers are admissible [Gonzalez Cisneros et al., 2015].

In this architecture the dynamics of the interaction are designed by designing the information flow filter $F(s)$. The goal is to find an F which provides good estimated absolute reference positions $\hat{r}$ for which the agents are in formation according to the given formation reference r. This problem is analogous to the formation control problem defined in Section 2.1.4 with $\hat{r}_i$ considered as agent position y_i. The objective of good reference tracking is measured by the consensus sensitivity function

$$S_F(s) = \mathcal{L}_{(m)}(I + \hat{F}(s)\mathcal{L}_{(m)})^{-1} \tag{3.2.6}$$

describing the relation of the formation reference input r and the relative estimation error e (also referred to as *reference disagreement*). Good reference tracking corresponds to $\bar{\sigma}(S_F) \approx 0$ for low frequencies.

For the application of loop shaping techniques to design the information flow filter, the consensus loop can be reformulated into the generalized plant form shown in Fig. 3.2.10 and proposed in Pilz et al. [2011]. Similar to the cooperative control architecture, here Ψ is considered as uncertainty. The Information Flow Filter (IFF), which is the system to be

designed, is considered as controller. The fictitious performance output z_i is defined as the estimation error e_i weighted by the shaping filter W_S, such that the weighted consensus sensitivity function $\hat{W}_S S_F$ is obtained as transfer function from reference r to z.

An important difference between the information flow filter design problem and the single-loop cooperative design problem is the absence of a dynamic plant model within the generalized plant of the IFF design problem. In terms of standard robust control terminology, this results in a direct throughput from the control input (which in this case is the estimated reference $\hat{r}$) and the measured output fed to the controller. For this reason the control sensitivity function equals the complementary sensitivity function, such that an explicit penalization of the control sensitivity is not meaningful and thus omitted here. When using LMI-based synthesis tools for IFF design with this problem formulation, the direct throughput from $\hat{r}_i$ to e_i can lead to difficulties. To avoid these, a low-pass filter $J(s)$ can be added as shown in Fig. 3.2.10 with a reasonably high bandwidth.

Direct IFF Design Regarding the formulation as an optimization problem, the absence of plant dynamics has an important impact: As complementary sensitivity and control sensitivity are equal, in a multi-objective formulation as in Fig. 3.2.4 the sensitivities to be shaped (S_F and KS_F) are complementary to each other. Thus, both objectives are no longer independent and the resulting optimization problem would reduce to finding the minimal positive real value γ fulfilling $\|\hat{W}_S S_F\|_\infty < \gamma$. As this problem has arbitrarily many solutions ("everything which is below $W_S^{-1}(s)$"), it is no longer meaningfully formulated.

In the context of this difficulty in the optimization problem it is good to know that in most cases it is not even reasonable to tackle the information flow filter design problem as an optimization problem at all. Instead, having the shaping filter W_S given to specify desired properties of the formation sensitivity function, $F(s)$ can be obtained directly:

Theorem 3.2.1. *Consider a multi-agent system as shown in Fig. 3.2.9 consisting of N agents equipped with an information flow filter $F(s) = W_S(s) - I$ and a local controller $K(s)$ stabilizing the agent dynamics $P(s)$. This system is stable for any number of agents N and arbitrary and switching communication topologies with any time-varying communication delays, if W_S satisfies*

$$\|W_S^{-1} - I\|_1 < 1. \tag{3.2.7}$$

Proof: The consensus loop of the considered multi-agent system can be represented as an LFT interconnection of a generalized plant $G_F(s)$, the information flow filter $F(s)$ and the adjacency matrix Ψ as shown in Fig. 3.2.10 (with $J = I$). We consider $T(s)$ to be the LFT interconnection of a single instance of the generalized plant and the information flow filter $F(s)$. For this system the closed-loop the following transfer functions are obtained:

$$T_{er} = (I + F)^{-1} = W_S^{-1}, \tag{3.2.8}$$

$$T_{pq} = -FT_{er} = -(W_S - I)W_S^{-1} = W_S^{-1} - I. \tag{3.2.9}$$

Accordingly, assumption (3.2.7) implies $\|T_{pq}\|_1 < 1$. From this finding the proof can be concluded using Theorem 3.1.1. ■

The role of W_S in this case is similar to that in the usual optimal design problem. However, here the information flow filter is obtained directly, such that the local formation sensitivity function exactly equals the shaping filter, corresponding to $\gamma = 1$.

Design Objectives for Information Flow Filter For the consensus-based formation control architecture the decoupling of interaction and local dynamics, provides the possibility to design the information flow filter, which determines the dynamics of the consensus process, independently of the dynamics of the agents. For holonomic agents to be controlled in cartesian space, it is furthermore possible to handle the consensus scheme separately for each dimension, such that only SISO design problems are to be solved.

The main property of the consensus loop to be influenced is its closed loop bandwidth ω_C, which determines how fast the agents can adapt their reference position $\hat{r}_i$ to changes of the formation reference r or the reference position of the neighbors. With respect to the bandwidth ω_{Pcl} of the local position control loop, the choice of ω_C can follow one of the following strategies:

$\omega_C > \omega_{Pcl}$ **Fast Consensus** The goal of this strategy is to achieve formation consensus as fast as possible, such that the bandwidth of the overall system is dominated by the local bandwidth ω_{Pcl}. This is reasonable if the physical capabilities of the agents are considered as bottleneck and the reference r is to be tracked as quickly as possible.

$\omega_C < \omega_{Pcl}$ **Slow Consensus** With this strategy the bandwidth of the entire system is to be determined in a range below the local bandwidth by designing the information flow filter. In contrast to the fast consensus strategy, here the local agents are able to track the position reference $\hat{r}_i$ almost exactly, as the high-frequency content of the formation reference is already attenuated by the consensus loop. This strategy is reasonable for example in teams of heterogeneous agents, for which more agile agents are to be prevented from executing fast movements which their team mates are not able to perform. Another usage case is to cope with low data rates of the communication channels preventing higher bandwidths of the consensus loop by limiting the sampling rate.

$\omega_C \approx \omega_{Pcl}$ **Matched Consensus** The idea of this strategy is to choose the consensus bandwidth (almost) equal to the local bandwidth and is a compromise of the two extreme strategies above.

The strategies of fast and slow consensus are illustrated in the following example:

Example 3.2.1. A consensus-based formation control scheme according to Fig. 3.2.9 is considered with point mass agents ($P(s) = m/s^2$) controlled by a first order lead

compensator. The information flow filter is obtained directly as

$$F(s) = W_S(s) - I \quad \text{from choosing } W_S(s) = \frac{s + \omega_C}{s}. \tag{3.2.10}$$

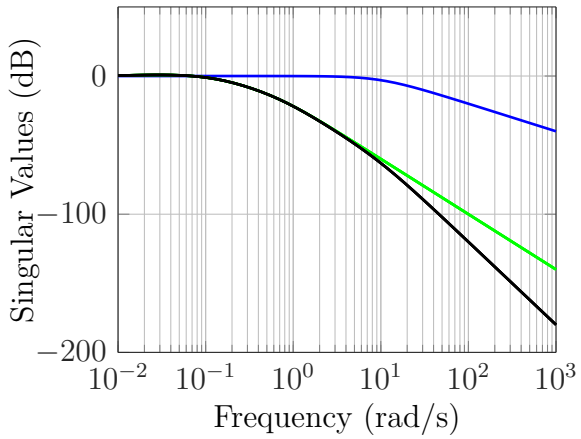


Figure 3.2.11: Fast Consensus

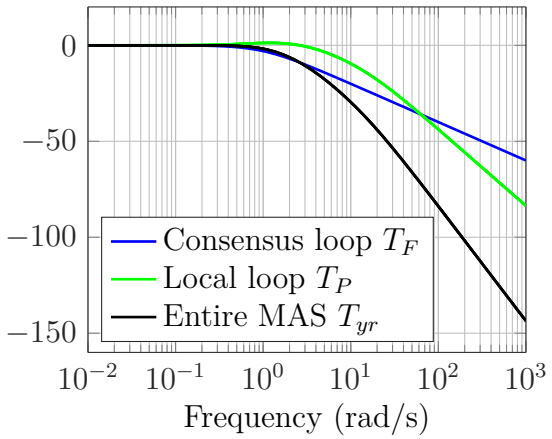


Figure 3.2.12: Slow Consensus

Fig. 3.2.11 and 3.2.12 show the local frequency responses of the transfer functions of the consensus loop $T_F = F(I + F)^{-1}$, of the locally controlled agent $T_P = \mathrm{Tf}(\hat{r}_i \to y_i)$ and of the combination of both ($T_{yr} = T_P T_F$). For the fast consensus case a heavy agent with a low bandwidth is considered, while for the consensus scheme a bandwidth $\omega_C = 10 rad/s$ is chosen. As clearly visible, for frequencies up to ω_C the frequency response of the entire system is almost equal to that of the local agent. Thus, the performance of the entire system is only limited by the physical capabilities of the agents. In contrast, for the slow consensus case, where more agile agents are considered, ω_C is chosen lower than the agent bandwidth. This causes the response of the entire MAS to decrease for frequencies above ω_C, although the agent bandwidth is significantly higher.

Time responses for formation establishment are given for both cases in Fig. 3.2.13 and 3.2.14, where the position y_i (solid) and the estimated position reference $\hat{r}_i$ (black dotted) are shown in x-direction. As clearly visible, for the slow consensus case the agents track their estimated references almost exactly. In contrast, for the fast consensus case a formation consensus is achieved quickly, but the agents need a comparatively long time to reach their target positions.

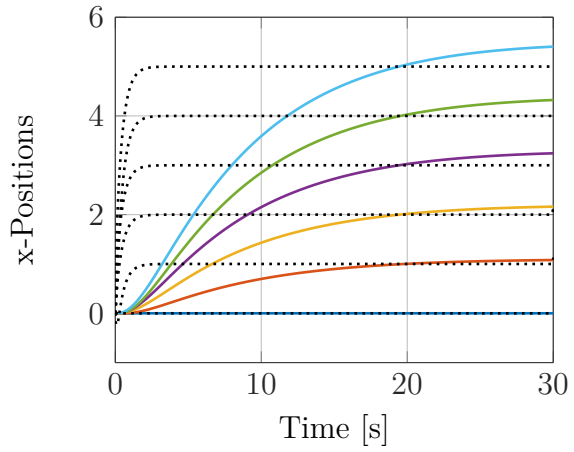


Figure 3.2.13: Fast Consensus

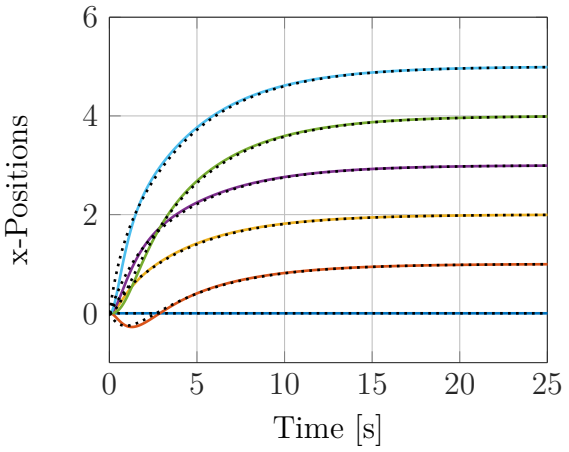


Figure 3.2.14: Slow Consensus

3.2.4 Extended Consensus-Based Control Scheme

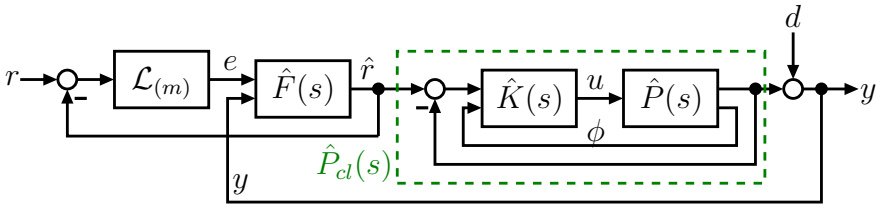


Figure 3.2.15: Extended information flow scheme

Despite the advantages of fast consensus and separate controller design, the consensus-based control scheme has an important drawback: As no feedback exists from the actual agent output to the consensus scheme, there is no possibility for the agents to react on any disturbances acting on neighbor agents. In the following, an extension of the consensus-based control scheme is introduced with the goal to combine the performance and design advantages of the consensus-based scheme with the ability of the agents to react on disturbances of other agents.

This scheme is shown in Fig. 3.2.15. Unlike in the consensus-based scheme, where information flow filter and position controller are only connected in a feed-forward way, here the agent position is fed back locally to an extended information flow filter $F(s)$. Thus, the coordination signal $\hat{r}$ can be influenced by the agent's output, which permits to transport information about an output disturbance d_y acting on the agent and enables the other agents to react on that. Nevertheless, $\hat{r}$ is used as reference estimation for the position controller.

For this architecture the local position control loop is considered in the same manner as in the consensus-based scheme, thus it can as well be designed independently. Assuming that such a controller $K(s)$ has been designed according to the objectives of reference tracking, disturbance rejection, noise attenuation and reasonable control effort with respect to the local control loop, the control objectives for the information flow filter are

1. $\mathrm{Tf}(r \to \hat{r}) \approx 1$ for formation consensus
2. $\mathrm{Tf}(d \to \hat{r})$ small for disturbance rejection
3. $\mathrm{Tf}(d \to \hat{r}) \approx 1$ for information propagation

Objective 3 follows from the fact that $\hat{r}_i$ is the signal which is transmitted to the neighbor agents. Obviously, the objectives 2 and 3 are contradictory. However, these contradictory claims become reasonable when having a closer look on what disturbance rejection in case of a formation control problem means: The geometrical shape of the formation is to be restored after an agent has been driven out of its place by an external disturbance. This can be accomplished both by the disturbed agent moving towards its desired position and by the other agents moving the whole formation towards the place of the disturbed agent. For the latter movement, it is necessary to propagate the information of the displacement caused by the disturbance to all other agents of the formation. For quickly restoring the formation, especially the signal components around the bandwidth ω_{Pcl} of the position-controlled agent need to be passed to the neighbors, which results in claiming objective 3 for frequencies in this area. Instead, objective 2 mainly is a claim regarding the steady state accuracy. For this reason, the conflict is resolved by formulating the objectives in a frequency-dependent manner.

The aforementioned objectives are similar to the frequency-dependent objectives of tracking and control effort limitation encountered in standard mixed sensitivity design problems. Thus, here it seems natural to formulate a multi-objective design problem as well: Like for the decoupled consensus-based control scheme, for the extended version the quality of reference consensus is measured by a consensus sensitivity function S_F. In order to adjust the coupling between the agents, a *coupling sensitivity* function S_C is to be formulated as relation between the reference r and a *coupling error* signal e_C measuring how well the desired information to be fed to the neighbors is contained in the transmitted signal $\hat{r}$ (i.e. the lower e_C, the closer $\hat{r}$ is to the signal to be transmitted). This signal can be chosen in different ways: In Bartels and Werner [2014] it is proposed to choose the local position error $e_C = e_P = \hat{r} - y$. A straight-forward choice based on Objective 3 is $e_C = \hat{r} - d$, which is considered in the following. The sensitivity functions and the corresponding disturbance sensitivity functions are obtained as

$$S_F = \mathrm{Tf}(r \to e) = \mathcal{L}_{(m)}(I + \hat{F}_1\mathcal{L}_{(m)} - \hat{F}_2\hat{T}_P)^{-1}(I - \hat{F}_2\hat{T}_P) \tag{3.2.11a}$$

$$S_{Fd} = \mathrm{Tf}(d \to e) = -\mathcal{L}_{(m)}(I + \hat{F}_1\mathcal{L}_{(m)} - \hat{F}_2\hat{T}_P)^{-1}\hat{F}_2\hat{T}_{Pd} \tag{3.2.11b}$$

$$S_C = \mathrm{Tf}(r \to e_C) = (I + \hat{F}_1\mathcal{L}_{(m)} - \hat{F}_2\hat{T}_P)^{-1}\hat{F}_1\mathcal{L}_{(m)} \tag{3.2.11c}$$

$$S_{Cd} = \mathrm{Tf}(d \to e_C) = (I + \hat{F}_1\mathcal{L}_{(m)} - \hat{F}_2\hat{T}_P)^{-1}\hat{F}_2\hat{T}_{Pd} - I \tag{3.2.11d}$$

where F is partitioned as $F = \begin{bmatrix} F_1 & F_2 \end{bmatrix}$ according to its inputs e_i and y_i and the transfer functions of the position-controlled agents are given as block-diagonal functions

$$\hat{T}_P = \mathrm{Tf}(\hat{r} \to y) = (I + \hat{P}\hat{K})^{-1}\hat{P}\hat{K} \tag{3.2.12a}$$

$$\hat{T}_{Pd} = \mathrm{Tf}(d \to y) = \mathrm{Tf}(\hat{r} \to e_P) = (I + \hat{P}\hat{K})^{-1} \tag{3.2.12b}$$

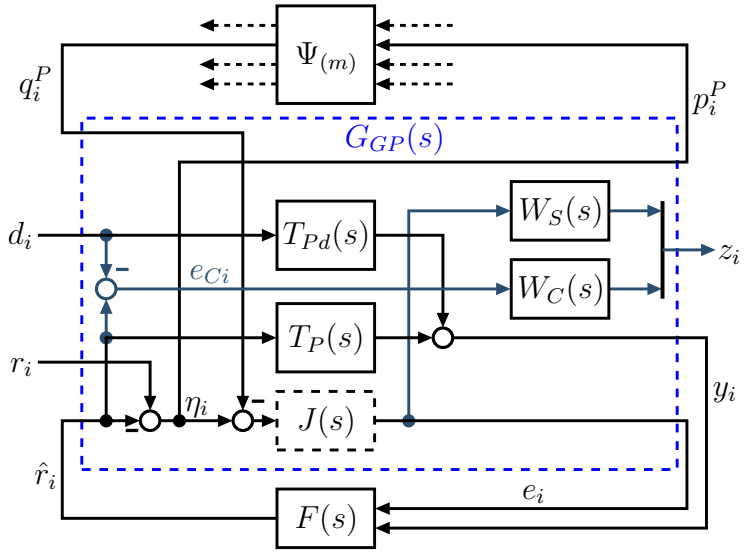


Figure 3.2.16: Generalized plant for extended information flow filter design

Details about obtaining these transfer functions are given in Appendix A.4.2. In terms of the sensitivity functions, the objectives can be summarized as follows:

$\omega \ll \omega_{Pcl}$	• $S_P \to 0$ for local reference tracking • $S_F \to 0$ for reference consensus • $S_{Fd} \to 0$ for disturbance rejection
$\omega \approx \omega_{Pcl}$	• S_{Cd} small for coupling • S_F small for fast reference consensus
$\omega \gg \omega_{Pcl}$	• $S_P \approx 1$ for noise rejection • $S_P K \to 0$ for reasonable control effort

The objectives concerning $S_P = \mathrm{Tf}(\hat{r}_i \to e_{Pi})$ and $S_P K$ are to be considered when designing $K(s)$. The synthesis problem for $F(s)$ can be formulated using the generalized plant setup shown in Fig. 3.2.16. Like in the decoupled IFF design, the shaping filter W_S is used to penalize the formation consensus error. To tune the coupling of the agents, a second performance channel z_C is introduced penalizing the coupling error e_C by a shaping filter $W_C(s)$. In contrast to the decoupled consensus-based architecture, here the consensus loop is influenced by the agent position y_i. Thus, the closed loop dynamics of the position-controlled agent have to be included into the IFF design problem, which is done by expressing y_i by $y_i = T_P \hat{r}_i + T_{Pd} d_i$ as given in (3.2.12).

An important aspect has to be considered in tuning the extended information flow filter

regarding the objectives in the active frequency range ($\omega \approx \omega_{Pcl}$): Enforcing small values for both S_F and S_{Cd} leads to a conflict. From (3.2.11b) and (3.2.11d) one can see that the complementary sensitivity corresponding to S_{Cd} only differs from S_{Fd} by the frequency-independent factor $-\mathcal{L}_{(m)}$. Thus, increasing the bandwidth of W_S for achieving faster consensus (increasing the bandwidth of S_F) also increases the bandwidth of S_{Fd}, which is shaped by W_S as well. The complementarity to S_{Cd} then leads to an increase of the bandwidth of S_{Cd} and moves its value at ω_{Pcl} towards 1, which contradicts the coupling objective. Thus, the choice of the shaping filter bandwidths is a trade-off between fast consensus and coupling.

Example 3.2.2. This example considers the control task from Example 3.2.1 to be accomplished by using the extended consensus-based control architecture shown in Fig. 3.2.15. To design the information-flow filter $F(s)$, a four-block mixed sensitivity problem is formulated using the generalized plant shown in Fig. 3.2.16. The agents are point masses with $m = 1$ controlled by the lead compensator used in Example 3.2.1. The shaping filters are chosen as

$$W_S(s) = \frac{3.333}{s + 0.01} I_3, \quad W_C(s) = 100 \frac{s + 1}{s + 1000} I_3. \tag{3.2.13}$$

This design problem is solved using the robust control approach from Pilz and Werner [2012a] based on the small gain theorem, which is reviewed in Section 3.1.1. Here Problem 3.1.2 is solved and the ℓ_1 norm is checked a posteriori.

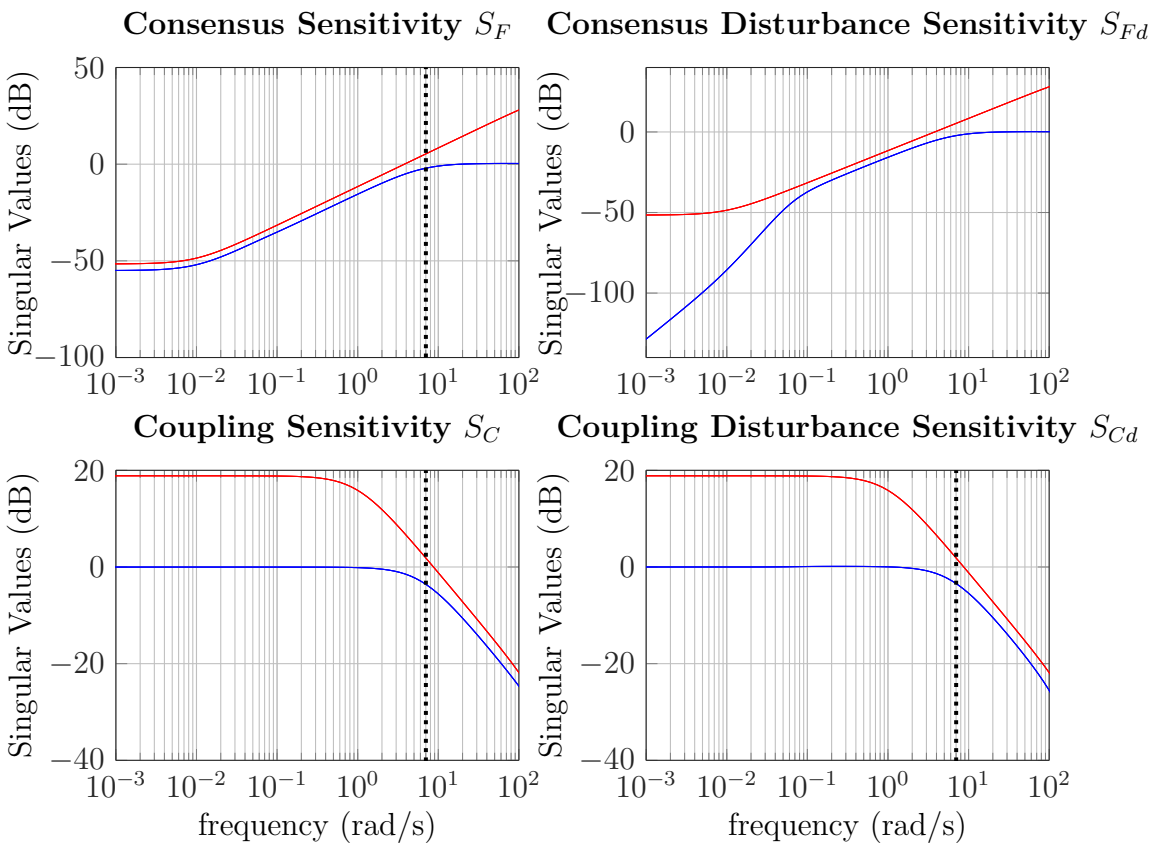


Figure 3.2.17: Sensitivity plots for design of an extended information flow filter; singular values of the inverse shaping filter (red) and the sensitivity function (blue) plotted over frequency

Fig. 3.2.17 shows the shaping filters together with the singular values of the resulting sensitivity functions plotted over frequency. The agent bandwidth ω_{Pcl} is indicated by a black dotted line. The design objectives regarding the steady state $\omega \ll \omega_{Pcl}$ are clearly fulfilled, as both S_F and S_{Fd} are small for low frequencies. The agent bandwidth almost coincides with the bandwidth of the consensus sensitivity S_F, which indicates a matched consensus case, i.e. the consensus loop reacts more or less as fast as the agents. At ω_{Pcl} the coupling disturbance sensitivity already has a value significantly below 1, such that good coupling can be expected.
Frequency responses of the resulting decomposed system ($\Psi = 0$) are plotted in Fig. 3.2.18 showing the transfer functions of the consensus loop $T_F = \mathrm{Tf}(r \to \hat{r})$, the local system $T_P = \mathrm{Tf}(\hat{r} \to y)$, the entire system $T_{yr} = \mathrm{Tf}(r \to y)$ and the coupling $T_{Fd} = \mathrm{Tf}(d \to \hat{r})$ (details see Appendix A.4.2). Corresponding to the matched consensus case observed in Fig. 3.2.17, the bandwidths of the consensus loop and the local loop almost coincide. The coupling transfer function is close to zero for very small frequencies, but close to 1 for high frequencies. Thus, in steady state the disturbance has almost no influence on the commanded position. T_P and T_{Fd}

intersect with magnitudes close to 1, which means that an active frequency range exists in which the agent is able to react on disturbances acting on neighbors.

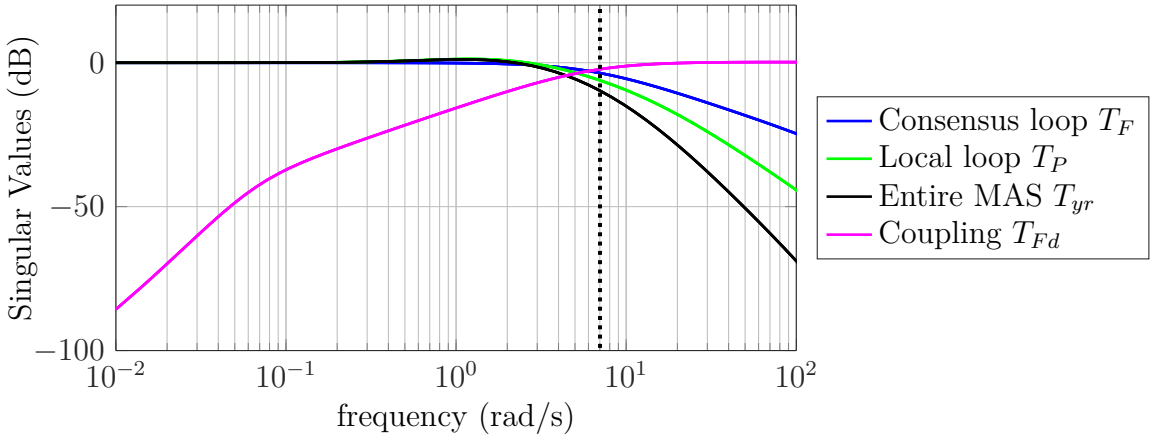


Figure 3.2.18: Singular value plots of the input-output transfer functions

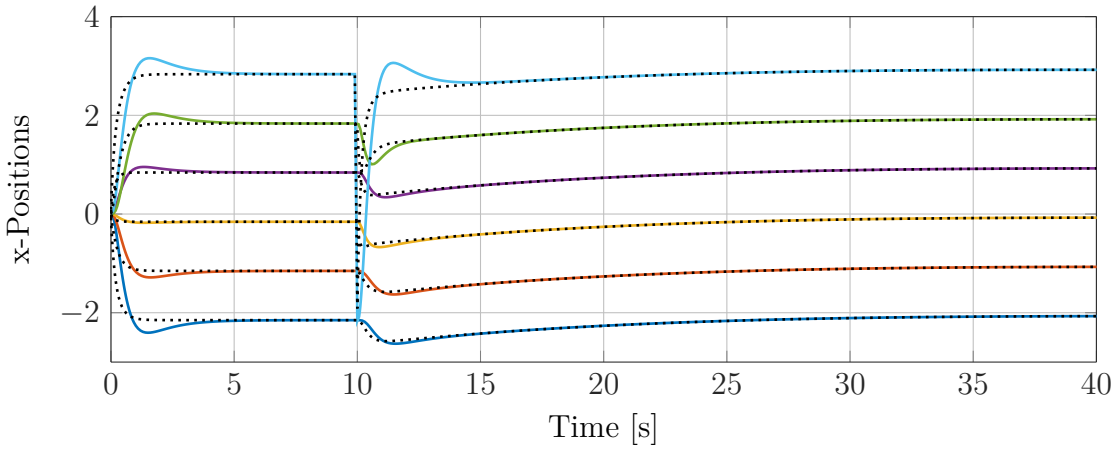


Figure 3.2.19: Position in x-direction (solid) and estimated reference $\hat{r}$ (black dotted), plotted over time for formation establishment and disturbance rejection

A simulated time response of the entire multi-agent system is shown in Fig. 3.2.19. The simulated scenario considers a group of six agents commanded to achieve a line formation with a spacing of 1. At $t = 10\,\text{sec}$ an output disturbance $d_{6,x} = -5\sigma(t - 10\,\text{sec})$ is acting on agent 6. As clearly visible, the other agents react on this disturbance and quickly move in negative x-direction. The formation is restored within approx. 4 seconds.

	(i)	(ii)	(iii)
Generalized Plant	Fig. 3.2.3 with $\phi_i = x_i$	Fig. 3.2.10	Fig. 3.2.16, modified by choosing $e_C = \hat{r} - y$
Shaping Filters	$W_S = \frac{1}{s+10^{-5}} I_3$ $W_K = \frac{s+0.1}{s+100} I_4$	$W_F = \frac{1}{3}\frac{0.01}{(z-1)} I_3$	$W_F = \frac{1}{3}\frac{0.01}{(z-1)} I_3$ $W_C = \frac{5z-4.995}{z} I_3$

Table 3.1: Synthesis specifications

3.3 Comparison and Benchmark

3.3.1 Comparison of Control Architectures

The simulation study presented in this section aims at a comparison of (i) the robust cooperative control scheme introduced in Section 3.2.2, (ii) the decoupled consensus-based control scheme from Section 3.2.3 and (iii) its extension presented in Section 3.2.4. The controllers used to generate the presented results were synthesized by means of the small gain theorem-based robust approach presented in Section 3.1.1. Specifications of the synthesis problem are given in Table 3.1.

The scenario simulated here is adopted from Pilz and Werner [2012a] and considers a group of five identical quad-rotor helicopters modeled by the linearized 12-th order model given in Section B.1, which is discretized here with a sampling time of 10ms. This group of agents is guided by a virtual leader with a given position. Although the simulation considers a 3D space, here only a change in the x-coordinate is commanded. Starting positions and desired relative placements of the quad-rotor agents are given as

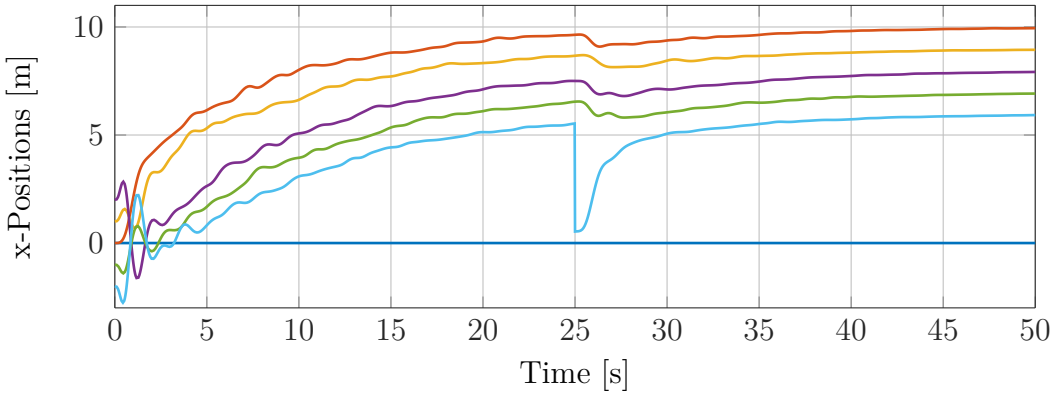
$$\begin{bmatrix} x_1(0) & x_2(0) & x_3(0) & x_4(0) & x_5(0) \end{bmatrix}^T = \begin{bmatrix} 0m & 1m & 2m & -1m & -2m \end{bmatrix}^T \tag{3.3.1}$$

$$\begin{bmatrix} r_{1,x} & r_{2,x} & r_{3,x} & r_{4,x} & r_{5,x} \end{bmatrix}^T = \begin{bmatrix} 10m & 9m & 8m & 7m & 6m \end{bmatrix}^T. \tag{3.3.2}$$

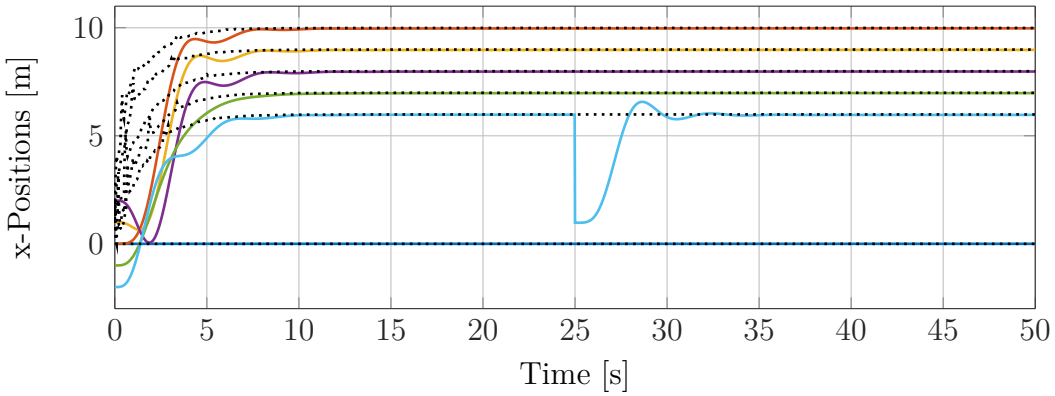
The communication topology is chosen as unweighted graph with an interaction matrix partitioned as

$$\Psi = \begin{bmatrix} \Psi_{FF} & \Psi_{FL} \\ 0 & 0 \end{bmatrix}, \quad \Psi_{FL} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \end{bmatrix}^T, \tag{3.3.3}$$

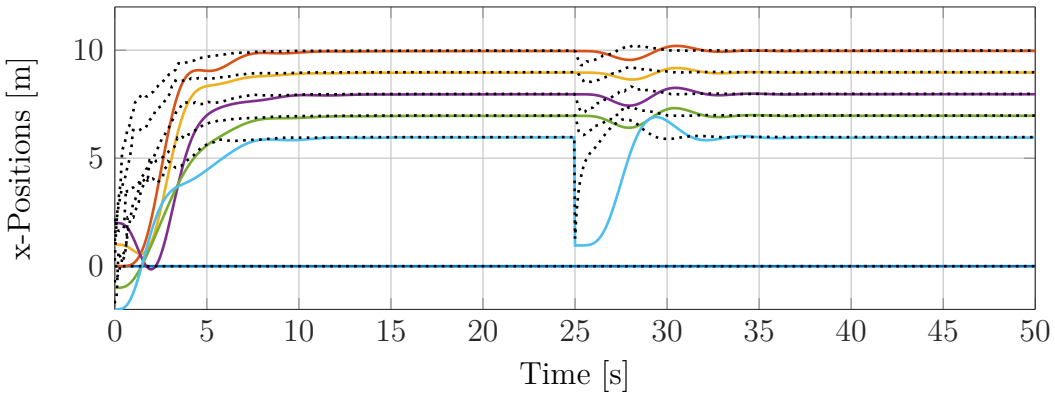
where Ψ_{FF} describes the topology among the followers and Ψ_{FL} describes that the virtual leader is connected to the agents 1 and 2. The topology among the followers is chosen randomly and is updated every 0.1s. Additionally, the communication links are affected by random time delays in the range of 1 to 6 sampling instances. At $t_d = 25$s agent 5 is affected by a step disturbance $d_y = -5\sigma(t - t_d)$ acting on its output.



(a) Cooperative control scheme



(b) Consensus-based control scheme



(c) Extended consensus-based control scheme

Figure 3.3.1: Position $y_{i,x}$ (solid) and estimated reference $\hat{r}_i$ (dotted) for different control architectures

The results in terms of the agents' positions in x-direction are shown in Fig. 3.3.1 for the three examined formation control schemes. For both cooperative schemes the estimated reference $\hat{r}$ is plotted as well. Comparing the time to reach the formation, one can see that for the cooperative architecture (Fig. 3.3.1a) an approximately three times larger settling time is needed compared to the consensus-based architectures. Additionally,

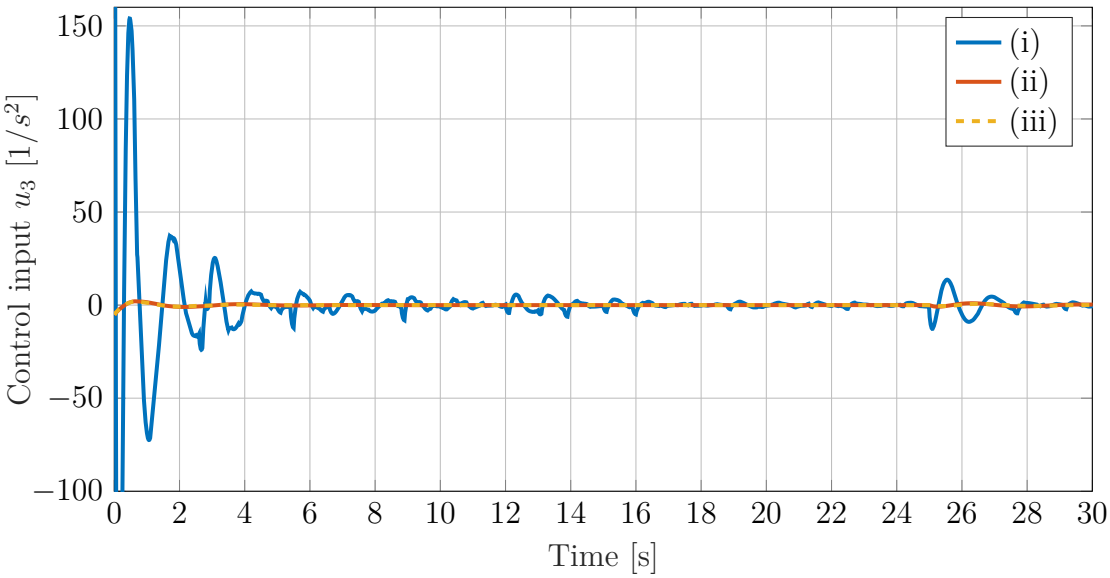


Figure 3.3.2: Control input u_3 of agent 5

here a significant dithering is visible during the whole movement, which suggests a high sensitivity to the topology changes. In contrast, the results shown in Fig. 3.3.1b and 3.3.1c are smooth and do not noticeably show such effects.

Rejecting the disturbance on agent 5 at $t = 25$s is performed rather quickly in all cases, where consensus-based and extended IFF formation control show a noticeable overshoot. As expected, in the fully decoupled consensus-based case (ii) the neighbor agents do not show any reaction on the disturbance. In both other cases a significant attempt to follow the disturbed agent is visible. As described in Section 3.2.3, the behavior of case (ii) is explained by the feed-forward architecture, in which no actual position data are transmitted to neighbor agents. In contrast, both for the cooperative scheme and the extended consensus-based scheme, the coordination data depend on the actual position data, such that the neighbor agents notice the disturbance on agent 5 and are able to react on it.

