TÉCNICO LISBOA

# Bearings-based Cooperative Navigation for Arbitrary Formation Topologies

**Pedro Alves Mendes**

Thesis to obtain the Master of Science Degree in

## Electrical and Computer Engineering

Supervisor: Prof. Pedro Tiago Martins Batista

### Examination Committee

Chairperson: Prof. João Fernando Cardoso Silva Sequeira
Supervisor: Prof. Pedro Tiago Martins Batista
Member of the Committee: Prof. Bruno João Nogueira Guerreiro

**December 2021**

# Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

# Acknowledgments

Firstly, I would like to thank my family and friends, whose support has been essential and deeply appreciated, not only throughout the duration of this work, but also throughout my life. I would also like to express my gratitude towards the guidance provided by my supervisor, Prof. Pedro Batista, whose rigor and expertise have elevated this work into what it is.

This work has received partial funding from the European Union under Horizon 2020 FET Proactive Programme via grant agreement No. 101017808 (RAMONES project), and was supported by the Fundação para a Ciência e a Tecnologia (FCT) through LARSyS - FCT Project UIDB/50009/2020 and through the FCT project DECENTER [LISBOA-01-0145-FEDER-029605], funded by the Programa Operacional Regional de Lisboa 2020 and PIDDAC programs.

# Abstract

This work compares several cooperative navigation solutions for formations of autonomous vehicles, equipped with depth sensors and capable of taking bearing measurements to their neighbors under a certain measurement topology. Two approaches based on the extended Kalman filter are described, one centralized and the other decentralized. Additionally, four other Kalman filter implementations based on systems with linear dynamics using artificial measurements are also described, one centralized and the remaining ones decentralized. The presented algorithms were chosen for their simplicity, robustness, and scalability, which are all important design parameters when choosing an observer. Special emphasis was given to algorithms that require minimal communication, since the operating environment might not allow for high-bandwidth and low-latency communication with current technology, as is the case in underwater applications. Additionally, only algorithms that can handle arbitrary measurement topologies were considered, since one of the objectives of this work is to investigate algorithms that are versatile enough to do so. These algorithms were subsequently implemented in a simulation environment and their performance was analyzed. Some results pertaining to each algorithm are presented and, following that, Monte Carlo results were obtained in order to investigate the impact of the measurement topology on the behavior of the algorithms. In particular, the root-mean-squared-error of the obtained estimates and their mean error were investigated.

# Keywords

Decentralized navigation, Kalman filter, Monte Carlo analysis, bearing measurements

# Resumo

Esta tese compara vários algoritmos descentralizados para navegação de uma formação de veículos autónomos. É assumido que cada veículo tem capacidade para fazer medidas de direção aos seus vizinhos, parametrizadas por ângulos de azimute e elevação, e condicionadas de maneira a que estas obedeçam a uma certa topologia de medição. Dois dos algoritmos apresentados são baseados no filtro de Kalman para equações não-lineares, um centralizado, e outro descentralizado. Os outros quatro algoritmos, dos quais um deles é centralizado, são baseados na modelação do sistema de maneira a que este seja linear em relação ao estado dos veículos. Os algoritmos aqui apresentados foram escolhidos com base na sua simplicidade, robustez, e escalabilidade. Todas estas qualidades são importantes no que toca à escolha de algoritmos para navegação. Dado que os veículos podem estar a operar numa situação em que a qualidade da comunicação pode ser reduzida, esta tese foca-se em algoritmos cuja quantidade de comunicação é baixa. Por fim, apenas algoritmos capazes de lidarem com topologias de medição arbitrária foram considerados para apresentação, visto que um dos objetivos desta tese é a análise de técnicas versáteis o suficiente para o fazer. Estes algoritmos foram implementados num ambiente de simulação e alguns resultados são apresentados. Posteriormente, a qualidade das estimativas é investigada, calculando o seu *root-mean-squared-error* e o seu erro médio, averiguando o impacto da topologia de medição, nomeadamente, as vantagens e desvantagens de topologias cíclicas e acíclicas.

# Palavras Chave

Navegação descentralizada, filtro de Kalman, análise Monte Carlo, medidas de direção

# Contents

# List of Figures

# List of Tables

# Acronyms

**AUV** autonomous underwater vehicle

**CEKF** centralized extended Kalman filter

**CKF** centralized Kalman filter

**DEKF** decentralized extended Kalman filter

**DKF-FCS** decentralized Kalman filter with full covariance sharing

**DKF-PCS** decentralized Kalman filter with partial covariance sharing

**EKF** extended Kalman filter

**IKF** independently interconnected Kalman fitler

**MPC** model predictive control

**RMSE** root-mean-squared-error

**SLTI** static-gain estimator

**UAV** unmanned aerial vehicle

**UGV** unmanned ground vehicle

**UV** unmanned vehicle

# 1

# Introduction

## 1.1 Motivation

For the past two decades, there has been an increased interest and effort towards the development of fully autonomous unmanned vehicle (UV) systems. This is, in large part, due to demand for applications that are repetitive, unpleasant, or dangerous for human agents. Examples of domains of application are agriculture, transportation, scientific exploration, resource mining, waste management, search and rescue missions, surveillance, and plenty of other applications, many among them military [1]. Unmanned vehicles can be distinguished by their respective fields of application and operating environment (space, water and ground). Unmanned aerial vehicles (UAVs) and autonomous underwater vehicles (AUVs) are of special interest. The navigation problem for these differs from that of the unmanned ground vehicle (UGV) navigation problem, not only due to the extra spatial dimension, but also due to the fact that their respective operating fluid's momentum will have a much larger effect on the vehicles' kinematics than in the UGV problem.

