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

CAS-2022-5262623

M.Sc. Thesis

Robust Formation Control against Observation Losses

Zhonggang Li

Abstract

Distributed formation control has received increasing attention in multiagent systems. Maintaining certain geometry in space is advantageous in many applications such as space interferometry and underwater sensing. At present, there is a variety of distributed solutions for agents to converge to desired formations and track a series of prescribed maneuvers. They typically rely on the relative kinematics e.g., relative positions of the neighboring agents as state observations for the local controller. In harsh working environments, the acquisition of the relative kinematics is challenged and observation losses might occur, which can be detrimental to the optimality of formation.

In this work, observation losses in noisy environments are addressed under a distributed formation control framework. Three types of solutions are proposed to enhance the robustness which is evaluated through the improvements of tracking error, convergence speed, and smoothness of trajectories in both random and permanent loss settings. Firstly, a relative localization technique is proposed using formation itself as a spatial constraint. Secondly, a dynamic model is established for the agents entailed by a Kalman filter-based solution. Finally, a fusion of the previous two types is inspired and it exhibits superior performance than both aforementioned types individually.

This work not only provides means of relative localization without additional sensor data but also shares insights into coping with random or permanent graph changes for stress-based formation control systems. This could potentially lead to the exploration of formation control with subgraphs or energy-efficient sensing as future directions.

Faculty of Electrical Engineering, Mathematics and Computer Science

Delft University of Technology

Robust Formation Control against Observation Losses

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

Zhonggang Li
born in Baotou, China

This work was performed in:

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

Delft University of Technology

Copyright © 2022 Circuits and Systems Group
All rights reserved.

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 **"Robust Formation Control against Observation Losses"** by **Zhonggang Li** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 25 August 2022

Chairman: ______________________________
dr. R.T.Rajan

Advisor: ______________________________
dr. R.T.Rajan

Committee Members: ______________________________
prof.dr.ir. G.J.T.Leus

____
dr. J.J.D.Loicq

Abstract

Distributed formation control has received increasing attention in multiagent systems. Maintaining certain geometry in space is advantageous in many applications such as space interferometry and underwater sensing. At present, there is a variety of distributed solutions for agents to converge to desired formations and track a series of prescribed maneuvers. They typically rely on the relative kinematics e.g., relative positions of the neighboring agents as state observations for the local controller. In harsh working environments, the acquisition of the relative kinematics is challenged and observation losses might occur, which can be detrimental to the optimality of formation.

In this work, observation losses in noisy environments are addressed under a distributed formation control framework. Three types of solutions are proposed to enhance the robustness which is evaluated through the improvements of tracking error, convergence speed, and smoothness of trajectories in both random and permanent loss settings. Firstly, a relative localization technique is proposed using formation itself as a spatial constraint. Secondly, a dynamic model is established for the agents entailed by a Kalman filter-based solution. Finally, a fusion of the previous two types is inspired and it exhibits superior performance than both aforementioned types individually.

This work not only provides means of relative localization without additional sensor data but also shares insights into coping with random or permanent graph changes for stress-based formation control systems. This could potentially lead to the exploration of formation control with subgraphs or energy-efficient sensing as future directions.

Acknowledgments

These two years in my student career have been a challenging time as the world is healing from the pandemic. I would like to thank the following people whose invaluable support helped me through the most difficult times.

I would first like to express my sincerest gratitude to my daily supervisor dr. Raj Thilak Rajan. Your academic guidance has been a valuable support to my academic career. I enjoy the weekly discussions and seminars very much and I look forward to our further collaboration. I would also like to thank you for funding my work at the SITB symposium.

I would next like to thank the Ph.D. candidates and other MSc students in the CAS group from whom I benefit profusely through our academic discussions. I also had a very pleasant time working with the teaching and support staff in EEMCS as a teaching assistant.

I would finally thank my family and my friends for their unconditional support and trust. My life is filled with love and joy because of you and I wish you all the best in your future endeavors.

Zhonggang Li
Delft, The Netherlands
25 August 2022

Contents

List of Figures

List of Tables

Introduction 1

1.1 Distributed Autonomous Systems

Autonomous systems are receiving increasing attention with emerging technologies such as self-driving vehicles, space rovers, self-piloting drones, etc. An autonomous system is usually one that can accomplish a set of tasks prescribed by the human in complex and changing environments, but without human inputs and interventions. The typical workflow of such systems includes sensing and interpretation (decision making) followed by actuation, which brings several communities together, e.g., signal processing, control, machine learning, and robotics. Recently, as the need and complexity of tasks rise rapidly, ideas of mobilizing multiple autonomous agents to cooperatively achieve a collective goal have been extensively researched and implemented in the industry. For instance, in energy management [1], sensing networks [2] in geology, reinforcement learning [3], satellite interferometry [4], cooperative localization [5], etc. These systems are often referred to as multiagent systems (MAS) where the autonomous agents can be either physical such as robots or virtual such as computation nodes in distributed optimization [6]. Because of the pursuit of scalability, computation and power efficiency, and robustness against single node failure, multiagent systems have a distributed nature, which also fits the philosophy of distributed autonomous systems (DAS). Among all the applications of MAS or DAS, some rely on a stable geometric pattern in space from the agents, i.e., formation, which naturally leads to the problem of formation control.

1.2 Formation Control

1.2.1 Applications of Formation Control

The application of formation control abounds with different types of agents. Depending on the goal of maintaining formation, we categorize and list a few applications in the following.

Spatial Sensing

Most of the applications of formation control are in line with sensing an unknown field by placing mobile nodes in space. Unmanned aerial vehicles (UAVs) are typically the choice as mobile platforms for their flexibility and coverage. For example, drone swarms in formation are used for 3D scene reconstruction [7] and aerial filming [8]. In underwater applications [9], autonomous underwater vehicles (AUVs) are often employed to take hydrographical measurements in the ocean in formation. There is also a need for formation control in space applications [10]. For instance, space-based interferometry is used to detect and unveil the origins of the cosmos and is favored over ground-based

setup due to its immunity to atmospheric disturbance and flexible reconfiguration [11].
Object Transport
Another major type of formation control application is cooperatively transporting objects. For instance, the possibility of drone delivery is being discussed and tested nowadays, and the deliveries with drone swarms in formation are also becoming a future potential [12]. Other than drones, multiple robots could also carry and transport large or heavy objects [13, 14], and recently the potential of transportation with autonomous vessels is being explored [15].
Economic Travel
One of the benefits of traveling in formation is fuel efficiency. It has been discovered that flying in the wake vortex field of an aircraft is aerodynamically efficient and thus can reduce the fuel consumption [16]. As such, for a group of aircraft, formation flying in a long-haul traveling is favored. Similar ideas are also realized for land transportation e.g., truck platooning [17].

Other purposes of formation control include showcase and entertainment. For instance, drone swarms can be used as a mobile display and the formation of the drones is important and needs to be reconfigurable [18].

1.2.2 State-of-the-Art Controllers

The majority of research makes efforts to design controllers targeting different types of agents based on different principles. There have been lots of surveys summarizing the merits and challenges of state-of-art controllers [9, 19, 20, 21]. The strategies behind controllers are versatile, e.g.,virtual structure [22, 23], potential field [24], leader-follower [25], etc., and there are often more than one applied in controller designs. But based on the variables that agents sense and control or how the formation is defined, the controller can roughly be categorized as follows [21, 26].

- Position-constrained controller [23], where the formation is defined by the individual positions of agents and capabilities of sensing absolute positions under a global reference frame are required. The inter-agent interaction is not necessary and the controller can track almost any formation.

- Displacement-constrained controller [27], where inter-agent relative positions are actively and locally sensed with the same orientation. This requires some interactions among agents and inter-agent communication might be needed. Since the constraints are on the relative position, it can track formation with translation changes.

- Distance-constrained controller [28], where inter-agent distances are maintained. This allows the controller to track formation with rotation and translation changes without compromising the inter-agent distances. More interactions among agents are needed for this type of controller.

- Bearing-constrained controller [29], where inter-agent bearings are sensed and maintained. Since bearings are invariant to scaling and translation, this type of controller can track formation with different translations and scaling without changing the constraints on the bearings. However, the rigidity of formation can not be guaranteed.

These controllers have been modified in trending research to be capable to track more variations of formations regardless of the sensing variable. For instance, by introducing complex Laplacians, [30] can track scaling with relative position measurements since complex Laplacians are invariant to translation, rotation, and scaling. A more powerful tool that has been brought to the spotlight recently is the stress matrix which can be considered as a generalized graph Laplacian. It is invariant to any affine transformation. As such, stress-based controllers can track formations up to any affine transformation once the stress is settled. For instance, [31] gives the graphical conditions of stabilizable formations using stress matrices and provides means of calculation. It is also able to achieve affine formations with relative position measurements. Moreover, [26] extends the scope of [31] and achieves affine formation maneuver control using a leader-follower strategy. This thesis will be established in this formation maneuver control framework and further address the problem of observation losses.

1.3 Observation Losses in Formation Control

In feedback control systems, the output states need to be observed using sensors. For formation control systems, the states are usually positions, relative positions, distances, or bearings as reviewed in Section 1.2.2. To understand the reason for observation losses, it is necessary to review the principle and typical setups for the acquisitions of the observations. In the case of affine formation control, they are relative positions.

1.3.1 Relative Localization Methods

Global navigation satellite systems (GNSS) such as Beidou and Galileo are usually the go-to solution for positioning, and relative positions could be simply acquired up to a translation. But GNSS is limited in some environments, and equipment for high-precision localization can be expensive. As such, relative positioning is more suitable for formation control of UAV or robot swarms. It mainly involves measuring the distance/bearing between agents and calculating the relative locations.

The algorithms of relative localization are generally based on radio frequency (RF) or optical signal according to [32]. The RF-based positioning algorithms include the following.

- Received signal strength (RSS) [33], where the signal attenuation can be modeled and measured, and then the distance can be estimated. The typical sensors used for this application are Bluetooth, Wi-Fi, and radio frequency identification (RFID).

- Time of arrival (TOA) [34], where the propagation speed of electromagnetic waves is known and the distance can be estimated by measuring the time of flight. Typical sensors used for TOA include ultra-wideband (UWB).

- Angle of arrival (AOA) [35], where the radio signals are received by an array of antennas, and the incidence angle of the signal can be estimated by the small phase differences of the received signal on each antenna. Bluetooth and Wi-Fi are typically used in this case.

- Time difference of arrival (TDOA) [36], where synchronized anchors at known locations (fixed ground points) receive the same signal up to a small time difference, based on which the position can be calculated. The sensors of this application also include UWB.

The optical signal-based positioning algorithm includes simultaneous localization and mapping (SLAM) which often uses LIDAR and cameras, and multi-camera target recognition and localization algorithms with motion capture systems.

As such, according to the elements involved in relative localization, the causes of observation losses can be as follows.

- Sensor-induced, which includes all kinds of sensor malfunctioning or calibration issues. They can be either random or permanent losses depending on the nature of the failure.

- Environment-induced, which includes interference of RF signals, communication delays or packet losses, and intrusion in line-of-sight based localization. These mostly cause random observation losses.

- System-induced, which includes scheduling issues of the sensors or computation overload. These also usually cause random losses.

- Target-induced, in which there are no target objects to localize. The targets can be offline due to e.g., maintenance and critical failure, and these issues usually cause permanent observation losses.

1.3.2 Frameworks against Observation Losses

The problem of observation losses is eventually converted to state estimation. In control theory literature, observation losses are referred to as intermittent observations and the arrival or availability of observation is usually modeled by a binary random variable. A paradigm of linear-quadratic-Gaussian (LQG) control under intermittent observations has been established, and Kalman filter-based state estimation solutions are well-studied [37, 38, 39] and they can optimally give state estimates and make partial observations full. For nonstandard LQG systems, there have also been solutions like interactive multiple models (IMM) [40], Gaussian sum filter [41], and deep learning filters [42] for state estimates.

Recently, with more interest focused on distributed sensor networks, distributed frameworks with intermittent observations are also established. But most literature still relies on the Kalman filter [43, 44] or variants of the Kalman filter [45]. However, the observation losses in formation control receive less attention. Since there are different strategies used in formation control techniques, observations are not limited to (relative) positions and are typically modeled as communication delays and packet losses. But they all cause the missing of key elements in the controller and can generally be considered loss of observation. There have been a few strategies developed for the observation losses for formation control systems. Learning the behavior of the subject of observation is considered in e.g., [46] where iterative learning control (ILC) methods are applied when the same task is repeatedly performed and the previous experience can be of contribution, or [47] where a long short-term memory (LSTM) method is used to predict the lost packets. Controllers with enhanced robustness to packet losses are designed in e.g., [48] where model predictive control (MPC) is used, and [49] where sliding mode control is implemented. However, most of the techniques work for under 30% random losses and the robust system designs for heavier random losses and even permanent losses generally remain unexplored. As such, we identify that the limitations in current research are mainly

- robust system design for affine formation control regime,
- coping mechanism for heavily lossy conditions, and
- solutions to permanent observation losses,

which serve as the objectives for this thesis.

1.4 Overview and Notation

This thesis is organized as follows as shown in Figure 1.1. Chapter 2 introduces the building blocks of distributed affine formation control. Chapter 3 describes the distributed control laws to achieve formation maneuvering and establishes the simulation setup on which the later chapters are built. Chapter 4 formulates the problem of observation losses and gives the first type of state estimator that uses formation itself as a spatial constraint. Chapter 5 extends the results in Chapter 4 with noisy observations and evaluates the performance statistically. Also, to address the noises, a smoothing technique is introduced on top of this estimator. Chapter 6 establishes a framework for the second type of state estimator and further proposes a fusion technique of both types of estimators. Chapter 7 pools all the proposed techniques and evaluates them in both random observation loss and permanent observation loss settings. The final chapter evaluates all proposed methods and reaches some conclusions. The future work of this thesis is also discussed in this chapter.

The notations used in this thesis are as follows.

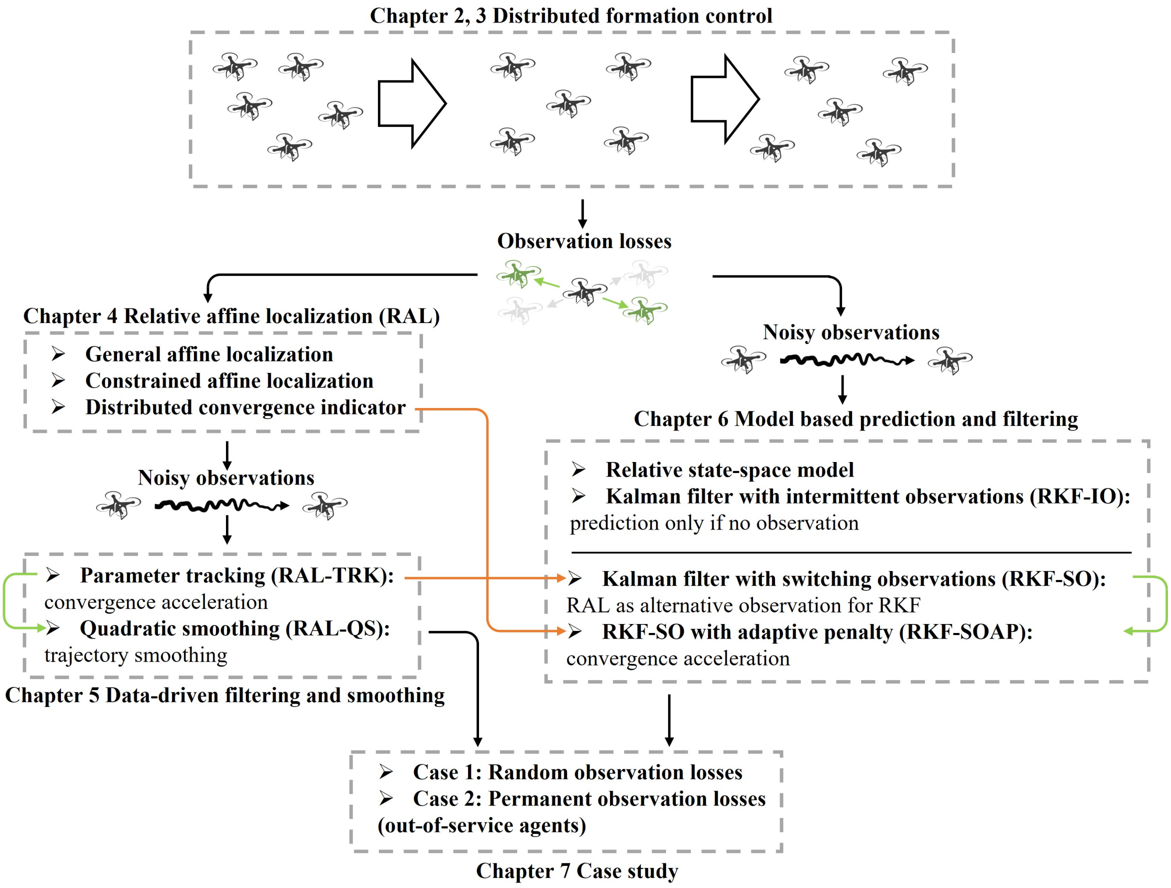

Chapter 2, 3 Distributed formation control
Observation losses
Chapter 4 Relative affine localization (RAL)
General affine localization
Constrained affine localization
Distributed convergence indicator
Noisy observations
Parameter tracking (RAL-TRK):
convergence acceleration
Quadratic smoothing (RAL-QS):
trajectory smoothing
Chapter 5 Data-driven filtering and smoothing
Noisy observations
Chapter 6 Model based prediction and filtering
Relative state-space model
Kalman filter with intermittent observations (RKF-IO):
prediction only if no observation
Kalman filter with switching observations (RKF-SO):
RAL as alternative observation for RKF
RKF-SO with adaptive penalty (RKF-SOAP):
convergence acceleration
Case 1: Random observation losses
Case 2: Permanent observation losses
(out-of-service agents)
Chapter 7 Case study

Figure 1.1: An overview of the structure of this thesis

Table 1.1: Notations used in this thesis

Symbol	Definition
a	Scalar
$\mathbf{a}$	Column vector
$\mathbf{A}$	Matrix
$\mathcal{A}$	Set
A_{ij}	Entries of the i-th row and j-th column of matrix $\mathbf{A}$
$\mathbf{1}_N$	All-one vector of length N
$\mathbf{0}_N$	All-zero vector of length N
$\mathbf{I}_N$	Identity matrix of size N
$\mathcal{N}_i$	Set of neighboring nodes of node i
$\boldsymbol{\mathcal{N}}(0,1)$	Normal distribution with mean 0 and variance 1
$\mathbb{R}^N$	Set of real vectors of length N
$\mathbb{R}^{N\times M}$	Set of real matrices of size N-by-M
$\otimes$	Kronecker product
$\lvert\cdot\rvert$	Cardinality of a set
$\hat{\cdot}$	estimate
$\lVert\cdot\rVert_2$	Euclidean norm
$\lVert\cdot\rVert_F$	Frobenius norm
$\mathbb{E}[\cdot]$	Expectation operator
$\mathrm{cov}[\cdot]$	Covariance operator
$\mathrm{P}[\cdot]$	Probability of a random variable
$\mathrm{tr}(\cdot)$	Trace of a matrix
$\mathrm{rank}(\cdot)$	Rank of a matrix
$\bar{A}(\cdot)$	Matrix augmentation operator
$\mathrm{diag}(\mathbf{a})$	Diagonal matrix with diagonal entries the elements of vector $\mathbf{a}$
$\mathrm{bdiag}(\mathbf{A}_i)_{i\in\mathcal{S}}$	block diagonal matrix with blocks $\mathbf{A}_i$ for all i in the set $\mathcal{S}$

Fundamentals of Affine Formation Control 2

In this chapter, the most essential building blocks of affine formation control are established. The notations on the graph theory are introduced first followed by the concepts of configuration and formation. Then some affine transformation-related terminology is defined with some visual explanations. After that, the concept of stress and its critical role in affine formation control is explained. The effect of the leader-follower strategy and its implications are emphasized due to its importance in dealing with observation losses in later chapters. Finally, we define the metrics with which the main performance of the methods developed throughout the thesis is evaluated.

2.1 Graph Theory

Consider the setup where N mobile agents are deployed in $\mathbb{R}^D$ where $N \geq D+1$. The agents and their pairwise connections are typically represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the vertex (node) set $\mathcal{V} = \{1, ..., N\}$ denotes the identities of agents and the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the information flow between agents. If edge $(i, j) \in \mathcal{E}$, the information can flow from agent j to agent i. In this thesis, we assume an undirected graph, $(i, j) \in \mathcal{E} \Leftrightarrow (j, i) \in \mathcal{E}$, meaning that the edges are bidirectional. The set of neighbors of node i is defined as $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$, to which the cardinality $|\mathcal{N}_i|$ denotes the number of neighbors of node i. The total number of undirected edges is denoted by M

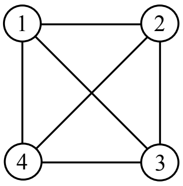
There is a variety of representations that can characterize a graph, e.g., adjacency matrix, incidence matrix, degree matrix, etc. Here we define the incidence matrix $\mathbf{B} \in \mathbb{R}^{N \times M}$ as follows,

$$B_{ij} = \begin{cases} 1, & \text{if } (i,j) \in \mathcal{E} \text{ and } i < j \\ -1, & \text{if } (i,j) \in \mathcal{E} \text{ and } i > j \\ 0, & \text{others} \end{cases}, \tag{2.1}$$

which will be used in the calculation of the stress matrix. An example of a graph and its incidence matrix is shown in Figure 2.1.

2.2 Configuration and Formation

We use $\mathbf{z}_i \in \mathbb{R}^D$ to denote the position of agent i and the *configuration* of all agents is define as $\mathbf{z} = [\mathbf{z}_1^T, ..., \mathbf{z}_N^T]^T \in \mathbb{R}^{DN}$. Similarly, we define $\mathbf{z}^* = [\mathbf{z}_1^{*T}, ..., \mathbf{z}_N^{*T}]^T \in \mathbb{R}^{DN}$ as the *target configuration* with $\mathbf{z}_i^* \in \mathbb{R}^D$ being the target position of agent i. Additionally, we define a generic configuration $\mathbf{p} = [\mathbf{p}_1^T, ..., \mathbf{p}_N^T]^T \in \mathbb{R}^{DN}$ called *nominal configuration* [26], which presents a general geometric pattern that agents are expected



(a)

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{bmatrix}$$

(b)

Figure 2.1: An example of a graph and its incidence matrix. By convention, the columns in the incidence matrix are grouped per agent in ascending order.

to maintain. The target configuration will be designed as a time-varying mapping of the nominal configuration to denote formation maneuvers. A *formation* is formally defined as the combination of the configuration of agents and the associated graph $(\mathcal{G}, \mathbf{z})$. Correspondingly, target formation and nominal formation are defined as $(\mathcal{G}, \mathbf{z}^*)$ and $(\mathcal{G}, \mathbf{p})$, respectively.

2.3 Affine Image and Affine Transformation

It is mentioned in the previous section that the target and nominal configuration are up to a time-varying mapping which, in this thesis, is considered to be an affine transformation. The set that contains all affine transformations of the nominal configuration is called an *affine image*.

Definition 2.1 (*Affine image* [31])**.** The affine image of the nominal configuration is defined as

$$\mathcal{A}(\mathbf{p}) = \{\mathbf{z} \in \mathbb{R}^{DN} : \mathbf{z} = (\mathbf{I}_N \otimes \mathbf{\Theta})\mathbf{p} + \mathbf{1}_N \otimes \mathbf{t}\}, \tag{2.2}$$

where $\mathbf{\Theta} \in \mathbb{R}^{D\times D}$ is a transformation matrix and $\mathbf{t} \in \mathbb{R}^D$ is a translation vector.

This leads to the formal definition of target configuration.

Definition 2.2 (*Target configuration*)**.** The time-varying target configuration has the form of

$$\mathbf{z}^*(k) = (\mathbf{I}_N \otimes \mathbf{\Theta}(k))\mathbf{p} + \mathbf{1}_N \otimes \mathbf{t}(k), \tag{2.3}$$

where $\mathbf{\Theta}(k) \in \mathbb{R}^{D\times D}$ and $\mathbf{t}(k) \in \mathbb{R}^D$ are in discrete time k. Sometimes we acquiesce that the nominal configuration is the initial target configuration $\mathbf{p} = \mathbf{z}^*(0)$. Note that the target configuration $\mathbf{z}^*(k)$ is in affine image $\mathcal{A}(\mathbf{p})$ for all k by definition.

As such, the desired position for any agent i is $\mathbf{z}_i^*(k) = \mathbf{\Theta}(k)\mathbf{p}_i + \mathbf{t}(k)$. This can also be written in an equivalent but a more compact form

$$\mathbf{z}_i^*(k) = [\mathbf{\Theta}(k) \mid \mathbf{t}(k)] \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix} = \bar{\mathbf{\Theta}}(k)\bar{\mathbf{p}}_i, \tag{2.4}$$

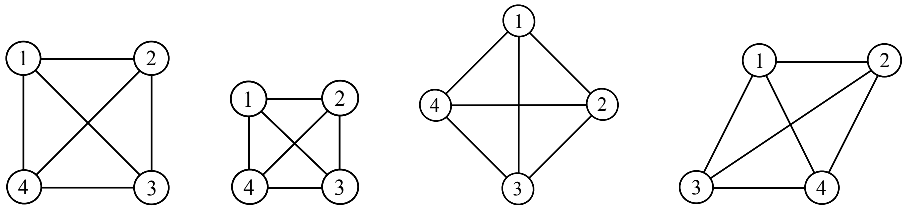
where the translation vector $\mathbf{t}(k)$ is appended on the matrix $\mathbf{\Theta}(k)$ and $\bar{\mathbf{p}}_i$ is an augmented vector. In practice, the time-varying formation can be designed by utilizing

Table 2.1: Several special cases of affine transformation

Transformation	$\bar{\mathbf{\Theta}}$	DOF in $\mathbb{R}^2$ ($\mathbb{R}^3$)	orientation	distances	angles	parallelism
translation	$[\mathbf{I} \mid \mathbf{t}]$	2 (3)	✓	✓	✓	✓
Euclidean (rigid)	$[\mathbf{R} \mid \mathbf{t}]$	3 (6)		✓	✓	✓
similarity	$[s\mathbf{R} \mid \mathbf{t}]$	4 (7)			✓	✓
affine	$[\mathbf{\Theta} \mid \mathbf{t}]$	6 (12)				✓

time-dependent functions for each parameter in $\bar{\mathbf{\Theta}}$, through which the target velocities and accelerations are also defined.

Geometrically, the affine transformation includes several basic geometric transformations such as scaling, rotation, shearing, translation, etc., and combinations of them. A graphical illustration of these geometrical transformations can be found in Figure 2.2. There are also several common combinations of these basic transformations as special cases of general affine transformation that are often adopted in practice for their preserved properties [50]. Table 2.1 shows the structure of $\bar{\mathbf{\Theta}}$, the degrees of freedom (DOF), and the preserved property of these cases, in which $\mathbf{I}$ is an identity matrix, $\mathbf{R}$ is a rotation matrix and s is a scalar.



(a) original (b) scaling (c) rotation (d) shearing

Figure 2.2: Illustration of some basic geometric transformations that constitute affine transformations.

2.4 Edge Weights and Stress Matrix

For a nominal formation $(\mathcal{G}, \mathbf{p})$, there exists a set of scalars $\{l_{ij}\}_{(i,j)\in\mathcal{E}}$, called *equilibrium stress*, that satisfy

$$\sum_{j\in\mathcal{N}_i} l_{ij}(\mathbf{p}_j - \mathbf{p}_i) = \mathbf{0}, \quad i \in \mathcal{V}. \tag{2.5}$$

From a mechanical perspective, the equilibrium stress balances the forces applied to each node by neighboring nodes and thus produces a stable formation [26]. (2.5) also suggests that the equilibrium stress can only be determined up to scalar because if l_{ij} satisfy (2.5), there exists a scalar ω such that ωl_{ij} also satisfy (2.5). Now we define a *stress matrix* that structurally stores these coefficients.

