Circuits and Systems
Mekelweg 4,
2628 CD Delft
The Netherlands
http://cas.tudelft.nl/

CAS-2021-4464397

M.Sc. Thesis

Edge State Kalman Filtering for Distributed Formation Control Systems

Martijn van der Marel B.Sc.

Abstract

Formation control problems consider a set of mobile agents with the underlying goal of attaining and maintaining a state where the relative positions of agents are stable in accordance with the desired configuration. Navigation for formation control is typically achieved through localization in a global reference frame, e.g., via GNSS. However, when a global reference frame is not shared among agents, a relative navigation approach is required.

Distributed filtering for relative localization in formation control systems is a relatively unexplored field. The absence of absolute positioning means motivates the need for a distributed filter that operates on the edges of the sensing graph of the multi-agent system. In this thesis, a data model for relative formation control problems and two edge-based Kalman filters are proposed. The first filter is designed for an individual edge. The second is a filter designed via decoupling of the optimal global filter which allows for the joint estimation of adjacent edges. It is shown that the joint filter is optimal under the decoupling constraints.

Monte Carlo results show that when random environmental disturbances are correlated among agents, the joint filter outperforms the local edge filter in a mean square error sense.

Lastly, systems are considered where inter-agent communications are unavailable, leading to biased prediction steps of the Kalman filters. We aim to minimize this effect through the proposal of a local Wiener filter which predicts the control actions of neighboring agents.

TU Delft

Faculty of Electrical Engineering, Mathematics and Computer Science

Delft University of Technology

Edge State Kalman Filtering for Distributed Formation Control Systems

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

Martijn van der Marel B.Sc.
born in Leiden, the Netherlands

This work was performed in:

Circuits and Systems Group
Department of Microelectronics
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

TU Delft

Delft University of Technology

Delft University of Technology
Department of
Microelectronics

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"Edge State Kalman Filtering for Distributed Formation Control Systems"** by **Martijn van der Marel B.Sc.** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: June 17 2021

Chairman: ______________________________
prof.dr.ir. G.J.T. Leus

Advisor: ______________________________
dr. R.T. Rajan

Committee Member: ______________________________
prof.dr. A. Yarovoy

Abstract

Formation control problems consider a set of mobile agents with the underlying goal of attaining and maintaining a state where the relative positions of agents are stable in accordance with the desired configuration. Navigation for formation control is typically achieved through localization in a global reference frame, e.g., via GNSS. However, when a global reference frame is not shared among agents, a relative navigation approach is required.

Distributed filtering for relative localization in formation control systems is a relatively unexplored field. The absence of absolute positioning means motivates the need for a distributed filter that operates on the edges of the sensing graph of the multi-agent system. In this thesis, a data model for relative formation control problems and two edge-based Kalman filters are proposed. The first filter is designed for an individual edge. The second is a filter designed via decoupling of the optimal global filter which allows for the joint estimation of adjacent edges. It is shown that the joint filter is optimal under the decoupling constraints.

Monte Carlo results show that when random environmental disturbances are correlated among agents, the joint filter outperforms the local edge filter in a mean square error sense.

Lastly, systems are considered where inter-agent communications are unavailable, leading to biased prediction steps of the Kalman filters. We aim to minimize this effect through the proposal of a local Wiener filter which predicts the control actions of neighboring agents.

Acknowledgments

The process of writing this thesis has undoubtedly been challenging at times. Hence, hereby I would like to show my appreciation for the support provided by the people around me during the project.

First and foremost, I would like to sincerely thank my supervisor dr. R.T. Rajan. I highly appreciate the numerous hours of discussions, advice and support that has been put into guiding me throughout the project. Through our shared passion for the subject, our weekly discussions have been a very pleasant experience. Additionally, I would like to express my gratitude for the support provided while writing and revising the paper now submitted to IEEE Control System Letters (L-CSS).

I would also like to thank all the CAS students and staff, particularly the group of MSc students with whom I have had the pleasure of exchanging our MSc project experiences on a weekly basis.

Lastly, I would like to thank my friends and family for their support, which has been immensely important for me throughout the past year. In particular I would like to mention my girlfriend Jitske and my roommate Roy whose support has been both invaluable and unconditional.

Martijn van der Marel B.Sc.
Delft, The Netherlands
June 17 2021

Contents

List of Figures

List of Tables

Introduction 1

1.1 Background

A major trend in science and engineering concerns decentralization. A task first performed by a central unit is now distributed among multiple agents which all perform a subset of the main task. As such, the main task is performed in a so-called *distributed* or *decentralized* manner. One of the advantages of decentralization is an intrinsic robustness to single point of failure (SPOF). If any single agent fails, a well-designed decentralized system can recover and continue with one fewer agent. In a decentralized system, no single central controller exists which can fail and shut down the whole system. Another advantage of decentralization is computational efficiency, as agents perform their work on a large task in parallel. Directly related is the scalability of distributed systems. Local parallel processing allows an increasing number of agents without increasing the load on a central processing unit or on communication with such a unit, as would be the case with centralized systems.

Decentralization is seen in countless research areas, from computer processor design to electricity grids. In this thesis, the focus is on decentralized systems with mobile agents, i.e., agents which can freely move around in Euclidean space. For various reasons, it may be desirable to keep the relative dynamical states of agents (i.e., position, velocity) with respect to other agents stable. If this is achieved, we say the agents are *in formation*. The general *formation control problem* therefore refers to letting a set of mobile agents converge to a desired formation, and keeping them there. Of course, there are countless variants of the formation control problem. These differ in e.g., the type of formation, the dynamics of the agents, the sensing and communication capabilities of agents, and the control hierarchy of the system.

Before discussing the state of the art and the goals and outline of this thesis, first some application areas of formation control systems are introduced.

1.2 Application Areas of Formation Control

Three application areas of formation control systems are discussed, with special interest in domains with limited absolute positioning means. First satellite arrays are considered, followed by UAV and AUV formations, respectively.

1.2.1 Satellite Interferometry

Since the invention of the first telescope, a desire in radio astronomy has been to increase the resolution of astronomical images. Originally, this meant increasing the size of telescope dishes. Interferometry provides an alternative. Interferometry involves using

Table 1.1: Selection of past and present-day multi-satellite interferometry projects.

Project name	Lead	Observation objective	Formation type
IRASSI [2]	University Munich	Cold space	Free-drifting
OLFAR [3]	PIPP OLFAR group	Early cosmos	Free-drifting
Darwin [5]	ESA	Extrasolar planets	Tetrahedral
Cluster II [6]	ESA	Earth magnetosphere	Tetrahedral
MMS [7]	NASA	Earth magnetosphere	Tetrahedral
LISA [4]	ESA	Gravitational waves	Triangular

arrays of telescopes which can achieve the same angular resolution as a much bigger single telescope [1]. Instead of ground-based astronomy, this also allows the use of satellite arrays which experience less atmospheric pollution compared to ground-based telescopes. The recent IRASSI (Infrared Astronomy Satellite Swarm Interferometry) [2] and OLFAR (Orbiting Low Frequency Array) [3] projects are examples of these types of systems. Satellite interferometry is however not only used for astronomical purposes, but can also be used to observe other phenomena, such as gravitational waves [4], extrasolar planets [5] or Earth's magnetosphere [6], [7].

There are two general methods of using swarms of satellites for interferometry. One is to have the satellites free-flying within a certain region, as is done in [2], [3], [8]. This requires less precise control over the satellites. However, for some applications more control is needed over the (relative) positions of the satellites, as in [5]–[7]. In these cases, millimeter- or even micrometer-level precision formation control is required. Formation control for these systems is usually based on relative distance measurements, since absolute localization, which on Earth is often done through GNSS systems such as GPS, is problematic in space. A variety of sensors is often used on a single satellite for relative localization, but for the most accurate position estimates, laser-based sensors are often used [2].

Translational movement of satellites is achieved through high-precision propulsion techniques. With thrusters placed such that applied forces are orthogonal to each other, the control inputs for translational movement can be decoupled into independent control inputs for all three dimensions [9]. If orbital dynamics are also ignored, the dynamics of a satellite can be modelled as a double integrator:

$$\dot{\mathbf{x}} = \mathbf{v}, \quad \dot{\mathbf{v}} = \frac{1}{M}\mathbf{u} \tag{1.1}$$

with $\mathbf{x}, \mathbf{v}, \mathbf{u} \in \mathbb{R}^3$ the position, velocity and thrust force of the satellite respectively, and M the mass of the satellite. Attitude dynamics and -control of satellites are more involved, and will not be considered in this thesis.

A selection of some projects involving multi-satellite systems are shown in Table 1.1. While many of the currently launched satellite formations consist of a limited number of satellites, recent projects like OLFAR include an increased number of (smaller and cheaper) satellites. Distributed formation control is therefore a promising field for the future of satellite interferometry projects.

1.2.2 UAV Formation Flight

Another application area for distributed formation control is that of Unmanned Aerial Vehicles (UAVs). Formations of UAVs can have a myriad of functions. Some examples include but are not limited to surveillance operations, surface mapping and crop inspection.

UAV-based systems can be roughly classified into two classes based on their dynamics. Fixed-wing UAVs need to maintain a certain velocity and are often modelled using modified unicycle models [10], which are nonholonomic and dimensions are not decoupled. Rotary wing UAVs are often modelled as single- or double integrators in two or three dimensions [11]. The single- and double integrator models can be justified because rotary wings UAVs can be in equilibrium when stationary. More complex dynamical models for these UAVs are also omnipresent in literature [12], [13]. However, the decoupled dimensions and holonomicity of the single- and double integrator models make them particularly attractive as a starting point for stability analysis of formation control algorithms.

Often, UAVs are equipped with GNSS modules, giving them an absolute reference frame. There are some specific applications where this is not possible, e.g., in indoor applications or when GNSS modules are not cost-effective. In these applications relative position measurements are required for localization. Usually, UAVs measure distances (e.g., via RSS, radar, lidar, stereo cameras) and bearings (e.g., via omnidirectional cameras), and then combine these measurements into a relative position estimate.

UAVs are particularly susceptible to disturbances from the environment in the form of wind [14]. The wind direction and magnitude are generally approximately uniform over the space spanned by a formation and as such the disturbances are often correlated among the UAVs in the formation. In addition, wind speed is correlated in time. Prior information on wind patterns might therefore be exploited to counteract its effect on formation control in UAV applications.

1.2.3 Underwater Sensing Formations

Another domain where absolute localization is compromised is underwater. GNSS systems function poorly underwater and vision is similarly limited. As such, distributed formations of Autonomous Underwater Vehicles (AUVs) based on local relative positioning are particularly suited to this domain. There are various functions a distributed formation could fulfill under water. Consider for example the search for naval mines [15], or the sampling of ocean currents [16]. Distributed AUVs can sweep ocean floor for mines quicker and are more cost-efficient than a centralized solution. The same argument can be made for taking spatial samples of currents.

Underwater agents often communicate acoustically between themselves and can use the acoustic energy to estimate their relative distance. Alternatively, laser-based rangefinders exist for higher precision rangefinding [17]. Through using multiple rangefinders on a single agent with some distance between them, bearings may be estimated using triangulation. Specifically, $D + 1$ spatially separated rangefinders are necessary to compute relative position in D-dimensional space. This method of relative position estimation increases in precision when the distance between the rangefinders increases. For underwater systems with acoustic rangefinders this approach is particularly useful because agents are generally relatively large in size and sound waves travel relatively slowly.

The dynamics of AUVs are highly dependent on the design and type of propulsion. While many systems are nonholonomic [18], holonomic vehicles with full translational freedom of movement exist as well [19]. The latter type of vehicles are particularly useful for maintaining stationary underwater formations.

1.3 State of the Art

In this section, relevant state of the art is summarized. First, the state of the art for relative formation control systems is covered, followed by a brief summary of recent work on distributed Kalman filtering for these systems.

1.3.1 Relative Formation Control

In [20], formation control problems are categorized into three groups, based on the sensing capabilities of agents.

1. In *position-based* control systems, agents share a common global reference frame. They are capable of measuring their own position with respect to this global reference frame. Hence, we call these measurements *absolute* state measurements. Systems equipped with GNSS modules generally belong to this category.

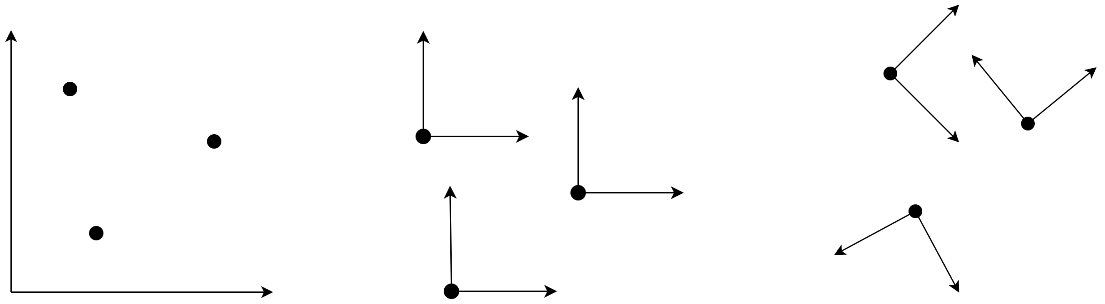
2. In *displacement-based* control systems, agents do not share a common reference frame, but their local reference frames do share a common orientation. In practice, this can be achieved through e.g., an onboard compass. Position measurements taken in a local reference frame are by definition *relative* state measurements, which are then generally also the variable to be controlled.

3. In *distance-based* control systems, agents share neither a common reference frame nor a common reference orientation. Inter-agent distances are controlled, while again only relative state measurements are available.

A comparison of the reference frames of agents for the three categories is shown in Figure 1.1 This thesis focuses on the second category of formation control systems, which, along with the third category are referred to as *relative formation control systems.*

Focus on relative formation control started with agents governed by single-integrator dynamics. One of the main reasons for this choice was that the formation control problem for multi-agent systems with single-integrator agents closely resembles consensus problems. Through extension of consensus algorithms, formations with translational degrees of freedom were hence among the first displacement-based formation control schemes proposed [21], [22].

From the consensus-derived translational formation control schemes, a desire arose to find a control scheme that allows for formations with freedom of rotation and scaling. As a result, the ordinary graph Laplacian that is used for consensus schemes was generalized to the complex [23], [24] and signed [24], [25] Laplacians. The type of formations



(a) Position-based

(b) Displacement-based

(c) Distance-based

Figure 1.1: An illustration comparing the reference frames for position-, displacement- and distance-based formation control systems.

proposed using these generalized Laplacians were dubbed the *similar* formation and the *affine* formation, respectively. The price for the extra degrees of freedom comes in the form of stricter conditions on the sensing graph $\mathcal{G}$: whereas for the translational formation control scheme, $\mathcal{G}$ having a spanning tree is sufficient (and necessary) to guarantee stability [22], stability of similar formation control systems requires 2-rootedness of the underlying graph [23], and affine formation control systems require universal rigidity [25].

Rigidity conditions and rotational degrees of freedom were generally seen in the domain of distance-based formation control. Hence, the similar and affine formation control schemes, which were proposed after the publication of [20], do not directly conform to either of the three categories as proposed in [20].

Since the introduction of the similar and affine formations, many different extensions and applications have been published. Some focus on time-varying sensing graphs [26], [27], others on different agent dynamics [10] or heterogeneous agents [28]. Of particular note are the extensions to rigid formations, proposed in the original publications [23], [25].

An overview of some relevant publications on relative formation control is given in Table 1.2.

1.3.2 Distributed Kalman Filtering

The history of filtering in control systems is extensive and will not be considered in much detail. The classic, widely known (and optimal) filter for linear Gauss-Markov models is the Kalman filter [33]. Following the popularity of decentralization and distributed systems, distributed adaptations of the Kalman filter were widespread.

Distributed localization in static systems is omnipresent in literature [34], [35], which is useful for applications such as sensor networks. However, localization in dynamic systems is altogether a different problem. For networks of mobile agents, [36] is of particular note for the extensive optimality analysis. For systems with relative observations with some anchors, [37] proposes a Kalman filter including an algorithm to compute the optimal Kalman gain subject to any sparsity pattern, i.e., for any arbitrary communication constraint. Lastly, in [38] additionally relative measurements with a set of

Reference	Formation Type	Global reference frame	Aligned orientation	Remarks
Lin et al. [29]	Translational	✗	✓	Consensus-based
Oh et al. [30]	Translational	✗	✗	Orientation alignment through consensus
Lin et al. [23]	Similar	✗	✓	Incl. extension to rigid formations
Lin et al. [25]	Affine	✗	✓	Incl. extension to rigid formations
Han et al. [27]	Rigid	✗	✓	Rigidity through leader-follower split
Xu et al. [28]	Affine	✗	✓	Heterogeneous agents
Dimarogonas et al. [31]	Rigid	✗	✗	Gradient-based, for tree graphs
Park et al. [32]	Rigid	✗	✗	Gradient-based, for complete graphs

Table 1.2: Overview of some relevant state of the art for relative formation control systems.

anchors are considered for a dynamical system. However, no control inputs are considered and hence the problem is purely that of distributed estimation. Despite including relative observations, [37] and [38], like [36], let agents track their absolute position with respect to some global reference frame (or anchor set). An anchorless alternative is hence still lacking.

1.4 Goals and Outline

The main goals of this thesis are to set up a data model for relative formation control systems with agents governed by single-integrator dynamics, and to design a distributed Kalman filter for these systems. Additionally, robustness of the filter under communication constraints is analyzed.

This thesis is structured as follows:

- In Chapter 2, some preliminary results on relative formation control are introduced, including notation standards.
- Chapter 3 introduces the affine formation control scheme as proposed in [25]. The extension to rigid formations and some benchmark simulations are also considered.
- In Chapter 4 a data model for a formation control system is motivated and introduced, followed by a Kalman filter designed from a local, single-edge perspective

on the filtering problem.

- In contrast with the local perspective, Chapter 5 introduces a global perspective on the filtering problem. First, a centralized filter for the global state-space model is considered. Then a distributed adaptation of this filter is proposed.

- Chapter 6 explores the robustness of the proposed filters under randomly failing communication links, or under a completely communication-less setting. A solution based on a Wiener predictor is introduced.

- The thesis is concluded by Chapter 7. Directions for future work are discussed, and a brief summary of contributions is included.

2 Preliminaries

This chapter provides a brief overview of the notation used throughout this report. Additionally, some preliminary theory is provided on graph theory, frameworks and their rigidity. For more extensive coverage of these subjects, the reader is referred to the various references in this chapter.

2.1 Notation

The notation standards for some of the commonly used symbols and operators throughout this thesis are described in Table 2.1.

2.2 Graph Theory

A graph is defined as a combination of a set of nodes and a set of edges between these nodes: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The node set $\mathcal{V} = \{1, 2, \dots, N\}$ indexes the N nodes. The edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, where equality implies a complete graph. The graphs considered in this

a	Local scalar
A	Global constant scalar
$\mathbf{a}$	Column vector
$\mathbf{A}$	Matrix
A_{ij}	Element on the ith row and jth column of matrix $\mathbf{A}$
$\mathcal{A}$	Set or graph
$\mathbf{1}_N$	Vector of all ones of length N
$\mathbf{0}_N$	Vector of all zeros of length N
$\mathbf{I}_N$	$N \times N$ identity matrix
$\mathbb{R}^N$	Set of N-dimensional real column vectors
$\mathbb{R}^{N \times M}$	Set of real matrices with N rows and M columns
$\mathbb{S}_+^N$	Set of real symmetric positive semi-definite $N \times N$ matrices
$\boldsymbol{\mathcal{N}}(0,1)$	Normal distribution with mean 0 and variance 1
$\otimes$	Kronecker product operator
$\circ$	Hadamard product operator
$\lVert \cdot \rVert$	Euclidean norm operator
$(\cdot)^\top$	Transpose operator
$\mathbb{E}(\cdot)$	Expectation operator
$\mathrm{tr}(\cdot)$	Trace operator
$\mathrm{bdiag}(\mathbf{A}_i)_{i \in \mathcal{S}}$	Block diagonal matrix with blocks $\mathbf{A}_i$ for all i in the set $\mathcal{S}$

Table 2.1: Notation standards used throughout the thesis.

thesis are considered to be undirected, meaning all edges are bidirectional: if and only if $(i, j) \in \mathcal{E}$, then $(j, i) \in \mathcal{E}$. The neighbor set of a node i is defined $\mathcal{N}_i = \{j : (j, i) \in \mathcal{E}\}$. The total number of undirected edges in the graph is denoted M. In addition, the *degree* of a node i is defined as the cardinality of the neighborhood of i, i.e., $M_i = |\mathcal{N}_i|$.

Next, a generalization of the graph Laplacian is defined.

Definition 2.1. *The generalized Laplacian* $\mathbf{L}$ *of an undirected graph* $\mathcal{G}$ *is an* N*-by-*N *matrix whose elements satisfy the following properties:*

$$L_{ij} = \begin{cases} -l_{ij} & \text{if } i \neq j \text{ and } j \in \mathcal{N}_i \\ 0 & \text{if } i \neq j \text{ and } j \notin \mathcal{N}_i \\ \sum\limits_{j \in \mathcal{N}_i} l_{ij} & \text{if } i = j \end{cases}$$

Note that the weights l_{ij} can be either positive or negative, in contrast with usual definitions of the Laplacian. In the remainder of this thesis, the adjective *generalized* is often omitted when talking about the generalized Laplacian. It is also worth mentioning that in some references (e.g., [39], [40]) the generalized Laplacian is referred to as the *stress matrix*.

Note that the Laplacian is a stochastic matrix, i.e., $\mathbf{L}\mathbf{1}_N = \mathbf{0}_N$. If the graph $\mathcal{G}$ is undirected the Laplacian is also symmetric: $L_{ij} = L_{ji}$. It is important to realize that the generalized Laplacian is not uniquely determined for a graph $\mathcal{G}$. As such, the set of all generalized Laplacians for an undirected graph $\mathcal{G}$ will be denoted $\mathcal{L}(\mathcal{G})$.

2.3 Configurations and Frameworks

In order to specify desired formations, we introduce the concepts of *configurations* of a graph. For a graph $\mathcal{G}$, a configuration $\mathbf{p} = [\mathbf{p}_1^\top, \mathbf{p}_2^\top, \dots, \mathbf{p}_N^\top]^\top \in \mathbb{R}^{DN}$ is a concatenated vector of vectors $\mathbf{p}_i \in \mathbb{R}^D$ relating node i of the graph to a location in D-dimensional Euclidean space. While the configuration vector $\mathbf{p}$ contains DN elements and thus is $\in \mathbb{R}^{DN}$, it is often said that a configuration is in $\mathbb{R}^D$ to indicate that the configuration is in D-dimensional Euclidean space.

Similar to [25], in this paper it will be assumed that all configurations are *generic.* A configuration is generic if the coordinates of its configuration do not satisfy any non-trivial algebraic equation with rational coefficients [40].

The combination of a graph $\mathcal{G}$ and a configuration $\mathbf{p}$ is called a *framework* $(\mathcal{G}, \mathbf{p})$.

2.4 Rigidity

The rigidity of frameworks is important to consider for formation control. Some definitions related to rigidity will be presented.

First, the concepts of *equivalence* and *congruence* will be defined.

Definition 2.2. *[25] Two frameworks in* $\mathbb{R}^D$ *with equal underlying graph* $\mathcal{G}$ *defined as* $(\mathcal{G}, \mathbf{p})$ *and* $(\mathcal{G}, \mathbf{q})$ *are equivalent if*

$$\|\mathbf{p}_i - \mathbf{p_j}\| = \|\mathbf{q}_i - \mathbf{q_j}\| \quad \forall \quad (i, j) \in \mathcal{E} \tag{2.1}$$

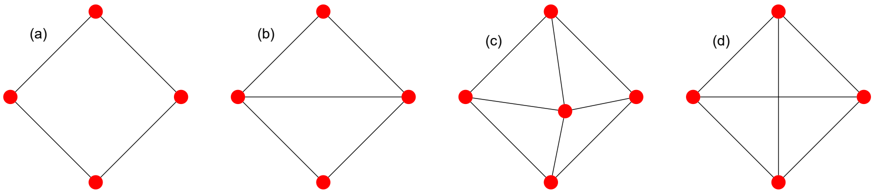


Figure 2.1: Four frameworks in $\mathbb{R}^2$ with increasing rigidity, from left to right.

In other words, equivalence means all edge lengths are equal.

Definition 2.3. *[25] Two frameworks in $\mathbb{R}^D$ with equal underlying graph $\mathcal{G}$ defined as $(\mathcal{G}, \mathbf{p})$ and $(\mathcal{G}, \mathbf{q})$ are congruent if*

$$\|\mathbf{p}_i - \mathbf{p_j}\| = \|\mathbf{q}_i - \mathbf{q_j}\| \quad \forall \quad i, j \in \mathcal{V} \tag{2.2}$$

That is, all inter-node lengths are equal. Two congruent frameworks are the same up to a rigid transformation (translation and rotation) in $\mathbb{R}^D$. Note that congruence implies equivalence by definition. Since congruence does not depend on the graph, it can be said to be a property of the configurations $\mathbf{p}$ and $\mathbf{q}$.

The definitions of equivalence and congruence will be used in turn for the rigidity definitions.

Definition 2.4. *[41] A framework $(\mathcal{G}, \mathbf{p})$ in $\mathbb{R}^D$ is locally rigid if an $\epsilon > 0$ exists such that for any configuration $\mathbf{q}$ in $\mathbb{R}^D$ with $|\mathbf{p} - \mathbf{q}| < \epsilon$, congruence between $(\mathcal{G}, \mathbf{p})$ and $(\mathcal{G}, \mathbf{q})$ must mean that $\mathbf{p} = \mathbf{q}$.*

In Figure 2.1 only framework (a) is not locally rigid. Instead, it is said to be *flexible*.