Unmanned aerial vehicle technology has undeniable economic and social potential. It could have a major impact on many applications, even being considered as a major part of the development of

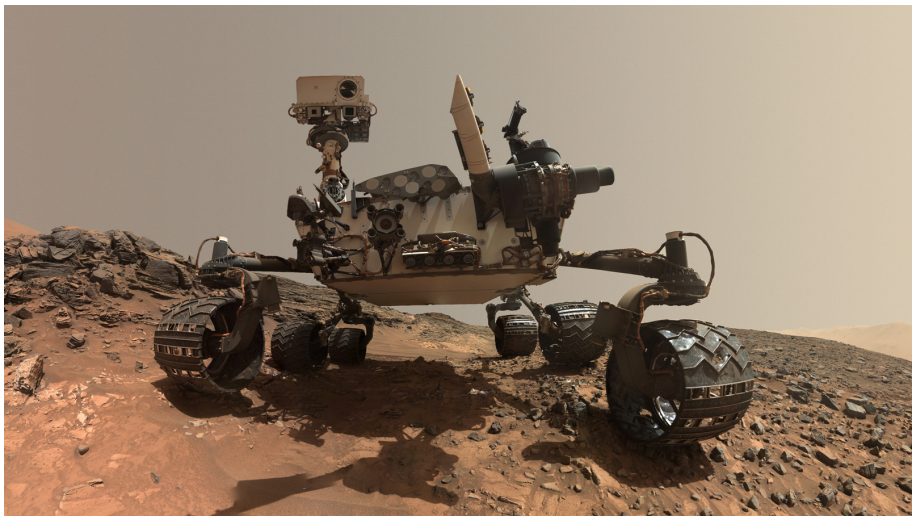


**Figure 1.1:** Mars rover, Curiosity. Image extracted from https://mars.nasa.gov/msl/home/

smart cities, playing an important role in transportation, environmental monitoring, security and delivery of goods [2]. In fact, some companies have already announced package delivery services using UAVs as delivery drones. UAVs have also successfully been used for crop monitoring [3] and aerobiological sampling [4], showing this technology's potential in agriculture as well. Furthermore, a Canadian company is currently using this type of vehicles in an effort to regenerate ecosystems on a global scale by using them to plant large amounts of trees [5]. Due to the fact that these vehicles can move freely in 3D-space, they can easily traverse rough terrain, making them a great potential asset when it comes to firefighting and disaster relief missions. Since these vehicles usually move at a relatively fast speed, and accidents have a high probability to result in their destruction, a solution for the navigation problem for these UVs must work at a fast rate and be very robust.

Similarly to the other types of vehicles, UGVs also have applications such as surveillance and payload delivery. They also have uses in agriculture, nuclear plant operations, firefighting efforts, and are regularly used for scientific purposes. As an example, the Mars rover presented in Fig. 1.1, Curiosity, is an example of a UGV, responsible for investigating the Gale crater on Mars autonomously.

Autonomous underwater vehicles also have many uses, though these are mostly for military and scientific purposes at present. Examples of these are oceanographic mapping, sea floor sample collection and underwater mine termination [6]. AUVs have been used by the oil and gas industry to map the seafloor, before starting underwater construction projects, and they can also be equipped with energy harvesting devices, such as triboelectric nanogenerators, that allow them to extract energy from the ocean, powering the devices needed for their operation [7]. One major aspect to have in consideration is that wireless communication underwater is severely limited between AUVs, which will limit the number of possible solutions depending on the mission at hand. On the other hand, these move rather slowly,

allowing for lower frequency solutions, compensating for the low communication capabilities underwater.

As UV technology improves, the number of concurrent operating UVs is expected to increase, giving rise to systems of increasing complexity, for which centralized approaches become impracticable. This is mainly due to the large computational resources and inter-UV communication necessary for this kind of solution, which some way or the other relies on all data being present on one central unit for computation of an updated navigation estimate for the whole formation. Unlike a fully centralized approach, decentralized approaches do not suffer from these problems, since each UV is responsible for computing its own state, relying mostly on measurements and limited communication with their immediate neighbors.

One of the main advantages of centralized solutions is the fact that all the data in the system is present for interpretation and manipulation within one central unit, meaning it is not limited in the information it has to work with. On the other hand, this is also why it can be problematic for large scale systems. While simpler to design, it is much more resource demanding and becomes even more demanding as more agents (more UVs) are added to the system. Also, if it fails, it can disturb the whole system dynamics, which, when composed of lots of agents, can be completely disruptive.