Definition 2.3 (*Stress matrix*)**.** Under undirected graphs, a stress matrix $\mathbf{L} \in \mathbb{S}^{N\times N}$

satisfies

$$L_{ij} = \begin{cases} 0, & \text{if } i \neq j \text{ and } (i,j) \notin \mathcal{E} \\ -l_{ij}, & \text{if } i \neq j \text{ and } (i,j) \in \mathcal{E} \ , \\ \sum_{j \in \mathcal{N}_i} l_{ij}, & \text{if } i = j \end{cases} \tag{2.6}$$

which has a similar structure to the graph Laplacian matrix and is sometimes referred to as the generalized Laplacian matrix [51] in some literature since the values can also be negative here.

There are several approaches to calculating the stress matrix and two common methods are introduced in Appendix A. The property of the stress matrix is closely related to the rigidity of formation. It has been stated in [26] that under undirected graph $\mathcal{G}$, formation $(\mathcal{G}, \mathbf{p})$ is universally rigid if and only if there exists a stress matrix $\mathbf{L} \in \mathbb{S}_+^N$ such that $\text{rank}(\mathbf{L}) = N - D - 1$. This implies that not all formations have a suitable stress matrix and the graph has to be carefully designed. Note that the stress matrix is also invariant to affine transformations [26, 31], hence no additional calculations are needed for time-varying target formations once the stress matrix is determined using the nominal formation. This property is crucial for flexible time-varying maneuvering with affine transformations.

2.5 Leader-Follower Strategy

The maneuverability is typically ensured by a technique called leader-follower strategy in which a small set $\mathcal{V}_l$ of agents are set aside to be the *leaders* and the rest $\mathcal{V}_f = \mathcal{V} \setminus \mathcal{V}_l$ are the *followers*. Without the loss of generality, the first $|\mathcal{V}_l|$ agents are considered to be leaders, and the rest $|\mathcal{V}_f| = N - |\mathcal{V}_l|$ agents are followers. As such, the configuration vector can also be split as $\mathbf{z} = [\mathbf{z}_l^T, \mathbf{z}_f^T]^T$, and so can the nominal configuration and target configuration. The time-varying target formation is prescribed to the leaders while the followers only need to follow the leaders and stay in formation without any knowledge of target configurations. There are several key problems the leader-follower strategy should address:

- The tracking of the followers if target positions are unknown.
- The requirement on the number and the spatial distribution of the leaders.

These questions will be answered in the following theorem but first, the term *affine localizability* will be defined.

Definition 2.4 (*Affine localizability* [26]). The nominal formation $(\mathcal{G}, \mathbf{p})$ is *affinely localizable* by the leaders if for any $\mathbf{z} = [\mathbf{z}_l^T, \mathbf{z}_f^T]^T \in \mathcal{A}(\mathbf{p})$, $\mathbf{z}_f$ can be uniquely determined by $\mathbf{z}_l$.

Affine localizability establishes a one-to-one relation between the positions of leaders and followers if the configuration is in the affine image. Next, a theorem asserts the conditions on the choice of leaders to achieve affine localizability. The theorem and its proof are given in [26], but we slightly change the narrative and provide the outline of an alternative proof. If we define a matrix augmentation operator $\bar{A}(\cdot)$ as

$$\bar{A}(\mathbf{A}) = [\mathbf{A} \mid \mathbf{1}], \tag{2.7}$$

where the all-one vector $\mathbf{1}$ is of appropriate size, the theorem can be stated as follows.

Theorem 2.1 (*Choice of leaders*)**.** Given the nominal formation $(\mathcal{G}, \mathbf{p})$ and the nominal configuration matrix $\mathbf{P} = [\mathbf{p}_1, ..., \mathbf{p}_N]^T \in \mathbb{R}^{N \times D}$ that satisfies $\bar{A}(\mathbf{P})$ full column rank, then $(\mathcal{G}, \mathbf{p})$ is affinely localizable if and only if the subconfiguration matrix for leaders $\mathbf{P}_l \in \mathbb{R}^{|\mathcal{V}_l| \times D}$ also satisfy $\bar{A}(\mathbf{P}_l)$ full column rank.

Proof. For any $\mathbf{z} \in \mathcal{A}(\mathbf{p})$, there exists $\bar{\mathbf{\Theta}} = [\mathbf{\Theta} \mid \mathbf{t}]$ such that the configuration matrix $\mathbf{Z} = [\mathbf{z}_1, ..., \mathbf{z}_N]^T \in \mathbb{R}^{N \times D}$ can be written by stacking individual positions as

$$\mathbf{Z} = \bar{A}(\mathbf{P})\bar{\mathbf{\Theta}}^T = \bar{\mathbf{P}}\bar{\mathbf{\Theta}}^T, \tag{2.8}$$

which admits a partition

$$\mathbf{Z}_l = \bar{A}(\mathbf{P}_l)\bar{\mathbf{\Theta}}^T = \bar{\mathbf{P}}_l\bar{\mathbf{\Theta}}^T, \tag{2.9}$$

$$\mathbf{Z}_f = \bar{A}(\mathbf{P}_f)\bar{\mathbf{\Theta}}^T = \bar{\mathbf{P}}_f\bar{\mathbf{\Theta}}^T. \tag{2.10}$$

(Sufficiency) If $\bar{A}(\mathbf{P}_l)$ full column rank of $D+1$, $\bar{\mathbf{\Theta}}^T$ can be uniquely determined as

$$\bar{\mathbf{\Theta}}^T = (\bar{\mathbf{P}}_l^T\bar{\mathbf{P}}_l)^{-1}\bar{\mathbf{P}}_l^T\mathbf{Z}_l, \tag{2.11}$$

which can then be used to determine $\mathbf{Z}_f$ using (2.10). Since $\mathbf{z}_f$ is only up to a reshaping of $\mathbf{Z}_f$, the nominal formation is affinely localizable.
(Necessity) If $\bar{A}(\mathbf{P}_l)$ is not full column rank, then (2.9) is an underdetermined system to which there are infinite number of solutions of $\bar{\mathbf{\Theta}}^T$. Hence, $\mathbf{Z}_f$ cannot be uniquely determined by (2.10), i.e., the nominal configuration is not localizable. ■

The implication of Theorem 2.1 is profound.

- For $\bar{\mathbf{P}}_l \in \mathbb{R}^{|\mathcal{V}_l| \times (D+1)}$ to be full column rank and have a left inverse, $|\mathcal{V}_l| \geq D+1$, i.e., the minimum number of leaders needed is $D+1$. This can also be described as $\{\mathbf{p}_i\}_{i \in \mathcal{V}_l}$ *affinely span* $\mathbb{R}^D$ [26], implying the leaders cannot be collinear in $\mathbb{R}^2$ or coplanar in $\mathbb{R}^3$ in geometry. There must be at least 3 agents as leaders in $\mathbb{R}^2$ and 4 in $\mathbb{R}^3$, which is not a strong requirement given that the size of practical networks can be very large. Some examples of the choice of leaders are given in Figure 2.3.

- Theorem 2.1 ensures that once the leaders track the prescribed target formation, the followers will also track the target formation without knowing the underlying affine transformation. Therefore, the affine formation maneuver can be achieved by controlling the small set of leaders either autonomously or manually.

- A generalized view of affine localizability is that given a small subset of all positions, the unknown positions can be affinely localized. This property can be used as an estimator when some observations of positions are unavailable. Later in Chapter 4, we will utilize this idea and further study the localization of relative positions to combat missing observations for affine formation control.

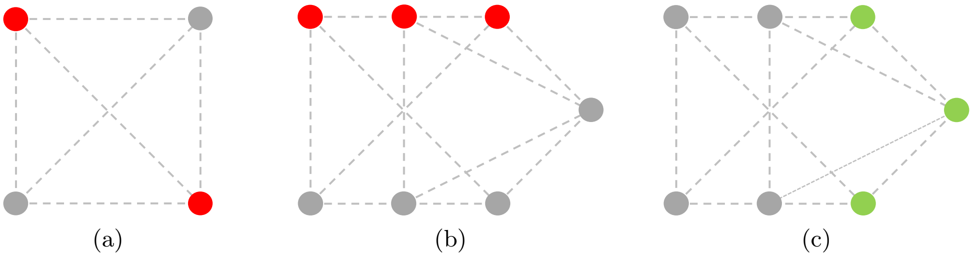


Figure 2.3: Some examples of the infeasible and feasible choices of leaders in $\mathbb{R}^2$. Colored nodes represent leaders. (a) infeasible due to $|\mathcal{V}_l| < D + 1$. (b) infeasible due to collinear leaders in $\mathbb{R}^2$. (c) feasible.

2.6 Evaluation Metric

We adopt *tracking error* as the universal metric to evaluate the performance of formation control.

Definition 2.5 (*Tracking error*)**.** The tracking error $\mathbf{e}_i(k) \in \mathbb{R}^D$ for agent i at discrete time k is defined as

$$\mathbf{e}_i(k) = \mathbf{z}_i(k) - \mathbf{z}_i^*(k). \tag{2.12}$$

To show the overall performance of the system, a scalar version of the tracking error is defined as

$$\delta(k) = \frac{1}{D|\mathcal{V}_f|} \sum_{i \in \mathcal{V}_f} \|\mathbf{e}_i(k)\|_2 \,, \tag{2.13}$$

which is all norms of $\mathbf{e}_i(k)$ averaged across all dimensions and all followers.

2.7 Summary

In this chapter, the most important concepts to understand affine formation control are presented. A few key statements that are directly involved in the following chapters are listed as follows.

- The stress matrix $\mathbf{L}$ which contains the equilibrium stress $\{l_{ij}\}_{(i,j)\in\mathcal{E}}$ can be calculated before control once the nominal formation is given.
- Leaders and followers shall take different control laws and only leaders are aware of the target configurations $\mathbf{z}^*(k)$.
- The idea of affine localizability can be generalized to determining unknown locations of agents if only a subset of agents' locations is given.

In the next chapter, controller designs for agents to track the time-varying target formation are introduced.

Distributed Formation Maneuvering 3

Having the basics of affine formation control, we now introduce the distributed controller designs for formation maneuvering in this chapter. First, the problem formulation is laid out describing the goal of the controllers. Then the control laws for both leaders and followers are introduced with sketches of proofs of convergences provided. Then a technique aiming to reduce the overhead of convergence is presented followed by simulations.

3.1 Problem Formulation

Recall that the position of agent i is denoted as $\mathbf{z}_i$. In this thesis, agents are assumed to be governed by single-integrator dynamics $\dot{\mathbf{z}}_i = \mathbf{u}_i$, which means the control input $\mathbf{u}_i$ is taken as the velocity information $\dot{\mathbf{z}}_i$. In the discrete domain, this relation can be translated into

$$\mathbf{z}_i(k+1) = \mathbf{z}_i(k) + \Delta t \mathbf{u}_i(k) \quad i \in \mathcal{V}, \tag{3.1}$$

which means the position of agent i at time $k+1$ is the position at k plus the control input $\mathbf{u}_i(k)$ between time interval Δt. The problem of distributed formation maneuvering can generally be stated as follows. Given a affinely localizable nominal formation $(\mathcal{G}, \mathbf{p})$ and prescribed time-varying target formations, design $\mathbf{u}_i$ for all $i \in \mathcal{V}$ such that $\mathbf{e}_i(k) \to \mathbf{0}$ as $k \to \infty$. Since the leader-follower strategy is adopted, this problem can be decomposed into two parts.

- Assuming that the tracking error for the leaders are always zero, i.e., $\mathbf{e}_i(k) = \mathbf{0}$ for all k and $i \in \mathcal{V}_l$, design $\mathbf{u}_i$ for $i \in \mathcal{V}_f$ using only local measurements and communications in the neighborhood to achieve $\mathbf{e}_i(k) \to \mathbf{0}$ for $i \in \mathcal{V}_f$ as $k \to \infty$ without revealing $\mathbf{z}_i^*(k)$ to the followers for any $i \in \mathcal{V}$.
- If the leaders are assumed to be autonomous, design $\mathbf{u}_i$ for $i \in \mathcal{V}_l$ using local measurements and communications in the neighborhood to achieve $\mathbf{e}_i(k) \to \mathbf{0}$ for $i \in \mathcal{V}_l$ as $k \to \infty$.

3.2 Control laws for the followers

The distributed control law under single-integrator dynamics to track formations with time-varying velocities is [26]

$$\mathbf{u}_i = -\frac{1}{\gamma_i} \sum_{j \in \mathcal{N}_i} l_{ij} (\mathbf{z}_{ij} - \dot{\mathbf{z}}_j) \quad i \in \mathcal{V}_f, \tag{3.2}$$

where $\gamma_i = \sum_{j \in \mathcal{N}_i} l_{ij}$, and l_{ij} are equilibrium stress that constitute $\mathbf{L}$. $\mathbf{z}_{ij} = \mathbf{z}_i - \mathbf{z}_j$ is considered as *relative position* observed by agent i, and the velocity term $\dot{\mathbf{z}}_j$ may be transmitted via a wireless communication network and acquired in different ways depending on the sensing capability of the agents. Assuming $\mathbf{z}_{ij}$ can be locally measured or calculated through wireless communication, control law (3.2) is fully distributed. The next theorem states the convergence of this control law.

Theorem 3.1 (*Convergence of control law (3.2)* [26]**).** If $\mathbf{e}_i(k) = \mathbf{0}$ for all k and $i \in \mathcal{V}_l$, and the velocity $\dot{\mathbf{z}}_l^*(k)$ is time-varying, then the tracking error $\mathbf{e}_i(k)$ for $i \in \mathcal{V}_f$ converges to zero globally and exponentially fast under control law (3.2).

The proof for this theorem is given in [26], hence we do not reiterate it here.

Theorem 3.1 claims if the leaders can always track their target positions, the followers can converge to their respective target positions. In practice, the tracking errors for the leaders are not zero, but if they are bounded and converging to zeros, then the tracking errors for the followers are also bounded and converging to zero.

Control law (3.2) is essentially a linear combination of stress l_{ij} and observations of relative positions and velocity information, which makes it very computationally efficient. The diagram of this control system is shown in Figure 3.1. Note that this control law is aimed at a general setup where the velocities for leaders are nonzero and time-varying (*dynamic formation*). In special cases where the leaders are in *static formation*, the control law (3.2) still applies but the velocity measurements are not needed. As adopted in [31, 26, 51] the control law can be simplified to

$$\mathbf{u}_i = -\sum_{j \in \mathcal{N}_i} l_{ij} \mathbf{z}_{ij} \quad i \in \mathcal{V}_f. \tag{3.3}$$

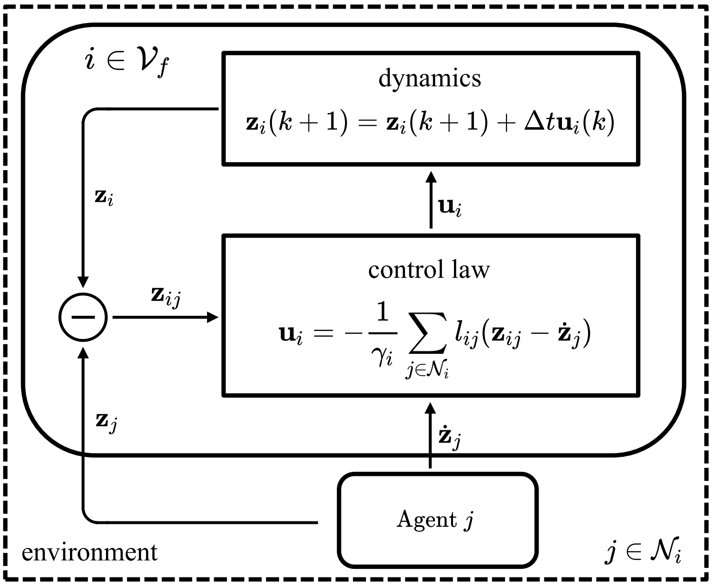

$i \in \mathcal{V}_f$
dynamics
$\mathbf{z}_i(k+1) = \mathbf{z}_i(k+1) + \Delta t \mathbf{u}_i(k)$
$\mathbf{u}_i$
$\mathbf{z}_i$
$\mathbf{z}_{ij}$
control law
$\mathbf{u}_i = -\frac{1}{\gamma_i} \sum_{j \in \mathcal{N}_i} l_{ij}(\mathbf{z}_{ij} - \dot{\mathbf{z}}_j)$
$\mathbf{z}_j$
$\dot{\mathbf{z}}_j$
Agent j
environment
$j \in \mathcal{N}_i$

Figure 3.1: An overview of distributed affine formation control systems

3.3 Control laws for the leaders

Since the leaders only take up a small subset of all agents, the study of leaders is, in most literature, omitted and they are simply assumed to perfectly track the target formation. Since the target positions are known to leaders, there are comparably more distributed solutions for the leaders. We introduce one simple controller design and give the proof of convergence for completeness but the leaders will not be in the scope of discussion for the rest of the thesis. Under single-integrator dynamics, the control law for the leaders is

$$\mathbf{u}_i = -\alpha(\mathbf{z}_i - \mathbf{z}_i^*) + \dot{\mathbf{z}}_i^* \quad i \in \mathcal{V}_l, \tag{3.4}$$

where α is the control gain that modulates the convergence speed, and $\dot{\mathbf{z}}_i^*$ is the target speed of the leaders. The following theorem asserts the convergence of control law (3.4).

Theorem 3.2 (*Convergence of control law (3.4)*)**.** Given the target formation $(\mathcal{G}, \mathbf{z}^*)$, the tracking error $\mathbf{e}_i(k)$ for all $i \in \mathcal{V}_l$ converges to zero exponentially fast.

Proof. For all $i \in \mathcal{V}_l$, by definition $\mathbf{e}_i = \mathbf{z} - \mathbf{z}^*$. As such, by substituting $\dot{\mathbf{z}}_i$ with $\mathbf{u}_i$ from (3.4),

$$\begin{aligned} \dot{\mathbf{e}}_i &= \dot{\mathbf{z}}_i - \dot{\mathbf{z}}_i^* \\ &= -\alpha(\mathbf{z}_i - \mathbf{z}_i^*) + \dot{\mathbf{z}}_i^* - \dot{\mathbf{z}}_i^* \\ &= -\alpha \mathbf{e}_i, \end{aligned} \tag{3.5}$$

which is an ordinary differential equation to which the solution is an exponential function decaying to zero. ■

It can be seen that the control systems for the leaders are detached from the ones for the followers, but leaders and followers are still interactive in terms of sensing and communication. Note that the premise for Theorem 3.1 is that the tracking errors for the leaders are strictly zero, which is not the case in practice for autonomous agents. But the tracking error for the leaders converges to zeros adopting this control law, and so will the tracking error for the followers.

3.4 Reconfiguration of Initial Positions

If the initial positions of agents are randomly drawn from a certain distribution, the distances that need to be covered from the initial positions to the target positions might not be the shortest possible, which can be regarded as an extra overhead introduced by random initialization. Figure 3.2 shows two cases of mapping between initial and target positions in which one clearly requires less distance to be covered. From an application's perspective, it is expensive to cover unnecessary distances in terms of fuel and time for applications e.g., space systems. This overhead might also cause chaos and increase the probability of collision e.g., a swarm of drones or robots in tight space.

In this section, we aim to reduce the overhead by remapping the agents and their prospective identity in target formation such that the distances to be covered from the initial locations to the target locations are minimized. Fortunately, this problem can be treated as a linear sum assignment problem (LSAP) which is a classical paradigm in linear programming and combinatorial optimization [52]. Note that this reconfiguration requires the agents to be homogeneous.

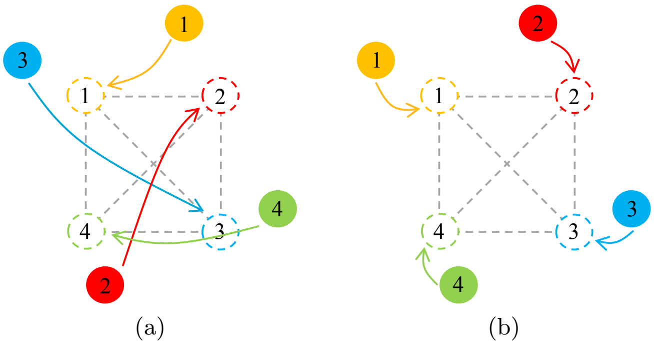


Figure 3.2: Two assignment cases with the same initialization. The dotted circles are the target positions and the filled circles are random initial positions. (a) is clearly less optimal than (b) in terms of the total distance to be covered to reach convergence and path crossing.

Define a cost matrix $\mathbf{C} \in \mathbb{R}^{N\times N}$

$$\mathbf{C} = \begin{bmatrix} \|\mathbf{z}_1^*(0) - \mathbf{z}_1(0)\|_2 & \cdots & \|\mathbf{z}_1^*(0) - \mathbf{z}_N(0)\|_2 \\ \vdots & \ddots & \vdots \\ \|\mathbf{z}_N^*(0) - \mathbf{z}_1(0)\|_2 & \cdots & \|\mathbf{z}_N^*(0) - \mathbf{z}_N(0)\|_2 \end{bmatrix}, \tag{3.6}$$

where the elements are the pairwise Euclidean distances between the initial positions and the target positions. LSAP gives the minimum sum of the cost of N entries that are not mutually in the same row or column of $\mathbf{C}$. It can be regarded as finding a permutation matrix $\mathbf{C}^p \in \mathbb{R}^{N\times N}$ to solve the linear program (LP) [53]

$$\begin{aligned} \min \quad & \sum_{i,j=1}^{N} C_{ij}C_{ij}^p \\ \text{s.t.} \quad & \sum_{i=1}^{N} C_{ij}^p = 1 \quad j = 1, ..., N \\ & \sum_{j=1}^{N} C_{ij}^p = 1 \quad i = 1, ..., N \\ & C^p \in \{0, 1\} \end{aligned}. \tag{3.7}$$

As such, the reassigned configuration vector is

$$\mathbf{z}^p(0) = (\mathbf{C}^p \otimes \mathbf{I}_D)\mathbf{z}(0), \tag{3.8}$$

where $\mathbf{z}(0)$ is the initial configuration vector $\mathbf{z}(0) = [\mathbf{z}_1(0)^T, ..., \mathbf{z}_N(0)^T]^T \in \mathbb{R}^{DN}$. Note that LP (3.7) can be efficiently solved by classical algorithms e.g. the Munkres Algorithm [54]. Section 3.5 will show the improvement of configuration reassignment, but it will be assumed that the initial configurations are reassigned by default in algorithms and simulations unless specified for the rest of the thesis.

3.5 Simulations

We establish a simulation framework and some simulation parameters will be fixed for the rest of the thesis. A hexagonal formation in $\mathbb{R}^2$ shown in Figure 3.3 is adopted as nominal formation which is also used in [51]. The graph has $N = 10$ nodes and $M = 30$ undirected edges. The initial positions of the agents are randomly drawn from the normal distribution $\boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{P}_0)$ where we take $\mathbf{P}_0 = 2\mathbf{I}_D$. For this discrete-time control system, we choose 1kHz as the control frequency meaning a $\Delta t = 1$ms interval between successive control inputs and a simulation duration of 60s ($K_{max} = 60000$) in which the formation maneuvers in $\mathbb{R}^2$ under affine transformations. The pseudocode for distributed formation control is presented in Algorithm 3.1 Note that all algorithms in this thesis will only show controls for the followers for simplicity and assume the initial positions are reconfigured by (3.8).

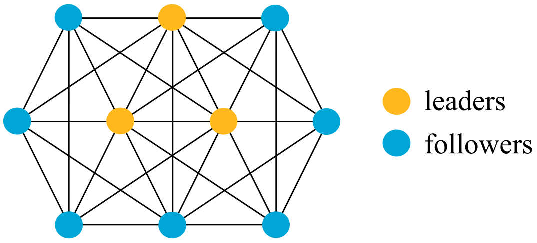


Figure 3.3: The nominal formation adopted in this thesis. The leaders are marked in orange on the graph and the followers in cyan.

Algorithm 3.1 *Distributed formation control*

```
Initialization
    Calculate stress matrix $\mathbf{L}$
    Draw initial positions $\mathbf{z}_i(0) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{P}_0)$
    $k = 0$
while $k \leq K_{max}$ do
    for $i \in \mathcal{V}_f$ do
        $\mathbf{u}_i(k) = -\frac{1}{\gamma_i} \sum_{j \in \mathcal{N}_i} l_{ij}(\mathbf{z}_{ij}(k) - \dot{\mathbf{z}}_j(k))$        ▷ Control law (3.2)
        $\mathbf{z}_i(k+1) = \mathbf{z}_i(k) + \Delta t \mathbf{u}_i(k)$
    end for
    $k = k + 1$
end while
```

Using Algorithm 3.1, the true and reference (target) trajectories are presented in Figure 3.4, where the agents start in random locations, then converge to the target positions, and finally track the maneuvers in space. Figure 3.5 shows the individual dynamics including position and velocity in both dimensions. It should be noted that the units for the figures can be scaled based on applications.

As mentioned in the theory, reconfiguring the initial positions will result in less distance to be covered towards convergence. Figure 3.6 shows the comparison of two cases

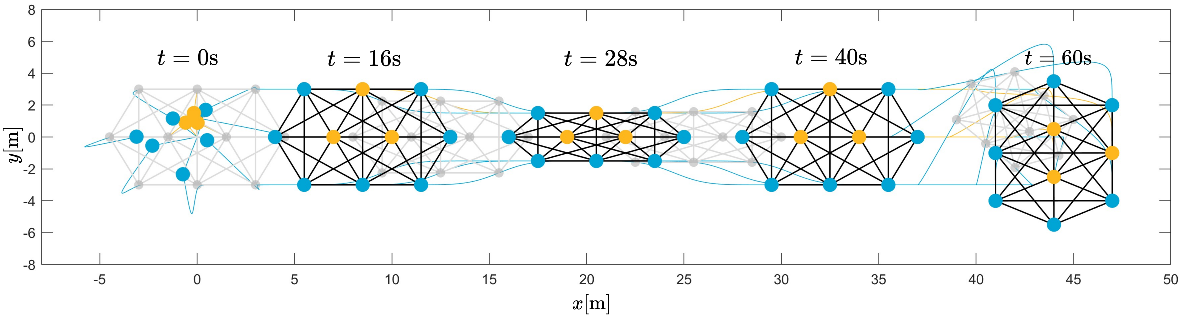


Figure 3.4: Trajectories of one experiment using Algorithm 3.1. The colored nodes denote the true positions of agents, and the translucent (light gray) nodes represent the target positions at other times.