Definition 2.5. *[41] A framework $(\mathcal{G}, \mathbf{p})$ in $\mathbb{R}^D$ is globally rigid if for any configuration $\mathbf{q}$ in $\mathbb{R}^D$ for which $(\mathcal{G}, \mathbf{q})$ and $(\mathcal{G}, \mathbf{p})$ are equivalent, $\mathbf{p}$ and $\mathbf{q}$ are also congruent.*

Framework (b) in Figure 2.1 is not globally rigid since the bottom node can be flipped along a horizontal axis such that it is in the same location as the top node. This flipped framework is equivalent to the shown framework, but they are not congruent (since the distance between the top and bottom node is not preserved). Frameworks (c) and (d) are globally rigid.

Definition 2.6. *[25] A framework $(\mathcal{G}, \mathbf{p})$ in $\mathbb{R}^D$ is universally rigid if for any configuration $\mathbf{q}$ in $\mathbb{R}^s$ for which $(\mathcal{G}, \mathbf{q})$ and $(\mathcal{G}, \mathbf{p})$ are equivalent, $\mathbf{p}$ and $\mathbf{q}$ are also congruent, for any $s > 0$.*

Framework (c) is not universally rigid since if the formation is considered to live in 3-dimensional space, a framework can be found which is equivalent but not congruent. For example, the center node may be lifted out of the plane with all nodes still maintaining the same edge distances.

There are numerous useful corollaries which follow from these definitions. Note for example that any formation where $\mathcal{G}$ is a complete graph is by definition universally rigid. As such, framework (d) in Figure 2.1 is universally rigid. This can be used as a starting point for defining a class of graphs which are generically universally rigid (i.e., graphs whose frameworks are universally rigid for generic configurations) in $\mathbb{R}^D$, called *trilateration graphs* [42]. A D-trilateration graph is a graph which is constructed in the following way: The first $D+1$ nodes form a complete graph (i.e. a D-simplex). Every subsequent node is adjacent to at least $D+1$ other nodes. The class of trilateration graphs can be used to determine whether a framework is universally rigid or to construct frameworks that have this property.

2.5 Affine and Rigid Formations

Next, the concepts of affine and rigid formations are introduced. Before introducing affine formation control in the next chapter, we define the *affine image*. A subset of the affine image is the *rigid image*, which we will use when extending the affine formation control toward rigid formations.

Definition 2.7. *[25] The affine image of* $\mathbf{p}$ *is defined as*

$$\mathcal{A}(\mathbf{p}) = \left\{ (\mathbf{I}_N \otimes \mathbf{A})\mathbf{p} + \mathbf{1}_N \otimes \mathbf{a} : \mathbf{A} \in \mathbb{R}^{D\times D}, \mathbf{a} \in \mathbb{R}^D \right\}$$

From this definition, note that if we denote $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}$ via a singular matrix decomposition, we can see the transformation of $\mathbf{p}$ as a rotation $\mathbf{V}$, a scaling along axes $\boldsymbol{\Sigma}$, another rotation $\mathbf{U}$ and finally a translation $\mathbf{a}$. If $\mathbf{A}$ is orthogonal, all its singular values are 1 and we get $\boldsymbol{\Sigma} = \mathbf{I}_D$ and so the transformation is *rigid*, meaning distances between the nodes are conserved and the only degrees of freedom are rotation and translation. This leads to the definition of the *rigid image*.

Definition 2.8. *The rigid image of* $\mathbf{p}$ *is defined as*

$$\mathcal{R}(\mathbf{p}) = \left\{ (\mathbf{I}_N \otimes \mathbf{A})\mathbf{p} + \mathbf{1}_N \otimes \mathbf{a} : \mathbf{A} \in \mathbb{R}^{D\times D}, \mathbf{a} \in \mathbb{R}^D, \mathbf{A}^\top \mathbf{A} = \mathbf{I}_D \right\}$$

A position vector $\mathbf{z}$ is said to be an *affine formation* of $\mathbf{p}$ if it is in the affine image of $\mathbf{p}$: $\mathbf{z} \in \mathcal{A}(\mathbf{p})$. Similarly, $\mathbf{z}$ is a *rigid formation* of $\mathbf{p}$ if $\mathbf{z} \in \mathcal{R}(\mathbf{p})$. This last statement is equivalent to stating that $\mathbf{z}$ and $\mathbf{p}$ are congruent.

2.6 Procrustes Error

Given a set of node positions $\mathbf{z}$, it is useful to quantify how close the set of nodes is to the rigid image $\mathcal{R}(\mathbf{p})$. For this purpose, we define the Procrustes error. The Procrustes method of determining the distance between $\mathbf{z}$ and $\mathcal{R}(\mathbf{p})$ can be seen as finding the projection $\mathbf{z}^*$ of $\mathbf{z}$ on $\mathcal{R}(\mathbf{p})$ and then computing the distance between $\mathbf{z}$ and $\mathbf{z}^*$. In detail, it works as follows: First a centering matrix is applied to the positions of the nodes to remove any translational degrees of freedom. Following that, the optimal rotation is found using the singular value decomposition solution to the orthogonal Procrustes problem [43]. The scalar Procrustes error is then defined as the Frobenius norm of the remaining difference between the configuration and optimally translated and rotated positions, normalized by the number of nodes.

Definition 2.9. *The Procrustes error e is defined as*

$$\begin{aligned} e = \min_{\mathbf{t},\boldsymbol{\Omega}} \quad & \frac{1}{N}\|\mathbf{Z}\boldsymbol{\Omega} + \mathbf{1}_N\mathbf{t}^\top - \mathbf{P}\|_F \\ s.t. \quad & \boldsymbol{\Omega}^\top\boldsymbol{\Omega} = \mathbf{I}_D \end{aligned}$$

where $\mathbf{Z} \in \mathbb{R}^{N\times D}$ and $\mathbf{P} \in \mathbb{R}^{N\times D}$ represent the node positions $\mathbf{z}$ and configuration $\mathbf{p}$ rearranged in matrix form, respectively. $\mathbf{t} \in \mathbb{R}^D$ and $\boldsymbol{\Omega} \in \mathbb{R}^{D\times D}$ respectively denote the translation vector and orthogonal rotation matrix.

Relative Formation Control 3

In this chapter, we consider a relative formation control problem in D-dimensional Euclidean space. The methods of [25] are used to present a local control law which allows a multi-agent system to converge to a rigid formation.

First, in Section 3.1 the problem context is introduced, including the dynamics of the set of homogeneous agents and the structure of the local control law. After the affine formation control problem is formulated in Section 3.2, necessary and sufficient conditions for the realizability and stability of the closed-loop system are given in Section 3.3. The control law weight design is discussed in Section 3.4, followed by the extension from affine to rigid formations in Section 3.5. To conclude the chapter, simulations in Section 3.6 show convergence of a set of randomly initialized agents to a rigid formation.

3.1 Agent Dynamics and Control Law

Before the problem can be formulated, first the dynamics of the agents and the structure of the control law will be set up. As discussed in Chapter 2, the desired configuration $\mathbf{p}$ denotes the desired formation of agents up to a rigid transformation. The locations of the agents are denoted by the vector $\mathbf{z} = [\mathbf{z}_1^\top, \mathbf{z}_2^\top, \dots, \mathbf{z}_N^\top]^\top \in \mathbb{R}^{DN}$, with $\mathbf{z}_i \in \mathbb{R}^D$ a vector denoting the location of agent i in D-dimensional Euclidean space. The locations of the agents over time are determined by single-integrator dynamics:

$$\dot{\mathbf{z}}_i = \mathbf{u}_i \quad i = 1, \dots, N \tag{3.1}$$

with $\mathbf{u}_i \in \mathbb{R}^D$ the control variable for agent i. The graph $\mathcal{G}$ denotes the sensing graph of the agents, meaning agent i can measure the relative positions of its neighbors $j \in \mathcal{N}_i$. As such, agent i has access to $\mathbf{z}_i - \mathbf{z}_j \quad j \in \mathcal{N}_i$. The control law should be a function of these measurements. Hence, the control law has the following structure.

$$\mathbf{u}_i = -\sum_{j \in \mathcal{N}_i} l_{ij}(\mathbf{z}_i - \mathbf{z}_j) \tag{3.2}$$

which is linear and locally computable for each agent. l_{ij} represent weights, which we will see correspond with the elements of the generalized Laplacian. The multi-agent system from the perspective of agent i is visualized in Figure 3.1.

The agent dynamics (3.1) and the control law (3.2) of all agents together can be aggregated in vectors to arrive at the closed-loop system in matrix form, written in terms of the generalized Laplacian:

$$\dot{\mathbf{z}} = -(\mathbf{L} \otimes \mathbf{I}_D)\mathbf{z} \tag{3.3}$$

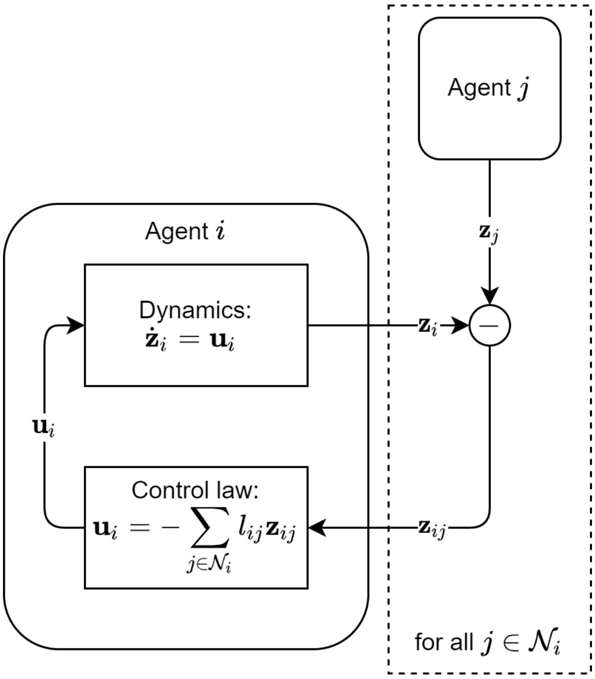

Agent j
$\mathbf{z}_j$
Agent i
Dynamics:
$\dot{\mathbf{z}}_i = \mathbf{u}_i$
$\mathbf{z}_i$
$-$
$\mathbf{u}_i$
Control law:
$\mathbf{u}_i = -\sum_{j \in \mathcal{N}_i} l_{ij} \mathbf{z}_{ij}$
$\mathbf{z}_{ij}$
for all $j \in \mathcal{N}_i$

Figure 3.1: Multi-agent system from the perspective of agent i.

3.2 Problem Formulation

With the definition of affine formations given in Section 2.5 and the agent dynamics and control law as defined in Section 3.1, a formulation is given for the affine formation control problem. The rigid formation control problem will be discussed as an extension of the affine case in Section 3.5.

Given a group of agents in a framework described by the graph $\mathcal{G}$ and the configuration $\mathbf{p}$, with dynamics described by (3.1). A distributed linear control law of the form of (3.2) should be applied for which

(i) $\mathcal{A}(\mathbf{p})$ equals the equilibrium set of the closed-loop system;

(ii) for any initial position vector $\mathbf{z}$ the closed-loop system converges to an equilibrium in $\mathcal{A}(\mathbf{p})$.

The two conditions are referred to as the *realizability* condition and the *stabilizability* condition, respectively. The variables open to design are the weight variables l_{ij} of the generalized Laplacian. In the following subsection, such a Laplacian will be shown to always exist such that both the realizability and the stabilizability conditions are satisfied, under certain conditions on the sensing graph. This will be done via the methods shown in [25].

3.3 Conditions for Realizability and Stabilizability

First, the realizability condition is considered. From observing the closed-loop system (3.3) it is obvious that the equilibrium set of the system is the set of points $\mathbf{z}$ satisfying

$(\mathbf{L} \otimes \mathbf{I}_D)\mathbf{z} = \mathbf{0}_{DN}$. As such, the equilibrium set of the closed-loop system is the null space of $(\mathbf{L} \otimes \mathbf{I}_D)$. The realizability condition is satisfied if the equilibrium set equals the affine image of the configuration: $(\mathbf{L} \otimes \mathbf{I}_D)\mathbf{z} = \mathbf{0}_{DN}$ if and only if $\mathbf{z} \in \mathcal{A}(\mathbf{p})$. We will see that a Laplacian satisfying this condition always exists if the underlying framework is globally rigid, formalized by the following theorem. The theorem is followed by an outline of the proof.

Theorem 3.1. *Given an undirected graph $\mathcal{G}$ with $N \geq D+2$ and a generic configuration $\mathbf{p}$ in $\mathbb{R}^D$. An affine formation of $\mathbf{p}$ is realizable over $\mathcal{G}$ if and only if $(\mathcal{G}, \mathbf{p})$ is globally rigid.*

Proof. For the proof, let us start with a result presented in [40] and [41] which states that for an undirected graph $\mathcal{G}$ with $N \geq D + 2$ and a generic configuration $\mathbf{p}$ in $\mathbb{R}^D$, a matrix $\mathbf{L} \in \mathcal{L}(\mathcal{G})$ satisfying $(\mathbf{L} \otimes \mathbf{I}_D)\mathbf{p} = \mathbf{0}_{DN}$ exists whose rank is $N - D - 1$ if and only if the framework $(\mathcal{G}, \mathbf{p})$ is globally rigid.

The condition $(\mathbf{L} \otimes \mathbf{I}_D)\mathbf{p} = \mathbf{0}_{DN}$ which is satisfied by such a Laplacian implies that $\mathbf{p}$ is an equilibrium of (3.3). If $\mathbf{p}$ is an equilibrium of (3.3), then all points in $\mathcal{A}(\mathbf{p})$ are also equilibria:

$$\begin{aligned}(\mathbf{L} \otimes \mathbf{I}_D)[(\mathbf{I}_N \otimes \mathbf{A})\mathbf{p} + \mathbf{1}_N \otimes \mathbf{a}] &= (\mathbf{L} \otimes \mathbf{A})\mathbf{p} \\ &= (\mathbf{I}_N \otimes \mathbf{A})(\mathbf{L} \otimes \mathbf{I}_D)\mathbf{p} \\ &= \mathbf{0}_{DN}\end{aligned}$$

for any $\mathbf{A} \in \mathbb{R}^{D \times D}$ and $\mathbf{a} \in \mathbb{R}^D$. As a result, we know that the affine image of $\mathbf{p}$ is a subset of the equilibrium set on the condition that the framework is globally rigid.

To complete the proof, the last step is to prove under which conditions the affine image of $\mathbf{p}$ equals the equilibrium set, i.e., we need to prove *all* equilibria are in the affine image of $\mathbf{p}$. We use the fact that if the coordinate vectors of $\mathbf{p}$ span $\mathbb{R}^D$, then $\mathcal{A}(\mathbf{p})$ is a linear subspace of dimension $D^2 + D$ as proven in [25]. From the result above on the existence of $\mathbf{L}$, we know that for a generic configuration $\mathbf{p}$, the null space of $\mathbf{L} \otimes \mathbf{I}_D$ has dimension $(D + 1)D$. Given that this null space has the same dimension as the linear subspace $\mathcal{A}(\mathbf{p})$ combined with the fact that $\mathcal{A}(\mathbf{p})$ is a subset of the equilibrium set, we can conclude that the equilibrium set and $\mathcal{A}(\mathbf{p})$ are equal, as long as $\mathcal{G}$ is globally rigid. This concludes the proof. □

The next condition to satisfy is the stabilizability condition. We will see that for this, a graphical condition even stronger than global rigidity is necessary and sufficient. The following theorem states that stabilizability of the continuous-time closed-loop system is guaranteed if and only if the framework $(\mathcal{G}, \mathbf{p})$ is universally rigid.

Theorem 3.2. *Given an undirected graph $\mathcal{G}$ with $N \geq D + 2$ and generic configuration $\mathbf{p}$ in $\mathbb{R}^D$. An affine formation of $\mathbf{p}$ is stabilizable over $\mathcal{G}$ if and only if $(\mathcal{G}, \mathbf{p})$ is universally rigid.*

Proof. We will extend on the relation between the existence of a Laplacian and rigidity of the graph via a result in [39]. For an undirected graph $\mathcal{G}$ with $N \geq D + 2$ and generic configuration $\mathbf{p}$ in $\mathbb{R}^D$, a positive semi-definite Laplacian $\mathbf{L} \in \mathcal{L}(\mathcal{G})$ exists with rank $N - D - 1$ if and only if the framework $(\mathcal{G}, \mathbf{p})$ is universally rigid. Notice the addition of positive semi-definiteness of the Laplacian when the global rigidity condition is tightened to universal rigidity. Positive semi-definiteness of $\mathbf{L}$ guarantees that its eigenvalues are all positive apart from the $D + 1$ zero eigenvalues. The Kronecker product with the identity matrix $\mathbf{I}_D$ only increases the multiplicity of the eigenvalues. As a result, the closed-loop system (3.3) has all eigenvalues in the left half-plane, and is thus asymptotically stable for continuous-time systems. □

3.4 Laplacian Weight Design

Given that a framework is universally rigid, we now know that a Laplacian exists for which the resulting system is stable and converging to an affine formation of the configuration. In this section, the design method proposed in [25] is explained which actually computes this Laplacian.

First, we introduce the vectors $\mathbf{q}_1, \ldots, \mathbf{q}_D \in \mathbb{R}^N$ with $\mathbf{q}_i$ representing the ith element of the configuration vector $\mathbf{p}_i$. For a generic configuration $\mathbf{p}$, the vectors $\mathbf{q}_1, \ldots, \mathbf{q}_D, \mathbf{1}_N$ are all linearly independent. As such, we introduce an orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{(N-D-1)\times N}$ whose orthonormal rows are orthogonal to $\mathbf{q}_1, \ldots, \mathbf{q}_D, \mathbf{1}_N$.

Second, we will write the Laplacian as a function of the weights to be designed. With $\mathbf{w} \in \mathbb{R}^M$ a vector containing the edge weights, the Laplacian can be written as follows:

$$\mathbf{L} = \mathbf{B}\text{diag}(\mathbf{w})\mathbf{B}^\top \tag{3.4}$$

where $\mathbf{B} \in \{-1, 0, 1\}^{N\times M}$ represents the standard incidence matrix of the underlying sensing graph. We now have three constraints for the Laplacian:

(i) The Laplacian should correspond to the underlying graph: $\mathbf{L} \in \mathcal{L}(\mathcal{G})$ which is satisfied if $\mathbf{L} = \mathbf{B}\text{diag}(\mathbf{w})\mathbf{B}^\top$.

(ii) The smallest eigenvalue not corresponding to one of the $D + 1$ zero eigenvalues should be positive: $\lambda_{\min}(\mathbf{QLQ}^\top) > 0$.

(iii) The configuration vectors $\mathbf{q}_1, \ldots, \mathbf{q}_D$ should be in the null space of the Laplacian: $\mathbf{Lq}_i = \mathbf{0}_N$ for all $i = 1, \ldots, D$.

An optimization problem incorporating these constraints can be set up as follows:

$$\begin{aligned} \max_{\mathbf{w}} \quad & \lambda \\ \text{s.t.} \quad & \mathbf{L} = \mathbf{B}\text{diag}(\mathbf{w})\mathbf{B}^\top \\ & 0 < \lambda < \bar{\lambda} \\ & \mathbf{QLQ}^\top \succ \lambda \mathbf{I}_{N-D-1} \\ & \mathbf{Lq}_i = \mathbf{0}_N \quad i = 1, \ldots, D \end{aligned} \tag{3.5}$$

which is a semi-definite program and thus convex. $\bar{\lambda}$ can be any positive constant and is added to make sure the solution to the problem is bounded. As per Theorem 3.2, we know that if the framework is universally rigid, then a solution is guaranteed to exist.

3.5 Extension to Rigid Formations

In many applications, a rigid formation (which allows only rotations and translations with respect to the configuration) is more favorable than an affine formation. Fortunately, the affine formation control law described above can be extended to force the system to converge to a rigid formation (i.e., congruent to the configuration). The concept is simple: if we force a subset $\mathcal{M} \subseteq \mathcal{V}$ of $D+1$ agents to keep to a rigid formation, the complete formation will also be rigid, even when all other agents $\mathcal{V} \setminus \mathcal{M}$ follow control law (3.2). This is formalized by the following theorem.

Theorem 3.3. *[25] Given a generic configuration $\mathbf{p}$ in $\mathbb{R}^D$. A configuration $\mathbf{z} \in \mathcal{A}(\mathbf{p})$ is congruent to $\mathbf{p}$ if and only if there exists a set $\mathcal{M}$ of at least $D+1$ agents such that the dimension of the convex hull of $\{\mathbf{z}_i : i \in \mathcal{M}\}$ is D and*

$$\|\mathbf{z}_i - \mathbf{z}_j\| = \|\mathbf{p}_i - \mathbf{p}_j\| \quad \forall \quad i, j \in \mathcal{M} \tag{3.6}$$

Proof. Note that for an affine image $\mathbf{z}_i = \mathbf{A}\mathbf{p}_i + \mathbf{a}$ for all i, (3.6) implies that $\mathbf{A}$ must be unitary:

$$\mathbf{A}^\top \mathbf{A} = \mathbf{I}_D \tag{3.7}$$

As such, for any pair (i, j) of agents

$$\|\mathbf{z}_i - \mathbf{z}_j\| = (\mathbf{p}_i - \mathbf{p}_j)^\top \mathbf{A}^\top \mathbf{A}(\mathbf{p}_i - \mathbf{p}_j) = \|\mathbf{p}_i - \mathbf{p}_j\| \tag{3.8}$$

and thus $\mathbf{z}$ is congruent to $\mathbf{p}$, i.e., $\mathbf{z} \in \mathcal{R}(\mathbf{p})$. □

Note that this property, which is often referred to as a *leader-follower* split, has been exploited in a number of papers (e.g., [27], [44]).

Also worth noting is that for trilateration graphs, the $D+1$ initial agents forming a simplex in D-dimensional space can be appointed the leaders. Since these agents form a complete subgraph, the distance-based control law proposed in [32] can be used, which guarantees a $\mathcal{K}_{D+1}$ (i.e., the complete graph of $D+1$ nodes) formation in $\mathbb{R}_D$ converges to a rigid formation.

Note finally that while the leader-follower split takes away from the homogeneity of agents, it scales particularly well for an increasing amount of agents, since the number of leader agents is constant at $D+1$ (which in real life means 4 agents at most for three-dimensional formations).

3.6 Simulations

Simulations were set up to illustrate the formation control proposed in [25]. The simulations consider agents governed by single-integrator dynamics (3.1) and local linear control laws (3.2). A subset of $D+1$ leader agents forming a complete subgraph are given a different control law to make sure the formation converges to a rigid formation rather than an affine formation as discussed in Section 3.5.

Agents have access to relative position measurements with respect to their neighbors, as characterized by the framework. This framework is defined as a hexagonal formation

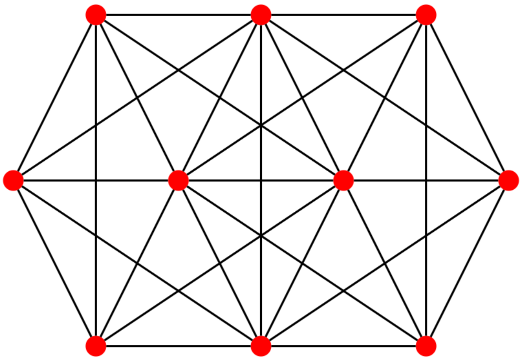


Figure 3.2: Hexagonal framework in $\mathbb{R}^2$.

in $\mathbb{R}^2$ consisting of $N = 10$ agents, as shown in Figure 3.2. The framework is universally rigid as its graph is a 2-trilateration graph.

Pseudocode for the affine formation control simulation is shown in Algorithm 1. Initial positions of agents were randomly drawn from a normal distribution $\boldsymbol{\mathcal{N}}(\boldsymbol{\mu}, \mathbf{P})$ with $\boldsymbol{\mu} = \mathbf{0}_D$ and $\mathbf{P} = \mathbf{I}_D$. For every iteration k, the control law (3.2) is applied[1]. Note also that the simulation is performed in discrete-time. We have seen that conditions for realizability and stability are for continuous-time dynamical systems. When introducing measurements and digital simulations, the step to discrete-time is inevitable. To preserve the stability of the simulated system, the time steps Δt of the simulation need to be chosen sufficiently small. For the single-integrator dynamics the discrete-time equivalent to $\dot{\mathbf{z}} = \mathbf{u}$ is $\mathbf{z}_{k+1} = \mathbf{z}_k + \Delta t \mathbf{u}_k$. For Algorithm 1 and all subsequent simulations we use $\Delta t = 0.001$. Algorithm 1 was run 50 times, and the results are shown in Figure 3.3. In Figure 3.3 the mean Procrustes error for varying initial positions is

Algorithm 1 Relative formation control

```
Constants: $N, D, \Delta t, \boldsymbol{\mu}, \mathbf{P}$
Known local constants: $l_{ij}$ for $i = 1, \dots, N$, $j \in \mathcal{N}_i$
Draw initial positions $\mathbf{z}_1^i$ from $\boldsymbol{\mathcal{N}}(\boldsymbol{\mu}, \mathbf{P})$
$k = 1$
while $k \leq k_{max}$ do
    for $i \in \mathcal{V}$ do
        $\mathbf{u}_k^i = -\sum_{j \in \mathcal{N}_i} l_{ij}(\mathbf{z}_k^i - \mathbf{z}_k^j)$        ▷ Control law
        $\mathbf{z}_{k+1}^i = \mathbf{z}_k^i + \Delta t \mathbf{u}_k^i$        ▷ Dynamics update
    end for
    $k = k + 1$
end while
```

[1] Although not specified in Algorithm 1, a set of $D + 1$ leader nodes use a different local control law to guarantee the rigidity of the system.