Another class of issues with centralized solutions has to do with communication. In underwater applications, where conventional wireless communication techniques do not work due to the severe attenuation of the electromagnetic radiation by the salty water, communication is usually performed using acoustic modems, which, as the name implies, converts data into pressure waves that travel through water at the speed of sound in this medium, i.e., around $1500\,\mathrm{m\,s^{-1}}$. So, not only is underwater communication more limited in terms of bandwidth, it also suffers from limitations in terms of latency, meaning centralized solutions for agents that are operating at great distances from each other become harder to implement and synchronize.

In decentralized solutions, agents are limited to work only with information local to them, that is, measurements to other agents in their proximity and communication with these. Thus, computation and communication requirements for any given agent scale mostly with the number of agents in its proximity, rather than with the whole system, as is the case in centralized approaches. This inherent difference between centralized and decentralized solutions means that the behavior of an agent should only directly affect the behavior of agents around it, naturally improving the robustness and scalability of the solution.

A major part of fully autonomous UV systems is the navigation aspect, by which each UV in a formation needs to be aware of its location, allowing for independent tackling of the control problem, which has a higher dependence on the mission at hand. The control objective might be, for example, keeping all the agents moving in a specified formation, or navigating all UVs to a specific location with arbitrary trajectories [8]. Regardless, the majority of the control solutions, whether centralized or decentralized, rely on accurate localization of the agents, which is an integral part of a well-operating UV system. This thesis tackles this issue, focusing on the research and development of decentralized navigation algo-

rithms, such that all the UVs in a formation can accurately estimate their position in the presence of disturbances and noisy measurements.

## 1.2 State of the art

While this thesis concerns itself mostly with the navigation aspect of a UV system, the control problem is much more widely studied in the literature, namely for UAVs, since these can easily be equipped with cheap satellite-based navigation systems, such as those provided by the Galileo or GPS systems. Nonetheless, given the similarities between the navigation and control problems, it is worthwhile to discuss some of the approaches in the literature to deal with the latter.

Most of the decentralized solutions to the control problem of guiding UVs from their initial positions to their respective goals are based on model predictive control (MPC) techniques. In [9], a control problem for arbitrary motion models and trajectories is defined, in which each agent solves its own optimal control problem at each time instance. The agents have two operating modes, one by which they try to accomplish their mission of arriving at a certain destination, and an emergency operating mode in case the optimal problem is not feasible, which in the authors' implementation simply attempts to stop the UV. The proposed solution is decentralized and admits time-varying interconnections between the UVs. However, as pointed out by the authors, the proposed scheme may lead to a deadlock or jerky behavior due to excessive operation in the emergency mode. The authors suggest some approaches to deal with this problem, such as introduction of right-of-way rules.

In [10], an MPC approach is also taken, albeit with a different cost function that depends on the time each UV takes to reach its goal. Collision avoidance and goal guidance objectives are taken care of by including them as constraints in the optimization problem, meaning there is no need for an emergency operating mode to deal with possible collisions. This solution requires that the first control command to give each UV be computed in a centralized manner; if a solution is found, then the problem is feasible and the decentralized sub-problems that are solved subsequently are also guaranteed to be feasible.

Another control problem that is often considered is that of formation stabilization, by which each UV must converge to specified locations given by range or bearing constraints to their neighbors. In order to deal with this problem, some solutions rely on the use of potential functions. For example, in [11], the authors take a graph theoretical approach, in which they exploit the unambiguity of the formation graph to generate a potential function with local minimum at the desired configuration, thus guarantying local asymptotic stability with collision avoidance to the desired formation. In [12], a similar approach is taken whereby the authors construct a feedback control law based on the bearing rigidity of the formation, guaranteeing local convergence to the desired spatial configuration if the bearing graph is rigid. Finally, [13] gives some more insight on the relationship between the rigidity of the underlying

formation graph and its consequences for the control problem.

A problem that is closely related to the navigation of UV systems is the localization of mobile sensor networks. Sensor networks consist of many spatially distributed devices that communicate between themselves and are equipped with sensors that allow them to make measurements about their neighbors and environment. Some nodes are capable of estimating their own position, known as *anchor* nodes. The main difference between sensor networks and UV systems is that the former is usually assumed to have a quite larger agent spatial density and typically does not assume a communication topology, allowing for data exchange between any two neighbors that are close enough.

A very interesting solution to the localization problem for mobile and static sensor networks is presented in [14]. It does not rely on range or bearing measurements and, instead, utilizes the limited radio range of the nodes to estimate the position of other nodes in the network. The solution consists of having each node maintain a polygon representing the set of its possible locations (such that the node's position is guaranteed to be in its polygon). Nodes then dilate their polygons by the radio range and broadcast the dilated versions of their polygons. This way, any node that has heard the sender's message must be within the radio range of the broadcasting node. Listening nodes then intersect their respective dilated polygons with the received ones, resulting in a smaller convex polygon. Since the listening node must be included in the dilated polygon of the broadcasting node, this new polygon represents the listening node's new set of possible locations. Over time, each node converges to a lower bound on its position estimate, which depends on parameters such as anchor density and number of sides of the polygons.