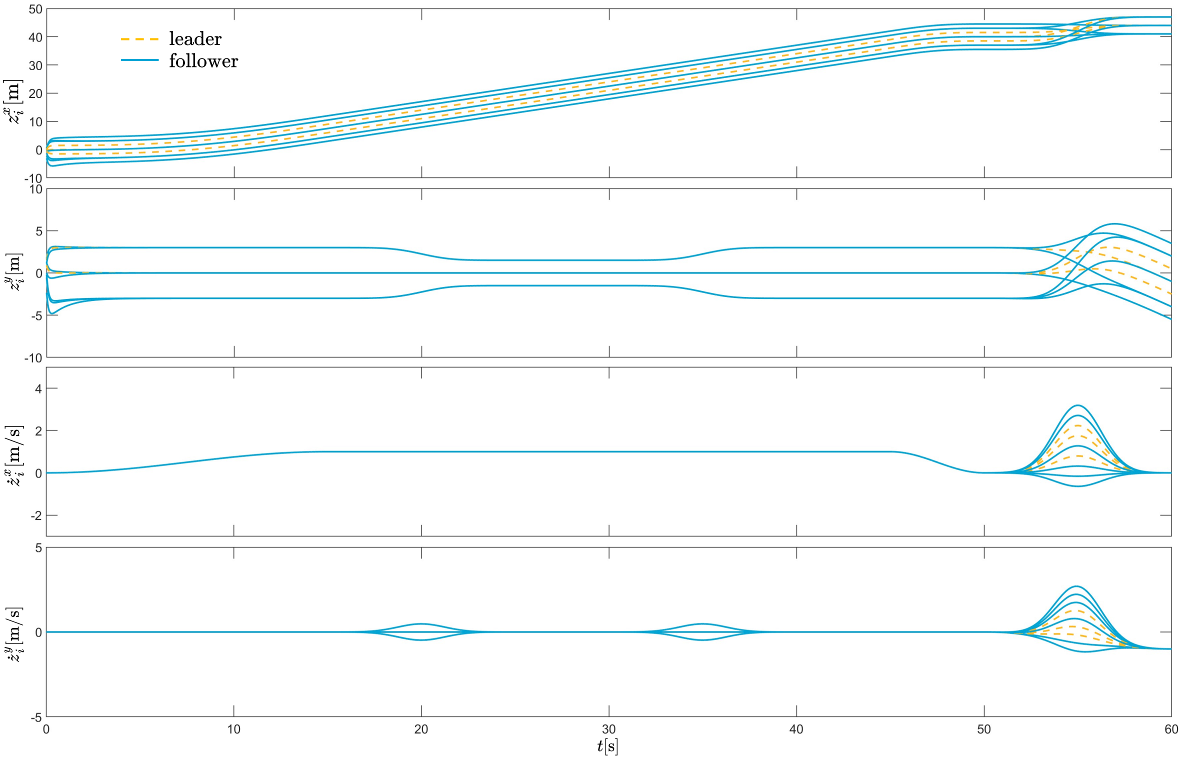
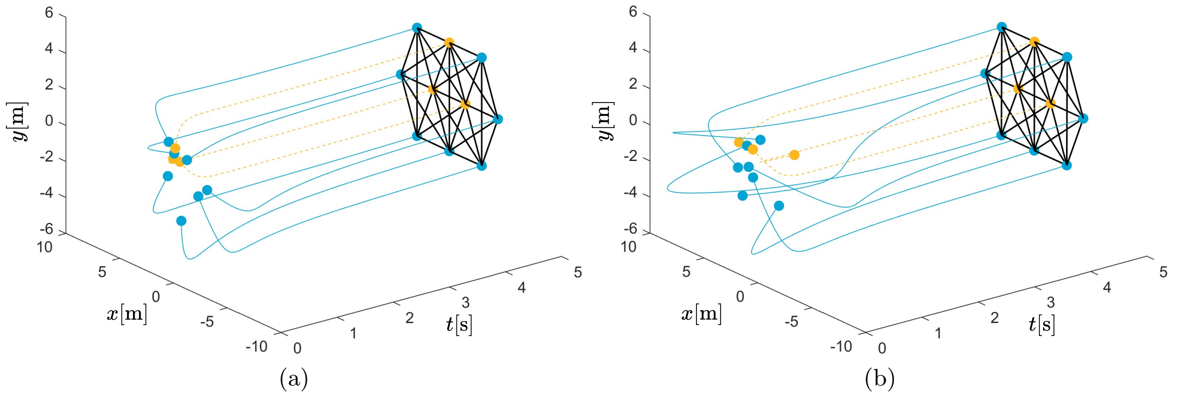


Figure 3.5: The individual dynamics of agents across time, z_i^x and z_i^y are the positions on both dimensions, and $\dot{z}_i^x$ and $\dot{z}_i^y$ are the velocities on both dimensions.

with random initialization. It can be observed that with reconfiguration the trajectories are less chaotic at the beginning and there is no path crossing. Figure 3.7 shows the improvements in the convergence time, where the curve with reconfiguration enabled is lower at the beginning on average over 50 Monte Carlo experiments[1]. It is easy to imagine that in very large networks with random initializations, the improvements can be significant. It is worth mentioning that Figure 3.7 shows both normal and log scale.

[1] For the rest of this thesis, the convergence plot will show the mean of 50 experiments with random initializations with the shaded area being the region of standard deviation unless specified.



(a)

(b)

Figure 3.6: Two experiments using Algorithm 3.1 for the first 5 seconds with LSAP reconfiguration of initial positions in (a) and without reconfiguration in (b).

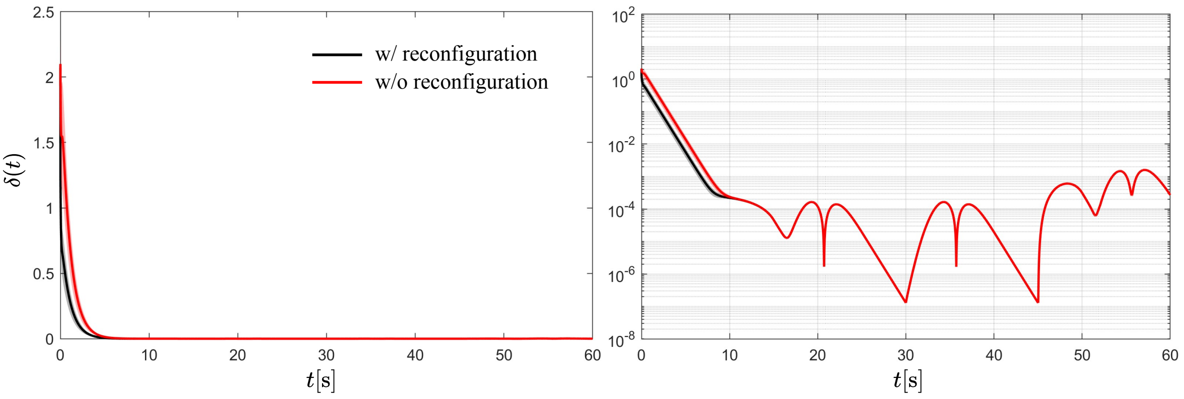


Figure 3.7: Tracking error δ of two cases under normal and log scale. Solid lines are the mean of 50 experiments and the shaded areas are the error standard deviation.

Since the log scale better presents the nuances, the convergence plots throughout the thesis will be presented this way. As seen in the figure, the tracking error is "bumpy" under the log scale, which is caused by the discretization of a continuous-time system. The time step Δt limits the precision of control, which is also why a 1kHz frequency is chosen for sufficient accuracy. But under the normal scale, the error curves are smooth.

3.6 Summary

This chapter presented the controller designs for distributed affine formation control under single-integrator dynamics. The goal of tracking the time-varying target positions is accomplished and verified with simulation. The main results can be summarized as

- Controller design for followers in general dynamic formation settings and convergence statement.
- Controller design for leaders with convergence proof.

- A reconfiguration of initial positions using an LSAP formulation which offers improvements on the convergence and possibility of collision.

all of which are verified with simulations. The controllers in this chapter are running in noise and failure-free environments. In the next chapter, the observation losses of the relative positions which are critical inputs for the controllers are modeled and tackled.

Relative Affine Localization 4

The previous chapters establish a paradigm of distributed formation control and set up a performance baseline in an ideal environment. In practice, such an environment is non-existent and the systems are always challenged with disturbances, resource allocations, power consumption limits, etc. From this chapter on, we aim to deal with a specific type of challenge: observation losses. For various reasons reviewed in Chapter 1, observation losses are detrimental to the controllers presented in the previous chapter. But fortunately, the affine formation itself can be considered as a spatial constraint that makes it possible to restore the missing observations.

We first motivate the idea of affine localization from the concept of affine localizability introduced in Definition 2.4. Then the theory of affine localization is presented followed by the extension to the relative domain. Next, the methods will be tailored to deal with affine transformations with additional constraints. To raise awareness of the convergence state, a distributed indicator function is designed and will be used in later chapters.

4.1 Motivation and Problem Formulation

In Chapter 2, the concept of affine localizability establishes that the positions of the followers can be localized by those of leaders if they satisfy certain conditions. The rationale behind this is that the underlying affine transformation parameters, $\boldsymbol{\Theta}$ and $\mathbf{t}$, can be uniquely determined by the positions of leaders. Since the formation is in affine image $\mathcal{A}(\mathbf{p})$, the followers' positions can be calculated by applying the same affine transformation to the nominal configuration. As is discussed in Chapter 2, we could generalize the concepts of "leaders" and "followers" to known and unknown (positions of) agents. However, in affine formation control systems, relative positions $\mathbf{z}_{ij}$ are of interest instead of the absolute ones. As such, a further adaptation of distributed localization of unknown relative positions is needed, which is named *relative affine localization* (RAL). Note that RAL is different from the relative localization methods introduced in Chapter 1 because it does not involve any sensors and conducts the estimation through known observations in the neighborhood. In this thesis, observations refer to the relative positions $\mathbf{z_{ij}}$ or edge states that are needed by the controllers although relative positions might be acquired by distance and bearing measurements in practice.

The formulation of affine localization from a global perspective is straightforward. The node set $\mathcal{V}$ of graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, like the leader-follower split, can be divided into $\mathcal{V} = (\mathcal{V}_k, \mathcal{V}_m)$ where $\mathcal{V}_k$ contains the nodes whose positions are known, and $\mathcal{V}_m$ contains the nodes whose positions are missing or unavailable. The goal is to restore the positions $\mathbf{z}_i$ for $i \in \mathcal{V}_m$ using those for $i \in \mathcal{V}_k$. For relative affine localization, we adopt a formulation in a distributed fashion. Locally, for each agent $i \in \mathcal{V}_f$, we split the set of neighbors into $\mathcal{N}_i = (\mathcal{N}_i^k, \mathcal{N}_i^m)$ in which $\mathcal{N}_i^k$ contains the neighbors w.r.t. whom the relative position observations are known, and $\mathcal{N}_i^m$ contains the neighbors w.r.t. whom the observations are missing or unavailable. Then $\left|\mathcal{N}_i^k\right|$ indicates the number of available observations for agent i. The goal is then to retrieve $\mathbf{z}_{ij}$ for $j \in \mathcal{N}_i^m$. Based on the nature of observation losses, the split of the set of neighbors can be random or permanent across time. For the introduction of solutions, we assume a random split until Chapter 7 where a permanent split will also be evaluated. Algorithm 4.1 describes the steps of affine formation control in presence of missing observations of relative positions. The results of the convergence are shown in Figure 4.1 where is it shown that under random loss of observations the formation will not converge optimally.

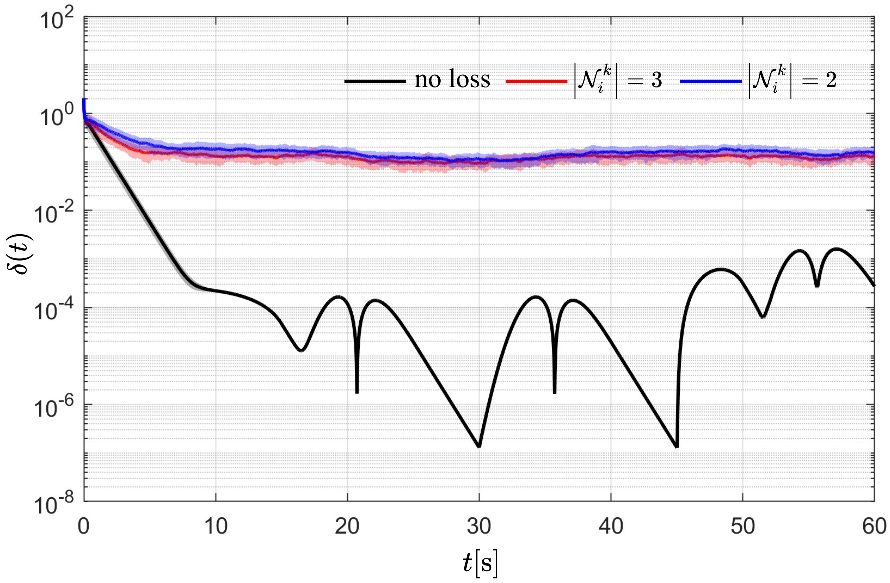


Figure 4.1: Simulations using Algorithm 4.1 with different numbers of known observations per agent $\left|\mathcal{N}_i^k\right|$. "no loss" is the case where there are no observation losses and is simulated with Algorithm 3.1

4.2 General Affine Localization

4.2.1 Localization with Absolute Measurements

We start with affine transformations established in (2.4), which is restated here

$$\mathbf{z}_i^*(k) = [\mathbf{\Theta}(k) \mid \mathbf{t}(k)] \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix} = \bar{\mathbf{\Theta}}(k)\bar{\mathbf{p}}_i \quad i \in \mathcal{V}, \tag{4.1}$$

where $\mathbf{z}_i^*(k)$ is the target position at time k and $\mathbf{p}_i$ is the nominal position. As is defined, the time-varying target configurations are always in affine image $\mathcal{A}(\mathbf{p})$. If the system has sufficiently converged using controller (3.2) and (3.4), $\mathbf{z}_i(k) \to \mathbf{z}_i^*(k)$ as $k \to \infty$ for all $i \in \mathcal{V}$. Hence, the configuration $\mathbf{z}(k) = [\mathbf{z}_1^T(k), ..., \mathbf{z}_N^T(k)]^T \in \mathbb{R}^{DN}$ can

Algorithm 4.1 *Distributed formation control with random observations losses*

```
Initialization
    Calculate stress matrix $\mathbf{L}$
    Draw initial positions $\mathbf{z}_i(0) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{P}_0)$
    $k = 0$
while $k \leq K_{max}$ do
    for $i \in \mathcal{V}_f$ do
        Define $\mathcal{N}_i = (\mathcal{N}_i^k, \mathcal{N}_i^m)$
        $\mathbf{u}_i(k) = -\frac{1}{\gamma_i} \sum_{j \in \mathcal{N}_i^k} l_{ij}(\mathbf{z}_{ij}(k) - \dot{\mathbf{z}}_j(k))$        ▷ Control law
        $\mathbf{z}_i(k+1) = \mathbf{z}_i(k) + \Delta t \mathbf{u}_i(k)$
    end for
    $k = k + 1$
end while
```

also be regarded to be in the affine image, which implies

$$\mathbf{z}_i(k) = [\boldsymbol{\Theta}(k) \mid \mathbf{t}(k)] \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix} = \bar{\boldsymbol{\Theta}}(k)\bar{\mathbf{p}}_i \tag{4.2}$$

for $i \in \mathcal{V}$. Aggregating for all agents, (4.2) can be written in matrix form

$$\mathbf{Z}(k) = [\mathbf{P} \mid \mathbf{1}_N] \, \bar{\boldsymbol{\Theta}}(k)^T = \bar{\mathbf{P}}\bar{\boldsymbol{\Theta}}(k)^T, \tag{4.3}$$

where $\mathbf{Z}(k) = [\mathbf{z}_1(k), ..., \mathbf{z}_N(k)]^T \in \mathbb{R}^{N \times D}$ and $\mathbf{P} = [\mathbf{p}_1, ..., \mathbf{p}_N]^T \in \mathbb{R}^{N \times D}$.

The following lemma formally states the necessary and sufficient conditions for affine localization, which can be regarded as a direct corollary of Theorem 2.1. The proof is omitted since it is essentially identical to the proof of Theorem 2.1.

Lemma 4.1 (*Global affine localization*)**.** Assuming configuration $\mathbf{z}(k) \in \mathbb{R}^{DN}$ is in affine image $\mathcal{A}(\mathbf{p})$, $\mathbf{z}_i(k)$ for $i \in \mathcal{V}_m$ can be uniquely determined if and only if $|\mathcal{V}_k| \geq D+1$ and $\{\mathbf{p}_i(k)\}_{i \in \mathcal{V}_k}$ affinely span $\mathbb{R}^D$.

Recollect that affine span is equivalent to full column rank for augmented configuration matrix as introduced in the discussion of Theorem 2.1. To better describe and distinguish the set of known positions, an observation (selection) matrix $\boldsymbol{\Phi}(k) \in \mathbb{R}^{|\mathcal{V}_k| \times N}$ is used such that the rows of $\boldsymbol{\Phi}(k)$ is drawn from $|\mathcal{V}_k|$ rows of an identity matrix $\mathbf{I}_N$ satisfying $\mathbf{1}_{|\mathcal{V}_k|}^T \boldsymbol{\Phi}(k) \mathbf{1}_N = |\mathcal{V}_k|$. Again, the observation matrix can be random or permanent depending on the setup. But it is assumed that the observation matrix $\boldsymbol{\Phi}(k)$ is always known. The selection effect can be visualized in Figure 4.2 where $\mathbf{X}(k)$ stacks $\mathbf{z}_i(k)$ for $i \in \mathcal{V}_k$. And (4.3) can be adapted as

$$\mathbf{X}(k) = \boldsymbol{\Phi}(k)\bar{\mathbf{P}}\bar{\boldsymbol{\Theta}}(k)^T = \mathbf{H}(k)\bar{\boldsymbol{\Theta}}(k)^T, \tag{4.4}$$

where $\mathbf{H}(k) \triangleq \boldsymbol{\Phi}(k)\bar{\mathbf{P}}$. As in the proof of Theorem 2.1, the parameter matrix $\bar{\boldsymbol{\Theta}}$ can be estimated from this linear equation with the formulation

$$\min_{\bar{\boldsymbol{\Theta}}(k)} \quad \left\| \mathbf{H}(k)\bar{\boldsymbol{\Theta}}(k)^T - \mathbf{X}(k) \right\|_F^2, \tag{4.5}$$

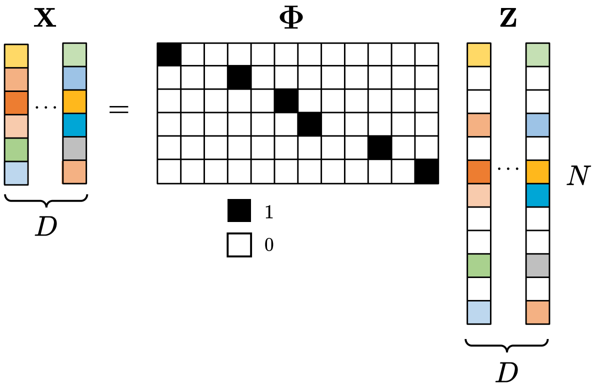


Figure 4.2: The selection effect of the observation matrix, where $\mathbf{\Phi}$ shall always be a wide matrix.

to which an analytical solution is given by

$$\bar{\mathbf{\Theta}}(k)^T = (\mathbf{H}(k)^T\mathbf{H}(k))^{-1}\mathbf{H}(k)^T\mathbf{X}(k). \tag{4.6}$$

The set of $\{\mathbf{z}_i(k)\}_{i\in\mathcal{V}_m}$ can then be estimated using (4.2). Note that this absolute localization is global and can not be readily applied in relative formation control since relative states $\mathbf{z}_{ij}(k)$ are needed, but it serves as a precursor to understanding the relative localization.

4.2.2 Decentralized Localization with Relative Measurements

The goal of decentralized localization for distributed formation control, is not to recover the absolute positions but to estimate relative positions in the neighborhood, i.e., $\{\mathbf{z}_{ij}(k)\}_{j\in\mathcal{N}_i}$ for $i \in \mathcal{V}_f$. The idea of relative affine localization, similar to absolute localization, is also a two-step estimation by first estimating transformation parameters. The difference is that for relative localization, the minimum number of observations required is reduced.

We first prepare some notations for the following theorem. Define relative nominal positions as $\mathbf{p}_{ij} = \mathbf{p}_i - \mathbf{p}_j$ for all $i \in \mathcal{V}_f$, and an local observation matrix $\mathbf{\Phi}_i(k) \in \mathbb{R}^{|\mathcal{N}_i^k|\times|\mathcal{N}_i|}$. Similar to absolute localization, the known relative positions are stored in $\mathbf{H}_i(k) \triangleq \mathbf{\Phi}_i(k)\mathbf{P}_i$, where $\mathbf{P}_i \in \mathbb{R}^{|\mathcal{N}_i|\times D}$ stacks all the nominal positions in the neighborhood $\{\mathbf{p}_{ij}\}_{j\in\mathcal{N}_i}$.

Theorem 4.1 (*Relative affine localization*)**.** Assume configuration $\mathbf{z}(k) \in \mathbb{R}^{DN}$ is in affine image $\mathcal{A}(\mathbf{p})$. For agent $i \in \mathcal{V}_f$, given the split $\mathcal{N}_i = (\mathcal{N}_i^k, \mathcal{N}_i^m)$, the missing relative positions $\{\mathbf{z}_{ij}(k)\}_{j\in\mathcal{N}_i^m}$ for can be locally and uniquely determined if and only if $\mathbf{H}_i(k)$ full column rank.

Proof. If configuration $\mathbf{z}(k) \in \mathbb{R}^{DN}$ is in affine image $\mathcal{A}(\mathbf{p})$, then by definition it holds that

$$\begin{aligned}\mathbf{z}_{ij}(k) &= \mathbf{z}_i(k) - \mathbf{z}_j(k) \\ &= \boldsymbol{\Theta}(k)\mathbf{p}_i + \mathbf{t}(k) - (\boldsymbol{\Theta}(k)\mathbf{p}_j + \mathbf{t}(k)) \\ &= \boldsymbol{\Theta}(k)\mathbf{p}_{ij} \end{aligned} \tag{4.7}$$

for $i \in \mathcal{V}_f, j \in \mathcal{N}_i$. Then a local set of linear equations could be established as

$$\mathbf{X}_i(k) = \boldsymbol{\Phi}_i(k)\mathbf{P}_i\boldsymbol{\Theta}(k)^T = \mathbf{H}_i(k)\boldsymbol{\Theta}(k)^T, \tag{4.8}$$

where $\mathbf{X}_i(k)$ stores $\mathbf{z}_{ij}(k) \in \{\mathbf{z}_{ij}(k)\}_{j\in\mathcal{N}_i^k}$.
(Sufficiency) If $\mathbf{H}_i(k)$ full column rank, then the global affine transformation matrix $\boldsymbol{\Theta}(k)$ can be locally and uniquely determined by

$$\boldsymbol{\Theta}_i(k)^T = (\mathbf{H}_i(k)^T\mathbf{H}_i(k))^{-1}\mathbf{H}_i(k)^T\mathbf{X}_i(k), \tag{4.9}$$

followed by the local determination of missing observation

$$\mathbf{z}_{ij}(k) = \boldsymbol{\Theta}_i(k)\mathbf{p}_{ij} \quad j \in \mathcal{N}_i^m. \tag{4.10}$$

(Necessity) If $\mathbf{H}_i(k)$ is not full-column rank, then there exists no left inverse of $\mathbf{H}_i(k)$ which indicates that unique and non-trivial local estimation for $\boldsymbol{\Theta}(k)$ does not exist. Then items in $\{\mathbf{z}_{ij}(k)\}_{j\in\mathcal{N}_i^m}$ cannot be uniquely determined.

■

The rank condition implies that $\left|\mathcal{N}_i^k\right| \geq D$ and geometry requirements on the observations. The split $\mathcal{N}_i = (\mathcal{N}_i^k, \mathcal{N}_i^m)$ that fulfill these conditions are called *geometrically feasible* for relative affine localization. Figure 4.3 gives some geometrically feasible and infeasible known relative positions. There is another intuitive understanding of the minimum number of positions or relative positions needed by Lemma 4.1 or Theorem 4.1, respectively. For general affine transformations, according to Table 2.1 the DOF are 6 and 12 for $\mathbb{R}^2$ and $\mathbb{R}^3$, respectively. DOF is the same as the number of parameters in $\bar{\boldsymbol{\Theta}}$ which needs at least the same number of independent equations to solve. For instance, 3 positions gives 6 equations in $\mathbb{R}^2$ which satisfy $D+1$ positions. For relative localization, the translation vector $\mathbf{t}(k)$ need not be determined according to (4.7), so fewer relative positions are needed to determine $\boldsymbol{\Theta}$.

However, for the circumstances RAL is not geometrically feasible, i.e., $\mathbf{H}_i(k)$ not full column rank, the following approaches can be considered.

- If $\boldsymbol{\Phi}_i(k)$ is locally designed, reconstruct such that it selects different relative positions or add more rows to it such that $\mathbf{H}_i(k)$ is full-rank.
- If $\boldsymbol{\Phi}_i(k)$ is determined by the environment but the formation is restricted and known to be a special case of affine transformation, e.g., rotation, then inversion-free solutions in the next section can be adopted.
- If $\boldsymbol{\Phi}_i(k)$ is determined by environment and none of the above is viable, one can turn to the model-based methods introduced in Chapter 6.

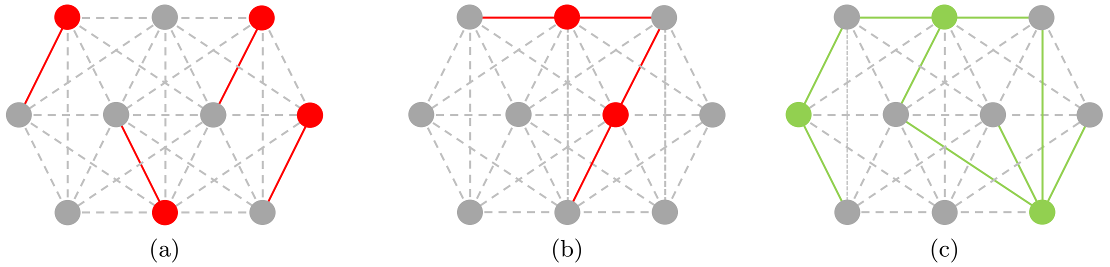


Figure 4.3: Some examples of infeasible and feasible available relative positions to enable RAL in $\mathbb{R}^2$. Colored nodes represent agents of interest. (a) infeasible due to $|\mathcal{N}_i^k| < D$ locally. (b) infeasible due to collinear relative positions. (c) feasible.

In practice, the assumption that $\mathbf{z}(k) \in \mathbb{R}^{DN}$ is in affine image $\mathcal{A}(\mathbf{p})$ almost never strictly holds, but we could still formulate the problem as

$$\min_{\mathbf{\Theta}(k)} \quad \left\|\mathbf{H}_i(k)\mathbf{\Theta}(k)^T - \mathbf{X}_i(k)\right\|_F^2, \tag{4.11}$$

and obtain estimates using an analytical solution

$$\hat{\mathbf{\Theta}}_i(k)^T = (\mathbf{H}_i(k)^T\mathbf{H}_i(k))^{-1}\mathbf{H}_i(k)^T\mathbf{X}_i(k). \tag{4.12}$$

Finally, the local RAL estimates of missing observations are

$$\mathbf{z}_{ij}^{\mathrm{ral}}(k) = \hat{\mathbf{\Theta}}_i(k)\mathbf{p}_{ij} \quad j \in \mathcal{N}_i^m. \tag{4.13}$$

For the known observations, we simply take them as they are, i.e.,

$$\mathbf{z}_{ij}^{\mathrm{ral}}(k) = \mathbf{z}_{ij}(k) \quad j \in \mathcal{N}_i^k, \tag{4.14}$$

since there are no better estimates than the true observations.

4.3 Constrained Relative Affine Localization

The previous section gives a solution to estimating parameters of general affine transformations. In practice, special cases of affine transformation (see Table 2.1) are sometimes demanded. For example, in [55], the formation requires a Euclidean (rigid) transform, and in [56, 57], the formation can be maneuvered up to a similarity transform. Under these circumstances, the degrees of freedom are reduced and the transformation parameters can be confined within a manifold. Although (4.12) is still applicable to these cases, we could further confine the solution space by levying constraints on 4.11 and relax the rank condition on $\mathbf{H}_i(k)$ in (4.12). In this section, we discuss the solution to several cases including scaling only, Euclidean transform, and similarity transform.