In Fig. 3.3.2 the control input u_3 representing the pitch torque around the y-axis, which is the control input mainly responsible for movements along the x-axis, is shown. As is clearly visible, for the cooperative scheme the control signal reaches much higher values than in the other cases and can be assumed to go beyond the actuator limits. Referring to the output results, in case of the consensus-based and extended consensus-based schemes a significantly better transient performance is reached with significantly less control effort.

From these observations the consensus-based and the extended consensus-based control schemes appear to clearly outperform the cooperative scheme. The only major advantage of the cooperative approach over the consensus-based one, which is the ability of the formation to react to disturbances of a single agent, turns out to be achievable with the extended consensus-based scheme at a significantly lower price in terms of performance. Comparing these results to those of the combined architecture shown in Pilz and Werner

[2012b] and Pilz [2013], the extended consensus-based scheme shows significantly better performance in terms of settling time and formation maintenance.

Further tests have shown that in most cases controllers synthesized by solving the $\mathcal{H}_\infty/\mathcal{H}_\infty$ problem 3.1.2 pass the *a posteriori* check of the ℓ_1 condition and provide the same performance compared to those obtained from solving the full $\mathcal{H}_\infty/\ell_1$ problem 3.1.1. Thus, as long as the results fulfill the ℓ_1 condition, solving Problem 3.1.2 can be seen as preferable method for the robust design approach.

3.3.2 Automated Controller Tuning

Although the controllers obtained as results of $\mathcal{H}_\infty$-optimal synthesis techniques achieve the optimal performance in the sense of the specified optimization problem, the way how this problem is formulated is not unique and has significant influence on the resulting performance. For problem formulations as given in Section 3.2 the tuning mainly consist of the choice of shaping filter coefficients.

In the following section, different synthesis techniques for cooperative controllers according to Section 3.2.2 will be examined comparing the achievable performance in a simulation study. In order to achieve a good comparability of the simulation results generated with the examined control approaches, the goal is to find for each approach individually that set of coefficients, which achieves the best performance in a common, pre-defined simulation scenario with respect to a common performance measure. As the coefficient-to-performance relationship including the behavior of the solver algorithm and the simulation scenario is difficult to describe and varies between the examined approaches, in this study an automated tuning technique was used based on a simple genetic algorithm Zalzala and Fleming [1997].

The present study considers synthesis problems with generalized plants constructed as in Fig. 3.2.3 and 3.2.6 with shaping filters W_S and W_K chosen as

$$W_S(s) = \frac{c_S\omega_S}{M_S} \cdot \frac{(s+\omega_S)^2}{(s+c_S\omega_S)^2}; \quad 0 < c_S < 1, \tag{3.3.4}$$

$$W_K(s) = \frac{c_K}{M_K} \cdot \frac{s+\omega_K}{s+c_K\omega_K}; \quad c_K > 1. \tag{3.3.5}$$

Consequently, the result to be found by the automated tuning procedure is the tuning parameter vector

$$c = \begin{bmatrix} M_S & \omega_S & c_S & M_K & \omega_K & c_K \end{bmatrix}^T \tag{3.3.6}$$

containing the coefficients of the shaping filters defined in (3.3.4) and (3.3.5). The problem of performance optimization is formulated here as minimization of the objective function $J(c)$ defined as the sum of the agent objective functions $J_i(c)$ for all agents i:

$$J(c) = \sum_{i \in \mathcal{V}} J_i(c), \tag{3.3.7}$$

$$J_i(c) = a_e\|e_i(t)\| + a_u\|u_i(t)\| + a_\infty e_{i,\infty} + a_{tr}m_{tr,i} + a_d m_{d,i}. \tag{3.3.8}$$

This function consists of the weighting factors $a_\bullet$ and expressions describing the criteria

- Error norm: $\|e_i(t)\| = \sqrt{\int_{t=0}^{\infty} |e_i(t)|^2 dt}$
- Control effort $\|u_i(t)\| = \sqrt{\int_{t=0}^{\infty} |u_i(t)|^2 dt}$
- Steady state error $e_{i,\infty} = |e_i(t \to \infty)|$
- Tracking overshoot $m_{tr,i} = \dfrac{\bar{y}_{tr,i} - r_i}{r_i}$
- Disturbance rejection overshoot $m_{d,i} = \dfrac{\bar{y}_{d,i} - r_i}{r_i}$

where $\bar{y}_{tr,i}$ and $\bar{y}_{d,i}$ are the peak values of $y_i(t)$ after a reference step change and an output disturbance step, respectively. For a reasonable comparison the weighting factors have been chosen equally for all examined approaches.

Starting with an initial population of individuals c_0, at iteration k for each individual the controller synthesis procedure is executed and the result is tested in a simulation of the scenario described in Section 3.3.3 using Simulink. The obtained results are evaluated by computing the value $J(c_k)$ from the data generated by the simulation. Based on this, a set of child individuals c_{k+1} is generated using the reproduction methods elitism, crossover and mutation [Zalzala and Fleming, 1997]. The whole procedure is repeated iteratively.

For reproduction into a new generation of individuals c_{k+1}, parents c_k are selected by ranking with respect to the objective function J. The fittest n_{elit} individuals are copied directly into the next generation (elitism). Furthermore, from the n_{par} fittest individuals forming a parent group, crossover is performed to produce n_{Xover} children. In addition, n_{mut} children are generated by random mutation. Commonly, mutation is performed by additive superposition of the parent individuals' parameters and random values [Zalzala and Fleming, 1997]. As the parameters to be found in this work are in terms of corner frequencies and magnitudes, which are desired positive and cover several orders of magnitude, it is more suitable to modify the mutation mechanism such that the additive mutation takes place in logarithmic scale:

$$c_{k+1,j} = c_{k,j} \cdot 10^{\xi}, \tag{3.3.9}$$

where ξ is a zero-mean random value. With $\log_{10}(c_{k+1,j}) = \log_{10}(c_{k,j}) + \xi$, in a double-logarithmic plot of the shaping filter this mechanism (3.3.9) appears as an additive shift.

Figure 3.3.3 shows the results of the automated tuning algorithm applied to the ISC approach for seven reproduction steps. In this case a population size of $n = 6$ was used, the initial population was created manually including (but not only) promising results of manual tuning. The following reproduction settings were applied:

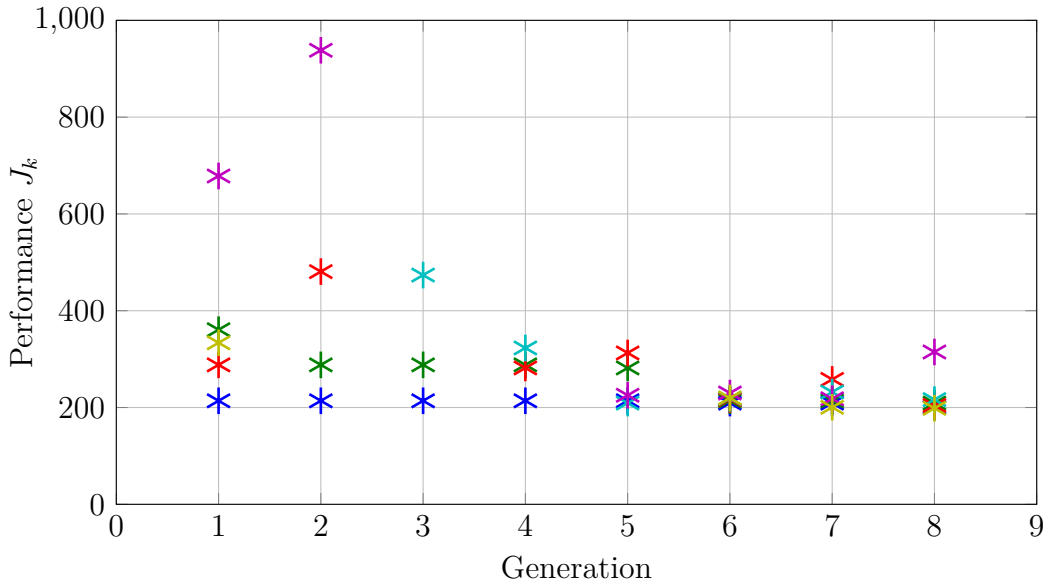


Figure 3.3.3: Output of the tuning algorithm for the ISC approach in terms of the objective function $J_k(c)$, 6 individuals

n_{elit}	n_{par}	n_{Xover}	n_{mut}
2	4	2	2

It is clearly visible that already after seven reproduction steps the performance values (in terms of $J_{k,i}$) of most individuals are very close together. This indicates good convergence properties of this algorithm. A reasonable convergence after a low number of steps is necessary in this study, as the number of iteration steps as well as the population size need to be kept low. The reason is the high computational effort, as in each iteration step both synthesis procedure and full simulation have to be done n times (once for every individual).

3.3.3 Comparison of Synthesis Techniques

This section presents the results of a simulation study comparing the achievable performance of the robust approach from Pilz and Werner [2012a] and the interconnection-scheduled approach from Hoffmann et al. [2015] presented in Chapter 3.1. The focus of this study is set on the application of these approaches to the synthesis of cooperative controllers according to Section 3.2.2. The synthesis problem is formulated based on a generalized plant constructed as in Section 3.2.2. For both approaches the tuning parameters of the shaping filters are obtained from the evolutionary tuning algorithm presented in the previous Section 3.3.2. The weighting factors for tuning the cost function of this algorithm are chosen as shown in Table 3.2.

a_e	a_u	a_∞	a_{tr}	a_d
1	5e-4	50	2	200

Table 3.2: Weighting factor values

Synthesis approach	**computation time [sec]**
Robust control, $\mathcal{H}_\infty/\mathcal{H}_\infty$ synthesis	24.45
ISC, $\mathcal{H}_\infty$ synthesis	10.37

Table 3.3: Computation time for controller synthesis

Discussion of the Synthesis Approaches Here some important features and differences of the examined synthesis approaches are discussed. While for the robust control approach the composition of the coordination data is specified in the generalized plant setup, the Interconnection-Scheduled Control (ISC) approach considers the coordination data as output signals of the controller. Accordingly, they are specified by the system matrices C_p^K, D_{pv}^K and D_{pq}^K of the controller (2.1.3). In contrast to the fixed generalized plant matrices, the latter are determined by the synthesis algorithm. This fact provides additional freedom for optimization and can be expected to reduce conservatism. In addition, as the ISC synthesis problem is adopted from LPV gain scheduling control, it can be easily extended to be combined with actual scheduling parameters, such that also LPV subsystems can be handled [Hoffmann et al., 2014].

In case of the robust control approach, communication delays are considered by the synthesis problem, such that robustness against arbitrarily high delays is guaranteed. The interconnection-scheduled approach does not provide any robustness guarantee against delays. In return, relinquishing the demand of such a formal guarantee can be expected to reduce the conservatism of the synthesis problem.

Regarding the computation effort, the benefit of the interconnection-scheduled approach is a convex LMI problem, which can be efficiently solved with medium effort. For the robust control approach, instead of the $\mathcal{H}_\infty/\ell_1$-Problem 3.1.1 in most cases it is sufficient to solve the $\mathcal{H}_\infty/\mathcal{H}_\infty$-Problem 3.1.2 instead, for which the computation effort relatively small. However, this does not immediately provide any stability guarantee and the corresponding ℓ_1 condition (3.1.4) has to be checked *a posteriori*. If it is violated, a computationally expensive iterative procedure has to be applied, which increases the controller order. In this study even solving the $\mathcal{H}_\infty/\mathcal{H}_\infty$ problem exceeded the computation time of the ISC synthesis (details are given in Table 3.3). For this reason the ISC approach can be considered as the computationally more attractive choice.

Simulation Scenario The benchmark scenario considered in this study is similar to the scenario examined in the previous simulation study of Section 3.3.1. Its subject is a formation flight of five identical quad-rotor helicopters as followers (index F) and a non-dynamic leader agent (index L) with given position trajectory. Accordingly, the topology

is structured as

$$\Psi = \begin{bmatrix} \Psi_{FF} & \Psi_{FL} \\ 0 & 0 \end{bmatrix}, \tag{3.3.10}$$

where Ψ_{FF} is the topology among the followers and Ψ_{FL} describes which follower agent receives data from the leader. Among the followers a directed topology (described by Ψ_{FF}) is randomly chosen and updated every 5 seconds, Ψ_{FL} is manually specified and time-invariant. The followers are modeled by the linearized model given in Section B.1. Their goal is to achieve and maintain a given formation with relative positions

$$\begin{bmatrix} r_{1,x} & r_{2,x} & r_{3,x} & r_{4,x} & r_{5,x} \end{bmatrix}^T = \begin{bmatrix} 1m & 2m & 3m & 4m & 5m \end{bmatrix}^T. \tag{3.3.11}$$

with respect to the leader. In addition, agent 5 is affected by an output disturbance $d_{i,x}(t) = -5 \cdot \sigma(t - t_{\text{dist}})$. Although motion is modeled in 3D, here we display only results in the x-direction. Unless stated otherwise, the local error is transmitted ($p_i^P = \eta_i$, switch position (ii)), the existing links are assumed ideal and no communication delays occur.

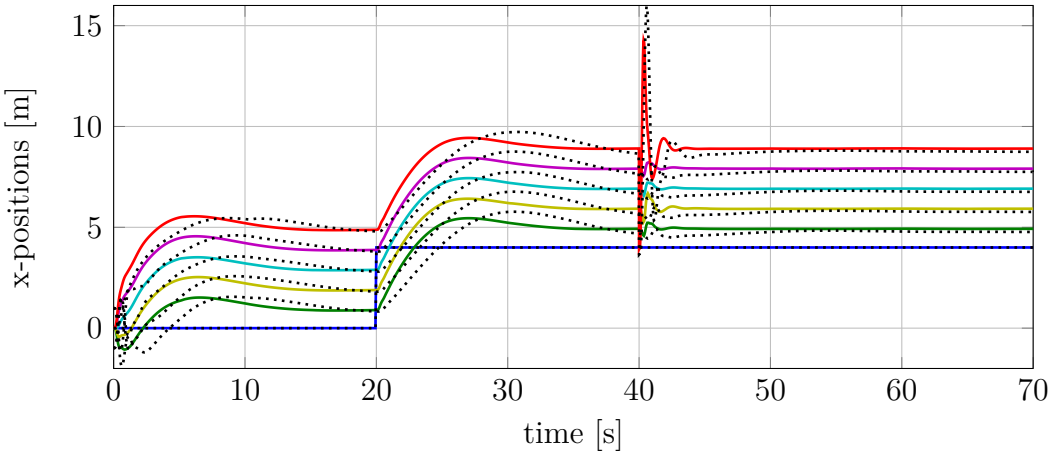
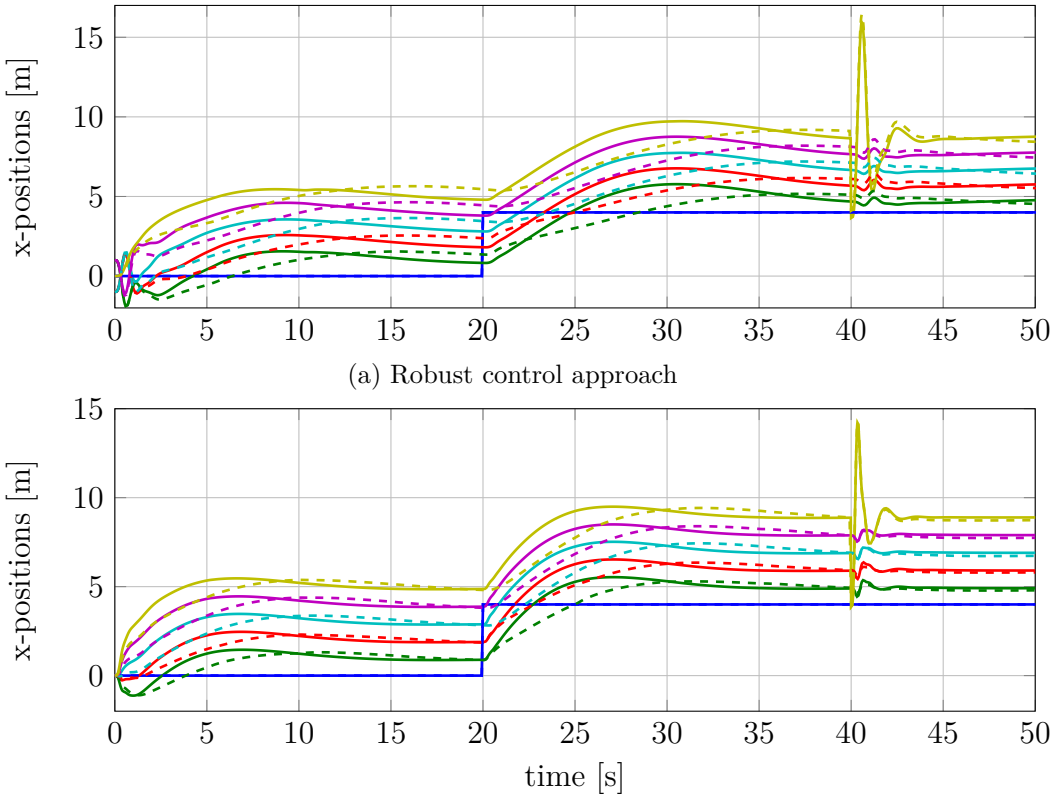


Figure 3.3.4: Positions $y_{i,x}$ for robust control (dotted) and interconnection-scheduled approach (solid), error data exchange, all agents connected to the leader (i.e. $\Psi_{FL} = [1 \cdots 1]^T$)

Simulation Results In the following, the results of the comparative simulation study are presented. Figure 3.3.4 shows the simulated agent positions for both robust control and interconnection-scheduled control (ISC). The leader position is plotted as dark blue solid line. It is observed that for the ISC-designed controller both formation establishment as well as the reaction to the leader movement is performed significantly faster. At the same time, this controller achieves a smaller overshoot of both leader tracking and disturbance rejection in comparison with the robust controller. This observation, which indicates superior performance of the ISC approach, coincides with the expectation of this approach to be less conservative.



(a) Robust control approach

(b) Interconnection-scheduled control approach

Figure 3.3.5: Positions $y_{i,x}$ with error data exchange;
solid: $\Psi_{FL} = [1 \cdots 1]^T$, dashed: $\Psi_{FL} = [1 \quad 1 \quad 0 \quad 0 \quad 0]^T$

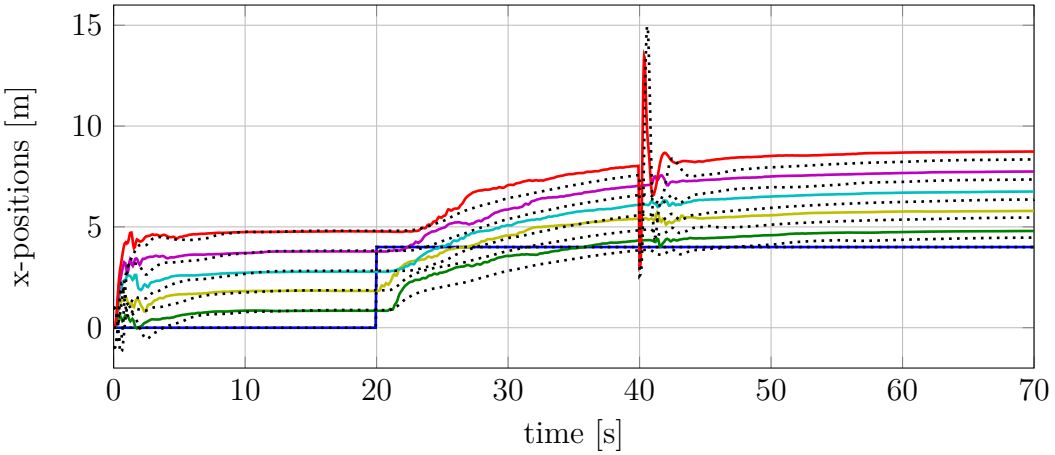
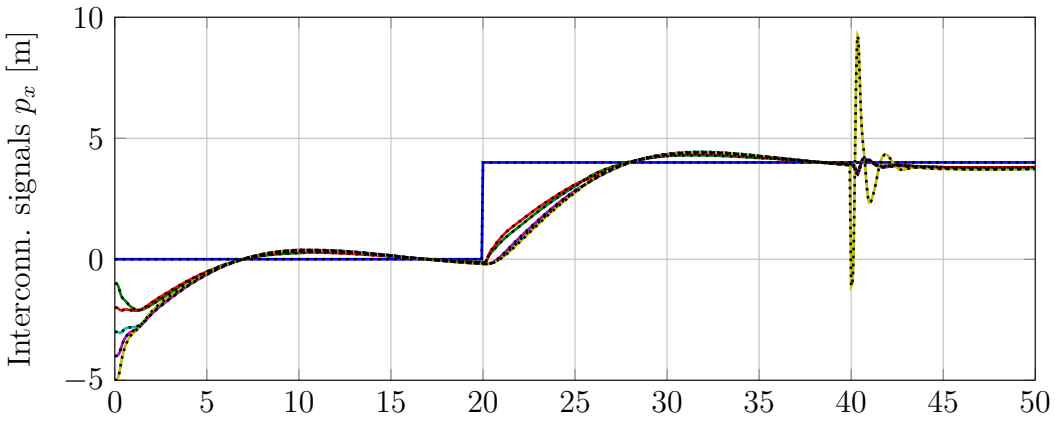


Figure 3.3.6: Positions $y_{i,x}$ for robust control (dotted) and interconnection-scheduled control (solid) for delayed communication; error data exchange, $\Psi_{FL} = [1 \quad 1 \quad 0 \quad 0 \quad 0]^T$

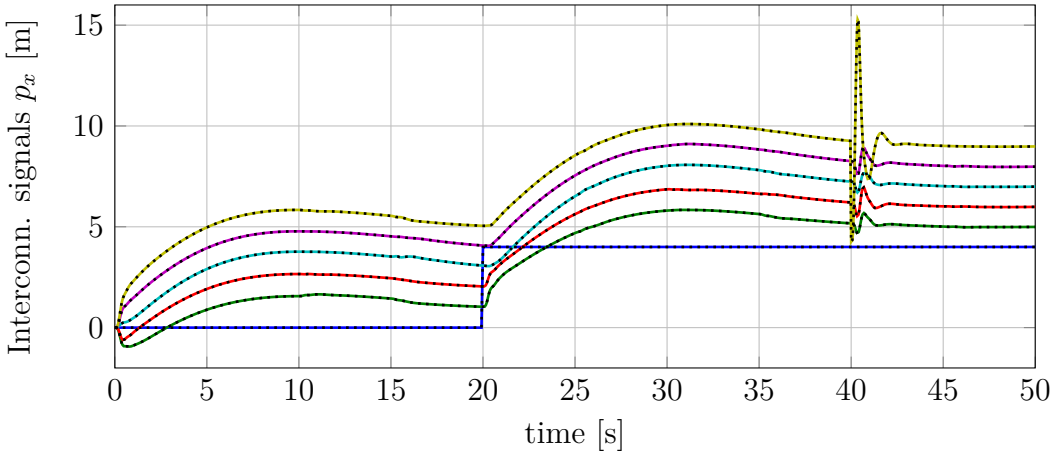
If some followers do not directly receive leader data, successful tracking requires to propagate the leader information through the group of followers. This fact can be expected to have a damping effect on the tracking behavior of the MAS. Exactly this observation can be made in Fig. 3.3.5: Limiting direct leader connection to agents 1 and 2 leads to a slow-down of the tracking behavior, meaning an increase of rise time and settling time together with a reduction of the overshoot. The formation accuracy is however maintained.

These observations are made in a similar manner for both robust control (Fig. 3.3.5a) and ISC (Fig. 3.3.5b) approach.

A potential drawback of the ISC approach is the lack of stability guarantee for delayed communication, which in turn promises less conservatism when delays are insignificant. As delays are likely to appear in practical applications, it is an interesting question how the ISC-designed controller performs in the presence of delays. Here we consider a scenario with time-varying random delays (updated every 0.1s), which take gaussian-distributed values with 1s mean value and a standard deviation of 0.5s. Fig. 3.3.6 shows results of both designs for this delay scenario. Although the ISC synthesis does not guarantee stability, in the presented example a stabilizing controller is obtained and even outperforms the robust controller: For both designs the tracking performance is significantly affected by the delay. The robust controller needs a significantly longer time to reach the desired position and shows a larger overshoot in the disturbance response. Even for larger delays no instability of the ISC setup was observed in this study, which indicates a good robustness against delays in spite of the lack of formal guarantee.



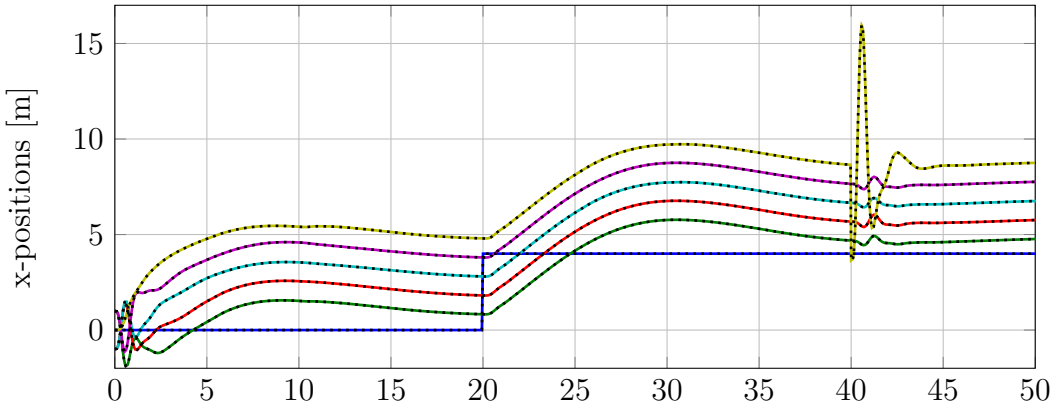
(a) Error data exchange $p_i^P = \eta_i$ (ii)



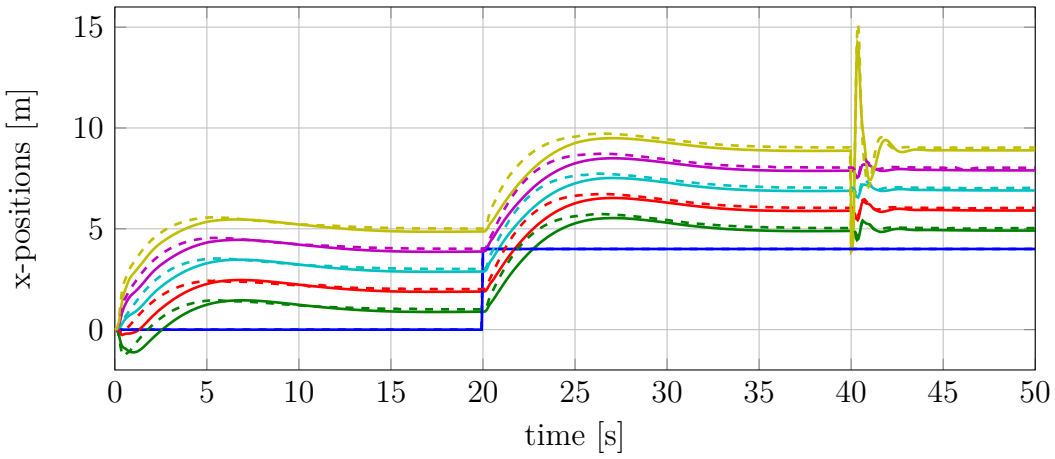
(b) Output data exchange $p_i^P = y_i$ (i)

Figure 3.3.7: x-Coordinate of the coordination signals p_i^K of the controller (solid colored) and p_i^P of the plant (black dotted), ISC approach, $\Psi_{FL} = [1 \quad 1 \quad 0 \quad 0 \quad 0]^T$

Besides not considering delays, the control architecture for which the ISC approach is used can be expected to contribute to a reduced conservatism: The direct interaction of the controllers provides an additional degree of freedom in the search for the optimal controller. Regarding the actual composition of p^K, an interesting observation can be made in Fig. 3.3.7: For both versions (i) and (ii) of choosing the plant coordination signal p^P, the actually transmitted data p^K show no visible difference to p^P. From this fact one can infer that, at least for the setup used in this work, the ISC synthesis chooses the interaction signal as a copy of the plant coordination signal defined in the generalized plant. By doing so, it imitates the robust control scheme instead of exploiting the additional degree of freedom. This effect is referred to as "Feed-Through Effect" in this work and is further examined in the following Section 3.3.4.



(a) Robust control approach

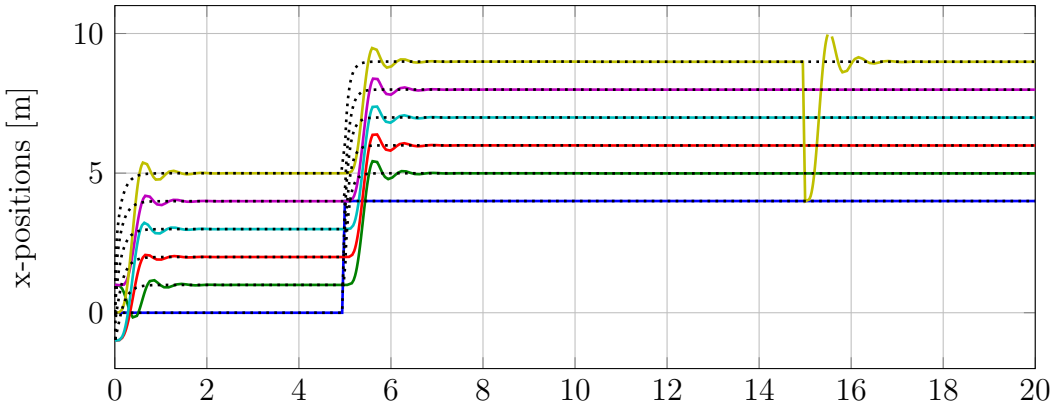


(b) Interconnection-scheduled control approach

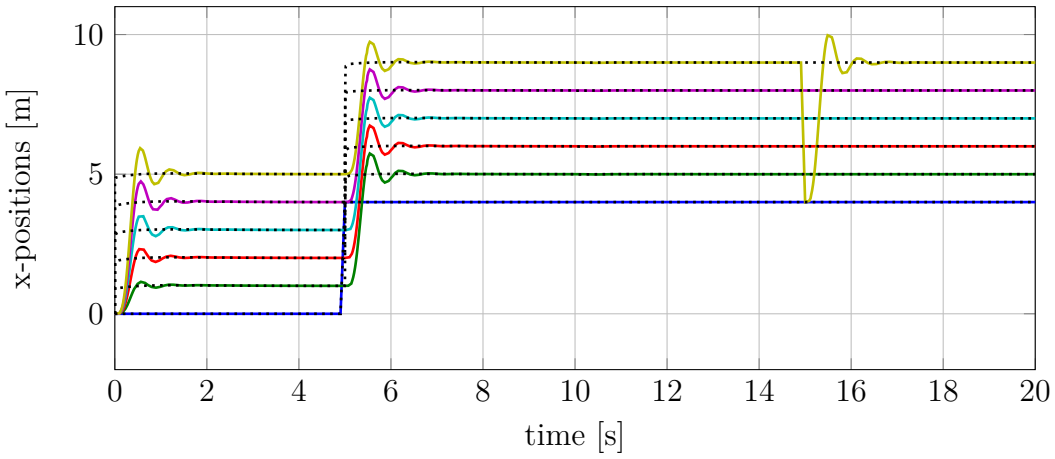
Figure 3.3.8: Positions $y_{i,x}$ for (i) output data exchange ($p_i^P = y_i$, dashed/dotted) and (ii) error data exchange ($p_i^P = \eta_i$, solid); $\Psi_{FL} = [1 \cdots 1]^T$

An important choice to be made in the problem formulation is on the signal to be used as plant coordination signal p_i^P in the sense of Fig.3.2.3 and 3.2.6. The results shown so far (except Fig. 3.3.7b) have been achieved using $p_i = \eta_i$. Another reasonable choice is $p_i^P = y_i$, results for which are shown in Fig. 3.3.8. As easily noticeable, for the robust

control approach both designs lead to equal results. This is plausible, because the two choices only differ in the dependence of p_i on the input w_i. As the robust control approach does not consider the cross-relation from w_i to p_i (only T_{zw} and T_{pq} are handled separately in Prob. 3.1.2), this difference has no effect. In contrast, the ISC design handles the system T as a whole and thus also depends on the cross-relation T_{pw}. Accordingly, in Fig. 3.3.8b we can notice a visible difference as for $p_i = y_i$ a slightly faster tracking behavior is observed.



(a) Robust control approach



(b) Interconnection-scheduled control approach

Figure 3.3.9: MAS response $y_{i,x}$ (solid) and estimated reference $\hat{r}_i$ (dotted) for consensus-based control, $\Psi_{FL} = [1 \quad 1 \quad 0 \quad 0 \quad 0]^T$

Results for the consensus-based formation control scheme are presented in Fig. 3.3.9, where the information flow filter has been designed using the robust control approach in one case and using the interconnection-scheduled approach in the other. Tuning values were obtained using the same genetic algorithm as for designing the cooperative controllers.

As we can observe comparing Fig. 3.3.9a and 3.3.9b, the ISC approach achieves a significantly faster consensus in both formation establishment and leader tracking. With both approaches a good formation accuracy is achieved and no noticeable overshoot of the esti-

mated reference appears. According to the results presented in Bartels and Werner [2014] with respect to the robust IFF design, the consensus-based control scheme achieves a significantly higher tracking performance compared to the cooperative scheme. The results of this work also confirm this observation for the interconnection-scheduled IFF design. Due to the architecture of the consensus-based scheme, it is expected regardless of the IFF design that a disturbance acting on a single agent will not lead to any reaction of the other agents. This is confirmed in the shown results.

3.3.4 A "Feed-Through" Effect in Controller Interaction

The previously presented study reveals an interesting observation regarding the controller interaction signals generated by the ISC controllers: The results of this study (see Fig. 3.3.7) show that for both tested problem formulations of the ISC approach, the controller interaction signal p_i^K is identical to the virtual plant interaction signal p_i^P. While, in contrast to the robust approach, here the controller matrices defining the interaction output are variables to be chosen by solving the synthesis problem, obviously a result is obtained which imitates the architecture of the robust approach, as the controllers *feed through* the interaction signal of the plant. The resulting numerical values of the controller matrices show that the signals p_i^P and p_i^K do not just look similar, but for this particular controller always are exactly identical: The synthesis problem is defined according to Theorem 3.1.4 using the generalized plant from Fig. 3.2.7. Solving this particular problem yields a controller of the form (2.1.3) with the following structure:

$$\begin{pmatrix} \dot{x}_i^K \\ \hline u_i \\ p_i^K \end{pmatrix} = \left[\begin{array}{c|cc} A^K & B_v^K & B_q^K \\ \hline C_u^K & D_{uv}^K & D_{uq}^K \\ 0 & [I \quad 0] & 0 \end{array}\right] \begin{pmatrix} x_i^K \\ \hline v_i \\ q_i^K \end{pmatrix}, \quad v_i = \begin{pmatrix} \eta_i \\ \phi_i \end{pmatrix} \tag{3.3.12}$$

This clearly shows that the plant interaction signal p_i^P, which is part of the measured output v_i, is directly transmitted to the neighbor controllers. As this observation contradicts the expectation of an additional degree of freedom to be provided by the ISC approach, it is worth to further examine this effect.

In the presented case study the described effect was observed for the synthesis of distributed controllers for multi-agent systems. However, the underlying synthesis technique originally is a gain-scheduling controller synthesis technique for LPV systems. This fact raises the question whether the same effect can be observed also for "ordinary" LPV gain-scheduling control problems, i.e. without a distributed control interpretation. In the following, the effect is indeed reproduced for a SISO LPV example.

SISO Example The subject of this example is the second order non-linear SISO system described by the differential equation

$$m\ddot{y} = -b\dot{y}^2 + u.$$

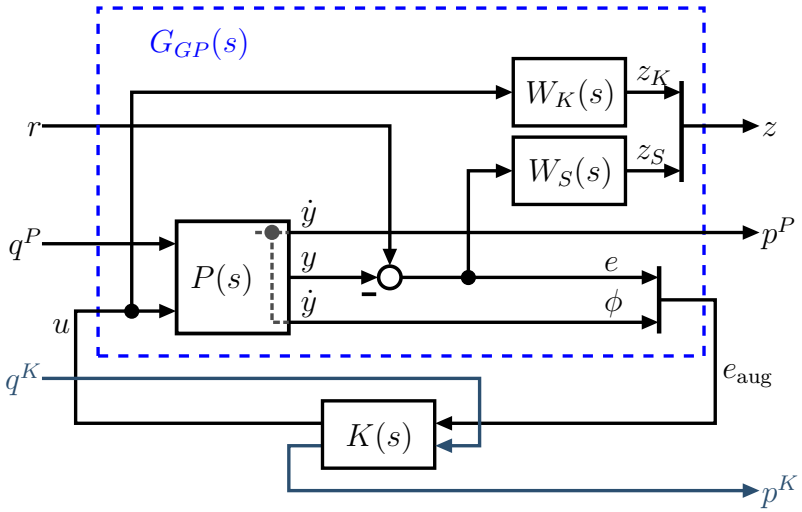


Figure 3.3.10: Generalized Plant of the gain-scheduled LPV control problem

This model describes the dynamics of a body with mass m, which is moved by input force u and is subject to aerodynamic drag $F_{\text{drag}} = -bv^2$ depending quadratically on the velocity. For this system a quasi-LPV model can be constructed with the state vector $x = [y \quad \dot{y}]^T$ describing position and velocity of the mass and the velocity $\dot{y}$ as scheduling parameter. An LFT representation of this system reads

$$\begin{bmatrix} \dot{x} \\ \hline p \\ \hline y_{\text{aug}} \end{bmatrix} = \left[\begin{array}{cc:c:c} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{b}{m} & \frac{1}{m} \\ \hdashline 0 & 1 & 0 & 0 \\ \hdashline 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right] \begin{bmatrix} x \\ \hline q \\ \hline u \end{bmatrix}, \tag{3.3.13}$$

$$q = \Delta p = \dot{y}^2; \quad \Delta = \dot{y}. \tag{3.3.14}$$

The examined problem is the design of a two-degree-of-freedom position controller for this system, to which the position error and one of the system outputs, either (i) the velocity $\dot{y}$ or (ii) the position y, is fed. Figure 3.3.10 shows the corresponding generalized plant for case (i).