A very different approach to the same problem is presented in [15]. The solution is based on rewriting the inter-node distance measurements as barycentric coordinate weights, and then using these to update the position estimates of the nodes. The only requirement is that the updating node must lie inside the convex hull of its neighbors, meaning it must communicate with the appropriate number of valid nodes whenever it performs an update. The advantage of using this representation is that it allows for the writing of an update rule with a sub-stochastic system matrix, such that with only one seed node, and given some easily satisfied conditions, the whole mobile network can eventually be localized, regardless of the initial estimates or number of nodes.

In order to deal with the navigation problem considered in this work, the most commonly used methods to solve the UV navigation problem are based on cooperative probabilistic techniques, from which Kalman and particle filters emerge as two possible solutions.

The most attractive aspect of particle filters is the multi-modality of the resulting particle distributions, which relies on the amount of particles itself. However, since communication might be limited between UVs, it is not practical to transmit all the information and thus parametrization of the particle distribution must first be performed using, for example, a Gaussian mixture model, which allows for keeping the multi-modal nature of the particle distribution. In [16], the author discusses this and proposes an algorithm

which forward propagates measurement data in time, coupling it with newer data to create hypotheses for the position of agents; then, using dead-reckoning information, it finds the most likely path through the sets of hypotheses. This algorithm inherently maintains several hypotheses about the state of each UV, updating them whenever new measurements are obtained.

Many of the decentralized approaches rely on the decoupling of the centralized Kalman filter equations such that each agent estimates only its state and covariance, communicating with neighbors to acquire the necessary quantities for updating its belief. A major complication when building a decentralized Kalman filter that is equivalent to its centralized counterpart, however, is the difficulty in estimating the correlation between different agents that do not communicate with each other. The work [17] presents some of the solutions in the literature to deal with this problem, giving some insight on the need for keeping track of the cross-correlation between agents when this type of decentralized solutions are considered. Ignoring the cross-correlation between agents will cause these to become overconfident in their estimates, which might cause the filter to ignore updates and diverge. Besides this, the cross-correlation between agents allow for the state estimate of all of them to be improved in a cooperative manner, meaning there are some clear benefits to correctly keeping track of it. A more recent solution that will be studied in some detail in Section 3.2 is presented in [18].

Kalman filter solutions (both centralized and decentralized) that rely directly on the range or bearing measurements have the additional complication that these models are nonlinear, meaning the extended Kalman filter (EKF) must be used. Because of this, the initial guess of the UV states must be relatively close to their true values, otherwise the Kalman filter may diverge.

Another way of solving the navigation problem, rather than trying to find a centralized equivalent approach, is to design decentralized observers and then ensure the interconnection between these systems leads to the correct solution. An example of this approach is given in [19], where a linear time-varying system that is globally observable, under certain restrictions, is created by defining an artificial output using bearing measurements. Then, a Kalman filter with globally exponentially stable error dynamics is applied to the subsystem and globally convergent dynamics for the whole formation are guaranteed for the case of acyclic tiered formations. In the considered topology, the UVs in the first tier have access to absolute position measurements, and UVs in the tiers below these only take bearing measurements and communicate with UVs in the tier directly above. This means that if the first tier estimators converge to their true values, the lower tier ones will eventually converge to their true values as well.

A similar approach is proposed in [20], where the authors augment the state vector of each UV to include the range to other UVs, and then augment the output to include range constraints between UVs participating in the measurement. While this approach has globally convergent observer error dynamics and allows for correct tracking of inter-UV ranges, which could be useful in dealing with the

collision avoidance problem discussed earlier, it is developed for continuous-time and as such can rely too heavily on communication.

In [21], two solutions for the navigation of a single UV based on bearing measurements to three stationary sources are proposed, both achieving globally convergent observer error dynamics given some easily satisfied conditions. One of these allows for keeping track of the range to these sources and is based on state augmentation; the other is based on the definition of an artificial output which is equivalent to the original nonlinear bearing output. This solution could easily be extended to mobile sources by allowing them to communicate their velocities to measuring UVs, thus serving to localize a formation in which all follower UVs can communicate with the sources.

Some other approaches are based on iterative optimization techniques. In [22], a distributed solution based on gradient optimization is proposed, in which the nodes start by exchanging information with their neighbors to reach a consensus on a step-size (geometrically fast) and then optimize according to the agreed upon value. This process is then repeated until some stopping criterion is met, making this method very reliant on communication. A more recent alternative is proposed in [23], in which the iterative minimization of a majorizer of a cost function is performed. This cost function has a form that allows it to be distributively minimized by the agents in the network. Each node broadcasts its location to its neighbors, which then compute a new estimate for their location. This solution achieves better results than a centralized EKF solution and is provably convergent to a stationary point, whereas the EKF can be unstable depending on the initial conditions.