4.3.1 Scaling Only

If the transformation only involves scaling, the transformation matrix $\mathbf{\Theta}$ degenerates to

$$\mathbf{\Theta} = \mathrm{diag}(s_1, ..., s_D), \tag{4.15}$$

where s_d, for $d = 1, ..., D$, are the scaling factors for each dimension. Then (4.11) could be reformulated as

$$\begin{aligned} \min_{\mathbf{\Theta}(k)} \quad & \|\mathbf{H}_i(k)\mathbf{\Theta}(k) - \mathbf{X}_i(k)\|_F^2 \\ \text{s.t.} \quad & \mathbf{\Theta}(k) \, \text{diagonal}, \end{aligned} \tag{4.16}$$

which could be solved numerically or analytically. For the analytical solution, since the diagonality of $\mathbf{\Theta}(k)$ decouples the dimensions, (4.16) could be decomposed into D parallel Least Squares problems, i.e.,

$$\left\| s_d(k)\mathbf{h}_i^d(k) - \mathbf{x}_i^d(k) \right\|_2^2 \quad d = 1, ..., D, \tag{4.17}$$

where $\mathbf{h}_i^d(k)$ and $\mathbf{x}_i^d(k)$ are the d-th column of $\mathbf{H}_i(k)$ and $\mathbf{X}_i(k)$, respectively. The solution is given by

$$\hat{s}_d = \frac{\mathbf{h}_i^d(k)^T \mathbf{x}_i^d(k)}{\mathbf{h}_i^d(k)^T \mathbf{h}_i^d(k)} \quad d = 1, ..., D. \tag{4.18}$$

Then the local estimate of the transformation matrix is

$$\hat{\mathbf{\Theta}}_i(k) = \operatorname{diag}(\hat{s}_1, ..., \hat{s}_D). \tag{4.19}$$

4.3.2 Euclidean Transform

Euclidean transform retains the rigidity of the formation, and the inter-agent Euclidean distances are preserved. As such, translations, rotations, and reflections are considered Euclidean transforms. Since translation is a nuisance in relative affine localization and reflection is not suitable for continuous maneuvering of formation, we only consider a rotation in this setup. This degenerates the parameters matrix $\mathbf{\Theta}$ to a rotation matrix whose columns are orthonormal. Then the constrained formulation can be written as

$$\begin{aligned} \min_{\mathbf{\Theta}(k)} \quad & \left\|\mathbf{H}_i(k)\mathbf{\Theta}(k)^T - \mathbf{X}_i(k)\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{\Theta}(k)^T\mathbf{\Theta}(k) = \mathbf{I}_D. \end{aligned} \tag{4.20}$$

This formulation is recognized as the orthogonal Procrustes problem (OPP) [58] where an orthonormal matrix is sought to approximate rotations between two body frames. There are numerical and analytical solutions [59] available, and we adopt a singular value decomposition (SVD) based solution from [60]. If $\mathbf{H}_i(k)^T\mathbf{X}_i(k)$ has singular value decomposition (SVD)

$$\mathbf{H}_i(k)^T\mathbf{X}_i(k) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{4.21}$$

then the rotation matrix is given by

$$\hat{\mathbf{\Theta}}_i(k) = \mathbf{U}\mathbf{V}^T. \tag{4.22}$$

4.3.3 Similarity Transform

In similarity transforms, all dimensions are uniformly scaled by a scalar s on top of a rigid transform, i.e., $\mathbf{\Theta}$ degenerates to $s\mathbf{R}$. Hence, the constraints on $\mathbf{\Theta}$ are relaxed from orthonormality to orthogonality. Note that the classic OPP actually requires $\det(\boldsymbol{\Theta}) = 1$, so this problem can be seen as a generalization that is formulated as

$$\begin{aligned} \min_{s,\mathbf{R}(k)} \quad & \left\| s\mathbf{H}_i(k)\mathbf{R}(k)^T - \mathbf{X}_i(k) \right\|_F^2 \\ \text{s.t.} \quad & \mathbf{R}(k)^T\mathbf{R}(k) = \mathbf{I}_D. \end{aligned} \tag{4.23}$$

Several numerical solutions are summarized by [61], but here a heuristic SVD-based solution is also provided as an extension of (4.21) and (4.22). If $\mathbf{H}_i$ and $\mathbf{X}_i$ both have economy-sized decomposition

$$\mathbf{H}_i = \mathbf{U}_H \boldsymbol{\Sigma}_H \mathbf{V}_H^T \tag{4.24}$$

$$\mathbf{X}_i = \mathbf{U}_X \boldsymbol{\Sigma}_X \mathbf{V}_X^T, \tag{4.25}$$

then the scaling factor is given by

$$\hat{s} = \frac{1}{D}\operatorname{tr}(\boldsymbol{\Sigma}_H^{-1}\boldsymbol{\Sigma}_X), \tag{4.26}$$

with $\hat{\mathbf{R}}$ given by the same equation as (4.22). The final estimate of transformation parameters is

$$\hat{\boldsymbol{\Theta}}_i(k) = \hat{s}\hat{\mathbf{R}}. \tag{4.27}$$

The solution of (4.26) can be interpreted as the following: $\boldsymbol{\Sigma}_H$ and $\boldsymbol{\Sigma}_X$ represents the energy on each (rotated) dimension before and after transformation, which reflects the scale of the geometry. $\boldsymbol{\Sigma}_H^{-1}\boldsymbol{\Sigma}_X$ divides the energy on each dimension and then these values are averaged by the trace and the normalizer.

Algorithm 4.2 describes the steps of affine formation control using RAL for both general and constrained cases. Several more details should be noted. The algorithm requires specific types of affine transformation for constrained estimation. If this is not known, the algorithm takes it as the default case and uses general estimation for the transformation parameters. If $|\mathcal{N}_i^m| = 0$ or $|\mathcal{N}_i^k| = |\mathcal{N}_i|$ meaning there are no observation losses, Algorithm 4.2 gives the same results as Algorithm 3.1.

Some results are shown in Figure 4.4, where the tracking errors of different numbers of known observations $|\mathcal{N}_i^k|$ are simulated. Generally, with RAL applied, the system is converging steadily compared to having no mechanism to combat observation losses in Figure 4.1. However, the convergence is slowed by a different rate based on the number of observations available compared to the reference where there are no missing observations. This is because the more observations agents can locally access, the better estimates RAL can produce to lead to a quick convergence. In the extreme case where RAL is barely viable ($|\mathcal{N}_i^k| = 2$), the formation is not converging unless the solution space is confined and constrained RAL is activated. In general, it can be asserted that RAL can deal with random observation losses at the cost of slower convergence.

4.4 Convergence Indicator

As discussed in the previous section, RAL generally requires sufficient convergence to give satisfying estimates. However, Figure 3.7 already shows that for the first few time

Algorithm 4.2 *Distributed formation control using RAL*

```
Initialization
    Calculate stress matrix $\mathbf{L}$
    Draw initial positions $\mathbf{z}_i(0) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{P}_0)$
    Globally known nominal positions $\mathbf{p}_i$ for $i \in \mathcal{V}$
    $k = 0$
while $k \leq K_{max}$ do
    for $i \in \mathcal{V}_f$ do
        Define $\mathcal{N}_i = (\mathcal{N}_i^k, \mathcal{N}_i^m)$
        if $\boldsymbol{\Theta}(k)$ is diagonal then                          ▷ Scaling only
            Compute $\hat{\boldsymbol{\Theta}}_i(k)$ based on (4.19)
        else if $\boldsymbol{\Theta}(k)$ is orthonormal then                  ▷ Rigid transform
            Compute $\hat{\boldsymbol{\Theta}}_i(k)$ based on (4.22)
        else if $\boldsymbol{\Theta}(k)$ is orthogonal then              ▷ Similarity transform
            Compute $\hat{\boldsymbol{\Theta}}_i(k)$ based on (4.27)
        else
            Compute $\hat{\boldsymbol{\Theta}}_i(k)$ based on (4.12)    ▷ General affine transformation
        end if
        Reconstruct $\mathbf{z}_{ij}^{\mathrm{ral}}(k)$ by (4.13) or (4.14)
        $\mathbf{u}_i(k) = -\frac{1}{\gamma_i} \sum_{j \in \mathcal{N}_i} l_{ij}(\mathbf{z}_{ij}^{\mathrm{ral}}(k) - \dot{\mathbf{z}}_j(k))$      ▷ Control law
        $\mathbf{z}_i(k+1) = \mathbf{z}_i(k) + \Delta t \mathbf{u}_i(k)$
    end for
    $k = k + 1$
end while
```

instances, the system is not sufficiently converged. This leads to inaccurate estimates of the transformation parameters and the missing relative positions which naturally raises questions such as how to locally know the convergence state to assess the quality of the estimates. A heuristic indicator would be the tracking error $\delta_f(k)$ defined in (2.13) that compares the current positions and the target positions. But the major limitation is that the target configuration $\mathbf{z}^*(k)$ is only exposed to the leaders, hence the majority of agents in the formation cannot locally compute the tracking error.

In this section, a distributed indicator function is designed based on the consensus of affine transformations. First, we present a lemma stating the property of the parameter consensus in convergence.

Lemma 4.2 (*Consensus of affine transformation*)**.** Given a formation $(\mathcal{G}, \mathbf{z}(k))$, a local estimate of $\boldsymbol{\Theta}_i(k)$ for agent i is identical to $\boldsymbol{\Theta}_j(k)$ for $j \in \mathcal{N}_i$ if $\mathbf{z}(k)$ is in affine image $\mathcal{A}(\mathbf{p})$.

Proof. Assuming the global underlying affine transformation is characterized by $\bar{\boldsymbol{\Theta}}(k) = [\boldsymbol{\Theta}(k) \mid \mathbf{t}(k)]$. If $\mathbf{z}(k)$ is in affine image $\mathcal{A}(\mathbf{p})$ then (4.8) share the same solution space and $\boldsymbol{\Theta}_i(k) = \boldsymbol{\Theta}(k)$ for all $i \in \mathcal{V}$. Hence, $\boldsymbol{\Theta}_i(k) = \boldsymbol{\Theta}_j(k) = \boldsymbol{\Theta}(k)$ for $j \in \mathcal{N}_i$. ■

Lemma 4.2 implies that if the system has perfectly converged to the target formation, then local estimates of the transformation parameters should be equal to the neighbor-

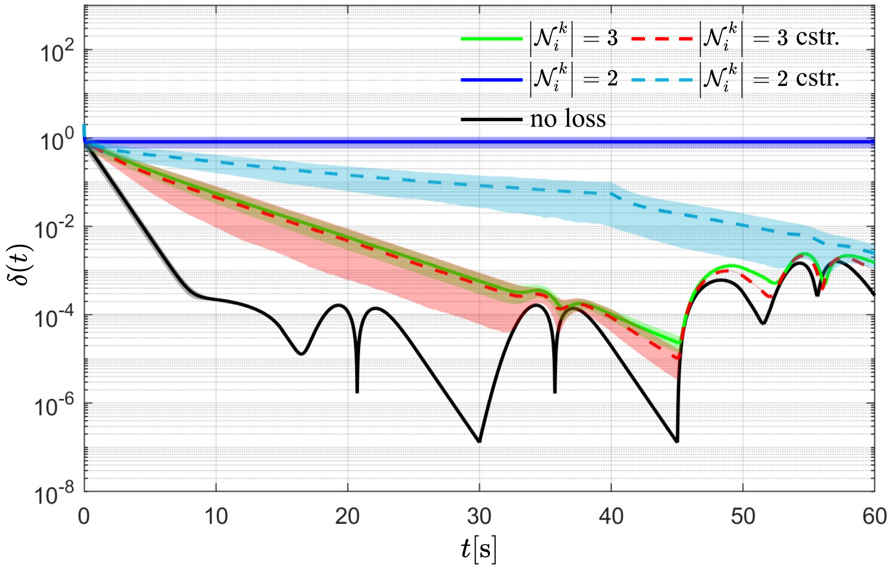


Figure 4.4: The convergence simulations of δ using RAL. "no loss" is the case where there are no missing observations and is generated with Algorithm 3.1. "cstr." or the dotted lines are using constrained RAL with the first 40 seconds being scaling only and 40-60 seconds being a similarity transform.

ing agents. As such, if the local estimates do not reach a consensus with neighbors, then the formation has not converged to the target formation. The converse statement of Lemma 4.2 is not true because $\mathbf{\Theta}_i(k) = \mathbf{\Theta}_j(k)$ does not necessarily mean $\mathbf{\Theta}_i(k) = \mathbf{\Theta}(k)$. However, in formation control systems, $\mathbf{z}(k)$ converges to $\mathbf{z}^*(k)$ as $k \to \infty$, $\mathbf{\Theta}_i(k)$ can be considered $\mathbf{\Theta}(k)$ with high confidence if $\mathbf{\Theta}_i(k) = \mathbf{\Theta}_j(k)$ for all $j \in \mathcal{N}_i$.

Inspired by Lemma 4.2, the local indicator function is designed as

$$f_i(\hat{\mathbf{\Theta}}_i(k), \{\hat{\mathbf{\Theta}}_j(k)\}_{j\in\mathcal{N}_i}) = \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i}\left\|\hat{\mathbf{\Theta}}_i(k) - \hat{\mathbf{\Theta}}_j(k)\right\|_F^2 \quad i \in \mathcal{V}_f, \tag{4.28}$$

where $\hat{\mathbf{\Theta}}_i(k)$ are the local estimate of transformation matrix using RAL. Note that the neighbors' estimates $\hat{\mathbf{\Theta}}_j(k)$ need to be transmitted through a wireless communication network.

This function approximates the trend of the tracking error $\delta_f(k)$, which can be seen from Figure 4.5, but the values are not necessarily the same. In the simulation, all neighbors' estimates $\hat{\mathbf{\Theta}}_j(k)$ are delayed by Δt in Figure 4.5 to reduce the demand on fast communication. As a result, the indicator function can well approximate the tracking error distributedly and will be employed in Chapter 6 to acquire the awareness of the convergence state of the system.

4.5 Summary

In this chapter, a practical challenge of observation losses is modeled and tackled using relative affine localization. In principle, the main contribution can be summarized as

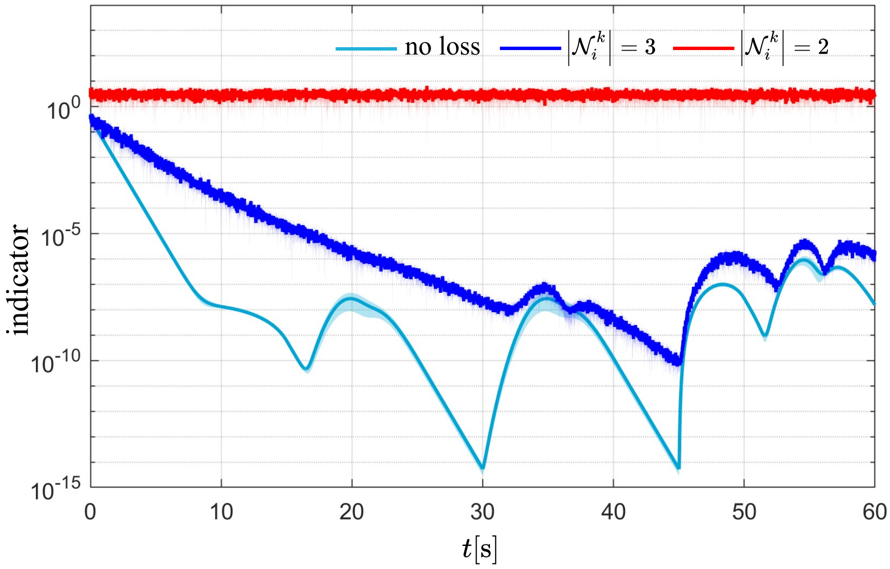


Figure 4.5: The mean values of the indicator function across the followers. The shaded areas are the standard deviation across the followers of 1 simulation, which is a bit different from the tracking error plots. The case of ”no loss” is when no observation losses occur but the estimation of transformation parameters $\mathbf{\Theta}(k)$ is still performed to calculate the function value.

- Relative localization (RAL) under general affine transformations is established to deal with observation losses.
- Constrained relative localization under special cases of affine transformations is derived to relax the conditions of geometrical feasibility for general RAL and improve the performance.
- A consensus-based convergence indicator is designed to locally evaluate the state of convergence.

Although the convergence indicator is not yet applied in this chapter, it will be utilized in later chapters. Although a prototype to deal with observation losses is proposed in this chapter, the system is still not practical since no noise of any kind has been modeled. In the next chapter, we will evaluate distributed formation control and relative affine localization under observation noises and propose some variants of RAL to improve the performance.

Data Driven Filtering and Smoothing 5

In Chapter 3 and 4, distributed formation control framework and ways using spatial constraints to combat missing observations are introduced. However, the theories are established in an idealized environment where there are no uncertainties of any kind. To approximate the reality, we introduce observation noises in this chapter and evaluate how the previous control framework and relative affine localization react. This leads to the two goals of the filtering:

- (primary) Minimize the impact of observation losses by maintaining the same level of tracking error as the normal case at the least cost of convergence speed.
- (secondary) Without compromising the primary goal, reduce the uncertainties introduced by observation noise to have smoother trajectories.

First, the noise modeling is given followed by the analysis of the statistical properties of the RAL as an estimator. Next, a tracking filter is performed in the time domain on the parameter level to expedite the convergence of RAL. Finally, a smoothing filter on the trajectory level is introduced to reduce the effect of noise. These filters use only available observations and RAL estimates without any assumption on the kinematics of agents or any other additional information, hence the name ”data-driven”.

5.1 Noise Modelling on Observations

In practice, the state observations are carried out by sensors onboard agents, which will inevitably introduce measurement noise. As no particular sensors are assumed for agents in this thesis, we simply model the observations as the true relative states $\mathbf{z}_{ij}(k)$ plus additive Gaussian noise. That is

$$\mathbf{y}_{ij}(k) = \mathbf{z}_{ij}(k) + \mathbf{v}_{ij}(k) \quad i \in \mathcal{V}_f,\ j \in \mathcal{N}_i, \tag{5.1}$$

where $\mathbf{v}_{ij}(k) \sim \mathcal{N}(\mathbf{0}_D, \mathbf{R})$ is the observation noise to the edge states and is often correlated over the dimension depending on the exact acquisition procedure of relative positions. It is assumed that observation noises are i.i.d. across all edges. Note that we only model noises on the observations of relative positions. The velocity term $\dot{\mathbf{z}}_j$ in control law (3.2) should generally be measured as well but we will not include its noise modeling and transmission losses in this thesis.

With this noise modeling on the relative states, the level of accuracy of distributed formation control is limited in addition to the time resolution Δt. As such, the tracking error can be assumed to have a decomposition

$$\mathbf{e}_i(k) = \boldsymbol{\varepsilon}_i(k) + \mathbf{n}_i(k), \tag{5.2}$$

where $\boldsymbol{\varepsilon}_i(k)$ is the deterministic error related to control and $\mathbf{n}_i(k)$ is the zero-mean stochastic error related to observation noise. This is an important assumption that will be depended on later in statistical analysis. The steps of affine formation control in presence of observation noise are described in Algorithm 5.1.

Algorithm 5.1 *Distributed affine formation control in observation noise*

```
Initialization
    Calculate stress matrix $\mathbf{L}$
    Draw initial positions $\mathbf{z}_i(0) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{P}_0)$
    Set noise covariance $\mathbf{R}$
    $k = 0$
while $k \leq K_{max}$ do
    for $i \in \mathcal{V}_f$ do
        for $j \in \mathcal{N}_i$ do
            Draw $\mathbf{v}_{ij}(k) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{R})$
            $\mathbf{y}_{ij}(k) = \mathbf{z}_{ij}(k) + \mathbf{v}_{ij}(k)$                ▷ Noisy observation
        end for
        $\mathbf{u}_i(k) = -\frac{1}{\gamma_i} \sum_{j \in \mathcal{N}_i} l_{ij}(\mathbf{y}_{ij}(k) - \dot{\mathbf{z}}_j(k))$        ▷ Control law
        $\mathbf{z}_i(k+1) = \mathbf{z}_i(k) + \Delta t \mathbf{u}_i(k)$
    end for
    $k = k + 1$
end while
```

5.2 Relative Affine Localization in Noise

Inevitably, the observation noise will negatively affect RAL and corrupt the estimation. If considering RAL as a statistical estimator, then the bias and variance of such estimator are worth investigating. Recall the local sensing model in the noiseless case (4.8), which is restated here

$$\mathbf{X}_i(k) = \mathbf{H}_i(k)\boldsymbol{\Theta}(k)^T, \tag{5.3}$$

where $\mathbf{X}_i(k), \mathbf{H}_i(k) \in \mathbb{R}^{|\mathcal{N}_i^k| \times D}$ are the configuration matrix stacking all the available relative states in the neighborhood and their corresponding nominal positions, respectively. According to observation model (5.1), (5.4) can be adapted as

$$\mathbf{Y}_i(k) = \mathbf{X}_i(k) + \mathbf{V}_i(k) = \mathbf{H}_i(k)\boldsymbol{\Theta}(k)^T + \mathbf{V}_i(k), \tag{5.4}$$

where $\mathbf{V}_i(k) = [\mathbf{v}_{i1}, ..., \mathbf{v}_{i|\mathcal{N}_i^k|}]^T$ in which $\mathbf{v}_{ij} \in \{\mathbf{v}_{ij}\}_{j \in \mathcal{N}_i^k}$ is the observation noise matrix for known relative positions. As such, general relative affine localization can then be adapted into

$$\min_{\boldsymbol{\Theta}(k)} \quad \left\|\mathbf{H}_i(k)\boldsymbol{\Theta}(k)^T - \mathbf{Y}_i(k)\right\|_F^2, \tag{5.5}$$

to which a distributed solution is given by

$$\hat{\boldsymbol{\Theta}}_i(k)^T = (\mathbf{H}_i(k)^T\mathbf{H}_i(k))^{-1}\mathbf{H}_i(k)^T\mathbf{Y}_i(k), \tag{5.6}$$

followed by

$$\hat{\mathbf{z}}_{ij}^{\mathrm{ral}}(k) = \hat{\mathbf{\Theta}}_i(k)\mathbf{p}_{ij} \tag{5.7}$$

for $i \in \mathcal{V}_f, j \in \mathcal{N}_i^m$. Again, for the known observations $j \in \mathcal{N}_i^k$, the estimates are taken as the observations

$$\hat{\mathbf{z}}_{ij}^{\mathrm{ral}}(k) = \mathbf{y}_{ij}(k). \tag{5.8}$$

Next, the statistical properties will be investigated, namely the bias and the variance of the estimator. From this point on, we only focus on the general affine localization on which we build other methods for simplicity, and constrained RAL should have a similar analysis. An overview of the systems in noisy cases is provided in Figure 5.1.

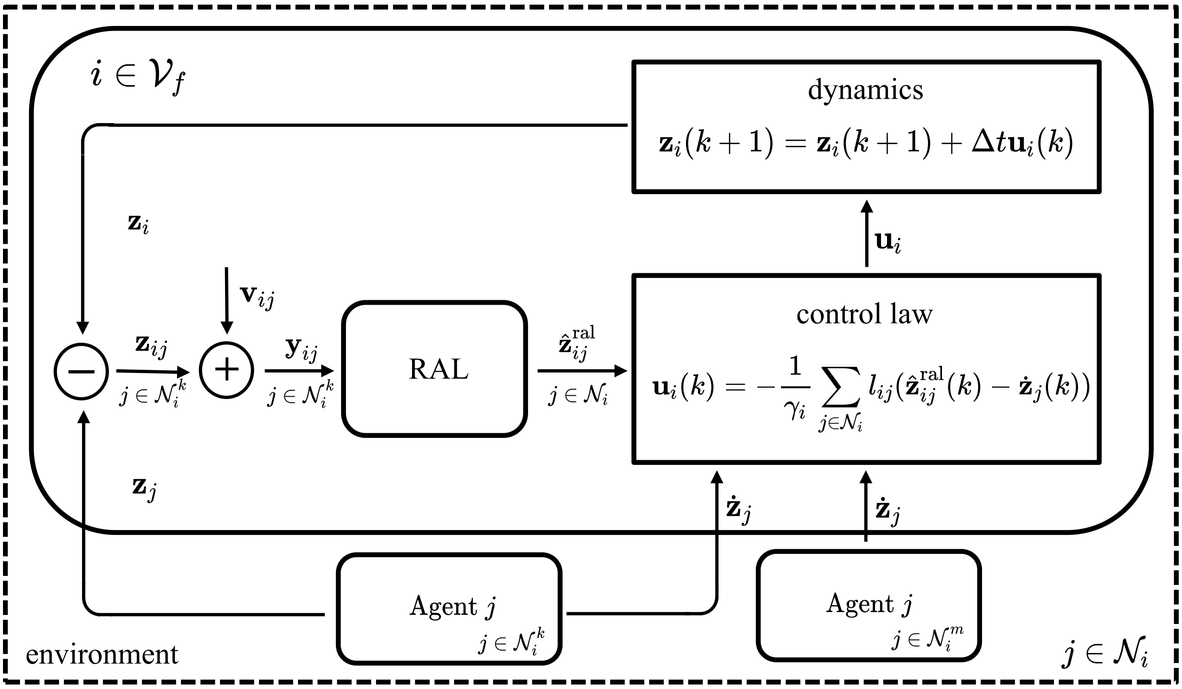


Figure 5.1: An overview of distributed affine formation control system with RAL implemented under observation noise.

5.2.1 Asymptotic Unbiasedness of RAL

Recall that the premise of relative affine localization is that the configuration $\mathbf{z}(k)$ is in affine image $\mathcal{A}(\mathbf{p})$. But in noisy environments, this is almost never achievable since the estimation is always subject to uncertainties. However, we can assume that the formation will converge to the point where the error related to observation noises is dominant over the control error. In other words, the $\boldsymbol{\varepsilon}_i(k)$ is negligible compared to $\mathbf{n}_i(k)$ in (5.2) if the system has sufficiently converged. As such, based on the definition of tracking error, we could write

$$\mathbf{z}_{ij}(k) = \mathbf{z}_{ij}^*(k) + \mathbf{e}_{ij}(k) = \mathbf{z}_{ij}^*(k) + \mathbf{n}_{ij}(k), \tag{5.9}$$

where $\mathbf{e}_{ij}(k) = \mathbf{e}_i(k) - \mathbf{e}_j(k)$ is the (relative) tracking error and $\mathbf{n}_{ij}(k) = \mathbf{n}_i(k) - \mathbf{n}_j(k)$ is the zero-mean stochastic tracking error of the relative states.

Theorem 5.1 (*Unbiasedness of relative affine localization*)**.** The relative state estimator given by (5.6) and (5.7) is unbiased given the observation model (5.1) if $k \to \infty$.