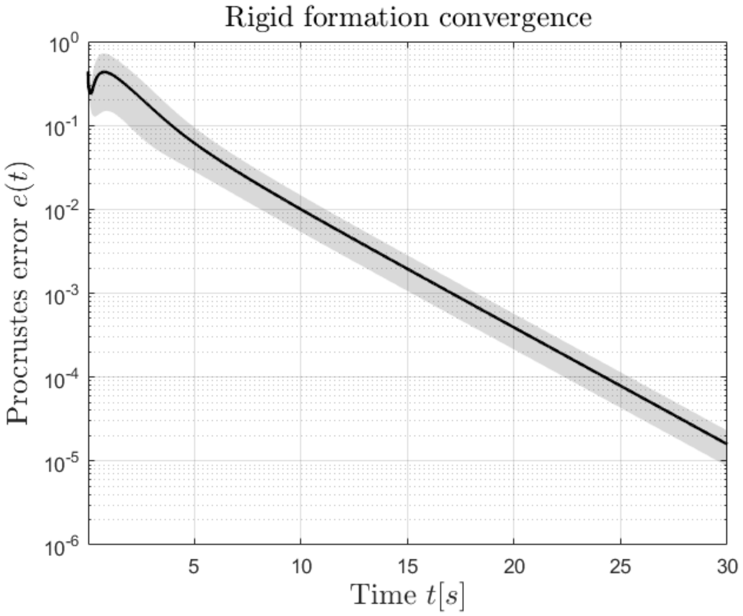


Figure 3.3: Convergence towards rigid formation over time for Algorithm 1.

plotted over a period of 30 seconds ($k_{max} = 30000$). The ±1 standard deviation regions are also shown. Figure 3.4 additionally shows the paths of individual agents and the final formation in $\mathbb{R}^2$, for a single run of the simulation. We notice that the Procrustes error decreases exponentially (note the logarithmic scale), after the initial 2 seconds where the Procrustes error is still unstable when agents rearrange themselves after their random initial positions. Clearly visible is that the final positions of the agents conform with a rigid transformation of the framework.

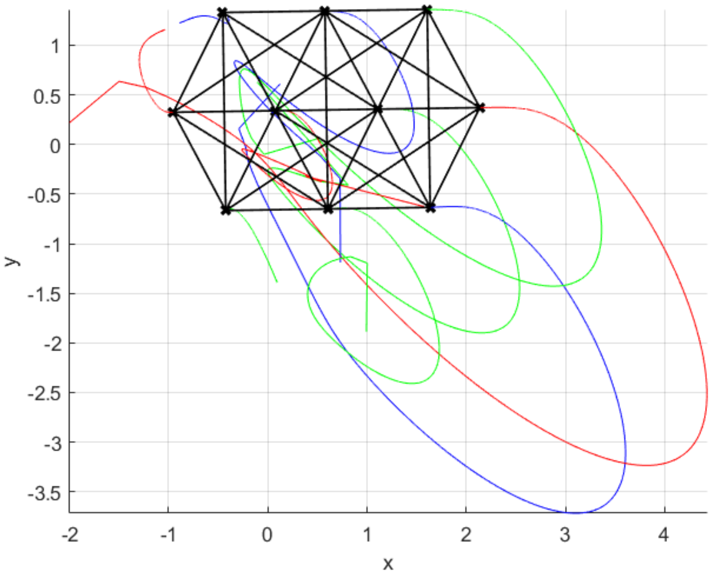


Figure 3.4: Paths of agents in $\mathbb{R}^2$ over time for Algorithm 1.

Local Approach to Edge State Filtering 4

In Chapter 3 the affine and rigid relative formation control problems were explored. Necessary and sufficient conditions were given for the convergence of a multi-agent system governed by single-integrator dynamics. When applying the proposed solution in real-life systems however, we have to deal with uncertainty. In control theory, uncertainty is a broad term, which is generally considered when a system is not perfectly known. This can e.g. be due to unknown deterministic system parameters or unmodeled dynamics. Alternatively, uncertainty is introduced in a system as randomness, i.e. statistical noise. This thesis focuses on uncertainty of the latter form.

In Section 4.1 a motivation is provided for modelling statistical uncertainty in formation control systems. Subsequently, in Section 4.2 a data model is proposed which attempts to model statistical uncertainty in relative formation control systems in two forms: measurement noise and process noise. Given sequences of noise-corrupted relative position measurements, we aim to optimally track the movement of the agents over time. This is a so-called *filtering* problem. In this chapter, we will approach the filtering problem from a local perspective, i.e., we consider a single arbitrary agent i which attempts to track its relative position to an arbitrary neighbor j. We start with an optimal instantaneous estimator (the MLE) in Section 4.3 and then move on to recursive Bayesian estimation (the MMSE estimator) in Section 4.4 and finally, a Kalman filter is proposed in Section 4.5. Contrary to many Kalman filters in literature (e.g., [36], [37]), the proposed Kalman filter operates on the *edges* of the sensing graph. Hence, the filter is referred to as the *edge-based Kalman filter*. Step by step, we will explore the performance of the considered methods via simulations, whose setup is explained in Section 4.2.3.

4.1 Motivation for Statistical Noise Modelling

Statistical uncertainty can be split into two categories. Measurement noise acts on the observations of an agent, while process noise acts on the dynamics of the agent. The motivations behind modelling each of these sources of uncertainty for relative formation control systems are discussed in order.

4.1.1 Motivation for Measurement Noise Modelling

The first form of modelled uncertainty is measurement noise. Perfect sensors don't exist, and as such most measurements are modelled as the sum of the true quantity to be measured and some noise drawn from a probability distribution function modelled after the sensor. If agents are homogeneous and carry the same sensors, the measurement noise can often be modelled via the same probability distribution function.

Additionally, for many types of sensors measurement noise is uncorrelated over time, i.e., the noise realizations on two measurements taken in succession are statistically independent.

On the other hand, noise on Cartesian relative position coordinates is more often correlated between the $x-$, $y-$ (and $z-$) coordinates. Consider for example a system where relative position measurements are done based on bearing and range measurements. The transformation from polar or spherical coordinates to Cartesian coordinates introduces correlation between the dimensions, even when the original range and bearing measurements were independent. We will see this via the following example.

Let us consider a simple $2D$ scenario showing that relative position estimations in Cartesian coordinates are correlated when combining range and bearing measurements. We assume the measurements of a range and bearing sensor to be normally distributed around the true value and independent with respect to each other:

$$\begin{bmatrix} R \\ \theta \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_R \\ \mu_\theta \end{bmatrix}, \begin{bmatrix} \sigma_R^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \right) \tag{4.1}$$

We then transform to Cartesian coordinates:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} R\cos(\theta) \\ R\sin(\theta) \end{bmatrix} \tag{4.2}$$

The Cartesian random variables X and Y are not independent, nor are they jointly Gaussian. The expected values of the Cartesian variables are

$$\mathbb{E}\left(\begin{bmatrix} X \\ Y \end{bmatrix} \right) = \mathbb{E}\left(\begin{bmatrix} R\cos(\theta) \\ R\sin(\theta) \end{bmatrix} \right) = \begin{bmatrix} \mu_R\cos(\mu_\theta)e^{-\frac{\sigma_\theta^2}{2}} \\ \mu_R\sin(\mu_\theta)e^{-\frac{\sigma_\theta^2}{2}} \end{bmatrix} \tag{4.3}$$

and the covariance matrix is given by

$$\mathrm{Cov}\left(\begin{bmatrix} X \\ Y \end{bmatrix} \right) = \begin{bmatrix} \mathrm{Var}(X) & \mathrm{Cov}(X,Y) \\ \mathrm{Cov}(X,Y) & \mathrm{Var}(Y) \end{bmatrix} \tag{4.4}$$

with

$$\begin{aligned} \mathrm{Var}(X) &= (\mu_R^2 + \sigma_R^2)(\frac{1}{2} + \frac{1}{2}e^{-2\sigma_\theta^2}\cos(2\mu_\theta)) - \mu_R^2\cos^2(\mu_\theta)e^{-\sigma_\theta^2} \\ \mathrm{Cov}(X,Y) &= (\mu_R^2 + \sigma_R^2)\frac{1}{2}\sin(2\mu_\theta)e^{-2\sigma_\theta^2} - \mu_R^2\frac{1}{2}\sin(2\mu_\theta)e^{-\sigma_\theta^2} \\ \mathrm{Var}(Y) &= (\mu_R^2 + \sigma_R^2)(\frac{1}{2} - \frac{1}{2}e^{-2\sigma_\theta^2}\cos(2\mu_\theta)) - \mu_R^2\sin^2(\mu_\theta)e^{-\sigma_\theta^2} \end{aligned}$$

which for this very simple case already grows rather complex. The main takeaway however is that the covariance matrix is no longer diagonal. In three-dimensional space, where two bearing measurements are needed, this covariance matrix grows even more complex. If solid assumptions can be made on the noise distribution of the range and bearing measurements, then the covariance matrix transformed to relative positions in

Euclidean space can be exploited when aiming to optimally estimate relative positions, as we will see in this chapter.

It is worth noting that in real life, relative positions are never measured directly. In many cases, range and bearing measurements are fused to find the relative position (e.g., [45], [46]). Range sensors are plentiful and exist for a variety of different applications. Laser-, radar- sound- or received signal strength-based sensors can all measure inter-agent distances with varying degree of accuracy, depending on the application. Bearing-only sensors include omnidirectional cameras. Some sensors can measure both bearing and range, such as laser rangefinders, stereo vision, radar, lidar and sonar [47], [48]. Note that for these sensors still, the range and bearing should be seen as separate, independent measurements, rather than a direct relative position measurement.

4.1.2 Motivation for Process Noise Modelling

The second form of random uncertainty appears in real-life systems in the form of random perturbations originating in the environment. These perturbations enter a system via the dynamics of agents, e.g., in UAV systems wind affects the movement of agents [14]. The disturbance can be modelled via some process noise source modelled after the type of disturbance it represents.

Many environmental disturbances are spatially and temporally correlated. A single gust of wind, for example can affect a group of UAVs over a period of time. Research on spatial and temporal correlation research in wind models is extensive and mostly data-based, see e.g. [49]. As for an example in a different application domain, the dynamics of a group of satellites may be disturbed by a period of increased solar radiation pressure [50], while AUVs experience disturbances in the form of oceanic currents.

It is a small step to see that disturbances acting on agents are consequently also often correlated. If one agent is disturbed by a gust of wind, its neighbor is likely to suffer a similar disturbance. Figure 4.1 visualizes this inter-agent disturbance correlation as a result of a spatially correlated environmental disturbance source. The disturbances acting on agents that are spatially close to each other are generally more highly correlated than disturbances between agents with a large distance between them. The assumption that environmental disturbances are often correlated between agents is key to the results of this thesis. Knowledge about the inter-agent disturbance correlation will be exploited by agents in counteracting the detrimental effect of these disturbances and keeping the multi-agent system in formation.

4.2 Data model

We consider two types of statistical noise sources affecting the formation control system. The first type is that of measurement noise. Since we consider the formation control problem in the context of relative navigation, i.e., agents measure only relative positions with respect to neighbors, the measurement noise can be modelled on the edges of the bidirectional sensing graph. The second type of modelled uncertainty is process noise, which models random disturbances on the dynamics of agents. Since the agents are represented by nodes in the sensing graph, this type of uncertainty is modelled on

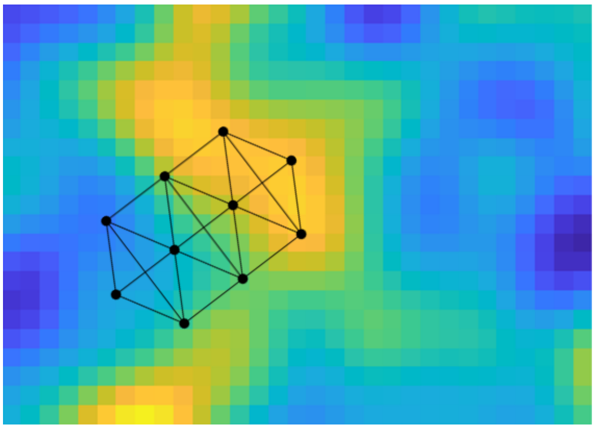


Figure 4.1: A multi-agent system in a spatial random field representing spatially correlated environmental disturbances.

the nodes of the sensing graph. In this section, an exact model for both types of uncertainties is presented.

4.2.1 Measurement Model

As a first step, uncertainty in the form of measurement noise is modelled. A multi-agent system is considered with agents performing relative position measurements with respect to its neighbors. The relative position between agents i and j can, in a more general sense, be considered to be the 'state' of edge (i, j), and it will be referred to as such. Additive noise is introduced on the edge states, denoted as $\mathbf{v}_{ij} \in \mathbb{R}^D$. The measurements $\mathbf{y}_{ij}$ available to agent i are modelled as follows:

$$\begin{aligned} \mathbf{y}_{ij} &= \mathbf{z}_i - \mathbf{z}_j + \mathbf{v}_{ij} \quad i = 1, \dots, N \quad j \in \mathcal{N}_i && (4.5) \\ &= \mathbf{z}_{ij} + \mathbf{v}_{ij} \quad i = 1, \dots, N \quad j \in \mathcal{N}_i && (4.6) \end{aligned}$$

where the edge state vector $\mathbf{z}_{ij} = \mathbf{z}_i - \mathbf{z}_j$ is defined for more compact notation. The probability distribution function of the noise source is characterized by both agents i and j. For some special cases, the noise source may be argued to be characterized only by the sensor on agent i, e.g., when the sensor is a camera or laser rangefinder. For homogeneous agents, one may even argue that the noise probability distribution is the same for all agent combinations. In the most general case however, the measurement is a function of both agents, e.g., in two-way ranging methods. As such, this more general case is chosen for the model. A block diagram showing where measurement noise enters the system is shown in Figure 4.2.

Measurements are assumed to be independent with respect to other measurements, both in time (i.e. measurements taken at different time instances are independent) and between different agents (i.e. measurements between agents i and j are independent of measurements between agents k and l, for any i,j,k and l).

We will assume the noise $\mathbf{v}_{ij}$ is Gaussian with zero mean and known covariance matrix $\mathbf{R}_{ij} \in \mathbb{S}_+^D$, which characterizes correlation between the D Euclidean dimensions.

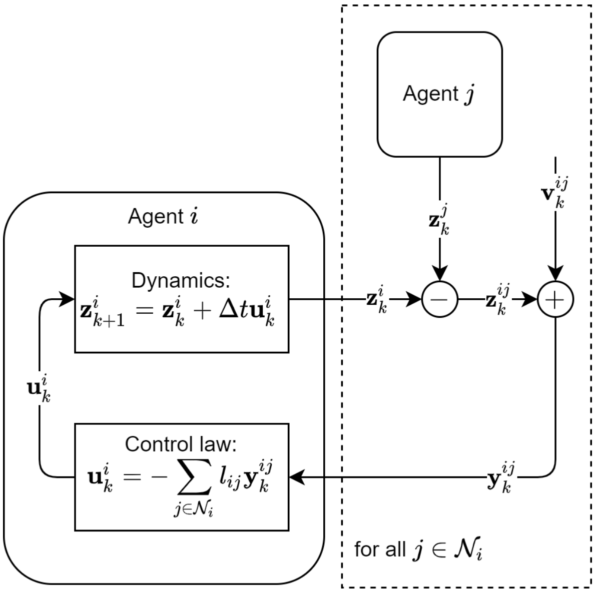

Agent j
$\mathbf{v}_k^{ij}$
$\mathbf{z}_k^j$
Agent i
Dynamics:
$\mathbf{z}_{k+1}^i = \mathbf{z}_k^i + \Delta t \mathbf{u}_k^i$
$\mathbf{z}_k^i$
$-$
$\mathbf{z}_k^{ij}$
$+$
$\mathbf{u}_k^i$
Control law:
$\mathbf{u}_k^i = -\sum_{j \in \mathcal{N}_i} l_{ij} \mathbf{y}_k^{ij}$
$\mathbf{y}_k^{ij}$
for all $j \in \mathcal{N}_i$

Figure 4.2: Measurement noise model.

$$\mathbf{v}_{ij} \sim \mathcal{N}(\mathbf{0}_D, \mathbf{R}_{ij}) \quad i = 1, \ldots, N \quad j \in \mathcal{N}_i \tag{4.7}$$

To further generalize the model, we may assume that the agents have access to multiple measurements per control update. We consider that agent i can perform T measurements over a time period indexed by k, in which $\mathbf{z}_k^{ij}$ is assumed to remain constant. Consider $\mathbf{y}_k^{ij} \in \mathbb{R}^{DT}$ as the measured relative positions between agent i and agent j, aggregated over T measurements. We will write the model for the measurements in the form of a Linear Gaussian model:

$$\mathbf{y}_k^{ij} = \mathbf{H}\mathbf{z}_k^{ij} + \mathbf{v}_k^{ij} \tag{4.8}$$

With $\mathbf{H} = \mathbf{1}_T \otimes \mathbf{I}_D \in \mathbb{R}^{DT \times D}$. $\mathbf{v}_k^{ij} \in \mathbb{R}^{DT}$ is a stacked vector of T realizations of noise modelled as in (4.7), i.e. zero-mean Gaussian and correlated with regard to dimension, but uncorrelated with regard to time: $\tilde{\mathbf{R}}_{ij} = \mathbf{I}_T \otimes \mathbf{R}_{ij}$.

4.2.2 Process Model

In addition to measurement errors, another source of uncertainty are unknown perturbations on the dynamics of the agents. This uncertainty is modelled as process noise $\mathbf{w}_i$ acting on the dynamics update of agent i (see Figure 4.3):

$$\mathbf{z}_{k+1}^i = \mathbf{z}_k^i + \Delta t \mathbf{u}_k^i + \mathbf{w}_k^i \quad i = 1, \ldots, N \tag{4.9}$$

Random disturbances are often correlated among agents because they originate in the environment in which the agents operate, as discussed in Section 4.1. Therefore the process noise will be modelled as a random vector $\mathbf{w}_k = \{\mathbf{w}_k^i\}_{i=1}^N \in \mathbb{R}^{DN}$ which stacks the process noises of all agents. Like the measurement noise model, the process noise is

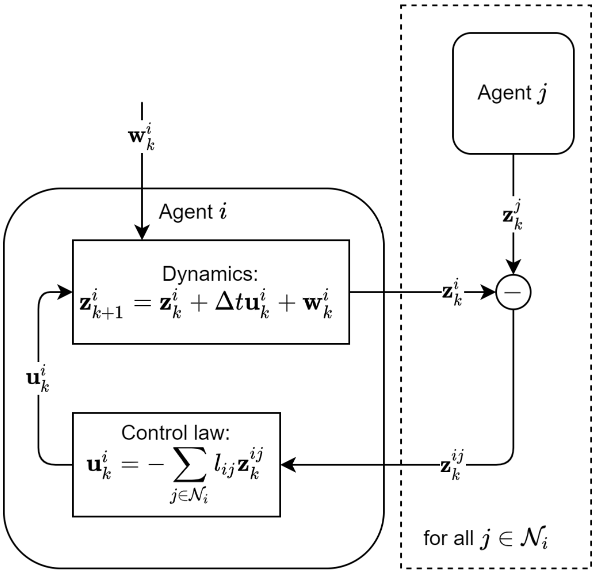

$\mathbf{w}_k^i$
Agent i
Dynamics:
$\mathbf{z}_{k+1}^i = \mathbf{z}_k^i + \Delta t \mathbf{u}_k^i + \mathbf{w}_k^i$
$\mathbf{u}_k^i$
Control law:
$\mathbf{u}_k^i = -\sum_{j \in \mathcal{N}_i} l_{ij} \mathbf{z}_k^{ij}$
Agent j
$\mathbf{z}_k^j$
$\mathbf{z}_k^i$
$-$
$\mathbf{z}_k^{ij}$
for all $j \in \mathcal{N}_i$

Figure 4.3: Disturbance uncertainty model.

assumed to have zero first-order moment and its second order moment is characterized via the covariance matrix $\mathbf{Q} \in \mathbb{S}_+^{DN}$. If we assume the process noise to be Gaussian, we can write:

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}_{DN}, \mathbf{Q}) \tag{4.10}$$

In multi-agent systems consisting of homogeneous agents, we might assume that the process noise $\mathbf{w}_i$ enters the dynamics of the agents in the same way for all agents. Then the process noise for a single agent is identically distributed with zero mean and covariance $\mathbf{Q}_D \in \mathbb{S}_+^D$. The covariance matrix $\mathbf{Q}$ then has some structure via the Kronecker product:

$$\mathbf{Q} = \mathbf{Q}_A \otimes \mathbf{Q}_D \tag{4.11}$$

where $\mathbf{Q}_A \in \mathbb{S}_+^N$ represents correlation of environmental disturbances between agents and $\mathbf{Q}_D$ represents correlation of these disturbances between dimensions.

As discussed in Section 4.1, environmental disturbances are often best modelled via a spatial (or spatiotemporal) covariance function. An often-used example of such a covariance function is the squared exponential function (see e.g. [51]):

$$[\mathbf{Q}_A]_{ij} = f(\|\mathbf{z}_i - \mathbf{z}_j\|) \tag{4.12}$$

$$= \sigma^2 \exp -\frac{\|\mathbf{z}_i - \mathbf{z}_j\|}{\theta^2} \tag{4.13}$$

where $[\mathbf{Q}_A]_{ij}$ is the disturbance covariance between agent i at position $\mathbf{z}_i$ and agent j at position $\mathbf{z}_j$. σ and θ are scaling parameters. With mobile agents whose position is changing over time, we thus see that the covariance structure also changes over time. Hence, we can add the subscript k to the covariance matrix ($\mathbf{Q}_k$) to indicate that it is time-varying. When the subscript is omitted, the covariance matrix is assumed to be constant in time. Note that with a time-varying covariance matrix, we can no longer assume that agents know the covariance matrix, and hence agents will need to estimate (part of) this covariance matrix, using knowledge of the covariance function instead.

4.2.3 Data Model for Simulations

Throughout this chapter and Chapters 5 and 6, formation control simulations are run to verify the performance of proposed estimators and filters. The general setup for these simulations is equivalent to the simulations in Section 3.6. The same hexagonal framework and control law are used, but for the simulations in this chapter and the following chapters, we introduce statistical noise. Hence, in this section the noise statistics used for the simulations are defined.

The measurement noise on the relative positions are modelled to be additive and Gaussian as described in Section 4.2.1. We assume all agents are homogeneous and therefore all measurements represented by edges of the sensing graph have the same covariance matrix. For the framework in $\mathbb{R}^2$, we use the following covariance matrix:

$$\mathbf{R}_{ij} = \sigma_v^2 \begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix} \quad i = 1, \dots, N \quad j \in \mathcal{N}_i$$

with varying values for the noise level σ_v. The measurement noise is considered to be independent with respect to time and between agents.

The process noise is considered to be structured via a Kronecker product. No correlation between dimensions is assumed: $\mathbf{Q}_D = \sigma_w^2 \mathbf{I}_D$. The process noise is correlated between agents, via the covariance matrix $\mathbf{Q}_A$. Correlation is assumed to be related to the distance between agents, i.e. if agents are close by, the disturbance affecting them is highly correlated, whereas this correlation decreases if the distance between agents becomes larger. Hence, the covariance matrix $\mathbf{Q}_A$ is defined by a standard squared exponential covariance function:

$$[\mathbf{Q}_A]_{ij} = \exp\big(-\frac{1}{2}\|\mathbf{p}_i - \mathbf{p}_j\|\big) \tag{4.14}$$

where $\mathbf{p}_i$ is the normalized configuration vector of agent i known in advance by the agents. Since (4.14) satisfies Mercer's condition [52] the matrix $\mathbf{Q}_A$ is guaranteed to be positive semi-definite and thus a valid covariance matrix. Note that a constant, known disturbance covariance matrix is chosen for the simulations. Ideally one would want to use real positions $\mathbf{z}_i$ to model the disturbance rather than the configuration positions $\mathbf{p}_i$, but these positions are not constant and more importantly not known by the agents. See Chapter 7 for further discussion on this subject.

The full $ND \times ND$ process noise covariance is $\mathbf{Q} = \mathbf{Q}_A \otimes \mathbf{Q}_D$. Process noise is further assumed to be uncorrelated in time, i.e. realizations $\mathbf{w}_k$ are assumed to be i.i.d. for all k.

The pseudocode for the affine formation control simulation with noise is thus as shown in Algorithm 2. In contrast with Algorithm 1, notice that now noise is added both on the dynamics via $\mathbf{w}_k$ and on the edge observations via $\mathbf{v}_k^{ij}$. The noise covariances $\mathbf{Q}$ and $\mathbf{R}_{ij}$ are as defined above. Lastly, the data model from the perspective of a single node i is also visualized in Figure 4.4. The main goal for the remainder of the thesis is to design a local filter which optimally tracks local edge states, visualized by the *filter* block in the block diagram.