In summary, there are many solutions to the problem considered in this work. The most common ones in the field of cooperative navigation are based on probabilistic techniques, such as the Kalman and particle filters, while others rely on building decentralized observers and then making interconnections between them, such that each observer ends up converging to a solution. Lastly, there are also techniques based on iterative optimization.

In this work, some of these approaches will be detailed such as to give an overview of the current state of the research in the field of decentralized navigation. This will be one of the main contributions of this work, as well as a comparison between these state of the art approaches, analyzing their overall performance under different types of measurement topologies by obtaining Monte Carlo results.

## 1.3 Objectives

The main objective of this work is the study of cooperative navigation techniques which allow for arbitrary measurement topologies. Additionally, this work intends to investigate the effect of different topologies on the estimation performance of the studied approaches, namely, the effect of having an acyclical versus a cyclical measurement topology. As a third objective, this work intends to provide a general overview

of some of the approaches in the literature.

## 1.4 Contributions

Several algorithms based on depth and bearing measurements are presented in this work. Two of these are EKF-based, and the remaining ones are based on creating artificial outputs, such that the resulting system model is linear relative to the state, allowing it to be studied using standard observability techniques.

In order to accomplish the main two objectives of this work, a simulation environment that is versatile enough to handle different missions was created. This allowed for designing missions and specifying measurement topologies for the implemented algorithms, from which a Monte Carlo analysis was made. This analysis revealed that, if the agents do not take into account enough information pertaining to the measurement topology, such as cross-measurement information, the presence of cycles can be very detrimental to the quality of the estimates, due to the reintroduction of the estimation error into the algorithms. The third objective is accomplished in Section 1.2.

During the development of this work, [24] was published and presented in the OCEANS 2021 San Diego - Porto conference, where a Monte Carlo study comparing the EKF-based algorithms presented in Section 3 was made.

## 1.5 Organization

This work is organized as follows. In Chapter 1, the decentralized navigation problem is motivated and some of current literature reviewed. In Chapter 2, the mathematical description of the problem is formalized, and, in Chapters 3 and 4, algorithms to solve it are presented. In Chapter 3, approaches based on the direct use of the extended Kalman filter are described, and, in Chapter 4, artificial measurements which are linear relative to the state of the system are considered, which allows for treating the observability of the resulting system model using regular linear system analysis techniques. Finally, in Chapter 5, the performance of the algorithms presented in this work is evaluated and compared via Monte Carlo results.

# 2

# Problem Description

## 2.1 Notation

Here, the notation adopted throughout this work is briefly defined. Vectors and matrices are represented in bold and their scalar entries are superscripted, such that $\mathbf{v} = (\mathbf{v}^i) \in \mathbb{R}^n$ and $\mathbf{A} = (\mathbf{A}^{ij}) \in \mathbb{R}^{m \times n}$. The identity and zero square matrices of size $n$ are represented as $\mathbf{I}_n$ and $\mathbf{0}_n$, respectively. If the zero matrix is not square, then it is represented as $\mathbf{0}_{m \times n} \in \mathbb{R}^{m \times n}$. The transpose operator is represented by $(\cdot)^T$ and $\mathrm{diag}(\cdot)$ builds a diagonal matrix from its inputs. Additionally, the Kronecker product is denoted by the symbol $\otimes$, such that, for $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{B} \in \mathbb{R}^{p \times q}$, one has

$$A \otimes B \coloneqq \begin{bmatrix} \mathbf{A}^{11}\mathbf{B} & \cdots & \mathbf{A}^{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ \mathbf{A}^{m1}\mathbf{B} & \cdots & \mathbf{A}^{mn}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{pm \times qn}.$$

If $\mathcal{S}$ denotes a set, $|\mathcal{S}|$ represents its cardinality, i.e., the number of elements in $\mathcal{S}$.

## 2.2 Problem statement

Consider a set of UVs, numbered from $1$ to $N$, operating in a 3D environment such that the movement of each agent in the inertial frame, $\{I\}$, is described by

$$\begin{cases} \dot{\mathbf{p}}_i(t) = \mathbf{R}_i(t)\mathbf{v}_{r_i}(t) + \mathbf{v}_{f_i}(t) \\ \dot{\mathbf{v}}_{f_i}(t) = \mathbf{0}_3 \end{cases},$$

where $\mathbf{p}_i(t) = \begin{bmatrix} \mathbf{p}_i^x(t) & \mathbf{p}_i^y(t) & \mathbf{p}_i^z(t) \end{bmatrix}^T$ represents the position of the $i^{\text{th}}$ UV, $\mathbf{R}_i \in SO(3)$ is this agent's rotation matrix, that transforms coordinates in its body frame to coordinates in the inertial frame, and $\mathbf{v}_{r_i}(t)$ is its local velocity relative to the fluid it is operating in, represented in the UV's body frame. Note that, in practical terms, $\mathbf{v}_{f_i}$ is a function of both time and the position, $\mathbf{p}_i$, of the agent. However, in nominal terms, it will be assumed to be constant. In practice, by appropriate tuning of the parameters of the filtering solution, it is possible to estimate slowly time-varying quantities.