To this control problem we apply the synthesis technique from Scherer [2001], which is the basis of the distributed controller synthesis technique by Hoffmann and Werner [2017]. This yields a gain-scheduled controller of the form

$$\begin{bmatrix} \dot{x}^K \\ u \\ p^K \end{bmatrix} = \begin{bmatrix} A^K & B_v^K & B_q^K \\ C_u^K & D_{uv}^K & D_{uq}^K \\ C_p^K & D_{pv}^K & D_{pq}^K \end{bmatrix} \begin{bmatrix} x^K \\ e_{\text{aug}} \\ q^K \end{bmatrix}, \tag{3.3.15}$$

$$q^K = \Delta_K p^K. \tag{3.3.16}$$

The multipliers computed as intermediate results have the same general form (3.1.19a) as in the distributed control problem. Here we consider both (a) the *D/G-scalings* approach from Dettori and Scherer [2001] imposing the structural constraints (3.1.20) and (b) full-block multipliers. For the D/G-scalings case, the system block matrix of the controller (3.3.15) is computed as

$$(i)\ \phi = \dot{y}: \left[\begin{array}{rrrr|rr|r} 55.36 & -7.64 & 82.81 & -728.20 & -1954.37 & 2159.20 & 66.04 \\ -31.29 & 0.22 & -28.52 & -105.43 & 2.90 & 94.66 & -2.01 \\ -1723.48 & 56.14 & -1304.29 & -103.83 & 1877.87 & -934.41 & -60.33 \\ 26.53 & -20.40 & -84.28 & -2835.03 & -63.74 & 3636.48 & 2.39 \\ \hline -3.62 & 0.14 & -0.58 & 0.68 & 4.03 & -4.10 & -0.13 \\ \hline -0.00 & -0.00 & -0.00 & -0.00 & -0.00 & -1.00 & 0.00 \end{array}\right]$$

$$(ii)\ \phi = y: \left[\begin{array}{rrrr|rr|r} -913.97 & -67760.10 & 296.79 & 16922.86 & 2333.95 & 840223.01 & -97.54 \\ -772.02 & -57519.62 & 177.84 & 14337.07 & -144.12 & 706300.87 & 3.68 \\ -114.46 & -4585.45 & -124.70 & 1755.53 & -135.86 & 155895.46 & 4.59 \\ 54.31 & 3958.51 & -10.46 & -1008.75 & -52.90 & -50916.11 & 0.47 \\ \hline 0.22 & 8.73 & -1.70 & -3.30 & 0.29 & -299.84 & -0.02 \\ \hline 0.00 & -0.03 & 0.00 & 0.25 & 0.02 & -0.42 & 0.01 \end{array}\right]$$

In the highlighted rows the same observation is made as in the previously presented study on distributed control: For choosing the velocity as the second feedback channel, the controller exactly reproduces the plant LFT loop signal, which in this case is the velocity $\dot{y}$. Although the interpretations are different, both synthesis problems have in common that an LFT form plant is considered which feeds the same signal both to the LFT loop and to the controller (here $\dot{y}$ or η_i in the distributed ISC problem). Interestingly, for feeding y instead of $\dot{y}$ to the controller of the SISO LPV control problem, the described feed-through effect does not appear (see above the row highlighted in blue).

The loop output signals p^P and p^K for the SISO LPV example are plotted in Figure 3.3.11 for all four examined cases. The feed-through effect is clearly observed for the case of the velocity fed to the controller in combination with multipliers restricted to D/G structure. For all other cases p^P and p^K show visible differences. This observation indicates that the feed-through effect is a general property of D/G-scalings for control problems in which the controller has access to the plant loop output signal p^P.

Preliminary results of this chapter have been previously published in Bartels and Werner [2014] and Bartels and Werner [2016].

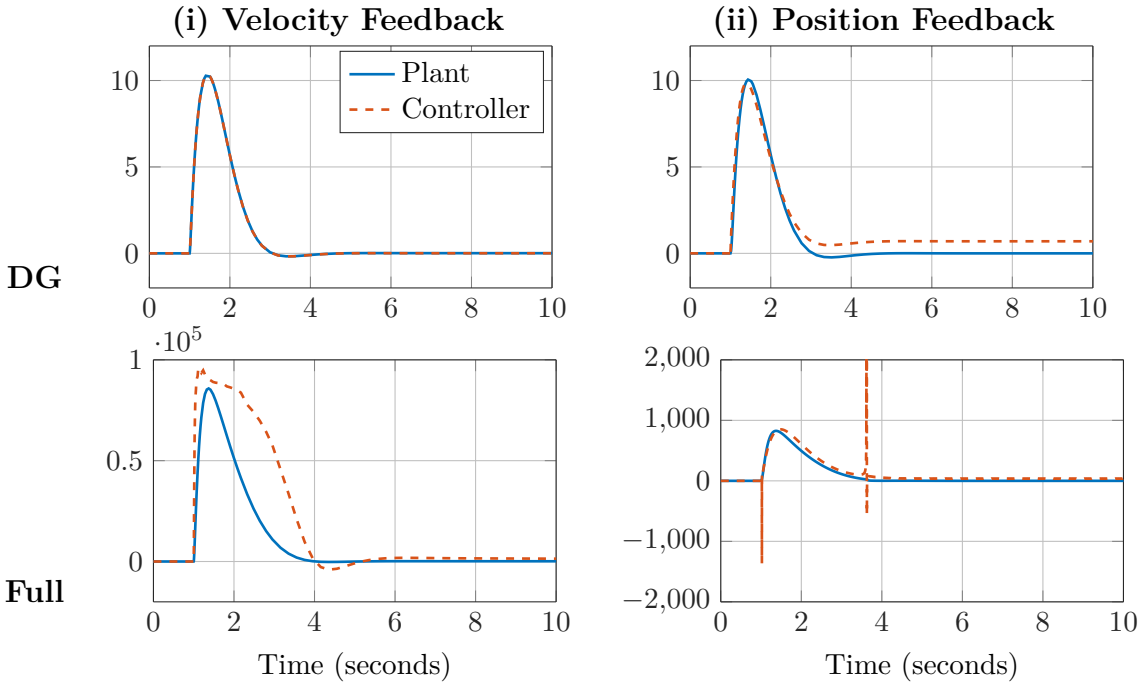


Figure 3.3.11: Loop output signals p of the 2-degree-of-freedom LPV gain-scheduling example

Chapter 4

Descriptor-Based Design of Distributed Controllers

As one of the contributions of this thesis, in this chapter a descriptor representation for multi-agent systems is introduced and a distributed controller design approach for systems in this representation is proposed. The initial idea behind this approach is based on Wollnack [2016] and consists of using the descriptor system framework to model both the local dynamics of the agents as well as the interaction within one descriptor model. This formulation is chosen in a way such that the resulting descriptor model affinely depends on the interaction matrix, which is considered as scheduling parameter. The benefit of this approach is twofold: The fact that a regular state space model can be expressed by infinitely many equivalent descriptor systems provides additional degrees of freedom to the synthesis problem. This fact promises a reduction of conservatism, while the affine formulation avoids the necessity to impose further constraints on the variables such as inherent in the Full-Block $\mathcal{S}$-Procedure (FBSP). Secondly, descriptor models describe a wider class of systems than state space models and for many systems provide a more natural way of modeling [Xia et al., 2009; Hill and Mareels, 1990; Sastry and Desoer, 1981]. This extends the proposed method to further classes of agent models which cannot be handled by existing approaches based on regular state space models.

This fact also motivated research on the consensus of multiple descriptor systems. In Yang and Liu [2012, 2014] a stabilizing output feedback controller is designed for fixed topologies. Consensus in case of switching topologies is handled in Xi et al. [2014], the case of delays is examined in Xi et al. [2012]. A state feedback approach to distributed formation control for swarms of descriptor systems is proposed in Meng et al. [2016] using Jordan decomposition of the interaction matrix. An output feedback approach for heterogeneous descriptor multi-agent systems is given in Ma et al. [2016].

This work is inspired by Masubuchi et al. [2003] and Masubuchi et al. [2004], where an LMI-based design approach for gain-scheduled control of descriptor LPV systems is proposed. The controller is obtained as descriptor system itself and for affinely parameter-dependent plants it inherits the affine dependency. Using this method we propose a new approach to design a distributed $\mathcal{L}_2$-optimal controller of the same architecture as in

Hoffmann and Werner [2017]. It provides scalability and robustness against topology changes.

In this work, the framework of descriptor systems is used to describe multi-agent systems in accordance with the formalism introduced in Section 2.1.1.

4.1 Descriptor Representation of Multi-Agent Systems

For the descriptor representation the framework of multi-agent systems introduced in Section 2.1.1 is extended by considering agents described by descriptor models: Descriptor models corresponding to plant model (2.1.2) and controller (2.1.3) are considered as

$$G: \left(\begin{array}{c} E\dot{x}_i \\ \hdashline z_i \\ p_i^P \\ v_i \end{array}\right) = \left[\begin{array}{c:ccc} A & B_w & B_q & B_u \\ \hdashline C_z & D_{zw} & D_{zq} & D_{zu} \\ C_p & D_{pw} & D_{pq} & D_{pu} \\ C_v & D_{vw} & D_{vq} & D_{vu} \end{array}\right] \left(\begin{array}{c} x_i \\ \hdashline w_i \\ q_i^P \\ u_i \end{array}\right), \tag{4.1.1}$$

$$K: \left(\begin{array}{c} E^K\dot{x}_i^K \\ \hdashline u_i \\ p_i^K \end{array}\right) = \left[\begin{array}{c:cc} A^K & B_v^K & B_q^K \\ \hdashline C_u^K & D_{uv}^K & D_{uq}^K \\ C_p^K & D_{pv}^K & D_{pq}^K \end{array}\right] \left(\begin{array}{c} x_i^K \\ \hdashline v_i \\ q_i^K \end{array}\right) \tag{4.1.2}$$

with signals $x_i \in \mathbb{R}^n$; $x_i^K \in \mathbb{R}^{n_t}$; $p_i^P, q_i^P \in \mathbb{R}^m$; $p_i^K, q_i^K \in \mathbb{R}^{m_K}$; $u_i \in \mathbb{R}^h$; $v_i \in \mathbb{R}^l$; $w_i \in \mathbb{R}^p$ and $z_i \in \mathbb{R}^q$. For the case of a singular agent model, i.e. E is singular, we assume the structure $E = \operatorname{diag}(E_t, 0)$ with a regular block E_t in $\mathbb{R}^{n_t \times n_t}$. This structural assumption does not restrict the set of admissible systems, as according to Bender and Laub [1985]; Bara [2010] every descriptor system with an arbitrary matrix E can be transformed into this structure using a singular value decomposition of E. The first n_t states in x_i are referred to here as *temporal states*, as they correspond to the differential part of the model, while the remaining states express algebraic constraints. The controller construction technique presented in this work yields a controller with a regular left factor E^K of the size of E_t.

Introducing the augmented state vector $\xi = \operatorname{vcat}(x, q^P, p^P)$ of size Nn_D with $n_D = n+2m$, this system can be reformulated as the following descriptor model:

$$\begin{aligned} \hat{\mathcal{E}}\dot{\xi} &= \mathscr{A}(\Psi)\xi + \hat{\mathcal{B}}_w w + \hat{\mathcal{B}}_u u \\ z &= \hat{\mathcal{C}}_z \xi + \hat{D}_{zw} w + \hat{D}_{zu} u \\ v &= \hat{\mathcal{C}}_v \xi + \hat{D}_{vw} w + \hat{D}_{vu} u \end{aligned} \tag{4.1.3}$$

with the augmented system matrices

$$\hat{\mathcal{E}} = \begin{bmatrix} \hat{E} & & \\ & 0 & \\ & & 0 \end{bmatrix}, \quad \mathscr{A}(\Psi) = \begin{bmatrix} \hat{A} & \hat{B}_q & 0 \\ \hat{C}_p & \hat{D}_{pq} & -I \\ 0 & -I & \Psi_{(m)} \end{bmatrix}, \tag{4.1.4}$$

$$\hat{\mathcal{B}}_w = \begin{bmatrix} \hat{B}_w \\ \hat{D}_{pw} \\ 0 \end{bmatrix}, \quad \hat{\mathcal{B}}_u = \begin{bmatrix} \hat{B}_u \\ \hat{D}_{pu} \\ 0 \end{bmatrix}, \quad \begin{aligned} \hat{\mathcal{C}}_z &= \begin{bmatrix} \hat{C}_z & \hat{D}_{zq} & 0 \end{bmatrix}, \\ \hat{\mathcal{C}}_v &= \begin{bmatrix} \hat{C}_v & \hat{D}_{vq} & 0 \end{bmatrix}. \end{aligned}$$

The augmented state equation in (4.1.3) is a composition of the state equation from the local state space model (4.1.1) describing the internal dynamics of the agent, and the algebraic relation $q^P = \Psi_{(m)} p^P$ (2.1.7b) describing the interaction of the agents. In the following, the calligraphic font (like $\mathcal{A}$, $\mathcal{B}$) will be used for the system matrices of the descriptor representation, while the normal font is used for the matrices of the local nominal model.

The descriptor-based controller synthesis technique presented in this thesis yields a controller inheriting the form of representation (4.1.3):

$$\begin{aligned} \hat{\mathcal{E}}^K \dot{\xi}^K &= \mathscr{A}^K(\Psi)\xi^K + \hat{\mathcal{B}}^K v \\ u &= \hat{\mathcal{C}}^K \xi^K + \hat{\mathcal{D}}^K v, \end{aligned} \tag{4.1.5}$$

where the matrix $\mathscr{A}^K(\Psi) \in \mathbb{R}^{n_{DK} \times n_{DK}}$ affinely depends on the interaction matrix Ψ. The closed loop consisting of this controller and the plant (4.1.3) then reads

$$\begin{aligned} \hat{\mathcal{E}}^{cl} \dot{\xi}^{cl} &= \mathscr{A}^{cl}(\Psi)\xi^{cl} + \hat{\mathcal{B}}^{cl} w \\ z &= \hat{\mathcal{C}}^{cl} \xi^{cl} + \hat{\mathcal{D}}^{cl} v, \end{aligned} \tag{4.1.6}$$

where $\xi^{cl} = \operatorname{vcat}(\xi, \xi^K)$ and $\hat{\mathcal{E}}^{cl} = \operatorname{diag}(\hat{\mathcal{E}}, \hat{\mathcal{E}}^K)$.

Assuming that Ψ is diagonalizable, the system (4.1.3) can be decomposed into a family of *modal subsystems* parameterized by the eigenvalues λ of Ψ:

$$\begin{aligned} \mathcal{E}\dot{\xi}_i &= \mathcal{A}(\lambda)\xi_i + \mathcal{B}_w w_i + \mathcal{B}_u u_i \\ z_i &= \mathcal{C}_z \xi_i + D_{zw} w + D_{zu} u_i \\ v_i &= \mathcal{C}_v \xi_i + D_{vw} w + D_{vu} u_i \end{aligned} \tag{4.1.7}$$

with

$$\begin{aligned}
\mathcal{E} &= \begin{bmatrix} E & & \\ & 0 & \\ & & 0 \end{bmatrix}, \quad \mathcal{A}(\lambda) = \begin{bmatrix} A & B_q & 0 \\ C_p & D_{pq} & -I \\ 0 & -I & \lambda I \end{bmatrix}, \\
\mathcal{B}_w &= \begin{bmatrix} B_w \\ D_{pw} \\ 0 \end{bmatrix}, \quad \mathcal{B}_u = \begin{bmatrix} B_u \\ D_{pu} \\ 0 \end{bmatrix}, \quad \mathcal{C}_z = \begin{bmatrix} C_z & D_{zq} & 0 \end{bmatrix}, \\
&\qquad\qquad\qquad\qquad\quad \mathcal{C}_v = \begin{bmatrix} C_v & D_{vq} & 0 \end{bmatrix}.
\end{aligned} \tag{4.1.8}$$

Accordingly, in this case the distributed controller (4.1.5) can as well be decomposed into a family of affinely parameter-dependent controllers

$$\begin{aligned}
\mathcal{E}^K \dot{\xi}_i^K &= \mathcal{A}^K(\lambda)\xi_i^K + \mathcal{B}^K v_i \\
u_i &= \mathcal{C}^K \xi_i^K + \mathcal{D}^K v_i
\end{aligned} \tag{4.1.9}$$

4.2 Controller Design for Descriptor Multi-Agent Systems

In this section we will provide a method to design an $\mathcal{L}_2$-optimal output feedback controller according to (4.1.9) for multi-agent systems in descriptor representation (4.1.3). This result is based on the controller design method presented in Masubuchi et al. [2003] and Masubuchi et al. [2004] for single LPV descriptor systems.

The approach presented here is based on the assumption that the interaction matrix Ψ is unitarily diagonalizable, i.e. the existence of a unitary matrix Z fulfilling

$$\Psi = Z\Lambda Z^T \text{ with } \Lambda \text{ diagonal}, \tag{4.2.1}$$

for which the diagonal elements λ of Λ are the eigenvalues of Ψ. The set $\mathbf{\Lambda}$ is defined as the set of all possible eigenvalues of admissible interaction matrices Ψ.

4.2.1 Stabilizing Controller Design

As first step, a synthesis problem for a stabilizing controller is derived. We consider the system (4.1.3), but the performance output z is neglected and the reference input w is assumed identical to zero. This yields the system

$$\begin{aligned}
\hat{\mathcal{E}}\dot{\xi} &= \mathscr{A}(\Psi)\xi + \hat{\mathcal{B}}_u u \\
v &= \hat{\mathcal{C}}_v \xi + \hat{D}_{vu} u,
\end{aligned} \tag{4.2.2}$$

for which the closed loop system using the controller (4.1.5) is obtained as

$$\hat{\mathcal{E}}^{cl}\dot{\xi}^{cl} = \mathscr{A}^{cl}(\Psi)\xi^{cl}. \tag{4.2.3}$$

Theorem 4.2.1. *Consider the multi-agent system (4.2.2) with a possibly time-varying interaction matrix $\Psi \in \mathbf{\Psi}$ that is unitarily diagonalizable at all times. There exists a distributed controller (4.1.5) that stabilizes the closed loop system (4.2.3) if there exist $X, Y, H \in \mathbb{R}^{n_D \times n_D}$, $F \in \mathbb{R}^{n_D \times h}$ and $G \in \mathbb{R}^{l \times n_D}$ with the structure*

$$X = \begin{bmatrix} X_{11} & 0 \\ X_{21} & X_{22} \end{bmatrix}, \qquad Y = \begin{bmatrix} Y_{11} & Y_{12} \\ 0 & Y_{22} \end{bmatrix}, \tag{4.2.4}$$
$$X_{11} = X_{11}^T \in \mathbb{R}^{n_t \times n_t}, \qquad Y_{11} = Y_{11}^T \in \mathbb{R}^{n_t \times n_t}$$

satisfying

$$\begin{bmatrix} \mathcal{E} & \\ & \mathcal{E}^T \end{bmatrix} \begin{bmatrix} Y^T & I \\ I & X \end{bmatrix} = P = P^T \geq 0 \tag{4.2.5}$$
$$Q + Q^T < 0 \quad \forall \lambda \in \mathbf{\Lambda} \tag{4.2.6}$$
$$\text{with } Q = \begin{bmatrix} \mathcal{A}(\lambda) Y^T + \mathcal{B}_u F^T & \mathcal{A}(\lambda) \\ H^T & X^T \mathcal{A}(\lambda) + G^T \mathcal{C}_v \end{bmatrix}. \tag{4.2.7}$$

Proof: The proof of this theorem combines the controller synthesis approach for descriptor systems proposed in Masubuchi et al. [2003] with the decomposition approach used in Hoffmann and Werner [2017]. As starting point we consider the global closed loop system (4.2.3) and derive a stability criterion on global level. Subsequently, this is decomposed to obtain the criterion on the level of a single subsystem.

According to Masubuchi et al. [2003] the stability of the global closed loop system is provided by the existence of a Lyapunov matrix $\mathscr{Y}_{cl} \in \mathbb{R}^{N(n_D+n_{DK}) \times N(n_D+n_{DK})}$ satisfying

$$\hat{\mathcal{E}}_{cl} \mathscr{Y}_{cl}^T = \mathscr{Y}_{cl} \hat{\mathcal{E}}_{cl}^T \geq 0 \tag{4.2.8}$$
$$\mathscr{A}_{cl}(\Psi) \mathscr{Y}_{cl}^T + \mathscr{Y}_{cl} \mathscr{A}_{cl}^T(\Psi) < 0 \quad \forall \Psi \in \mathbf{\Psi}. \tag{4.2.9}$$

As $\mathscr{A}_{cl}$ contains the unknown controller matrices, the inequality (4.2.9) is bi-linear in the variables. To obtain an LMI problem, Masubuchi et al. [2003] propose a change of variables, which transfers the problem (4.2.8, 4.2.9) into a linear problem of the form (4.2.5, 4.2.6) in the variables $\mathscr{X}, \mathscr{Y}, \mathscr{F}, \mathscr{G}, \mathscr{H}$. With the Lyapunov matrix being partitioned into $Nn_D \times Nn_D$ blocks as

$$\mathscr{Y}_{cl} = \begin{bmatrix} \mathscr{Y}_{11} & \mathscr{Y}_{12} \\ \mathscr{Y}_{21} & \mathscr{Y}_{22} \end{bmatrix} \tag{4.2.10}$$

and $\mathscr{S} = \mathscr{Y}_{12} \mathscr{Y}_{22}^{-1} \mathscr{Y}_{21}$, the new variables are chosen as

$$\begin{aligned} &\mathscr{X} = (\mathscr{Y}_{11} - \mathscr{S})^{-T}, \quad \mathscr{Y} = \mathscr{Y}_{11}, \\ &\mathscr{F} = \mathscr{S} \cdot (\hat{\mathcal{C}}^K)^T, \quad \mathscr{G} = -(\hat{\mathcal{B}}^K)^T \mathscr{X}, \\ &\mathscr{H} = \left(\mathscr{Y} (\mathscr{A}(\Psi) - \hat{\mathcal{B}}^K \hat{\mathcal{C}}_v)^T + \mathscr{S} (\hat{\mathcal{B}}_u \hat{\mathcal{C}}^K - \mathscr{A}^K(\Psi))^T \right) \mathscr{X}. \end{aligned} \tag{4.2.11}$$

For further details the reader is referred to Masubuchi et al. [2003]. With this change of variables (4.2.11), the corresponding controller and $\mathscr{Y}_{cl}$ satisfy (4.2.8, 4.2.9) if $\mathscr{X}, \mathscr{Y}, \mathscr{F}, \mathscr{G}, \mathscr{H}$ fulfill

$$\begin{bmatrix} \hat{\mathcal{E}} & \\ & \hat{\mathcal{E}}^T \end{bmatrix} \begin{bmatrix} \mathscr{Y}^T & I \\ I & \mathscr{X} \end{bmatrix} = \mathscr{P} = \mathscr{P}^T \geq 0, \tag{4.2.12}$$

$$\mathcal{Q} + \mathcal{Q}^T < 0 \quad \forall \Psi \in \mathbf{\Psi} \tag{4.2.13}$$

$$\text{with } \mathcal{Q} = \begin{bmatrix} \mathscr{A}(\Psi)\mathscr{Y}^T + \hat{\mathcal{B}}_u \mathscr{F}^T & \mathscr{A}(\Psi) \\ \mathscr{H}^T & \mathscr{X}^T \mathscr{A}(\Psi) + \mathscr{G}^T \hat{\mathcal{C}}_v \end{bmatrix}.$$

In order to decompose this LMI problem, we constrain the variables to have a block-diagonal structure: For $\mathscr{X}$, $\mathscr{Y}$ and $\mathscr{H}$ we impose

$$\mathscr{X} = \begin{bmatrix} \hat{X}_{11} & \hat{X}_{12} & \hat{X}_{13} \\ \hat{X}_{21} & \hat{X}_{22} & \hat{X}_{23} \\ \hat{X}_{31} & \hat{X}_{32} & \hat{X}_{33} \end{bmatrix} \tag{4.2.14}$$

with partitioning $\begin{bmatrix} Nn & Nm & Nm \end{bmatrix}$ and each sub-block being repeated block-diagonal with N repetitions of the corresponding subsystem-level variables. Likewise, we partition $\mathscr{F}$ and $\mathscr{G}$ into block-diagonal sub-blocks as

$$\mathscr{F} = \begin{bmatrix} \hat{F}_1^T & \hat{F}_2^T & \hat{F}_3^T \end{bmatrix}^T, \quad \mathscr{G} = \begin{bmatrix} \hat{G}_1 & \hat{G}_2 & \hat{G}_3 \end{bmatrix}. \tag{4.2.15}$$

With this structure of $\mathscr{X}$ and $\mathscr{Y}$ the coupling condition (4.2.12) already has a structure which allows decomposing it into N identical expressions (4.2.5). The decomposition of the LMI condition (4.2.13) is obtained using the unitarily diagonalizing transformation Z: The inequality (4.2.13) is post-multiplied with the transformation matrix diag $(\mathcal{Z}, \mathcal{Z})$ and pre-multiplied with its inverse, where $\mathcal{Z} = \text{diag}(Z_{(n)}, Z_{(m)}, Z_{(m)})$. The structural constraints on the matrix variables have been chosen such that they commute with this transformation matrix. This yields

$$\tilde{\mathcal{Q}} + \tilde{\mathcal{Q}}^T < 0 \text{ with} \tag{4.2.16}$$

$$\tilde{\mathcal{Q}} = \begin{bmatrix} \mathcal{Z}^T \mathscr{A}(\Psi) \mathcal{Z} \mathscr{Y}^T + \hat{\mathcal{B}}_u \mathscr{F}^T & \mathcal{Z}^T \mathscr{A}(\Psi) \mathcal{Z} \\ \mathscr{H}^T & \mathscr{X}^T \mathcal{Z}^T \mathscr{A}(\Psi) \mathcal{Z} + \mathscr{G}^T \hat{\mathcal{C}}_v \end{bmatrix}$$

From the structure of $\mathscr{A}$ given in (4.1.4) we obtain for the transformed version of $\mathscr{A}$

$$\mathcal{Z}^T \mathscr{A}(\Psi) \mathcal{Z} = \begin{bmatrix} \hat{A} & \hat{B}_q & 0 \\ \hat{C}_p & \hat{D}_{pq} & -I \\ 0 & -I & Z_{(m)}^T \Psi_{(m)} Z_{(m)} \end{bmatrix} \tag{4.2.17}$$

where $Z_{(m)}^T \Psi_{(m)} Z_{(m)}$ can be replaced by the diagonal $\Lambda_{(m)}$. Thus, condition (4.2.13) can be decomposed into N instances of the subsystem-level condition (4.2.6) parameterized by λ. Consequently, we have transformed the global-level LMI problem into a family of parameter-dependent LMI problems in common variables X, Y, F, G, H with respect to the modal subsystem (4.1.7). From a solution to this problem, a controller can be reconstructed by solving the decomposed version of (4.2.11) for the controller matrices yielding

$$\begin{aligned} \mathcal{A}^K &= \left((\mathcal{A}(\lambda) + X^{-T} G^T \mathcal{C}_v) Y^T - X^{-T} H^T + \mathcal{B}_u F^T\right) S^{-T}, \\ \mathcal{B}^K &= -X^{-T} G^T, \quad \mathcal{C}^K = F^T S^{-T}, \quad \mathcal{D}^K = 0, \quad \mathcal{E}^K = \mathcal{E} \\ &\quad \text{with } S = Y - X^{-T}. \end{aligned} \tag{4.2.18}$$

The structural constraints on the subsystem-level Lyapunov matrices given in (4.2.4) result from the symmetry requirement of P in the coupling condition (4.2.5). ■

As the matrix $\mathcal{A}(\lambda)$ of the modal subsystem (4.1.7) is an affine function of λ, the controller coefficient matrix $\mathcal{A}^K(\lambda)$ constructed according to (4.2.18) is an affine function of λ as well. Another fact resulting from $\mathcal{A}(\lambda)$ being an affine function is that the LMI constraint (4.2.6) is convex in the parameter λ. This is an important observation, as it allows us to infer condition (4.2.6) being fulfilled for all λ in an interval $[\underline{\lambda}, \overline{\lambda}]$ from just checking that it is fulfilled for the bounds $\underline{\lambda}$ and $\overline{\lambda}$. Thus, with this observation we avoid the requirement mentioned in Theorem 4.2.1 of checking the LMI condition for infinitely many values of λ. For interaction matrices Ψ fulfilling (2.1.17), by Gershgorin's circle theorem we obtain the bounds $\lambda \in [-1,\ 1]$.

According to (4.2.11), the variable global-level $\mathscr{H}$ is an affine function of Ψ. From the decomposition, the corresponding local-level variable is thus obtained as affine function $H(\lambda) = H_0 + \lambda H_1$ of λ. However, a parameter-dependency in H can lead to a controller matrix $\mathcal{A}^K(\lambda)$ with parameter dependency in all elements. This would require the number of interaction channels to be equal to the number of states of the descriptor subsystem model (4.1.7) and prohibit using a reduced number. To avoid this, a parameter-independent choice $H = H_0$ is proposed accepting conservatism introduced by this restriction. In the numerical example tested in this work no significant benefit of allowing a parameter-dependent matrix H was observed.

4.2.2 Controller Synthesis for Performance

In Theorem 4.2.1 a synthesis problem on subsystem level is considered, providing a stabilizing controller. It is straightforward to extend this problem to the synthesis of a controller with guaranteed $\mathcal{L}_2$ performance. In Masubuchi et al. [2003] and Masubuchi et al. [2004] the synthesis problem adopted in this work already provides $\mathcal{L}_2$-optimal design. Using this, we can state the following:

Theorem 4.2.2. *Consider the multi-agent system in descriptor representation (4.1.3) with a possibly time-varying interaction matrix $\Psi \in \boldsymbol{\Psi}$ that is unitarily diagonalizable at all times. There exists a distributed controller (4.1.5) providing that the closed loop*

system G^{cl} from (4.1.6) is stable and fulfills $\|G^{cl}\|_{\mathcal{L}_2} < \gamma$, if there exist $X, Y, H \in \mathbb{R}^{n_D \times n_D}$, $F \in \mathbb{R}^{n_D \times h}$ and $G \in \mathbb{R}^{l \times n_D}$ with the structure (4.2.4), which satisfy (4.2.5) and

$$\begin{bmatrix} Q(\lambda) + Q^T(\lambda) & Q_B & Q_C^T \\ Q_B^T & -\gamma I & D_{zw}^T \\ Q_C & D_{zw} & -\gamma I \end{bmatrix} < 0 \quad \forall \lambda \in \mathbf{\Lambda} \tag{4.2.19}$$

with Q from (4.2.7) and

$$Q_B = \begin{bmatrix} \mathcal{B}_w \\ X^T \mathcal{B}_w + G^T D_{vw} \end{bmatrix}, \tag{4.2.20a}$$

$$Q_C = \begin{bmatrix} \mathcal{C}_z Y^T + D_{zu} F^T & \mathcal{C}_z \end{bmatrix} \tag{4.2.20b}$$

Proof: The proof of this theorem is an extension of the proof of Theorem 4.2.1. We apply the optimal synthesis conditions from Masubuchi et al. [2004], which provide that the global system G^{cl} being stable and fulfilling $\|G^{cl}\|_{\mathcal{L}_2} < \gamma$ is implied by the existence of $\mathscr{X}, \mathscr{Y}, \mathscr{F}, \mathscr{G}, \mathscr{H}$ fulfilling (4.2.12) and the extended LMI condition

$$\begin{bmatrix} \mathcal{Q}(\Psi) + \mathcal{Q}^T(\Psi) & \mathcal{Q}_B & \mathcal{Q}_C^T \\ \mathcal{Q}_B^T & -\gamma I & \hat{D}_{zw}^T \\ \mathcal{Q}_C & \hat{D}_{zw} & -\gamma I \end{bmatrix} < 0 \quad \forall \Psi \in \mathbf{\Psi} \tag{4.2.21}$$

with $\mathcal{Q}$ as in (4.2.13) and

$$\mathcal{Q}_B = \begin{bmatrix} \hat{\mathcal{B}}_w \\ \mathscr{X}^T \hat{\mathcal{B}}_w + \mathscr{G}^T \hat{D}_{vw} \end{bmatrix}, \tag{4.2.22a}$$

$$\mathcal{Q}_C = \begin{bmatrix} \hat{\mathcal{C}}_z \mathscr{Y}^T + \hat{D}_{zu} \mathscr{F}^T & \hat{\mathcal{C}}_z \end{bmatrix}. \tag{4.2.22b}$$

For the global-level variables $\mathscr{X}, \mathscr{Y}, \mathscr{F}, \mathscr{G}, \mathscr{H}$ we assume the same sizes and structural constraints as in Theorem 4.2.1. Using the extended transformation diag $(\mathcal{Z}, \mathcal{Z}, Z_{(p)}, Z_{(q)})$ with $\mathcal{Z} = \operatorname{diag}(Z_{(n)}, Z_{(m)}, Z_{(m)})$, we obtain a version of the extended LMI condition (4.2.21) with all blocks being block-diagonal. This allows the decomposition into a family of parameter-dependent conditions (4.2.19) and concludes the proof. Note that as long as Z is unitary, diagonalizing G^{cl} by means of Z does not change its induced $\mathcal{L}_2$ gain, see Eichler and Werner [2013] and Remark 10 in Massioni and Verhaegen [2009]. ■

4.2.3 Construction of the Interconnected Controller

The controller synthesis method described in Section 4.2.1 and 4.2.2 yields a parameter-dependent local controller (4.1.9) in descriptor form, where the coefficient matrix $\mathcal{A}^K(\lambda)$

is obtained from (4.2.18) as an affine function of the parameter λ. In the following we describe how this parameter-dependent controller can be converted into an interconnected controller $K(s)$ of the form (2.1.3).

In the controller construction law (4.2.18) the parameter-dependence comes in by the plant coefficient matrix $\mathcal{A}(\lambda)$. The latter can be written as $\mathcal{A}(\lambda) = \mathcal{A}_0 + \lambda\mathcal{A}_1$. For the controller this leads to the expression $\mathcal{A}^K = \mathcal{A}_0^K + \lambda\mathcal{A}_1^K$ with

$$\begin{aligned} \mathcal{A}_0^K &= \left((\mathcal{A}_0 + X^{-T}G^T\mathcal{C}_v)Y^T - X^{-T}H^T + \mathcal{B}_u F^T\right) S^{-T}, \\ \mathcal{A}_1^K &= \mathcal{A}_1 Y^T S^{-T}. \end{aligned} \tag{4.2.23}$$

The state equation of the controller (4.1.9) then reads

$$\mathcal{E}^K \dot{\xi}_i^K = \mathcal{A}_0^K \xi_i^K + \mathcal{B}_v^K v_i + \lambda \mathcal{A}_1^K \xi_i^K. \tag{4.2.24}$$

Introducing the controller coordination signal $q_i^K = \lambda p_i^K$ and a factorization $\mathcal{A}_1^K = \mathcal{B}_q^K \mathcal{C}_p^K$, the parameter-dependent controller can be formulated as an LFT interconnection of a nominal (and thus parameter-independent) controller in descriptor form and a scheduling block λI:

$$\left(\begin{array}{c} \mathcal{E}^K \dot{\xi}_i^K \\ \hdashline u_i \\ p_i^K \end{array}\right) = \left[\begin{array}{c:cc} \mathcal{A}_0^K & \mathcal{B}^K & \mathcal{B}_q^K \\ \hdashline \mathcal{C}^K & \mathcal{D}^K & 0 \\ \mathcal{C}_p^K & 0 & 0 \end{array}\right] \left(\begin{array}{c} \xi_i^K \\ \hdashline v_i \\ q_i^K \end{array}\right) \tag{4.2.25a}$$

$$q_i^K = \lambda I_{m_K} p_i^K. \tag{4.2.25b}$$

For implementation it is often desirable to convert the nominal controller given by (4.2.25a) into a regular state space representation (4.1.2). From (4.2.18) we know that $\mathcal{E}^K$ is equal to $\mathcal{E}$ given in (4.1.8) with a regular sub-block $E_t \in \mathbb{R}^{n_t \times n_t}$. From this we can infer that the first n_t elements of ξ_i^K are the temporal states to be present as elements of x_i^K in (4.1.2). Partitioning the system matrices of the nominal controller (4.2.25a) accordingly, we obtain the controller K in (4.1.2) as

$$\left[\begin{array}{c:cc} A^K & B_v^K & B_q^K \\ \hdashline C_u^K & D_{uv}^K & D_{uq}^K \\ C_p^K & D_{pv}^K & D_{pq}^K \end{array}\right] = \begin{bmatrix} \mathcal{A}_{011}^K & \mathcal{B}_1^K \\ \mathcal{C}_1^K & \begin{bmatrix} \mathcal{D}^K & 0 \\ 0 & 0 \end{bmatrix} \end{bmatrix} - \begin{bmatrix} \mathcal{A}_{012}^K \\ \mathcal{C}_2^K \end{bmatrix} {\mathcal{A}_{022}^K}^{-1} \begin{bmatrix} \mathcal{A}_{021}^K & \mathcal{B}_2^K \end{bmatrix} \tag{4.2.26}$$

$$\text{with } \mathcal{A}_0^K = \begin{bmatrix} \mathcal{A}_{011}^K & \mathcal{A}_{012}^K \\ \mathcal{A}_{021}^K & \mathcal{A}_{022}^K \end{bmatrix}, \quad \begin{bmatrix} \mathcal{B}^K & \mathcal{B}_q^K \end{bmatrix} = \begin{bmatrix} \mathcal{B}_1^K \\ \mathcal{B}_2^K \end{bmatrix}, \quad \begin{bmatrix} \mathcal{C}^K \\ \mathcal{C}_p^K \end{bmatrix} = \begin{bmatrix} \mathcal{C}_1^K & \mathcal{C}_2^K \end{bmatrix},$$

partitioned according to $\mathcal{A}_{011}^K \in \mathbb{R}^{n_t \times n_t}$. As $E^K = E_t$, for G chosen as regular state space model with $E = I$ (implying $n_t = n$), we obtain $E^K = I$ as well.

The N parallel instances of nominal controllers $K(s)$, each scheduled by LFT connection with one of the diagonal elements λ_i of Λ, are obviously equal to an LFT connection of the block-diagonal controller $\hat{K}(s)$ and the diagonalized pattern matrix Λ. This setup is equal to the LFT of $\hat{K}$ and Ψ, as one can show using a suitable Kronecker extension

of the transformation Z, which commutes with the block-diagonal system $\hat{K}(s)$. In the same manner as in Hoffmann et al. [2015], this setup of $\hat{K}$ and Ψ is interpreted as an interconnection of the controllers by the network described by Ψ.

For practical reasons it is desirable to keep the size m_K of the scheduling block, which is equivalent to the number of communication channels required to implement the distributed controller, as low as possible. For this purpose we can use the structure of the plant coefficient matrices given in (4.1.8), where the parameter-dependent part is already concentrated in the lower right $m \times m$-block of $\mathcal{A}$:

$$\mathcal{A}(\lambda) = \begin{bmatrix} \mathcal{A}_{11} & \mathcal{A}_{12} \\ \mathcal{A}_{21} & \lambda I \end{bmatrix} = \begin{bmatrix} \mathcal{A}_{11} & \mathcal{A}_{12} \\ \mathcal{A}_{21} & 0 \end{bmatrix} + \lambda \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}}_{\mathcal{A}_1} \tag{4.2.27}$$

Assuming a fully populated right factor $R = Y^T S^{-T}$ partitioned according to $\mathcal{A}$, we obtain from (4.2.23)

$$\mathcal{A}_1^K = \mathcal{A}_1 R = \begin{bmatrix} 0 \\ R_2 \end{bmatrix} \quad \text{for } R = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}. \tag{4.2.28}$$

Thus, a reasonable choice for obtaining controller coordination signals of size m is

$$\mathcal{B}_q^K = \begin{bmatrix} 0 \\ I_m \end{bmatrix}, \quad \mathcal{C}_p^K = R_2. \tag{4.2.29}$$

Example 4.2.1. (Coordination of multiple electric circuits)

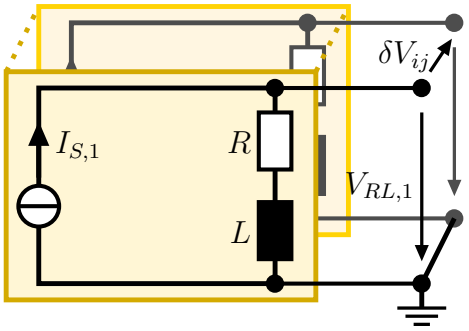


Figure 4.2.1: System of multiple electric circuits with inductive load

This example illustrates the applicability of the descriptor-based controller design approach for multi-agent systems with descriptor systems as agent models. The system under consideration consists of multiple electric circuits as shown in Fig. 4.2.1. Each circuit consists of an inductive load consisting of resistor $R = 1\Omega$ and inductance $L = 1\text{H}$, driven by an ideal current source providing the source current $I_{S,i}$. From Kirchhoff's laws a transfer function from source current $I_{S,i}$ to output voltage $V_{RL,i}$ is obtained as $P(s) = R + Ls$. As this transfer function is not proper (has more zeros

than poles), a regular state space representation does not exist. Nevertheless, there exists a descriptor model with state vector $x_i = [I_{S,i} \quad V_{RL,i}]^T$:

$$\begin{bmatrix} L & \\ & 0 \end{bmatrix} \dot{x}_i = \begin{bmatrix} -R & 1 \\ 1 & 0 \end{bmatrix} x_i + \begin{bmatrix} 0 \\ -1 \end{bmatrix} I_{S,i} \tag{4.2.30a}$$

$$V_{RL,i} = \begin{bmatrix} 0 & 1 \end{bmatrix} x_i. \tag{4.2.30b}$$

As an analogy to the formation control problem, the goal is to control the output voltages $V_{RL,i}$ such that they achieve a "formation", meaning that a specified "spacing" in terms of the voltages δV_{ij} measured between the output terminals of some two circuits i and j is reached. We assume that each current source is governed by a local controller of the form (4.1.2) with $u_i = I_{S,i}$ and $\phi_i = y_i = V_{RL,i}$, which is interacting with those of other circuits. Accordingly, the whole setup is a multi-agent system in the sense of Section 2.1.1.