Algorithm 2 Relative formation control with modelled noise

```
Unknown global constants: $N, \Delta t, \boldsymbol{\mu}, \mathbf{P}, \mathbf{Q}, \mathbf{R}_{ij}$ for $i = 1, \dots, N$, $j \in \mathcal{N}_i$
Known global constants: $D$
Known local constants: $l_{ij}$ for $i = 1, \dots, N$, $j \in \mathcal{N}_i$
Draw initial positions $\mathbf{z}_1^i$ from $\boldsymbol{\mathcal{N}}(\boldsymbol{\mu}, \mathbf{P})$
$k = 1$
while $k \leq k_{max}$ do
    Draw process noise $\mathbf{w}_k$ from $\boldsymbol{\mathcal{N}}(\mathbf{0}_{DN}, \mathbf{Q})$
    for $i \in \mathcal{V}$ do
        for $j \in \mathcal{N}_i$ do
            Draw observation noise $\mathbf{v}_k^{ij}$ from $\boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{R}^{ij})$
            $\mathbf{y}_k^{ij} = \mathbf{z}_k^i - \mathbf{z}_k^j + \mathbf{v}_k^{ij}$          ▷ Measurements
        end for
        $\mathbf{u}_k^i = -\sum_{j \in \mathcal{N}_i} l_{ij} \mathbf{y}_k^{ij}$          ▷ Control law
        $\mathbf{z}_{k+1}^i = \mathbf{z}_k^i + \Delta t \mathbf{u}_k^i + \mathbf{w}_k^i$          ▷ Dynamics update
    end for
    $k = k + 1$
end while
```

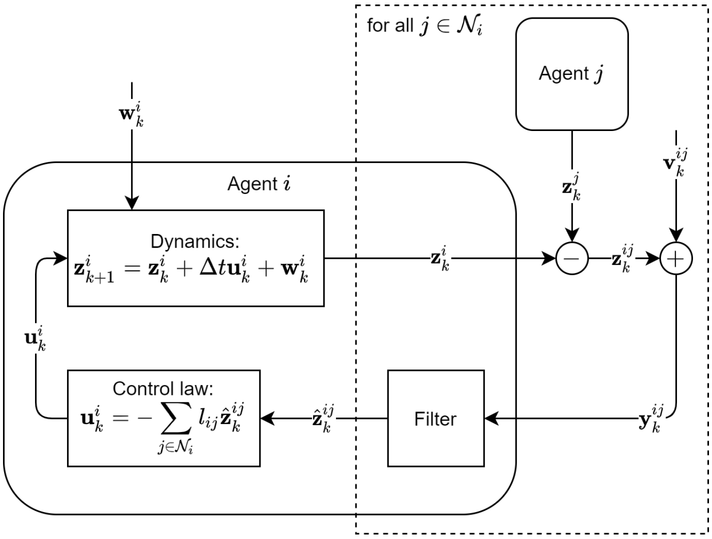

for all $j \in \mathcal{N}_i$
Agent j
$\mathbf{w}_k^i$
$\mathbf{v}_k^{ij}$
Agent i
$\mathbf{z}_k^j$
Dynamics:
$\mathbf{z}_{k+1}^i = \mathbf{z}_k^i + \Delta t \mathbf{u}_k^i + \mathbf{w}_k^i$
$\mathbf{z}_k^i$
$-$
$\mathbf{z}_k^{ij}$
$+$
$\mathbf{u}_k^i$
Control law:
$\mathbf{u}_k^i = -\sum_{j \in \mathcal{N}_i} l_{ij} \hat{\mathbf{z}}_k^{ij}$
$\hat{\mathbf{z}}_k^{ij}$
Filter
$\mathbf{y}_k^{ij}$

Figure 4.4: Data model for the local closed-loop system for an arbitrary agent i.

4.3 Maximum likelihood estimator

Given that uncertainty is introduced on the edges (via noisy measurements) and the nodes (via random disturbances) of a multi-agent system, we aim to optimally estimate the edge states such that the influence of the uncertainty in the closed-loop system is minimized. In this section, we will introduce a maximum likelihood estimator (MLE) for each edge. This estimator does not take into account evolution of the system over time, hence it can be called instantaneous. This estimator will mainly be used as a benchmark for future estimators and filters.

The parameter to be estimated is the relative position $\mathbf{z}_{ij} = \mathbf{z}_i - \mathbf{z}_j \in \mathbb{R}^D$ of an agent i relative to a neighbor j. We wish to estimate this parameter using only the T measurements $\mathbf{y}_{ij}$ as in (4.8). The maximum likelihood estimator is defined as

$$\hat{\mathbf{z}}_k^{ij} = \arg\min_{\mathbf{z}_k^{ij}} \mathrm{p}(\mathbf{z}_k^{ij}; \mathbf{y}_k^{ij}) \tag{4.15}$$

Using the noise model (4.8), which is linear with Gaussian noise, the maximum likelihood estimator takes a standard form:

$$\begin{aligned} \hat{\mathbf{z}}_{ij} &= (\mathbf{H}^\top \tilde{\mathbf{R}}_{ij}^{-1} \mathbf{H})^{-1} \mathbf{H}^\top \tilde{\mathbf{R}}_{ij}^{-1} \mathbf{y}_{ij} \\ &= \frac{1}{T} \mathbf{H}^\top \mathbf{y}_k^{ij} \end{aligned} \tag{4.16}$$

where $\mathbf{H} = \mathbf{1}_T \otimes \mathbf{I}_D$ is the observation matrix as introduced in Section 4.2 and $\tilde{\mathbf{R}}_{ij} = \mathbf{I}_T \otimes \mathbf{R}_{ij}$ is the known measurement covariance matrix. For the linear Gaussian model, the MLE is efficient, i.e. it equals the Minimum Variance Unbiased Estimator (MVUE).

The posterior covariance of the MLE estimate is given by the inverse of the Fisher information matrix:

$$\begin{aligned} \mathbf{\Sigma}_{ij}^{\text{post,MLE}} &= (\mathbf{H}^\top \tilde{\mathbf{R}}_{ij}^{-1} \mathbf{H})^{-1} \\ &= \frac{1}{T} \mathbf{R}_{ij} \end{aligned} \tag{4.17}$$

The posterior covariance indicates the variance of the obtained estimate. Algorithm 2 can now be adapted by applying the MLE at every iteration. The affine formation control simulation with MLE is described by Algorithm 3. Note that in this algorithm, T independent measurements are aggregated for each edge. The MLE (4.16) is then applied to optimally estimate each edge state. Then, the control law uses the estimated edge state as inputs rather than the direct measurements as in Algorithm 2. Note that for $T = 1$, the MLE reduces to the estimator-less case, and Algorithms 2 and 3 are equivalent.

The simulation specified by Algorithm 3 is run for varying measurement rates T. Note that with a time interval of $\Delta t = 0.001$ seconds, agents measure at a rate of T kHz. Convergence results for the hexagonal formation for various T are shown in Figure 4.5, where only observation noise is applied ($\sigma_v = 0.1$, $\sigma_w = 0$). The MLE results are compared against the simulation without estimator, as described in Algorithm 2. As expected, the simulations with uncertainty converge to a rigid formation up to some

Algorithm 3 Relative formation control using MLE

```
Unknown global constants: $N, \Delta t, \boldsymbol{\mu}, \mathbf{P}, \mathbf{Q}$
Known global constants: $D, T, \mathbf{H}$
Known local constants: $l_{ij}, \mathbf{R}_{ij}$ for $i = 1, \dots, N$, $j \in \mathcal{N}_i$
$\tilde{\mathbf{R}}_{ij} = \mathbf{I}_T \otimes \mathbf{R}_{ij}$ for $i = 1, \dots, N$, $j \in \mathcal{N}_i$
Draw initial positions $\mathbf{z}_1^i$ from $\boldsymbol{\mathcal{N}}(\boldsymbol{\mu}, \mathbf{P})$
$k = 1$
while $k \leq k_{max}$ do
    Draw process noise $\mathbf{w}_k$ from $\boldsymbol{\mathcal{N}}(\mathbf{0}_{DN}, \mathbf{Q})$
    for $i \in \mathcal{V}$ do
        for $j \in \mathcal{N}_i$ do
            Draw observation noise $\mathbf{v}_k^{ij}$ from $\boldsymbol{\mathcal{N}}(\mathbf{0}_{DT}, \tilde{\mathbf{R}}_{ij})$
            $\mathbf{y}_k^{ij} = \mathbf{H}(\mathbf{z}_k^i - \mathbf{z}_k^j) + \mathbf{v}_k^{ij}$        ▷ Measurements
            $\hat{\mathbf{z}}_k^{ij} = \frac{1}{T}\mathbf{H}^\top \mathbf{y}_k^{ij}$        ▷ MLE
        end for
        $\mathbf{u}_k^i = -\sum_{j \in \mathcal{N}_i} l_{ij} \hat{\mathbf{z}}_k^{ij}$        ▷ Control law
        $\mathbf{z}_{k+1}^i = \mathbf{z}_k^i + \Delta t \mathbf{u}_k^i + \mathbf{w}_k^i$        ▷ Dynamics update
    end for
    $k = k + 1$
end while
```

noise floor. When the MLE is applied, this noise floor is lower than when the noisy measurements are directly used in the control law. This is expected since the MLE has access to T measurements compared to the single measurement for the estimator-less case. The noise floor for the MLE is a factor of roughly $\sqrt{T}$ smaller than without estimator. We also note that the convergence variance similarly scales down by the same factor (note that the error axis in Figure 4.5 is logarithmic), visible through the plotted standard deviation region. Convergence speed seems to be unaffected by the introduced uncertainty, as both noisy simulations converge just as quickly as the noiseless simulation, up until the moment their noise floor is met. From these results, we know that improvement in Procrustes error floor is possible. However, there is still room for improvement. The MLE is instantaneous, i.e., it does not take into account past information which might be exploited. In the next sections we will expand on the MLE with the intent of improving convergence further.

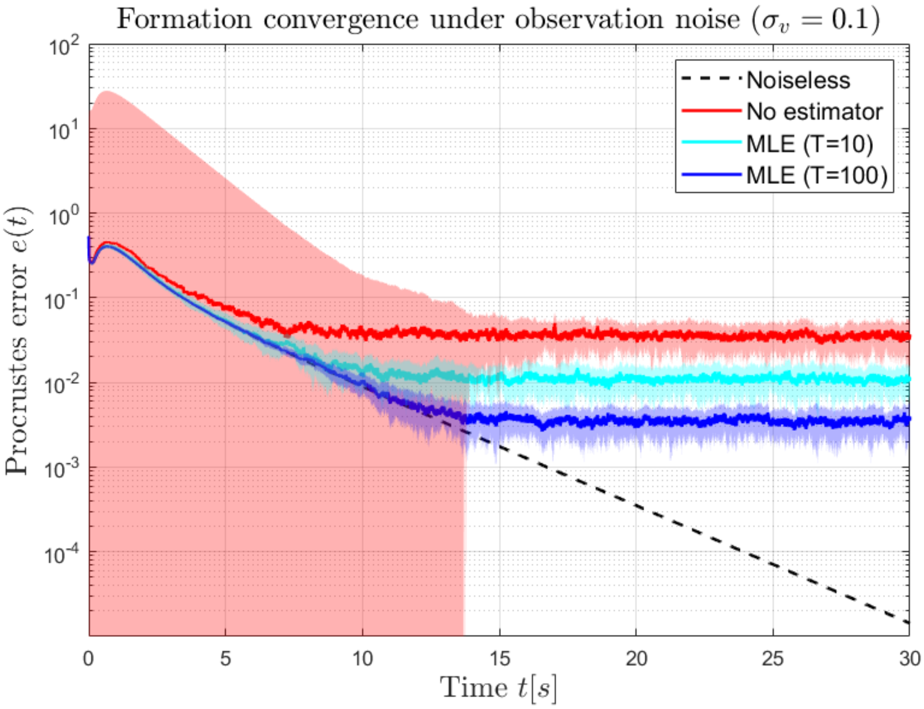


Figure 4.5: Convergence of hexagonal formation. The plot shows the mean convergence for 50 runs of Algorithms 2 and 3, and the ±1 standard deviation regions.

4.4 Minimum Mean Square Error Estimator

Extending on the MLE estimator proposed in Section 4.3, we will now attempt to include temporal information in the edge state estimates. As such, we enter the Bayesian realm of estimation. Since we know that the relative position of agent i with respect to agent j will not drastically change over a short period of time, we will use the relative position estimate at time index $k-1$ as a prior for the estimate of the relative position at time index k. When the Minimum Mean Square Error (MMSE) estimator is used for consecutive time indices, we can call this an MMSE filter.

Consider the prior for $\mathbf{z}_k^{ij}$ to be Gaussian with mean $\hat{\mathbf{z}}_{k-1}^{ij}$ and some fixed covariance $\mathbf{\Sigma}_{ij}$:

$$\mathbf{z}_k^{ij} \sim \mathcal{N}(\hat{\mathbf{z}}_{k-1}^{ij}, \mathbf{\Sigma}_{ij}) \tag{4.18}$$

The observation model is the linear Gaussian model (4.8), repeated here:

$$\mathbf{y}_k^{ij} = \mathbf{H}\mathbf{z}_k^{ij} + \mathbf{v}_k^{ij} \tag{4.19}$$

Minimizing the covariance in a minimum mean square error sense gives the following estimator (see e.g., [53]):

$$\hat{\mathbf{z}}_k^{ij} = \hat{\mathbf{z}}_{k-1}^{ij} + \mathbf{\Sigma}_{ij}\mathbf{H}^\top\left(\mathbf{H}\mathbf{\Sigma}_{ij}\mathbf{H}^\top + \tilde{\mathbf{R}}_{ij}\right)^{-1}(\mathbf{y}_k^{ij} - \mathbf{H}\hat{\mathbf{z}}_{k-1}^{ij}) \tag{4.20}$$

We can get an expression for the posterior covariance of the MMSE relative position estimate:

$$\mathbf{\Sigma}_{ij}^{\text{post,MMSE}} = (\mathbf{\Sigma}_{ij}^{-1} - \mathbf{H}^\top \tilde{\mathbf{R}}_{ij}^{-1} \mathbf{H})^{-1} \tag{4.21}$$

Note that this posterior covariance is always smaller than the posterior covariance of the MLE (4.17). Generally, when the covariance of the prior is small, then the posterior covariance will be small as well. In the limit, when the covariance of the prior becomes very large, the MMSE estimator reduces to the MLE estimator.

Algorithm 4 describes the simulation with MMSE filter. Notice that the MMSE filter uses the previous estimate as the prior mean in the next iteration. The choice of the prior covariance $\mathbf{\Sigma}_{ij}$ is very significant here. When chosen too large, the MMSE filter does not show any improvement over the MLE (i.e., the prior information given is too weak). With decreasing $\mathbf{\Sigma}_{ij}$ more weight is placed on the prior (i.e., the previous estimate). This will likely result in better estimates when the previous estimate is good and the system is near steady state (i.e., the agents are near formation and control inputs are small). However, convergence speed may become slower since the MMSE filter will give past measurements larger weights, which may already be outdated.

Algorithm 4 Relative formation control using MMSE filter

```
Unknown global constants: $N, \Delta t, \boldsymbol{\mu}, \mathbf{P}$
Known global constants: $D, T, \mathbf{H}$
Known local constants: $l_{ij}, \mathbf{\Sigma}_{ij}, \mathbf{R}_{ij}$ for $i = 1, \dots, N$, $j \in \mathcal{N}_i$
$\tilde{\mathbf{R}}_{ij} = \mathbf{I}_T \otimes \mathbf{R}_{ij}$ for $i = 1, \dots, N$, $j \in \mathcal{N}_i$
Draw initial positions $\mathbf{z}_1^i$ from $\boldsymbol{\mathcal{N}}(\boldsymbol{\mu}, \mathbf{P})$
$k = 1$
while $k \leq k_{max}$ do
    Draw process noise $\mathbf{w}_k$ from $\boldsymbol{\mathcal{N}}(\mathbf{0}_{DN}, \mathbf{Q})$
    for $i \in \mathcal{V}$ do
        for $j \in \mathcal{N}_i$ do
            Draw observation noise $\mathbf{v}_k^{ij}$ from $\boldsymbol{\mathcal{N}}(\mathbf{0}_{DT}, \tilde{\mathbf{R}}_{ij})$
            $\mathbf{y}_k^{ij} = \mathbf{H}(\mathbf{z}_k^i - \mathbf{z}_k^j) + \mathbf{v}_k^{ij}$    ▷ Measurements
            $\hat{\mathbf{z}}_k^{ij} = \hat{\mathbf{z}}_{k-1}^{ij} + \mathbf{\Sigma}_{ij}\mathbf{H}^\top \big(\mathbf{H}\mathbf{\Sigma}_{ij}\mathbf{H}^\top + \tilde{\mathbf{R}}_{ij}\big)^{-1}(\mathbf{y}_k^{ij} - \mathbf{H}\hat{\mathbf{z}}_{k-1}^{ij})$    ▷ MMSE
        end for
        $\mathbf{u}_k^i = -\sum_{j \in \mathcal{N}_i} l_{ij}\hat{\mathbf{z}}_k^{ij}$    ▷ Control law
        $\mathbf{z}_{k+1}^i = \mathbf{z}_k^i + \Delta t \mathbf{u}_k^i + \mathbf{w}_k^i$    ▷ Dynamics update
    end for
    $k = k + 1$
end while
```

This effect can indeed be observed via the formation convergence shown in Figure

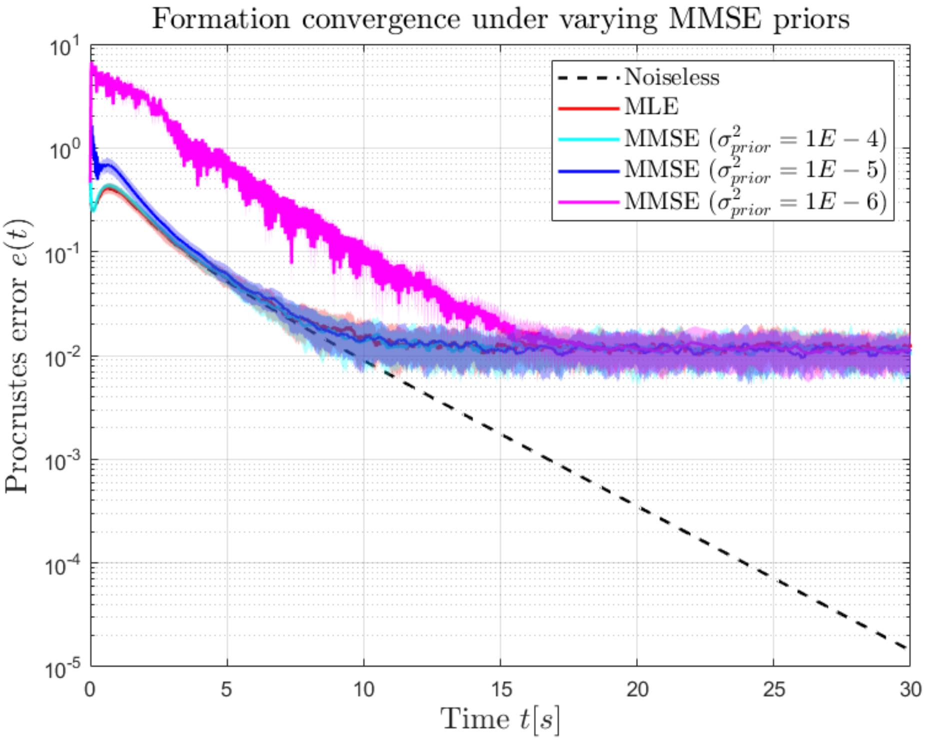


Figure 4.6: Convergence of hexagonal formation. The plot shows the mean convergence for 50 runs of Algorithms 3 and 4 (both using $T = 10$) for varying σ^2_{prior}, and the ±1 standard deviation regions.

4.6. Three priors were used of the form $\mathbf{\Sigma}_{ij} = \sigma^2_{prior}\mathbf{I}_D$. For a tighter prior (i.e., smaller σ^2_{prior}) the initial convergence of the MMSE filter is clearly slower and initially even diverges. Near steady state, the supposed positive effect of the prior is not immediately visible. The steady state error is equivalent for all priors, and additionally equal to that of the MLE. Upon zooming in, it can be seen that a tighter prior does make the steady state estimates less erratic over time. This can be desirable since it will make sure that the control input (which is a linear function of the filtered edge states) is less erratic as well, which is desirable in practical applications.

Ideally, we would want two improvements over the MMSE estimator. First would be a varying prior based on the uncertainty at that moment in the simulation, instead of a fixed prior. Second is a prediction element which includes the control action of the agent in the prior for the next estimate. The edge-based distributed Kalman filter introduced in the next section incorporates both of these elements.

4.5 Edge-based Kalman Filter

So far, edge state smoothing via the MLE (Section 4.3) and the MMSE estimator (Section 4.4) have been introduced. Already, the MMSE estimator is similar to a Kalman filter for the state estimation of a single edge. In this section, we will extend the MMSE filter to arrive at a proposed filter which is called the edge Kalman filter (Edge-KF).

From the MMSE estimator introduced in Section 4.4, we can identify two shortcomings. Both shortcomings originate in the prior used for the estimate. First of all, the prior used for the estimation of the edge state has a mean which is the previous estimate. However, since the agents are mobile, they move and the previous estimate might be outdated. Since agents know the control signal they apply, a prediction can be made for the next time instance based on the previous estimated state and the predicted movement of the agent. As such, we exploit the fact that the state space model of agents is known. However, note that while an agent knows its own movement, the edge state is a function of two agents. The control inputs therefore will need to be communicated locally to allow unbiased prediction of the edge state. Secondly, the prior used in the MMSE estimator has fixed covariance matrix. This is usually not desirable. For example, initially an agent may know very little about its edge states due to there being few aggregated measurements. However, over time the state estimates will improve and once near steady state, edge states will be relatively stable. As such, more weight can then be placed on the prior information by decreasing the prior covariance $\mathbf{\Sigma}_{ij}$. The Kalman filter introduces solutions to both of these problems.

We will now see what the agent model and prediction and update equations will look like for the edge-based distributed Kalman filter (Edge-KF). To do this, we will look at the system from the perspective of an agent i observing one of its neighbors j.

For agent i, the discrete-time dynamics are as follows:

$$\mathbf{z}^i_{k+1} = \mathbf{z}^i_k + \Delta t \mathbf{u}^i_k + \mathbf{w}^i_k \quad i = 1, \dots, N \tag{4.22}$$

The index $k \in \mathbb{N}$ is the time index, and $\Delta t \in \mathbb{R}$ is a scalar denoting the time step period of the filter.

The observations of agents with respect to their neighbors follow the data model in Section 4.2:

$$\mathbf{y}^{ij}_k = \mathbf{H}\mathbf{z}^{ij}_k + \mathbf{v}^{ij}_k \quad i = 1, \dots, N \quad j \in \mathcal{N}_i \tag{4.23}$$

with $\mathbf{v}^{ij}_k$ zero-mean and with covariance $\tilde{\mathbf{R}}_{ij}$.

We can rewrite (4.22) by combining the dynamics of agents i and j to describe the dynamics of the edge (i, j):

$$\mathbf{z}^{ij}_{k+1} = \mathbf{z}^{ij}_k + \Delta t(\mathbf{u}^i_k - \mathbf{u}^j_k) + \mathbf{w}^{ij}_k \quad i = 1, \dots, N \quad j \in \mathcal{N}_i \tag{4.24}$$

The process noise for the edge $\mathbf{w}^{ij}_k$ can be expressed as a sum of process noise on the node, to comply with the data model set up in Section 4.2:

$$\mathbf{w}^{ij}_k = \mathbf{w}^i_k - \mathbf{w}^j_k \quad i = 1, \dots, N \quad j \in \mathcal{N}_i \tag{4.25}$$

In terms of the global process noise vector $\mathbf{w}_k$, we can write the edge process noise $\mathbf{w}^{ij}_k$ as follows:

$$\mathbf{w}^{ij}_k = \mathbf{B}_{ij}\mathbf{w}_k \quad i = 1, \dots, N \quad j \in \mathcal{N}_i \tag{4.26}$$

where $\mathbf{B}_{ij} \in \mathbb{R}^{D \times DN}$ is a selection matrix that can be written as $\mathbf{b}_{ij} \otimes \mathbf{I}_D$. The ith entry of $\mathbf{b}_{ij}$ is 1, the jth entry is -1 and all other entries are zero. As such, it can

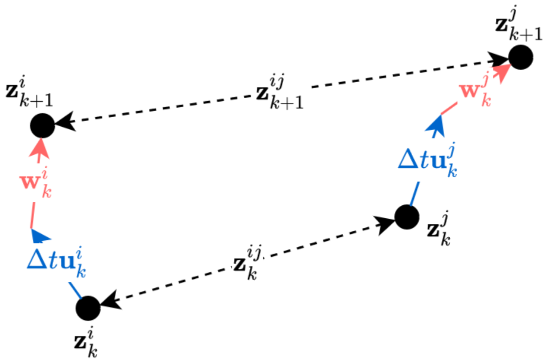


Figure 4.7: Dynamics of the edge state $\mathbf{z}^{ij}_k$ in $\mathbb{R}^2$.

be considered a transposed column of the incidence matrix of the bidirectional sensing graph. If $\mathbf{w}_k$ is Gaussian, then $\mathbf{w}^{ij}_k$ will be too, since (4.26) is linear. The covariance of $\mathbf{w}^{ij}_k$ can be written $\mathbf{Q}_{ij} = \mathbf{B}_{ij}\mathbf{Q}\mathbf{B}^\top_{ij}$.

The edge dynamics are visualized in Figure 4.7 for a pair of mobile agents in $\mathbb{R}^2$.

We notice that the edge state dynamics (4.24) and observations (4.23) are linear and Gaussian, having used the data model described in Section 4.2.

The proposed edge-based Kalman filter updates can thus be introduced as a series of parallel Kalman filters for each agent, one for each of its neighbors. The exact Kalman filter prediction and update equations will now be examined in more detail.

The edge-based Kalman filter updates consist of two steps. The first is the prediction step, where the relative positions for the next time step are predicted based on the control input and current estimated state. Second is the update step, where the new measurement is examined and the relative position estimate is updated with this new information, using the predicted state as a prior.

Edge-KF initialization

The initialization is similar to that of the centralized Kalman filter, where we exploit the fact that the initial positions of agents are independent. The expected initial edge state is zero:

$$\hat{\mathbf{z}}^{ij}_{0|0} = \mathbb{E}(\mathbf{z}^{ij}_0) = \mathbb{E}(\mathbf{z}^i_0) - \mathbb{E}(\mathbf{z}^j_0) = \boldsymbol{\mu} - \boldsymbol{\mu} = \mathbf{0}_D \tag{4.27}$$

The initial covariance then becomes

$$\begin{aligned}\boldsymbol{\Sigma}^{ij}_{0|0} &= \mathbb{E}(\mathbf{z}^{ij}_0\mathbf{z}^{ij\top}_0) - \mathbb{E}(\mathbf{z}^{ij}_0)\mathbb{E}(\mathbf{z}^{ij}_0)^\top \\ &= \mathbb{E}((\mathbf{z}^i_0 - \mathbf{z}^j_0)(\mathbf{z}^i_0 - \mathbf{z}^j_0)^\top) \\ &= \mathbb{E}(\mathbf{z}^i_0\mathbf{z}^{i\top}_0) + \mathbb{E}(\mathbf{z}^j_0\mathbf{z}^{j\top}_0) - 2\mathbb{E}(\mathbf{z}^i_0\mathbf{z}^{j\top}_0) \\ &= 2\mathbf{P}\end{aligned} \tag{4.28}$$

which indicates that the initial uncertainty on the edge states is twice as high as the uncertainty on the node states.