Since solutions are usually implemented on a digital computer, the continuous-time kinematics must be discretized, resulting in

$$\begin{cases} \mathbf{p}_i(t_{k+1}) = \mathbf{p}_i(t_k) + T\mathbf{v}_{f_i}(t_k) + \mathbf{u}_i[k] \\ \mathbf{v}_{f_i}(t_{k+1}) = \mathbf{v}_{f_i}(t_k) \end{cases}, \tag{2.1}$$

where

$$\mathbf{u}_i[k] = \int_{t_k}^{t_{k+1}} \mathbf{R}_i(t)\mathbf{v}_{r_i}(t)dt \tag{2.2}$$

and $T$ is the sampling time. In state-space form, letting the state of the $i^{\text{th}}$ agent be defined as

$$\mathbf{x}_i[k] \coloneqq \begin{bmatrix} \mathbf{p}_i(t_k) \\ \mathbf{v}_{f_i}(t_k) \end{bmatrix}, \tag{2.3}$$

and following (2.1), the motion model of an agent is given by

$$\mathbf{x}_i[k+1] = \mathbf{A}\mathbf{x}_i[k] + \mathbf{B}\mathbf{u}_i[k],$$

where

$$\mathbf{A} \coloneqq \begin{bmatrix} \mathbf{I}_3 & T\mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \tag{2.4}$$

and

$$\mathbf{B} \coloneqq \begin{bmatrix} \mathbf{I}_3 \\ \mathbf{0}_3 \end{bmatrix}. \tag{2.5}$$

The UVs are equipped with sensors that enable them to make measurements about themselves, such as depth and orientation measurements; and about their neighbors, such as range or bearing measurements. In addition to this, they are also capable of some degree of communication between

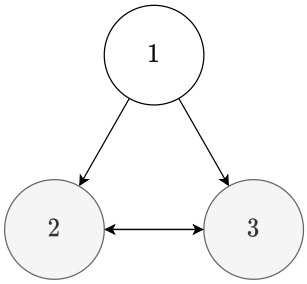



**Figure 2.1:** Example measurement graph.

themselves, enabling them to share quantities such as position estimates with their neighbors or algorithm specific data.

At this point, it is assumed that if the $j^{\text{th}}$ UV is capable of taking measurements about the $i^{\text{th}}$ agent, then there is a bidirectional communication link between the two. The formation's measurement configuration can then be represented with a single directed graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of UVs and $\mathcal{E}$ is the set of directed edges, representing measurement information flow. The $j^{\text{th}}$ UV takes measurements about the $i^{\text{th}}$ UV if there is a directed edge leaving node $i$ and entering node $j$, i.e., if there is an edge $e_{ij} = (i, j)$. The neighbor set of the $i^{\text{th}}$ UV is defined as the set of UVs that it takes measurements about, i.e., $\mathcal{N}_i = \{j : (j, i) \in \mathcal{E}\}$. It is also assumed that $\mathcal{V}$ can be further separated into two disjoint subsets, $\mathcal{V}_\mathcal{L}$ and $\mathcal{V}_\mathcal{F}$, such that $\mathcal{V}_\mathcal{L} \cup \mathcal{V}_\mathcal{F} = \mathcal{V}$. The set $\mathcal{V}_\mathcal{L}$ contains the so-called *leader* UVs, which are assumed to able to estimate their position with some accuracy by themselves, and the set $\mathcal{V}_\mathcal{F}$ contains the *follower* UVs, that must estimate their state based on measurements about their neighbors and communication with them. Note that, according to this definition for the measurement graph, the leader agents have no neighbors, since they do not take measurements about any other UVs.

**Example 1.** Consider the measurement graph, $\mathcal{G}$, presented in Fig. 2.1. In this example, the leader set is $\mathcal{V}_\mathcal{L} = \{1\}$, and the follower set is $\mathcal{V}_\mathcal{F} = \{2, 3\}$, which is graphically represented with grayed out nodes. As per the previous definitions, the $2^{\text{nd}}$ UV takes measurements about and receives information from agents 1 and 3. Likewise for the $3^{\text{rd}}$ UV, which takes measurements about agents 1 and 2, such that $\mathcal{N}_3 = \{1, 2\}$. The neighbor sets of UVs $1$ and $2$ are $\mathcal{N}_1 = \emptyset$ and $\mathcal{N}_2 = \{1, 3\}$, respectively.