Proof. Since there is no operation across time, we drop the time mark k in the notation in this proof for conciseness and clarity. Starting from (5.6) and by incorporating (5.4),

$$\begin{aligned}\hat{\mathbf{\Theta}}_i^T &= (\mathbf{H}_i^T\mathbf{H}_i)^{-1}\mathbf{H}_i^T\mathbf{Y}_i \\ &= (\mathbf{H}_i^T\mathbf{H}_i)^{-1}\mathbf{H}_i^T(\mathbf{H}_i\mathbf{\Theta}_i^T + \mathbf{V}_i) \\ &= \mathbf{\Theta}_i^T + (\mathbf{H}_i^T\mathbf{H}_i)^{-1}\mathbf{H}_i^T\mathbf{V}_i. \end{aligned} \tag{5.10}$$

Then by substituting $\hat{\mathbf{\Theta}}_i$ in (5.7), we have

$$\begin{aligned}\hat{\mathbf{z}}_{ij}^{\text{ral}} &= \hat{\mathbf{\Theta}}_i\mathbf{p}_{ij} \\ &= \mathbf{\Theta}_i\mathbf{p}_{ij} + \mathbf{V}_i^T\mathbf{H}_i(\mathbf{H}_i^T\mathbf{H}_i)^{-1}\mathbf{p}_{ij} \\ &= \mathbf{z}_{ij}^* + \mathbf{V}_i^T\mathbf{q}_{ij}, \end{aligned} \tag{5.11}$$

with $\mathbf{q}_{ij} \triangleq \mathbf{H}_i(\mathbf{H}_i^T\mathbf{H}_i)^{-1}\mathbf{p}_{ij}$. If $k \to \infty$, we assume (5.9) comes into effect and the bias

$$\begin{aligned}\mathbb{E}[\hat{\mathbf{z}}_{ij}^{\text{ral}} - \mathbf{z}_{ij}] &= \mathbb{E}[\mathbf{z}_{ij}^* + \mathbf{V}_i^T\mathbf{q}_{ij} - (\mathbf{z}_{ij}^* + \mathbf{n}_{ij})] \\ &= \mathbb{E}[\mathbf{V}_i^T\mathbf{q}_{ij} - \mathbf{n}_{ij}] \\ &= \mathbf{0}. \end{aligned} \tag{5.12}$$

This is easy to show since

$$\mathbf{V}_i^T\mathbf{q}_{ij} = [\mathbf{v}_{i1}, \mathbf{v}_{i1}, ..., \mathbf{v}_{i|\mathcal{N}_i^k|}]\mathbf{q}_{ij} \tag{5.13}$$

is essentially a linear combination of zero-mean noise terms $\mathbf{v}_{ij}$ (columns of the matrix $\mathbf{V}_i^T$), which is also zero-mean. It is also easy to show that $\mathbf{n}_{ij}$ is zero-mean because if the observation $\mathbf{y}_{ij}$ is zero-mean and the controller (3.2) is linear to the observations, then errors of positions are also zero-mean. ■

The property of unbiasedness is very important because it can typically lead to instability and divergence for autonomous systems if the observations are always biased and there are other means to correct it.

5.2.2 Variance of RAL

The variance of relative affine localization is straightforward. From (5.11), it is obvious that the uncertainties come from the term containing the noise matrix $\mathbf{V}_i$. As such,

$$\begin{aligned}\text{cov}[\hat{\mathbf{z}}_{ij}^{\text{ral}}] &= \text{cov}[\mathbf{z}_{ij}^* + \mathbf{V}_i^T\mathbf{q}_{ij}] \\ &= \text{cov}[\mathbf{V}_i^T\mathbf{q}_{ij}] \\ &= \text{cov}[q_{ij}(1)\mathbf{v}_{i1} + \cdots + q_{ij}(|\mathcal{N}_i^k|)\mathbf{v}_{i|\mathcal{N}_i^k|}], \end{aligned} \tag{5.14}$$

which is the covariance of a linear combination of Gaussian variables. Applying the property of Gaussian distribution and assuming the noises are i.i.d across edges,

$$\begin{aligned}\mathrm{cov}[\hat{\mathbf{z}}_{ij}^{\mathrm{ral}}] &= (q_{ij}^2(1) + \cdots + q_{ij}^2(|\mathcal{N}_i^k|))\mathbf{R} \\ &= (\mathbf{q}_{ij}^T \mathbf{q}_{ij})\mathbf{R}. \end{aligned} \tag{5.15}$$

The steps of relative affine localization with observation noise (parameter tracking involved) are described in Algorithm 5.2 and the convergence is shown in Figure 5.3 together with the variants of RAL which will be introduced in the following sections. As can be expected, the tracking errors are truncated and maintained at a certain noise level post-convergence. Otherwise, similar conclusions to the noiseless case can be drawn. When observation losses occur, the tracking error will be compromised and the engagement of relative affine localization can make up for the performance decrease at a cost of slower convergence.

Algorithm 5.2 *Relative affine localization with observation noise (RAL)*

```
Initialization
    Calculate stress matrix $\mathbf{L}$
    Draw initial positions $\mathbf{z}_i(0) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{P}_0)$
    Globally known nominal positions $\mathbf{p}_i$ for $i \in \mathcal{V}$
    Set noise covariance $\mathbf{R}$
    $k = 0$
while $k \leq K_{max}$ do
    for $i \in \mathcal{V}_f$ do
        Define $\mathcal{N}_i = (\mathcal{N}_i^k, \mathcal{N}_i^m)$
        for $j \in \mathcal{N}_i$ do
            Draw $\mathbf{v}_{ij}(k) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{R})$
            $\mathbf{y}_{ij}(k) = \mathbf{z}_{ij}(k) + \mathbf{v}_{ij}(k)$          ▷ Noisy observation
        end for
        Compute $\hat{\boldsymbol{\Theta}}_i(k)$ based on (5.6)
        Reconstruct $\hat{\mathbf{z}}_{ij}^{\mathrm{ral}}(k)$ based on (5.7) and (5.8)
        $\mathbf{u}_i(k) = -\frac{1}{\gamma_i} \sum_{j \in \mathcal{N}_i} l_{ij}(\hat{\mathbf{z}}_{ij}^{\mathrm{ral}}(k) - \dot{\mathbf{z}}_j(k))$          ▷ Control law
        $\mathbf{z}_i(k+1) = \mathbf{z}_i(k) + \Delta t \mathbf{u}_i(k)$
    end for
    $k = k + 1$
end while
```

5.3 Parameter Tracking in Time Domain

The key to relative affine localization is to estimate the transformation parameters $\boldsymbol{\Theta}(k)$ first. Algorithm 5.2 in the previous sections can be considered estimation with spatial information without considering their temporal correlations. In fact, the parameters, by design, are not abruptly changing but usually present smooth curves subject to some underlying functions. This entails high correlations to the transformation parameters between successive time instances, which makes the exploration in the time domain beneficial. In this section, we present a Least squares-based filter that tracks the evolution of time-varying transformation parameters and presents better performances in the RAL settings.

5.3.1 Data Model and Problem Formulation

For the convenience of derivation, we use a vectorized notation

$$\boldsymbol{\theta} = \text{vec}(\boldsymbol{\Theta}) \tag{5.16}$$

to denote the transformation parameters. Assume that at time instance k, local estimates of affine transformation parameters $\hat{\boldsymbol{\theta}}_i(k)$ using RAL are acquired and presented as

$$\hat{\boldsymbol{\theta}}_i(k) = \boldsymbol{\theta}(k) + \boldsymbol{\epsilon}(k), \tag{5.17}$$

where $\boldsymbol{\theta}(k)$ is the local ground truth (which is also the global ground truth), and $\boldsymbol{\epsilon}(k)$ is the error induced by insufficient convergence. The problem can now be stated as: estimate the true affine transformation parameters given a collection of local estimates across the time series, i.e., estimate $\boldsymbol{\theta}(K-1)$ given $\boldsymbol{\theta}^{\text{ral}}(k)$ for $k = 0, ..., K-1$, where $K-1$ is the "current" time instance.

5.3.2 Adaptive Tracking with Sequential Least Squares

We first assume that the system is static with $\boldsymbol{\theta}(k) = \boldsymbol{\theta}$, for $k = 0, ..., K-1$. Without any statistical modeling on the error $\boldsymbol{\epsilon}(k)$, a Least squares approach is considered in which we minimize the squared error

$$J(\boldsymbol{\theta}) = \sum_{k=0}^{K-1} \left\| \hat{\boldsymbol{\theta}}_i(k) - \boldsymbol{\theta} \right\|_2^2 . \tag{5.18}$$

Differentiating with respect to $\boldsymbol{\theta}$ and setting the result to zero yields

$$\hat{\boldsymbol{\theta}}^{\text{trk}} = \frac{1}{K} \sum_{k=0}^{K-1} \hat{\boldsymbol{\theta}}_i(k), \tag{5.19}$$

which is the sample mean estimator [62].

Now we extend this to the dynamic formation by using the sequential Least squares approach to recursively update the estimation. The problem now becomes that having estimated $\hat{\boldsymbol{\theta}}_i^{\text{trk}}(K-1)$, estimate $\hat{\boldsymbol{\theta}}_i^{\text{trk}}(K)$ with new $\hat{\boldsymbol{\theta}}_i(k)$ obtained from (5.6). The solution is

$$\begin{aligned}\hat{\boldsymbol{\theta}}_i^{\mathrm{trk}}(K) &= \frac{1}{K+1}\sum_{k=0}^{K}\hat{\boldsymbol{\theta}}_i(k)\\ &= \frac{1}{K+1}(\sum_{k=0}^{K-1}\hat{\boldsymbol{\theta}}_i(k)+\hat{\boldsymbol{\theta}}_i(K))\\ &= \frac{K}{K+1}\hat{\boldsymbol{\theta}}_i^{\mathrm{trk}}(K-1)+\frac{1}{K+1}\hat{\boldsymbol{\theta}}_i(K) && (5.20a)\\ &= \hat{\boldsymbol{\theta}}_i^{\mathrm{trk}}(K-1)+\frac{1}{K+1}(\hat{\boldsymbol{\theta}}_i(K)-\hat{\boldsymbol{\theta}}_i^{\mathrm{trk}}(K-1)), && (5.20b)\end{aligned}$$

where (5.20a) suggests that this filter is taking a weighted average between the past estimate and the current input, and the weight for the current input is proportionally decreasing as time progresses since the previous estimate is based on more samples. On the other hand, (5.20b) indicates that the filter is adding a correction term on top of the previous estimate. $\hat{\boldsymbol{\theta}}_i(K)-\hat{\boldsymbol{\theta}}_i^{\mathrm{trk}}(K-1)$ can also be regarded as an error in predicting $\hat{\boldsymbol{\theta}}_i^{\mathrm{trk}}(K)$ with the previous estimator. If the previous estimator is sufficiently accurate and the system is time-invariant, the correction term will converge to zero. Otherwise, the estimator will keep being updated which serves the purpose of adaptive filtering in time-varying systems. As a generalization, we modify (5.20b) and give a update equation

$$\hat{\boldsymbol{\theta}}_i^{\mathrm{trk}}(k) = \hat{\boldsymbol{\theta}}_i^{\mathrm{trk}}(k-1)+\mu(\hat{\boldsymbol{\theta}}_i(k)-\hat{\boldsymbol{\theta}}_i^{\mathrm{trk}}(k-1)), \tag{5.21}$$

where μ is a small coefficient to set constant weights to the correction term for better tracking of time-varying parameters. It should be noted that since only linear operations are involved in (5.22), direct computation on the unvectorized version $\boldsymbol{\Theta}$ is possible,

$$\hat{\boldsymbol{\Theta}}_i^{\mathrm{trk}}(k) = \hat{\boldsymbol{\Theta}}_i^{\mathrm{trk}}(k-1)+\mu(\hat{\boldsymbol{\Theta}}_i(k)-\hat{\boldsymbol{\Theta}}_i^{\mathrm{trk}}(k-1)). \tag{5.22}$$

Finally, the local RAL estimates of missing observations are again

$$\mathbf{z}_{ij}^{\mathrm{trk}}(k) = \hat{\boldsymbol{\Theta}}_i^{\mathrm{trk}}(k)\mathbf{p}_{ij} \quad j\in\mathcal{N}_i^m, \tag{5.23}$$

and $\mathbf{z}_{ij}^{\mathrm{trk}}(k)=\mathbf{y}_{ij}(k)$ for $j\in\mathcal{N}_i^k$ as before.

The choice of μ should be application dependent. For fast maneuvering systems where there are lots of rotations and scaling involved and are traveling with high speed, μ should be larger to capture the fast change of parameters. For slow maneuvering systems where translation or low-speed maneuvers are dominant, μ could be smaller. In the extreme case where the system is static, μ could be chosen as $1/(k+1)$ as a decaying weight and get optimal filtering in a Least squares sense.

Algorithm 5.3 describes the steps of performing this filter in RAL settings. To distinguish this alternation, this method is marked RAL-TRK for ”tracking”. Since the algorithm employs previous estimates, the initial value for $\hat{\boldsymbol{\Theta}}_i^{\mathrm{trk}}(0)$ can be set as $\mathbf{I}_D$. When the hyperparameter μ is taken as 0.1, the simulation result using this filter is presented in Figure 5.3, where the RAL-TRK case exhibits clear improvements in the convergence speed compared to the RAL case. A potential reason is the subtraction in (5.22) cancels out some bias between two successive estimates of the transformation parameter and thus results in faster convergence. From another perspective, historical information is taken into account when given an estimate, thus the robustness is enhanced. Although the speed of convergence is improved using the proposed method, the trajectories are still noisy under the impact of observation noises, which will be addressed in the following section.

Algorithm 5.3 *Relative affine localization with parameter tracking (RAL-TRK)*

```
Initialization
    Calculate stress matrix $\mathbf{L}$
    Draw initial positions $\mathbf{z}_i(0) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{P}_0)$
    Globally known nominal positions $\mathbf{p}_i$ for $i \in \mathcal{V}$
    Set noise covariance $\mathbf{R}$
    $\hat{\boldsymbol{\Theta}}_i^{\text{trk}}(0) = \mathbf{I}_D$ and set $\mu$                ▷ Initialize local parameters
    $k = 0$
while $k \leq K_{max}$ do
    for $i \in \mathcal{V}_f$ do
        Define $\mathcal{N}_i = (\mathcal{N}_i^k, \mathcal{N}_i^m)$
        for $j \in \mathcal{N}_i$ do
            Draw $\mathbf{v}_{ij}(k) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{R})$
            $\mathbf{y}_{ij}(k) = \mathbf{z}_{ij}(k) + \mathbf{v}_{ij}(k)$                ▷ Noisy observation
        end for
        Compute $\hat{\boldsymbol{\Theta}}_i(k)$ based on (5.6)
        $\hat{\boldsymbol{\Theta}}_i^{\text{trk}}(k) = \hat{\boldsymbol{\Theta}}_i^{\text{trk}}(k-1) + \mu(\hat{\boldsymbol{\Theta}}_i(k) - \hat{\boldsymbol{\Theta}}_i^{\text{trk}}(k-1))$                ▷ Parameter tracking
        Reconstruct $\mathbf{z}_{ij}^{\text{trk}}(k)$ based on (5.23)
        $\mathbf{u}_i(k) = -\frac{1}{\gamma_i} \sum_{j \in \mathcal{N}_i} l_{ij}(\mathbf{z}_{ij}^{\text{trk}}(k) - \dot{\mathbf{z}}_j(k))$                ▷ Control law
        $\mathbf{z}_i(k+1) = \mathbf{z}_i(k) + \Delta t \mathbf{u}_i(k)$
    end for
    $k = k + 1$
end while
```

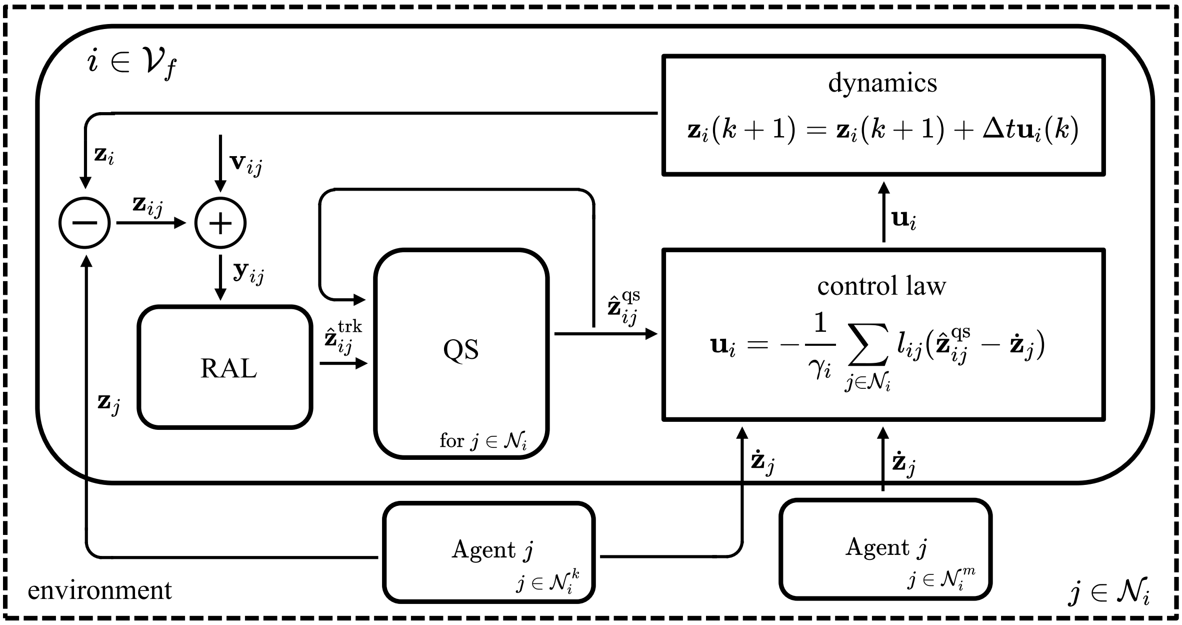
5.4 Quadratic Smoothing

5.4.1 Motivation

As seen in the previous section, the tracking error is bounded by the observation noise and it is fluctuating. This implies that the trajectories of agents are not smooth and subject to unnecessary ”zig-zags” which is undesired in practice. A typical way to reduce the mean and variance of the error is to introduce multiple parallel observations of one relative state. If the observation noises of them are i.i.d., some estimators such as the Maximum likelihood estimator (MLE) can be tailored to accomplish the task [51]. However, in the setup of this thesis, it would be impractical to assume parallel observations due to e.g. limited onboard resources that cause observation losses in the first place. Fortunately, we could still rely on temporal information to reduce the variance. Note that RAL-TRK also uses temporal information but on the parameter level $\mathbf{\Theta}(k)$. Since it is a linear filter, the noise variance can not be reduced. In this section, we introduce a nonlinear method using quadratic reconstruction and smoothing (QS) [63]. In some literature, it is also called Whittaker-Eilers Smoother [64].

The smoothing will be performed on the trajectories with RAL-TRK employed. Although the agents are moving in $\mathbb{R}^D$, we treat the trajectories of each dimension as separate 1D signals. Note agents perform the smoothing on the (relative) trajectories $\mathbf{z}_{ij}(k)$ for all k, which is the trajectory of neighbors observed by agent i.

5.4.2 Formulation



$i \in \mathcal{V}_f$
dynamics
$\mathbf{z}_i(k+1) = \mathbf{z}_i(k+1) + \Delta t\mathbf{u}_i(k)$
$\mathbf{z}_i$
$\mathbf{v}_{ij}$
$-$
$\mathbf{z}_{ij}$
$+$
$\mathbf{y}_{ij}$
$\mathbf{u}_i$
control law
$\mathbf{u}_i = -\frac{1}{\gamma_i}\sum_{j\in\mathcal{N}_i} l_{ij}(\hat{\mathbf{z}}_{ij}^{\text{qs}} - \dot{\mathbf{z}}_j)$
$\hat{\mathbf{z}}_{ij}^{\text{qs}}$
RAL
$\hat{\mathbf{z}}_{ij}^{\text{trk}}$
QS
for $j \in \mathcal{N}_i$
$\mathbf{z}_j$
$\dot{\mathbf{z}}_j$
$\dot{\mathbf{z}}_j$
Agent j
$j \in \mathcal{N}_i^k$
Agent j
$j \in \mathcal{N}_i^m$
environment
$j \in \mathcal{N}_i$

Figure 5.2: An overview of the RAL framework with quadratic smoothing implemented.

Now we lay the formulation of the smoother and an overview is shown in Figure 5.2. The observation model is still assumed to be (5.1), and since the smoothing is done in time series, we define a window length W_l to limit the number of past samples involved in the computation. We first define a reconstructed signal matrix for every $i \in \mathcal{V}_f, j \in \mathcal{N}_i$ as

$$\hat{\mathbf{Z}}_{ij}^{\text{qs}}(k) = \begin{bmatrix} \hat{\mathbf{z}}_{ij}^{\text{qs}}(k - W_l + 1)^T \\ \vdots \\ \hat{\mathbf{z}}_{ij}^{\text{qs}}(k)^T \end{bmatrix} \in \mathbb{R}^{W_l \times D}, \tag{5.24}$$

whose columns are denoted by $\hat{\mathbf{z}}_{ij}^d \in \mathbb{R}^{W_l}$ for $d = 1, ..., D$. Correspondingly, a signal matrix containing past estimates and current observations is defined as

$$\tilde{\mathbf{Y}}_{ij}(k) = \begin{bmatrix} \hat{\mathbf{z}}_{ij}^{\text{qs}}(k - W_l + 1)^T \\ \vdots \\ \hat{\mathbf{z}}_{ij}^{\text{qs}}(k-1)^T \\ \tilde{\mathbf{y}}_{ij}(k)^T \end{bmatrix} \in \mathbb{R}^{W_l \times D}, \tag{5.25}$$

where $\tilde{\mathbf{y}}_{ij}(k)$ is

$$\tilde{\mathbf{y}}_{ij}(k) = \begin{cases} \mathbf{y}_{ij}(k), & \text{if } j \in \mathcal{N}_i^k \\ \hat{\mathbf{z}}_{ij}^{\text{trk}}(k), & \text{if } j \in \mathcal{N}_i^m \end{cases}, \tag{5.26}$$

in which we use $\tilde{\mathbf{y}}_{ij}^d \in \mathbb{R}^{W_l}$ to denote the d-th column of $\tilde{\mathbf{Y}}_{ij}(k)$ that can be regarded as observation vector of the d-th dimension of the trajectory. (5.26) indicates that we use the available observation as it is but uses RAL-TRK estimates as an observation for the missing relative states. The structure of (5.25) suggests that the past smoothed signals are reused in the current smoothing, which in some sense is also a recursive method.

Quadratic smoothing is essentially a regularized Least square fitting in which a quadratic smoothing function serves to penalize the variations. Here we adopt the first-order difference in the quadratic function

$$J_{\text{quad}}(\hat{\mathbf{z}}_{ij}^d) = \left\| \mathbf{W} \hat{\mathbf{z}}_{ij}^d \right\|_2^2, \tag{5.27}$$

where $\mathbf{W} \in \mathbb{R}^{W_l - 1 \times W_l}$ is a sparse bidiagonal matrix

$$\mathbf{W} = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix}. \tag{5.28}$$

As such, the cost function of quadratic smoothing is

$$J_{\text{qs}}(\hat{\mathbf{z}}_{ij}^d) = \left\| \hat{\mathbf{z}}_{ij}^d - \tilde{\mathbf{y}}_{ij}^d \right\|_2^2 + \nu \left\| \mathbf{W} \hat{\mathbf{z}}_{ij}^d \right\|_2^2, \tag{5.29}$$

where $\nu \geq 0$ parameterizes the trade-off of the Least square term and the regularization term. Analytical solutions can be derived for this formulation as

$$\hat{\mathbf{z}}_{ij}^d = (\mathbf{I} + \nu \mathbf{W}^T \mathbf{W})^{-1} \tilde{\mathbf{y}}_{ij}^d \quad d = 1, ..., D, \tag{5.30}$$

with which we could aggregate D dimensions and filter altogether in the equivalent form

$$\hat{\mathbf{Z}}_{ij}^{\text{qs}}(k) = (\mathbf{I} + \nu \mathbf{W}^T \mathbf{W})^{-1} \tilde{\mathbf{Y}}_{ij}(k). \tag{5.31}$$

Note that (5.31) is numerically efficient since $\mathbf{I} + \nu \mathbf{W}^T \mathbf{W}$ is a sparse tridiagonal matrix and solvers such as sparse Cholesky solver [65] or sparse LU-based solver [66] are available. Besides, if W_l and ν are decided to be fixed, the inversion could be calculated at initialization. Hence a wider range of selection for the window length W_l is possible without being too computationally expensive.

There are a few comments regarding the formulation (5.29). As is suggested in the equation, the Least squares term ensures that the reconstructed trajectory is as close to the observation as possible. If $\nu = 0$ or very small, there will be no smoothing effect and the reconstructed trajectory should come out identical to the observation. On the other hand, if ν is very large, the variation penalty is dominant in the cost, and the resulting trajectory will be very smooth. However, this also jeopardizes the unbiasedness of the estimates. From a statistics point of view, ν balances the bias and the variance of the smoother. Existing literature such as [67] that studies the choice of ν usually deals with static systems (e.g. images, point clouds, etc.) and is less sensitive to the induced bias. As such, as of now, there is yet an optimal tuning method for this formation control system and ν should be heuristically chosen based on specific applications. Algorithm 5.4 shows the steps to perform quadratic smoothing (RAL-QS) under relative affine localization and distributed formation control settings.

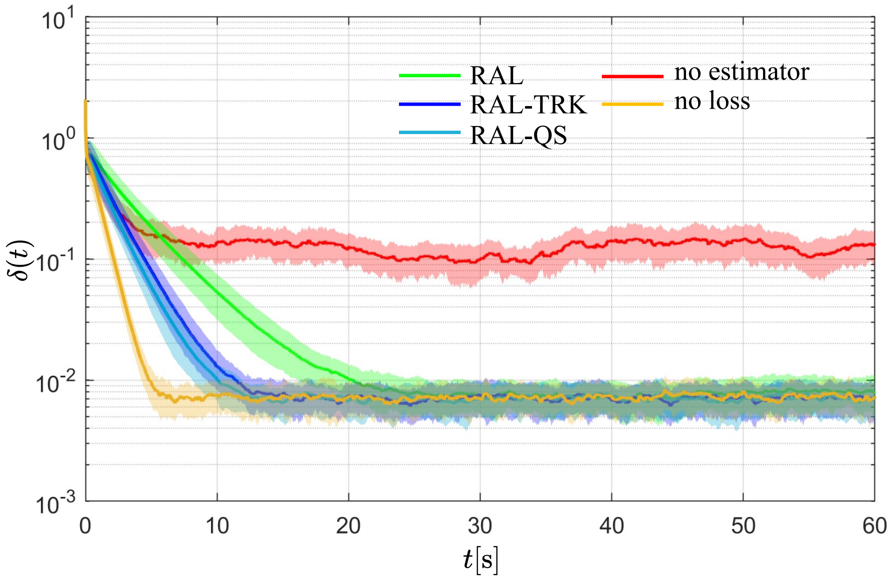
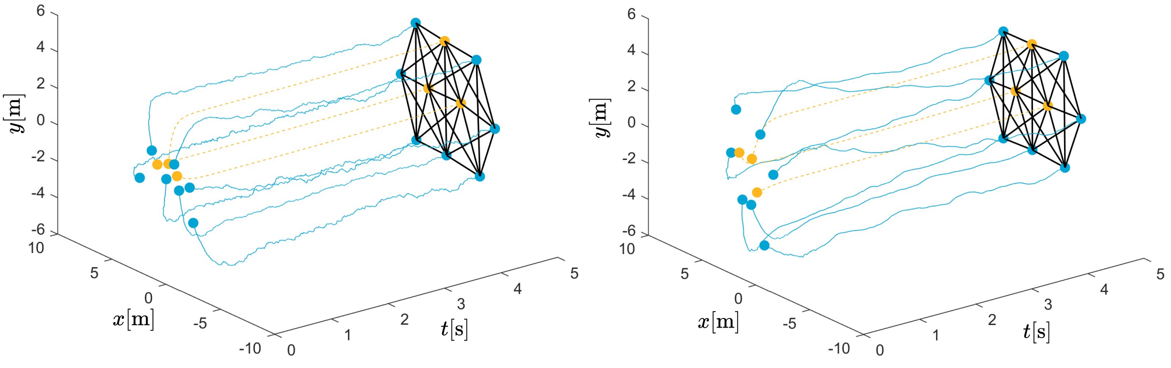


Figure 5.3: The simulation results of proposed RAL-based methods. The noise covariance is chosen as $\sigma_v^2\mathbf{I}$ with $\sigma_v = 0.1$. The number of known observations per agent is chosen as $\left|\mathcal{N}_i^k\right| = 3$. The references for each method are: RAL (Algorithm 5.2), RAL-TRK (Algorithm 5.3), and RAL-QS (Algorithm 5.4).