For controller synthesis, the LMI problem given in Theorem 4.2.2 was solved and the controller was constructed according to (4.2.23) and (4.2.29). The generalized plant model was constructed as shown in Fig. 3.2.6, version (ii), with $\phi_i = y_i$ and shaping filters chosen as

$$W_S(s) = \frac{\omega_S}{M_S} \cdot \frac{1}{s + \omega_S} \tag{4.2.31}$$

$$W_K(s) = \frac{c_K}{M_K} \cdot \frac{s + \omega_K}{s + c_K \omega_K}; \quad c_K > 1. \tag{4.2.32}$$

with coefficients

M_S	ω_S	M_K	ω_K	c_K
0.4	0.2	1e4	5e-3	1e6

All existing communication links are assumed ideal and no communication delays occur. The pattern matrix Ψ is chosen as unweighted adjacency matrix of an undirected topology with degree 2 at each node:

$$\Psi = \begin{bmatrix} 0 & 0.5 & 0 & 0 & 0.5 \\ 0.5 & 0 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0 & 0.5 & 0 \end{bmatrix}, \quad \Lambda = \operatorname{diag} \begin{pmatrix} -0.809 \\ -0.809 \\ 0.309 \\ 0.309 \\ 1 \end{pmatrix}.$$

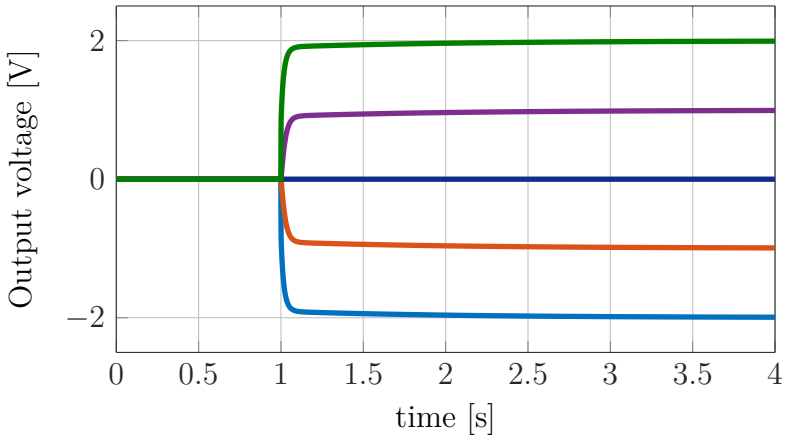


Figure 4.2.2: Simulation results: Output voltage V_{RL} for five circuits

Figure 4.2.2 shows the output voltage $V_{RL,i}$ as simulation result for five circuits. The system is switched on at $t = 1$s with a reference vector given as $r = [1 \quad 2 \quad 3 \quad 4 \quad 5]^T$V corresponding to $\delta V_{ij} = 1$V for all circuit pairs of neighboring index. As visible in Figure 4.2.2, the desired goal is achieved.

4.3 Comparison of Synthesis Approaches

This section examines the relationship between the controller synthesis approach proposed in this thesis for multi-agent systems in descriptor representation and the existing approach from Hoffmann and Werner [2017] using an LFT representation consisting of regular state space models. Both approaches share the same roots in robust control theory and Lyapunov-based stability analysis. Corresponding to the bounded real lemma (Theorems A.2.1 and A.2.2) stated among others in Gahinet [1992]; Scherer [2000], a stability and performance condition for descriptor systems is given in Masubuchi et al. [2003] based on Masubuchi et al. [1997]. For the multi-agent system in descriptor representation this reads

Theorem 4.3.1. ***Bounded Real Lemma*** *Masubuchi et al. [2003] The descriptor system $\mathscr{G}^{cl}$ given in (4.1.6) is admissible [1] for all parameter matrices $\Psi \in \mathbf{\Psi}$ and has a bounded induced $\mathcal{L}_2$ gain $\|\mathscr{G}^{cl}(\Psi)\|_{\mathcal{L}_2} < \gamma$, if the following equivalent statements hold:*

1. *There exists a generalized Lyapunov matrix $\mathscr{Y}_{cl} \in \mathbb{R}^{N(n_D+n_{DK})\times N(n_D+n_{DK})}$ fulfilling*

$$\hat{\mathcal{E}}_{cl}\mathscr{Y}_{cl}^T = \mathscr{Y}_{cl}\hat{\mathcal{E}}_{cl}^T \geq 0 \tag{4.3.1}$$

$$\begin{bmatrix} \mathscr{Y}_{cl}\mathscr{A}_{cl}^T + \mathscr{A}_{cl}\mathscr{Y}_{cl}^T & \mathscr{Y}_{cl}\hat{\mathcal{C}}_{cl}^T & \hat{\mathcal{B}}_{cl} \\ \hat{\mathcal{C}}_{cl}\mathscr{Y}_{cl}^T & -\gamma I & \hat{\mathcal{D}}_{cl} \\ \hat{\mathcal{B}}_{cl}^T & \hat{\mathcal{D}}_{cl}^T & -\gamma I \end{bmatrix} < 0 \quad \forall \Psi \in \mathbf{\Psi}. \tag{4.3.2}$$

[1] "Admissible" in Masubuchi et al. [2003] means that the pencil $s\hat{\mathcal{E}}_{cl} - \mathscr{A}_{cl}$ is regular, impulse-free and has no unstable exponential modes.

2. There exists a generalized Lyapunov matrix $\mathscr{X}_{cl} \in \mathbb{R}^{N(n_D+n_{DK})\times N(n_D+n_{DK})}$ *fulfilling*

$$\hat{\mathscr{E}}_{cl}^T \mathscr{X}_{cl} = \mathscr{X}_{cl}^T \hat{\mathscr{E}}_{cl} \geq 0 \tag{4.3.3}$$

$$\begin{bmatrix} \mathscr{X}_{cl}^T \mathscr{A}_{cl} + \mathscr{A}_{cl}^T \mathscr{X}_{cl} & \mathscr{X}_{cl}^T \hat{\mathscr{B}}_{cl} & \hat{\mathscr{C}}_{cl}^T \\ \hat{\mathscr{B}}_{cl}^T \mathscr{X}_{cl} & -\gamma I & \hat{\mathscr{D}}_{cl}^T \\ \hat{\mathscr{C}}_{cl} & \hat{\mathscr{D}}_{cl} & -\gamma I \end{bmatrix} < 0 \quad \forall \Psi \in \mathbf{\Psi}. \tag{4.3.4}$$

While in this theorem no constraints are stated on the Lyapunov matrix itself, the bounded real lemma formulated for the regular state space representation (see Theorem 3.1.2) requires the Lyapunov matrix to be symmetric and positive definite. Instead, here symmetry and positive semidefiniteness are required for the expression $\mathscr{Y}_{cl}\hat{\mathscr{E}}_{cl}^T$ (or $\mathscr{X}_{cl}^T\hat{\mathscr{E}}_{cl}$, respectively). To illustrate the differences between the descriptor approach and the existing state space approach, the bounded real lemma for the descriptor representation can be reformulated according to the multiplier form of (3.1.9): For comparison, the class of systems under consideration is limited to those covered by the existing approach, which are characterized by $E = I_n, E^K = I_{n_K}$ and thus $n_t = n$. Using a suitable permutation on the closed loop descriptor system leads to a re-ordered state vector $\xi = \mathrm{vcat}(x, x^K, q, q^K, p, p^K)$ and $\hat{\mathcal{E}} = \mathrm{diag}(\hat{E}, \hat{E}^K, 0, 0)$. This allows a partitioning into temporal and algebraic states, i.e. $\xi = \mathrm{vcat}(\xi_1, \xi_2)$ with $\xi_1 = \mathrm{vcat}(x, x^K)$. The system matrices can be partitioned accordingly. The Lyapunov matrix is partitioned as

$$\mathscr{X} = \begin{bmatrix} \mathcal{X} & \mathcal{N} \\ \mathcal{M} & \mathcal{P} \end{bmatrix}, \quad \mathcal{X} \in \mathbb{R}^{N(n+n_K)\times N(n+n_K)}. \tag{4.3.5}$$

From (4.3.3) one can infer $\mathcal{N} = 0$ and $\mathcal{X} = \mathcal{X}^T > 0$. The bounded real lemma (4.3.4) is now reformulated as

$$\begin{bmatrix} * \\ * \\ * \\ * \end{bmatrix}^T \left[\begin{array}{cc:cc:cc} 0 & \mathcal{X} & & & & \\ \mathcal{X} & 0 & & & & \\ \hdashline & & 0 & \mathcal{M} & & \\ & & \mathcal{M}^T & 0 & & \\ \hdashline & & & & 0 & \mathcal{P} \\ & & & & \mathcal{P}^T & 0 \\ \hdashline & & & & & \\ & & & & & \end{array}\right.$$
$$\left.\begin{array}{cc} & \\ & \\ & \\ & \\ 1/\gamma I & 0 \\ 0 & -\gamma I \end{array}\right] \left[\begin{array}{ccc} \mathscr{A}_{11}^{cl} & \mathscr{A}_{12}^{cl} & \mathcal{B}_1^{cl} \\ I & 0 & 0 \\ \hdashline \mathscr{A}_{21}^{cl} & \mathscr{A}_{22}^{cl}(\Psi) & \mathcal{B}_2^{cl} \\ I & 0 & 0 \\ \hdashline \mathscr{A}_{21}^{cl} & \mathscr{A}_{22}^{cl}(\Psi) & \mathcal{B}_2^{cl} \\ 0 & I & 0 \\ \hdashline \mathcal{C}_1^{cl} & \mathcal{C}_2^{cl} & \mathcal{D}^{cl} \\ 0 & 0 & I \end{array}\right] < 0, \tag{4.3.6}$$

A comparison of this expression with the bounded real lemma (3.1.9) for the regular state space representation shows that $\mathcal{X}$ of (4.3.6) corresponds to the Lyapunov matrix $\mathscr{X}$ of (3.1.9) subject to the same constraints. Meanwhile, the additional variables $\mathcal{M}$ and $\mathcal{P}$ are not constrained at all and represent additional degrees of freedom. For the synthesis problem in descriptor formulation this fact promises a reduction of conservatism.

Table 4.1 gives an overview of the LMI sizes, the contained decision variables and their additional constraints for the LMI problem finally obtained after applying the described

	FBSP-based approach	Descriptor-based approach
Decision variables	• $X, Y \in \mathbb{R}^{n\times n}$ • $R, S, \tilde{R}, \tilde{S} \in \mathbb{R}^{m\times m}$ • $\gamma \in \mathbb{R}_+$	• $X_{11}, Y_{11} \in \mathbb{R}^{n\times n}$ • $X_{21}^T, Y_{12} \in \mathbb{R}^{n\times 2m}$ • $X_{22}, Y_{22} \in \mathbb{R}^{2m\times 2m}$ • $H \in \mathbb{R}^{(n+2m)\times(n+2m)}$ • $F \in \mathbb{R}^{(n+2m)\times h}$ • $G \in \mathbb{R}^{l\times(n+2m)}$ • $\gamma \in \mathbb{R}_+$
Constraints	• $X = X^T$ • $Y = Y^T$ • $\begin{bmatrix} X & I \\ I & Y \end{bmatrix} > 0$ • $R = R^T > 0,\ \tilde{R} = \tilde{R}^T < 0$ • $S + S^T = 0,\ \tilde{S} + \tilde{S}^T = 0$	• $X_{11} = X_{11}^T$ • $Y_{11} = Y_{11}^T$ • $\begin{bmatrix} Y_{11} & I \\ I & X_{11} \end{bmatrix} > 0$
LMI sizes	• Primal LMI: $n_{\mathcal{N}_X}$ • Dual LMI: $n_{\mathcal{N}_Y}$	• LMI for $\underline{\lambda}$: $2(n+2m)+p+q$ • LMI for $\overline{\lambda}$: $2(n+2m)+p+q$

Table 4.1: Properties of the synthesis problems in FBSP and Descriptor approach

steps such as the full block $\mathcal{S}$-procedure, the projection lemma or the change of variables, respectively. Here the correspondence between X, Y of the FBSP approach and X_{11}, Y_{11} of the descriptor-based approach is visible, the same is true for the performance indicator γ. In contrast, for the other variables no obvious correspondence is observed. The descriptor-based approach comprises more and larger matrix variables, which in contrast to the FBSP multipliers $R, S, \tilde{R}, \tilde{S}$ are not subject to any constraints at all. This again corresponds to the additional degree of freedom.

The size of the LMIs to be solved in the FBSP-based approach depend on $n_{\mathcal{N}_X}$ and $n_{\mathcal{N}_Y}$, which denote the dimension of the kernel spaces $\mathcal{N}_X$ and $\mathcal{N}_Y$. These spaces defined in (3.1.19c, 3.1.19d) are the null spaces of the feedback output matrix $[C_v \quad D_{vq} \quad D_{vw}]$ and the transposed feedback input matrix $[{B_u}^T \quad {D_{pu}}^T \quad {D_{zu}}^T]$, which are the spaces to project on when using the projection lemma to eliminate the controller matrices. Referring to

(4.1.1), their dimensions are bounded by

$$n + m + p - l \leq n_{\mathcal{N}_X} \leq n + m + p, \tag{4.3.7a}$$
$$n + m + q - h \leq n_{\mathcal{N}_Y} \leq n + m + q \tag{4.3.7b}$$

depending on the rank of the input and output matrices. This shows that the descriptor-based approach requires LMIs of a larger size to be solved and thus unveils a drawback of the descriptor-based approach.

To study the effect of these properties on the resulting control performance, the proposed synthesis approach was applied to the same numerical example for which results were obtained using the FBSP approach as part of the simulation study presented in Section 3.3.3. It considers the formation control problem for quad-rotor helicopters. Here the results for both approaches will be compared:

Example 4.3.1. (Formation control of quad-rotor helicopters)

The scenario of this example is a formation flight of five quad-rotor helicopters, each modeled by a linearized unstable 12^{th} order state space model proposed in Lara et al. [2006]. Their goal is to achieve and maintain a given geometric formation with relative spacing as in $r_x = [1 \quad 2 \quad 3 \quad 4 \quad 5]^T$. Motion is modeled in 3D, but here only results in x-direction will be displayed.

For controller synthesis the generalized plant was constructed in the same manner as in Example 4.2.1 with shaping filters of the same form, but with different tuning values:

M_S	ω_S	M_K	ω_K	c_K
0.01	0.01	1e4	1e3	1e4

For solving the main LMI problem the following computation times were measured:

Synthesis method	computation time [sec]
Descriptor approach	5.53
FBSP approach	5.06

In this example they are almost equal, where the FBSP approach is slightly faster. The expected longer computation time for the descriptor method due to its larger LMI problem is confirmed by this result, but in this example the magnitude of this effect is observed to be marginal.

For simulations of this example the assumptions about the communication topology and the pattern matrix Ψ were adopted from Example 4.2.1.

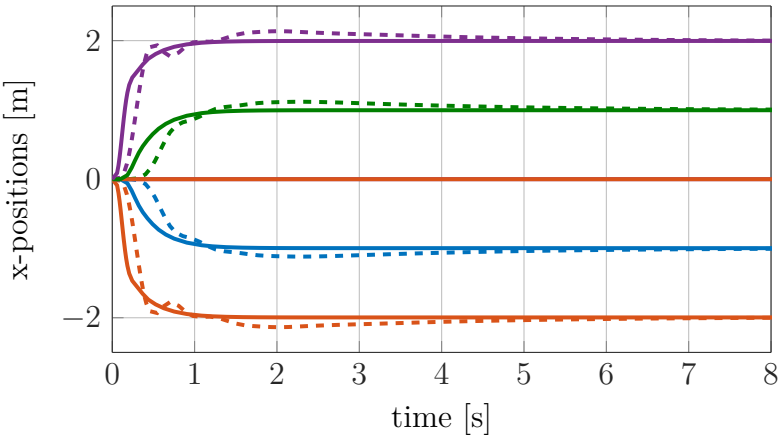


Figure 4.3.1: Simulation results $y_{i,x}$ for five quad-rotor helicopters, using the resulting controllers from the descriptor-based (solid) and the FBSP-based approach (dashed)

Fig. 4.3.1 shows simulation results of a formation flight in terms of the position of the agents in x-direction. The results show a successful formation establishment for both approaches. While the rise time is almost equal, the FBSP-based approach suffers from an oscillatory response with a significant overshoot resulting in a considerably longer settling time.

Although the agents in the previous example are modeled as regular state space models and thus allow the usage of existing state space approaches (as presented here for comparison), the performance benefit shows that it is nevertheless worth to use the descriptor framework to handle the interaction.

4.4 Extension to Non-symmetric Interaction

The synthesis approach for descriptor multi-agent systems presented so far is limited to interaction topologies represented by an unitarily diagonalizable interaction matrix Ψ. The interconnection-scheduled approach from Hoffmann et al. [2015], from which the decomposition methodology was adopted, in its basic version suffers from the same limitation. However, as described in Section 3.1.2, Hoffmann and Werner [2017] propose a method to overcome this limitation by using an augmented system model: It is possible to find matrices V and W with $VW = aI$ (a is some real scalar) such that $\Psi = V\tilde{\Psi}W$ with $\tilde{\Psi}$ unitarily diagonalizable (and thus normal). Accordingly, it is an obvious idea to adopt this method to the descriptor synthesis approach as well.

Having substituted the interaction matrix as $\Psi = V\tilde{\Psi}W$ within the loop consisting of agent dynamics and interaction matrix (see Fig. 3.1.1), the transformation matrices W and V can be considered part of an augmented agent model by introducing

$$\tilde{p} = W_{(m)}p, \quad q = V_{(m)}\tilde{q}. \tag{4.4.1}$$

With $\tilde{\Psi} \in \mathbb{R}^{\beta N \times \beta N},\ \beta \in \mathbb{N}$, an augmented version of the system model (2.1.7a) is obtained as

$$\hat{\tilde{G}} : \left(\begin{array}{c} \dot{x} \\ \hdashline z \\ \tilde{p}^P \\ v \end{array}\right) = \left[\begin{array}{c:ccc} \hat{A} & \hat{B}_w & \hat{B}_q V_{(m)} & \hat{B}_u \\ \hdashline \hat{C}_z & \hat{D}_{zw} & \hat{D}_{zq} V_{(m)} & \hat{D}_{zu} \\ W_{(m)} \hat{C}_p & W_{(m)} \hat{D}_{pw} & W_{(m)} \hat{D}_{pq} V_{(m)} & W_{(m)} \hat{D}_{pu} \\ \hat{C}_v & \hat{D}_{vw} & \hat{D}_{vq} V_{(m)} & \hat{D}_{vu} \end{array}\right] \left(\begin{array}{c} x \\ \hdashline w \\ \tilde{q}^P \\ u \end{array}\right) \tag{4.4.2}$$

$$\tilde{q}^P = \tilde{\Psi}_{(m)} \tilde{p}^P \tag{4.4.3}$$

with interaction channels $\tilde{p}_i^P, \tilde{q}_i^P$ of size $\tilde{m} = \beta m$. The decomposability of this augmented system model is preserved by choosing augmentation matrices composed of diagonal blocks such as (3.1.24).

The augmented LFT state space model (4.4.2, 4.4.3) corresponds to an augmented version of the descriptor model (4.1.3) in terms of the state vector $\tilde{\xi} = \mathrm{vcat}(x, \tilde{q}^P, \tilde{p}^P)$ with the following system matrices:

$$\begin{aligned} \tilde{\mathscr{A}}(\tilde{\Psi}) &= \begin{bmatrix} \hat{A} & \hat{B}_q V_{(m)} & 0 \\ W_{(m)} \hat{C}_p & W_{(m)} \hat{D}_{pq} V_{(m)} & -I_{\beta N m} \\ 0 & -I_{\beta N m} & \tilde{\Psi}_{(m)} \end{bmatrix} \\ \hat{\tilde{\mathcal{B}}}_w &= \begin{bmatrix} \hat{B}_w \\ W_{(m)} \hat{D}_{pw} \\ 0 \end{bmatrix}, \quad \hat{\tilde{\mathcal{B}}}_u = \begin{bmatrix} \hat{B}_u \\ W_{(m)} \hat{D}_{pu} \\ 0 \end{bmatrix}, \\ \hat{\tilde{\mathcal{C}}}_z &= \begin{bmatrix} \hat{C}_z & \hat{D}_{zq} V_{(m)} & 0 \end{bmatrix}, \quad \hat{\tilde{\mathcal{C}}}_v = \begin{bmatrix} \hat{C}_v & \hat{D}_{vq} V_{(m)} & 0 \end{bmatrix}. \end{aligned} \tag{4.4.4}$$

With $\tilde{\Psi}$ being unitarily diagonalizable at all times, one could naively assume this problem formulation to fulfill the conditions for Theorems 4.2.1 and 4.2.2 to be applied. However, the decomposition of the LMI problem in Theorem 4.2.1 makes use of the fact that the system matrices of the global system model commute with the transformation matrix diagonalizing the interaction matrix. For the original system model, this property is ensured by the structure of the global system model, where the system matrices have the block-diagonal structure $I_N \otimes M$ with the number of repetitions corresponding to the size of Ψ. In the augmented system model (4.4.4), this property is no longer given. For this property to be given, the augmented system matrices would need to be block-diagonal with $2N$ repetitions, corresponding to $\tilde{\Psi}$, which is not the case. Thus, Theorems 4.2.1 and 4.2.2 are not necessarily applicable to this model. For this reason, an extension of the descriptor approach to multi-agent systems with non-symmetric interaction is not straight-forward and remains unsolved in this thesis.

Chapter 5

Experiments on Multi-Agent Systems

This chapter presents the experimental work conducted in the scope of this thesis. The goal of this work was the development of a test platform and based thereon an experimental examination and validation of the theoretical approaches presented in the previous chapters. Although the experimental results actually achieved in the scope of this work do not provide a comprehensive validation of the proposed theoretical concepts, a proof-of-concept is provided for different approaches to the realization and examination of distributed formation control schemes on multi-agent systems.

5.1 Experimental Platforms

The experimental platforms considered in this thesis are mainly oriented on multiple quad-rotor helicopters (also short-named *quad-copters*) as mobile agents, which collaborate according to the concept of multi-agent systems. In this context, two different types of experimental platforms were examined: Firstly, the development of fully autonomous quad-rotor devices in out-door operation was continued based on preceding experimental research of the institute [Pilz, 2013]. The key feature of this work is an actual realization of the considered agent capabilities regarding interaction and control by means of on-board hardware and direct agent-to-agent data exchange. The major challenge is the precise localization of the agents.

The second type of experimental work was motivated by the advantages of camera-based indoor localization (see Section 5.1.3) and the desire to conduct indoor formation flight experiments avoiding the drawbacks of outdoor experiments such as wind influences or the demand for a suitable outdoor test field. To cope with the limited available flight space, which is constrained by the camera vision space, parts of the MAS functionalities to be tested are implemented virtually on a laboratory PC, which is referred to as Hardware In the Loop (HIL) experiment. In this concept either a reduced number of physical agents with full functional range is operated and complemented by further virtual agents, or

	Hummingbird	**CX-10**
Manufacturer	Ascending Technologies	Cheerson
Size [mm]	540 × 540 × 85	40 × 40 × 22
Weight [g]	530	14.2
Flight time [min]	20	5
Sensors	• 3-axes gyro • 3-axes accelerometer • 3-axes magnetometer • barometric height sensor • GPS	• 3-axes gyro • 3-axes accelerometer (both not accessible)
User-programmable CPU	LPC2146 ARM by NXP	none

Table 5.1: Properties of the quad-rotors used in the scope of this thesis

smaller agents with reduced on-board capabilities are used with an external realization of the missing functionalities.

5.1.1 Quad-Rotor Helicopters

For the preceding experimental research of the institute, a hardware platform had been designed in a joint project of the Institute of Reliability Engineering, the Institute of Aircraft System Engineering and the Institute of Control Systems [Witt et al., 2011] and experienced several modifications. Although some formation flight experiments had been successfully conducted using this hardware (results are published in Pilz [2013]), at the time the work reported in this thesis started, it still suffered from significant weaknesses especially affecting its reliability. For this reason, the decision was taken to purchase a fleet of four commercially produced quad-copters of the type *Hummingbird* developed and produced by the company *Ascending Technologies*[1]. These devices came fully equipped with an accessible on-board computation unit, the required sensors and wireless communication hardware to provide fully autonomous operation. While these devices showed a remarkable benefit in terms of flight properties and reliability compared to the previous hardware, the localization of the quad-rotor, which is an important requirement for successful position control, still remained an open challenge.

[1] `http://www.asctec.de`, more detailed information is provided in the AscTec Research Line wiki at `http://wiki.asctec.de` [Ascending Technologies, 2017a]

In context of indoor formation flight experiments, low-cost miniature quad-copters *Cheerson CX-10* were chosen as hardware platform. These devices, of which 16 units were purchased, significantly reduce the space requirements at the price of less on-board capabilities.

For future experiments, an interesting perspective is given by the *Crazyflie 2.0* miniature quad-copter provided by the manufacturer *Bitcraze*[2] as open source development platform. It is equipped with open-accessible sensing and computation hardware comparable to the Hummingbird on-board hardware, while size and weight allow indoor experiments at low space consumption. However, this quad-copter only gained maturity during the time this thesis was composed and therefore was not available for experiments in the scope of this thesis.

Table 5.1 shows the fundamental properties of the quad-rotor helicopters used for experiments in the scope of this thesis. In the following, the hardware and software setup is described for both types of quad-rotor helicopters.

Hummingbird On-Board Hardware

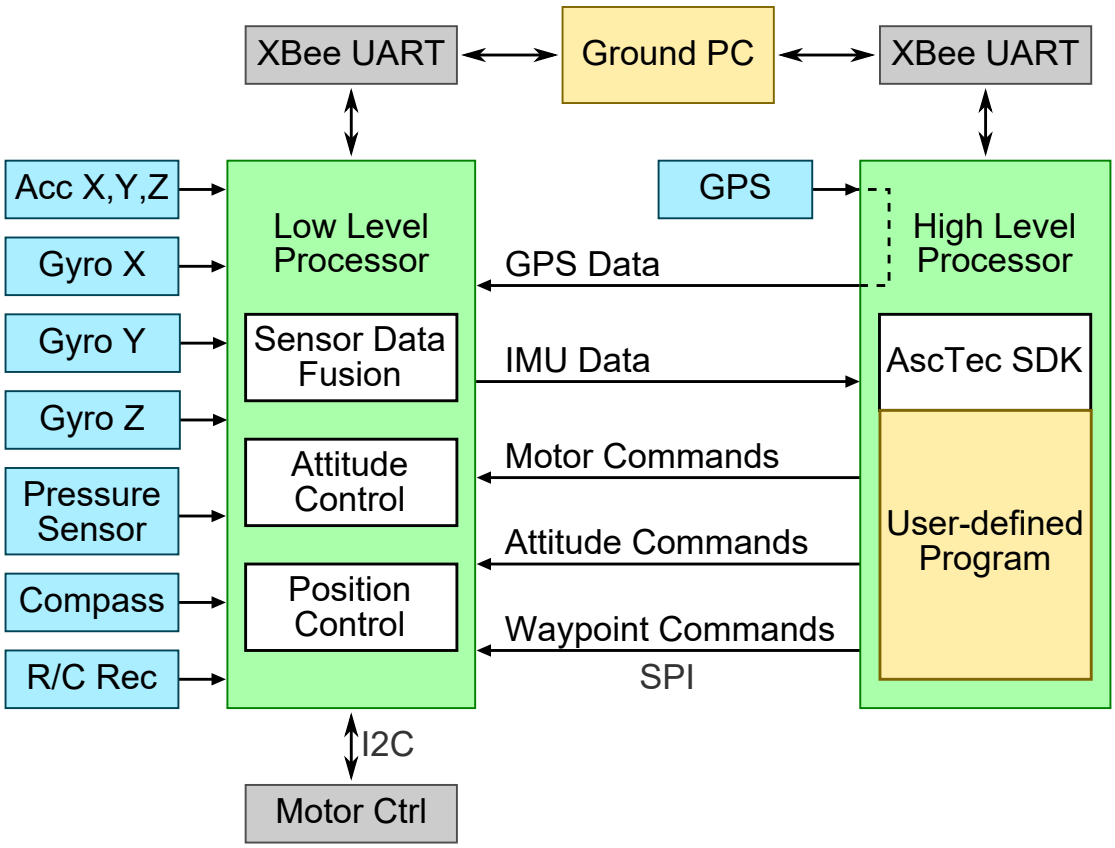


Figure 5.1.1: On-board setup of Hummingbird quad-rotor

Each Hummingbird quad-rotor is equipped with a main control unit named *AscTec Au-*

[2]Detailed information is available at `https://www.bitcraze.io` and `https://wiki.bitcraze.io`

toPilot. This module comprises two LPC2146 ARM microprocessors [Ascending Technologies, 2012], an Inertial Measurement Unit (IMU) and various interfaces. Figure 5.1.1 schematically shows the setup of the AscTec AutoPilot. The Low Level Processor (LLP) is programmed by the manufacturer to provide low level functionalities comprising sensor data fusion, attitude control and waypoint trajectory control (referred to as position control by AscTec). It is not intended to be programmed by the user. In contrast, the High Level Processor (HLP) is intended to execute custom code and can be programmed via a JTAG interface.

Except for the GPS receiver, which is connected to the HLP, all sensors are connected to the LLP. Both processors can exchange data via an SPI interface running at 1 kHz. It allows the HLP to access sensor data and the LLP to receive flight control commands from the HLP. For communication with a ground station, both processors can be connected to *XBee* radio modules via UART. While this interface is not prepared by the manufacturer to establish a direct link between multiple quad-rotors, such a link was enabled in the scope of this work by modifying the on-board software accordingly [Gonzalez Cisneros, 2014].

On-Board Software

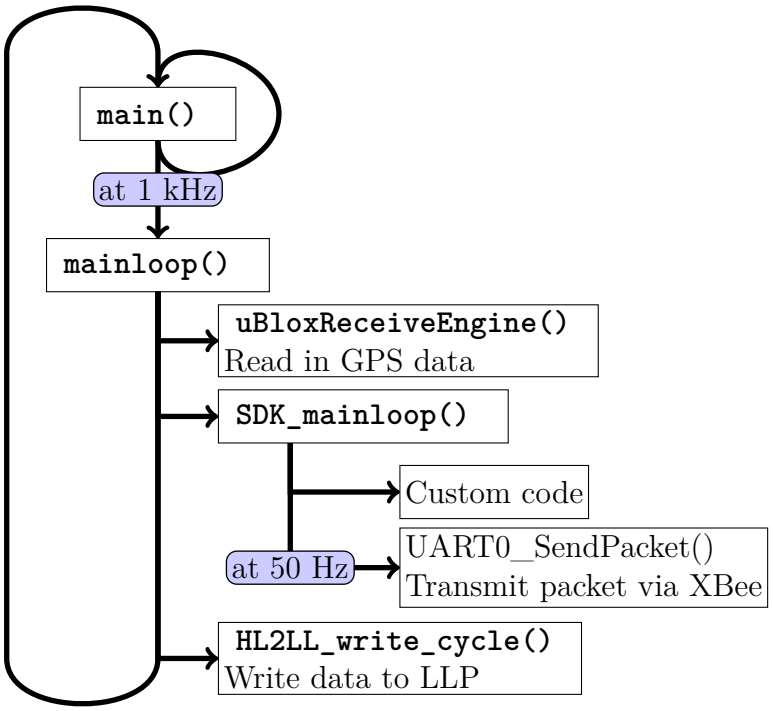


Figure 5.1.2: Structure of Hummingbird HLP on-board program

To support the development and upload of custom C-code to the HLP, an eclipse-based SDK is provided by AscTec together with a C-coded software framework providing many basic functionalities as a wrapper for custom code. The structure of this wrapper program

is shown in Fig. 5.1.2: The function `mainloop()` is triggered by a timer at a frequency of 1 kHz and calls functions to execute the tasks to be fulfilled repeatedly. Besides reading in GPS data and writing data to the LLP, this comprises the function `SDK_mainloop()` designated to contain custom code such as a controller. For the development of custom code, AscTec provides a *Simulink toolkit*. It contains a Simulink model with pre-defined interfaces, from which C-code can be generated automatically using the *Simulink coder*.

The sensor data obtained from the LLP and the GPS receiver are stored in global variables, which are accessible within the functions executed at the HLP. In the same manner, global variables are defined for transmitting values such as control input data to the LLP. On the LLP control data are accepted in four different modes:

1. Direct motor access: Set individual motor thrust directly
2. Motor mapping: Command $(\phi, \theta, \dot{\psi}, F_{Th})$ for feed-forward mapping into motor thrust
3. Attitude control: Command attitude controller references $(r_\phi, r_\theta, r_\psi)$ for ϕ, θ, $\dot{\psi}$ and thrust setpoint F_{Th}
4. Waypoint control: Command waypoints for trajectory controller

The waypoint mode is not part of the Simulink toolkit provided by AscTec, but was included in this framework in the scope of this thesis [Bilgin, 2014].

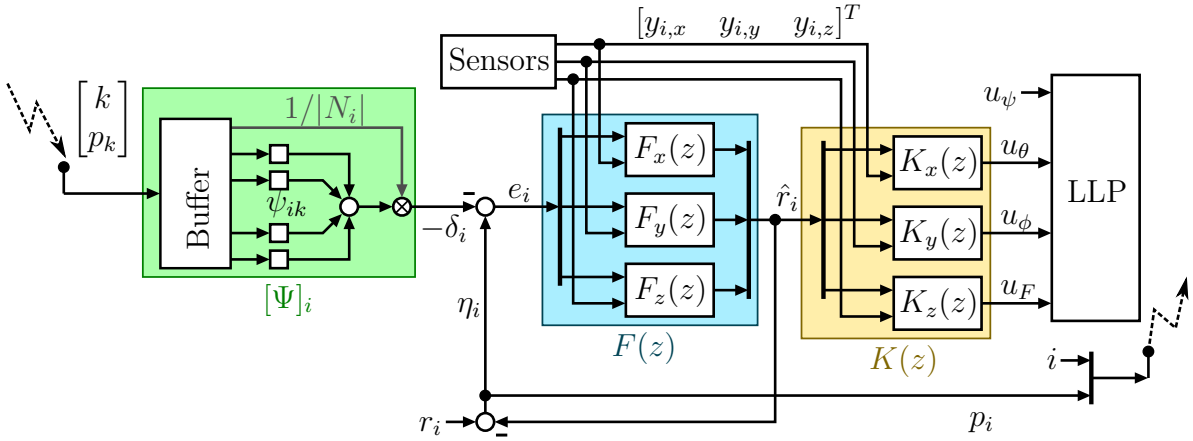


Figure 5.1.3: Structure of the Hummingbird on-board software implementing the extended consensus-based formation control scheme

In the scope of this work, the consensus-based formation control scheme according to Section 3.2.3 and the extended version according to Section 3.2.4 were implemented based on the Hummingbird framework. Fig. 5.1.3 shows a block diagram of the on-board software implementing the extended consensus-based control scheme. The blocks $F_\bullet(z)$ and $K_\bullet(z)$ are discrete-time state space implementations of the extended information flow filter and the local position controller. Outputs of the controllers are reference values of total thrust, pitch and roll angle as inputs to the local attitude controller, which is executed on the LLP operated in Mode 3 with inputs $u = [r_\psi \quad r_\theta \quad r_\phi \quad F_{Th}]$. The yaw rate reference is set to zero.

CX-10 Hardware and Software

As the CX-10 quad-rotor is a low-cost design focused on the needs of hobby usage, the on-board hardware is kept lean and no official documentation is available. Here we refer to a third party review [RCGroups.com, 2017]. The on-board hardware consists of an STM32 F050K4 microcontroller, an Infineon 6050C IMU (3-axes gyro sensors and 3-axes accelerometer) and a BK2421 radio receiver chip. The design does not provide any custom programming of the microcontroller and no interface is provided to access the IMU data. To manually control the device, a remote control unit is provided using a proprietary protocol for sending control commands to the internal attitude controller. For the implementation of custom outer control schemes, an interface module was developed in the scope of this thesis to transmit control commands from a PC to the quad-rotor by imitating the remote control protocol [3]. This module consists of an Arduino microcontroller board with USB interface and an NRF24L01+ radio transceiver module. The commands are accepted on the USB port in terms of the total thrust and reference values of roll and pitch angle as well as yaw rate, i.e. $u = [F_{Th} \quad r_\phi \quad r_\theta \quad r_\psi]$. This corresponds to mode 3 of the Hummingbird quad-rotor.

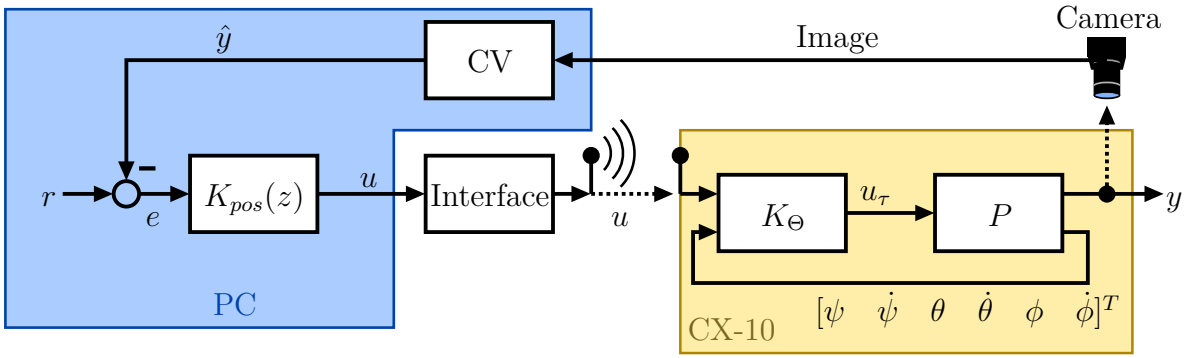


Figure 5.1.4: Position control scheme for a CX-10 quad-rotor with camera-based localization

In contrast to the Hummingbird quad-rotors, which provide programmable on-board hardware, for the CX-10 all control levels above the local attitude control had to be implemented externally. For this reason the position control loop is closed using a central lab PC performing both localization and position control. The control architecture is shown in Fig. 5.1.4. Position feedback is provided by the camera-based localization system, for which the camera periodically feeds images to the PC. These images are processed by a Computer Vision algorithm providing a 3D position estimation $\hat{y}$ for each quad-rotor visible in the camera image. This position estimation is compared to the reference r to obtain the error, which is fed to the controller $K_{\text{pos}}(z)$. The controller output data u are transmitted to the quad-rotor using the transmission protocol interface.

[3] This work was done by Peter Paulsen based on code provided at `github.com` by the users "samuelpowell" and "goebish".