Consider now that the UVs in $\mathcal{V}_\mathcal{F}$ are equipped with pressure gauges and attitude and heading reference systems, so that they can determine their own depth and orientation, as well some device that allows them to take bearing measurements to neighboring UVs. As an example, an ultra-short baseline acoustic positioning system readily gives bearing measurements [25]. Then, according to the measurement graph, at time $t_k$, in addition to its noisy attitude measurement, given by its rotation matrix,

UV $3$ has access to the following information

$$\begin{cases} z_3(t_k) = \mathbf{p}_3^z(t_k) + e_1(t_k) \\ \theta_{31}(t_k) = {}^{\mathcal{B}}\theta(\mathbf{p}_3(t_k), \mathbf{p}_1(t_k)) + e_2(t_k) \\ \phi_{31}(t_k) = {}^{\mathcal{B}}\phi(\mathbf{p}_3(t_k), \mathbf{p}_1(t_k)) + e_3(t_k) \\ \theta_{32}(t_k) = {}^{\mathcal{B}}\theta(\mathbf{p}_3(t_k), \mathbf{p}_2(t_k)) + e_4(t_k) \\ \phi_{32}(t_k) = {}^{\mathcal{B}}\phi(\mathbf{p}_3(t_k), \mathbf{p}_2(t_k)) + e_5(t_k) \\ \hat{\mathbf{p}}_1(t_k) = \mathbf{p}_1(t_k) + \mathbf{e}_6(t_k) \\ \hat{\mathbf{p}}_2(t_k) = \mathbf{p}_2(t_k) + \mathbf{e}_7(t_k) \end{cases},$$

where $\mathbf{p}_3$ represents the position of UV $3$, $\hat{\mathbf{p}}_j$ is a position estimate of UV $j \in \{1, 2\}$ that is communicated to UV $3$, ${}^{\mathcal{B}}\theta(\mathbf{p}_i, \mathbf{p}_j)$ and ${}^{\mathcal{B}}\phi(\mathbf{p}_i, \mathbf{p}_j)$ are functions that return the noiseless bearing angles, measured by an agent with index $i$ about another agent with index $j$, represented in agent $i$'s body frame. The quantities $e_1$, $e_2$, $e_3$, $e_4$, $e_5$, $\mathbf{e}_6$ and $\mathbf{e}_7$ are unknown errors terms, due to, for instance, measurement noise or estimation errors.

The problem of decentralized navigation considered in this work is then to estimate the position, $\mathbf{p}_i$, of each UV, as well as its local fluid velocity, $\mathbf{v}_{f_i}(t)$, constrained by the fact that the agents only have access to local information that they can capture, be it through measurements or limited communication with their neighbors. In addition to the decentralized navigation approaches that will be discussed, centralized solutions are presented as well, in order to establish a baseline for comparison with their decentralized alternatives.

# 3

# Approaches Based on the Extended Kalman Filter

The most straightforward approach to the navigation problem is by using the measurements captured by the UVs directly, employing an extended Kalman filter, which requires the linearization of the observation model. An issue with EKF-based approaches is that, due to their linearization step, the initial state estimate of each UV must be close enough to the true state of its respective agent and might require fine tuning of the filter parameters, otherwise, convergence to the true solution is not guaranteed. This reduces the time efficiency of missions since an initial setup (using GPS, for example) is necessary, which might require all agents to be at the surface to do so.

## 3.1 Centralized extended Kalman filter

While the centralized extended Kalman filter (CEKF) has access to all data, this comes with some serious drawbacks, such as heavy reliance on communication between UVs and lack of scalability. In

some cases, the implementation of a fully centralized approach can become very cumbersome or even unfeasible. Since all the data must be available at a single unit for computation, some information might need to travel through long distances, thus resulting in the introduction of a delay into the system, which might not be easy to deal with. Additionally, the amount of information shared and the high correlation between agents might make the algorithm more sensitive to outliers, which may even cause it to diverge. Regardless, centralized approaches have the potential to give the "best" estimates, and, as such, the CEKF is presented here as a baseline for comparison with its decentralized counterpart, presented in Section 3.2.

### 3.1.1 Motion updates

Define the whole state as

$$\mathbf{x}[k] \coloneqq \begin{bmatrix} \mathbf{x}_1[k] \\ \vdots \\ \mathbf{x}_N[k] \end{bmatrix} \in \mathbb{R}^{6N},$$

where each $\mathbf{x}_i$ is defined as in (2.3), representing the position and local fluid velocity of each UV. Then, considering $\mathbf{A}$ and $\mathbf{B}$ as defined in (2.4) and (2.5), the complete system motion model is given by

$$\mathbf{x}[k+1] = \mathbf{A}_c\mathbf{x}[k] + \mathbf{B}_c\mathbf{u}[k],$$

where

$$\begin{cases} \mathbf{A}_c = \mathbf{I}_N \otimes \mathbf{A} \\ \mathbf{B}_c = \mathbf{I}_N \otimes \mathbf{B} \\ \mathbf{u}[k] = \begin{bmatrix} \mathbf{u}_1[k] \\ \vdots \\ \mathbf{u}_N[k] \end{bmatrix} \end{cases}, \tag{3.1}$$

with $\mathbf{u}_i[k]$ defined in (2.2), and $N = |\mathcal{V}|$ is the number of agents. The prediction equations for the CEKF are then