As the previous convergence simulations that use the setup introduced in Section 3.5, the simulation results of RAL-TRK are shown in Figure 5.3 and Figure 5.4 in comparison with the previously proposed methods including RAL and RAL-TRK. It is clear that with quadratic smoothing applied, the tracking error is not compromised but the trajectories are smoothed. The hyperparameters chosen for this simulation are $W_l = 500$ and $\nu = 100$ although they are not necessarily optimal need to search for better choices.

Algorithm 5.4 *Relative affine localization with quadratic smoothing (RAL-QS)*

```
Initialization
    Calculate stress matrix $\mathbf{L}$
    Draw initial positions $\mathbf{z}_i(0) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{P}_0)$
    Globally known nominal positions $\mathbf{p}_i$ for $i \in \mathcal{V}$
    Set noise covariance $\mathbf{R}$
    $\hat{\mathbf{\Theta}}_i^{\text{trk}}(0) = \mathbf{I}_D$ and set $\mu$
    Set $W_l$ and $\nu$
    Construct $\mathbf{W}$ and calculate $(\mathbf{I} + \nu \mathbf{W}^T \mathbf{W})^{-1}$
    $k = 0$
while $k \leq K_{max}$ do
    for $i \in \mathcal{V}_f$ do
        Define $\mathcal{N}_i = (\mathcal{N}_i^k, \mathcal{N}_i^m)$
        for $j \in \mathcal{N}_i$ do
            Draw $\mathbf{v}_{ij}(k) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{R})$
            $\mathbf{y}_{ij}(k) = \mathbf{z}_{ij}(k) + \mathbf{v}_{ij}(k)$                              ▷ Noisy observation
        end for
        Compute $\hat{\mathbf{\Theta}}_i(k)$ based on (5.6)
        $\hat{\mathbf{\Theta}}_i^{\text{trk}}(k) = \hat{\mathbf{\Theta}}_i^{\text{trk}}(k-1) + \mu(\hat{\mathbf{\Theta}}_i(k) - \hat{\mathbf{\Theta}}_i^{\text{trk}}(k-1))$      ▷ Parameter tracking
        Reconstruct $\mathbf{z}_{ij}^{\text{trk}}(k)$ based on (5.23)
        Construct $\tilde{\mathbf{Y}}_{ij}(k)$ based on (5.25)
        $\hat{\mathbf{Z}}_{ij}^{\text{qs}}(k) = (\mathbf{I} + \nu \mathbf{W}^T \mathbf{W})^{-1} \tilde{\mathbf{Y}}_{ij}(k)$          ▷ Quadratic smoothing
        $\mathbf{u}_i(k) = -\frac{1}{\gamma_i} \sum_{j \in \mathcal{N}_i} l_{ij}(\hat{\mathbf{z}}_{ij}^{\text{qs}}(k) - \dot{\mathbf{z}}_j(k))$          ▷ Control law
        $\mathbf{z}_i(k+1) = \mathbf{z}_i(k) + \Delta t \mathbf{u}_i(k)$
    end for
    $k = k + 1$
end while
```



(a) (b)

Figure 5.4: Two trajectories for the first 5 seconds with (a) using RAL-TRK and (b) RAL-QS applied. Simulation parameters are $\left|\mathcal{N}_i^k\right| = 3$, and to clearly show the smoothing effect, $\sigma_v = 1$ is chosen for both figures.

5.5 Summary

In this section, we put the system in a more realistic, namely noisy environment and build filtering techniques on top of the relative affine localization framework from the previous chapter. Some notable contributions can be highlighted as

- The influence of the observation noise is numerically shown with simulations.
- Theoretical investigation on the statistical property of relative affine localization as an estimator.
- A Least squares-based parameter tracking technique is introduced with improvements in the speed of convergence (RAL-TRK).
- A data-driven smoother/filter based on the principle of quadratic smoothing (RAL-QS) is introduced showing less noisy trajectories.

So far we have established a framework to tackle random missing observations and observation noise using the relative affine localization technique. However, RAL and its related methods are limited by the geometrical feasibility and suffer from a slower convergence speed. In the next chapter, an additional kinematics model will be established for agents, and the fusion of the kinematics model, available observations, and RAL estimates will be carried out using a Kalman filter framework.

Model Based Prediction and Filtering 6

The methods to tackle missing observations and smoothing in the previous chapters are built upon relative affine localization which is fundamentally limited by its geometry feasibility. We call it a *RAL-denied* situation if this constraint can not be satisfied. As such, other sources of estimating the missing state observation in RAL denied situations are needed and we turn to the relative kinematics model of these agents. In this chapter, we perform simultaneous missing state estimation and smoothing in noisy environments under a Kalman Filter framework. First, a general kinematics model for the agents is established without any assumption about the agents' true dynamics. A complete state-space model can then be formed by incorporating the observation model in the previous chapter. Next, a series of Kalman filters will be tailored to accommodate the need for smoothing and state estimation. Finally, we present a RAL-KF fusion scheme that exhibits superior performance.

6.1 Relative State-Space Model

In this section, we introduce a state-space model on agents to pave the way for the Kalman filters. Since the relative positions are of interest for the distributed controllers, the state-space model is built on the relative kinematics of the agents, hence is called the relative state-space model. Here we adopt a constant acceleration model to approximate the kinematics of agents since this model is very general with minimum assumptions and is independent of the dynamics in the control framework. As such, it can be easily adapted to other formation control frameworks. Some other motivations for this choice are:

- A constant acceleration model is a generalization of the constant velocity model which could capture finer details from the movement.
- If the true dynamics of agents are acceleration-varying which is usually the case, the constant acceleration model is still able to approximate well if the time step Δt is small.

As such, we define the extended (relative) state vector $\mathbf{z}'_{ij} \in \mathbb{R}^{3D}$ as

$$\mathbf{z}'_{ij}(k) = \mathrm{vec}([\mathbf{z}_{ij}(k), \dot{\mathbf{z}}_{ij}(k), \ddot{\mathbf{z}}_{ij}(k)]^T), \tag{6.1}$$

where $\dot{\mathbf{z}}_{ij}(k)$ and $\ddot{\mathbf{z}}_{ij}(k)$ are the (relative) velocity and acceleration, respectively. An example of (6.1) when $D = 2$ is

$$\mathbf{z}'_{ij}(k) = \begin{bmatrix} z^x_{ij}(k) \\ \dot{z}^x_{ij}(k) \\ \ddot{z}^x_{ij}(k) \\ z^y_{ij}(k) \\ \dot{z}^y_{ij}(k) \\ \ddot{z}^y_{ij}(k) \end{bmatrix}, \tag{6.2}$$

where x and y are the two dimensions.

6.1.1 Observation Model

Assuming that agents carry homogeneous sensors, the observation noise is also i.i.d. if there is no cross-agent correlation. Same as the RAL settings, we assume again that agents can access at most 1 observation of each edge state per control interval. Recall the noise modeling on the observation in (5.1), which is restated here:

$$\mathbf{y}_{ij}(k) = \mathbf{z}_{ij}(k) + \mathbf{v}_{ij}(k), \tag{6.3}$$

where $\mathbf{v}_{ij}(k) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}, \mathbf{R})$ is the observation noise. Then the observation equation for the edge state-space model is

$$\mathbf{y}_{ij}(k) = \mathbf{G}\mathbf{z}'_{ij}(k) + \mathbf{v}_{ij}(k), \tag{6.4}$$

where $\mathbf{G} \in \mathbb{R}^{D\times 3D}$ is the observation matrix and has the following structure

$$\mathbf{G} = \mathbf{I}_D \otimes [1\ 0\ 0], \tag{6.5}$$

which is essentially selecting the (relative) positions as observation and implying that other states are not observable.

6.1.2 State Transition Model

The state transition model can be then written as

$$\mathbf{z}'_{ij}(k+1) = \mathbf{F}\mathbf{z}'_{ij}(k) + \mathbf{w}_{ij}(k), \tag{6.6}$$

where the state transition matrix $\mathbf{F} \in \mathbb{R}^{3D\times 3D}$ is

$$\mathbf{F} = \mathbf{I}_D \otimes \mathbf{F}^d = \mathbf{I}_D \otimes \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \tag{6.7}$$

and $\mathbf{w}_{ij}(k) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}, \mathbf{Q})$ is the process noise.

The process noise $\mathbf{w}_{ij}(k)$ is added to approximate the inaccuracy of the state transition model caused by assuming constant acceleration. The covariance matrix $\mathbf{Q}$ that characterizes this uncertainty can be represented by a block matrix e.g., for $D = 2$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}^x & \mathbf{Q}^{xy} \\ \mathbf{Q}^{yx} & \mathbf{Q}^y \end{bmatrix}. \tag{6.8}$$

Assuming the kinematics are uncorrelated across dimensions, $\mathbf{Q}$ will have a block diagonal structure, i.e., $\mathbf{Q} = \mathrm{bdiag}\{\mathbf{Q}^d\}_{d=1,\ldots,D}$. Given that the error of acceleration induces errors in velocity and position, $\mathbf{Q}^d$ should also be structured. Let $\mathbf{Q}_a^d \in \mathbb{R}^{3\times 3}$ denote the covariance matrix of a small error in acceleration in dimension d which has a structure

$$\mathbf{Q}_a^d = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_w^2 \end{bmatrix} = \sigma_w^2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{6.9}$$

we can then project this error covariance to $\mathbf{Q}^d$ by

$$\mathbf{Q}^d = \mathbf{F}^d \mathbf{Q}_a^d \mathbf{F}^{d^T} = \sigma_w^2 \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 & \Delta t \\ \frac{\Delta t^2}{2} & \Delta t & 1 \end{bmatrix}. \tag{6.10}$$

Assuming identical noise covariance for each dimension, the final covariance matrix $\mathbf{Q}$ has the structure

$$\mathbf{Q} = \mathbf{I}_D \otimes \mathbf{Q}^d = \sigma_w^2 \left(\mathbf{I}_D \otimes \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 & \Delta t \\ \frac{\Delta t^2}{2} & \Delta t & 1 \end{bmatrix} \right), \tag{6.11}$$

where σ_w^2 is the variance of acceleration and can be thus regarded as the process noise variance.

6.2 Relative Kalman Filter

In the case where there are no missing observations, a relative Kalman filter (RKF) can be derived based on the relative state-space model. Kalman filters under affine formation control frameworks based on the relative kinematics are studied in [51]. The major difference between this work and the Kalman filters in [51] is the state transition model, where we take the general kinematics of an autonomous agent whereas [51] uses the exact dynamics (3.1) by which the agents are governed for the formation control framework. Using the exact dynamics is advantageous in predicting the states but it is not generalizable to other systems and strong requirements for the communication network is needed.

RKF initialization

Without any prior knowledge of the relative velocity and the acceleration, we simply take zeros for their initial states. Since the spawn locations $\mathbf{z}_i(0)$ for the agents are random and subject to $\boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{P}_0)$, we can derive that

$$\hat{\mathbf{z}}_{ij}^{\mathrm{rkf}}(0|0) = \mathbb{E}[\mathbf{z}_{ij}(0)] = \mathbb{E}[\mathbf{z}_i(0) - \mathbf{z}_j(0)] = \mathbf{0}_D. \tag{6.12}$$

As such, we set the initial state vector to be zero, that is

$$\hat{\mathbf{z}}'_{ij}(0|0) = \mathbf{0}_{3D}. \tag{6.13}$$

For the initial estimate covariance $\mathbf{\Sigma}_{ij}(0|0)$, the values should be large for the (relative) velocities and accelerations since no prior knowledge is given. But for the relative positions, we can simply take $2\mathbf{P}_0$ since $\mathbf{z}_{ij}(0) = \mathbf{z}_i(0) - \mathbf{z}_j(0)$. If taking the setup introduced in Section 3.5 where $\mathbf{P}_0 = 2\mathbf{I}_D$ and assume the same level of confidence for (relative) velocities and accelerations, we take the initial estimate covariance as

$$\mathbf{\Sigma}_{ij}(0|0) = 4\mathbf{I}_{3D} \tag{6.14}$$

This initialization is rather heuristic and can also be modeled in other ways but the initialization has little impact on the convergence of the Kalman filter and the formation.

RKF prediction stage

The prediction stage is where the system extrapolates the next states based on the state transition model. The estimate covariance will also be predicted based on $\mathbf{Q}$, the confidence of this model. The prediction is conducted as follows

$$\hat{\mathbf{z}}'_{ij}(k|k-1) = \mathbf{F}\hat{\mathbf{z}}'_{ij}(k-1|k-1) \tag{6.15a}$$

$$\mathbf{\Sigma}_{ij}(k|k-1) = \mathbf{F}\mathbf{\Sigma}_{ij}(k-1|k-1)\mathbf{F}^T + \mathbf{Q} \tag{6.15b}$$

for $i \in \mathcal{V}_f, j \in \mathcal{N}_i$. This estimate is given without any input from the observation, and the final estimate will be calculated in the correction stage by taking a weighted average between this estimate and the observations. This prediction is also important in this context since if no observations are available, we can only count on this prediction.

RKF correction stage

The correction stage is where the filter takes a weighted average of the prediction and observation. The weight is calculated based on their confidence or uncertainty and it is named the Kalman gain $\mathbf{K}_{ij}(k)$. The correction steps are performed as

$$\mathbf{K}_{ij}(k) = \mathbf{\Sigma}_{ij}(k|k-1)\mathbf{G}^T(\mathbf{G}\mathbf{\Sigma}_{ij}(k|k-1)\mathbf{G}^T + \mathbf{R})^{-1} \tag{6.16a}$$

$$\hat{\mathbf{z}}'_{ij}(k|k) = \hat{\mathbf{z}}'_{ij}(k|k-1) + \mathbf{K}_{ij}(k)(\mathbf{y}_{ij}(k) - \mathbf{G}\hat{\mathbf{z}}'_{ij}(k|k-1)) \tag{6.16b}$$

$$\mathbf{\Sigma}_{ij}(k|k) = (\mathbf{I}_{3D} - \mathbf{K}_{ij}(k)\mathbf{G})\mathbf{\Sigma}_{ij}(k|k-1) \tag{6.16c}$$

for $i \in \mathcal{V}_f, j \in \mathcal{N}_i$. After the correction of relative states and covariances, we can extract the position estimates for the controller (3.2) using

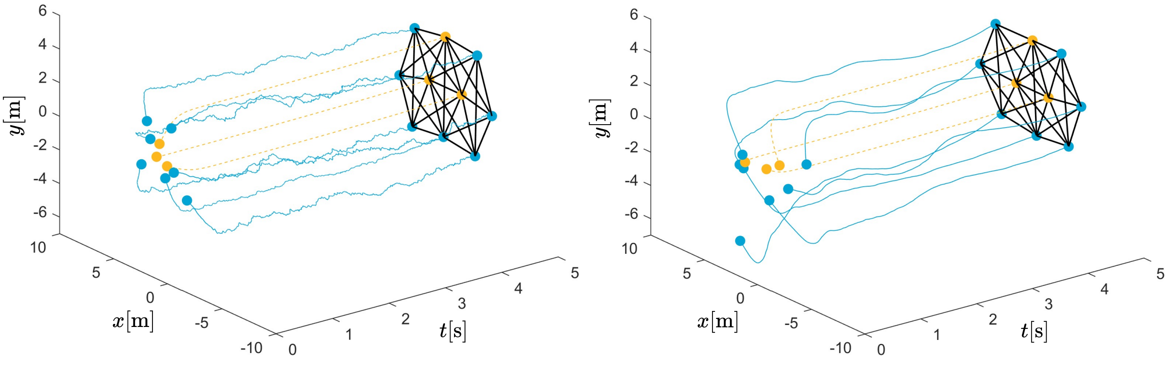
$$\hat{\mathbf{z}}^{\text{rkf}}_{ij}(k) = \mathbf{G}\hat{\mathbf{z}}'_{ij}(k|k). \tag{6.17}$$

Note that only (6.16b) in the two alternating stages is observation dependent, hence the other steps can be precalculated and wait for the arrival of observations.

The pseudocode to perform RKF in affine formation control is described in Algorithm 6.1. The smoothing effect is shown in Figure 6.1 where the trajectories are less noisy after filtering. The convergence of RKF is shown in Figure 6.2 in comparison with the variant of RKF for observation losses in the next section.

Algorithm 6.1 *Distributed formation control with relative Kalman filter (RKF)*

```
Initialization
    Calculate stress matrix $\mathbf{L}$
    Draw initial positions $\mathbf{z}_i(0) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{P}_0)$
    Set noise covariance $\mathbf{R}$ and $\mathbf{Q}$
    $\hat{\mathbf{z}}'_{ij}(0|0) = \mathbf{0}_{3D}$                                          ▷ Initialize the state
    $\boldsymbol{\Sigma}_{ij}(0|0) = 4\mathbf{I}_{3D}$                                     ▷ Initialize the covariance
    $k = 0$
while $k \leq K_{max}$ do
    for $i \in \mathcal{V}_f$ do
        for $j \in \mathcal{N}_i$ do
            $\hat{\mathbf{z}}'_{ij}(k|k-1) = \mathbf{F}\hat{\mathbf{z}}'_{ij}(k-1|k-1)$                  ▷ Prediction
            $\boldsymbol{\Sigma}_{ij}(k|k-1) = \mathbf{F}\boldsymbol{\Sigma}_{ij}(k-1|k-1)\mathbf{F}^T + \mathbf{Q}$
            Draw $\mathbf{v}_{ij}(k) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{R})$
            $\mathbf{y}_{ij}(k) = \mathbf{z}_{ij}(k) + \mathbf{v}_{ij}(k)$                       ▷ Noisy observation
            $\mathbf{K}_{ij}(k) = \boldsymbol{\Sigma}_{ij}(k|k-1)\mathbf{G}^T(\mathbf{G}\boldsymbol{\Sigma}_{ij}(k|k-1)\mathbf{G}^T + \mathbf{R})^{-1}$
            $\hat{\mathbf{z}}'_{ij}(k|k) = \hat{\mathbf{z}}'_{ij}(k|k-1) + \mathbf{K}_{ij}(k)(\mathbf{y}_{ij}(k) - \mathbf{G}\hat{\mathbf{z}}'_{ij}(k|k-1))$    ▷ Correction
            $\boldsymbol{\Sigma}_{ij}(k|k) = (\mathbf{I}_{3D} - \mathbf{K}_{ij}(k)\mathbf{G})\boldsymbol{\Sigma}_{ij}(k|k-1)$
            $\hat{\mathbf{z}}^{\mathrm{rkf}}_{ij}(k) = \mathbf{G}\hat{\mathbf{z}}'_{ij}(k|k)$                ▷ Extract position estimate
        end for
        $\mathbf{u}_i(k) = -\frac{1}{\gamma_i}\sum_{j \in \mathcal{N}_i} l_{ij}(\hat{\mathbf{z}}^{\mathrm{rkf}}_{ij}(k) - \dot{\mathbf{z}}_j(k))$      ▷ Control law
        $\mathbf{z}_i(k+1) = \mathbf{z}_i(k) + \Delta t\mathbf{u}_i(k)$
    end for
    $k = k + 1$
end while
```



(a)

(b)

Figure 6.1: Two trajectories with no observation losses for the first 5 seconds with (a) no RKF implemented and (b) RKF implemented. To clearly show the smoothing effect, $\sigma_v = 1$ is chosen for both figures.

6.3 Relative Kalman Filter with Intermittent Observations

In the event of random observation losses, an estimate of the relative state has to be computed. A binary random variable $\beta_{ij}(k)$ is used to characterize the availability of observations, i.e., $\beta_{ij}(k) = 1$ if $\mathbf{y}_{ij}(k)$ is available and $\beta_{ij}(k) = 0$ if $\mathbf{y}_{ij}(k)$ is missing. A heuristic approach to approximate the missing observations by using previous estimates is

$$\hat{\mathbf{z}}_{ij}(k) = \begin{cases} \hat{\mathbf{z}}_{ij}(k-1), & \text{if } \beta_{ij}(k) = 0 \\ \mathbf{y}_{ij}(k), & \text{if } \beta_{ij}(k) = 1 \end{cases}. \tag{6.18}$$

Intuitively, this approach should perform better than using zeros for the missing observations as described in Algorithm 4.1 if the majority of observations are available. This approach is adopted in many distributed consensus algorithms in event of loss of data or asynchronousness. As such, we take this approach as a comparison to our methods using the Kalman filter.

With the dynamics modeled for the agents in (6.6), it is natural to turn to this model when there are no observations. In the field of navigation, this is sometimes referred to as *dead reckoning* [68, 69] which is a technique to infer the location of an object using accurately measured velocity, heading, etc. However, this kind of method requires precise modeling of the dynamics as well as up-to-date and accurate measurements to provide good estimates. These conditions are not satisfied with the state-space model in Section 6.1 since the dynamic model is an approximate model and the observations only contain positions. As such, using (6.6) to predict relative positions for long periods is infeasible and at least a certain amount of observations is needed to correct and update the model.

A Kalman filter with intermittent observations is inspired [37, 38] based on the above reasoning. The initialization and prediction stages for RKF with intermittent observations (RKF-IO) are identical to those in Section 6.2, only the correction stage is different in presentation and is shown as follows.

RKFIO correction stage

$$\mathbf{K}_{ij}(k) = \mathbf{\Sigma}_{ij}(k|k-1)\mathbf{G}^T(\mathbf{G}\mathbf{\Sigma}_{ij}(k|k-1)\mathbf{G}^T + \mathbf{R})^{-1} \tag{6.19a}$$

$$\hat{\mathbf{z}}'_{ij}(k|k) = \hat{\mathbf{z}}'_{ij}(k|k-1) + \beta_{ij}(k)\mathbf{K}_{ij}(k)(\mathbf{y}_{ij}(k) - \mathbf{G}\hat{\mathbf{z}}'_{ij}(k|k-1)) \tag{6.19b}$$

$$\mathbf{\Sigma}_{ij}(k|k) = (\mathbf{I}_{3D} - \beta_{ij}(k)\mathbf{K}_{ij}(k)\mathbf{G})\mathbf{\Sigma}_{ij}(k|k-1) \tag{6.19c}$$

with $\beta_{ij}(k)$ being either 0 or 1 for $i \in \mathcal{V}_f, j \in \mathcal{N}_i$. It is obvious that the correction for the state and the covariance prediction is identical to (6.16b) and (6.16c) if $\beta_{ij}(k) = 1$, and the state and the covariance are taken as predicted if $\beta_{ij}(k) = 0$.

In the Kalman filter with intermittent observation literature e.g. [37], the focus was on mathematically showing that there exists a critical value λ_c on the probability of $\beta_{ij}(k) = 1$. If $\lambda > \lambda_c$ the Kalman filter has a guaranteed convergence, and if otherwise, the system diverges under some initial conditions. In this thesis, we omit the details on the theoretical derivations of this critical value and use numerical simulations to show the performance under the entire formation control framework. The steps to perform RKF with intermittent observations are described in Algorithm 6.2, and some simulations are shown in Figure 6.2. The results show that when there are no missing observations, RKF maintains the same level of tracking error as the reference case, whereas RAL-TRK is also capable but at cost of slower convergence. When observation losses occur, using no estimators and using the previous estimates both cause a large compromise on the tracking error, but RKF-IO significantly reduces this compromise and the tracking error is slightly higher than the reference. The higher error of RKF-IO is due to the trade-off of observation losses, but with a kinematics model as a backup, the system is more robust.

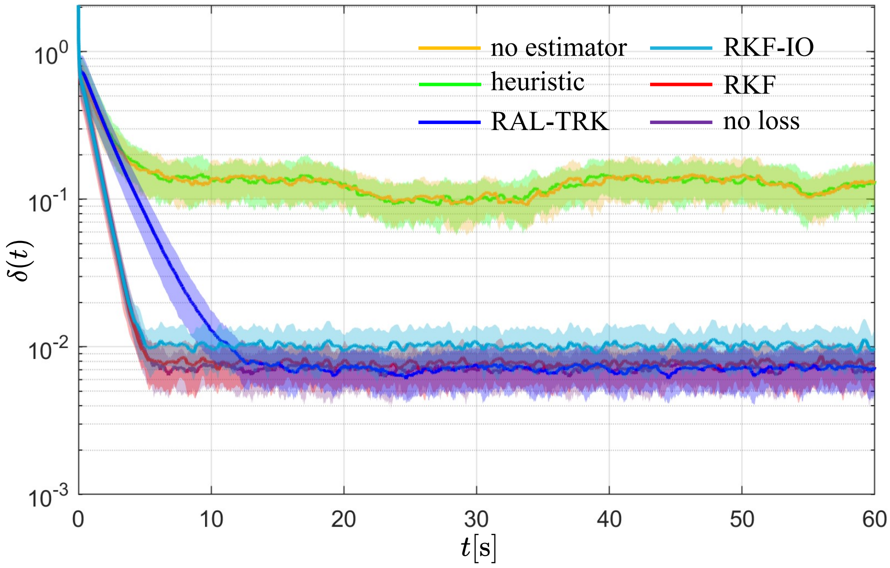


Figure 6.2: Convergence simulations and comparisons of some proposed methods. The simulation parameters for the Kalman filters are $\sigma_v = 0.1$, $\sigma_w = 0.001$ and the number of known observations is again chosen as $\left|\mathcal{N}_i^k\right| = 3$. For the "no loss" and "RKF" cases, there are no missing observations. Some references to the presented methods are: heuristic (6.18), RAL-TRK (Algorithm 5.3), RKF (Algorithm 6.1), and RKF-IO (Algorithm 6.2).

6.4 Relative Kalman Filter with Switching Observations

The previous approach using RKF with intermittent observations propagates the predicted states directly when no observation arrives. As seen in the simulation, the tracking error is boosted compared to the full observation case and RAL. In this chapter, we aim to minimize this compromise by fusing RAL estimates into the Kalman filter framework. The way to accomplish this is to treat RAL estimates as alternative observations in addition to the true ones and switch the observation model for the relative Kalman filter.