5.1.2 Communication and Interaction

The existence of data exchange links between the agents is a crucial prerequisite for the implementation of cooperative control in a multi-agent system. In case of mobile robots or even flying devices as agents, as considered in this thesis, it is clear that wireless communication among the agents is required. The theoretical concept of multi-agent systems according to Section 2.1.1 considers a peer-to-peer network, i.e. the agents communicate to their neighbors by direct links only and no forwarding of packets takes place. Thus, no additional network components such as routers or coordinator nodes are considered. With respect to the OSI model, for the realization of such a system the realization of the physical and the Media Access Control (MAC) layer can be assumed sufficient. Besides the agent-to-agent communication network, for practical implementations additional links from the agents to a ground node are required for monitoring and safety purposes.

Full Realization

For the experimental platforms examined in this thesis, wireless communication hardware operating in the 2.4 GHz Industrial, Scientific and Medical (ISM) band was used. The original equipment of the Hummingbird quad-copters comprises *XBee Pro* radio modules implementing the IEEE 802.15.4 standard. With a transmission power of 10 mW, a range of 750 m is stated by the manufacturer [Digi International, 2015]. The IEEE 802.15.4 standard defines both physical and MAC layer, where the latter follows the Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) principle. For networking functionalities such as routing, the usage of a network layer according to another standard is intended, for which some standards such as *ZigBee* are suited to the IEEE 802.15.4 MAC layer. However, in this work only physical and MAC layer are used. The radio modules are configured to use the same channel across the network, such that all messages are transmitted as broadcast and can be received by all agents within the transmission range. As long as no further disturbing effects are acting, the resulting topology is a proximity graph.

A drawback of the XBee modules with respect to future research is their fixed implementation of the IEEE 802.15.4 MAC layer, which does not allow the implementation of a custom MAC protocol. However, for the performance of distributed control schemes, in some cases such as large swarms with a high density of agents, the MAC protocol can be a critical feature. To allow experiments on this aspect, some tests were performed in this work using the NRF24L01+ transceiver in connection with a simple custom CSMA/CA protocol [Farnbacher, 2017]. The results indicate a good suitability of this hardware for experiments on distributed control.

Above the two basic layers of the wireless links, a custom layer has been developed and implemented in the scope of Gonzalez Cisneros [2014] to realize the interaction considered throughout this thesis (see Section 2.1.1). The main part is the *Buffer*, which processes data packets received by the on-board radio module, manages the information about the connected neighbors and provides the control-related data q_i to the controller. Fig. 5.1.5

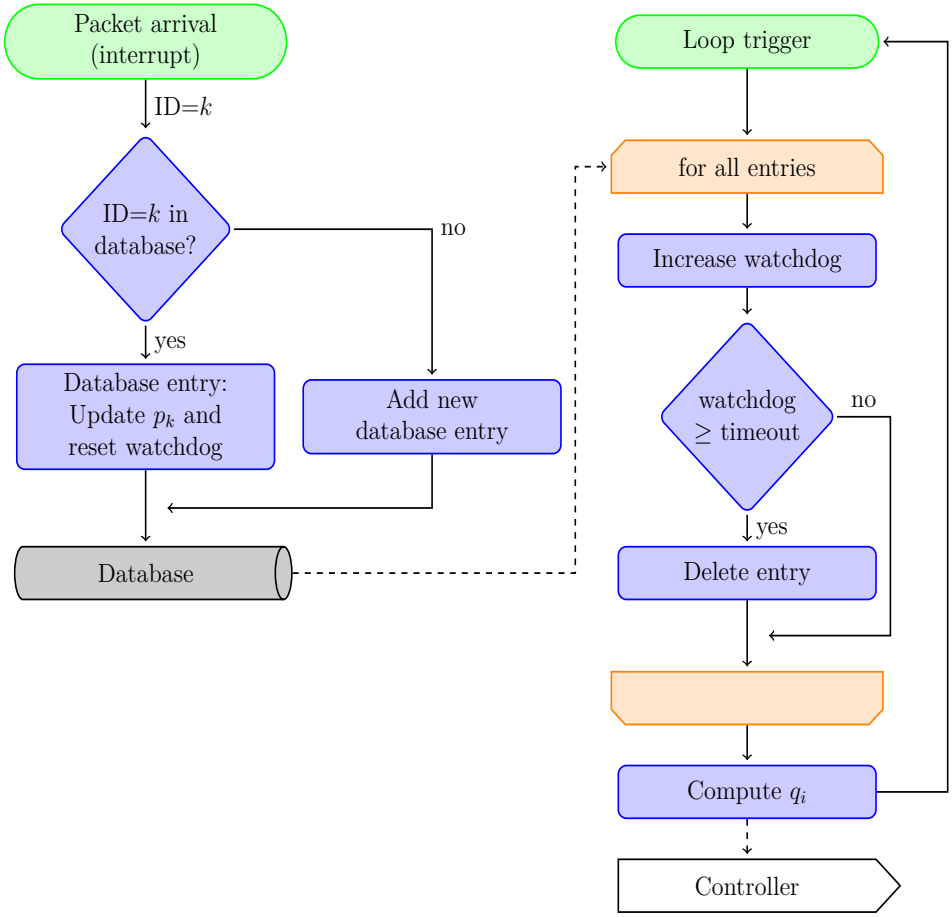


Figure 5.1.5: Flow diagram of the buffer

shows the program flow of this buffer. Each data packet is assembled as

$$P_i = \begin{bmatrix} i & p_{i,x} & p_{i,y} & p_{i,z} \end{bmatrix}^T, \tag{5.1.1}$$

where i is the ID of the transmitting agent. To manage the incoming connections, each agent maintains a database, which contains one entry for each connected neighbor and thus represents the set N_i. Each entry contains the neighbor ID, the latest coordination data p_k received from this neighbor and a watchdog timer. An incoming packet P_k is processed by updating the entry for ID k or, if no such entry exists, creating a new one. To detect a link failure, the watchdog timer is used. It is increased in every main loop cycle and is reset when a packet from the particular agent is processed. In each loop cycle all entries with a watchdog value exceeding a given threshold (timeout period) are deleted from the database. Exceeding the threshold means that at least the timeout period has passed since the last packet from the respective neighbor has arrived. Thus, it is assumed that the link with this neighbor no longer exists. At each controller execution step, the number of neighbors $d_i = |N_i|$ is determined by counting the entries (after deleting the timeout entries) and the value q_i is computed as

$$q_i = \frac{1}{d_i} \sum_{k \in N_i} p_k \tag{5.1.2}$$

summing up all neighbor data currently stored in the database. This corresponds to choosing the scaled adjacency matrix $\mathcal{A}$ as pattern matrix Ψ.

Hardware-in-the-Loop Realization

The communication setup described so far aims at a full realization of the considered agent interaction and is a reasonable approach for fully equipped agents such as the Hummingbird or Crazyflie. However, it requires the on-board capabilities to implement the data management as well as the agent-to-agent connections. For agents with reduced functional range, such as the CX-10, the interaction part can be emulated on the central laboratory PC as an extension of the hardware-in-the-loop setup shown in Fig 5.1.4. In the experiments conducted with CX-10 quad-rotors in this work, the communication was realized by simply implementing the matrix representation $q = \Psi_{(m)}p$ for a given interaction matrix Ψ.

Besides allowing agents with reduced on-board capabilities, this realization has a second benefit: In recent research programmes such as the DFG Priority Programme "Cyber-Physical Networking"[4], a focus is set on the data transmission network and its role in the control scheme. This comprises simulation studies with an advanced and more detailed representation of the communication network. In this context the HIL framework allows to combine the advanced network model with actual agent hardware and offers a rapid-prototyping testbed for experimental validation of advanced communication protocols and control schemes. This is particularly attractive for experiments in which the MAS is to be tested under critical conditions like large distances or critical communication load. For such scenarios, the risk of communication failures or instability of formation control is fairly high. For a full on-board realization, this will likely lead to losing control of one or more agents and thus means a significant risk of serious damage. In contrast, in a HIL realization the interaction takes place within the lab PC program and allows critical outputs to be trapped, such that the physical agents remain in safe operation.

5.1.3 Localization

An important prerequisite for successful implementation of formation control in multi-robot systems and at the same time one of the major challenges is the accurate localization of the robots. At the beginning of the work reported in this thesis, the equipment to be used did not provide a satisfactory localization solution. For this reason, a significant part of the experimental effort was spent for developing and testing suitable localization solutions for the experimental platforms used. To allow formation control of quad-copters, it is reasonable to claim that the localization error must not exceed the order of magnitude of the dimensions of a single quad-copter. As second relevant property, the sampling rate must be significantly higher than the bandwidth of the motion dynamics of the quad-copter. While a factor of 2 is the theoretical limit to capture the motion, from practical

[4]Priority Programme "Cyber-Physical Networking" (SPP 1914) by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), started in 2016, `www.spp1914.de`

Technology	Usage	Precision[5]	Comments
GPS uncorrected (NEO-6P)	outdoor	1 m	
GPS with PPP (SBAS)	outdoor	0.83 m	
GPS with PPP (IGS)	outdoor	0.36 m	requires internet access
RTK-GPS	outdoor	0.27 m	requires internet access
Fusion of GPS, Optical Flow, IMU	outdoor	$\approx$ 0.5 m	
Camera: top view	indoor	3 mm	CX-10

Table 5.2: Localization technologies tested within this work

considerations a reasonable margin to claim is a factor of 10 or 20. For the Hummingbird, a bandwidth in the range of 0.5 Hz has been estimated for the translational dynamics [Kunkel, 2016], such that a sampling frequency of 5 to 10 Hz is required. For the CX-10 a slightly higher bandwidth is expected.

The localization attempts examined in the framework of this thesis can be divided into three different technological approaches: The first one, which continues the work previously done in the Institute of Control Systems, uses the widely used Global Positioning System (GPS) for absolute localization and aims at increasing the accuracy by applying correction techniques. The second approach extends GPS-based localization by data fusion with accelerometer data and velocity measurements from an optical flow sensor. A third and separate technology aims at indoor localization and utilizes a computer vision technique based on one or two stationary cameras. The goal of the presented research is to test the mentioned technologies for the localization of quad-rotor helicopters, determine the achievable performance and assess the suitability for formation flight experiments, which allow a validation of the theoretical concepts of this thesis. In the following, the examined localization techniques are briefly introduced. An overview is given in Table 5.2 with typical precision Distance Root Mean Square (DRMS) values for a series of static measurements.

GNSS-Based Localization

In their original configuration, the Hummingbird quad-copters are equipped with a *u-blox* LEA-6T GPS receiver module, which directly provides a GNSS position fix to the HLP. For this configuration, stationary experiments yielded a localization precision in the range of 1.46 m [Kahlefendt, 2015]. In order to improve the accuracy, in the scope of this thesis experiments were carried out using the correction techniques Precise Point Positioning (PPP) and Real-Time Kinematics (RTK). For these experiments, a u-blox NEO-6P receiver module was used, which is able to output raw data and to carry out the PPP calculations internally. Without corrections, this receiver provides a precision of approx. 1 m.

The functional principle of PPP is based on carrier phase measurements and internal models of disturbance mechanisms such as satellite antenna offset, earth tides, ephemeris

[5]Distance Root Mean Square (DRMS)

errors and atmospheric effects. These models are used to estimate the error caused by the particular effect and to correct the position result accordingly. To achieve a significant benefit, it is necessary to know the time-varying parameters of the disturbance models, which are provided by external services. Data about satellite clock and ephemeris errors as well as ionospheric delays are provided via geo-stationary satellites by Satellite Based Augmentation System (SBAS), such as the European Geostationary Navigation Overlay Service (EGNOS). Many GNSS receivers are able to improve the achievable accuracy by using SBAS correction data without implementing the full PPP algorithm.

In our experiments, a considerable benefit was observed for using the PPP algorithm of the NEO-6P based on the EGNOS-SBAS data, which lead to a precision of approx. 0.8 m. This result was further improved by performing external corrections using the software RTKLIB [6] executed on a PC and using correction data of the International GNSS Service (IGS) obtained via internet. For this configuration, a precision of 36 cm was achieved [Schalk, 2013]. As an alternative to PPP, RTK-GPS was tested by Schalk [2013] and Kahlefendt [2015]. RTK-GPS is an extension of differential GPS in combination with carrier phase measurement. Accordingly, correction data need to be provided by a base station. In this work, correction data from the IGS station WARN1 (Warnemünde, Germany) and the commercial SAPOS [7] service were used. In static experiments, this method lead to the best achieved precision of 27 cm.

Both techniques providing precision values below 0.5 m depend on correction data sourced via internet and computations which are not performed by the receiver module itself. To allow usage of these techniques on board of the quad-copter, a significant extension of the hardware setup had to be implemented [Kahlefendt, 2015]: As computation unit, a type B Raspberry Pi single-board computer was added to the Hummingbird quad-copter and connected to the GPS receiver module by its serial interface. On this computer, an ArchLinux operating system and RTKLIB were installed and used to execute the RTK and PPP computations. The weight and power consumption of the Raspberry Pi are sufficiently small to allow a normal flight of the quad-copter with the Raspberry Pi attached to it and supplied by the on-board 5 V power supply. At the same time, it provides sufficient computational power to perform the corrections. Although the Raspberry Pi B does not contain a radio communication interface itself[8], it is equipped with USB ports allowing the usage of commonly available WiFi or mobile radio USB modules to receive correction data via internet. In the scope of Kahlefendt [2015], a WiFi module was used and connected to the campus WiFi network of TUHH.

In the experiments, both techniques (RTK and PPP) provided a significant improvement of the achievable accuracy. However, the quality was found to highly depend on the circumstances, especially shadowing effects of large buildings or other obstacles are able to severely degrade the accuracy (urban canyon effect). In addition, providing correction data from external sources to a flying quad-copter is challenging and the usage of WiFi

[6] RTKLIB is an open source program package for GNSS positioning, download and further information are available at `http://www.rtklib.com/`

[7] Satellitenpositionierungsdienst der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland, `www.sapos.de`

[8] More recent versions such as Raspberry Pi 3 are equipped with an on-board WiFi interface.

turned out to be highly prone to connection failures. These facts impede the usage of corrected GNSS as high-precision localization technique of unmanned aerial vehicles. An additional drawback of GNSS-based technologies is the low sampling frequency of 1 to 5 Hz, which on its own is critical with respect to capturing the motion of a quad-copter.

Inertial and Optical Flow Measurements, Data Fusion

Besides the application of correction techniques to the GNSS localization itself, a commonly used technique to improve the localization quality is multi-sensor fusion using further sensor data such as inertial measurements.

The Hummingbird quad-copter is equipped with an IMU containing a 3D accelerometer, which provides 3D acceleration data. Although theoretically the position is obtained simply by double integration of the acceleration measurement, in practical applications the noise corrupting the acceleration signal leads to a drift of the position result and thus makes it unusable. Nevertheless, acceleration data are very useful to reduce the noise of direct position measurements by data fusion. A commonly used fusion technique for such cases is the Kalman filter [Wendel, 2011]. Besides an improvement of the precision, this method also allows an increase of the position sampling rate by using model-based interpolation and acceleration data available at higher frequency.

An additional promising sensor technology in this context is *Optical Flow* sensing, a computer-vision technique allowing to directly measure the velocity with respect to the ground. It uses an image sequence from a vertically oriented camera mounted on the quad-copter. The evaluation of the brightness pattern of this sequence provides a 2D motion field on the image plane. If the attitude and the altitude over ground are known from other sources such as gyro sensors and ultrasonic or laser-based distance sensors, the 2D motion field can be translated into a 3D motion of the camera with respect to the ground. The main requirements for this method to be effective are a sufficiently structured environment along both horizontal dimensions and a sufficiently strong and homogeneous illumination of the environment.

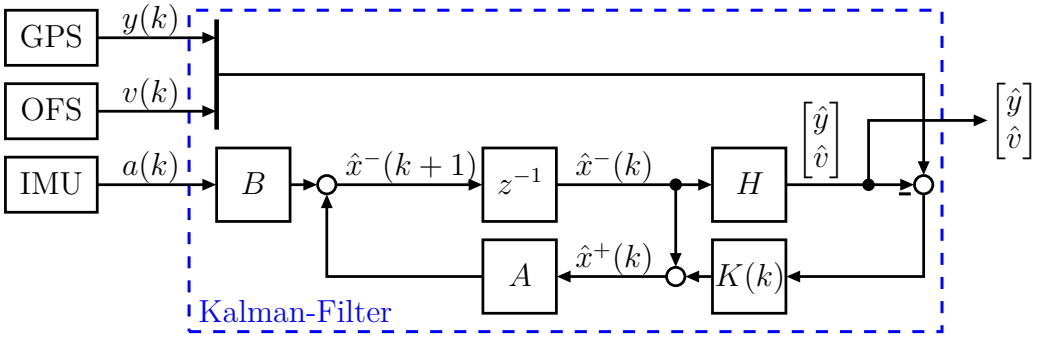


Figure 5.1.6: Kalman filter for fusion of GPS, optical flow (OFS) and IMU data

In the scope of this thesis, initial experiments were conducted with a multi-sensor fusion setup based on the data of IMU, GPS and optical flow sensors installed on a Hummingbird quad-copter [Laß, 2016; Konda, 2016]. The original on-board IMU and GPS sensors were

used in combination with a PX4Flow [9] optical flow sensor (OFS) module. This module is equipped with camera, gyroscope, ultrasonic distance sensor and an image processing unit for providing readily processed velocity data. Figure 5.1.6 shows the implemented architecture of the Kalman filter used for data fusion. It is based on a state space model of the quad-copter dynamics with system matrices A, B and H, where H is an augmented output matrix for the augmented output vector $y_{\text{aug}} = [y^T \; v^T]^T$ with the velocity $v = \dot{y}$ as additional output. The Kalman gain K is dynamically updated as

$$K(k) = P^-(k)H^T(HP^-(k)H^T + R)^{-1} \quad \text{with} \tag{5.1.3}$$

$$P^+(k) = (I - K(k)H)P^-(k), \tag{5.1.4}$$

$$P^-(k+1) = AP^+(k)A^T + Q. \tag{5.1.5}$$

The matrices Q and R are covariance matrices of the system noise and of the measurement noise, which are used to tune the filter.

Figure 5.1.7: Position estimations for rectangle trajectory [Konda, 2016]

Results obtained from the data fusion approach are illustrated in Fig. 5.1.7 for a scenario examined in Konda [2016], in which the quad-copter was manually moved along a rectangular path. A quantitative statement about the accuracy is not given due to the lack of a sufficiently accurate ground truth, but a qualitative statement can be clearly identified. The position trajectory resulting from integrating the optical flow results shows a significant drift, while the GPS trajectory is significantly subject to noise, as visible by the

[9] Part of the *Pixhawk* framework, see `www.pixhawk.org` for details

non-smooth course. Furthermore, along the south-east edge the GPS trajectory clearly deviates from a straight line, which is ascribed to shadowing effects of a tree in this area. The filtered trajectory clearly reduces both disturbing effects of the GPS measurements and produces an almost rectangular trajectory.

The present results, together with the fact that no external services or infrastructure beyond regular GNSS are required, indicate this approach to be most promising for outdoor field experiments. Further improvements can be expected from a dynamic adjustment of the covariance matrices using the positioning quality indicator provided by GPS and the optical flow sensor, which would allow the Kalman filter to adapt to the varying quality of input data depending on the environmental conditions.

Camera-Based Localization

Besides the previously described localization techniques using on-board sensors, an external camera-based localization approach was implemented and tested in the context of this work. This technique allows a localization of multiple quad-copters for indoor lab experiments at a significantly higher precision compared to GPS-based localization. The localization system employed in this thesis is an OpenCV-based design and has been developed at the institute as part of multiple student projects. An initial version has been developed in the scope of Milhan [2015] for 2D-localization of autonomous ground vehicles. An extension to 3D-localization and usage with Hummingbird quad-rotor helicopters has been done by Steinmetz [2015] and Elsner [2015]. Further extensions are the adaptation to CX-10 small-size quad-copters [Berghoff, 2016] and a multi-camera setup [Murthy, 2017].

The camera-based localization setup using a single camera is illustrated in Figure 5.1.8. The camera is mounted to the ceiling of the laboratory and is directed downward. In most experiments a Logitech HD Pro Webcam C920 was used. This camera is connected via USB to an image processing PC, which executes a computer vision algorithm using OpenCV functions to detect the quad-copter in the image and to determine its location. For on-board control of Hummingbird quad-copters, the localization result is transmitted to the quad-copter using the XBee wireless communication link. The employed hardware is able to provide localization results with a frequency of 30 Hz, for a small number of agents mainly limited by the frame rate of the camera. This sampling rate is sufficient with respect to the dynamics of the quad-copters to be localized.

To allow a reliable and precise localization of the quad-copter, it is equipped with characteristic markers of precisely known size, shape and mounting position. Two different marker setups were tested and are illustrated in Fig. 5.1.9. The first setup consists of four styrofoam balls with a characteristic color and was used for both Hummingbird and CX-10. Three balls are mounted in the plane of the quad-copter body forming a stretched isosceles triangle, a fourth one is mounted on a stick above the center of the quad-copter. As the mounting positions are neither co-planar nor rotation-symmetric, knowing the positions of the four markers on the camera image allows to determine the full pose of the quad-copter, i.e. the 3D localization and the 3D orientation. The second setup consists of

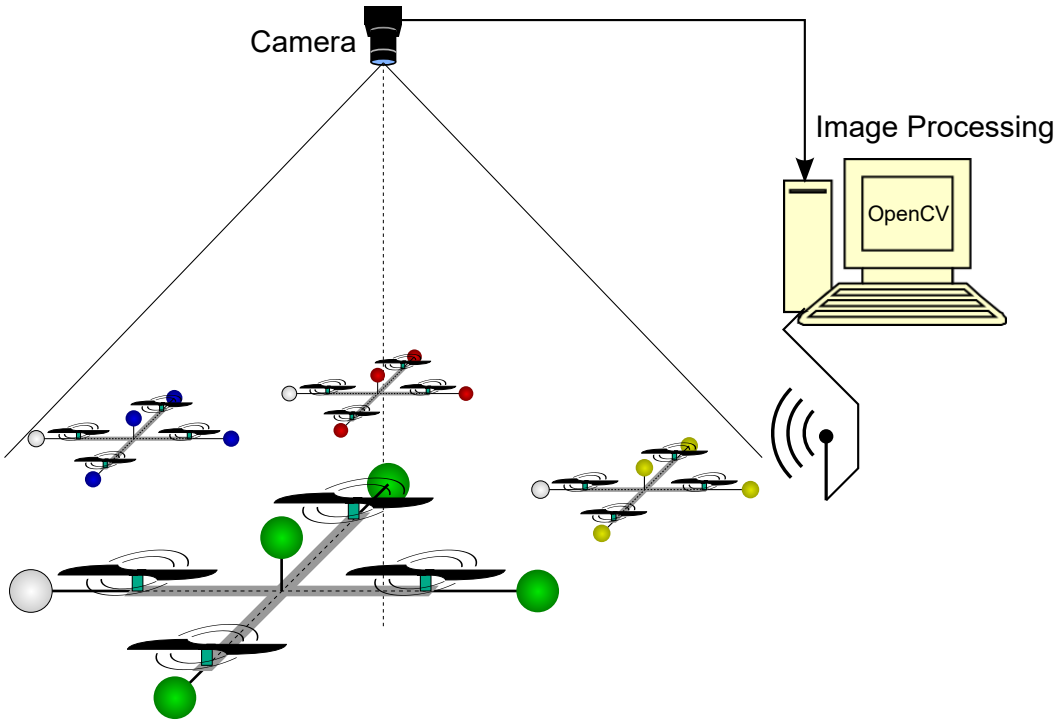
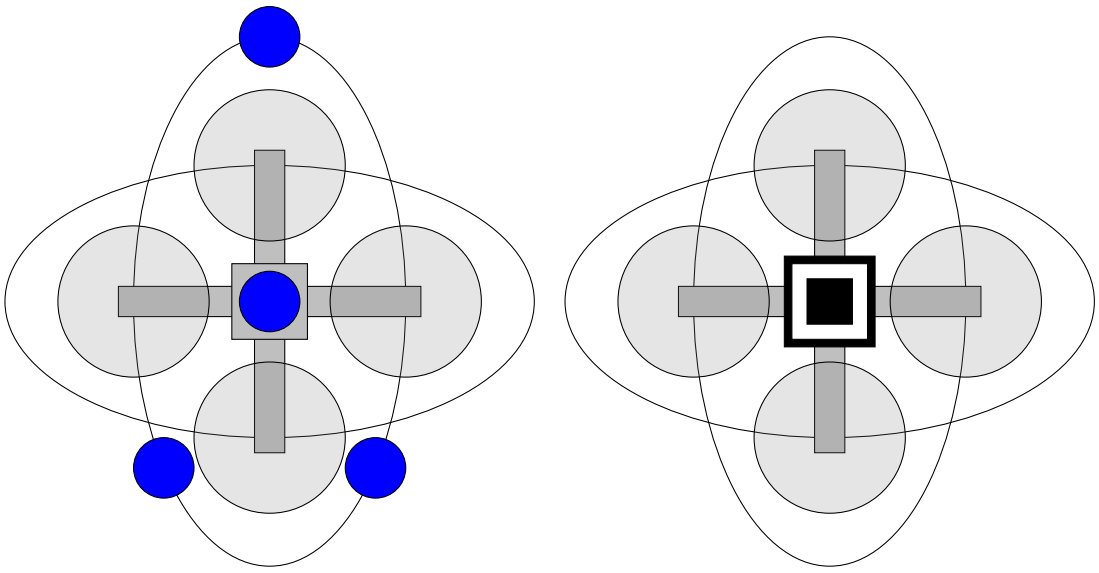


Figure 5.1.8: Schematic view of the camera-based localization setup using a single camera



(a) Blue ball markers

(b) Rectangle marker

Figure 5.1.9: Different marker setups for quad-copter localization [Kunkel, 2016]

a flat, rectangular marker plate with a black-white-black figure printed on it. This setup does not provide information about the orientation (except a 90°-ambiguous yaw angle result), but proved to be less sensitive to the ambient light conditions [Kunkel, 2016]. In most cases determining the orientation by the camera setup is dispensable, as sufficiently accurate orientation measurements are provided by the on-board IMU.

The functional principle of the camera based localization is based on finding the marker setup of a quad-copter in the camera image, determining its position and size in the image and transforming this 2D information into 3D world coordinates using knowledge about the real size and shape of the marker setup. To detect the markers, the following image processing toolchain is used:

- Blurring of the image, i.e. low-pass filtering, to reduce noise
- Conversion to greyscale (square marker) or HSV color space (ball marker)
- Threshold decision into a binary image according to brightness or color hue value
- Contour tracking to identify contours of the marker
- Polygon fitting to retrieve the shape of the marker

The height of the quad-copter can be determined from the size of the image of the marker setup. Let L_m be a length in pixels measured in one dimension of the image, where the total number of pixels is L_{image}. Then, the distance h between camera and marker is obtained as

$$h = \frac{l_m L_{\text{image}}}{2 \tan(\alpha/2) L_m}, \tag{5.1.6}$$

where l_m is the corresponding real world length of the marker (which is known) and α is the opening angle of the camera. If h, and thus the z coordinate of the quad-copter, is known, the x and y coordinates can be computed from the location of the center of the marker in the image by simple geometry. To determine the full pose using the ball markers, the location of each marker in the image is determined by computing the center of gravity of each polygon provided by the image processing. This result is used to determine the pose of the real marker setup by means of the Pose from Orthography and Scaling with ITerations (POSIT) algorithm proposed in Dementhon and Davis [1995]. Given a non-coplanar 3D object model consisting of at least 4 points, a 2D image of the object on which all model points are identified, and the camera properties (focal length and principal point), this algorithm is able to determine the pose of the object by solving the Perspective-n-Point problem.

Quad-copter	Quantity	x	y	z
Hummingbird	Mean error [mm] at $p_1 = [0\ 0\ 2]$m	0	20	2
Hummingbird	Mean error [mm] at $p_2 = [0.6\ 0.6\ 2]$m	20	50	10
Hummingbird	Standard deviation [mm]	0.3	0.2	5
CX-10	Mean error [mm] at $p_1 = [0\ 0\ 1.62]$m	-1	-2	-4.5
CX-10	Mean error [mm] at $p_2 = [0.6\ 0\ 1.62]$m	3.5	-2	-10
CX-10	Standard deviation [mm] at p_1	0.22	0.24	4.5
CX-10	Standard deviation [mm] at p_2	3	0.24	6

Table 5.3: Stationary localization error

The achievable accuracy of the camera-based localization was evaluated experimentally in the scope of Steinmetz [2015] for the Hummingbird and in Berghoff [2016] for the CX-10 quad-copter. However, due to the lack of a facility or method providing a reliable and sufficiently accurate ground-truth localization of a flying quad-copter, a meaningful evaluation could only be carried out for a stationary quad-copter. Results are shown in Table 5.3. The experiments consisted of recording a series of measurements from the camera system while the quad-copter was in rest at a position determined by mechanical measurement devices. This experiment was repeated for different positions p on the ground and for each position the mean error over time as well as the standard deviation were computed from the recorded measurements.

In Table 5.3 the mean error is shown for two different locations. At p_1, which is located on the optical axis of the camera, a rather small mean error is observed, while the mean error in locations further outside the optical axis (such as p_2) is considerably larger. Further examinations in Steinmetz [2015] indicate that the relationship between the distance from the optical axis and the measurement error can be well approximated by a polynomial, which can be used to compensate this systematic error. A second important characteristic is the standard deviation around the mean value of one measurement series, which numbers the precision. For this quantity no significant dependency of the distance to the optical axis was observed. Here, an important observation is the large difference between the height error (z value) and the sub-millimeter horizontal error (x and y direction). This observation is plausible with respect to the error propagation which can be expected from determining the height based on measurements of the horizontal size and position of the marker.

Further improvement of the accuracy of the camera-based localization system is expected from adding a side-view camera. In such a setup, the 3D position of a ball marker can be determined by triangulation methods such as described in Hartley and Zisserman [2003]. This promises significantly better error propagation properties and thus a higher accuracy. First experiments with a two-camera setup in the scope of this thesis support this expectation [Murthy, 2017].

As visible in Table 5.2, the accuracy achieved with the camera-based localization system is clearly superior compared to the other techniques tested in this work. The theoretical research presented in this thesis focuses the control schemes governing the interaction level of the agents rather than quad-copter flight control. For experimental research corresponding to this focus, the localization accuracy can be considered more important than the ability of the implemented control scheme to operate under real field conditions. In this context, the results given in this section lead to the conclusion that among the tested localization techniques the camera-based method is found most suitable for the purposes of this thesis. Thus, the following experimental results were obtained by using this method.

Local Positioning System

For future research, radio-based Local Positioning Systems (LPSs) are a promising alternative to the localization technologies described so far. In particular, the *Loco Positioning system* is provided by Bitcraze as accessory of the Crazyflie quad-copter and thus is suited to the use case of quad-copter localization. However, this system is the result of recent development and was not yet available for the work in the scope of this thesis.

Generally, LPSs use time-of-flight measurements between multiple stationary radio modules with precisely known positions (*anchors*) and a mobile radio module (*tag*) attached to each mobile robot. Currently, the Loco system can be operated in two different modes:

1. In the Two-Way Ranging (TWR) mode, the tag exchanges messages with the anchor and determines the distance to the anchor based on the round trip time. The location of the tag is calculated by triangulation based on the distances to multiple anchors.

2. In the Time Difference of Arrival (TDoA) mode, the tag uses the time difference between the arrival of packets from a master anchor and those of other anchors. Based on this, for each anchor the difference of the distances to this anchor and to the master anchor can be computed, which results in a hyperbola of possible tag locations. The tag location is determined as the intersection of the hyperbolas determined for the different anchors.

The TWR method has the drawback of requiring individual bilateral communication between a particular anchor and each tag. In case of multi-agent systems with a large number of agents, this leads to a high communication load and can critically reduce the available sampling rate. In contrast, the TDoA method relies on unilateral communication from anchor to tag, such that the capabilities of the system do not depend on the number of tags. However, the TDoA method poses the challenge of synchronization among the anchors and thus the TDoA version of Loco is still in an early stage of development.

5.2 Formation Control Experiments

Based on the experimental framework described in the previous sections, experiments were carried out in different configurations, comprising a single and multiple real agents. As important steps, the implementation comprises the modeling of the quad-copters, design of a local position controller, implementation of a communication network to enable the considered interaction as well as the synthesis and implementation of a cooperative controller.

The development of a local position controller is described in Appendix B.2 for both types of quad-copters. This comprises the realistic modeling of the quad-copter dynamics as important prerequisite to the usage of model-based controller design techniques. For this purpose, different identification experiments were conducted and models were obtained using system identification techniques (see Appendix B.2.2 for details). Based on these

models, different controller design techniques were tested: PID control, observer-based LQG and LQGI state feedback and $\mathcal{H}_\infty$ control. See Appendix B.2.4 for details.

In the following, the conducted formation control experiments are described together with the applied controller design. Results are provided and evaluated.

5.2.1 HIL Test of Extended Consensus-Based Formation Control

Experiments addressing the extended consensus-based approach were conducted in a formation control scenario of Hummingbird quad-copters. At the same time, the experiments were envisaged as proof-of-concept of the Hummingbird-based test platform for cooperative control. For the scope of this thesis, the remaining challenges concerning the accurate localization of the quad-copters did not yet allow a reasonable experimental setup for multiple Hummingbird quad-copters at the same time. Instead, for this thesis a Hardware In the Loop (HIL) setup was realized, in which a single Hummingbird quad-copter was operated in a realistically emulated cooperative control setup.

Formation Controller Synthesis and Implementation

The goal of this experiment is to implement and test the control architecture described in Section 3.2.4, of which the general setup is shown in Fig. 3.2.15. This control scheme is realized within the existing framework of the Hummingbird quad-copters by implementing an on-board software as shown in Fig. 5.1.3. While the design of the local position controllers $K_\bullet$ is covered in Appendix B.2.4, here we will focus the design of the Information Flow Filter (IFF) F.

The information flow filter design problem is formulated as robust control problem according to Section 3.2.4. A slightly modified version of the generalized plant of Fig. 3.2.16 is used here, at which the coupling error is defined as $e_C = \hat{r} - y$ (as proposed in Bartels and Werner [2014]) and the disturbance rejection transfer function T_{Pd} is neglected, i.e. assumed to be equal to identity. Following the assumption that the coupling between movements in different spatial dimensions is sufficiently suppressed by the local controllers and using the fact that a cartesian space is considered, the IFF design problem is solved separately for each dimension. The shaping filters are chosen as

$$W_S(s) = \frac{1}{M_S} \cdot \frac{\omega_S}{s + \omega_S}, \quad W_C(s) = \frac{c}{M_C} \cdot \frac{s + \omega_C}{s + c \cdot \omega_C}, \quad J(s) = \frac{200}{s + 200} \tag{5.2.1}$$

$$\begin{aligned} \text{with } & M_S = -34.5\text{dB}, \quad \omega_S = 0.01 \frac{\text{rad}}{\text{s}}, \\ & c = 1000, \\ & M_C = 120\text{dB}, \quad \omega_C = 5 \frac{\text{rad}}{\text{s}}. \end{aligned}$$

As a discrete time representation of the IFF is required for implementation and the agent model is identified in discrete time, the synthesis problem is formulated in discrete time as well, using discretized versions of the shaping filters (5.2.1).

A straight forward way to obtain the information flow filter is to solve the $\mathcal{H}_\infty/\mathcal{H}_\infty$ Problem 3.1.2 from Pilz and Werner [2012a] on the described setup with shaping filters according to (5.2.1). A well-known disadvantage of such LMI-based $\mathcal{H}_\infty$ design methods is that the structure of the controller (here the IFF) cannot be pre-specified and the controller order is fixed to the order of the generalized plant. From an implementation point of view this property can be critical, as a high controller order can exceed the computational capabilities of the hardware to be used. In case of the IFF implementation on the Hummingbird, such problems were experienced as well [Farnbacher, 2016]. As alternative method, the function *hinfstruct* of Matlab was used, which employs nonsmooth optimization techniques from Apkarian and Noll [2006] to solve the problem for a controller with given structure. Thereby, the IFF design problem was solved for a third order filter with the structural constraint of $F_{\hat{r}e}$ being strictly proper. The resulting filter reads

$$F_\bullet(z) = \left[\begin{array}{ccc|cc} 0.9987 & 0 & 0 & 0.03298 & 0 \\ 0 & 0.9584 & -0.06729 & 0 & 0.03231 \\ 0 & 0.03231 & 0.9989 & 0 & 5.37e-4 \\ \hline 2.248 & -0.4217 & -1.898 & 0 & 1 \end{array}\right]. \tag{5.2.2}$$

Experiment Setup

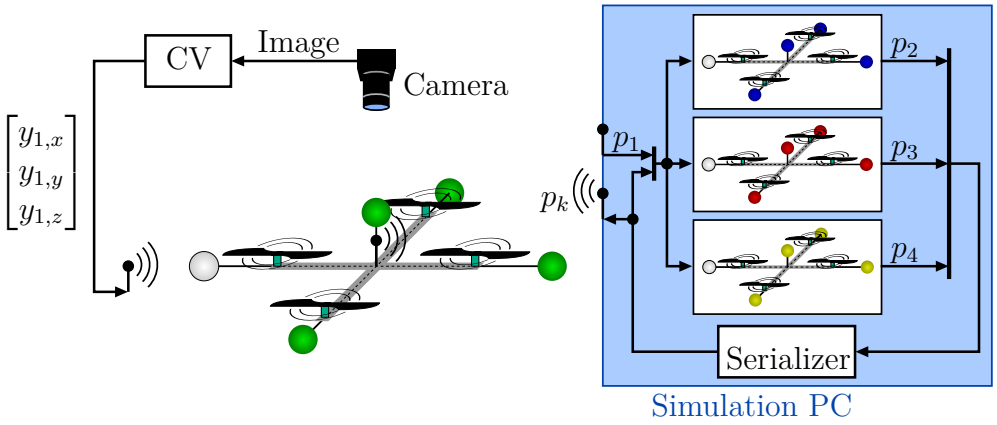


Figure 5.2.1: Setup of the hardware-in-the-loop formation control experiment using a single Hummingbird quad-copter

The experimental setup employed to test the extended consensus-based formation control scheme consists of a single Hummingbird quad-copter, the Computer Vision (CV)-based localization system described in Section 5.1.3 and a simulation PC. This setup is illustrated in Fig. 5.2.1 and consists of two parts: The localization system determines the 3D position of the real quad-copter and transmits this position data to the quad-copter by the wireless XBee link.

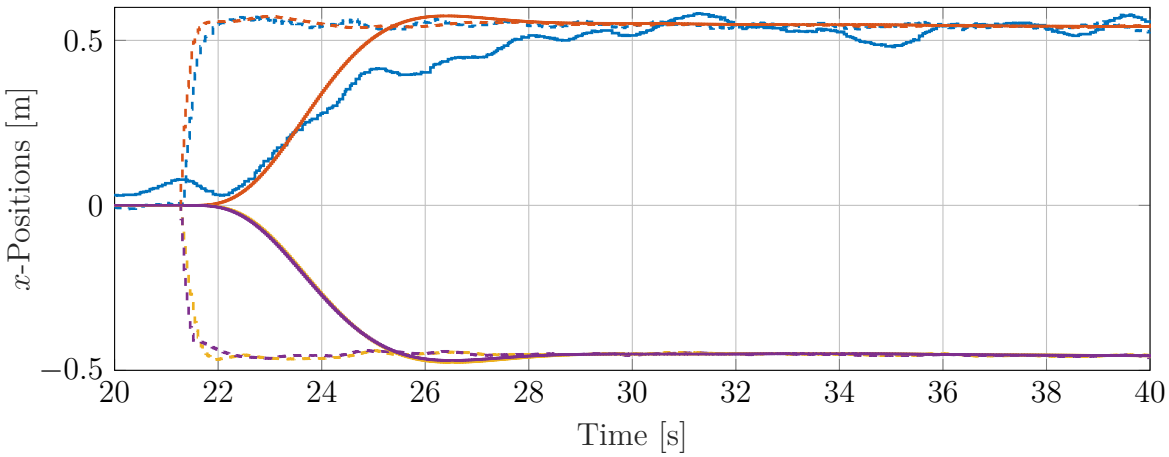


Figure 5.2.2: Agent positions in x-direction (solid) and estimated reference $\hat{r}$ (dashed) for test flight 1

The second part is the virtual MAS, which is realized by means of the simulation PC. On this PC, a semi-real-time Simulink model is executed and simulates three further quad-copters. Each virtual agent consists of a dynamic model of the Hummingbird quad-copter and replicates the on-board control scheme implemented on the real Hummingbird. From the coordination output data of the virtual agents, the serializer module generates a stream of messages according to the communication protocol implemented on the real quad-copter. These messages are transmitted to the real quad-copter, but are as well fed to the virtual agents together with messages received from the real agent. The communication topology is chosen randomly and changes with a frequency of 1 Hz. The links among the virtual agents have a 30 ms time delay, but apart from that are assumed ideal. By this means, the real quad-copter is operated in a realistic replication of a swarm of identical quad-copters.