Edge-KF prediction step

The prediction step is computed by an agent before acquiring a new set of measurements, and is given by the following equations:

$$\hat{\mathbf{z}}^{ij}_{k+1|k} = \hat{\mathbf{z}}^{ij}_{k|k} + \Delta t(\mathbf{u}^i_k - \mathbf{u}^j_k) \qquad i = 1, \ldots, N \quad j \in \mathcal{N}_i \tag{4.29}$$

$$\mathbf{\Sigma}^{ij}_{k+1|k} = \mathbf{\Sigma}^{ij}_{k|k} + \mathbf{Q}_{ij} \qquad i = 1, \ldots, N \quad j \in \mathcal{N}_i \tag{4.30}$$

The prediction for the edge state follows the edge dynamics equation (4.24). Note that agent i which is tracking the edge state requires the control input of agent j. As such, to make an unbiased prediction the control inputs need to be communicated between neighbors. In Chapter 6, we will see what happens if the neighbor control input is not available and the prediction is biased. The predicted covariance $\mathbf{\Sigma}^{ij}_{k+1|k}$ increases in the prediction step to indicate that, since process noise is added to the system, uncertainty about the state of the system increases. A large process noise covariance $\mathbf{Q}_{ij}$ means a large increase in uncertainty, and vice versa.

Edge-KF update step

The update step weighs the newly acquired measurements $\mathbf{y}^{ij}_k$ against the predicted state $\hat{\mathbf{z}}^{ij}_{k|k-1}$, and updates the estimation as follows:

$$\hat{\mathbf{z}}^{ij}_{k|k} = \hat{\mathbf{z}}^{ij}_{k|k-1} + \mathbf{K}^{ij}_k\big(\mathbf{y}^{ij}_k - \mathbf{H}\hat{\mathbf{z}}^{ij}_{k|k-1}\big) \qquad i = 1, \ldots, N \quad j \in \mathcal{N}_i \tag{4.31}$$

$$\mathbf{\Sigma}^{ij}_{k|k} = \big(\mathbf{I}_D - \mathbf{K}^{ij}_k\mathbf{H}\big)\mathbf{\Sigma}^{ij}_{k|k-1} \qquad i = 1, \ldots, N \quad j \in \mathcal{N}_i \tag{4.32}$$

$$\mathbf{K}^{ij}_k = \mathbf{\Sigma}^{ij}_{k|k-1}\mathbf{H}^\top\big(\mathbf{H}\mathbf{\Sigma}^{ij}_{k|k-1}\mathbf{H}^\top + \tilde{\mathbf{R}}_{ij}\big)^{-1} \qquad i = 1, \ldots, N \quad j \in \mathcal{N}_i \tag{4.33}$$

This update step works identically to the update step for state tracking in the distributed Kalman filter in [36]. The Kalman gain $\mathbf{K}^{ij}_k$ is dependent on the covariance of the measurement noise $\tilde{\mathbf{R}}_{ij}$. If the uncertainty on the measurements is very small, then the Kalman gain is close to identity meaning the estimated state closely follows the measurement. If the measurement noise is large, the estimated state will place more weight on the prediction based on the agent's known control actions and measurements in the past.

The edge-based Kalman filter (Edge-KF) is tested via simulations described by Algorithm 5. Two simulations were performed. For the first, only measurement noise was applied ($\sigma_v = 0.1, \sigma_w = 0$). For the second, additionally process noise was applied ($\sigma_v = 0.1, \sigma_w = 0.001$). The results are shown in Figures 4.8 and 4.9. The initialization equations (4.27-4.28), prediction equations (4.29-4.30) and update equations (4.31-4.33) all feature in Algorithm 5. The control law then takes the updated edge estimates as inputs.

Figure 4.8 shows the average evolution of the Procrustes error over time. The steady state error for the Edge-KF is approximately 10 times lower than the MLE and MMSE errors after convergence which is significantly lower. Under process noise however, the noise floor difference is reduced. Figures 4.8 and 4.9 show that when observation noise is dominant, the Edge-KF is a clear improvement in steady-state Procrustes error over

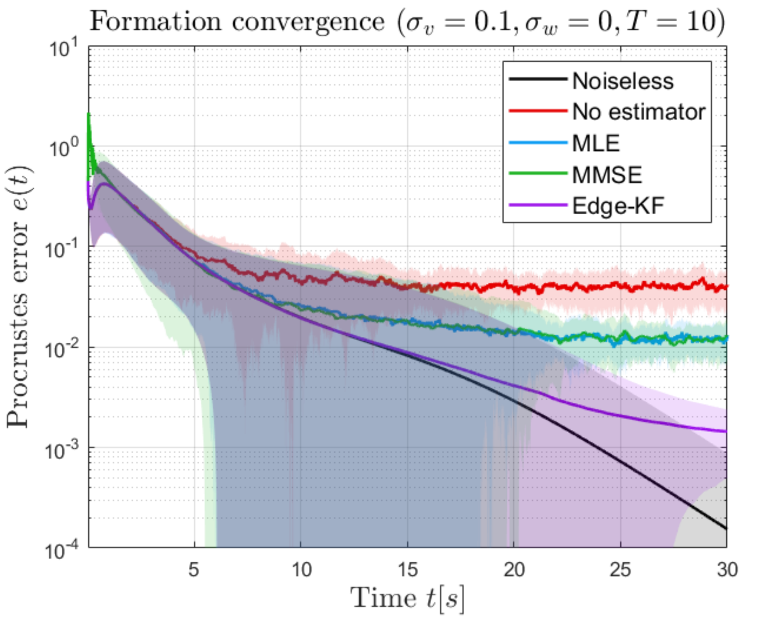


Figure 4.8: Convergence of a hexagonal formation for MLE, MMSE and Edge-KF with observation noise. The plot shows the mean convergence for 50 runs of Algorithms 2, 3, 4 and 5, and the ±1 standard deviation regions.

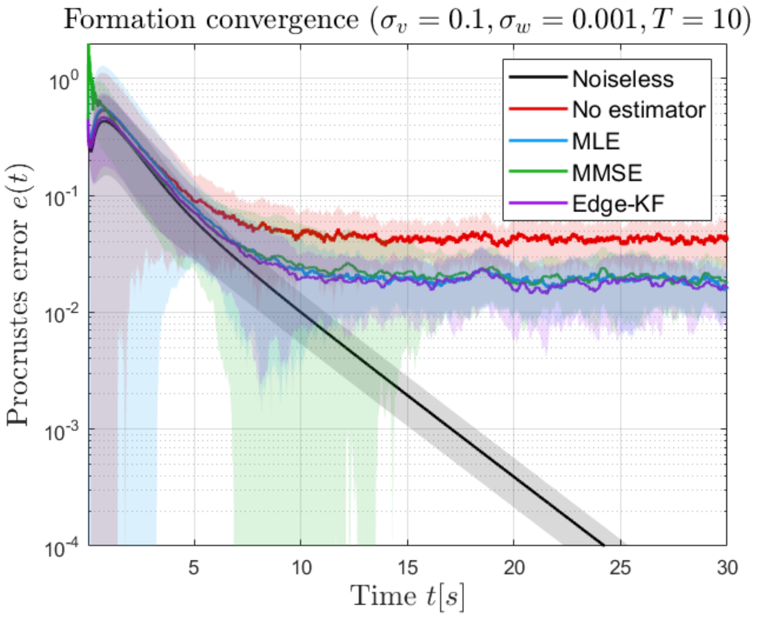


Figure 4.9: Convergence of a hexagonal formation for MLE, MMSE and Edge-KF with observation and process noise. The plot shows the mean convergence for 50 runs of Algorithms 2, 3, 4, and 5, and the ±1 standard deviation regions.

Algorithm 5 Relative formation control using Edge-KF

```
Unknown global constants: $N, \boldsymbol{\mu}$
Known global constants: $D, T, \Delta t, \mathbf{H}, \mathbf{P}$
Known local constants: $l_{ij}, \mathbf{R}_{ij}, \mathbf{Q}_{ij}$ for $i = 1, \dots, N,\ j \in \mathcal{N}_i$
$\tilde{\mathbf{R}}_{ij} = \mathbf{I}_T \otimes \mathbf{R}_{ij}$ for $i = 1, \dots, N,\ j \in \mathcal{N}_i$
Draw initial positions $\mathbf{z}_1^i$ from $\boldsymbol{\mathcal{N}}(\boldsymbol{\mu}, \mathbf{P})$
$\hat{\mathbf{z}}_{0|0}^{ij} = \mathbf{0}_D$ for all $i \in \mathcal{V}, j \in \mathcal{N}_i$
$\boldsymbol{\Sigma}_{0|0}^{ij} = 2\mathbf{P}$ for all $i \in \mathcal{V}, j \in \mathcal{N}_i$
$\mathbf{u}_0^i = \mathbf{0}_D$ for all $i \in \mathcal{V}$
$k = 1$
while $k \leq k_{max}$ do
    Draw process noise $\mathbf{w}_k$ from $\boldsymbol{\mathcal{N}}(\mathbf{0}_{DN}, \mathbf{Q})$
    for $i \in \mathcal{V}$ do
        for $j \in \mathcal{N}_i$ do
            $\hat{\mathbf{z}}_{k|k-1}^{ij} = \hat{\mathbf{z}}_{k-1|k-1}^{ij} + \Delta t \mathbf{u}_{k-1}^i - \Delta t \mathbf{u}_{k-1}^j$
            $\boldsymbol{\Sigma}_{k|k-1}^{ij} = \boldsymbol{\Sigma}_{k-1|k-1}^{ij} + \mathbf{Q}_{ij}$
            Draw observation noise $\mathbf{v}_k^{ij}$ from $\boldsymbol{\mathcal{N}}(\mathbf{0}_{DT}, \tilde{\mathbf{R}}_{ij})$
            $\mathbf{y}_k^{ij} = \mathbf{H}(\mathbf{z}_k^i - \mathbf{z}_k^j) + \mathbf{v}_k^{ij}$                ▷ Measurements
            $\mathbf{K}_k^{ij} = \boldsymbol{\Sigma}_{k|k-1}^{ij} \mathbf{H}^\top (\mathbf{H} \boldsymbol{\Sigma}_{k|k-1}^{ij} \mathbf{H}^\top + \tilde{\mathbf{R}}_{ij})^{-1}$
            $\hat{\mathbf{z}}_{k|k}^{ij} = \hat{\mathbf{z}}_{k|k-1}^{ij} + \mathbf{K}_k^{ij} (\mathbf{y}_k^{ij} - \mathbf{H} \hat{\mathbf{z}}_{k|k-1}^{ij})$
            $\boldsymbol{\Sigma}_{k|k}^{ij} = (\mathbf{I}_D - \mathbf{K}_k^{ij} \mathbf{H}) \boldsymbol{\Sigma}_{k|k-1}^{ij}$
        end for
        $\mathbf{u}_k^i = -\sum_{j \in \mathcal{N}_i} l_{ij} \hat{\mathbf{z}}_{k|k}^{ij}$                ▷ Control law
        $\mathbf{z}_{k+1}^i = \mathbf{z}_k^i + \Delta t \mathbf{u}_k^i + \mathbf{w}_k^i$                ▷ Dynamics update
    end for
    $k = k + 1$
end while
```

the MLE and MMSE estimators. When process noise is dominant, the Edge-KF shows less significant improvement in steady-state Procrustes error compared to the MLE and MMSE estimators. However, the improvement in estimation error is significant. The norm of the difference between the true edge states and the estimated edge states are plotted in Figure 4.10, defined as follows:

$$\epsilon_k = \sqrt{\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} \|\mathbf{z}_k^{ij} - \hat{\mathbf{z}}_k^{ij}\|^2} \tag{4.34}$$

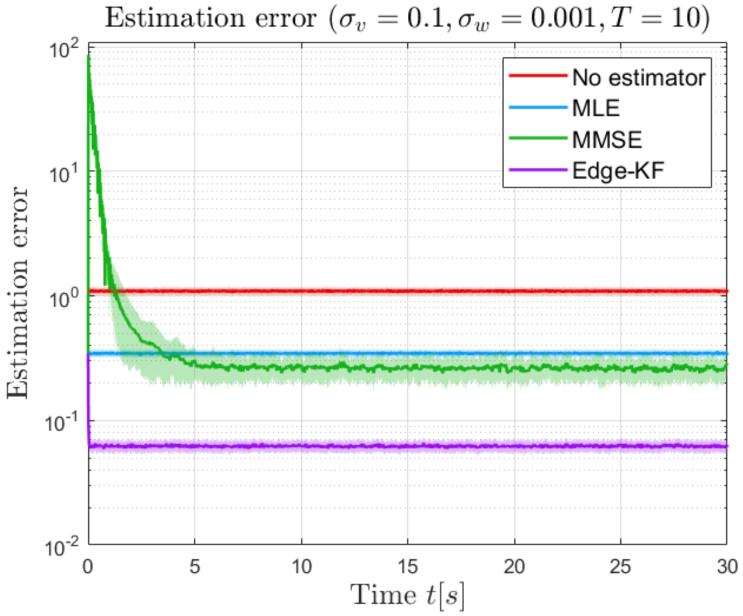
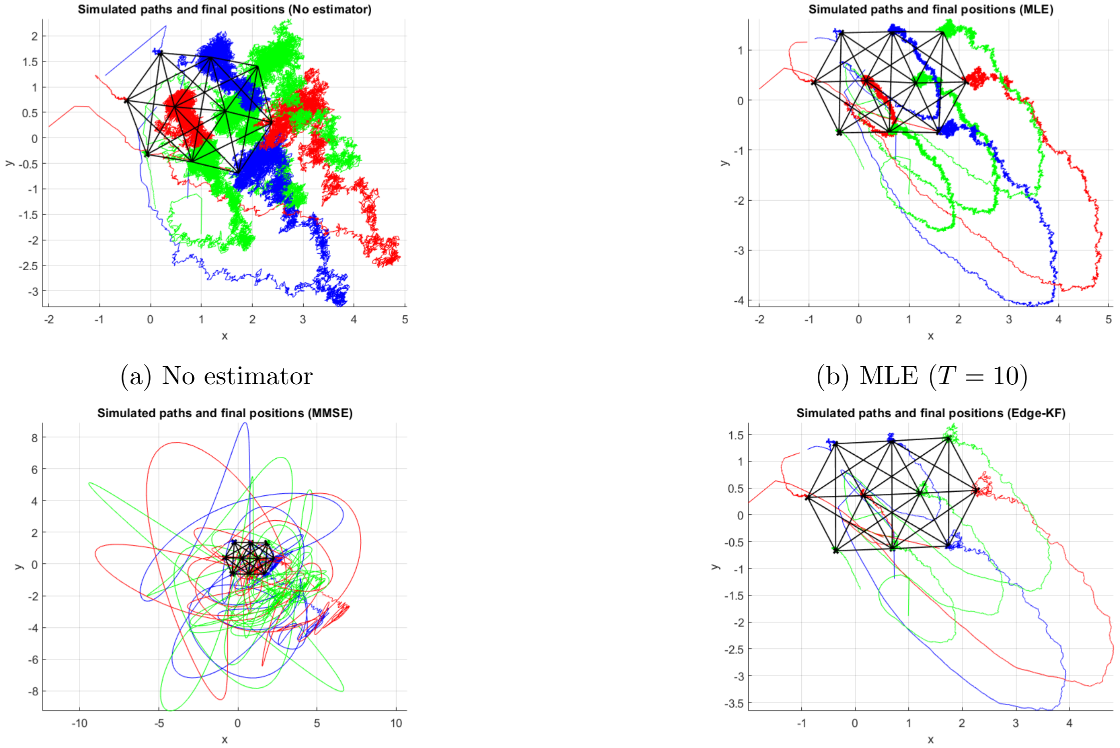


Figure 4.10: Estimation errors for MLE, MMSE and Edge-KF with observation and process noise. The plot shows the mean estimation error for 50 runs of Algorithms 2, 3, 4, and 5, and the ±1 standard deviation regions.

After initially suffering from the rapid movement of agents during the first phase of the simulation, the MMSE estimator shows improved steady-state estimation error compared to the MLE. However, the Kalman filter shows even better performance. Therefore, the conclusion can be drawn that although the superiority of Edge Kalman filter is not directly visible from the Procrustes error when faced with dominant process noise. However, from 4.10 the improved estimation accuracy is evident.

The paths of agents under measurement and process noise in $\mathbb{R}^2$ for a single run of the simulations are also shown in Figure 4.11.



(a) No estimator

(b) MLE ($T = 10$)

(c) MMSE ($T = 10$)

(d) Edge-KF ($T = 10$)

Figure 4.11: Paths of agents in $\mathbb{R}^2$ over time for the hexagonal framework ($\sigma_v = 0.1, \sigma_w = 0.001$).

5 Global Approach to Edge State Filtering

In Chapter 4 a data model for the formation control problem was introduced, and estimators and filters were proposed by approaching the filtering problem locally. This makes sense, since the formation control problem is a multi-agent problem, which we desire to solve in a decentralized manner. However, from the data model we have seen that the system is heavily interconnected, for example via the disturbance model. As such, in this chapter we will take a global perspective on the multi-agent system. Using a global state-space model, a centralized Kalman filter will be considered. The next step is to find a way to distribute the centralized Kalman filter over the individual agents. Via this approach, we again arrive at the edge-based Kalman filter seen in the previous section, but also a distributed joint method is presented which is shown to have superior performance compared to the Edge-KF through abusing the coupled disturbance model.

First, we introduce a global state-space model in Section 5.1. The centralized global Kalman filter is introduced in Section 5.2. Then, in Section 5.3 a distributed adaptation of the centralized filter is proposed, dubbed the *joint edge-based Kalman filter*. Section 5.4 concludes the chapter with simulations showing the performance of the proposed filters.

5.1 Global State-Space Model

To consider the the multi-agent system from a global perspective, a state-space model of the multi-agent system is presented in this section. Initially, the state-space model is written in terms of the node states. However, we will introduce a state transformation which makes the global state a concatenation of the edge states instead. This will allow the direct application of a centralized edge-based Kalman filter in the next section.

In discrete-time, and with additive process noise we can write the dynamics of the global system as a concatenation of the local dynamics (4.9):

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \Delta t \mathbf{u}_k + \mathbf{w}_k \tag{5.1}$$

where $\mathbf{z}_k = \{\mathbf{z}_k^i\}_{i=1}^N \in \mathbb{R}^{DN}$ is a concatenated vector of the positions of the N mobile agents in D-dimensional space. Similarly, the control input $\mathbf{u}_k = \{\mathbf{u}_k^i\}_{i=1}^N \in \mathbb{R}^{DN}$ is a concatenation of the control inputs of all agents. The process noise, as described in Section 4.2 is Gaussian with zero mean and covariance matrix $\mathbf{Q}$. The relative positions can similarly be written as a linear function:

$$\mathbf{y}_k = (\mathbf{B}^\top \otimes \mathbf{H})\mathbf{z}_k + \mathbf{v}_k \tag{5.2}$$

where $\mathbf{y}_k \in \mathbb{R}^{2MTD}$ is a vector with all the observations made by agents. Remember that $\mathbf{H} = \mathbf{1}_T \otimes \mathbf{I}_D$ is the observation matrix, where T is the number of observations per filter update. M is the number of edges in the undirected sensing graph. Since every edge is bidirectional, the number of relative position observations is thus $2MT$. $\mathbf{B} \in \mathbb{R}^{N \times 2M}$ is the incidence matrix of the bidirectional sensing graph. Without loss of generality, the convention is chosen that the incidence matrix is ordered such that the columns are grouped per agent, starting with all edges directed towards agent 1 and ending with edges directed towards agent N. A similar convention is chosen in [37]. The observation noise $\mathbf{v}_k$ is, as introduced in Section 4.2, zero-mean Gaussian. Its covariance matrix $\mathbf{R}$ is block-diagonal with matrix $\mathbf{R}_{ij}$ for observations of agent i with respect to agent j. In case of homogeneous agents (i.e. $\mathbf{R}_{ij} = \mathbf{R}_D$ for all i, j) we can write the covariance matrix of $\mathbf{y}_k$ as $\mathbf{R} = \mathbf{I}_{2MT} \otimes \mathbf{R}_D$.

Now, note that using $\mathbf{z}_k$ as the state is not particularly useful for the formation control problem considered. After all, the local inputs to the control law (3.2) are relative positions. Since the measurements are also considered to be on the edges of the sensing graph, it is natural to also choose the edge state as the tracking variable.

We introduce the vector $\mathbf{x} \in \mathbb{R}^{2MD}$ which is a concatenation of all edge states. We have the following relation between node states $\mathbf{z}$ and edge states $\mathbf{x}$:

$$\mathbf{x} = (\mathbf{B}^\top \otimes \mathbf{I}_D)\mathbf{z} \tag{5.3}$$

We will use the matrix $\tilde{\mathbf{B}}^\top = (\mathbf{B}^\top \otimes \mathbf{I}_D)$ for compact notation. We can now transform the state space model by pre-multiplying the state update equation with $\tilde{\mathbf{B}}^\top$. The measurement equation can likewise be rewritten:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \tilde{\mathbf{B}}^\top \mathbf{u}_k + \tilde{\mathbf{B}}^\top \mathbf{w}_k \tag{5.4}$$

$$\mathbf{y}_k = (\mathbf{I}_{2M} \otimes \mathbf{H})\mathbf{x}_k + \mathbf{v}_k \tag{5.5}$$

This change of variable does not change the linearity of the system, and the noise is still Gaussian, although the distribution of the process noise $\bar{\mathbf{w}}_k = \tilde{\mathbf{B}}^\top \mathbf{w}_k$ becomes $\boldsymbol{\mathcal{N}}(\mathbf{0}_{ND}, \tilde{\mathbf{B}}^\top \mathbf{Q} \tilde{\mathbf{B}})$.

5.2 Centralized Edge Kalman Filter

Using the edge-based global state-space model introduced in 5.1, an optimal Kalman filter is introduced in this section which we call the *centralized edge-based Kalman filter* (C-Edge-KF) because the filter can only be applied centrally, using information from the whole multi-agent system. The advantages of distributability (robustness, scalability, parallel execution of tasks) have been discussed earlier. However, the centralized edge-based Kalman filter will be a starting point for the design of distributed filters.

With the linear and Gaussian state update (5.4) and measurement (5.5) equations, the standard Kalman initialization, prediction and update equations can be applied.

C-Edge-KF initialization

$$\begin{aligned}\hat{\mathbf{x}}_{0|0} &= \tilde{\mathbf{B}}^\top(\mathbf{1}_N \otimes \boldsymbol{\mu}) && (5.6)\\ &= (\mathbf{B}^\top \otimes \mathbf{I}_D)(\mathbf{1}_N \otimes \boldsymbol{\mu})\\ &= (\mathbf{B}^\top \mathbf{1}_N \otimes \boldsymbol{\mu})\\ &= \mathbf{0}_{2MD}\\ \boldsymbol{\Sigma}_{0|0} &= \tilde{\mathbf{B}}^\top(\mathbf{I}_N \otimes \mathbf{P})\tilde{\mathbf{B}} && (5.7)\\ &= (\mathbf{B}^\top \otimes \mathbf{I}_D)(\mathbf{I}_N \otimes \mathbf{P})(\mathbf{B} \otimes \mathbf{I}_D)\\ &= \mathbf{B}^\top\mathbf{B} \otimes \mathbf{P}\end{aligned}$$

Since we know that $\mathbf{B}^\top$, as the incidence matrix, has rows summing to 0, the product $\mathbf{B}^\top\mathbf{1}_N$ equals $\mathbf{0}_{2M}$. This means that no matter the initial distribution of the agents, the initial edge state estimates are always zero as long as the initial agent positions are independent and identically distributed (i.i.d.).

C-Edge-KF prediction step

$$\hat{\mathbf{x}}_{k+1|k} = \hat{\mathbf{x}}_{k|k} + \Delta t \tilde{\mathbf{B}}^\top \mathbf{u}_k \tag{5.8}$$

$$\boldsymbol{\Sigma}_{k+1|k} = \boldsymbol{\Sigma}_{k|k} + \tilde{\mathbf{B}}^\top \mathbf{Q} \tilde{\mathbf{B}} \tag{5.9}$$

C-Edge-KF update step

$$\mathbf{K}_k = \boldsymbol{\Sigma}_{k|k-1}\tilde{\mathbf{H}}^\top(\tilde{\mathbf{H}}\boldsymbol{\Sigma}_{k|k-1}\tilde{\mathbf{H}}^\top + \mathbf{R})^{-1} \tag{5.10}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \tilde{\mathbf{H}}\hat{\mathbf{x}}_{k|k-1}) \tag{5.11}$$

$$\boldsymbol{\Sigma}_{k|k} = (\mathbf{I}_{2MD} - \mathbf{K}_k\tilde{\mathbf{H}})\boldsymbol{\Sigma}_{k|k-1} \tag{5.12}$$

where $\tilde{\mathbf{H}} = (\mathbf{I}_{2M} \otimes \mathbf{H})$ is used for compact notation.

Algorithm 6 combines the Kalman filter iteration steps described in (5.6-5.12). Note again from this algorithm that the initialization, prediction and update steps are performed in one step for all agents.

The dimension of the estimated edge states $\hat{\mathbf{x}}$ is $2MD$ which is generally larger than ND which is the dimension of the concatenation of node states. As a result, one might argue that tracking node states is computationally more efficient than tracking edge states. This is true for the centralized filter. However, note that in a distributed filter, nodes would need to communicate estimated node states to neighbors in order to convert node states to edge states. Since we assume that agents do not share a common reference frame, this is not feasible. Through this reasoning, the choice of edge states as a tracking variable is justified.