Algorithm 6.2 *Relative Kalman filters with intermittent observations (RKF-IO)*

```
Initialization
    Calculate stress matrix $\mathbf{L}$
    Draw initial positions $\mathbf{z}_i(0) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{P}_0)$
    Set noise covariance $\mathbf{R}$ and $\mathbf{Q}$
    $\hat{\mathbf{z}}'_{ij}(0|0) = \mathbf{0}_{3D}$    ▷ Initialize the state
    $\boldsymbol{\Sigma}_{ij}(0|0) = 4\mathbf{I}_{3D}$    ▷ Initialize the covariance
    $k = 0$
while $k \leq K_{max}$ do
    for $i \in \mathcal{V}_f$ do
        Define $\mathcal{N}_i = (\mathcal{N}_i^k, \mathcal{N}_i^m)$
        for $j \in \mathcal{N}_i$ do
            $\hat{\mathbf{z}}'_{ij}(k|k-1) = \mathbf{F}\hat{\mathbf{z}}'_{ij}(k-1|k-1)$    ▷ Prediction
            $\boldsymbol{\Sigma}_{ij}(k|k-1) = \mathbf{F}\boldsymbol{\Sigma}_{ij}(k-1|k-1)\mathbf{F}^T + \mathbf{Q}$
            Draw $\mathbf{v}_{ij}(k) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{R})$
            $\mathbf{y}_{ij}(k) = \mathbf{z}_{ij}(k) + \mathbf{v}_{ij}(k)$    ▷ Noisy observation
            if $j \in \mathcal{N}_i^k$ then    ▷ Correction only for available observation
                $\mathbf{K}_{ij}(k) = \boldsymbol{\Sigma}_{ij}(k|k-1)\mathbf{G}^T(\mathbf{G}\boldsymbol{\Sigma}_{ij}(k|k-1)\mathbf{G}^T + \mathbf{R})^{-1}$
                $\hat{\mathbf{z}}'_{ij}(k|k) = \hat{\mathbf{z}}'_{ij}(k|k-1) + \mathbf{K}_{ij}(k)(\mathbf{y}_{ij}(k) - \mathbf{G}\hat{\mathbf{z}}'_{ij}(k|k-1))$
                $\boldsymbol{\Sigma}_{ij}(k|k) = (\mathbf{I}_{3D} - \mathbf{K}_{ij}(k)\mathbf{G})\boldsymbol{\Sigma}_{ij}(k|k-1)$
            else
                $\hat{\mathbf{z}}'_{ij}(k|k) = \hat{\mathbf{z}}'_{ij}(k|k-1)$
                $\boldsymbol{\Sigma}_{ij}(k|k) = \boldsymbol{\Sigma}_{ij}(k|k-1)$
            end if
            $\hat{\mathbf{z}}_{ij}^{\mathrm{rkf}}(k) = \mathbf{G}\hat{\mathbf{z}}'_{ij}(k|k)$    ▷ Extract position estimate
        end for
        $\mathbf{u}_i(k) = -\frac{1}{\gamma_i}\sum_{j \in \mathcal{N}_i} l_{ij}(\hat{\mathbf{z}}_{ij}^{\mathrm{rkf}}(k) - \dot{\mathbf{z}}_j(k))$    ▷ Control law
        $\mathbf{z}_i(k+1) = \mathbf{z}_i(k) + \Delta t \mathbf{u}_i(k)$
    end for
    $k = k + 1$
end while
```

6.4.1 Observation Models

Recollect in Theorem 5.1 from Chapter 5 that RAL gives unbiased estimates, so we can model the RAL observation as

$$\hat{\mathbf{z}}_{ij}^{\mathrm{ral}}(k) = \mathbf{z}_{ij}(k) + \tilde{\mathbf{v}}_{ij}(k), \tag{6.20}$$

where $\tilde{\mathbf{v}}_{ij}(k)$ is the equivalent noise of RAL estimates and has zero mean. The covariance of this noise, however, is tricky and trivial to find. With simple maneuver of (6.20) and express it in terms of (5.11) and (5.9), we have

$$\begin{aligned} \tilde{\mathbf{v}}_{ij}(k) &= \hat{\mathbf{z}}_{ij}^{\mathrm{ral}}(k) - \mathbf{z}_{ij}(k) \\ &= \mathbf{V}_i^T(k)\mathbf{q}_{ij}(k) - \mathbf{n}_{ij}(k), \end{aligned} \tag{6.21}$$

in which the $\mathbf{n}_{ij}$, due to dynamics of the system, is hard to accurately characterize in terms of covariance. But it should resemble structure of the true observation noise covariance which is $\mathbf{v}_{ij}(k) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}, \mathbf{R})$. We here assume that the covariance for this term is $\mathbf{R}$ which means that the uncertainty for the true relative positions with respect to the target positions is the same level of observation noise. As such, we can express the covariance of the equivalent noise $\tilde{\mathbf{v}}_{ij}(k)$ as

$$\begin{aligned}\tilde{\mathbf{R}} &= \text{cov}[\mathbf{V}_i^T(k)\mathbf{q}_{ij}(k) - \mathbf{n}_{ij}(k)] \\ &= (1 + \mathbf{q}_{ij}^T\mathbf{q}_{ij})\mathbf{R}. \end{aligned} \tag{6.22}$$

Now we conclude the switching observation models as

$$\hat{\mathbf{y}}_{ij}(k) = \begin{cases} \hat{\mathbf{z}}_{ij}^{\text{ral}}(k), \text{with } \tilde{\mathbf{R}}, & \text{if } j \in \mathcal{N}_i^m \\ \mathbf{y}_{ij}(k), \text{with } \mathbf{R}, & \text{if } j \in \mathcal{N}_i^k \end{cases}. \tag{6.23}$$

The steps to perform relative Kalman filter with switching observations (RKF-SO) are described in Algorithm 6.3 and an overview of such systems is shown in Figure 6.3. The performance can be seen in Figure 6.4 in the next section in comparison with several other filters, but the conclusion is that the compromise on the mean of the tracking error with RKF-IO is reduced using RKF-SO. However, since RAL observations are employed, the drawback of slow convergence is also inherited. This is going to be addressed in the next section.

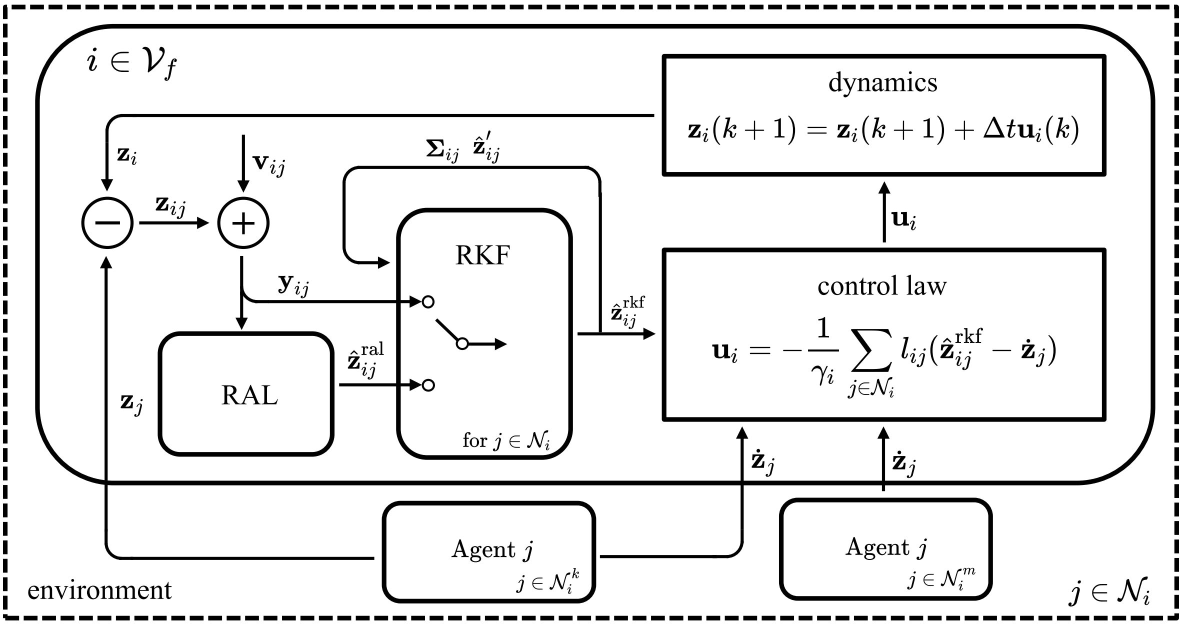

$i \in \mathcal{V}_f$
dynamics
$\mathbf{z}_i(k+1) = \mathbf{z}_i(k+1) + \Delta t\mathbf{u}_i(k)$
$\mathbf{z}_i$
$\mathbf{v}_{ij}$
$\boldsymbol{\Sigma}_{ij}$ $\hat{\mathbf{z}}'_{ij}$
$\mathbf{z}_{ij}$
$\mathbf{u}_i$
RKF
$\mathbf{y}_{ij}$
control law
$\hat{\mathbf{z}}_{ij}^{\text{rkf}}$
$\mathbf{u}_i = -\frac{1}{\gamma_i}\sum_{j\in\mathcal{N}_i} l_{ij}(\hat{\mathbf{z}}_{ij}^{\text{rkf}} - \dot{\mathbf{z}}_j)$
$\hat{\mathbf{z}}_{ij}^{\text{ral}}$
$\mathbf{z}_j$
RAL
for $j \in \mathcal{N}_i$
$\dot{\mathbf{z}}_j$
$\dot{\mathbf{z}}_j$
Agent j
$j \in \mathcal{N}_i^k$
Agent j
$j \in \mathcal{N}_i^m$
environment
$j \in \mathcal{N}_i$

Figure 6.3: An overview of the relative Kalman filter with switching observations

6.4.2 Improved Convergence with Convergence Indicator

As mentioned, RKF-SO takes advantage of RAL estimates but also absorbs their drawback. It is natural to motivate a strategy where we use RKF-IO at the beginning to

Algorithm 6.3 *Relative Kalman filter with switching observations (RKF-SO)*

```
Initialization
    Calculate stress matrix $\mathbf{L}$
    Draw initial positions $\mathbf{z}_i(0) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{P}_0)$
    Set noise covariance $\mathbf{R}$ and $\mathbf{Q}$
    $\hat{\boldsymbol{\Theta}}_i^{\mathrm{trk}}(0) = \mathbf{I}_D$ and set $\mu$
    $\hat{\mathbf{z}}'_{ij}(0|0) = \mathbf{0}_{3D}$                                   ▷ Initialize the state
    $\boldsymbol{\Sigma}_{ij}(0|0) = 4\mathbf{I}_{3D}$                                ▷ Initialize the covariance
    $k = 0$
while $k \leq K_{max}$ do
    for $i \in \mathcal{V}_f$ do
        Define $\mathcal{N}_i = (\mathcal{N}_i^k, \mathcal{N}_i^m)$
        for $j \in \mathcal{N}_i$ do
            Draw $\mathbf{v}_{ij}(k) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{R})$
            $\mathbf{y}_{ij}(k) = \mathbf{z}_{ij}(k) + \mathbf{v}_{ij}(k)$                 ▷ Noisy observation
            $\hat{\mathbf{z}}'_{ij}(k|k-1) = \mathbf{F}\hat{\mathbf{z}}'_{ij}(k-1|k-1)$     ▷ Prediction
            $\boldsymbol{\Sigma}_{ij}(k|k-1) = \mathbf{F}\boldsymbol{\Sigma}_{ij}(k-1|k-1)\mathbf{F}^T + \mathbf{Q}$
        end for
        Compute $\hat{\boldsymbol{\Theta}}_i(k)$ based on (5.6)
        $\hat{\boldsymbol{\Theta}}_i^{\mathrm{trk}}(k) = \hat{\boldsymbol{\Theta}}_i^{\mathrm{trk}}(k-1) + \mu(\hat{\boldsymbol{\Theta}}_i(k) - \hat{\boldsymbol{\Theta}}_i^{\mathrm{trk}}(k-1))$
        Reconstruct $\mathbf{z}_{ij}^{\mathrm{trk}}(k)$ based on (5.23)
        Construct $\hat{\mathbf{y}}_{ij}(k)$ based on (6.23)
        for $j \in \mathcal{N}_i$ do                                                     ▷ Correction
            $\mathbf{K}_{ij}(k) = \boldsymbol{\Sigma}_{ij}(k|k-1)\mathbf{G}^T(\mathbf{G}\boldsymbol{\Sigma}_{ij}(k|k-1)\mathbf{G}^T + \mathbf{R})^{-1}$
            $\hat{\mathbf{z}}'_{ij}(k|k) = \hat{\mathbf{z}}'_{ij}(k|k-1) + \mathbf{K}_{ij}(k)(\hat{\mathbf{y}}_{ij}(k) - \mathbf{G}\hat{\mathbf{z}}'_{ij}(k|k-1))$
            $\boldsymbol{\Sigma}_{ij}(k|k) = (\mathbf{I}_{3D} - \mathbf{K}_{ij}(k)\mathbf{G})\boldsymbol{\Sigma}_{ij}(k|k-1)$
            $\hat{\mathbf{z}}_{ij}^{\mathrm{rkf}}(k) = \mathbf{G}\hat{\mathbf{z}}'_{ij}(k|k)$    ▷ Extract position estimate
        end for
        $\mathbf{u}_i(k) = -\frac{1}{\gamma_i}\sum_{j \in \mathcal{N}_i} l_{ij}(\hat{\mathbf{z}}_{ij}^{\mathrm{rkf}}(k) - \dot{\mathbf{z}}_j(k))$    ▷ Control law
        $\mathbf{z}_i(k+1) = \mathbf{z}_i(k) + \Delta t\mathbf{u}_i(k)$
    end for
    $k = k + 1$
end while
```

induce fast convergence and use RKF-SO post-convergence to have lower tracking error. This can be achieved by modulating the covariance structure for RAL observations as $\tilde{\mathbf{R}}^{\mathrm{mod}}(k) = \alpha\tilde{\mathbf{R}}$. If α is very large, then the Kalman gain $\mathbf{K}_{ij}(k)$ will be very small and give small weight to the observations. In this way, the Kalman filter leans towards the prediction and degenerates to a Kalman filter with intermittent observations.

In general, to achieve the fast convergence, we need to design function of time $\tilde{\mathbf{R}}^{\mathrm{mod}}(k) = \xi_i(k, \tilde{\mathbf{R}})$. This is challenging because the agents need to be locally aware of the convergence state of the entire formation.

Time thresholding

This simplest way to design this $\xi_i(k, \tilde{\mathbf{R}})$ is to choose a large value α (e.g., $\alpha = 1e5$) and pick a time threshold k_c based on experience. The function of $\xi_i(k, \tilde{\mathbf{R}})$ is then expressed as

$$\xi_i(k, \tilde{\mathbf{R}}) = \begin{cases} \alpha\tilde{\mathbf{R}}, & \text{for } k < k_c \\ \tilde{\mathbf{R}}, & \text{for } k \geq k_c \end{cases}, i \in \mathcal{V}_f. \tag{6.24}$$

Although this method is easy enough, the choice of k_c relies too much on experience and is not generalizable to other formation designs. It is not robust to sudden post-convergence disturbance either. For instance, if the system needs to reconverge after a sudden disturbance, then the k_c will not be of any help.

Error triggering

Fortunately, when we introduced a convergence indicator in Section 4.4 based on the fact that the difference of local estimates of affine transformation parameters in the neighborhood reflects the state of convergence. The curve of this indicator resembles that of the tracking error although the values might be different. This property can be explored for the design of $\xi_i(k, \tilde{\mathbf{R}})$. If we use an error $\epsilon_i(k)$ to denote the function in (4.28), then we can also set a fixed ϵ_c as a trigger threshold. Then the function $\xi_i(k, \tilde{\mathbf{R}})$ can be designed as

$$\xi_i(k, \tilde{\mathbf{R}}) = \begin{cases} \alpha\tilde{\mathbf{R}}, & \text{for } \epsilon_i(k) \geq \epsilon_c \\ \tilde{\mathbf{R}}, & \text{for } \epsilon_i(k) < \epsilon_c \end{cases}, i \in \mathcal{V}_f, \tag{6.25}$$

where α is still a large number (e.g., $\alpha = 1e5$) and $\epsilon_i(k) = f_i(\hat{\mathbf{\Theta}}_i(k), \{\hat{\mathbf{\Theta}}_j(k)\}_{j\in\mathcal{N}_i})$ as defined in (4.28). The advantage of this method is clear that whenever the error $\epsilon_i(k)$ is large implying an insufficiently converged formation, $\xi_i(k, \tilde{\mathbf{R}})$ will keep using the large value until the formation is converged. As such quick reconvergence will be ensured in event of sudden disturbance. However, the drawback of this design is also obvious. In noisy environments, the error ϵ_i is also noisy, and the choice of the threshold ϵ_c to prevent unnecessary triggering and mistriggering is tricky.

Adaptive penalty

Finally, we present a design to adaptively penalize the noise covariance for RAL observations and no threshold needs to be decided. Function $\xi_i(k, \tilde{\mathbf{R}})$ is designed as

$$\xi_i(k, \tilde{\mathbf{R}}) = \tilde{\mathbf{R}} + \epsilon_i(k)\mathbf{I}_D, \tag{6.26}$$

for $i \in \mathcal{V}_f$. As can be seen, an additional term is added to penalize $\tilde{\mathbf{R}}$. When the system is insufficiently converged, large $\epsilon_i(k)$ will make the penalty dominating and when the system is converged, small $\epsilon_i(k)$ will result in negligible influence on the structure of $\tilde{\mathbf{R}}$. The advantage of this method is no thresholding or triggering is needed based on experience. But one should beware of the values of $\epsilon_i(k)$ and make sure it generates sufficient and negligible penalty at appropriate times. The procedures to perform adaptive penalty in relative Kalman filter with switching observations (RKF-SOAP) frameworks are described in Algorithm 6.4 and some simulations are shown in Figure 6.4. It could be seen that using RKF-SOAP the convergence is accelerated compared to the RKF-IO level with the error mean still lower than the RKF-IO case.

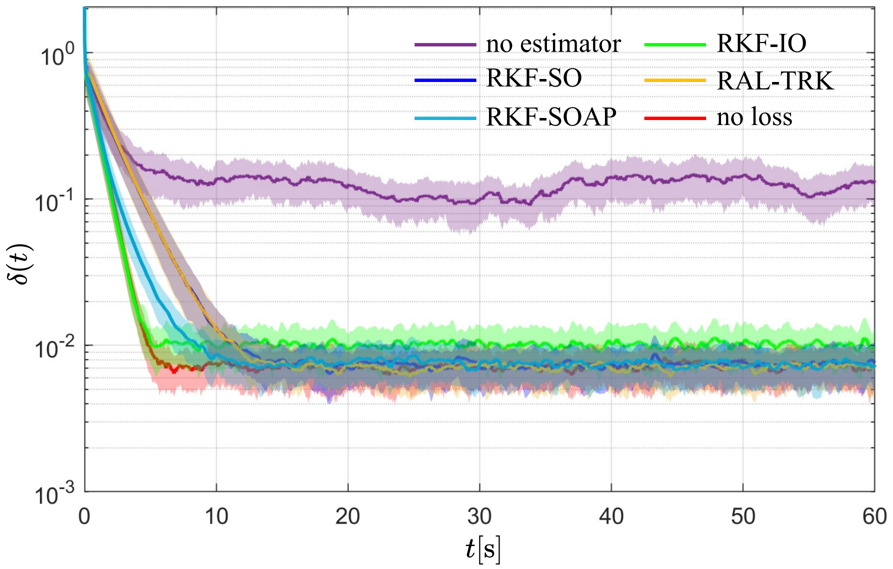


Figure 6.4: Simulation result for RKF-SOAP with comparison to the previous methods. The simulation parameters are $\sigma_v = 0.1$, $\sigma_w = 0.001$ and the number of available observation per agent is chosen as $\left|\mathcal{N}_i^k\right| = 3$. Some reference to the presented methods are: RAL-TRK (Algorithm 5.3), RKF-IO (Algorithm 6.2), RKF-SO (Algorithm 6.3), and RKF-SOAP (Algorithm 6.4).

6.5 Summary

In this chapter, a dynamic model is established to approximate the relative kinematics of agents. Subsequently, a franchise of relative Kalman filters based on this model is derived to battle missing observations. Some key conclusions can be summarized as follows.

- In the full-observation case RKF could reduce the variance of the tracking error resulting in a smoother trajectory.
- In event of observation losses, RKF-IO could significantly reduce the compromise on the mean of the tracking error.
- With RAL treated as alternative observations, RKF-SO could further reduce the mean of the error compared to RKF-IO, and an improved version of RKF-SOAP could further accelerate the convergence.

In general, the model-based methods do not have geometry limits and could thus deal with theoretically any amount of observation loss with a trade-off on the tracking error. In the next chapter, all the methods introduced so far to deal with observation losses will be pooled together and tested with more realistic scenarios.

Algorithm 6.4 *RKF with switching observations using adaptive penalty (RKF-SOAP)*

```
Initialization
    Calculate stress matrix $\mathbf{L}$
    Draw initial positions $\mathbf{z}_i(0) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{P}_0)$
    Set noise covariance $\mathbf{R}$ and $\mathbf{Q}$
    $\hat{\mathbf{\Theta}}_i^{\text{trk}}(0) = \mathbf{I}_D$ and set $\mu$
    $\hat{\mathbf{z}}'_{ij}(0|0) = \mathbf{0}_{3D}$                                        ▷ Initialize the state
    $\mathbf{\Sigma}_{ij}(0|0) = 4\mathbf{I}_{3D}$                                     ▷ Initialize the covariance
    $k = 0$
while $k \leq K_{max}$ do
    for $i \in \mathcal{V}_f$ do
        Define $\mathcal{N}_i = (\mathcal{N}_i^k, \mathcal{N}_i^m)$
        for $j \in \mathcal{N}_i$ do
            Draw $\mathbf{v}_{ij}(k) \sim \boldsymbol{\mathcal{N}}(\mathbf{0}_D, \mathbf{R})$
            $\mathbf{y}_{ij}(k) = \mathbf{z}_{ij}(k) + \mathbf{v}_{ij}(k)$                     ▷ Noisy observation
            $\hat{\mathbf{z}}'_{ij}(k|k-1) = \mathbf{F}\hat{\mathbf{z}}'_{ij}(k-1|k-1)$                     ▷ Prediction
            $\mathbf{\Sigma}_{ij}(k|k-1) = \mathbf{F}\mathbf{\Sigma}_{ij}(k-1|k-1)\mathbf{F}^T + \mathbf{Q}$
        end for
        Compute $\hat{\mathbf{\Theta}}_i(k)$ based on (5.6)
        $\hat{\mathbf{\Theta}}_i^{\text{trk}}(k) = \hat{\mathbf{\Theta}}_i^{\text{trk}}(k-1) + \mu(\hat{\mathbf{\Theta}}_i(k) - \hat{\mathbf{\Theta}}_i^{\text{trk}}(k-1))$
        Reconstruct $\mathbf{z}_{ij}^{\text{trk}}(k)$ based on (5.23)
        $\tilde{\mathbf{R}} = \tilde{\mathbf{R}} + \epsilon_i(k)\mathbf{I}_D$
        Construct $\hat{\mathbf{y}}_{ij}(k)$ based on (6.23)
        for $j \in \mathcal{N}_i$ do                                              ▷ Correction
            $\mathbf{K}_{ij}(k) = \mathbf{\Sigma}_{ij}(k|k-1)\mathbf{G}^T(\mathbf{G}\mathbf{\Sigma}_{ij}(k|k-1)\mathbf{G}^T + \mathbf{R})^{-1}$
            $\hat{\mathbf{z}}'_{ij}(k|k) = \hat{\mathbf{z}}'_{ij}(k|k-1) + \mathbf{K}_{ij}(k)(\hat{\mathbf{y}}_{ij}(k) - \mathbf{G}\hat{\mathbf{z}}'_{ij}(k|k-1))$
            $\mathbf{\Sigma}_{ij}(k|k) = (\mathbf{I}_{3D} - \mathbf{K}_{ij}(k)\mathbf{G})\mathbf{\Sigma}_{ij}(k|k-1)$
            $\hat{\mathbf{z}}_{ij}^{\text{rkf}}(k) = \mathbf{G}\hat{\mathbf{z}}'_{ij}(k|k)$                  ▷ Extract position estimate
        end for
        $\mathbf{u}_i(k) = -\frac{1}{\gamma_i}\sum_{j \in \mathcal{N}_i} l_{ij}(\hat{\mathbf{z}}_{ij}^{\text{rkf}}(k) - \dot{\mathbf{z}}_j(k))$                     ▷ Control law
        $\mathbf{z}_i(k+1) = \mathbf{z}_i(k) + \Delta t\mathbf{u}_i(k)$
    end for
    $k = k + 1$
end while
```

Case Study 7

The previous chapters have brought to light three types of methods to battle observation losses, namely RAL-based methods (RAL, RAL-TRK, and RAL-QS), RKF-based methods (RKF and RKF-IO), and RAL-RKF fusion (RKF-SO and RKF-SOAP). Since the principles of these methods are different, they also exhibit different capabilities in dealing with certain forms of observation losses. In this chapter, we pool these methods and evaluate their advantages and disadvantages in two practical scenarios.

Here in Table 7.1 we list the methods to be evaluated and their hyperparameters for simulation. Note that if one method incorporates the hyperparameters of another method, the hyperparameters are chosen the same as the other method in this table. The noise variance, as a global simulation parameter, is chosen as $\sigma_v = 0.1$.

Table 7.1: Methods to be evaluated and their hyperparameter choices

Methods	Reference	Hyperparameters	Value
RAL-TRK	Algorithm 5.3	μ: weight for the residual error	0.1
RAL-QS	Algorithm 5.4	W_l: window length	500
		ν: weight for the regularizer	100
RKF-IO	Algorithm 6.2	σ_w: confidence for the dynamic model	0.001
RKF-SO	Algorithm 6.3	-	-
RKF-SOAP	Algorithm 6.4	-	-

7.1 Random Loss of Observations

The observation losses in the previous chapters are formulated by splitting the set of neighbors $\mathcal{N}_i$ into $\mathcal{N}_i^k$ and $\mathcal{N}_i^m$, and randomly construct the $\mathcal{N}_i^k$ with a fixed cardinality for every iteration. In this way, it is easy to ensure the geometrical viability for RAL and an equivalent percentage of observation losses can be also calculated. But this is not truly random in a statistical manner.

In this section, we model the arrival of observations of each relative position as a Bernoulli process across time. Again, we use the binary variable $\beta_{ij}(k)$ to denote the availability and we define the probability $\mathrm{P}[\beta_{ij}(k) = 1] = \lambda$. Correspondingly, $\mathrm{P}[\beta_{ij}(k) = 0] = 1 - \lambda$. As such, for each relative position, there exists a sequence $\boldsymbol{\beta}_{ij} = [\beta_{ij}(0), \beta_{ij}(2), ..., \beta_{ij}(K_{max} - 1)]$ for $i \in \mathcal{V}_f, j \in \mathcal{N}_i$. Using this formulation, it is inevitable that RAL is denied for some k as is explained in an example in Figure 7.1. It can be expected that the chance of RAL being denied increases as λ decreases, which is also one of the motivations to derive model-based filtering, i.e., the relative Kalman filter family. Figure 7.2 simulates the probability of geometrically feasible situations for

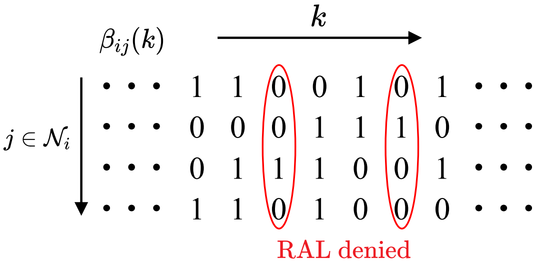


Figure 7.1: Examples of RAL-denied situations. As circled in red, for some agent i if there are not at least 2 neighbors in $\mathbb{R}^2$ to whom the observations are available, RAL is denied.

RAL under the graph in Figure 3.3 as λ decreases with 100,000 experiments. As can be seen, the chance of RAL being available decreases slowly with high λ, and decreases drastically as λ continues to decrease. This means under slightly lossy conditions, RAL could work and so do the RAL-based methods, but under extremely lossy conditions, RAL could barely work and provide little support to the related methods. As a compromise, we employ RAL estimates when available and use zeros values instead when RAL is denied. The methods built on top of RAL are also treated this way.