Experimental Results

In the scenario tested in this experiment, the agents start in rest at the origin and are commanded at $t = 21$s to move into a 1×1m square formation. In addition, at $t = 45$s, an output disturbance acts on the virtual Agent 4 and changes its position by 1 m.

Figures 5.2.2 and 5.2.3 show the results of this experiment in terms of the estimated reference positions $\hat{r}_i$ and the actual positions y_i of the four agents. The plotted position of the real quad-copter (Agent 1) is measured by the camera-based localization system, while the plots of the other agents show simulation results of the dynamic models. It is clearly visible that the IFF of the real quad-copter acts similar to the simulated ones and the real position actually follows the estimated reference with similar characteristics as the responses of the virtual agents. Nevertheless, the position response of the real quad-copter is subject to visibly higher fluctuations compared to the virtual ones. The reason can be assumed to be found in external influences on the real agent (such as aerodynamic effects) and noise induced by the localization system. The chosen setup in

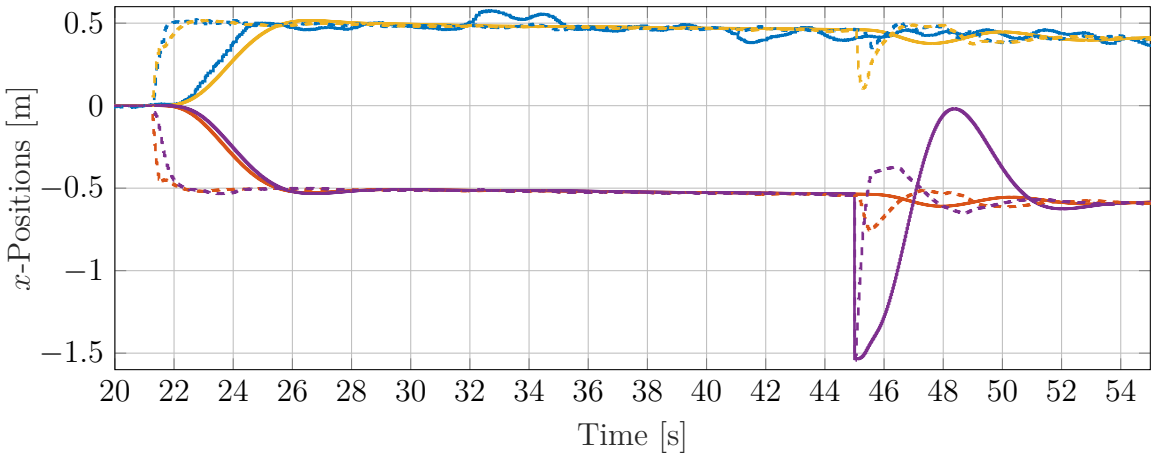


Figure 5.2.3: Agent positions in y-direction (solid) and estimated reference $\hat{r}$ (dashed) for test flight 2

this experiment, in particular the IFF design and the topology, lead to a small coupling between the disturbed Agent 4 and the real Agent 1. For this reason, the reaction of the estimated reference of Agent 1 is, although it is visible, below the noise level of the position and thus does not lead to a visible reaction in the measured position of Agent 1. Nevertheless, the presented results correspond to the expectations from simulation studies of the extended consensus-based control scheme, such as contained in Example 3.2.2.

5.2.2 Consesus-based Control of Multiple Quad-Copters

Subject of the second experiment performed in the scope of Nguyen Cong [2017] is the application of consensus-based formation control to a group of three CX-10 miniature quad-copters. Besides the validation of the consensus-based control approach, this experiment aims at a proof-of-concept of the MAS control testbed based on multiple miniature quad-copters in connection with camera-based localization.

The Information Flow Filter (IFF) has been designed by means of the optimization method, i.e. formulating the IFF design problem in analogy to the optimal control problem (according to Section 3.2.3) and solving the $\mathcal{H}_\infty/\mathcal{H}_\infty$ control problem 3.1.2 for $F(z)$. The generalized plant has been constructed according to Fig. 3.2.10, but with $p_i = \eta_i$ as coordination output instead of $\hat{r}_i$. To achieve good implementation properties, the synthesis problem has been formulated in 1D and solved for a 2nd-order IFF using the *hinfstruct* solver of Matlab. For $T_s = 25$ ms the result reads

$$F(z) = \frac{0.7462z - 0.7398}{z^2 - 1.991z + 0.9913} = \left[\begin{array}{cc|c} 1.9913 & -0.9913 & 1 \\ 1 & 0 & 0 \\ \hline 0.7462 & -0.7398 & 0 \end{array}\right]. \tag{5.2.3}$$

The setup of this experiment is a simple implementation of the consensus-based control scheme of Pilz et al. [2011] shown in Fig. 3.2.9. The position control loop was realized

by means of the camera-based localization system as shown in Fig. 5.1.4 of Section 5.1.1. For this purpose, the computer vision software is extended by multiple instances of the position controller $K_{pos}(z)$, realized as discrete-time state space model, once for each real quad-copter. The consensus loop is realized as a global discrete-time closed-loop state space model

$$\begin{aligned} x(k+1) &= (\hat{A}_F - \hat{B}_F \mathcal{L}_{(m)} \hat{C}_F) x(k) + \hat{B}_F \mathcal{L}_{(m)} r(k) \\ \hat{r}(k) &= \hat{C}_F x(k) \\ \text{for } F(z) &= \left[\begin{array}{c|c} A_F & B_F \\ \hline C_F & 0 \end{array}\right] \text{ and } \hat{A}_F = I_N \otimes A_F, \ldots \end{aligned} \tag{5.2.4}$$

for a given Laplacian matrix $\mathcal{L}$ and the information flow filter $F(z)$ of (5.2.3). Two instances of this dynamic system model (one for each spatial dimension x and y) are implemented as further extension of the computer vision program, while the height is controlled to track a common fixed value. Thus, all elements of the distributed control scheme of Fig. 3.2.9 except for the attitude control loop as inner part of the position control loop are realized centrally on the computer vision PC. This realization is necessary due to the limitations of the on-board hardware of the CX-10 quad-copters. Nevertheless, due to the block-diagonal structure of the system matrices in (5.2.4), this implementation contains an independent instance of the IFF $F(z)$ for each agent. Thus, despite the central realization, the realized control scheme can still be considered as distributed.

As examined scenario, a leader-follower setup of a virtual leader (Agent 1) and two real agents has been chosen. Therefore, the consensus loop system (5.2.4) is implemented for an interaction matrix chosen as Laplacian

$$\mathcal{L} = \begin{bmatrix} 0 & 0 & 0 \\ -1/2 & 1 & -1/2 \\ -1/2 & -1/2 & 1 \end{bmatrix} \tag{5.2.5}$$

and $N = 3$ instances of $F(z)$. The formation is guided by assigning a value to x_1 (the first state of the IFF instance corresponding to Agent 1) such that $\hat{r}_1$ takes the desired leader position. The values of $\hat{r}_2$ and $\hat{r}_3$ are fed as reference to the position controllers controlling the real quad-copters.

For the presented test flight, the desired formation is fixed as $r_1 = [0.1\ 0]^T$ m, $r_2 = [0\ 0]^T$ m, $r_3 = [0.3\ 0.3]^T$ m. During the experiment, the position of the virtual leader is changed step-wise along the x-axis.

Figure 5.2.4 shows the resulting estimated references $\hat{r}_i$ obtained from this experiment together with the measured positions y_i of the CX-10 quad-copters acting as Agents 2 and 3. It is clearly visible that the references $\hat{r}_i$ of the follower agents precisely follow the leader position and maintain the desired formation. For the present design, the response time of the consensus loop is very short compared to the dominant time constant of the real quad-copters.

The real agents oscillate around their reference positions with deviations in the range of 10 cm, which is in the range of the leader movements. Nevertheless, it is clearly

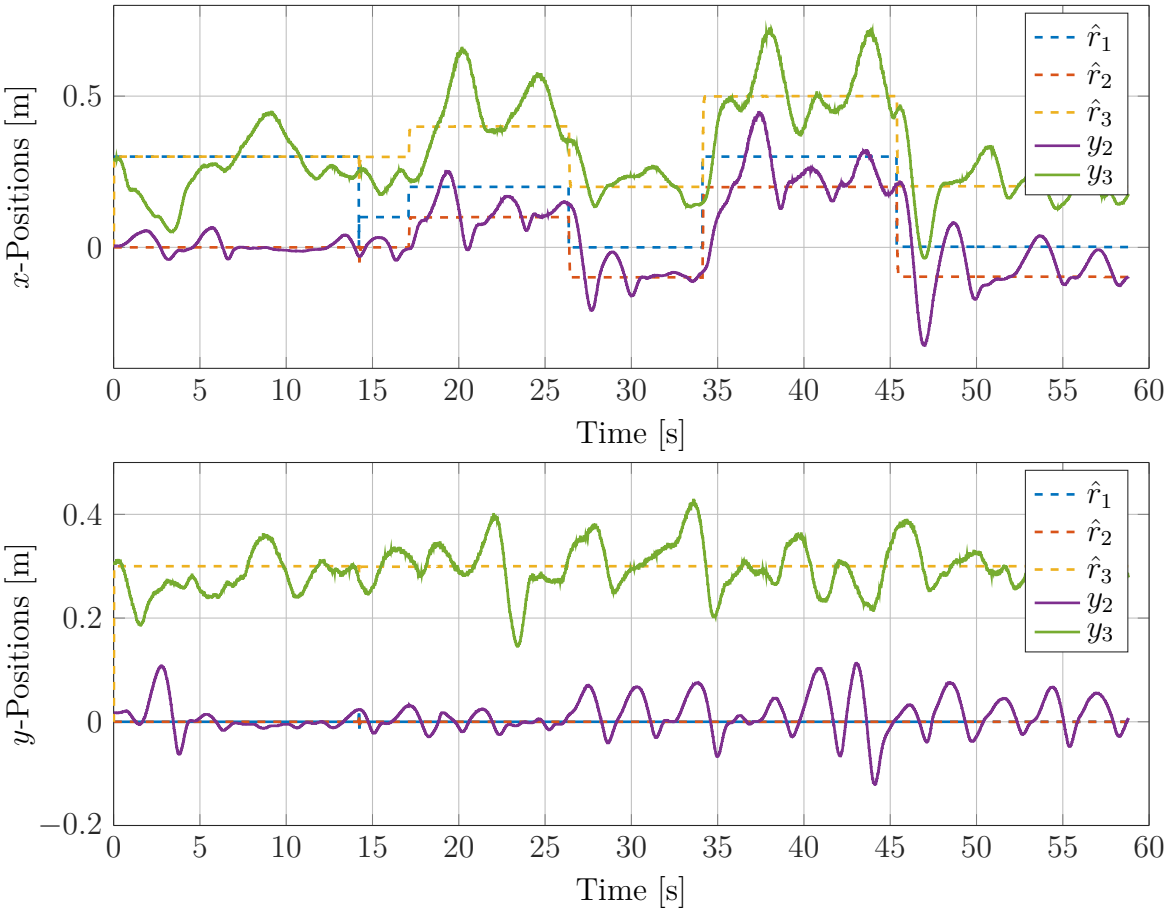


Figure 5.2.4: Agent positions (solid) and estimated reference $\hat{r}$ (dashed) for the leader movement scenario

visible that the mean value of the oscillation follows the reference value, which indicates a successful reference tracking. The main reason of the observed oscillations can be assumed to be poor tuning of the position controllers. In this experiment, manually tuned PID controllers are used. An improvement can be expected from a model-based design of the position controller (such as $\mathcal{H}_\infty$ or LQGI control) using the models of the quad-copter dynamics identified in Singh [2016].

Chapter 6

Conclusions and Outlook

6.1 Conclusions

For several years the topic of distributed control of multi-agent systems has attracted the interest of researchers; formation control of autonomous aerial vehicles is a popular example. In this context different control strategies have been developed providing robust stability of the multi-agent system even for the case of topology changes. One of the main aspects of this thesis aims at revising and comparing the existing control strategies. For this purpose simulation studies have been conducted and presented.

Among the existing control strategies, there exist both architectures considering a direct coupling by exchange of agent positions as well as consensus-based architectures, at which reference positions are negotiated. For the latter, in this thesis design objectives are worked out and illustrated by simulation results. A comparative simulation study reveals a performance benefit of the consensus-based approach, which comes at the price of a disturbance rejection drawback. Motivated by these results, in this thesis a new architecture is introduced combining the advantages of both approaches regarding performance, synthesis effort and formation maintenance. Simulation results illustrate the effectiveness of the combined approach.

For the examination of controller synthesis techniques, a unified framework has been established, in which the dynamics of the multi-agent system are modeled and performance criteria are formulated. Based on this framework, a simulation study has been conducted to compare the achievable performance of a robust control approach based on the Small Gain Theorem and an interconnection-scheduled (ISC) approach adopting LPV gain scheduling techniques. To find suitable tuning parameter values regarding a common set of objectives, a genetic algorithm has been designed and successfully applied.

From both control approaches satisfactory results regarding tracking performance and formation accuracy are achieved, while the ISC approach outperforms the robust one. This can be attributed to conservatism of the robust approach due to its guarantee of robustness against arbitrary communication delays. A simulation study including delays did not show a clear benefit of this guarantee, as both approaches led to stabilizing controllers.

An expected benefit of the ISC approach resulting from the freedom of optimizing the composition of the coordination data was not achieved. Further investigations attribute this result to an effect of the underlying synthesis technique and reproduce it in an LPV gain scheduling example. In spite of this observation, also supported by the synthesis effort observations, one can conclude that the ISC approach proves more advantageous.

As the second main finding of this thesis, a new modeling framework has been introduced using the descriptor system formulation to describe both interaction and local dynamics of a multi-agent system in a combined model. It is shown that multi-agent systems with unknown or changing topology can be efficiently modeled as affine LPV descriptor systems parameterized by the adjacency eigenvalue. For this class of systems a distributed controller synthesis approach is developed providing a controller of the same form as provided by the ISC approach. The new method is based on an existing controller synthesis approach for LPV descriptor systems and has been successfully applied to numerical examples. Simulation results show that the proposed method outperforms the compared existing ISC approach. In addition, the proposed method is demonstrated to be applicable to distributed control of interconnections of descriptor systems. This extends the scope of distributed robust control techniques to a broader class of systems not yet covered by the existing approaches.

The studies of theoretical approaches to control schemes and controller synthesis have been accompanied by experimental work aiming at an experimental validation of the theoretical findings and corresponding simulation studies. The usage of quad-copter agents as experimental platform has been adopted from previous work in this field, while new attempts have been taken especially in terms of hardware and localization techniques. This comprises both the improvement of the previously followed outdoor field testing strategy as well as the introduction of a new indoor testing concept, which partially makes use of small-scale quad-copters and Hardware In the Loop (HIL) experiments.

The localization of the agents was identified as a critical challenge, for which different technologies have been examined. For outdoor field testing, correction mechanisms for Global Navigation Satellite System (GNSS) localization have been examined as well as a multi-sensor fusion approach based on GPS, accelerometers and optical flow measurements. In both cases a significant improvement with respect to standard GPS has been reached, although GNSS correction results were found highly sensitive to environmental conditions and failures of correction data supply. As an important component of the indoor testbed, a camera-based localization system has been developed and successfully tested. The results clearly outperform the other examined technologies, although they are only available in a very limited testfield.

Using the indoor testbed, experiments on multi-agent systems were successfully conducted and proved the suitability of this experimental strategy. First results were obtained for the consensus-based control strategy and its extended version, which confirm the expectations inferred from theoretical results. However, the present experimental results do only in a limited extent provide deeper insight into particular properties of the examined approaches or the validity of the presented simulation studies. To achieve this goal, further

experiments will need to be conducted in future work. Nevertheless, the results achieved in this thesis form a significant basis for such future experiments.

6.2 Outlook

In spite of the insight gained by the work presented in this thesis, several open questions remain in the field of multi-agent systems. In this section a brief outline of these open questions is given.

The simulation studies presented in this thesis comprise a scenario with communication delays. As an interesting result, for this scenario a robust control approach providing a stability guarantee in case of communication delays is observed to be outperformed by the interconnection-scheduled control approach, which so far does not provide this formal guarantee. For further research this poses the question of finding a way to formally prove the stability of the ISC approach for systems with communication delays. In this context it is desirable to extend the existing modeling framework used for the ISC controller design to include communication delays. An analysis approach considering delays is proposed in Eichler and Werner [2015] and Mirali and Werner [2016] based on the framework of Integral Quadratic Constraints (IQCs).

Besides delays, in a real implementation further limitations of the communication links will be present. As in most cases digital technologies are used to transmit the signals, their values have to be quantized. Such a quantization always adds quantization noise to the transmitted signal, whose level depends on the resolution of the quantization. To achieve a reasonable performance in systems with strictly limited channel capacity, it is desirable to extend the existing controller synthesis approaches in order to provide robustness against quantization effects.

The approaches presented in this thesis are based on the consideration of the interaction topology being an exogenous uncertainty the control scheme has to cope with. Although robustness is an important objective with respect to reliability and safety, in many implementation cases the system designers still have strong influences on many of the properties of the communication network. This fact encourages a more holistic view on the control problems including, besides the controller design, as well the design of the communication interface with properties such as media access and the weighting and interpretation of incoming data.

Preliminary works such as Xiao and Boyd [2004], Pilz and Werner [2013] and Eichler and Werner [2014] regarding the weighting of communication links show a significant influence of link weights on the performance of the global system. This result motivates further research on techniques to find optimal weights. A first approach for multi-agent systems is presented in Pilz and Werner [2013] using a gradient-based optimization technique. For consensus schemes more general weighting strategies are proposed in Mirali and Werner [2017] based on findings from Olshevsky and Tsitsiklis [2011] and Bertrand and Moonen [2013], the application to cooperative control is examined in Mirali et al. [2017]. An approach beyond choosing the weights of existing links consists of *topology*

control strategies [Frey and Simplot, 2007; Neumann and Frey, 2015], which consider the set of actually realized communication links to be a specific selection from the set of technically feasible links. Mathematically, this approach allows not only to choose the values of the interaction matrix within a given zero-pattern, but provides a further freedom to design the interaction matrix. Especially for systems with a dense population of agents and comparatively large achievable communication ranges, such strategies are expected to be beneficial in order to avoid performance degradation resulting from a communication channel overload.

In the second part of this thesis, a new controller synthesis approach is proposed using a descriptor representation of the multi-agent system. Simulation results for this approach are promising and indicate a performance benefit in comparison with the existing ISC approach. However, so far the new approach is based on the assumption that the interaction matrix Ψ is unitarily diagonalizable. As this assumption imposes a strong restriction on the admissible topologies, for instance prohibiting directed interaction graphs such as required for leader-follower-systems, an extension of the presented approach is an open problem which needs to be solved for full applicability of the proposed approach. The decomposition technique considered in the descriptor approach is adopted from the ISC approach as presented in Hoffmann et al. [2015]. Accordingly, the ISC approach is basically limited to unitarily diagonalizable interaction matrices in the same manner. For the ISC approach this problem was solved by introducing the *Normalization* technique proposed in Hoffmann and Werner [2017]. The underlying idea is to re-formulate the system model such that the actual interaction matrix is replaced by a virtual, unitarily diagonalizable one. For the descriptor approach a similar approach is reasonable, however, the derivation of such an extended version is still an open problem.

While first experimental results have been achieved for the consensus-based control strategy, a comprehensive experimental validation of the concepts examined in this thesis is still open to future work. The recent development in the field of miniature quad-copters has brought up new solutions to the challenges identified in the present and previous work. In particular, new small-scale quad-copters such as the *Crazyflie* are available with full on-board equipment for autonomous operation. This is complemented by new radio-based localization technologies such as the *Loco positioning system*, which allow a precise localization in a medium-sized testfield. These technologies can be expected to allow new experimental insight into the considered control strategies.

Appendix A

Fundamentals and Auxiliary Content

A.1 LFT Representation

A very useful and therefore important formulation to describe interconnections of transfer elements is the Linear Fractional Transformation (LFT), which is defined e.g. in Zhou et al. [1996]. Therein, the general LFT operator is defined as:

Definition A.1.1. *Lower LFT*

$$\underbrace{\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}}_{M} * \Delta = M_{11} + M_{12}\Delta(I - M_{22}\Delta)^{-1}M_{21} = M(\Delta)$$

Definition A.1.2. *Upper LFT*

$$\Delta * \underbrace{\begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix}}_{N} = N_{22} + N_{21}\Delta(I - N_{11}\Delta)^{-1}N_{12} = N(\Delta)$$

These expressions can be interpreted as a loop interconnection of a static block Δ and a suitably partitioned matrix M or N (shown in Figure A.1.1):

$$\begin{pmatrix} z_1 \\ z_\Delta \end{pmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{pmatrix} w_1 \\ w_\Delta \end{pmatrix}; \quad \begin{pmatrix} z_\Delta \\ z_1 \end{pmatrix} = \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix} \begin{pmatrix} w_\Delta \\ w_1 \end{pmatrix} \tag{A.1.1}$$

$$w_\Delta = \Delta \cdot z_\Delta \tag{A.1.2}$$

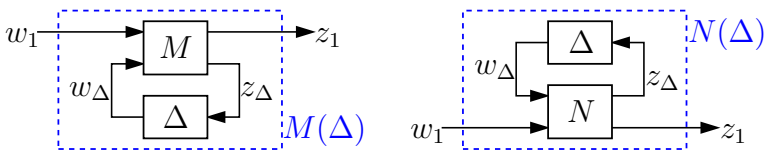


Figure A.1.1: Lower (left) and Upper (right) LFT interconnection

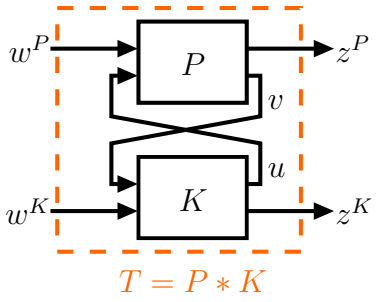


Figure A.1.2: Interconnection of two systems by Redheffer star product

Redheffer Star Product The Redheffer star product can be seen as an extension the previously described LFT representation to describe interconnections of two transfer matrices. For the interconnection shown in Fig. A.1.2, it is defined in Zhou et al. [1996] as

Definition A.1.3. *For two compatibly partitioned matrices*

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}, \quad K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix},$$

for which $P_{22}K_{11}$ is well defined and square and $I - P_{22}K_{11}$ is invertible, the star product is defined as

$$P * K = \begin{bmatrix} P * K_{11} & P_{12}(I - K_{11}P_{22})^{-1}K_{12} \\ K_{21}(I - P_{22}K_{11})^{-1}P_{21} & P_{22} * K \end{bmatrix} \tag{A.1.3}$$

This definition for matrices P and K can be extended to a corresponding definition for dynamic systems $P(s)$ and $K(s)$ in state space representation. The star product of the dynamic systems

$$P : \begin{pmatrix} \dot{x} \\ z^P \\ v \end{pmatrix} = \left[\begin{array}{c|cc} A & B_w & B_u \\ \hline C_z & D_{zw} & D_{zu} \\ C_v & D_{vw} & D_{vu} \end{array}\right] \begin{pmatrix} x \\ w^P \\ u \end{pmatrix} \text{ and}$$

$$K : \begin{pmatrix} \dot{x}^K \\ u \\ z^K \end{pmatrix} = \left[\begin{array}{c|cc} A^K & B_v^K & B_w^K \\ \hline C_u^K & D_{uv}^K & D_{uw}^K \\ C_z^K & D_{zv}^K & D_{zw}^K \end{array}\right] \begin{pmatrix} x^K \\ v \\ w^K \end{pmatrix},$$

yields a state space representation of the connected system as

$$P * K : \begin{pmatrix} \dot{\bar{x}} \\ z^{\bar{P}} \\ z^K \end{pmatrix} = \left[\begin{array}{c|cc} \bar{A} & \bar{B}_1 & \bar{B}_2 \\ \hline \bar{C}_1 & \bar{D}_{11} & \bar{D}_{12} \\ \bar{C}_2 & \bar{D}_{21} & \bar{D}_{22} \end{array}\right] \begin{pmatrix} \bar{x} \\ w^{\bar{P}} \\ w^K \end{pmatrix}. \tag{A.1.4}$$

Details of this representation are given in Zhou et al. [1996].

A.2 Linear Parameter-Varying (LPV) Control

Linear Parameter-Varying (LPV) control is a method used to control nonlinear systems like those appearing in flight control [Apkarian et al., 1995], robotics [Hashemi et al., 2009] or vehicle control [Kaiser, 2014]. The basic idea of LPV control is to describe a nonlinear and/or time-varying system by a system model which has the form of a linear system (usually a state space model), but with coefficients functionally depending on (possibly) time-varying parameters. This parameter variation expresses the change of the system dynamics with time and/or operating point. Such systems are accordingly called LPV systems. This concept was introduced in Shamma [1988], since then a large number of contributions such as Apkarian and Gahinet [1995]; Wu [1995]; Scherer [2000]; Dettori [2001] and Wu and Dong [2006] have made it become a powerful control technique. A mayor advantage of LPV control is the applicability of many concepts developed and well-known for linear systems, such as H_∞ controller design.

Typically an LPV system is described as state space model [Apkarian et al., 1995; Gahinet et al., 1995]

$$G(\theta) := \begin{cases} \dot{x} &= A(\theta)x + B(\theta)u \\ y &= C(\theta)x + D(\theta)u \end{cases} \tag{A.2.1}$$

with input signal vector $u(t)$ and output signal vector $y(t)$. The state space matrices are functions of the parameter vector $\theta(t) = \begin{bmatrix}\theta_1 & \dots & \theta_p\end{bmatrix}^T$ ranging in a predefined parameter set $\mathcal{P} \subset \mathbb{R}^p$. A special case of this representation are affine LPV systems, characterized by affine parameter dependence of the state space matrices [Gahinet et al., 1995]:

$$M = \begin{bmatrix} A(\theta) & B(\theta) \\ C(\theta) & D(\theta) \end{bmatrix} = \begin{bmatrix} A_0 & B_0 \\ C_0 & D_0 \end{bmatrix} + \sum_{i=1}^{p} \theta_i \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix} \tag{A.2.2}$$

A wider range of LPV systems, including affine LPV systems, can be described in an LFT representation. This is defined as LFT interconnection of an LTI system describing the nominal system dynamics and a parameter block Δ representing the parameter-dependency [Genc, 2002]. For the lower LFT the system equations read

$$\begin{pmatrix} \dot{x} \\ y \\ z_\Delta \end{pmatrix} = \left[\begin{array}{cc:c} A_0 & B_0 & B_\Delta \\ C_0 & D_0 & D_{y\Delta} \\ \hdashline C_\Delta & D_{\Delta u} & D_\Delta \end{array}\right] \begin{pmatrix} x \\ u \\ w_\Delta \end{pmatrix} \tag{A.2.3}$$

$$w_\Delta = \Delta(\theta) \cdot z_\Delta \tag{A.2.4}$$

Applying Def. A.1.1 to the system block matrix M of the nominal system (A.2.3), one obtains the system block matrix of the general parameter-dependent LPV representation (A.2.1) as $M * \Delta$. The LFT representation of an affine LPV system can be obtained by decomposing the summand matrices of the affine representation (A.2.2) [Genc, 2002]:

$$\tilde{P}_i = \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix} = \begin{bmatrix} L_i \\ W_i \end{bmatrix} \begin{bmatrix} R_i & Z_i \end{bmatrix} \quad \text{with } R_i \in \mathbb{R}^{q_i \times n},\ \operatorname{rank}(\tilde{P}_i) = q_i \tag{A.2.5}$$

$$\Rightarrow \theta_i \tilde{P}_i = \begin{bmatrix} L_i \\ W_i \end{bmatrix} \begin{bmatrix} \theta_i I_{q_i} \end{bmatrix} \begin{bmatrix} R_i & Z_i \end{bmatrix} \tag{A.2.6}$$

$$\Rightarrow M(\Delta) = \begin{bmatrix} A_0 & B_0 \\ C_0 & D_0 \end{bmatrix} + \begin{bmatrix} L_1 & \cdots & L_p \\ W_1 & \cdots & W_p \end{bmatrix} \begin{bmatrix} \theta_1 I_{q_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \theta_p I_{q_p} \end{bmatrix} \begin{bmatrix} R_1 & Z_1 \\ \vdots & \vdots \\ R_p & Z_p \end{bmatrix} \tag{A.2.7}$$

$$= \left[\begin{array}{cc:ccc} A_0 & B_0 & L_1 & \cdots & L_p \\ \hdashline C_0 & D_0 & W_1 & \cdots & W_p \\ \hdashline R_1 & Z_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_p & Z_p & 0 & \cdots & 0 \end{array}\right] * \begin{bmatrix} \theta_1 I_{q_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \theta_p I_{q_p} \end{bmatrix}$$

A.2.1 Bounded Real Lemma

Analyzing an LPV system $G(\theta)$ of the form (A.2.1) depending on a varying parameter vector θ, an important question to assess is whether this system is (asymptotically) stable for all parameter trajectories within a bounded parameter set Θ. In addition to stability, for optimal controller design also the question is posed whether the relation between input u and output y fulfills a performance condition. This condition is often formulated as the induced $\mathcal{L}_2$ gain (see Appendix C) being bounded by a fixed value γ for all parameter values θ. A sufficient condition for both criteria being fulfilled is given by the *Bounded Real Lemma* in terms of an LMI condition:

Theorem A.2.1. ***Bounded Real Lemma*** *Scherer [2000] The system $G(\theta)$ given in (A.2.1) is asymptotically stable for all parameter values θ and has a bounded induced $\mathcal{L}_2$ gain $\|G(\theta)\|_{\mathcal{L}_2} < \gamma$, if there exists a symmetric Lyapunov matrix $X > 0$ fulfilling*

$$\begin{bmatrix} * \\ * \end{bmatrix}^T \left[\begin{array}{cc:cc} 0 & X & 0 & 0 \\ X & 0 & 0 & 0 \\ \hdashline 0 & 0 & \frac{1}{\gamma} I & 0 \\ 0 & 0 & 0 & -\gamma I \end{array}\right] \left[\begin{array}{cc} A(\theta) & B(\theta) \\ I & 0 \\ \hdashline C(\theta) & D(\theta) \\ 0 & I \end{array}\right] < 0 \quad \forall \theta \tag{A.2.8}$$

For this theorem a dual version can be formulated in terms of the inverse of the Lyapunov matrix, i.e. in terms of $Y = X^{-1}$:

Theorem A.2.2. ***Dual Bounded Real Lemma*** *Dettori [2001] The system $G(\theta)$ given in (A.2.1) is asymptotically stable for all parameter values θ and has a bounded induced $\mathcal{L}_2$ gain $\|G(\theta)\|_{\mathcal{L}_2} < \gamma$, if there exists a symmetric dual Lyapunov matrix $Y > 0$ fulfilling*

$$\begin{bmatrix} * \\ * \end{bmatrix}^T \left[\begin{array}{cc:cc} 0 & Y & 0 & 0 \\ Y & 0 & 0 & 0 \\ \hdashline 0 & 0 & \gamma I & 0 \\ 0 & 0 & 0 & -\frac{1}{\gamma} I \end{array}\right] \left[\begin{array}{cc} I & 0 \\ -A^T(\theta) & -C^T(\theta) \\ \hdashline 0 & I \\ -B^T(\theta) & -D^T(\theta) \end{array}\right] > 0 \quad \forall \theta \tag{A.2.9}$$

As stated in Genc [2002], the expression (A.2.8) can be rewritten as

$$\begin{bmatrix} A^T(\theta)X + XA(\theta) & XB(\theta) & C^T(\theta) \\ B^T(\theta)X & -\gamma I & D^T(\theta) \\ C(\theta) & D(\theta) & -\gamma I \end{bmatrix} < 0, \tag{A.2.10}$$

the same is true for the dual version (A.2.9):

$$\begin{bmatrix} YA^T(\theta) + A(\theta)Y & YC^T(\theta) & B(\theta) \\ C(\theta)Y & -\gamma I & D(\theta) \\ B^T(\theta) & D^T(\theta) & -\gamma I \end{bmatrix} < 0 \tag{A.2.11}$$

A.3 Kronecker Calculus

The *Kronecker product* $C = A \otimes B$ is a very useful mathematical tool for describing multi-agent systems. For $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$, it is defined as

$$C = A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \in \mathbb{R}^{mp\times nq}. \tag{A.3.1}$$

Relevant properties of the Kronecker product are

$$A \otimes (B + C) = A \otimes B + A \otimes C, \tag{A.3.2a}$$
$$(A + B) \otimes C = A \otimes C + B \otimes C, \tag{A.3.2b}$$
$$(A \otimes B) \otimes C = A \otimes (B \otimes C), \tag{A.3.2c}$$
$$(\alpha A) \otimes B = A \otimes (\alpha B) = \alpha(A \otimes B) \text{ for any scalar } \alpha, \tag{A.3.2d}$$
$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \text{ if } AC \text{ and } BD \text{ defined}, \tag{A.3.2e}$$
$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1} \text{ iff } A, B \text{ invertible}, \tag{A.3.2f}$$
$$(A \otimes B)^T = A^T \otimes B^T \tag{A.3.2g}$$

A.4 Auxiliary Proofs and Derivations

A.4.1 Projection into Agreement Space

Proof of (2.1.37): From $\mathscr{A} = \text{Span}(B)$ with $B = \mathbf{1}_N \otimes I_p$, the projection matrix $P_{\mathscr{A}}$ into the agreement space $\mathscr{A}$ is given as

$$P_{\mathscr{A}} = B(B^TB)^{-1}B^T = \frac{1}{N}(\mathbf{1}_N \otimes I_p)(\mathbf{1}_N^T \otimes I_p). \tag{A.4.1}$$

Then the agreement part of x is obtained as

$$\mathbf{1} \otimes x_a = P_{\mathscr{A}}x = \mathbf{1}_N \otimes \frac{1}{N}\sum_{i=1}^{N} x_i. \tag{A.4.2}$$

A.4.2 Transfer Functions of Extended Consensus-Based Control

For the local position control loop as shown in Fig. 3.2.15, one obtains

$$y = d + \hat{P}\hat{K}(\hat{r} - y) \tag{A.4.3a}$$
$$\Rightarrow (I + \hat{P}\hat{K})y = d + \hat{P}\hat{K}\hat{r} \tag{A.4.3b}$$
$$\Rightarrow y = (I + \hat{P}\hat{K})^{-1}(d + \hat{P}\hat{K}\hat{r}) \tag{A.4.3c}$$
$$e_P = \hat{r} - y = \hat{r} - (d + \hat{P}\hat{K}e_P) \tag{A.4.3d}$$
$$\Rightarrow (I + \hat{P}\hat{K})e_P = \hat{r} - d \tag{A.4.3e}$$

Thus, the sensitivity function S_P, the disturbance sensitivity function S_{Pd} and the reference-to-output transfer function T_P are

$$\hat{S}_P = \mathrm{Tf}(\hat{r} \to e_P) = (I + \hat{P}\hat{K})^{-1} \tag{A.4.4}$$
$$\hat{S}_{Pd} = \mathrm{Tf}(d \to e_P) = -\hat{S}_P \tag{A.4.5}$$
$$\hat{T}_P = \mathrm{Tf}(\hat{r} \to y) = (I + \hat{P}\hat{K})^{-1}\hat{P}\hat{K} \tag{A.4.6}$$
$$\hat{T}_{Pd} = \mathrm{Tf}(d \to y) = \hat{S}_P. \tag{A.4.7}$$

The equality of S_P and T_{Pd} results from modeling the disturbance as output disturbance affecting the measured feedback signal.

The locally position-controlled agents are considered to be operated as a multi-agent system using an information flow filter as in Fig. 3.2.15. Then, the formation consensus signal $\hat{r}$ is computed by the information flow filter $\hat{F} = \begin{bmatrix}\hat{F}_1 & \hat{F}_2\end{bmatrix}$ as

$$\hat{r} = \hat{F}_1\mathcal{L}_{(m)}(r - \hat{r}) + \hat{F}_2 y, \quad y = \hat{T}_P\hat{r} + \hat{T}_{Pd}d \tag{A.4.8}$$
$$\Rightarrow (I + \hat{F}_1\mathcal{L}_{(m)} - \hat{F}_2\hat{T}_P)\hat{r} = \hat{F}_1\mathcal{L}_{(m)}r + \hat{F}_2\hat{T}_{Pd}d \tag{A.4.9}$$

The formation consensus error is defined as $e = \mathcal{L}_{(m)}(r - \hat{r})$. From these expressions the consensus sensitivity function S_F and other transfer functions concerning the global system can be formulated:

$$\begin{aligned} S_F = \mathrm{Tf}(r \to e) &= \mathcal{L}_{(m)}(I - (I + \hat{F}_1\mathcal{L}_{(m)} - \hat{F}_2\hat{T}_P)^{-1}\hat{F}_1\mathcal{L}_{(m)}) \\ &= \mathcal{L}_{(m)}(I + \hat{F}_1\mathcal{L}_{(m)} - \hat{F}_2\hat{T}_P)^{-1}(I - \hat{F}_2\hat{T}_P) \end{aligned} \tag{A.4.10}$$
$$S_{Fd} = \mathrm{Tf}(d \to e) = -\mathcal{L}_{(m)}(I + \hat{F}_1\mathcal{L}_{(m)} - \hat{F}_2\hat{T}_P)^{-1}\hat{F}_2\hat{T}_{Pd} \tag{A.4.11}$$
$$T_F = \mathrm{Tf}(r \to \hat{r}) = (I + \hat{F}_1\mathcal{L}_{(m)} - \hat{F}_2\hat{T}_P)^{-1}\hat{F}_1\mathcal{L}_{(m)} \tag{A.4.12}$$
$$T_{Fd} = \mathrm{Tf}(d \to \hat{r}) = (I + \hat{F}_1\mathcal{L}_{(m)} - \hat{F}_2\hat{T}_P)^{-1}\hat{F}_2\hat{T}_{Pd} \tag{A.4.13}$$

Choosing the coupling error signal as $e_C = \hat{r} - d$, the coupling sensitivity functions are obtained as

$$S_C = \mathrm{Tf}(r \to e_C) = T_F \tag{A.4.14}$$
$$S_{Cd} = \mathrm{Tf}(d \to e_C) = T_{Fd} - I = (I + \hat{F}_1\mathcal{L}_{(m)} - \hat{F}_2\hat{T}_P)^{-1}\hat{F}_2\hat{T}_{Pd} - I. \tag{A.4.15}$$

Transfer functions to the output of the global multi-agent system are obtained as

$$T_{yr} = \mathrm{Tf}(r \to y) = \hat{T}_P T_F \tag{A.4.16}$$

$$T_{yd} = \mathrm{Tf}(d \to y) = \hat{T}_P T_{Fd} + \hat{T}_{Pd} \tag{A.4.17}$$

Appendix B

Control of Quad-Rotor Helicopters

B.1 Dynamics of Quad-Rotor Helicopters

In the following, the physical fundamentals of those mobile robots are introduced, which are used as experimental testbed and example systems throughout this thesis.