Algorithm 6 Relative formation control using C-Edge-KF

```
Unknown global constants: $\boldsymbol{\mu}$
Known global constants: $D, N, M, T, \Delta t, \mathbf{P}, \mathbf{B}, \mathbf{H}, \mathbf{R}, \mathbf{Q}, \mathbf{L}$
$\tilde{\mathbf{B}} = \mathbf{B} \otimes \mathbf{I}_D$
$\tilde{\mathbf{H}} = \mathbf{I}_{2M} \otimes \mathbf{H}$
Draw initial positions $\mathbf{z}_1^i$ from $\boldsymbol{\mathcal{N}}(\boldsymbol{\mu}, \mathbf{P})$ for $i = 1, \ldots, N$
$\hat{\mathbf{x}}_{0|0} = \mathbf{0}_{2MD}$
$\boldsymbol{\Sigma}_{0|0} = \mathbf{B}^\top \mathbf{B} \otimes \mathbf{P}$
$\mathbf{u}_0 = \mathbf{0}_{DN}$
$k = 1$
while $k \leq k_{max}$ do
    $\hat{\mathbf{x}}_{k|k-1} = \hat{\mathbf{x}}_{k-1|k-1} + \Delta t \tilde{\mathbf{B}}^\top \mathbf{u}_{k-1}$
    $\boldsymbol{\Sigma}_{k|k-1} = \boldsymbol{\Sigma}_{k-1|k-1} + \tilde{\mathbf{B}}^\top \mathbf{Q} \tilde{\mathbf{B}}$
    Draw observation noise $\mathbf{v}_k$ from $\boldsymbol{\mathcal{N}}(\mathbf{0}_{2MDT}, \mathbf{R})$
    $\mathbf{y}_k = \tilde{\mathbf{H}} \tilde{\mathbf{B}}^\top \mathbf{z}_k + \mathbf{v}_k$                                   ▷ Measurements
    $\mathbf{K}_k = \boldsymbol{\Sigma}_{k|k-1} \tilde{\mathbf{H}}^\top (\tilde{\mathbf{H}} \boldsymbol{\Sigma}_{k|k-1} \tilde{\mathbf{H}}^\top + \mathbf{R})^{-1}$
    $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \tilde{\mathbf{H}} \hat{\mathbf{x}}_{k|k-1})$
    $\boldsymbol{\Sigma}_{k|k} = (\mathbf{I}_{2MD} - \mathbf{K}_k \tilde{\mathbf{H}}) \boldsymbol{\Sigma}_{k|k-1}$
    $\mathbf{u}_k^i = -\sum_{j \in \mathcal{N}_i} l_{ij} \hat{\mathbf{x}}_{k|k}^{ij}$ for $i = 1, \ldots, N$                    ▷ Control law
    Draw process noise $\mathbf{w}_k$ from $\boldsymbol{\mathcal{N}}(\mathbf{0}_{DN}, \mathbf{Q})$
    $\mathbf{z}_{k+1} = \mathbf{z}_k + \Delta t \mathbf{u}_k + \mathbf{w}_k$                              ▷ Dynamics update
    $k = k + 1$
end while
```

5.3 Joint Edge Kalman Filter

The main goal of this section is to adapt the centralized Kalman filter of Section 5.2 such that agents can track their own local states using only local information. In other words, we try to find a distributed solution. For this purpose, we will place sparsity restrictions on the structure of the Kalman gain matrix. It will be shown that a Kalman gain matrix with a certain block-diagonal structure allows distribution of the filter over the agents. Furthermore, we will explore how to compute the optimal Kalman gain subject to these sparsity constraints.

The following theorem states under which conditions on the Kalman gain matrix the centralized edge-based Kalman filter decouples and can be distributed over the individual agents.

Theorem 5.1. *The centralized edge-based Kalman filter given by the equations 5.6-*

5.12 decouples over the agents if the Kalman gain $\mathbf{K}_k$ is restricted such that it is a block-diagonal matrix with $\mathbf{K}_k^i \in \mathbb{R}^{|\mathcal{N}_i|D \times |\mathcal{N}_i|TD}$ the i^{th} of N block matrices.

Proof. We divide the state estimates $\hat{\mathbf{x}}_{k|k}$ into N local edge states denoted by $\hat{\mathbf{x}}_{k|k}^i \in \mathbb{R}^{|\mathcal{N}_i|D}$. Similarly, the covariance matrix $\mathbf{\Sigma}_{k|k}$ is split into the covariance matrices $\mathbf{\Sigma}_{k|k}^i \in \mathbb{S}_+^{|\mathcal{N}_i|D}$. For a block-diagonal Kalman gain matrix, the prediction and update equations can all be computed locally, provided communication between neighboring agents is available:

- The state prediction equation (5.8) is a vector sum which can be split into N sums that can be performed locally as long as all agents have access to the control inputs of their neighbors. I.e., agent i can compute the matrix product $(\mathbf{B}_i^\top \otimes \mathbf{I}_D)\mathbf{u}_k$, with $\mathbf{B}_i \in \mathbb{R}^{N \times |\mathcal{N}_i|}$ the matrix aggregating the columns of the incidence matrix $\mathbf{B}$ corresponding to edges directed toward agent i.

- The predictive covariance equation (5.9) as a matrix sum can be decoupled similarly. The disturbance covariance summand is the matrix $\tilde{\mathbf{B}}_i^\top \mathbf{Q} \tilde{\mathbf{B}}_i$, which thus only requires agent i to have (partial) knowledge of the constant matrix $\mathbf{Q}$.

- The state update equation (5.11) is decoupled since the block-diagonal structure of $\mathbf{K}_k$ ensures the edge measurements performed by agent i are only used for the updated state estimate of edges tracked by agent i. Hence, measurement information is only used locally. Similarly, the predicted local edge states $\hat{\mathbf{x}}_{k|k-1}^i$ are also used exclusively for the updated state estimate of agent i.

- The posterior covariance equation (5.12) describes the posterior covariance matrix only for the optimal Kalman gain provided in (5.10). Given that the block-diagonal Kalman gain matrix is generally not optimal, the Joseph form of the posterior global covariance matrix should be used instead:

$$\mathbf{\Sigma}_{k|k} = \mathbf{K}_k \mathbf{R} \mathbf{K}_k^\top + (\mathbf{I}_{2MD} - \mathbf{K}_k \tilde{\mathbf{H}}) \mathbf{\Sigma}_{k|k-1} (\mathbf{I}_{2MD} - \mathbf{K}_k \tilde{\mathbf{H}})^\top \tag{5.13}$$

  The matrix $\mathbf{I}_{2MD} - \mathbf{K}_k \tilde{\mathbf{H}}$ keeps the block-diagonal structure with the i^{th} block being $\mathbf{I}_{|\mathcal{N}_i|D} - \mathbf{K}_k^i \mathbf{H}$. Hence, the local covariance update $\mathbf{\Sigma}_{k|k}^i$ is a function of only the local gain matrix $\mathbf{K}_k^i$, the local measurement covariance $\mathbf{R}_i$ and the local prior covariance $\mathbf{\Sigma}_{k|k-1}^i$.

Since all Kalman filter iteration steps are shown to be decoupled, the whole filter can be said to be decoupled as well. □

While Theorem 5.1 states that the Kalman filter iterations can be decoupled given a block-diagonal Kalman gain matrix, we do not yet have a way to compute the optimal Kalman gain subject to the block-diagonality constraints, let alone a decentralized way to do so. The Kalman filter aims to minimize the error in a mean square error sense. Since the Kalman filter is unbiased, minimizing the mean square error is equivalent to minimizing the trace of the posterior covariance matrix:

$$\min_{\mathbf{K}_k} \quad \operatorname{tr}\left(\mathbf{\Sigma}_{k|k}\right) \tag{5.14}$$

This minimization problem can be rewritten with additional constraints on the structure of the gain matrix.

$$\begin{aligned} \min_{\mathbf{K}_k} \quad & \operatorname{tr}\left(\mathbf{\Sigma}_{k|k}\right) \\ \text{s.t.} \quad & \mathbf{K}_k \in \mathcal{K} \end{aligned} \tag{5.15}$$

where $\mathcal{K} \subseteq \mathbb{R}^{2MD \times 2MTD}$ is a set representing all matrices that satisfy some arbitrary sparsity pattern, i.e., with arbitrary elements forced to zero. In [37] a centralized closed-form solution to (5.15) is provided, which is valid for any sparsity pattern. However, in the special case of a block-diagonal sparsity constraint, we can solve (5.15) via a decentralized method, as shown via Theorem 5.2.

Theorem 5.2. *Let $\mathcal{K}$ denote the set of all block-diagonal matrices with $|\mathcal{N}_i|D \times |\mathcal{N}_i|TD$ the dimensions of the i^{th} of N block matrices. Then (5.15) is solved by*

$$\mathbf{K}_k^i = \mathbf{\Sigma}_{k|k-1}^i \mathbf{H}_i^\top \left(\mathbf{H}_i \mathbf{\Sigma}_{k|k-1}^i \mathbf{H}_i^\top + \mathbf{R}_i\right)^{-1} \tag{5.16}$$

for all i, with $\mathbf{H}_i = \mathbf{I}_{|\mathcal{N}_i|} \otimes \mathbf{H}$ and $\mathbf{R}_i$ a block-diagonal matrix with blocks $\mathbf{R}_{ij}$ with $j \in \mathcal{N}_i$.

Proof. First, the Joseph form of the posterior covariance (5.13) is substituted into (5.15):

$$\begin{aligned} \min_{\mathbf{K}_k} \quad & \operatorname{tr}\left(\mathbf{K}_k \mathbf{R} \mathbf{K}_k^\top\right) + \operatorname{tr}\left((\mathbf{I}_{2MD} - \mathbf{K}_k \tilde{\mathbf{H}}) \mathbf{\Sigma}_{k|k-1} (\mathbf{I}_{2MD} - \mathbf{K}_k \tilde{\mathbf{H}})^\top\right) \\ \text{s.t.} \quad & \mathbf{K}_k \in \mathcal{K} \end{aligned} \tag{5.17}$$

Note again that in addition to $\mathbf{K}_k$, the matrix $\mathbf{I}_{2MD} - \mathbf{K}_k \tilde{\mathbf{H}}$ is also block-diagonal. As a result, we rewrite the problem by applying Lemma 5.1:

$$\min_{\mathbf{K}_k^i} \quad \operatorname{tr}\left(\mathbf{K}_k^i \mathbf{R}_i \mathbf{K}_k^{i\top}\right) + \operatorname{tr}\left((\mathbf{I} - \mathbf{K}_k^i \mathbf{H}_i) \mathbf{\Sigma}_{k|k-1}^i (\mathbf{I} - \mathbf{K}_k^i \mathbf{H}_i)^\top\right), \quad i = 1, \ldots, N \tag{5.18}$$

Note that the problem is decoupled and the structural constraint on the gain matrix is gone. Without this constraint, (5.18) is solved by the standard optimal Kalman gain, which is shown in (5.16). □

Lemma 5.1. *With $\mathbf{A}$ a block-diagonal matrix and $\mathbf{F}$ a full matrix of appropriate sizes:*

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \ldots & \mathbf{0} & \mathbf{A}_N \end{bmatrix}, \mathbf{F} = \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} & \ldots & \mathbf{F}_{1N} \\ \mathbf{F}_{21} & \mathbf{F}_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{F}_{N1} & \ldots & \ldots & \mathbf{F}_{NN} \end{bmatrix}$$

the trace of $\mathbf{A}\mathbf{F}\mathbf{A}^\top$ is distributable:

$$\begin{aligned} \operatorname{tr}(\mathbf{A}\mathbf{F}\mathbf{A}^\top) = \operatorname{tr}(\mathbf{A}_1 \mathbf{F}_{11} \mathbf{A}_1^\top) &+ \operatorname{tr}(\mathbf{A}_2 \mathbf{F}_{22} \mathbf{A}_2^\top) \\ &+ \ldots \\ &+ \operatorname{tr}(\mathbf{A}_N \mathbf{F}_{NN} \mathbf{A}_N^\top) \end{aligned}$$

Since Theorems 5.1 and 5.2 allow the complete decoupling of the centralized edge-based Kalman filter with local optimal gain computation, we can now introduce a distributed edge-based Kalman filter which we will call the *joint edge-based Kalman filter* (J-Edge-KF). The adjective *joint* is used to indicate that agents jointly filter their adjacent edge states. The local initialization, prediction and update equations of the joint edge-based Kalman filter are now given for agent i.

J-Edge-KF initialization

$$\hat{\mathbf{x}}^i_{0|0} = \mathbf{0}_{|\mathcal{N}_i|D} \tag{5.19}$$

$$\mathbf{\Sigma}^i_{0|0} = \mathbf{B}_i^\top \mathbf{B}_i \otimes \mathbf{P} \tag{5.20}$$

J-Edge-KF prediction step

$$\hat{\mathbf{x}}^i_{k+1|k} = \hat{\mathbf{x}}^i_{k|k} + \Delta t \tilde{\mathbf{B}}_i^\top \mathbf{u}_k \tag{5.21}$$

$$\mathbf{\Sigma}^i_{k+1|k} = \mathbf{\Sigma}^i_{k|k} + \tilde{\mathbf{B}}_i^\top \mathbf{Q} \tilde{\mathbf{B}}_i \tag{5.22}$$

J-Edge-KF update step

$$\mathbf{K}^i_k = \mathbf{\Sigma}^i_{k|k-1} \mathbf{H}_i^\top \left(\mathbf{H}_i \mathbf{\Sigma}^i_{k|k-1} \mathbf{H}_i^\top + \mathbf{R}_i \right)^{-1} \tag{5.23}$$

$$\hat{\mathbf{x}}^i_{k|k} = \hat{\mathbf{x}}^i_{k|k-1} + \mathbf{K}^i_k \left(\mathbf{y}^i_k - \mathbf{H}_i \hat{\mathbf{x}}^i_{k|k-1} \right) \tag{5.24}$$

$$\mathbf{\Sigma}^i_{k|k} = \left(\mathbf{I}_{|\mathcal{N}_i|D} - \mathbf{K}^i_k \mathbf{H}_i \right) \mathbf{\Sigma}^i_{k|k-1} \tag{5.25}$$

Pseudocode used for simulating a noisy system using J-Edge-KF is shown in Algorithm 7. Note how, compared to the centralized filter in Algorithm 6, the prediction and update equations are performed locally. The only non-constant non-local information required for an agent to perform the algorithm is the control inputs of its neighbors, which can be achieved using local one-hop communication.

As a last remark, note that the edge-based Kalman filter (Edge-KF) introduced in Section 4.5 can be derived in a similar way to the joint edge-based filter. To do so, we tighten the Kalman gain matrix structure constraints such that the Kalman gain is block diagonal with $2M$ blocks of dimensions $D \times TD$. This decouples the filter per edge. The expected performance of the edge-based filter is lower than that of the joint edge-based filter, since the latter allows agents to use information gathered from all its edge state measurements jointly. The state space models of the edges are coupled via the disturbance model. This suggests that joint estimation can lead to improvement.

5.4 Simulations

The performance of the joint edge-based Kalman filter (J-Edge-KF) are compared against the MLE and Edge-KF filter in this section. The simulation setup is equivalent to the setup described in Sections 3.6 and 4.2.3.

Algorithm 7 Relative formation control using J-Edge-KF

```
Unknown global constants: $N, \boldsymbol{\mu}$
Known global constants: $D, T, \Delta t, \mathbf{P}, \mathbf{Q}$
Known local constants: $l_{ij}, \mathbf{B}_i, \mathbf{R}_i, \mathbf{H}_i$ for $i = 1, \ldots, N$, $j \in \mathcal{N}_i$
Draw initial positions $\mathbf{z}_1^i$ from $\boldsymbol{\mathcal{N}}(\boldsymbol{\mu}, \mathbf{P})$ for $i = 1, \ldots, N$
$\hat{\mathbf{x}}_{0|0}^i = \mathbf{0}_{|\mathcal{N}_i|D}$ for $i = 1, \ldots, N$
$\boldsymbol{\Sigma}_{0|0}^i = \mathbf{B}_i^\top \mathbf{B}_i \otimes \mathbf{P}$ for $i = 1, \ldots, N$
$\mathbf{u}_0 = \mathbf{0}_{DN}$
$k = 1$
while $k \leq k_{max}$ do
    for $i \in \mathcal{V}$ do
            $\hat{\mathbf{x}}_{k|k-1}^i = \hat{\mathbf{x}}_{k-1|k-1}^i + \Delta t \tilde{\mathbf{B}}_i^\top \mathbf{u}_{k-1}$
            $\boldsymbol{\Sigma}_{k|k-1} = \boldsymbol{\Sigma}_{k-1|k-1} + \tilde{\mathbf{B}}^\top \mathbf{Q} \tilde{\mathbf{B}}$
        Draw observation noise $\mathbf{v}_k^i$ from $\boldsymbol{\mathcal{N}}(\mathbf{0}_{|\mathcal{N}_i|DT}, \mathbf{R}_i)$
        $\mathbf{y}_k^i = \mathbf{H}_i \mathbf{B}_i^\top \mathbf{z}_k + \mathbf{v}_k^i$                ▷ Measurements
            $\mathbf{K}_k^i = \boldsymbol{\Sigma}_{k|k-1}^i \mathbf{H}_i^\top (\mathbf{H}_i \boldsymbol{\Sigma}_{k|k-1}^i \mathbf{H}_i^\top + \mathbf{R}_i)^{-1}$
            $\hat{\mathbf{x}}_{k|k}^i = \hat{\mathbf{x}}_{k|k-1}^i + \mathbf{K}_k^i (\mathbf{y}_k^i - \mathbf{H}_i \hat{\mathbf{x}}_{k|k-1}^i)$
            $\boldsymbol{\Sigma}_{k|k}^i = (\mathbf{I}_{|\mathcal{N}_i|D} - \mathbf{K}_k^i \mathbf{H}_i) \boldsymbol{\Sigma}_{k|k-1}^i$
        $\mathbf{u}_k^i = -\sum_{j \in \mathcal{N}_i} l_{ij} \hat{\mathbf{x}}_{k|k}^{ij}$                ▷ Control law
    end for
    Draw process noise $\mathbf{w}_k$ from $\boldsymbol{\mathcal{N}}(\mathbf{0}_{DN}, \mathbf{Q})$
    $\mathbf{z}_{k+1} = \mathbf{z}_k + \Delta t \mathbf{u}_k + \mathbf{w}_k$                ▷ Dynamics update
    $k = k + 1$
end while
```

First, we consider the evolution of the trace of the posterior covariance matrix, shown in Figure 5.1.

The posterior covariance of the MLE is constant as given by (4.17). The Kalman filters initially have the same uncertainty, but over time the uncertainty decreases. As expected, the joint filter has smaller steady-state uncertainty compared to the Edge-KF, due to the coupling of the edge states via the disturbance correlation which is exploited by the joint filter. The centralized Kalman filter has even lower steady-state uncertainty, as is expected being the optimal filter for the global system.

The estimation errors show a similar story. In Figure 5.2 the norm of the difference between the true edge states and the estimated edge states is plotted.

Again, we note that the joint approach delivers superior performance in the form of a lower estimation error. Convergence happens at a similar time-scale compared to

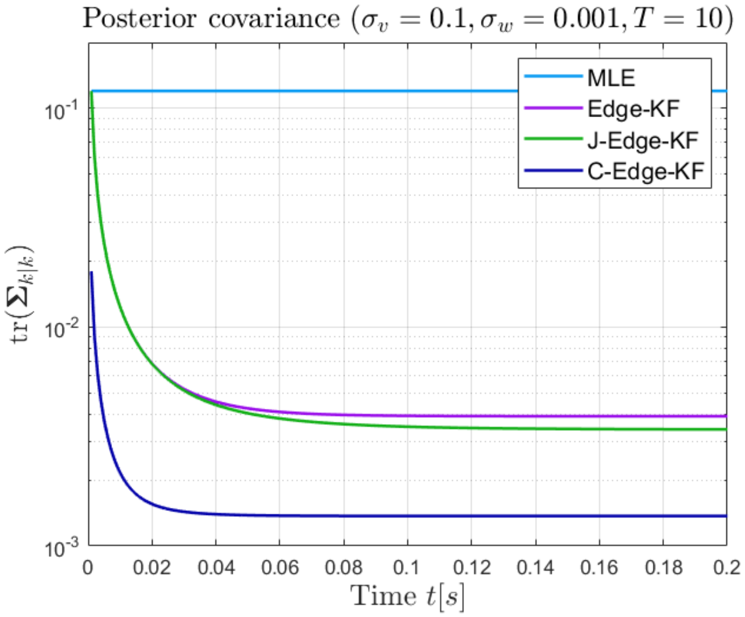


Figure 5.1: Trace of the posterior covariance matrix over time.

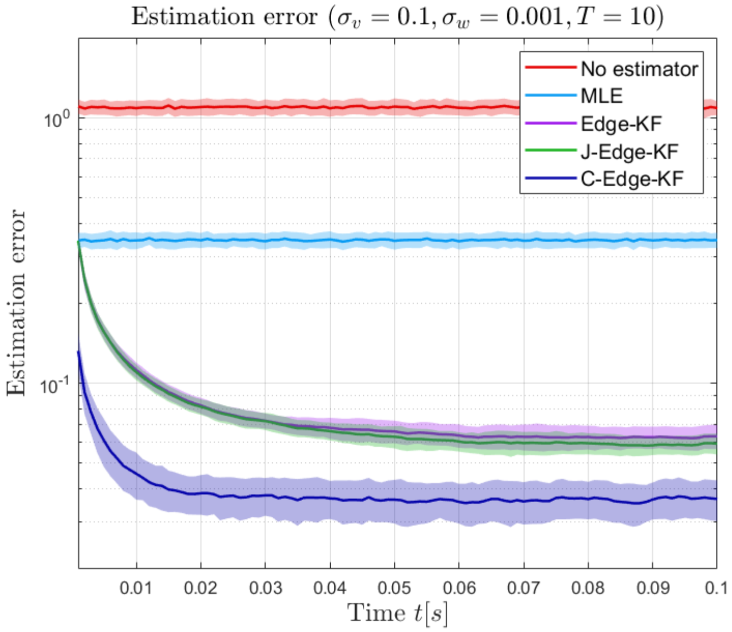


Figure 5.2: Estimation errors ϵ_k over time. The plot shows the mean estimation error for 50 runs of Algorithms 2, 3, 5, 6 and 7, and the ±1 standard deviation regions.

the posterior covariance.

Lastly, the Procrustes errors are compared in Figure 5.3. Differences between the algorithms are less obvious although again the centralized Kalman filter seems to have the smallest steady-state Procrustes error, followed by the joint Kalman filter, the edge

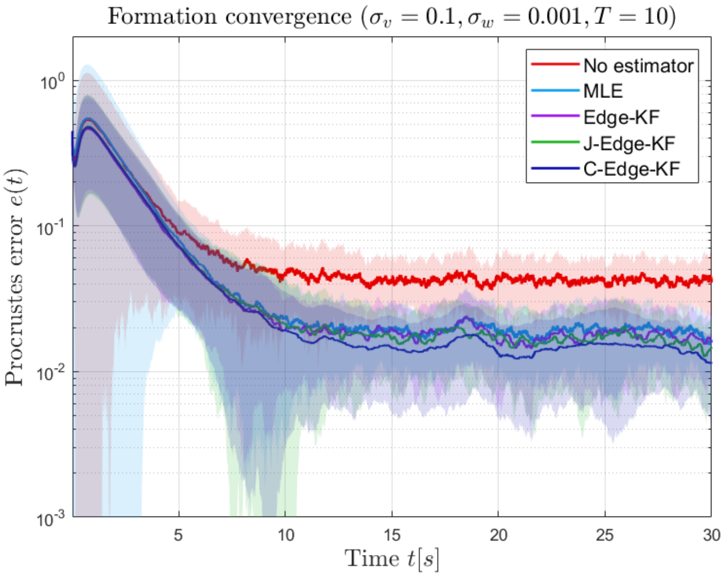


Figure 5.3: Procrustes errors over time. The plot shows the mean Procrustes error for 50 runs of Algorithms 2, 3, 5, 6 and 7, and the ±1 standard deviation regions.

Kalman filter and the MLE respectively. Convergence speed is practically equal for all algorithms, and is similar to the convergence speed in the noiseless case.

6 Robust Communication-less Edge State Filtering

The distributed Kalman filters proposed in the previous chapters are distributed in the sense that the filters can be run locally. However, the filters can not be run completely without inter-agent dependency, as seen from the dynamics of the edge (i, j) given by (4.24) and repeated here:

$$\mathbf{z}_{k+1}^{ij} = \mathbf{z}_k^{ij} + \Delta t(\mathbf{u}_k^i - \mathbf{u}_k^j) + \mathbf{w}_k^{ij}, \quad \mathbf{w}_k^{ij} \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{Q}_{ij}) \tag{6.1}$$

To make an unbiased prediction of the evolution of the edge state, the control inputs of both adjacent nodes are needed. Therefore, agent i needs not only its own control input $\mathbf{u}_k^i$, but also the control input $\mathbf{u}_k^j$ of its neighboring agent j to make an unbiased prediction on the evolution of the edge state $\mathbf{z}_k^{ij}$. In multi-agent systems with robust neighbor-to-neighbor data links, agents can communicate their computed control input to neighbors and this allows the distributed Kalman filters to run as intended. However, in this section two scenarios are explored where agents may not have direct access to the control input of their neighbors.

1. The first scenario is motivated by a desire for the formation control system to be robust to intermittently failing communication links. We will assume that generally, communication links are available between neighboring agents. However, at random time steps communication may be delayed or a communication link is temporarily suspended. As a result, the control input $\mathbf{u}_k^j$ of agent j at time step k is considered unknown to agent i. However, agent i has access to all past control inputs of agent j, i.e., $\mathbf{u}_l^j, l = 1, \ldots, k-1$. The problem to be solved is a one-step prediction. The estimate $\hat{\mathbf{u}}_k^j$ can then be used in the prediction step of the Kalman filter.