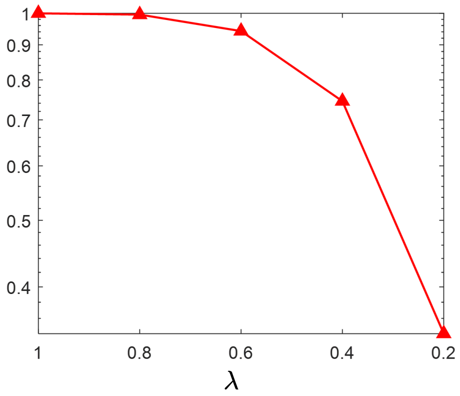


Figure 7.2: Probability of satisfying geometric feasibility for RAL with different λ

To conveniently present the result with different values of λ, an average tracking error across time is defined as

$$\bar{\delta} = \frac{1}{K_{max}} \sum_{k=0}^{K_{max}-1} \delta(k). \tag{7.1}$$

Figure 7.3 shows the result of average tracking with different methods. Generally, the average tracking errors increase as λ decreases as a general trade-off of losing more observations. However, different methods exhibit different resilience against the extent of observation losses. In the case of no estimator, the error is boosted even with a few observation losses. RAL-based methods, RAL-TRK and RAL-QS, show resilience in slightly lossy conditions, e.g., when $\lambda = 0.8$. The RKF-based method shown in the figure, namely RKF-IO, shows strong resilience in even very lossy conditions, e.g., when $\lambda = 0.2$. The fusion methods, however, present vastly different performances. Due to

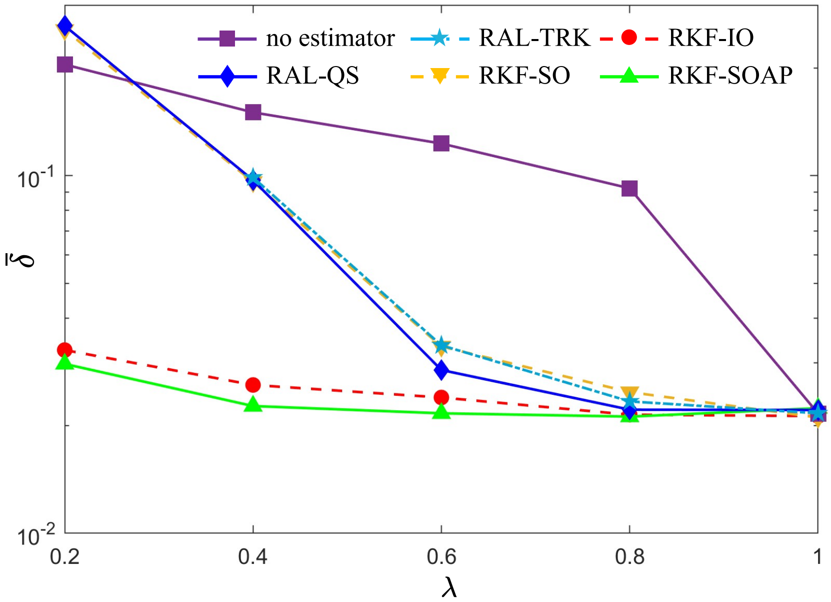


Figure 7.3: Average tracking error of several methods under different Bernoulli probability λ. The simulations are also averaged over 10 experiments.

the mixture of RAL observations, RKF-SO cannot perform decent correction in very lossy environments and thus presents a high error when λ is small. RKF-SOAP, on the other hand, tracks the convergence state by the convergence indicator and relies more on the RAL observations when they are more reliable. As such, not only is it resilient to very lossy conditions, but it also performs better than RKF-IO since RAL gives extra observations. But one should also expect a heavier computation load of RKF-IO as a trade-off.

There are several other perspectives to interpret the random loss of observations. If we consider observation loss a passive situation, we can also find scenarios in which observations are not taken actively. With the same requirement of control interval, the frequency of sensing could be decreased for energy efficiency if acquiring observations is expensive in some applications. Also, if the agents in the network are not synchronized, it can be modeled this way as observation loss.

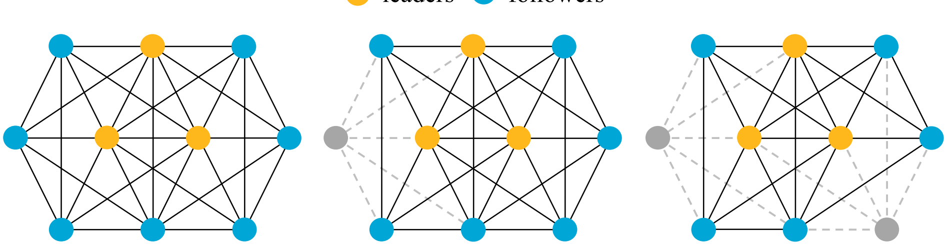


Figure 7.4: An example of out-of-service agent

7.2 Out-of-Service Agents

In the previous scenario, observations are regarding the relative positions meaning that the edges are temporarily unavailable in a graph. Another scenario that also entails unavailable edges is the loss of nodes in a graph which results in permanent observation losses. As claimed in Chapter 1, observations could be unavailable if the objects to be observed are gone. In practice, agents could be offline or out-of-service due to e.g., maintenance, scheduling, failure, etc. For expensive applications such as space missions and deep ocean explorations, a single node failure should not affect the rest of the system. As such, constant state estimation shall be performed regarding the missing edges as if a substitute virtual node is still running with the system.

To simulate such a scenario, a sequence of formations is designed in Figure 7.4 in which the first one is the complete graph, the second one loses one node, and the last one loses one more. The duration for each case from left to right in Figure 7.4 in simulation is $[0, 10)$ seconds, $[10, 35)$ seconds, and $[35, 60]$ seconds. The results for our proposed methods are shown in Figure 7.5. Two clear diverging curves are "no estimator" and RKF-IO. The behavior when no estimator is applied is different compared to the random observation loss scenario where the system is suboptimally converging or nonconverging but not diverging. This is because when edges are permanently lost, the underlying graph and equilibrium stress are fundamentally changed. The formation is not stabilizable under previous equilibrium stress. For the RKF-IO method, the system is diverging because no observations at all are served as corrections for the Kalman filter and the dynamic model is too outdated to give satisfactory estimates. It cannot track even the slightest maneuver in the formation. However, since two out-of-service nodes do not cause heavy edge losses, the RAL-based methods and the fusion type of methods perform well and the system remains almost unaffected.

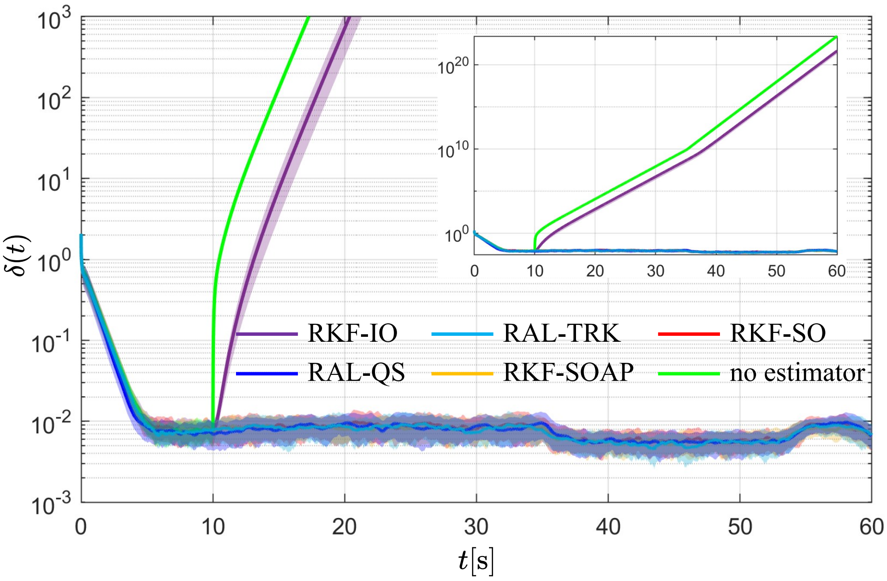


Figure 7.5: Convergence plot of the simulation. A complete view of the figure is shown on the top right.

Conclusions and Future Work 8

8.1 Conclusions

In this thesis, the problem of observation losses in the affine formation control and maneuver is addressed with several state estimators proposed. These methods are tested over random observation loss (ROL) scenario and out-of-service agents scenario which is equivalent to permanent observation loss (POL). The qualitative conclusion is summarized in Table 8.1.

Table 8.1: Comparison and evaluation of proposed methods (r.a.=robustness against)

Criteria	RAL	RAL-TRK	RAL-QS	RKF-IO	RKF-SO	RKF-SOAP
r.a. light ROL	good	good	good	good	good	good
r.a. heavy ROL	bad	bad	bad	good	bad	good
r.a. POL	good	good	good	bad	good	good
Smoothing effect			✓	✓	✓	✓
Convergence speed	low	medium	medium	fast	medium	fast
Computation load	low	low	medium	medium	high	high

In response to the objectives proposed in Section 1.3.2, we summarize the contribution of this thesis as follows.

- Relative affine localization, a state estimator using affine formation as spatial constraints, is put forward with necessary and sufficient conditions given on the applicability. Variants of this estimator for convergence acceleration and trajectory smoothing are also proposed. They can cope with light random observation losses and permanent observation losses.

- Relative Kalman filter, as a model-based state estimator, is established as a framework in which the Kalman filter with intermittent observations is proposed to battle observation losses. This framework can deal with both light and heavy random observation losses with smoothing capabilities.

- An RAL-RKF fusion technique is proposed to inherit the merits of the two types of estimators and a modulation mechanism is applied to overcome the shortcoming. As a result, the fusion methods can cope with all aforementioned observation loss scenarios with comparably the best performance.

8.2 Future Work

Although this thesis addresses the problem of observation loss for affine formation control systems, there are still some loose ends and interesting future directions to be discussed.

An important assumption when dealing with observation losses in this thesis is that the agents are notified of which edges are missing. However, from a sensing perspective, there is still a long way from e.g. sensor malfunctioning to asserting that the observation is unavailable. As such, the detection and decision of observation losses from sensor outputs should be addressed in the future as a premise of this thesis. But this has to deal with specific sensors for relative localization and should be studied with certain applications.

Although the theoretical results assure that RAL works only if the formation has sufficiently converged, the numerical results show that RAL also works prior to convergence at the cost of slower convergence speed. We also see that the system does not converge if only a small number of observations are available (see $\left|\mathcal{N}_i^k\right| = 2$ case in Figure 4.4). As such, analysis of the convergence condition of RAL and the theoretical potential bound on the number of available relative positions for convergence are still to be explored.

In this thesis, we always compare the results of the proposed methods with the no-loss case, which is considered to be a lower bound for observation losses. However, more rigorous derivations of theoretical bounds in the event of observation losses can be made. This will provide a clearer view of the proposed methods in terms of their performance.

In data-driven filtering, there are lots of hyperparameters involved in the proposed methods. We mostly give heuristic values for these hyperparameters and show the numerical results. However, these values are not necessarily the optimal ones and a systematic search for better choices can be done in the future.

In the model-based estimation and filtering, i.e., the relative Kalman filter family, only relative positions $\mathbf{z}_{ij}$ are used as partial state observations since we would like the state-space model to be generalizable to other controllers. However, for the particular controller (3.2) adopted for this thesis, velocity information $\dot{\mathbf{z}}_j$ is available and could be potentially used as observations as well. Since the quality of the dynamic model influences the performance of the Kalman filter, more and better models could be used to establish an IMM type of filter and potentially reach even lower tracking error than the no-loss case.

It has been shown that RAL-related methods can deal with permanent observation losses by substituting with virtual nodes and edges in the graph to remain the validity of equilibrium stress. Conversely, for those graphs that do not satisfy the condition (universal rigidity) to have a working stress matrix for affine formation control, it is also an idea to add virtual nodes and edges for them to acquire a working stress matrix to perform affine maneuvers.

Bibliography

[1] C.-S. Karavas, G. Kyriakarakos, K. G. Arvanitis, and G. Papadakis, "A multi-agent decentralized energy management system based on distributed intelligence for the design and control of autonomous polygeneration microgrids," *Energy Conversion and Management*, vol. 103, pp. 166–179, 2015.

[2] F. Walter, D. Gräff, F. Lindner, P. Paitz, M. Köpfli, M. Chmiel, and A. Fichtner, "Distributed acoustic sensing of microseismic dources and wave propagation in glaciated terrain," *Nature Communications*, vol. 11, 2020.

[3] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proceedings of the 33rd International Conference on Machine Learning*, vol. 48, 2016, pp. 1928–1937. [Online]. Available: https://proceedings.mlr.press/v48/mniha16.html

[4] D. Becker, M. D. Lachmann, S. T. Seidel, H. Ahlers, A. N. Dinkelaker, J. Grosse, O. Hellmig, H. Muentinga, V. Schkolnik, T. Wendrich, A. Wenzlawski, B. Weps, R. Corgier, T. Franz, N. Gaaloul, W. Herr, D. Luedtke, M. Popp, S. Amri, H. Duncker, M. Erbe, A. Kohfeldt, A. Kubelka-Lange, C. Braxmaier, E. Charron, W. Ertmer, M. Krutzik, C. Laemmerzahl, A. Peters, W. P. Schleich, K. Sengstock, R. Walser, A. Wicht, P. Windpassinger, and E. M. Rasel, "Space-borne bose-einstein condensation for precision interferometry," *Nature*, vol. 562, no. 7727, p. 391–395, 2018.

[5] S. Minaeian, J. Liu, and Y.-J. Son, "Vision-based target detection and localization via a team of cooperative uav and ugvs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 1005–1016, 2016.

[6] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.

[7] S. Milani and A. Memo, "Impact of drone swarm formations in 3d scene reconstruction," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 2598–2602.

[8] K. C. W. Goh, R. B. C. Ng, Y.-K. Wong, N. J. H. Ho, and M. C. H. Chua, "Aerial filming with synchronized drones using reinforcement learning," *Multimedia Tools and Applications*, vol. 80, no. 12, pp. 18 125–18 150, 2021.

[9] B. Das, B. Subudhi, and B. B. Pati, "Cooperative formation control of autonomous underwater vehicles: An overview," *International Journal of Automation and Computing*, vol. 13, no. 3, pp. 199–225, 2016.

[10] G.-P. Liu and S. Zhang, "A survey on formation control of small satellites," *Proceedings of the IEEE*, vol. 106, no. 3, pp. 440–457, 2018.

[11] J. D. Monnier, "A realistic roadmap to formation flying space interferometry," *arXiv: Instrumentation and Methods for Astrophysics*, 2019.

[12] X. Liu, K. Lam, B. Alkouz, B. Shahzaad, and A. Bouguettaya, "Constraint-based formation of drone swarms," in *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events (PERCOM Workshops)*, 2022.

[13] F. Yang, S. Liu, and F. Liu, "Cooperative transport strategy for formation control of multiple mobile robots," *Journal of Zhejiang University-Science C (Computers & Electronics)*, vol. 11, no. 12, pp. 931–938, 2010.

[14] H. W. Agung and I. Nilkhamhang, "Hierarchical decentralized lqr control for formation-keeping of cooperative mobile robots in material transport tasks," *Engineering Journal-Thailand*, vol. 25, no. 12, pp. 37–50, 2021.

[15] L. Chen, H. Hopman, and R. R. Negenborn, "Distributed model predictive control for cooperative floating object transport with multi-vessel systems," *Ocean Engineering*, vol. 191, 2019.

[16] S. Unterstrasser, "The contrail mitigation potential of aircraft formation flight derived from high-resolution simulations," *Aerospace*, vol. 7, no. 12, 2020.

[17] D. Yang, A. Kuljpers, G. Dane, and T. van der Sande, "Impacts of large-scale truck platooning on dutch highways," in *21st Euro Working Group on Transportation Meeting (EWGT 2018)*, vol. 37, 2019, pp. 425–432.

[18] H. Ahn, D.-T. Le, D. T. Nguyen, and H. Choo, "Real-time drone formation control for group display," in *Proceedings of the 13th International Conference on Ubiquitous Information Management and Communication (IMCOM) 2019*, vol. 935, 2019, pp. 778–785.

[19] Q. Ouyang, Z. Wu, Y. Cong, and Z. Wang, "Formation control of unmanned aerial vehicle swarms: A comprehensive review," *Asian Journal of Control*.

[20] M. A. Kamel, X. Yu, and Y. Zhang, "Formation control and coordination of multiple unmanned ground vehicles in normal and faulty situations: A review," *Annual Reviews in Control*, vol. 49, pp. 128–144, 2020.

[21] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109814004038

[22] N. H. M. Li and H. H. T. Liu, "Formation uav flight control using virtual structure and motion synchronization," in *2008 American Control Conference*, 2008, pp. 1782–1787.

[23] M. A. Lewis and K. H. Tan, "High precision formation control of mobile robots using virtual structures," *Autonomous Robots*, vol. 4, pp. 387–403, 1997.

[24] X. Liu, S. S. Ge, and C.-H. Goh, “Formation potential field for trajectory tracking control of multi-agents in constrained space,” *International Journal of Control*, vol. 90, no. 10, pp. 2137–2151, 2017. [Online]. Available: https://doi.org/10.1080/00207179.2016.1237044

[25] S. Lin, R. Jia, M. Yue, and Y. Xu, “On composite leader–follower formation control for wheeled mobile robots with adaptive disturbance rejection,” *Applied Artificial Intelligence*, vol. 33, no. 14, pp. 1306–1326, 2019. [Online]. Available: https://doi.org/10.1080/08839514.2019.1685182

[26] S. Zhao, “Affine formation maneuver control of multiagent systems,” *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4140–4155, 2018.

[27] Z. Li, Z. Duan, G. Chen, and L. Huang, “Consensus of multiagent systems and synchronization of complex networks: A unified viewpoint,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, pp. 213–224, 2010.

[28] R. Suttner and Z. Sun, “Formation shape control based on distance measurements using lie bracket approximations,” *SIAM Journal on Control and Optimization*, vol. 56, no. 6, pp. 4405–4433, 2018.

[29] S. Zhao and D. Zelazo, “Translational and scaling formation maneuver control via a bearing-based approach,” *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 429–438, 2017.

[30] Z. Han, L. Wang, Z. Lin, and R. Zheng, “Formation control with size scaling via a complex laplacian-based approach,” *IEEE Transactions on Cybernetics*, vol. 46, no. 10, pp. 2348–2359, 2016.

[31] 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.

[32] S. Chen, D. Yin, and Y. Niu, “A survey of robot swarms relative localization method,” *Sensors*, vol. 22, no. 12, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/12/4424

[33] A. Olesiński and Z. Piotrowski, “An adaptive energy saving algorithm for an rssi-based localization system in mobile radio sensors,” *Sensors*, vol. 21, no. 12, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/12/3987

[34] I. Sobron, I. Landa, I. Eizmendi, and M. Velez, “Adaptive toa estimation with imperfect los and nlos knowledge in uwb positioning systems,” in *2020 IEEE Sensors*, 2020, pp. 1–4.

[35] B. Wang, Y. Wang, X. Qiu, and Y. Shen, “Ble localization with polarization sensitive array,” *IEEE Wireless Communications Letters*, vol. 10, no. 5, pp. 1014–1017, 2021.

[36] S. Bottigliero, D. Milanesio, M. Saccani, and R. Maggiora, "A low-cost indoor real-time locating system based on tdoa estimation of uwb pulse sequences," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.

[37] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[38] Y. Mo and B. Sinopoli, "Kalman filtering with intermittent observations: Tail distribution and critical value," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 677–689, 2012.

[39] J. Xu, X. Wu, and Y. Tang, "Optimal linear exponential quadratic gaussian estimation with intermittent observations," *IEEE Control Systems Letters*, vol. 3, no. 4, pp. 936–941, 2019.

[40] H. Lin, J. Lam, M. Z. Q. Chen, Z. Shu, and Z.-G. Wu, "Interacting multiple model estimator for networked control systems: Stability, convergence, and performance," *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 928–943, 2019.

[41] H. Lin, H. Su, Z. Shu, Z.-G. Wu, and Y. Xu, "Optimal estimation in udp-like networked control systems with intermittent inputs: Stability analysis and suboptimal filter design," *IEEE Transactions on Automatic Control*, vol. 61, no. 7, pp. 1794–1809, 2016.

[42] Q. Zhang, G. Yin, and L. Y. Wang, "A deep filtering approach for control of partially observed systems," *IEEE Control Systems Letters*, vol. 5, no. 4, pp. 1189–1194, 2021.

[43] Y. Liang, Y. Li, S. Chen, G. Qi, and A. Sheng, "Event-triggered kalman consensus filter for sensor networks with intermittent observations," *International Journal of Adaptive Control and Signal Processing*, vol. 35, no. 8, pp. 1478–1497, 2021.

[44] W. Li, Y. Jia, and J. Du, "Distributed kalman consensus filter with intermittent observations," *Journal of the Franklin Institute-Engineering and Applied Mathematics*, vol. 352, no. 9, SI, pp. 3764–3781, 2015.

[45] G. Wang, N. Li, and Y. Zhang, "Diffusion nonlinear kalman filter with intermittent observations," *Processing of the Institution of Mechanical Engineering Part G-Journal of Aerospace Engineering*, vol. 232, no. 15, pp. 2775–2783, 2018.

[46] Y. Wu, J. Zhang, Y. Ge, Z. Sheng, and Y. Fang, "A dropout compensation ilc method for formation tracking of heterogeneous multi-agent systems with loss of multiple communication packets," *Applied Science-Basel*, vol. 10, no. 14, 2020.

[47] L. Sedghi, J. John, M. Noor-A-Rahim, and D. Pesch, "Formation control of automated guided vehicles in the presence of packet loss," *Sensors*, vol. 22, no. 9, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/9/3552

[48] Z. Meng, C. Hu, L. Cao, and N. Wang, "Distributed model predictive control for multiple unmanned ground vehicles formation with packet loss," in *Proceedings of the 38th Chinese Control Conference (CCC)*, 2019, pp. 6124–6129.

[49] A. Nandanwar, N. K. Dhar, D. Malyshev, L. Rybak, and L. Behera, "Stochastic event-based super-twisting formation control for multiagent system under network uncertainties," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 2, pp. 966–978, 2022.

[50] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[51] M. van der Marel, "Edge state kalman filtering for distributed formation control systems," Master's thesis, Delft University of Technology, 2021.

[52] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*. Society for Industrial and Applied Mathematics (SIAM), 2009. [Online]. Available: https://books.google.nl/books?id=nHIzbApLOr0C

[53] R. E. Burkard and E. Çela, *Linear Assignment Problems and Extensions*. Boston, MA: Springer US, 1999, pp. 75–149. [Online]. Available: https://doi.org/10.1007/978-1-4757-3023-4_2

[54] J. R. Munkres, "Algorithms for the Assignment and Transportation Problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, March 1957.

[55] H. Garcia de Marina, B. Jayawardhana, and M. Cao, "Distributed rotational and translational maneuvering of rigid formations and their applications," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 684–697, 2016.

[56] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control and object transport in dynamic environments via constrained optimization," *The International Journal of Robotics Research*, vol. 36, pp. 1000 – 1021, 2017.

[57] J. Alonso-Mora, E. Montijano, T. Nägeli, O. Hilliges, M. Schwager, and D. Rus, "Distributed multi-robot formation control in dynamic environments," *Autonomous Robots*, vol. 43, 06 2019.

[58] S. Prince, *Computer Vision: Models Learning and Inference*. Cambridge University Press, 2012.

[59] T. Viklands, "Algorithms for the weighted orthogonal procrustes problem and other least squares problems," 2006.

[60] P. H. Schonemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.

[61] R. Everson, "Orthogonal, but not orthonormal, procrustes problems," in *Advances in Computational Mathematics. Available from http://www.ee.ic.ac.uk/research/neural/everson*, 1997.

[62] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. USA: Prentice-Hall, Inc., 1993.

[63] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[64] P. H. C. Eilers, "A perfect smoother," *Analytical Chemistry*, vol. 75, pp. 3631–3636, 2003.

[65] E. G. Ng and B. W. Peyton, "Block sparse cholesky algorithms on advanced uniprocessor computers," *SIAM Journal on Scientific Computing*, vol. 14, no. 5, pp. 1034–1056, 1993. [Online]. Available: https://doi.org/10.1137/0914063

[66] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, "Maintaining lu factors of a general sparse matrix," *Linear Algebra and its Applications*, vol. 88-89, pp. 239–270, 1987. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0024379587901121

[67] S. Zuliana and A. Perperoglou, "Two dimensional smoothing via an optimised whittaker smoother," *Big Data Analytics*, vol. 2, 03 2017.

[68] M. Zhang, J. Yang, J. Zhao, and Y. Dai, "A dead-reckoning based local positioning system for intelligent vehicles," in *2019 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, 2019, pp. 513–517.

[69] H. R. Park, D. J. Hyun, H. S. Yang, and H. S. Park, "A dead reckoning sensor system and a tracking algorithm for mobile robot," in *2009 ICCAS-SICE*, 2009, pp. 5559–5563.

[70] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

Stress Matrix A

The equilibrium stress, as mentioned in Chapter 2, ensures a stable formation and it is used in control law (3.2). In this appendix, we summarize two common methods to acquire the stress matrix in the existing literature.

Let $\mathbf{P} = [\mathbf{p}_1, ..., \mathbf{p}_N]^T \in \mathbb{R}^{N \times D}$ be the nominal configuration matrix and $\overline{\mathbf{P}} = [\mathbf{P} \mid \mathbf{1}_N] = \mathbf{U}\Sigma\mathbf{V}^T$ be the augmented configuration matrix which then admits a singular value decomposition (SVD). The first $D+1$ columns of $\mathbf{U}$ are denoted as $\mathbf{U}_1$ and the rest $N-D-1$ columns are denoted as $\mathbf{U}_2$. We also specify $\mathbf{b}_i^T \in \mathbb{R}^M$, $i \in \mathcal{V}$ as the rows of the incidence matrix defined in (2.1). An alternative definition of the stress matrix $\mathbf{L} \in \mathbb{S}^N$ is

$$\mathbf{L} = \mathbf{B}\text{diag}(\mathbf{w})\mathbf{B}^T, \tag{A.1}$$

where $\mathbf{w} \in \mathbb{R}^M$ is an edge-related weighting vector. As such, the stress matrix can also be seen as a weighted graph Laplacian matrix. And to calculate the stress matrix, it suffices to determine the vector $\mathbf{w}$.

A.1 Method 1

An convex optimization based solution proposed in [31] is

$$\begin{aligned} \max_{\mathbf{w}} \quad & \lambda \\ \text{s.t.} \quad & 0 < \lambda \leq \overline{\lambda} \\ & \mathbf{L} = \mathbf{B}\text{diag}(\mathbf{w})\mathbf{B}^T \ , \\ & \mathbf{Q}\mathbf{L}\mathbf{Q}^T \succ \lambda \mathbf{I}_{N-D-1} \\ & \mathbf{L}\mathbf{P} = \mathbf{0} \end{aligned} \tag{A.2}$$

where $\overline{\lambda}$ can be set to any positive value that guarantees a bounded result and $\mathbf{Q}$ can be acquired through the nullspace of $\overline{\mathbf{P}}$. More details on the formulation and the reasoning for this optimization problem can be found in [31]. This optimization problem can be numerically solved by toolboxes such as CVXPY [70].

A.2 Method 2

Another solution features solving a feasibility problem of a linear matrix inequality (LMI) [26]. Firstly, a matrix $\mathbf{E}$ is constructed as

$$\mathbf{E} = \begin{bmatrix} \overline{\mathbf{P}}^T \mathbf{B}\text{diag}(\mathbf{b}_1) \\ \vdots \\ \overline{\mathbf{P}}^T \mathbf{B}\text{diag}(\mathbf{b}_n) \end{bmatrix} \in \mathbb{R}^{N(D+1) \times M}, \tag{A.3}$$

whose nullspace has q orthonormal basis $\mathbf{r}_1, ..., \mathbf{r}_q \in \mathbb{R}^M$. Then we define $\mathbf{M}_i = \mathbf{U}_2^T \mathbf{B} \text{diag}(\mathbf{r}_i) \mathbf{B}^T \mathbf{U}_2$ for $i = 1, ..., q$. As such, the edge weight vector can be calculated as

$$\mathbf{w} = \sum_{i=1}^{q} c_i \mathbf{r}_i, \tag{A.4}$$

where the coefficients $c_1, ..., c_q$ satisfy the LMI

$$\sum_{i=1}^{q} c_i \mathbf{M}_i \succ 0. \tag{A.5}$$

Generally, this feasibility problem can be numerically solved with toolboxes. However, there are cases where the LMI can be simplified and can be solved very efficiently. For instance, if $\mathbf{E}$ has only one-dimensional nullspace spanned by $\mathbf{r}_1$. c_1 can be rapidly determined to be either 1 or -1 depending on the definiteness of $\mathbf{M}_1$. As such, the weight vector $\mathbf{w}$ is either $\mathbf{r}_1$ or $-\mathbf{r}_1$. This situation usually occurs with simple formations where there are small numbers of nodes and edges.