As quad-rotor helicopters are a popular subject of research, various literature sources exist about their flight dynamics. In Castillo et al. [2004] and Bouabdallah et al. [2004] nonlinear models of rotational and translational dynamics are derived using the Lagrangian method, a linearized model is given in Lara et al. [2006]. Bouabdallah [2007] provides a rather comprehensive work considering results of both Euler-Lagrange and Newton-Euler formalism.

Figure B.1.1 shows the basic design of a quad-rotor helicopter and the conventions used in this work. The Position in 3-dimensional space is described by the translational coordinates x, y and z in an earth-fixed world coordinate system following the *north-east-down* (NED) convention. The orientation is expressed by the rotational coordinates ϕ (called *roll*), θ (*pitch*) and ψ (*yaw*). The body of the quad-rotor is assumed symmetric with the

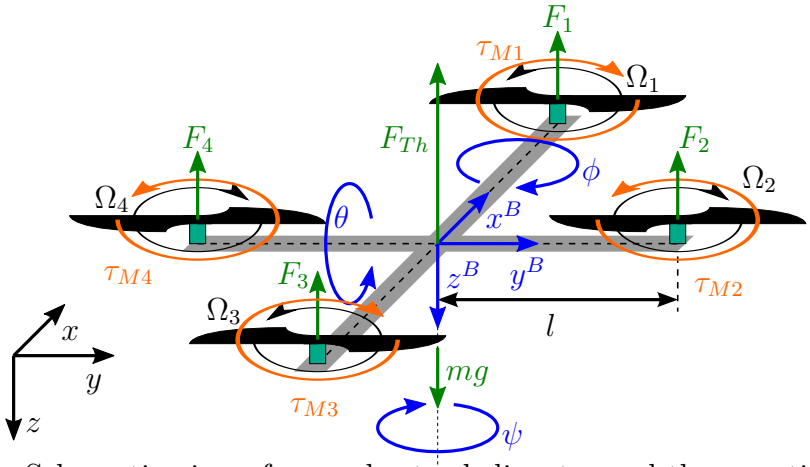


Figure B.1.1: Schematic view of a quad-rotor helicopter and the quantities describing it

center of gravity located at the center of the body. The body-fixed frame (x^B, y^B, z^B) rotates with the quad-rotor body and is used to describe quantities with respect to the body. A vehicle frame is defined as oriented along the world frame, but having its origin at the center of the quad-rotor body.

Each of the four rotors is driven by an electric motor and is assigned an individual torque. The rotors convert their rotation with angular speed Ω_i into a thrust force

$$F_i = b\Omega_i^2$$

pointing in z^B-direction perpendicular to the rotor plane (b is a thrust factor) [Bouabdallah, 2007]. In addition, each rotor exerts a reaction torque τ_{Mi} on the body opposite to its rotation. As the motor-rotor sets are fixed to the body, the rotor planes are always parallel to each other and to the plane spanned by the body. The total thrust force is thus obtained as the sum of all rotor thrust forces:

$$F_{Th} = F_1 + F_2 + F_3 + F_4$$

Opposite rotors rotate in equal direction, adjacent rotors in opposite direction. A difference in the thrust force of two opposite rotors causes a torque around the x^B- or y^B-axis, while a thrust difference of adjacent rotors causes a torque around the z^B-axis perpendicular to the frame:

$$\tau_\phi = l(F_4 - F_2) \tag{B.1.1a}$$
$$\tau_\theta = l(F_1 - F_3) \tag{B.1.1b}$$
$$\tau_\psi = \tau_{M1} + \tau_{M3} - \tau_{M2} - \tau_{M4}. \tag{B.1.1c}$$

The directions of rotation are chosen such that for equal motor torques the resulting torques acting on the body are canceled out, as visible from (B.1.1c). Using the Lagrangian method, in Bouabdallah [2007] the equations of motion for the rotational dynamics are obtained as

$$\begin{aligned} J_{xx}\ddot{\phi} &= \dot{\theta}\dot{\psi}(J_{yy} - J_{zz}) - J_r\dot{\theta}\Omega_r + \tau_\phi, \\ J_{yy}\ddot{\theta} &= \dot{\phi}\dot{\psi}(J_{zz} - J_{xx}) - J_r\dot{\phi}\Omega_r + \tau_\theta, \\ J_{zz}\ddot{\psi} &= \dot{\phi}\dot{\theta}(J_{xx} - J_{yy}) + \tau_\psi \end{aligned} \tag{B.1.2}$$

assuming a diagonal inertia matrix $J = \mathrm{diag}(J_{xx}, J_{yy}, J_{zz})$ for the quad-rotor, equal inertias J_r for the rotors and using the net rotor speed $\Omega_r = \Omega_1 + \Omega_3 - \Omega_2 - \Omega_4$.

For the translational movement the quad-rotor can be seen as a rigid body of mass m, on which a force F is acting. This force is obtained as superposition of the total thrust force acting along the body-frame axis z^B and the gravity force acting along the earth-fixed z-axis. Using the rotation matrix R_B^V mapping body to vehicle frame yields

$$F = m\begin{pmatrix}0\\0\\g\end{pmatrix} + R_B^V\begin{pmatrix}0\\0\\-F_{Th}\end{pmatrix} \tag{B.1.3}$$

Using the rotation matrix R_B^V as defined in Castillo et al. [2004] together with Newton's second law of motion, the translational equations of motion are obtained as

$$\begin{aligned} m\ddot{x} &= -\sin(\theta)F_{Th}, \\ m\ddot{y} &= \cos(\theta)\sin(\phi)F_{Th}, \\ m\ddot{z} &= -\cos(\theta)\cos(\phi)F_{Th} + mg. \end{aligned} \tag{B.1.4}$$

Linear Model For the non-linear equations of motion Lara et al. [2006] propose a method to derive a linearized state-space model, which is usable for LTI controller synthesis techniques. To simplify the rotational dynamics to be handled by a linear controller, Castillo et al. [2004] propose a non-linear control law to cancel the rotational nonlinearities. In terms of the body torques $\tau = [\tau_\psi \quad \tau_\theta \quad \tau_\phi]^T$ as control inputs it is defined as

$$\tau = C(\Theta, \dot{\Theta})\dot{\Theta} + J\tilde{\tau} \tag{B.1.5}$$

with $\tilde{\tau} = [\tilde{\tau}_\psi \quad \tilde{\tau}_\theta \quad \tilde{\tau}_\phi]^T$ as new inputs to be computed by the linear controller. The term $C(\Theta, \dot{\Theta})\dot{\Theta}$ is defined to consist of the gyroscopic and centrifugal terms of (B.1.2), such that (B.1.2) reads

$$J\ddot{\Theta} = -C(\Theta, \dot{\Theta})\dot{\Theta} + \tau \tag{B.1.6}$$

in terms of the orientation vector $\Theta = [\psi \quad \theta \quad \phi]^T$. Inserting (B.1.5) in (B.1.2) cancels the gyroscopic and centrifugal terms and leads to a linear double integrator model for the rotational dynamics in terms of the new input variables:

$$\begin{aligned} \ddot{\psi} &= \tilde{\tau}_\psi \\ \ddot{\theta} &= \tilde{\tau}_\theta \\ \ddot{\phi} &= \tilde{\tau}_\phi \end{aligned} \tag{B.1.7}$$

Following the lines of Lara et al. [2006], a non-linear state space model is formulated out of (B.1.4) and (B.1.7) by choosing the state vector $\mathbf{x}$ and the input vector u as [1]

$$\mathbf{x} = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} & \psi & \dot{\psi} & \theta & \dot{\theta} & \phi & \dot{\phi} \end{bmatrix}^T, \tag{B.1.8}$$

$$u = \begin{bmatrix} F_{Th} - mg & \tilde{\tau}_\psi & \tilde{\tau}_\theta & \tilde{\tau}_\phi \end{bmatrix}^T. \tag{B.1.9}$$

With this choice of u the translational equations of motion read

$$\begin{aligned} \ddot{x} &= -\sin(\theta)(\frac{1}{m}u_1 + g), \\ \ddot{y} &= \cos(\theta)\sin(\phi)(\frac{1}{m}u_1 + g), \\ \ddot{z} &= -\cos(\theta)\cos(\phi)\frac{1}{m}u_1. \end{aligned} \tag{B.1.10}$$

[1] Note that in Lara et al. [2006] the sign of the z coordinate is chosen differently and therefore $B_{61} = 1/m$ is stated positive. Here the z-axis is always chosen to point down, such that consistent right-handed coordinate systems are used.

The linear model is obtained by Taylor series linearization around the equilibrium point $\mathbf{x} = 0$, which leads to a state space model with

$$A = \left[\begin{array}{cccccc:cccccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right], \quad B = \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{m} & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right]. \quad \text{(B.1.11)}$$

B.2 System Identification and Local Control

B.2.1 Position Control Scheme

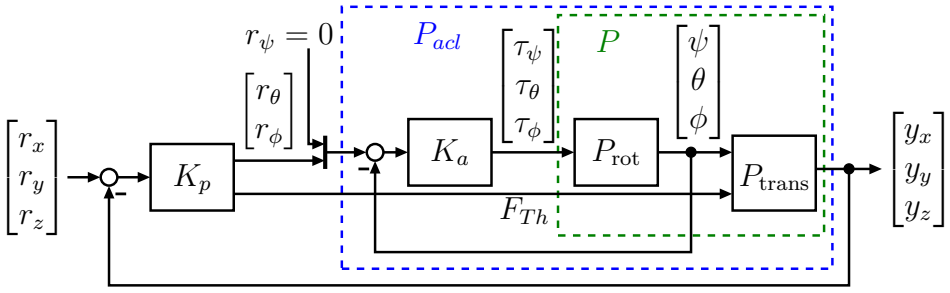


Figure B.2.1: Cascaded position control loop of a quad-copter

As an important prerequisite for the implementation of the formation control schemes examined in Section 5, a suitable position controller had to be designed and implemented for both types of quad-copters. In both cases, the cascaded control scheme shown in Fig. B.2.1 has been chosen. It consists of an inner loop controlling the attitude of the quad-copter and an outer loop controlling its position in 3D space. The plant model P describing the dynamics of the quad-copter can be separated into a rotational model P_{rot} and a translational model P_{trans}. The rotational model describes the dynamic relationship between the torques around the spatial axes and the resulting attitude angles, as described by (B.1.2). Meanwhile, as given in (B.1.4), the translational model maps the attitude angles and the total thrust force F_{Th} to the position. The rotational dynamics are controlled by an attitude controller K_a, which assigns the torques in roll, pitch and yaw direction in order to track the attitude reference. To control the position, a position controller K_p is considered to assign the references for roll, pitch and total thrust in order to track the position reference. As the quad-copter has four independent actuators, the

attitude-controlled system P_{acl} can be controlled by four inputs, i.e. r_ψ, r_θ, r_ϕ and F_{Th}. This allows controlling the position in 3D space and the yaw angle independently. In this thesis we focus on tracking a position in 3D space and for simplicity request the yaw rate to be zero, which turned out to be sufficient to compensate yaw disturbances.

Due to the cascaded structure of this control scheme, the inner controller can be designed and tested independently of the outer controllers. As the focus of this thesis is set to the level of cooperative control rather than the local control of agent dynamics and both experimental platforms were delivered with existing and sufficiently well-tuned local attitude controllers K_a, it was decided to utilize the existing attitude controllers and not to spend further effort on the design of K_a. Accordingly, in the following, the focus will be set on designing the position controller K_p and formation controllers on top of the local control scheme shown in Fig. B.2.1.

B.2.2 Identification of Agent Dynamics

The design techniques for cooperative control schemes examined in this thesis are model-based, which means that a model of the agent dynamics is required to design the controllers. Accordingly, for implementation and experimental validation on real hardware, a model of the utilized hardware needs to be derived. In the context of this thesis, multiple different modeling approaches were used for different purposes or parts of the system. While the white box model described in Appendix B.1 was used for most simulations, mostly grey box and black box models were used to design the controllers implemented for experimental research. The latter models were obtained from experimental data of both types of quad-copters used for experiments, i.e. Hummingbird [Kunkel, 2016] and CX-10 [Singh, 2016].

Grey-box modeling and identification of physical properties

As described in Section B.1, the fundamental flight dynamics of quad-copters are well known and can be described by models given in literature such as Castillo et al. [2004]; Bouabdallah et al. [2004] and Bouabdallah [2007]. However, for obtaining useful models in the context of experimental work, the physical model parameters still need to be determined for the particular quad-copter in use. While the mass m and the arm length l are determined easily, for the devices used for this thesis the inertia matrix J and the relationship between the motor inputs and the resulting thrust force F_{Th} needed to be identified experimentally, according to the grey-box approach.

Assuming J to be diagonal, the moments of inertia were identified by means of a torsional test rig, in which the quad-copter is hinged and attached to a torsional spring, such that it can perform a rotational oscillation around one axis. If the stiffness of the rotational spring is known, the moment of inertia of the quad-copter can be determined by measuring the frequency of the oscillation. For the Hummingbird quad-copter, the inertia values were determined accordingly from time measurements of the oscillation period. The spring stiffness was determined using an equivalent experiment with a test body of known

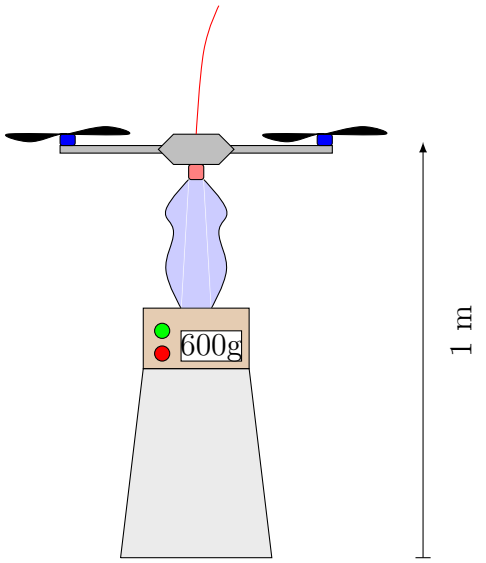


Figure B.2.2: Test rig for experimental identification of the thrust force [Kunkel, 2016]

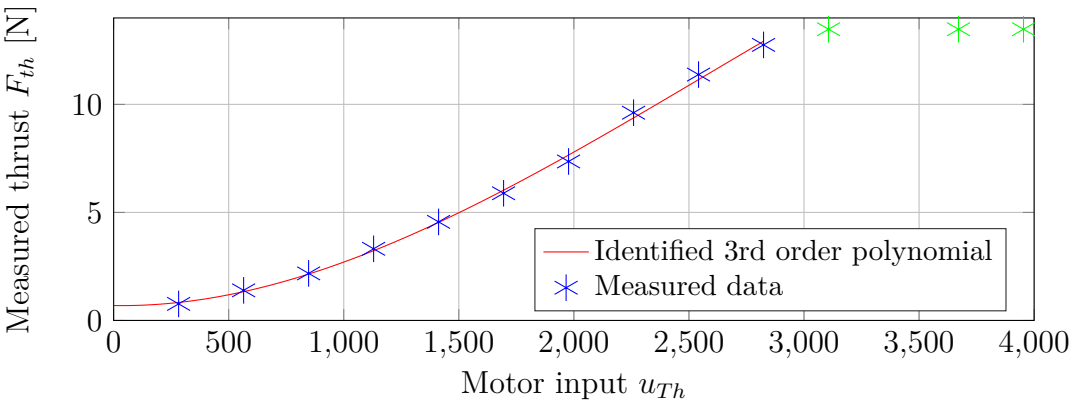


Figure B.2.3: Results for experimental identification of the thrust force [Kunkel, 2016]

inertia. Using a grey-box model based on the results of this method, simulation results for the rotational dynamics of a Hummingbird quad-copter were generated and compared to measurement data from free flight [Scheibner, 2014]. The results show a poor match, for which a failed identification of the inertia values is a plausible explanation.

The stationary relationship between motor inputs and thrust force of the Hummingbird was determined experimentally using the test rig shown in Figure B.2.2 [Scheibner, 2014; Kunkel, 2016]. The quad-copter is mounted in horizontal pose on top of a support structure weighing on an electronic balance. In order to simulate free flight conditions and to avoid aerodynamic ground effects, a thin support structure is used. The weight of the support structure is sufficiently high to prevent the quad-copter from lifting off even at high motor power. The quad-copter is operated in direct motor access mode (Mode 1), in which the LLP accepts an integer input value u_{Th} in the range [0, 4095] for each motor. Using this setup, the motor input values (all motors are commanded equal values at the same time) are increased step-wise and the resulting weight difference with respect to the

un-powered motors is measured by the balance. Figure B.2.3 shows the results of this experiment. For values below a threshold of $u_{Th} \approx 3000$, the resulting thrust increases with the input value and can be approximated by a third order polynomial. Above the threshold, the thrust force saturates at a value of approx. 14.5 N [Kunkel, 2016]. The relationship for the non-saturated range is determined as

$$F_{Th,z} \approx -3.2571 \cdot 10^{-10} u_{Th}^3 + 2.5176 \cdot 10^{-6} u_{Th}^2 - 1.809 \cdot 10^{-4} u_{Th} + 0.69 \text{ N}. \quad \text{(B.2.1)}$$

An identification of the dynamics of the motor-rotor subsystem was not possible with the described setup due to the high latency of the balance. Due to the low mass of the plastic propellers and the small motor size, the dynamics are assumed negligible.

Black-box modeling

Although properly tuned attitude controllers are available for both types of quad-copters, in both cases the manufacturer does not provide precise information about these provided attitude controllers. For the present experimental framework this means a limited knowledge of the agent dynamics. In addition, it turned out to be difficult to precisely determine all relevant physical properties of the quad-copters and the validation of corresponding grey-box models failed. For this reason, the black box method was chosen for the experimental part of this thesis. For small tilt angles, previous results indicate that a linear model describes the dynamic behavior sufficiently well and therefore in this work the system identification is limited to linear models as well.

Corresponding to the goal of designing a position controller K_p, the goal of the system identification is to obtain a model for the dynamics of the attitude-controlled quad-copter, thus of P_{acl}. For the structure shown in Fig.l B.2.1 with r_ψ fixed to zero, a 3×3 LTI model is to be identified:

$$\begin{bmatrix} y_x \\ y_y \\ y_z \end{bmatrix} = P_{acl}(s) \begin{bmatrix} r_\theta \\ r_\phi \\ u_{Th} \end{bmatrix}, \quad \text{(B.2.2)}$$

where $u_{Th} = F_{Th} - mg$ is the net thrust, i.e. the deviation from the hovering thrust. As the system under consideration already contains an internal attitude controller, it is plausible to assume that cross couplings between pitch and roll axes are sufficiently compensated by this controller for being negligible in modeling the closed-loop system. This assumption is confirmed by observations in Kunkel [2016]. Furthermore, the linearization presented in Section B.1 neglects the effect of the total thrust on the horizontal movement, which is reasonable for small values of the net thrust. These facts suggest that considering P_{acl} diagonal is a reasonable simplification, which is therefore used in this thesis. This allows to identify the dynamics in the different spatial dimensions as SISO systems in separate experiments.

As the black-box identification technique solely relies on input and output data, it is important to excite the system with a sufficient rich input signal and to capture the output signal, in this case the motion of the quad-copter, with high accuracy and a sufficient

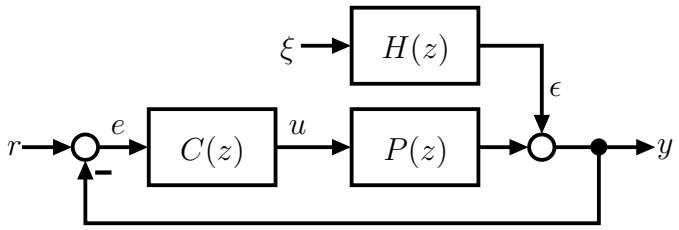


Figure B.2.4: Generic system setup for closed-loop system identification

sampling frequency. The input signal being *sufficiently rich* means exciting all relevant frequencies. To fulfill this requirement, a Pseudo-Random Binary Sequence (PRBS) is generated as input signal according to Landau and Zito [2006]. As design parameters, the bandwidth and the rise time of the system are estimated from measurements of the closed-loop step response obtained using a manually tuned Proportional-Integral-Differential (PID) position controller [Kunkel, 2016; Singh, 2016]. For capturing the output signals, in the scope of this thesis the requirements were only fulfilled by the camera-based indoor localization system. However, for this system the available flying space is strongly limited by the vision field of the camera. Especially regarding the tilt angles, this fact requires a tight limit on the assignable input amplitude in order to avoid the quad-copter leaving the area in which a localization is available. For this reason, it was decided to perform a system identification in closed-loop operation, meaning with an active position controller K_p in place.

The actual system identification has been accomplished in a two-step procedure [Kunkel, 2016; Singh, 2016]: In the first step, for each axis an initial PID position controller has been designed. In case of the Hummingbird, for this purpose an initial black-box identification has been performed in open-loop operation (i.e. without position feedback). The corresponding experiments consisted of assigning PRBS signals as input signals in terms of pitch and roll reference and net thrust and tracking the position of the quad-copter using the camera-based system. The resulting model was used to tune the PID controller [Kunkel, 2016]. In case of the CX-10, an initial PID controller was designed in a manual try-and-error process [Singh, 2016].

The second and main identification step was performed in closed loop, meaning the outer loop in Fig. B.2.1 was closed using the initial PID controller as position controller K_p. During the experiment, a PRBS signal was assigned as position reference for one particular dimension and zero for the other dimensions. This corresponds to commanding a movement along one of the spatial axes. As output, the position was measured using the camera-based localization system. In addition, the control signal $u = \begin{bmatrix} r_\theta & r_\phi & u_{Th} \end{bmatrix}^T$ produced by the position controller was recorded. This experiment was performed for all three spatial dimensions in the same manner.

Figure B.2.4 shows the generic setup in which closed-loop system identification takes place. Therein, $P(z)$ describes the real plant dynamics, here referring to the attitude-controlled agent model P_{acl}, and $C(z)$ is the controller. The plant output is corrupted by a disturbance signal ϵ, which is modeled as colored noise resulting from filtering white noise ξ by the noise filter $H(z)$. For this setup, control input and output signals can be

written as

$$y = P(z)u + H(z)\xi \tag{B.2.3}$$
$$u = T_{ur}(z)r - T_{ur}(z)H(z)\xi \tag{B.2.4}$$
$$\text{with } T_{ur}(z) = (I + C(z)P(z))^{-1}C(z)$$

For identifying the plant model $P(z)$ from the recorded closed loop data, two different approaches were tested [Kunkel, 2016; Singh, 2016]: The *Direct Approach* considers the identification problem as an open-loop problem using the recorded control signal u as input and the measured output y as output, ignoring the feedback loop. This method has the advantage of being simple, but tends to produce biased results in case of a significant amount of noise ϵ. The reason is that, due to the feedback loop, the considered input signal u depends on the measured output signal y and thus itself contains the noise contained in y. Thus, u and y are correlated and the closed-loop experiment can be non-informative [Zhu, 2001; Ljung, 1999]. As an alternative method to overcome the bias, the two-step version of the *Joint Input-Output Approach* proposed by van den Hof et al. [1992] has been tested: This method utilized the fact that the control input u from (B.2.4) consists of a part $u_r = T_{ur}(z)r$, which only depends on r and thus is not correlated with y, and a noise part colored by $H_1(z) = T_{ur}(z)H(z)$. This allows identifying the transfer function from r to u with open-loop identification methods as first step, yielding an estimate $\hat{T}_{ur}$. The second step follows from inserting (B.2.4) in (B.2.3), leading to the open-loop model structure

$$y = P(z)u_r + H_2(z)\xi; \quad H_2(z) = (I - P(z)T_{ur}(z))H(z). \tag{B.2.5}$$

According to this expression, the plant model $P(z)$ can be identified by open-loop methods using the measured output signal y and an artificial noise-free input signal $\hat{u}_r = \hat{T}_{ur}r$.

In this work, both direct and joint input-output approach were applied to experimental data of the Hummingbird quad-copter using subspace identification as identification technique, yielding 1×1 state space models for the diagonal elements of P_{acl} [Kunkel, 2016]. For the first step of the joint input-output identification, the input transfer function was identified as two different types of models, ARX and Finite Impulse Response (FIR).

Results of the closed-loop system identification of the Hummingbird quad-copter are shown in Fig. B.2.5 for the direct method and in Fig. B.2.6 for the joint input-output method. The plotted results were obtained from cross-validation experiments, for which each spatial dimension was excited subsequently by two different PRBS sequences generated with identical parameters and the position was measured as output. The measurements of the first sequence were used to identify the model. Then, a simulated response to the second input sequence was computed for the identified model. The results of this simulation are shown in Figs.B.2.5 and B.2.6 together with the measured response to the second input sequence. The resulting models are given numerically in Appendix B.2.3.

For both direct and joint input-output method, the results of horizontal movement (x and y) show a good match with slightly better results for the joint input-output method.

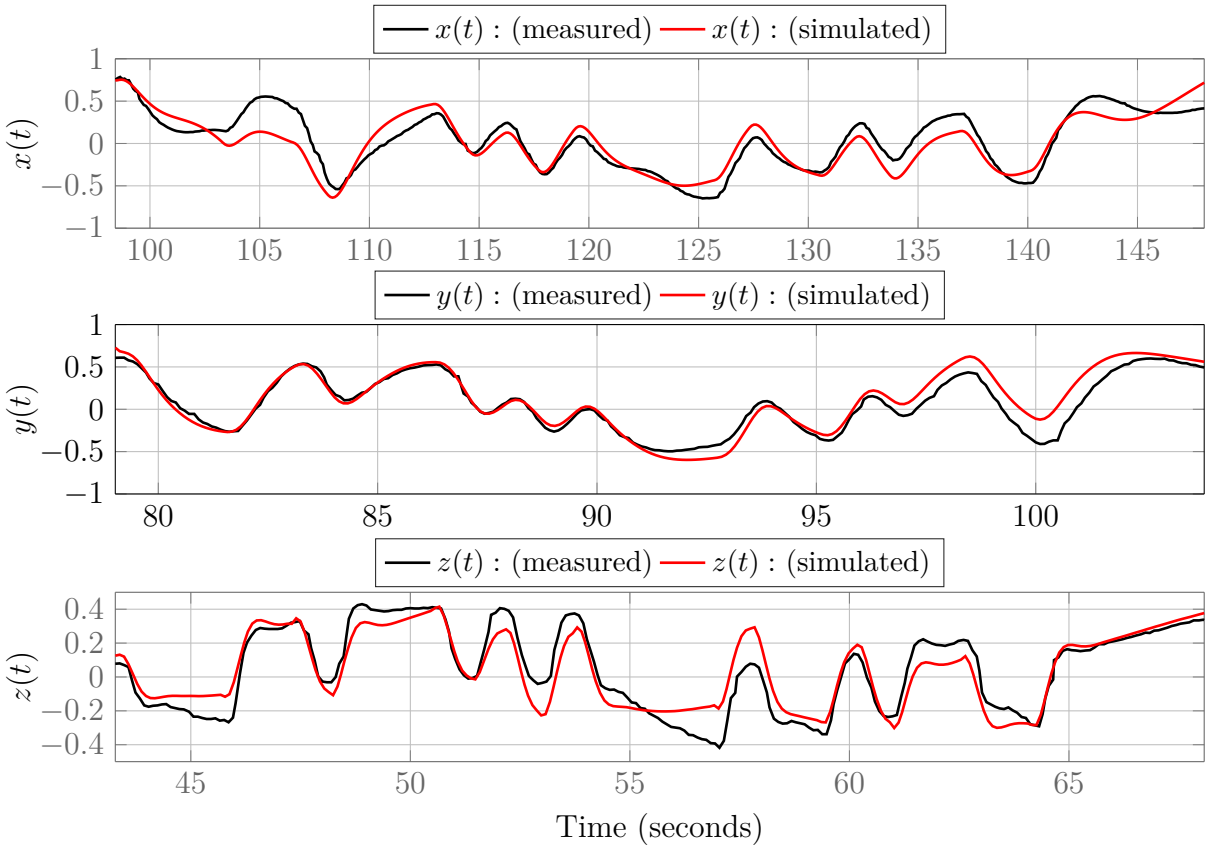


Figure B.2.5: Cross-validation of the Hummingbird state-space models identified by means of the direct approach [Kunkel, 2016]

The vertical movement results show a more significant mismatch for the direct and ARX-based joint input-output method. For this reason, the joint input-output identification was repeated using an FIR input model. The results obtained for this method are satisfactory and show a significant improvement compared to the ARX approach. Accordingly, the joint input-output method with FIR input model for the vertical dimension led to the best results among the tested methods and will be used in the following.

Models of the CX-10 small-scale quad-copter were identified in a similar manner by means of closed-loop experiments [Singh, 2016]. Again, both direct and indirect methods were tested. In contrast to the subspace identification of the Hummingbird, for the CX-10 transfer functions were identified using least squares estimation, for which each spatial dimension was treated separately. For indirect closed-loop identification, here a transfer function inversion was used. For this method the closed-loop transfer function $T_{yr}(z)$ from reference r to output y is identified experimentally. Assuming that the controller transfer function $C(z)$ is known, the plant transfer function is determined by expressing the closed-loop transfer function in terms of the plant and the controller transfer functions according to Fig. B.2.4 and solving for the plant transfer function:

$$P(z) = (C(z)(I - T_{yr}(z)))^{-1} T_{yr}(z) \tag{B.2.6}$$

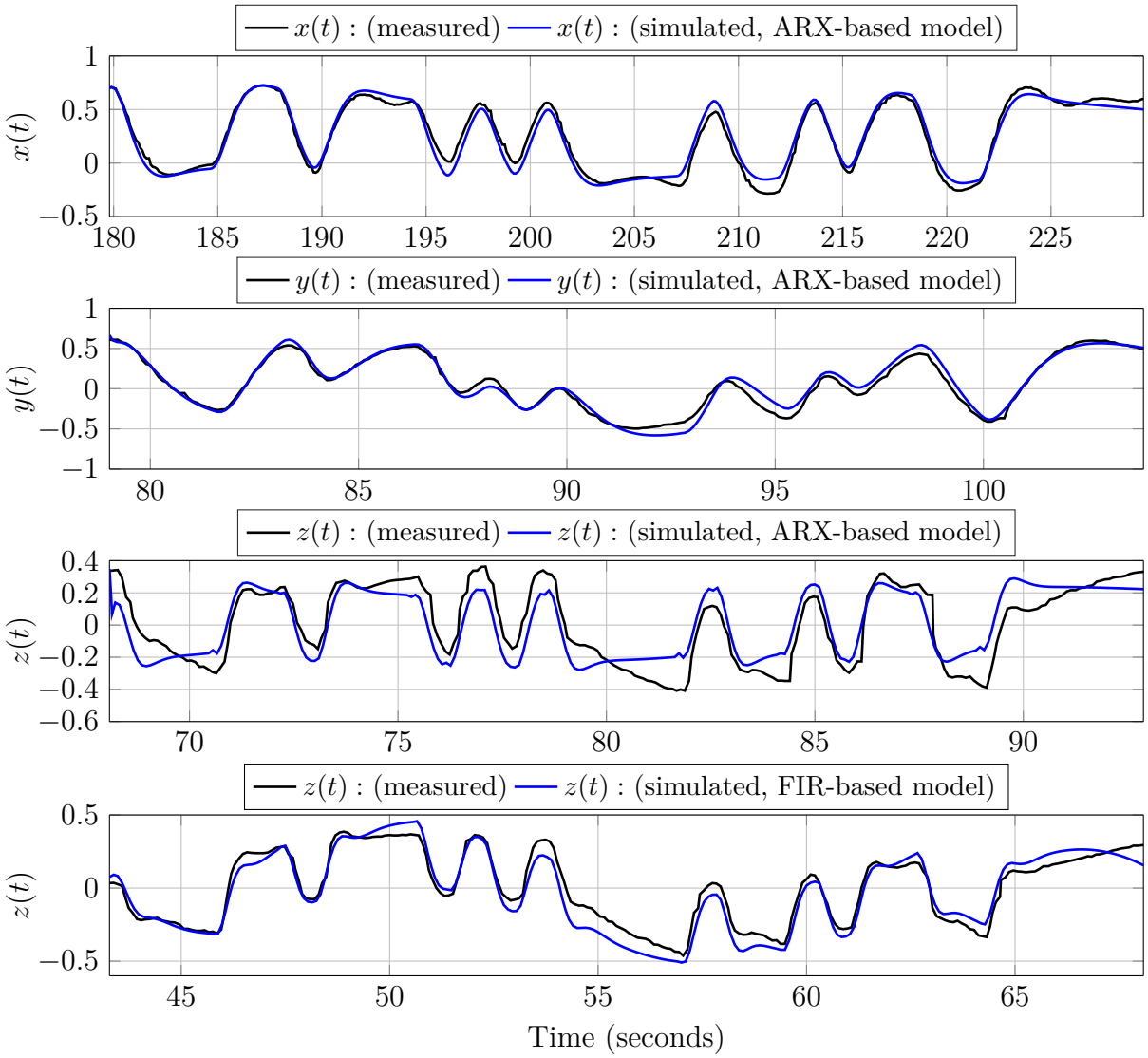


Figure B.2.6: Cross-validation of the Hummingbird state-space models identified by means of the joint input-output approach with ARX input model [Kunkel, 2016]

Cross validation of the results obtained for the CX-10 reveals a significant mismatch for the direct method, but the indirect method leads to satisfactory results.

B.2.3 Identified Agent Models

Models of the locally controlled Hummingbird quad-copter were finally identified by means of closed-loop black-box identification using the joint input-output method [Kunkel, 2016]. The identification is done separately for each dimension, neglecting any cross-coupling between the movements in different spatial dimensions. This corresponds to the diagonal

representation

$$\begin{bmatrix} y_x \\ y_y \\ y_z \end{bmatrix} = \begin{bmatrix} P_{acl,x}(s) & & \\ & P_{acl,y}(s) & \\ & & P_{acl,z}(s) \end{bmatrix} \begin{bmatrix} r_\theta \\ r_\phi \\ u_{Th} \end{bmatrix}. \tag{B.2.7}$$

The resulting models obtained from using an ARX input model are as follows:

$$P_{acl,x} = \left(\begin{array}{c|c} A_x & B_x \\ \hline C_x & D_x \end{array}\right) = \left(\begin{array}{ccc|c} -2.371 & -2.968 & -0.03677 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline -0.000155 & -0.01294 & -0.312 & 0 \end{array}\right) \tag{B.2.8a}$$

$$P_{acl,y} = \left(\begin{array}{c|c} A_y & B_y \\ \hline C_y & D_y \end{array}\right) = \left(\begin{array}{ccc|c} -2.229 & -2.664 & -0.343 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline -0.00341 & 0.0637 & 0.3843 & 0 \end{array}\right) \tag{B.2.8b}$$

$$P_{acl,z} = \left(\begin{array}{c|c} A_z & B_z \\ \hline C_z & D_z \end{array}\right) = \left(\begin{array}{ccc|c} 0.4462 & 0 & 0 & 0.1923 \\ 0 & -0.444 & -4.499 & -0.5004 \\ 0 & 4.846 & -4.57 & 1.234 \\ \hline 1.349 & 0.3073 & -0.07997 & 0 \end{array}\right) \tag{B.2.8c}$$

B.2.4 Local Controller Synthesis

This section describes the design, implementation and testing of the local position controller. For position control of the Hummingbird quad-copter, four different control techniques were tested: PID control, observer-based LQG and LQGI state feedback and $\mathcal{H}_\infty$ control. Among those, the LQGI state feedback controller achieved the best results, while the $\mathcal{H}_\infty$ controller suffered from problems in implementation.

LQGI Position Control of Hummingbird Quad-Copter

The architecture of the LQGI controller is shown in Fig. B.2.7 in discrete-time version. It is proposed e.g. in Skogestad and Postlethwaite [2005] and is an integal-augmented version of the well-known LQG architecture, which promises an improved steady-state accuracy compared to the standard LQG design. The plant dynamics are described by a strictly proper state space model with system matrices A, B and C, which is used in the state observer to observe the state vector $x(k)$ of the plant. The control input $u(k)$ is computed as weighted sum of the observed state $\hat{x}(k)$ and the integrator state $x^I(k)$, which is the integrated control error. The design parameters of this control scheme are the observer gain L and the state feedback gains F_S and F_I. For controller synthesis, i.e. determining F_S and F_I, the integral-augmented system is re-formulated as augmented system model with states $x_{aug} = [x^T \; x^{I^T}]^T$ and the optimal state feedback problem is solved, e.g. by solving the algebraic Riccati equation as proposed in Skogestad and Postlethwaite [2005]. This yields the state feedback gain $F = [F_S \; F_I]$. The optimal choice of L can be found

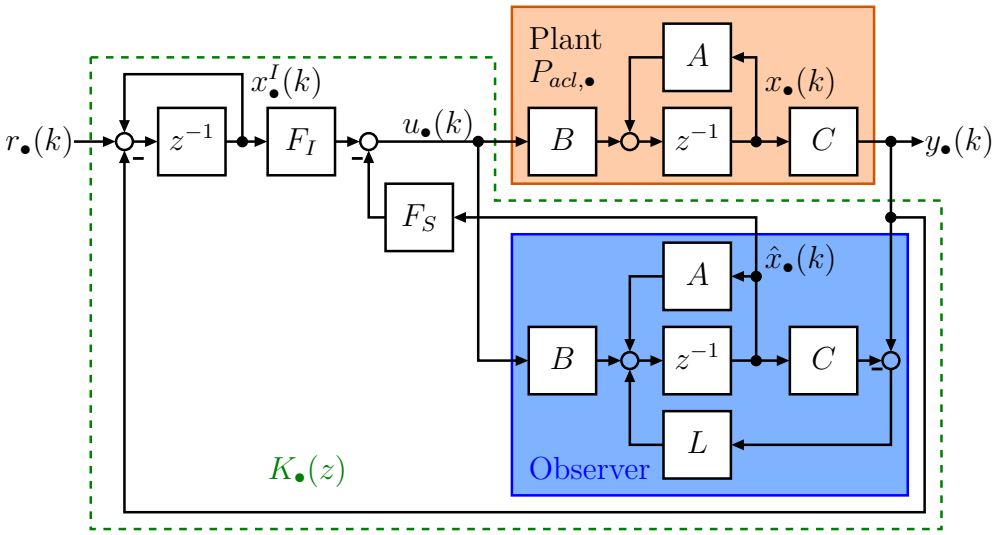


Figure B.2.7: Block diagram of the LQGI control scheme

by the same tools, exploiting the duality of the observer design problem and the optimal state feedback problem [Skogestad and Postlethwaite, 2005].

For local position control of the Hummingbird quad-copter, in this work the LQGI control scheme was applied separately for each spatial dimension. In Fig. B.2.7 this is indicated by the index $\bullet$ as place holder for the spatial dimensions, where $u_\bullet$ and $y_\bullet$ correspond to (B.2.2). The controller parameters were optimized by solving algebraic Riccati equations [Farnbacher, 2016]. As plant model, the results of the previously described black-box system identification by means of the joint input-output method with FIR input model were applied.

The obtained position controller was validated in a flight experiment as part of Farnbacher [2016]. First, the quad-copter was commanded to lift up to a height of 1 m, to sink down to 0.5 m height and to horizontally follow a given sequence of static reference points at the same height. This reference trajectory is shown in Fig. B.2.8 together with the measured trajectory of the quad-copter.

Fig. B.2.9 shows the position plotted over time. The reference steps of 0.5 m are followed with a rise time of approx. 4 sec with negligible overshoot. During the static periods, oscillations with an amplitude of 5 to 10 cm around the target position are observed.