2. The second scenario is motivated by a desire to design a filter that functions without *any* neighbor-to-neighbor communication. In fact, one of the main advantages of the control law for formation control proposed in [25] is the lack of a inter-agent communication requirement. In many mobile multi-agent systems, not requiring a communication module on every agent is a great advantage. As such, an attempt will be made to estimate the control input $\mathbf{u}_k^j$ of agent j at time step k based on local information known to agent i. The estimate $\hat{\mathbf{u}}_k^j$ can then be used in the prediction step of the Kalman filter.

For both scenarios, the eventual goal is to compute an estimate $\hat{\mathbf{u}}_k^j$ and a posterior covariance matrix $\boldsymbol{\Sigma}_{\mathbf{u},k}^{ij}$. We can then rewrite the prediction equations of the Kalman filter as

$$\hat{\mathbf{z}}_{k+1|k}^{ij} = \hat{\mathbf{z}}_{k|k}^{ij} + \Delta t(\mathbf{u}_k^i - \hat{\mathbf{u}}_k^j) \tag{6.2}$$

$$\boldsymbol{\Sigma}_{k+1|k}^{ij} = \boldsymbol{\Sigma}_{k|k}^{ij} + \mathbf{Q}_{ij} + \Delta t^2 \boldsymbol{\Sigma}_{\mathbf{u},k}^{ij} \tag{6.3}$$

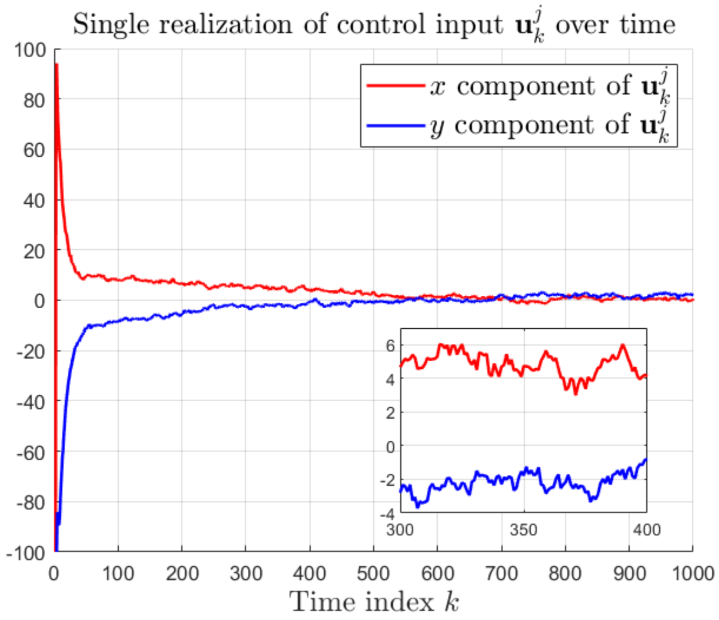


Figure 6.1: One realization of the control input process for a single agent.

Note that by making this change, we are modelling the control input of the neighbor as if it were a normally distributed random variable:

$$\mathbf{u}_k^j \sim \mathcal{N}(\hat{\mathbf{u}}_k^j, \mathbf{\Sigma}_{\mathbf{u},k}^{ij}) \tag{6.4}$$

In other words, we assume the true control input of agent j to be a random vector normally distributed and centered around the estimated control input $\hat{\mathbf{u}}_k^j$. The posterior predictive covariance matrix $\mathbf{\Sigma}_{\mathbf{u},k}^{ij}$ represents the uncertainty in the estimate $\hat{\mathbf{u}}_k^j$.

6.1 Scenario 1: Random Communication Link Failure

Given the control inputs of agent j up until one time step back, the challenge for agent i is to make a one-step ahead prediction which should result in an estimate $\hat{\mathbf{u}}_k^j$. Note that, unlike the state evolution of agents, there is no direct local model for the evolution of control inputs over time. From the linear control law (3.2), we notice that the control input of agent j depends on the states of neighbors of j in the network. These states are again dependent on control inputs of neighbors of j which again depend on neighbors of neighbors, etc. In the end, agent i lacks all the information necessary to model the evolution of the control input of its neighbor j using the control law (3.2). Therefore, a simpler local model is needed which models the evolution of $\mathbf{u}_k^j$ over time, and allows agent i to extrapolate the past control inputs one step into the future.

A single realization of the control input of an agent is shown in Figure 6.1. Clearly, the control input process shows some form of autocorrelation, i.e. samples show some form of dependence on past values. One of the options to model the control inputs over time is an autoregressive model of order p, denoted AR(p). This is a model of the following form:

$$\mathbf{u}_t^j = \sum_{l=1}^{p} \mathbf{a}_l \circ \mathbf{u}_{t-l}^j + \mathbf{n}_t^j, \quad \mathbf{n}_t^j \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \boldsymbol{\Sigma}_\mathbf{n}) \tag{6.5}$$

where the vectors $\mathbf{a}_l, l = 1, \ldots, p$ are the model coefficients, and $\mathbf{n}_t^j$ is white noise which is independent of the past control inputs $\mathbf{u}_l^j, l = 1, \ldots, t-1$.

There are various methods of finding a suitable model order p and estimating the model coefficients. One could fit the coefficients based on recorded training sequences, using the Yule-Walker equations.
Note furthermore that by using the model (6.5), the evolution of the control input of agent j is assumed to be a zero-mean wide sense stationary random process. In steady state (i.e. agents are in formation) this assumption is generally good. For scenarios where agents are initially in random positions, the control input process is generally not first order stationary (see e.g. Figure 6.1) and additionally the mean is generally not zero, which may limit the accuracy of the model. However, for AR processes of small orders the stationarity assumption may be a good approximation.

Now, the problem to be solved is that of one-step prediction of $\mathbf{u}_k^j$ given that the evolution of the control inputs of agent j is described by an AR(p) process of the form of (6.5), and given the set of past observations $\mathbf{u}_l^j, l = k-p, \ldots, k-1$.

We will attempt to make a linear prediction using the Wiener-Hopf equations. The linear filter has the following form:

$$\hat{\mathbf{u}}_k^j = \sum_{l=1}^{p} \mathbf{W}_l \mathbf{u}_{k-l}^j \tag{6.6}$$

To derive the optimal filter coefficients $\mathbf{W}_l$ via the Wiener-Hopf equations, first we need to introduce the autocorrelation matrix of the control input process. We define the autocorrelation matrix at lag τ of the control input process as

$$\mathbf{R}_\mathbf{u}(\tau) = \mathbb{E}\left(\mathbf{u}_t^j \mathbf{u}_{t+\tau}^{j\top}\right) \tag{6.7}$$

which, due to the assumption of wide sense stationarity, does not depend on the time index t but only on the time difference τ. The Wiener-Hopf equations for the autoregressive process of order p are

$$\begin{aligned} \begin{bmatrix} \mathbf{R}_\mathbf{u}(0) & \ldots & \mathbf{R}_\mathbf{u}(p-1) \\ \vdots & \ddots & \vdots \\ \mathbf{R}_\mathbf{u}(p-1) & \ldots & \mathbf{R}_\mathbf{u}(0) \end{bmatrix} \begin{bmatrix} \mathbf{W}_1 \\ \vdots \\ \mathbf{W}_p \end{bmatrix} &= \begin{bmatrix} \mathbf{R}_\mathbf{u}(1) \\ \vdots \\ \mathbf{R}_\mathbf{u}(p) \end{bmatrix} \\ \mathbf{R}_{\mathbf{uu},p}\mathbf{W} &= \mathbf{R}_{\mathbf{u},1} \end{aligned} \tag{6.8}$$

Solving for $\mathbf{W}$ gives us $\mathbf{W} = \mathbf{R}_{\mathbf{uu},p}^{-1}\mathbf{R}_{\mathbf{u},1}$.

Let us see what the Wiener predictor looks like for an AR(1) process with model coefficients $\mathbf{a}_1$. The autocorrelation matrix for this model is $\mathbf{R}_\mathbf{u}(\tau) = \mathrm{diag}(\mathbf{a}_1)^{|\tau|}$. The Wiener-Hopf equations then become

$$\begin{aligned} \mathbf{R}_\mathbf{u}(0)\mathbf{W}_1 &= \mathbf{R}_\mathbf{u}(1) \\ \mathbf{W}_1 &= \mathrm{diag}(\mathbf{a}_1) \end{aligned} \tag{6.9}$$

which means the prediction for the control input at time step k is given by $\hat{\mathbf{u}}_k^j = \mathbf{a}_1 \circ \mathbf{u}_{k-1}^j$, which makes sense, considering the model (6.5) holds.

The posterior predictive covariance $\boldsymbol{\Sigma}_{\mathbf{u},k}^{ij}$ represents the uncertainty in the prediction $\hat{\mathbf{u}}_k^j$. When the past control inputs are known, the uncertainty in the prediction is only caused by the process noise $\mathbf{n}_k^j$. The posterior predictive covariance is thus given by

$$\boldsymbol{\Sigma}_{\mathbf{u},k}^{ij} = \boldsymbol{\Sigma}_{\mathbf{n}} \tag{6.10}$$

which is a parameter of the autoregressive model that can be estimated by performing trial runs for the desired system.

The pseudocode for the simulations for Scenario 1 is presented in Algorithm 8.

6.2 Scenario 2: Full Communication-less Setting

For the second scenario, we consider a setting where inter-agent communication is not available, due to e.g., environmental or hardware constraints. As a result, agent i lacks not only the current control input of its neighbor j, but also all past control inputs of j. For this reason, we split the problem of estimation of $\mathbf{u}_k^j$ into two sub-problems. First, estimates $\hat{\mathbf{u}}_l^j, l = 1, \dots, k-1$ are made based on all past measurements $\mathbf{y}_l^{ij}, l = 1, \dots, k$. Then, a one-step predictor is designed which estimates the control input $\mathbf{u}_k^j$ based on all past estimated control inputs $\hat{\mathbf{u}}_l^j, l = 1, \dots, k-1$.

At time step t, after obtaining the measurement $\mathbf{y}_t^{ij}$, the state estimate $\hat{\mathbf{z}}_{t|t}^{ij}$ is available. We will use these and the previous state estimate to get a model for the control input $\mathbf{u}_{t-1}^j$. We start again with the edge dynamics equation (6.1), but shifted back one time step:

$$\mathbf{z}_t^{ij} = \mathbf{z}_{t-1}^{ij} + \Delta t(\mathbf{u}_{t-1}^i - \mathbf{u}_{t-1}^j) + \mathbf{w}_{t-1}^{ij} \tag{6.11}$$

We now substitute the state estimates $\hat{\mathbf{z}}_{t|t}^{ij}$ and $\hat{\mathbf{z}}_{t-1|t-1}^{ij}$, and rewrite such that all the known terms are on the left-hand side:

$$\frac{1}{\Delta t}\left(\hat{\mathbf{z}}_{t-1|t-1}^{ij} - \hat{\mathbf{z}}_{t|t}^{ij}\right) + \mathbf{u}_{t-1}^i = \mathbf{u}_{t-1}^j - \frac{1}{\Delta t}\mathbf{w}_{t-1}^{ij} \tag{6.12}$$

Since the disturbance $\mathbf{w}_{t-1}^{ij}$ is assumed to be zero-mean, the maximum likelihood estimator for $\mathbf{u}_{t-1}^j$ is given by

$$\hat{\mathbf{u}}_{t-1}^j = \frac{1}{\Delta t}\left(\hat{\mathbf{z}}_{t-1|t-1}^{ij} - \hat{\mathbf{z}}_{t|t}^{ij}\right) + \mathbf{u}_{t-1}^i \tag{6.13}$$

Keeping in mind that the state estimates have posterior covariances $\boldsymbol{\Sigma}_{t|t}^{ij}$ and $\boldsymbol{\Sigma}_{t-1|t-1}^{ij}$ respectively, we can write the posterior covariance of the estimate $\hat{\mathbf{u}}_{t-1}^j$ as

$$\boldsymbol{\Sigma}_{\hat{\mathbf{u}},t-1}^{ij} = \frac{1}{\Delta t^2}\left(\mathbf{Q}_{ij} + \boldsymbol{\Sigma}_{t|t}^{ij} + \boldsymbol{\Sigma}_{t-1|t-1}^{ij}\right) \tag{6.14}$$

Note that for an increasing Kalman filter frequency (i.e. a decreasing Δt), the uncertainty in the estimate increases with inverse squared proportionality. Hence, for

Algorithm 8 Relative formation control using Edge-KF with random link failures

```
Unknown global constants: $N, \boldsymbol{\mu}, P_{\text{fail}}$
Known global constants: $D, T, \Delta t, \mathbf{H}, \mathbf{P}, \mathbf{a}_1, \boldsymbol{\Sigma}_{\mathbf{n}}$
Known local constants: $l_{ij}, \mathbf{R}_{ij}, \mathbf{Q}_{ij}$ for $i = 1, \ldots, N,\ j \in \mathcal{N}_i$
$\tilde{\mathbf{R}}_{ij} = \mathbf{I}_T \otimes \mathbf{R}_{ij}$ for $i = 1, \ldots, N,\ j \in \mathcal{N}_i$
Draw initial positions $\mathbf{z}_1^i$ from $\boldsymbol{\mathcal{N}}(\boldsymbol{\mu}, \mathbf{P})$
$\hat{\mathbf{z}}_{0|0}^{ij} = \mathbf{0}_D$ for all $i \in \mathcal{V}, j \in \mathcal{N}_i$
$\boldsymbol{\Sigma}_{0|0}^{ij} = 2\mathbf{P}$ for all $i \in \mathcal{V}, j \in \mathcal{N}_i$
$\mathbf{u}_0^i = \mathbf{0}_D$ for all $i \in \mathcal{V}$
$k = 1$
while $k \leq k_{max}$ do
    Draw process noise $\mathbf{w}_k$ from $\boldsymbol{\mathcal{N}}(\mathbf{0}_{DN}, \mathbf{Q})$
    for $i \in \mathcal{V}$ do
        for $j \in \mathcal{N}_i$ do
            Draw random link failure variable $p_k^{ij}$ from $\boldsymbol{\mathcal{U}}(0, 1)$
            if $p < P_{\text{fail}}$ then                                          ▷ Link failure
                $\hat{\mathbf{u}}_{k-1}^j = \mathbf{a}_1 \circ \mathbf{u}_{k-2}^j$                               ▷ One-step prediction
                $\hat{\mathbf{z}}_{k|k-1}^{ij} = \hat{\mathbf{z}}_{k-1|k-1}^{ij} + \Delta t \mathbf{u}_{k-1}^i - \Delta t \hat{\mathbf{u}}_{k-1}^j$
                $\boldsymbol{\Sigma}_{k|k-1}^{ij} = \boldsymbol{\Sigma}_{k-1|k-1}^{ij} + \mathbf{Q}_{ij} + \boldsymbol{\Sigma}_{\mathbf{n}}$
            else                                                                ▷ No link failure
                $\hat{\mathbf{z}}_{k|k-1}^{ij} = \hat{\mathbf{z}}_{k-1|k-1}^{ij} + \Delta t \mathbf{u}_{k-1}^i - \Delta t \mathbf{u}_{k-1}^j$
                $\boldsymbol{\Sigma}_{k|k-1}^{ij} = \boldsymbol{\Sigma}_{k-1|k-1}^{ij} + \mathbf{Q}_{ij}$
            end if
            Draw observation noise $\mathbf{v}_k^{ij}$ from $\boldsymbol{\mathcal{N}}(\mathbf{0}_{DT}, \tilde{\mathbf{R}}_{ij})$
            $\mathbf{y}_k^{ij} = \mathbf{H}(\mathbf{z}_k^i - \mathbf{z}_k^j) + \mathbf{v}_k^{ij}$                              ▷ Measurements
            $\mathbf{K}_k^{ij} = \boldsymbol{\Sigma}_{k|k-1}^{ij} \mathbf{H}^\top (\mathbf{H} \boldsymbol{\Sigma}_{k|k-1}^{ij} \mathbf{H}^\top + \tilde{\mathbf{R}}_{ij})^{-1}$
            $\hat{\mathbf{z}}_{k|k}^{ij} = \hat{\mathbf{z}}_{k|k-1}^{ij} + \mathbf{K}_k^{ij} (\mathbf{y}_k^{ij} - \mathbf{H} \hat{\mathbf{z}}_{k|k-1}^{ij})$
            $\boldsymbol{\Sigma}_{k|k}^{ij} = (\mathbf{I}_D - \mathbf{K}_k^{ij} \mathbf{H}) \boldsymbol{\Sigma}_{k|k-1}^{ij}$
        end for
        $\mathbf{u}_k^i = -\sum_{j \in \mathcal{N}_i} l_{ij} \hat{\mathbf{z}}_{k|k}^{ij}$                                    ▷ Control law
        $\mathbf{z}_{k+1}^i = \mathbf{z}_k^i + \Delta t \mathbf{u}_k^i + \mathbf{w}_k^i$                             ▷ Dynamics update
    end for
    $k = k + 1$
end while
```

systems with high Kalman filter frequency, it may be more advantageous to use some sort of averaging filter for the estimation of $\mathbf{u}_{t-1}^{j}$, with the underlying assumption that the control inputs of agents do not change abruptly over time.

Given that the estimates $\hat{\mathbf{u}}_{l}^{j}, l = 1, \ldots, k-1$ are all available for agent i, the next step is to extrapolate this sequence to predict the control input at time index k. This step is similar to the approach taken in the scenario with intermittent communication link failures. However, the difference here is that the past observations of the random process are corrupted by noise. We will write the observation at time t as the true control input with additive white Gaussian noise with covariance $\boldsymbol{\Sigma}_{\hat{\mathbf{u}},t}^{ij}$:

$$\hat{\mathbf{u}}_{t}^{j} = \mathbf{u}_{t}^{j} + \mathbf{m}_{t}^{j}, \quad \mathbf{m}_{t}^{j} \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_{D}, \boldsymbol{\Sigma}_{\hat{\mathbf{u}},t}^{ij}) \tag{6.15}$$

Note that the covariance matrix $\boldsymbol{\Sigma}_{\hat{\mathbf{u}},t}^{ij}$ depends on the time index t. We can write the autocorrelation matrix of the noisy process as

$$\begin{aligned}
\mathbf{R}_{\hat{\mathbf{u}}}(t,\tau) &= \mathbb{E}\left(\hat{\mathbf{u}}_{t}^{j}\hat{\mathbf{u}}_{t+\tau}^{j\top}\right) \\
&= \mathbb{E}\left((\mathbf{u}_{t}^{j} + \mathbf{m}_{t}^{j})(\mathbf{u}_{t+\tau}^{j} + \mathbf{m}_{t+\tau}^{j})^{\top}\right) \\
&= \mathbb{E}\left(\mathbf{u}_{t}^{j}\mathbf{u}_{t+\tau}^{j\top}\right) + \mathbb{E}\left(\mathbf{u}_{t}^{j}\mathbf{m}_{t+\tau}^{j\top}\right) + \mathbb{E}\left(\mathbf{m}_{t}^{j}\mathbf{u}_{t+\tau}^{j\top}\right) + \mathbb{E}\left(\mathbf{m}_{t}^{j}\mathbf{m}_{t+\tau}^{j\top}\right) \\
&= \mathbf{R}_{\mathbf{u}}(\tau) + \delta(\tau)\boldsymbol{\Sigma}_{\hat{\mathbf{u}},t}^{ij}
\end{aligned} \tag{6.16}$$

where the delta function $\delta(\tau)$ reflects the fact that the noise process is white, i.e. samples of $\mathbf{m}_{t}^{j}$ are uncorrelated. Note that since the covariance matrix $\boldsymbol{\Sigma}_{\hat{\mathbf{u}},t}^{ij}$ depends on the time index t, the autocorrelation matrix will also change over time, and thus we no longer consider the process to be second order stationary. For the AR(1) process, the optimal Wiener filter for the prediction of the control input at time step k is then given by

$$\begin{aligned}
\mathbf{W}_{1} &= \mathbf{R}_{\hat{\mathbf{u}}}(k-1,0)^{-1}\mathbf{R}_{\mathbf{u}}(1) \\
&= \operatorname{diag}(\mathbf{a}_{1})\left(\mathbf{I}_{D} + \boldsymbol{\Sigma}_{\hat{\mathbf{u}},k-1}^{ij}\right)^{-1}
\end{aligned} \tag{6.17}$$

The posterior predictive covariance of the predictor with noisy past observations is a function of both the uncertainty introduced in the evolution model (via the noise $\mathbf{n}_{k}^{j}$), and the prior uncertainty in the past observations. We can write the posterior predictive covariance as

$$\boldsymbol{\Sigma}_{\mathbf{u},k}^{ij} = \sum_{l=1}^{p} \operatorname{diag}(\mathbf{a}_{l})^{2}\boldsymbol{\Sigma}_{\hat{\mathbf{u}},k-l}^{ij} + \boldsymbol{\Sigma}_{\mathbf{n}} \tag{6.18}$$

Algorithm 9 contains pseudocode for the simulations without inter-agent communication.

6.3 Simulations

For both scenarios simulation results are presented here. Simulation setup was generally equivalent to earlier simulations as described in Section 3.6 and Section 4.2.3. For

Algorithm 9 Relative formation control using Edge-KF without communication

```
Unknown global constants: $N, \boldsymbol{\mu}$
Known global constants: $D, T, \Delta t, \mathbf{H}, \mathbf{P}, \mathbf{a}_1, \boldsymbol{\Sigma}_{\mathbf{n}}$
Known local constants: $l_{ij}, \mathbf{R}_{ij}, \mathbf{Q}_{ij}$ for $i = 1, \dots, N,\ j \in \mathcal{N}_i$
$\tilde{\mathbf{R}}_{ij} = \mathbf{I}_T \otimes \mathbf{R}_{ij}$ for $i = 1, \dots, N,\ j \in \mathcal{N}_i$
Draw initial positions $\mathbf{z}^i_1$ from $\boldsymbol{\mathcal{N}}(\boldsymbol{\mu}, \mathbf{P})$
$\hat{\mathbf{z}}^{ij}_{0|0} = \mathbf{0}_D$ for all $i \in \mathcal{V}, j \in \mathcal{N}_i$
$\boldsymbol{\Sigma}^{ij}_{0|0} = 2\mathbf{P}$ for all $i \in \mathcal{V}, j \in \mathcal{N}_i$
$\mathbf{u}^i_0 = \mathbf{0}_D$ for all $i \in \mathcal{V}$
$k = 1$
while $k \leq k_{max}$ do
    Draw process noise $\mathbf{w}_k$ from $\boldsymbol{\mathcal{N}}(\mathbf{0}_{DN}, \mathbf{Q})$
    for $i \in \mathcal{V}$ do
        for $j \in \mathcal{N}_i$ do
            $\hat{\mathbf{u}}^j_{k-2} = \frac{1}{\Delta t}\left(\hat{\mathbf{z}}^{ij}_{k-2|k-2} - \hat{\mathbf{z}}^{ij}_{k-1|k-1}\right) + \mathbf{u}^i_{k-2}$    ▷ Delayed estimate
            $\boldsymbol{\Sigma}^{ij}_{\hat{\mathbf{u}},k-2} = \frac{1}{\Delta t^2}\left(\mathbf{Q}_{ij} + \boldsymbol{\Sigma}^{ij}_{k-1|k-1} + \boldsymbol{\Sigma}^{ij}_{k-2|k-2}\right)$    ▷ Post. cov. delayed estimate
            $\mathbf{W}_1 = \mathrm{diag}(\mathbf{a}_1)\left(\mathbf{I}_D + \boldsymbol{\Sigma}^{ij}_{\hat{\mathbf{u}},k-2}\right)^{-1}$
            $\hat{\mathbf{u}}^j_{k-1} = \mathbf{W}_1 \hat{\mathbf{u}}^j_{k-2}$    ▷ One-step prediction
            $\hat{\mathbf{z}}^{ij}_{k|k-1} = \hat{\mathbf{z}}^{ij}_{k-1|k-1} + \Delta t \mathbf{u}^i_{k-1} - \Delta t \hat{\mathbf{u}}^j_{k-1}$
            $\boldsymbol{\Sigma}^{ij}_{k|k-1} = \boldsymbol{\Sigma}^{ij}_{k-1|k-1} + \mathbf{Q}_{ij} + \mathrm{diag}(\mathbf{a}_1)^2 \boldsymbol{\Sigma}^{ij}_{\hat{\mathbf{u}},k-2} + \boldsymbol{\Sigma}_{\mathbf{n}}$
            Draw observation noise $\mathbf{v}^{ij}_k$ from $\boldsymbol{\mathcal{N}}(\mathbf{0}_{DT}, \tilde{\mathbf{R}}_{ij})$
            $\mathbf{y}^{ij}_k = \mathbf{H}(\mathbf{z}^i_k - \mathbf{z}^j_k) + \mathbf{v}^{ij}_k$    ▷ Measurements
            $\mathbf{K}^{ij}_k = \boldsymbol{\Sigma}^{ij}_{k|k-1}\mathbf{H}^\top(\mathbf{H}\boldsymbol{\Sigma}^{ij}_{k|k-1}\mathbf{H}^\top + \tilde{\mathbf{R}}_{ij})^{-1}$
            $\hat{\mathbf{z}}^{ij}_{k|k} = \hat{\mathbf{z}}^{ij}_{k|k-1} + \mathbf{K}^{ij}_k(\mathbf{y}^{ij}_k - \mathbf{H}\hat{\mathbf{z}}^{ij}_{k|k-1})$
            $\boldsymbol{\Sigma}^{ij}_{k|k} = (\mathbf{I}_D - \mathbf{K}^{ij}_k \mathbf{H})\boldsymbol{\Sigma}^{ij}_{k|k-1}$
        end for
        $\mathbf{u}^i_k = -\sum_{j \in \mathcal{N}_i} l_{ij} \hat{\mathbf{z}}^{ij}_{k|k}$    ▷ Control law
        $\mathbf{z}^i_{k+1} = \mathbf{z}^i_k + \Delta t \mathbf{u}^i_k + \mathbf{w}^i_k$    ▷ Dynamics update
    end for
    $k = k + 1$
end while
```

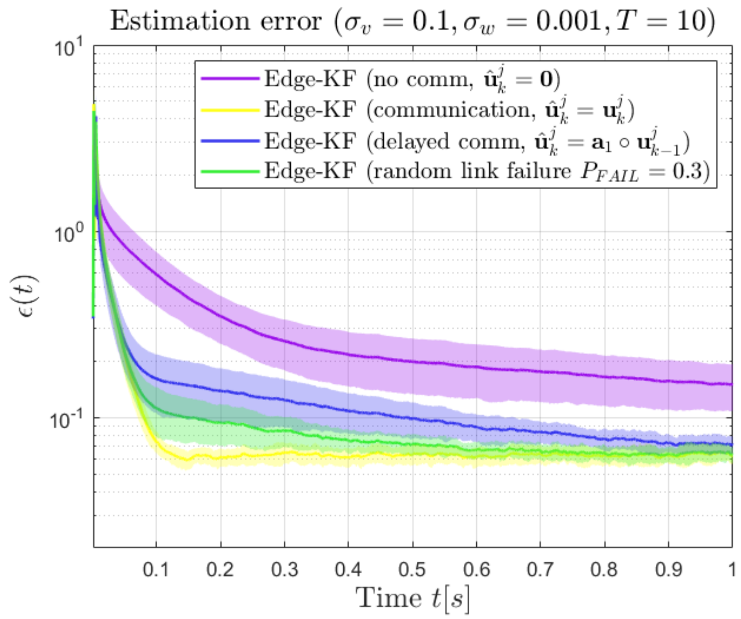


Figure 6.2: Estimation errors for randomly failing communication scenario with observation and process noise. The plot shows the mean estimation error for 50 runs of Algorithm 8, and the ±1 standard deviation regions.

the random failure model, we consider a constant scalar probability $P_{\text{fail}} \in [0, 1]$ of a communication link loss at time index k. The probability of failure of a link is independent of the probability of failure of all other links. Past control inputs of neighbors are assumed to be known for all $l = 1, \ldots, k-1$. The control input evolution model is taken to be an AR(1) process with $\mathbf{a}_1 = 0.8\mathbf{1}_D$, chosen based on trial runs of the simulation. For $P_{\text{fail}} \to 0$, the simulation is equivalent to the earlier simulations in Chapter 4. On the other hand, for $P_{\text{fail}} \to 1$ we approximate a case where all communication is delayed by one time step. As a benchmark, the case of an agent not having the control input of its neighbor available and consequently not using it in its edge filter prediction, is considered, i.e., $\hat{\mathbf{u}}_k^j = \mathbf{0}_D$.

The mean estimation errors for the Scenario 1 simulations are shown in Figure 6.2.

Estimation error convergence is generally slower as the probability of failure increases, although for any probability of failure the steady state estimation error is near equal. For the benchmark case ($\hat{\mathbf{u}}_k^j = \mathbf{0}_D$) however, both the convergence speed and steady state error suffer, i.e., the estimation error is inconsistent due to the biased predictions.

For the communication-less scenario, the Wiener predictor with past noisy measurements was applied, the results of which are shown in Figure 6.3. It is clear from the figure that the proposed solution initially experiences difficulties. The main reason for this is that the uncertainty for the control estimates is initially high. This means that the predictions similarly have a larger uncertainty. However, over time the estimated past control inputs contribute to the faster convergence of the proposed solution compared to the benchmark case ($\hat{\mathbf{u}}_k^j = \mathbf{0}_D$). Steady state estimation errors do not converge to the clairvoyant (with communication) estimation error floor. Hence, it can

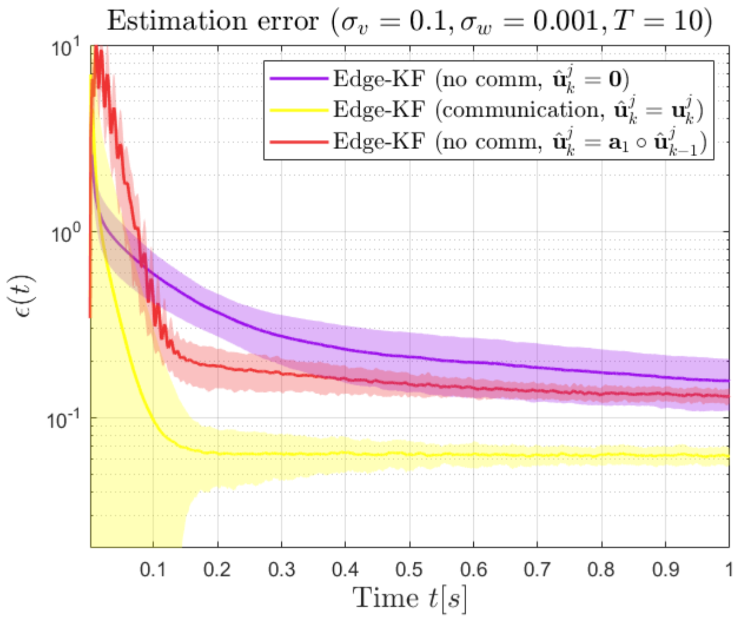


Figure 6.3: Estimation errors for communication-less scenario with observation and process noise. The plot shows the mean estimation error for 50 runs of Algorithm 9, and the ±1 standard deviation regions.

be concluded that there might still be some bias in the predictor leading to inconsistent estimates. The source for this bias might lie in the imperfect assumptions, e.g., the AR control input evolution model which may not be the best suited model for this problem. More discussion on this can be found in the next and final chapter.

7 Conclusion and Future Work

7.1 Summary

In this thesis an attempt has been made to apply distributed filtering techniques to relative formation control systems, motivated by a desire to make these systems more robust to uncertainty introduced in real-life applications. The main contributions of the thesis can be summarized as follows:

- A statistical data model has been proposed with the purpose of modelling sources of uncertainties in relative formation control systems. The model is motivated using real-world formation control examples. A simulation environment has been set up to test the effect of these noise sources on the formation control process.
- A local edge-based Kalman filter has been introduced, which is the optimal filter for individual edge tracking. Since relative positions are tracked, this filter requires no absolute anchor points or shared reference frame. Simulations have shown the superior performance of this filter compared to e.g., the MLE.
- As an extension of the local edge-based Kalman filter, a joint filter is proposed via a global filter design approach. It is shown that the joint filter is superior to the single-edge filter when the edge state space models are coupled when environmental disturbances are spatially correlated.
- Finally, a Wiener predictor has been proposed for systems where inter-agent communication links are (partially) unavailable. Simulations have been performed to show the robustness of the filter to failing communication links.

Part of the contributions of this thesis have been included in a paper submitted to IEEE Control Systems Letters (L-CSS), where it is currently in review.

7.2 Discussion and Future Work

Contrasting the contributions of this thesis, naturally there are also some loose ends and open discussion points, which are reflected upon in this section.

Firstly, let us consider the scope of the contributions of this thesis. The formation control method considered in [25] is limited to agents governed by single-integrator dynamics. Most real-life applications have dynamics which, while often decently approached via the single-integrator model, are more complicated. Dynamics are often nonholonomic or nonlinear. Similarly, the input to be controlled is often an acceleration rather than a velocity, making double-integrator dynamical models better suited. Much work has been done to extend relative formation control schemes to more general dynamical models [27], [44]. It should be noted that the distributed Kalman filters proposed in this thesis are by themselves in no ways limited to single-integrator dynamics, and are easily extendable to any linear state-space model. The requirement of observations being a linear function of the relative state of the two agents still remains, e.g., observations could include measurements of the relative velocity of agents.

The data model for environmental disturbances in relative formation control systems is chosen to reflect real-life disturbances such as wind, radiation pressure or oceanic currents. As indicated in Section 4.1, such disturbances are generally correlated not only spatially but also temporally. In this thesis, the temporal correlation of process noise has not been considered in the design of the proposed filters. An example of spatiotemporal models of statistical disturbances can be seen e.g., in [51]. It should be noted that the Kalman filter is optimal for statistically independent noise processes only. In situations with temporally correlated noise, other filters may be superior. Specifically, one may look at *kriged* Kalman filters [54], which are rooted in Gaussian process regression.

Another important point to consider for the spatial disturbance model is that in this thesis, the process noise covariance matrix $\mathbf{Q}$ indicating covariances of process noise between agents, is *known* and *constant*. As indicated in Section 4.2, in formation control systems we deal with mobile agents. Therefore, if we assume the environmental disturbances acting on agent i and j to be spatially correlated via some kernel function $f(\|\mathbf{z}_i - \mathbf{z}_j\|)$, then the covariance matrix $\mathbf{Q}_k$ is really *time-varying* (hence the subscript k) and *unknown*. Fortunately, agents can locally estimate the locally relevant covariances using their edge state estimates and the known kernel function. This extension of the data model has not been included in the simulations and the effect on the performance of the filters would be an interesting direction to explore.

The joint edge-based filter has been proposed in Chapter 5 as the optimal distributed filter for the given data model, and generally an improvement over the local edge-based Kalman filter proposed in Chapter 4. However, it should be noted that improvement is a direct result of the coupling of the dynamics of edges via the correlation of environmental disturbances acting on the agents. The joint filter utilizes the prior knowledge of the distribution of these disturbances to improve its estimate. If the correlation of external disturbances between agents is very weak or not present, then the joint filter shows no estimation improvement and, through the larger dimension of the joint tracked state, only adds a larger computational complexity for agents.

Lastly, consider the model for control input evolution in Chapter 6. An autoregressive model has been chosen in this thesis. However, looking at the true control input processes, a more extensive model may yield better results. Particularly, the process model should include some mean that is changing over time. The autoregressive model assumes a constant mean of zero. However, an autoregressive integrated moving average (ARIMA) model to include the non-stationarity of the mean of the process. See e.g., [55] for an example of forecasting for ARIMA models.

Bibliography

[1] J. E. Baldwin, "Ground-based interferometry: the past decade and the one to come," in *Interferometry for Optical Astronomy II*, W. A. Traub, Ed., vol. 4838, SPIE, Feb. 2003, p. 1. DOI: 10.1117/12.457192.

[2] H. Linz, D. Bhatia, L. Buinhas, M. Lezius, E. Ferrer, R. Förstner, K. Frankl, M. Philips-Blum, M. Steen, U. Bestmann, W. Hänsel, R. Holzwarth, O. Krause, and T. Pany, "InfraRed Astronomy Satellite Swarm Interferometry (IRASSI): Overview and study results," *Advances in Space Research*, vol. 65, no. 2, pp. 831–849, Jan. 2020, ISSN: 18791948. DOI: 10.1016/j.asr.2019.06.022.

[3] S. Mok, J. Guo, E. Gill, and R. Rajan, "Autonomous Mission Planning for OLFAR: A Satellite Swarm in Lunar Orbit for Radio Astronomy," *71st International Astronautical Congress (IAC)*, 2020.

[4] J. J. Bik, P. N. Visser, and O. Jennrich, "LISA satellite formation control," *Advances in Space Research*, vol. 40, no. 1, pp. 25–34, Jan. 2007, ISSN: 02731177. DOI: 10.1016/j.asr.2007.05.025.

[5] C. S. Cockell, A. Léger, M. Fridlund, T. M. Herbst, L. Kaltenegger, O. Absil, C. Beichman, W. Benz, M. Blanc, A. Brack, A. Chelli, L. Colangeli, H. Cottin, F. Coudé Du Foresto, W. C. Danchi, D. Defrère, J. W. Den Herder, C. Eiroa, J. Greaves, T. Henning, K. J. Johnston, H. Jones, L. Labadie, H. Lammer, R. Launhardt, P. Lawson, O. P. Lay, J. M. LeDuigou, R. Liseau, F. Malbet, S. R. Martin, D. Mawet, D. Mourard, C. Moutou, L. M. Mugnier, M. Ollivier, F. Paresce, A. Quirrenbach, Y. D. Rabbia, J. A. Raven, H. J. Rottgering, D. Rouan, N. C. Santos, F. Selsis, E. Serabyn, H. Shibai, M. Tamura, E. Thiébaut, F. Westall, and G. J. White, "Darwin - A mission to detect and search for life on extrasolar planets," *Astrobiology*, vol. 9, no. 1, pp. 1–22, Mar. 2009, ISSN: 15311074. DOI: 10.1089/ast.2007.0227.

[6] C. P. Escoubet and R. Schmidt, "Cluster II: Plasma measurements in three dimensions," *Advances in Space Research*, vol. 25, no. 7-8, pp. 1305–1314, Jan. 2000, ISSN: 02731177. DOI: 10.1016/S0273-1177(99)00639-0.

[7] S. A. Fuselier, W. S. Lewis, C. Schiff, R. Ergun, J. L. Burch, S. M. Petrinec, and K. J. Trattner, *Magnetospheric Multiscale Science Mission Profile and Operations*, Mar. 2016. DOI: 10.1007/s11214-014-0087-x.

[8] L. Plice, A. D. Perez, and S. West, "Helioswarm: Swarm Mission Design in High Altitude Orbit for Heliophysics," in *Astrodynamics Specialist Conference*, Aug. 2019.

[9] E. Canuto, L. Massotti, A. Molano-Jimenez, and C. Perez, "Long-distance, drag-free, low-thrust, low-Earth-orbit formation control," in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 18, IFAC Secretariat, Jan. 2010, pp. 297–302, ISBN: 9783902661968. DOI: 10.3182/20100906-5-jp-2022.00051.

[10] Y. Xu, Z. Lin, and S. Zhao, "Distributed Affine Formation Tracking Control of Multiple Fixed-Wing UAVs," in *Chinese Control Conference, CCC*, vol. 2020-July, IEEE Computer Society, Jul. 2020, pp. 4712–4717, ISBN: 9789881563903. DOI: 10.23919/CCC50068.2020.9188925.

[11] F. Liao, R. Teo, J. L. Wang, X. Dong, F. Lin, and K. Peng, "Distributed Formation and Reconfiguration Control of VTOL UAVs," *IEEE Transactions on Control Systems*

Technology, vol. 25, no. 1, pp. 270–277, Jan. 2017, ISSN: 10636536. DOI: 10.1109/TCST.2016.2547952.

[12] X. Peng, K. Guo, and Z. Geng, “Full State Tracking and Formation Control for Under-Actuated VTOL UAVs,” *IEEE Access*, vol. 7, pp. 3755–3766, 2019, ISSN: 21693536. DOI: 10.1109/ACCESS.2018.2889370.

[13] Y. Zou, Z. Zhou, X. Dong, and Z. Meng, “Distributed Formation Control for Multiple Vertical Takeoff and Landing UAVs with Switching Topologies,” *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 4, pp. 1750–1761, Aug. 2018, ISSN: 10834435. DOI: 10.1109/TMECH.2018.2844306.

[14] I. Sarras and H. Siguerdidjane, “On the guidance of a UAV under unknown wind disturbances,” *2014 IEEE Conference on Control Applications, CCA 2014*, Jan. 2014. DOI: 10.1109/CCA.2014.6981442.

[15] A. J. Healey, “Application of formation control for multi-vehicle robotic minesweeping,” in *Proceedings of the IEEE Conference on Decision and Control*, vol. 2, 2001, pp. 1497–1502. DOI: 10.1109/cdc.2001.981106.

[16] T. Curtin, J. Bellingham, J. Catipovic, and D. Webb, “Autonomous Oceanographic Sampling Networks,” *Oceanography*, vol. 6, no. 3, pp. 86–94, 1993, ISSN: 10428275. DOI: 10.5670/oceanog.1993.03.

[17] A. Laux, L. Mullen, P. Perez, and E. Zege, “Underwater laser range finder,” in *Ocean Sensing and Monitoring IV*, vol. 8372, SPIE, Jun. 2012, 83721B, ISBN: 9780819490506. DOI: 10.1117/12.919280.

[18] T. J. Tarn, G. A. Shoults, and S. P. Yang, “A dynamic model of an underwater vehicle with a robotic manipulator using Kane’s method,” *Autonomous Robots*, vol. 3, no. 2-3, pp. 269–283, 1996, ISSN: 09295593. DOI: 10.1007/BF00141159.

[19] K. Ehlers, B. Meyer, and E. Maehle, “Full Holonomic Control of the Omni-directional AUV SMART-E - VDE Conference Publication,” in *ISR/Robotik 2014; 41st International Symposium on Robotics*, VDE, Jun. 2014, ISBN: 978-3-8007-3601-0.

[20] K. K. Oh, M. C. Park, and H. S. Ahn, “A survey of multi-agent formation control,” *Automatica*, vol. 53, pp. 424–440, Mar. 2015, ISSN: 00051098. DOI: 10.1016/j.automatica.2014.10.022.

[21] R. Olfati-Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004, ISSN: 00189286. DOI: 10.1109/TAC.2004.834113.

[22] W. Ren, R. W. Beard, and T. W. McLain, “Coordination Variables and Consensus Building in Multiple Vehicle Systems,” *Lecture Notes in Control and Information Sciences*, vol. 309, pp. 171–188, 2005, ISSN: 01708643. DOI: 10.1007/978-3-540-31595-7{_}10.

[23] Z. Lin, L. Wang, Z. Han, and M. Fu, “Distributed formation control of multi-agent systems using complex laplacian,” *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1765–1777, 2014, ISSN: 00189286. DOI: 10.1109/TAC.2014.2309031.

[24] Z. M. Han, Z. Y. Lin, M. Y. Fu, and Z. Y. Chen, *Distributed coordination in multi-agent systems: a graph Laplacian perspective*, Jun. 2015. DOI: 10.1631/FITEE.1500118.

[25] Z. Lin, L. Wang, Z. Chen, M. Fu, and Z. Han, “Necessary and sufficient graphical conditions for affine formation control,” *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 2877–2891, 2016, ISSN: 00189286. DOI: 10.1109/TAC.2015.2504265.

[26] L. Wang, Z. Han, Z. Lin, and M. Fu, “A linear approach to formation control under directed and switching topologies,” in *Proceedings - IEEE International Conference on*

Robotics and Automation, Institute of Electrical and Electronics Engineers Inc., Sep. 2014, pp. 3595–3600. DOI: 10.1109/ICRA.2014.6907378.

[27] T. Han, Z. Lin, W. Xu, and M. Fu, "Three-dimensional formation merging control of second-order agents under directed and switching topologies," in *IEEE International Conference on Control and Automation, ICCA*, IEEE Computer Society, 2014, pp. 225–230, ISBN: 9781479928378. DOI: 10.1109/ICCA.2014.6870924.

[28] Y. Xu, D. Luo, D. Li, Y. You, and H. Duan, "Affine formation control for heterogeneous multi-agent systems with directed interaction networks," *Neurocomputing*, vol. 330, pp. 104–115, Feb. 2019, ISSN: 18728286. DOI: 10.1016/j.neucom.2018.11.023.

[29] Z. Lin, M. Broucke, and B. Francis, "Local control strategies for groups of mobile autonomous agents," *IEEE Transactions on Automatic Control*, vol. 49, no. 4, pp. 622–629, Apr. 2004, ISSN: 00189286. DOI: 10.1109/TAC.2004.825639.

[30] K. K. Oh and H. S. Ahn, "Formation control and network localization via orientation alignment," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 540–545, 2014, ISSN: 00189286. DOI: 10.1109/TAC.2013.2272972.

[31] D. V. Dimarogonas and K. H. Johansson, "On the stability of distance-based formation control," in *Proceedings of the IEEE Conference on Decision and Control*, 2008, pp. 1200–1205, ISBN: 9781424431243. DOI: 10.1109/CDC.2008.4739215.

[32] M. C. Park, Z. Sun, B. D. Anderson, and H. S. Ahn, "Distance-based control of Kn formations in general space with almost global convergence," *IEEE Transactions on Automatic Control*, vol. 63, no. 8, pp. 2678–2685, Aug. 2018, ISSN: 00189286. DOI: 10.1109/TAC.2017.2776524.

[33] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering, Transactions of the ASME*, vol. 82, no. 1, pp. 35–45, Mar. 1960, ISSN: 1528901X. DOI: 10.1115/1.3662552.

[34] A. Simonetto and G. Leus, "Distributed maximum likelihood sensor network localization," *IEEE Transactions on Signal Processing*, vol. 62, no. 6, pp. 1424–1437, Mar. 2014, ISSN: 1053587X. DOI: 10.1109/TSP.2014.2302746.

[35] U. A. Khan, S. Kar, and J. M. F. Moura, "Distributed Sensor Localization in Random Environments using Minimal Number of Anchor Nodes," *IEEE Transactions on Signal Processing*, vol. 57, no. 5, pp. 2000–2016, Feb. 2008. DOI: 10.1109/TSP.2009.2014812.

[36] D. Marelli, M. Zamani, M. Fu, and B. Ninness, "Distributed Kalman filter in a network of linear systems," *Systems and Control Letters*, vol. 116, pp. 71–77, Jun. 2018, ISSN: 01676911. DOI: 10.1016/j.sysconle.2018.04.005.

[37] D. Viegas, P. Batista, P. Oliveira, and C. Silvestre, "Discrete-time distributed Kalman filter design for formations of autonomous vehicles," *Control Engineering Practice*, vol. 75, pp. 55–68, Jun. 2018, ISSN: 09670661. DOI: 10.1016/j.conengprac.2018.03.014.

[38] S. Lu, C. Lin, Z. Lin, R. Zheng, and G. Yan, "Distributed Kalman filter for relative sensing networks," in *Chinese Control Conference, CCC*, vol. 2015-September, IEEE Computer Society, Sep. 2015, pp. 7541–7546, ISBN: 9789881563897. DOI: 10.1109/ChiCC.2015.7260835.

[39] A. Y. Alfakih, "On bar frameworks, stress matrices and semidefinite programming," *Mathematical Programming*, vol. 129, no. 1, pp. 113–128, Sep. 2011, ISSN: 00255610. DOI: 10.1007/s10107-010-0389-z.

[40] S. J. Gortler, A. D. Healy, and D. P. Thurston, "Characterizing Generic Global Rigidity," *Ad Hoc Networks*, vol. 10, no. 3, pp. 623–634, Oct. 2007. DOI: 10.1016/j.adhoc.2011.06.016.

[41] R. Connelly, "Generic global rigidity," *Discrete and Computational Geometry*, vol. 33, no. 4, pp. 549–563, Oct. 2005, ISSN: 01795376. DOI: 10.1007/s00454-004-1124-4.

[42] T. Eren, D. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. Anderson, and P. N. Belhumeur, "Rigidity, computation, and randomization in network localization," in *Proceedings - IEEE INFOCOM*, vol. 4, 2004, pp. 2673–2684, ISBN: 0780383559. DOI: 10.1109/INFCOM.2004.1354686.

[43] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, Mar. 1966, ISSN: 00333123. DOI: 10.1007/BF02289451.

[44] Y. Xu, D. Luo, Y. You, and H. Duan, "Distributed adaptive affine formation control for heterogeneous linear networked systems," *IEEE Access*, vol. 7, pp. 23 354–23 364, 2019, ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2898974.

[45] C. Fischer, K. Muthukrishnan, and M. Hazas, *SLAM for Pedestrians and Ultrasonic Landmarks in Emergency Response Scenarios*. Elsevier, Jan. 2011, vol. 81, pp. 103–160. DOI: 10.1016/B978-0-12-385514-5.00003-3.

[46] C. Soares, F. Valdeira, and J. Gomes, "Range and Bearing Data Fusion for Precise Convex Network Localization," *IEEE Signal Processing Letters*, vol. 27, pp. 670–674, 2020, ISSN: 15582361. DOI: 10.1109/LSP.2020.2988178.

[47] M. C. Deans and M. Hebert, "Bearings-Only Localization and Mapping," Ph.D. dissertation, USA, 2005, ISBN: 0542292831.

[48] M. Coppola, *Relative Localization for Collision Avoidance in Micro Air Vehicle teams Using on-board processing and sensors in indoor environments (MSc thesis)*. Delft, Jun. 2016.

[49] Q. Zhu, J. Chen, D. Shi, L. Zhu, X. Bai, X. Duan, and Y. Liu, "Learning Temporal and Spatial Correlations Jointly: A Unified Framework for Wind Speed Prediction," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 1, pp. 509–523, Jan. 2020, ISSN: 19493037. DOI: 10.1109/TSTE.2019.2897136.

[50] A. K. Sinha and M. M. Zaheer, "Modelling of force and torque due to solar radiation pressure acting on interplanetary spacecraft," in *Advances in Intelligent Systems and Computing*, vol. 714, Springer Verlag, 2019, pp. 525–535. DOI: 10.1007/978-981-13-0224-4{_}47.

[51] V. Roy, A. Simonetto, and G. Leus, "Spatio-temporal field estimation using kriged kalman filter (KKF) with sparsity-enforcing sensor placement," *Sensors (Switzerland)*, vol. 18, no. 6, Jun. 2018, ISSN: 14248220. DOI: 10.3390/s18061778.

[52] J. Mercer, "XVI. Functions of positive and negative type, and their connection the theory of integral equations," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 209, no. 441-458, pp. 415–446, Jan. 1909, ISSN: 0264-3952. DOI: 10.1098/rsta.1909.0016.

[53] S. M. Kay, *Fundamentals of statistical signal processing*. Englewood Cliffs, N.J.: Prentice Hall PTR, 1993, ISBN: 9780133457117.

[54] K. V. Mardia, C. Goodall, E. J. Redfern, and F. J. Alonso, "The Kriged Kalman Filter," *Test*, vol. 7, no. 2, pp. 217–282, 1998, ISSN: 11330686. DOI: 10.1007/bf02565111.

[55] C. Li and J. W. Hu, "A new ARIMA-based neuro-fuzzy approach and swarm intelligence for time series forecasting," *Engineering Applications of Artificial Intelligence*, vol. 25,

no. 2, pp. 295–308, Mar. 2012, ISSN: 09521976. DOI: 10.1016/j.engappai.2011.10.005.