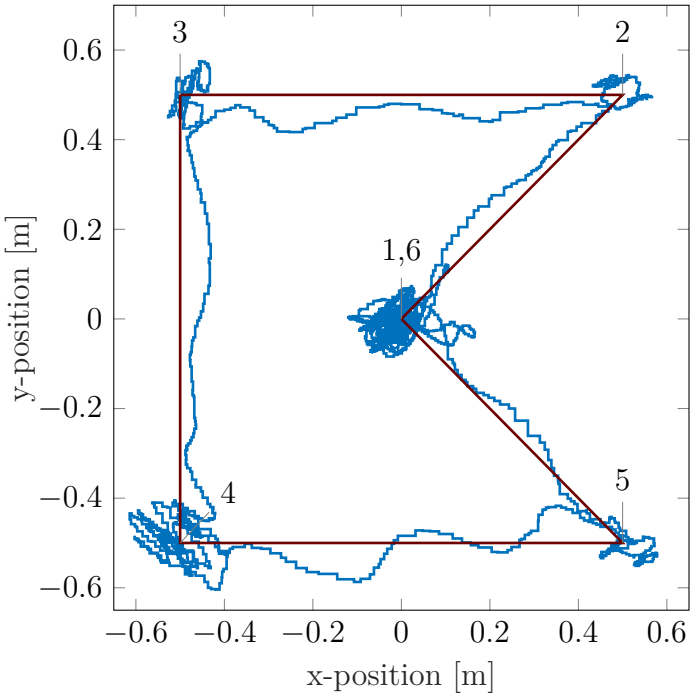


Figure B.2.8: Reference trajectory (red) and measured flight trajectory of the Hummingbird quad-copter (blue) [Farnbacher, 2016]

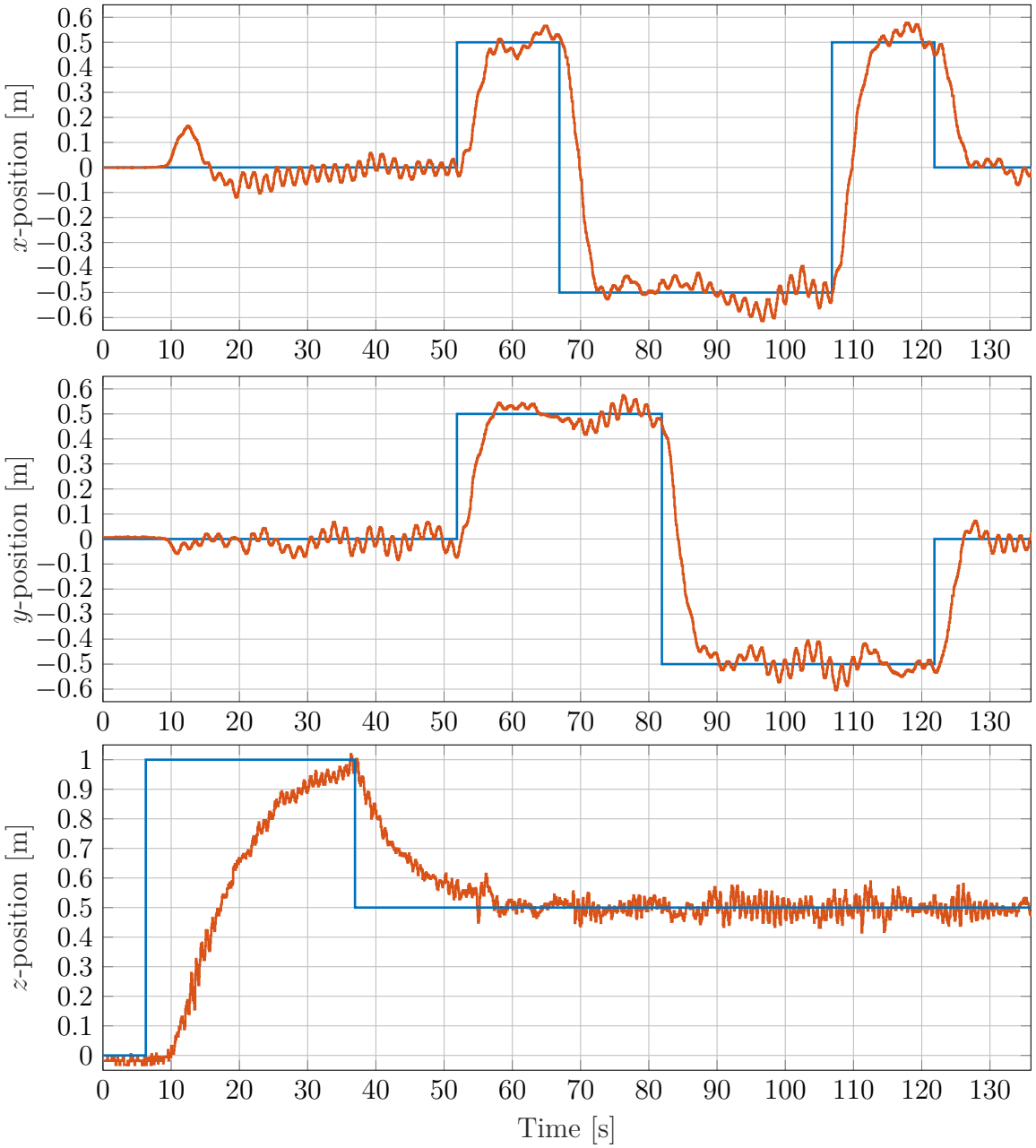


Figure B.2.9: Measured response of the flying Hummingbird quad-copter (red) for step-wise changes of the reference position (blue) [Farnbacher, 2016]

Appendix C

Notation and Symbols

1. **Abbreviations**

ARX Auto-Regressive with eXogenous input
CSMA/CA Carrier Sense Multiple Access/Collision Avoidance
CV Computer Vision
DRMS Distance Root Mean Square
EGNOS European Geostationary Navigation Overlay Service
FBSP Full-Block $\mathcal{S}$-Procedure
FIR Finite Impulse Response
GNSS Global Navigation Satellite System
GPS Global Positioning System
HIL Hardware In the Loop
HLP High Level Processor
HSV Hue-Saturation-Value
IEEE Institute of Electrical and Electronics Engineers
IFF Information Flow Filter
IGS International GNSS Service
IMU Inertial Measurement Unit
IQC Integral Quadratic Constraint
ISC Interconnection-Scheduled Control
ISM Industrial, Scientific and Medical
JTAG Joint Test Action Group
LFT Linear Fractional Transformation
LLP Low Level Processor
LMI Linear Matrix Inequality

LPS	Local Positioning System
LPV	Linear Parameter-Varying
LQG	Linear-Quadratic-Gaussian
LQGI	Linear-Quadratic-Gaussian with Integral augmentation
LTI	Linear Time-Invariant
MAC	Media Access Control
MAS	Multi-Agent System
OSI	Open Systems Interconnection
PID	Proportional-Integral-Differential
PPP	Precise Point Positioning
PRBS	Pseudo-Random Binary Sequence
RTK	Real-Time Kinematics
SBAS	Satellite Based Augmentation System
SDK	Software Development Kit
SGT	Small Gain Theorem
SISO	Single Input Single Output
SPI	Serial Peripheral Interface
TDoA	Time Difference of Arrival
TWR	Two-Way Ranging
UART	Universal Asynchronous Receiver Transmitter
UAV	Unmanned Aerial Vehicle
USB	Universal Serial Bus

2. **Complex Numbers**

$\Re(z)$	Real part of complex number z
$\Im(z)$	Imaginary part of complex number z

3. **Sets**

$\mathbb{R}$	Set of real numbers
$\mathbb{R}^{n\times m}$	Set of real matrices of dimension $n \times m$
$\mathbb{C}$	Set of complex numbers
$\mathbb{C}^{n\times m}$	Set of complex matrices of dimension $n \times m$

$\mathcal{V}$	Set of vertices in a graph $\mathcal{G}$, representing the agents in a multi-agent system
$\mathcal{E}$	Set of edges in a graph $\mathcal{G}$, representing the interaction links in a multi-agent system
N_i	Set of neighbors of agent i
$\bar{\Theta}$	Closed unit disk $\bar{\Theta} = \{z \in \mathbb{C} \mid \lvert z\rvert \leq 1\}$
$\bar{\mathbf{P}}$	Perron disk $\bar{\mathbf{P}} = \{z \in \mathbb{C} \mid \lvert z-1\rvert \leq 1\}$
$\text{eig}(M)$	Set of eigenvalues of matrix M
$\mathbf{\Psi}$	Set of admissible interaction pattern matrices
$\mathbf{\Lambda}$	Set of possible eigenvalues of interaction matrices $\Psi \in \mathbf{\Psi}$
Θ	Set of scheduling parameters of an LPV system

4. **Vectors and Matrices**

$\mathbf{1}_n$	Vector of size n filled with ones: $\mathbf{1}_n = \begin{bmatrix}1 & \cdots & 1\end{bmatrix}^T$
I_N	Identity matrix of size $N \times N$
0_N	Zero matrix of size $N \times N$
$[*]^T M \begin{bmatrix}\ddots\end{bmatrix}$	Short-hand notation for quadratic matrix expressions; $*$ stands for a repetition of the right factor

5. **Graph Theory**

$\mathcal{A}$	Adjacency matrix (row-wise scaled)
$\mathcal{A}^0$	Unscaled adjacency matrix
$\mathcal{L}$	Laplacian matrix (row-wise scaled)
$\mathcal{L}^0$	Unscaled Laplacian matrix
d_i^{in}	In-degree of vertex i
d_i^{out}	Out-degree of vertex i
$\mathcal{D}$	Degree matrix
ψ_{ik}	Weight of the link from vertex k to vertex i
Ψ	Weighting matrix

6. **Operations on Vectors, Matrices and Systems**

$\text{vcat}(v_1, \cdots, v_n) = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$	Vertical concatenation of vectors
$\text{diag}(M, N) = \begin{bmatrix} M & 0 \\ 0 & N \end{bmatrix}$	Block-diagonal concatenation of matrices or systems
$\underset{i=1}{\overset{N}{\text{diag}}} M_i = \begin{bmatrix} M_1 & & \\ & \ddots & \\ & & M_N \end{bmatrix}$	Block-diagonal concatenation of matrices or systems
$\hat{M} = I_N \otimes M$	Block-diagonal aggregation of identical matrices or systems

$M_{(n)} = M \otimes I_n$	Kronecker extension of matrix M
$\bar{\sigma}(M)$	Maximum singular value of matrix M
$\rho(M)$	Spectral radius of matrix M
$M * \Delta, \Delta * N$	LFT interconnection of M and Δ, see Section A.1
$P * K$	Redheffer star product of matrices or systems P and K, see Section A.1
$\mathrm{Tf}(u \to y)$	Transfer function from input u to output y
$G(s) = \left[\begin{array}{c\|c} A & B \\ \hline C & D \end{array}\right]$	Transfer function of the LTI system G described by the system matrices A, B, C, D

7. **Norms on Signals and Systems**

The following definitions are given for a vector $v \in \mathbb{R}^n$, a signal $v(t) : \mathbb{R}^+ \mapsto \mathbb{R}^n$ and a stable system G with input $u(t) : \mathbb{R}^+ \mapsto \mathbb{R}^q$, $\|u\|_2 < \infty$ and output $y(t) : \mathbb{R}^+ \mapsto \mathbb{R}^p$, $\|y\|_2 < \infty$.

$\|v\|_p = (\sum_{i=1}^n \|v_i\|^p)^{\frac{1}{p}}$	general vector-p-norm
$\|v\| = \|v\|_2 = \sqrt{v^T v}$	Euclidean length of vector v
$\|v(t)\|_p = \left(\int_{-\infty}^{\infty} \|v(\tau)\|_p^p d\tau\right)^{\frac{1}{p}}$	Signal-p-norm of signal $v(t)$
$\|v(t)\|_2 = \sqrt{\int_{-\infty}^{\infty} v^T(\tau) v(\tau) d\tau}$	Signal-2-norm of signal $v(t)$
$\|v(t)\|_\infty = \sup_t \|v(t)\|_\infty$	Signal-infinity-norm of signal $v(t)$
$\|G\|_{\mathcal{L}_2} = \sup_{u(t)\neq 0,\, \|u\|_2<\infty} \frac{\|y(t)\|_2}{\|u(t)\|_2}$	Induced $\mathcal{L}_2$ gain of system G
$\|G(s)\|_\infty = \sup_\omega \bar{\sigma}(G(j\omega))$	$\mathcal{H}_\infty$ system norm of LTI system $G(s)$, special case of induced $\mathcal{L}_2$ gain
$\|G\|_1 = \|G\|_{\mathcal{L}_\infty} = \sup_{u(t)\neq 0} \frac{\|y(t)\|_\infty}{\|u(t)\|_\infty}$	Induced $\mathcal{L}_\infty$ gain of system G [Rieber, 2007]
$\|G(z)\|_1 = \max_{i\in[1,p]} \sum_{k=1}^q \sum_{h=0}^\infty \|\acute{G}_{ik}(h)\|$	ℓ_1 norm of discrete-time LTI system $G(z)$ with Markov parameters $\acute{G}(t)$; special case of induced $\mathcal{L}_\infty$ gain [Rieber, 2007]

Bibliography

aerzteblatt.de (2018). Ortungsdrohnen bei Wasserrettung haben sich bewährt. [Online]. Available: `https://www.aerzteblatt.de/nachrichten/96633/Ortungsdrohnen-bei-Wasserrettung-haben-sich-bewaehrt`. (accessed 2019-03-17).

Apkarian, P. and Gahinet, P. (1995). A Convex Characterization of Gain-Scheduled H∞ Controllers. *IEEE Transactions on Automatic Control*, pages 853–864.

Apkarian, P., Gahinet, P., and Becker, G. (1995). Self-Scheduled H∞ Control of Linear Parameter-varying Systems: a Design Example. *Automatica*, 31(9):1251–1261.

Apkarian, P. and Noll, D. (2006). Nonsmooth H∞-Synthesis. *IEEE Transactions on Automatic Control*, 51(1):71–86.

Ascending Technologies (2012). AscTec Hummingbird with AutoPilot: User's Manual. Technical report.

Ascending Technologies (2017a). AscTec Research Line wiki. [Online]. Available: `http://wiki.asctec.de/display/AR/AscTec+Research+Home`. (accessed 31-03-2017).

Ascending Technologies (2017b). UAV Inspection & Monitoring. [Online]. Available: `http://www.asctec.de`. (accessed 2017-06-05).

Awad, A., Chapman, A., Schoof, E., Narang-Siddarth, A., and Mesbahi, M. (2015). Time-scale separation on networks: Tracking, consensus, and state-dependent interactions. In *54th IEEE Conference on Decision and Control (CDC)*.

Bara, G. I. (2010). Dilated LMI conditions for the robust analysis of uncertain parameter-dependent descriptor systems. In *American Control Conference*, pages 5451–5457.

Bartels, M., Liu, Q., Kaiser, G., and Werner, H. (2013). LPV Torque Vectoring for an Electric Vehicle Using Parameter-Dependent Lyapunov Functions. In *American Control Conference*.

Bartels, M. and Werner, H. (2014). Cooperative and Consensus-Based Approaches to Formation Control of Autonomous Vehicles. In *19th IFAC World Congress*.

Bartels, M. and Werner, H. (2016). Robust and Gain-Scheduled Formation Control Techniques for Multi-Agent Systems - Comparison and Benchmark in a Unified Framework. In *American Control Conference*.

Bender, D. and Laub, A. (1985). The linear-quadratic optimal regulator for descriptor systems. In *1985 24th IEEE Conference on Decision and Control*, pages 957–962. IEEE.

Berghoff, S. (2016). *Adaptation and Test of a Camera-Based Localization System for Positioning of Miniature Quad-Rotor Helicopters*. Bachelor thesis, Hamburg University of Technology, Hamburg, Germany.

Bertrand, A. and Moonen, M. (2013). Seeing the Bigger Picture: How Nodes Can Learn Their Place Within a Complex Ad Hoc Network Topology. *IEEE Signal Processing Magazine*, 30(3):71–82.

Bilgin, A. (2014). *Implementation and Comparison of Formation Control Schemes for a Swarm of Quad-Rotor Helicopters using Robust Control Techniques*. Master thesis, Hamburg University of Technology, Hamburg, Germany.

BMWi (2019). ...mit Drohnen. Unbemanntes Fliegen im Dienst von Mensch, Natur und Gesellschaft. brochure, Bundesministerium für Wirtschaft und Energie (BMWi).

Bohlin, T. (2006). *Practical Grey-box Process Identification: Theory and Applications*. Advances in industrial control. Springer-Verlag, London.

Bouabdallah, S. (2007). *Design and control of quadrotors with application to autonomous flying*. PhD thesis, Ecole Polytechnique Federale De Lausanne, Lausanne, France.

Bouabdallah, S., Noth, A., and Siegwart, R. (2004). PID vs LQ control techniques applied to an indoor micro quadrotor. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 2451–2456. IEEE/RSJ.

Carli, R., Chiuso, A., Schenato, L., and Zampieri, S. (2011). Optimal Synchronization for Networks of Noisy Double Integrators. *IEEE Transactions on Automatic Control*, 56(5):1146–1152.

Castillo, P., Dzul, A., and Lozano, R. (2004). Real-time stabilization and tracking of a four-rotor mini rotorcraft. *IEEE Transactions on Control Systems Technology*, 12(4):510–516.

Corfmat, J. P. and Morse, A. S. (1976). Decentralized control of linear multivariable systems. *Automatica*, 12(5):479–495.

Datar, A., Schneider, D., Mirali, F., Werner, H., and Frey, H. (2018). A memory weighted protocol for sampled-data systems subjected to packet dropouts. In *2018 Annual American Control Conference (ACC)*. IEEE.

Dementhon, D. F. and Davis, L. S. (1995). Model-based object pose in 25 lines of code. *Int. J. Comput. Vision (International Journal of Computer Vision)*, 15(1-2):123–141.

Dettori, M. (2001). *LMI Techniques for Control with Application to a Compact Disc Player Mechanism*. PhD thesis, Delft University of Technology, Delft, Netherlands.

Dettori, M. and Scherer, C. W. (2001). LPV Design for a CD Player: an Experimental Evaluation of Performance. In *40th IEEE Conference on Decision and Control*, pages 4711–4716.

DHL (2016). Successful Trial Integration of DHL Parcelcopter into Logistics Chain. *DHL Press Releases*.

Digi International (2015). *XBee / XBee-PRO RF Modules 802.15.4 Product Manual*. S edition.

Ebrahimi Dehshalie, M., Menhaj, M. B., Ghasemi, A., and Karrari, M. (2018). Finite-time distributed global optimal control for linear time-varying multi-agent systems: A dynamic output-feedback perspective. *IET Control Theory & Applications*, 12(9):1267–1275.

Eichler, A. (2016). *Interconnected Systems - Analysis and Distributed Controller Synthesis*. Ph.D. Thesis, Hamburg University of Technology, Hamburg, Germany. Werner, Herbert.

Eichler, A., Hoffmann, C., and Werner, H. (2013a). Robust Control of Decomposable LPV Systems Under Time-Invariant and Time-Varying Interconnection Topologies (Part 1). In *52nd IEEE Conference on Decision and Control*.

Eichler, A., Hoffmann, C., and Werner, H. (2013b). Robust Control of Decomposable LPV Systems Under Time-Invariant and Time-Varying Interconnection Topologies (Part 2). In *52nd IEEE Conference on Decision and Control*.

Eichler, A. and Werner, H. (2013). Performance Bounds on Decomposable Systems. In *American Control Conference*.

Eichler, A. and Werner, H. (2014). Closed-Form Solution for Optimal Convergence Speed of Multi-Agent Systems with Discrete-Time Double-Integrator Dynamics. *Systems & Control Letters*, 71:7–13.

Eichler, A. and Werner, H. (2015). Improved IQC Description to Analyze Interconnected Systems with Time-Varying Time-Delays. In *American Control Conference*.

Elsner, A. (2015). *Position Control of a Quad-Rotor Helicopter Using a Camera-Based Positioning System*. Student project work, Hamburg University of Technology, Hamburg, Germany.

Farnbacher, J. (2016). *Implementation and Experimental Validation of Formation Control Schemes for a Swarm of Quad-Rotor Helicopters using Robust Control Techniques*. Master thesis, Hamburg University of Technology, Hamburg, Germany.

Farnbacher, J. (2017). *Assessment of a data transmission platform using the nRF24L01+ radio transceiver for networked formation control of mobile robots*. Student project report, Hamburg University of Technology, Hamburg, Germany.

Fathian, K., Summers, T. H., and Gans, N. R. (2018). Robust Distributed Formation Control of Agents with Higher-Order Dynamics. *IEEE Control Systems Letters*, page 1.

Fax, J. A. and Murray, R. M. (2004). Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476.

Frey, H. and Simplot, D. (2007). Localized Topology Control Algorithms for Ad Hoc and Sensor Networks. In Nayak, A. and Stojmenovic, I., editors, *Handbook of Applied Algorithms: Solving Scientific, Engineering, and Practical Problems*. John Wiley and Sons.

Gahinet, P. (1992). A convex parametrization of H-infinity suboptimal controllers. In *31st IEEE Conference on Decision and Control*, pages 937–942.

Gahinet, P., Nemirovski, A., Laub, A., and Chilali, M. (1995). *LMI Control Toolbox*. The MathWorks, Inc, release 13 edition.

Genc, A. U. (2002). *Linear Parameter-Varying Modelling and Robust Control of Variable Cam Timing Engines*. PhD thesis, University of Cambridge, UK.

Gonzalez Cisneros, P. S. (2014). *Implementation of an Information Flow Filter on a Swarm of Quad-Rotor Helicopters Using Robust Control Techniques*. Student project work, Hamburg University of Technology, Hamburg, Germany.

Gonzalez Cisneros, P. S., Hoffmann, C., Bartels, M., and Werner, H. (2015). Linear Parameter-Varying Controller Design for a Nonlinear Quad-Rotor Helicopter Model for Use in Distributed Formation Control Schemes. In *54th IEEE Conference on Decision and Control*.

Gonzalez Cisneros, P. S., Hoffmann, C., Bartels, M., and Werner, H. (2016). Linear Parameter-Varying Controller Design for a Nonlinear Quad-Rotor Helicopter Model for High Speed Trajectory Tracking. In *American Control Conference*.

Grogan, S., Pellerin, R., and Gamache, M. (2018). The use of unmanned aerial vehicles and drones in search and rescue operations – a survey. In *PROLOG Conference*.

Gronemeyer, M., Bartels, M., and Horn, J. (2016). Comparison of Consensus Loop Designs with a Mission Error Signal. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*.

Gronemeyer, M., Bartels, M., Werner, H., and Horn, J. (2017). Using particle swarm optimization for source seeking in multi-agent systems. In *20th IFAC World Congress*.

Hackbarth, A., Kreuzer, E., and Solowjow, E. (2015). HippoCampus: A Micro Underwater Vehicle for Swarm Applications. In *International Conference on Intelligent Robots and Systems*.

Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge University Press, Cambridge, UK and New York, 2nd ed. edition.

Hashemi, S. M., Abbas, H. S., and Werner, H. (2009). LPV modelling and control of a 2-DOF robotic manipulator using PCA-based parameter set mapping. In *48th IEEE Conference on Decision and Control*, pages 7418–7423.

Hill, D. J. and Mareels, I. M. (1990). Stability theory for differential/algebraic systems with application to power systems. *IEEE Transactions on Circuits and Systems*, 37(11):1416–1423.

Hoffmann, C. (2015). *Linear Parameter-Varying Control of Systems of High Complexity*. PhD thesis, Hamburg University of Technology, Hamburg, Germany.

Hoffmann, C., Eichler, A., and Werner, H. (2013). Distributed Control of Linear Parameter-Varying Decomposable Systems. In *American Control Conference*, pages 2386–2391.

Hoffmann, C., Eichler, A., and Werner, H. (2014). Control of Heterogeneous Groups of LPV Systems Interconnected through Directed and Switching Topologies. In *American Control Conference*.

Hoffmann, C., Eichler, A., and Werner, H. (2015). Control of Heterogeneous Groups of LPV Systems Interconnected through Directed and Switching Topologies. *IEEE Transactions on Automatic Control*, 60(7):1904–1909.

Hoffmann, C. and Werner, H. (2015). Control of Heterogeneous LPV Systems Interconnected through Arbitrary and Switching Directed Topologies. In *54th IEEE Conference on Decision and Control*.

Hoffmann, C. and Werner, H. (2017). Convex Distributed Controller Synthesis for Interconnected Heterogeneous Subsystems via Virtual Normal Interconnection Matrices. *IEEE Transactions on Automatic Control*, 62(10):5337–5342.

Jadbabaie, A., Lin, J., and Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001.

Kahlefendt, C. (2015). *Implementierung und Untersuchung von Verfahren zur Erhöhung der Genauigkeit der GPS-basierten Positionsbestimmung eines Quadrokopters*. Bachelor thesis, Hamburg University of Technology, Hamburg, Germany.

Kaiser, G. (2014). *Linear Parameter-Varying Control for an Electric Vehicle*. PhD thesis, Hamburg University of Technology, Hamburg, Germany.

Kanistras, K., Martins, G., Rutherford, M. J., and Valavanis, K. P. (2013). A survey of unmanned aerial vehicles (UAVs) for traffic monitoring. In *International Conference on Unmanned Aircraft Systems (ICUAS), 2013*, pages 221–234. IEEE.

Klein, D. J., Bettale, P. K., Triplett, B. I., and Morgansen, K. A. (2008). Autonomous underwater multivehicle control with limited communication: Theory and experiment. In *2nd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles*.

Konda, Y. (2016). *Positioning of a Quad-Rotor Helicopter Using Data Fusion of GPS, Inertial and Optical Flow Measurements*. Bachelor thesis, Hamburg University of Technology, Hamburg, Germany.

Kunkel, F. (2016). *System Identification and Position Control of a Quadrocopter by Means of Camera-Based Position Tracking*. Master thesis, Hamburg University of Technology, Hamburg, Germany.

Lafferriere, G., Williams, A., Caughman, J., and Veerman, J. J. P. (2005). Decentralized control of vehicle formations. *Systems & Control Letters*, 54(9):899–910.

Landau, I. D. and Zito, G. (2006). *Digital Control Systems: Design, identification and implementation*. Communications and Control Engineering. Springer, London.

Langbort, C., Chandra, R. S., and D'Andrea, R. (2004). Distributed control design for systems interconnected over an arbitrary graph. *IEEE Transactions on Automatic Control*, 49(9):1502–1519.

Lara, D., Sanchez, A., Lozano, R., and Castillo, P. (2006). Real-Time Embedded Control System for VTOL Aircrafts: Application to stabilize a quad-rotor helicopter. In *International Conference on Control Applications*.

Laß, M. B. (2016). *Extension of the position measurement setup for position control of a quad-rotor helicopter by integration of an optical flow sensor*. Student project work, Hamburg University of Technology, Hamburg, Germany.

Ljung, L. (1999). *System identification: Theory for the user*. Prentice Hall information and system sciences series. Prentice Hall PTR, Upper Saddle River NJ, 2nd ed. edition.

Ma, Q., Xu, S., Lewis, F. L., Zhang, B., and Zou, Y. (2016). Cooperative Output Regulation of Singular Heterogeneous Multiagent Systems. *IEEE transactions on cybernetics*, 46(6):1471–1475.

Massioni, P. (2014). Distributed control for alpha-heterogeneous dynamically coupled systems. *Systems & Control Letters*, 72(0):30–35.

Massioni, P. and Verhaegen, M. (2008). Distributed control of vehicle formations: A decomposition approach. In *47th IEEE Conference on Decision and Control*, pages 2906–2912.

Massioni, P. and Verhaegen, M. (2009). Distributed Control for Identical Dynamically Coupled Systems: A Decomposition Approach. *IEEE Transactions on Automatic Control*, 54(1):124–135.

Massioni, P. and Verhaegen, M. (2010). A full block S-procedure application to distributed control. In *American Control Conference*.

Masubuchi, I., Akiyama, T., and Saeki, M. (2003). Synthesis of output feedback gain-scheduling controllers based on descriptor LPV system representation. In *42nd IEEE Conference on Decision and Control*, pages 6115–6120.

Masubuchi, I., Kamitane, Y., Ohara, A., and Suda, N. (1997). H∞ control for descriptor systems: A matrix inequalities approach. *Automatica*, 33(4):669–673.

Masubuchi, I., Kato, J., Saeki, M., and Ohara, A. (2004). Gain-Scheduled Controller Design Based on Descriptor Representation of LPV Systems: Application to Flight Vehicle Control. In *43rd IEEE Conference on Decision and Control*.

Mazur, M. (2016). Six Ways Drones Are Revolutionizing Agriculture. *MIT Technology Review*.

Meng, F., Shi, Z., and Zhong, Y. (2016). Distributed formation control of singular multi-agent systems. In *2016 12th IEEE International Conference on Control and Automation (ICCA)*, pages 915–920. IEEE.

Mesbahi, M. and Egerstedt, M. (2010). *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, Princeton and Oxford.

Milhan, P. G. (2015). *Design and Implementation of an Image Processing Platform to Detect Position and Orientation of Mobile Robots in a Plane*. Bachelor thesis, Hamburg University of Technology, Hamburg, Germany.

Mirali, F., Mendez Gonzalez, A., and Werner, H. (2017). Improving the Performance of Cooperative Control Using Novel Weighting Strategies. In *Proc. 20th IFAC World Congr.*

Mirali, F. and Werner, H. (2016). Delay Robustness of Interconnected Systems: IQC-Based Analysis for Special Interconnection Structures. In *American Control Conference*.

Mirali, F. and Werner, H. (2017). Distributed Weighting Strategies for Improved Convergence Speed of First-Order Consensus. In *American Control Conference*.

Mirali, F. and Werner, H. (2018). A Novel Sequence Weighting Method for First-Order Consensus Problems. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 97–102.

Murthy, S. P. (2017). *Positioning 3-D Quad-Copters Using Image Processing*. Student project work, Hamburg University of Technology, Hamburg, Germany.

Nelles, O. (2001). *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. Engineering online library. Springer, Berlin.

Neumann, F. and Frey, H. (2015). Foundation of Reactive Local Topology Control. *IEEE Communications Letters*, 19(7):1213–1216.

Nguyen Cong, T. (2017). *Implementation and Test of a Distributed Formation Control Scheme for a Swarm of Small-Scale Quad-Rotor Helicopters using Robust Control Techniques*. Student project work, Hamburg University of Technology, Hamburg, Germany.

Niemann, C. (2016). Wasserrettung aus der Luft. *Drohnenmagazin*, 1/2016:24–29.

Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420.

Olfati-Saber, R., Fax, J. A., and Murray, R. M. (2007). Consensus and Cooperation in Networked Multi-Agent Systems. *Proceedings of the IEEE*, 95(1):215–233.

Olfati-Saber, R. and Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533.

Olshevsky, A. and Tsitsiklis, J. (2011). Convergence Speed in Distributed Consensus and Averaging. *SIAM Review*, 53(4):747–772.

Pan, Y.-J., Werner, H., Huang, Z., and Bartels, M. (2017). Distributed cooperative control of leader–follower multi-agent systems under packet dropouts for quadcopters. *Systems & Control Letters*, 106:47 – 57.

Pilz, U. (2013). *Cooperative Control of Multi-Agent Systems with Application to Quad-Rotor Helicopters*. PhD thesis, Hamburg University of Technology, Hamburg, Germany.

Pilz, U., Popov, A., and Werner, H. (2009). Robust Controller Design for Formation Flight of Quad-Rotor Helicopters. In *48th IEEE Conference on Decision and Control*, pages 8322–8327.

Pilz, U., Popov, A., and Werner, H. (2011). An Information Flow Filter Approach to Cooperative Vehicle Control. In *18th World Congress*, pages 7432–7437.

Pilz, U. and Werner, H. (2012a). An H∞/ℓ1 Control Approach to Cooperative Control of Multi-Agent Systems. In *51st IEEE Conference on Decision and Control*.

Pilz, U. and Werner, H. (2012b). Robust Stability, Performance and Information Flow in Cooperative Control of Multi-Agent Systems. *IEEE Transactions on Control Systems Technology*.

Pilz, U. and Werner, H. (2013). Convergence Speed in Formation Control of Multi-Agent Systems - a Robust Control Approach. In *52nd IEEE Conference on Decision and Control*.

Popov, A. (2012). *Design of Distributed and Fixed-Structure Controllers for Cooperative Vehicle Control*. PhD thesis, Hamburg University of Technology, Hamburg, Germany.

Popov, A. and Werner, H. (2009). A Robust Control Approach to Formation Control. In *European Control Conference*, pages 4428–4433.

Popov, A. and Werner, H. (2012). Robust Stability of a Multi-Agent System under Arbitrary and Time-Varying Communication Topologies and Communication Delays. *IEEE Transactions on Automatic Control*, 57(9):2343–2347.

RCGroups.com (2017). Cheerson cx-10: Mini-Review by user "SeByDocKy". [Online]. Available: https://www.rcgroups.com/forums/showthread.php?2183413-Cheerson-cx-10. (accessed 10-04-2017).

Ren, W. and Beard, R. W. (2008). *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications*. Communications and Control Engineering. Springer-Verlag London Limited, London.

Reynolds, C. W. (1987). Flocks, herds and schools: a distributed behavioural model. *Computer Graphics*, 21(4):25–34.

Rieber, J. M. (2007). *Control of uncertain systems with L_1 and quadratic performance objectives*. PhD thesis, Universität Stuttgart, Stuttgart.

Sastry, S. and Desoer, C. A. (1981). Jump behavior of circuits and systems. *IEEE Transactions on Circuits and Systems*, 28(12):1109–1124.

Schalk, L. (2013). *Untersuchungen zu Qualität und Verbesserungsmöglichkeiten der GPS-basierten Positionsbestimmung eines Quadrokopters*. Bachelor thesis, Hamburg University of Technology, Hamburg, Germany.

Scheibner, W. (2014). *Systemidentifikation und Modellierung eines Quadrokopters*. Bachelor thesis, Hamburg University of Technology, Hamburg, Germany.

Scherer, C. W. (2000). Robust Mixed Control and LPV Control with Full Block Scalings. In El Ghaoui, L. and Niculescu, S.-I., editors, *Advances in Linear Matrix Inequality Methods in Control*. SIAM.

Scherer, C. W. (2001). LPV Control and Full Block Multipliers. *Automatica*, 37(3):361–375.

Scorletti, G. and Duc, G. (2001). An LMI approach to dencentralized H-infinity control. *International Journal of Control*, 74(3):211–224.

Seyboth, G. S., Schmidt, G. S., and Allgöwer, F. (2012). Cooperative control of linear parameter-varying systems. In *American Control Conference*.

Shakhatreh, H., Sawalmeh, A., Al-Fuqaha, A., Dou, Z., Almaita, E., Khalil, I., Othman, N., Khreishah, A., and Guizani, M. (2018). Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access*, 7.

Shamma, J. S. (1988). *Analysis and Design of Gain Scheduled Control Systems*. PhD thesis, Massachusetts Institute of Technology.

Singh, S. (2016). *System Identification of Miniature Quad-Rotor Helicopters using a Camera-Based Positioning System*. Student project work, Hamburg University of Technology, Hamburg, Germany.

Skogestad, S. and Postlethwaite, I. (2005). *Multivariable Feedback Control - Analysis and Design*. John Wiley & Sons, Ltd, 2nd edition.

Steinmetz, F. (2015). *Extension of a camera-based indoor positioning system for positioning of flying objects in 3D space*. Bachelor thesis, Hamburg University of Technology, Hamburg, Germany.

Timotin, D. (1995). Redheffer Products and Characteristic Functions. *Journal of Mathematical Analysis and Applications*, 196(3):823 – 840.

Tsitsiklis, J. N. and Athans, M. (1984). Convergence and asymptotic agreement in distributed decision problems. *IEEE Transactions on Automatic Control*, 29(1):42–50.

van den Hof, P., Schrama, R., and Bosgra, O. H. (1992). An indirect method for transfer function estimation from closed loop data. In *31st IEEE Conference on Decision and Control*, pages 1702–1706.

Wendel, J. (2011). *Integrierte Navigationssysteme*. Oldenbourg Verlag, 2nd edition.

Werner, H. and Frey, H. (2015). *Analysis and synthesis of combined cooperative control and topology control over wireless network models: Project Proposal*. DFG Priority Programme "Cyber-Physical Networking" (SPP 1914).

Witt, J., Annighöfer, B., Falkenberg, O., and Weltin, U. (2011). Design of a High Performance Quad-Rotor Robot Based on a Layered Real-Time System Architecture. In *4th International Conference on Intelligent Robotics and Applications*, pages 312–323.

Wolfe, J., Chichka, D., and Speyer, J. (1996). Decentralized controllers for unmanned aerial vehicle formation flight. In *Guidance, Navigation, and Control Conference*. American Institute of Aeronautics and Astronautics.

Wollnack, S. (2016). Personal communication. Hamburg University of Technology.

Wu, F. (1995). *Control of Linear Parameter Varying Systems*. PhD thesis, University of Berkeley, California, USA.

Wu, F. and Dong, K. (2006). Gain-Scheduling Control of LFT Systems Using Parameter-Dependent Lyapunov Functions. *Automatica*, 42(1):39–50.

Xi, J., Meng, F., Shi, Z., and Zhong, Y. (2012). Delay-dependent admissible consensualization for singular time-delayed swarm systems. *Systems & Control Letters*, 61(11):1089–1096.

Xi, J., Yao, Z., Liu, H., Liu, G., and Zhong, Y. (2014). Admissible consensus for descriptor multi-agent systems with switching topologies. In *Proceedings of the 33rd Chinese Control Conference*, pages 1138–1143. IEEE.

Xia, Y., Boukas, E.-K., Shi, P., and Zhang, J. (2009). Stability and stabilization of continuous-time singular hybrid systems. *Automatica*, 45(6):1504–1509.

Xiao, L. and Boyd, S. P. (2004). Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78.

Yang, X.-R. and Liu, G.-P. (2012). Necessary and Sufficient Consensus Conditions of Descriptor Multi-agent Systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 59(11):2669–2677.

Yang, X.-R. and Liu, G.-P. (2014). Consensus of descriptor multi-agent systems via dynamic compensators. *IET Control Theory & Applications*, 8(6):389–398.

Zalzala, A. M. S. and Fleming, P. (1997). *Genetic Algorithms in Engineering Systems*. IET, Stevenage, UK.

Zheng, S., Zhang, X., and Sheng, S. (2017). Simultaneous fault detection and control protocol design for coordination of multi-agent systems. In *2017 29th Chinese Control And Decision Conference (CCDC)*, pages 6425–6430.

Zhou, K., Doyle, J., and Glover, K. (1996). *Robust and Optimal Control*. Prentice-Hall, N.J., USA.

Zhu, Y. (2001). *Multivariable system identification for process control*. Pergamon, Amsterdam and London.

Zou, Y., Zhou, Z., Dong, X., and Meng, Z. (2018). Distributed Formation Control for Multiple Vertical Takeoff and Landing UAVs with Switching Topologies. *IEEE/ASME Transactions on Mechatronics*.

Author's Publications

Publications of the Author

- Bartels, M. and Werner, H. (2016). Robust and Gain-Scheduled Formation Control Techniques for Multi-Agent Systems - Comparison and Benchmark in a Unified Framework. In *American Control Conference*

- Bartels, M. and Werner, H. (2014). Cooperative and Consensus-Based Approaches to Formation Control of Autonomous Vehicles. In *19th IFAC World Congress*

- Bartels, M., Liu, Q., Kaiser, G., and Werner, H. (2013). LPV Torque Vectoring for an Electric Vehicle Using Parameter-Dependent Lyapunov Functions. In *American Control Conference*

Co-Authored Publications

- Gronemeyer, M., Bartels, M., Werner, H., and Horn, J. (2017). Using particle swarm optimization for source seeking in multi-agent systems. In *20th IFAC World Congress*

- Pan, Y.-J., Werner, H., Huang, Z., and Bartels, M. (2017). Distributed cooperative control of leader–follower multi-agent systems under packet dropouts for quadcopters. *Systems & Control Letters*, 106:47 – 57

- Gonzalez Cisneros, P. S., Hoffmann, C., Bartels, M., and Werner, H. (2016). Linear Parameter-Varying Controller Design for a Nonlinear Quad-Rotor Helicopter Model for High Speed Trajectory Tracking. In *American Control Conference*

- Gronemeyer, M., Bartels, M., and Horn, J. (2016). Comparison of Consensus Loop Designs with a Mission Error Signal. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*