$$\begin{cases} \hat{\mathbf{x}}[k+1|k] = \mathbf{A}_c\hat{\mathbf{x}}[k] + \mathbf{B}_c\mathbf{u}[k] \\ \mathbf{\Sigma}[k+1|k] = \mathbf{A}_c\mathbf{\Sigma}[k|k]\mathbf{A}_c^T + \mathbf{Q}_c \end{cases} \tag{3.2}$$

where $\hat{\mathbf{x}}$ and $\mathbf{\Sigma}$ are the state estimate mean and covariance matrix, respectively, and $\mathbf{Q}_c$ is the centralized process noise covariance matrix. The process noise of each agent can be independently parameterized via $\mathbf{Q}_i$ and $\mathbf{Q}_c$ is then built as $\mathbf{Q}_c = \mathrm{diag}(\mathbf{Q}_1, \ldots, \mathbf{Q}_N)$.

### 3.1.2 Measurement updates

Let $\mathbf{y}_i[k] = \mathbf{h}_i(\mathbf{x}[k])$ be a measurement taken by an agent with index $i$, and let the complete measurement vector, $\mathbf{y}[k]$, be the concatenation of all the individual measurement vectors, as in

$$\mathbf{y}[k] = \mathbf{h}(\mathbf{x}[k]) = \begin{bmatrix} \mathbf{h}_1(\mathbf{x}[k]) \\ \vdots \\ \mathbf{h}_N(\mathbf{x}[k]) \end{bmatrix}.$$

In order to perform the update step of the CEKF, the Jacobian of the measurement model must be computed, as in

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \partial\mathbf{h}_1/\partial\mathbf{x} \\ \vdots \\ \partial\mathbf{h}_N/\partial\mathbf{x} \end{bmatrix} = \begin{bmatrix} \partial\mathbf{h}_1/\partial\mathbf{x}_1 & \cdots & \partial\mathbf{h}_1/\partial\mathbf{x}_N \\ \vdots & \ddots & \vdots \\ \partial\mathbf{h}_N/\partial\mathbf{x}_1 & \cdots & \partial\mathbf{h}_N/\partial\mathbf{x}_N \end{bmatrix}, \tag{3.3}$$

where the discrete-time dependency was omitted for readability. Since the real state is unknown, the Jacobian, $\mathbf{J} = \mathbf{J}(\mathbf{x}[k])$, is approximated by $\hat{\mathbf{J}} = \mathbf{J}(\hat{\mathbf{x}}[k+1|k])$, hence one of the reasons why a good enough initial state estimate is necessary.

Since EKF-based approaches allow for arbitrary measurement models, the general update equations will be presented here, and some specific measurement models will be described in the following section. Upon receiving measurements, the CEKF update equations are given by

$$\begin{cases} \hat{\mathbf{x}}[k+1|k+1] = \hat{\mathbf{x}}[k+1|k] + \mathbf{K}\left(\mathbf{y}[k+1] - \hat{\mathbf{y}}[k+1]\right) \\ \boldsymbol{\Sigma}[k+1|k+1] = \left(\mathbf{I}_{6N} - \mathbf{K}\hat{\mathbf{J}}\right)\boldsymbol{\Sigma}[k+1|k] \end{cases},$$

where $\mathbf{K} = \boldsymbol{\Sigma}[k+1|k]\hat{\mathbf{J}}^T\left(\hat{\mathbf{J}}\boldsymbol{\Sigma}[k+1|k]\hat{\mathbf{J}}^T + \mathbf{R}_c\right)^{-1}$ is the Kalman gain and $\mathbf{R}_c$ is the centralized measurement vector noise covariance matrix. Lastly, $\hat{\mathbf{y}}[k+1] = \mathbf{h}(\hat{\mathbf{x}}[k+1|k])$ is the expected value of the measurement vector, given the current state estimate.

### 3.1.3 Measurement models

Some common measurement models will now be introduced. In particular, equations for position, depth, range and bearing measurements are presented and the Jacobians of these models are computed. For ease of representation, it is assumed that the measuring UV, with index $i$, is the one with that occupies the first positions of the total state vector, such that $\mathbf{x} = \begin{bmatrix}\mathbf{x}_i^T & \mathbf{x}_j^T & \dots & \mathbf{x}_N^T\end{bmatrix}^T$. In case this is not verified, the terms must be shifted to their correct positions according to their indices.

Note that any combination of these measurements can be considered simultaneously through appropriate concatenation of measurement vectors and Jacobian matrices. This is a major strength of the Kalman filter, which easily integrates all this information and produces a single estimate with increased certainty, a process known as sensor fusion.

#### 3.1.3.A Position measurements

If the UV making a measurement has direct access to position measurements $\mathbf{y}_i = \mathbf{h}_i(\mathbf{x}) = \mathbf{p}_i(t_k)$, its relevant part in (3.3) is given by

$$\frac{\partial \mathbf{h}_i}{\partial \mathbf{x}}(\mathbf{x}) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times(6N-6)} \end{bmatrix}.$$

It will often be assumed that the only UVs capable of acquiring this kind of measurements are leader UVs, which usually remain at the top or near the surface, where position measurements are readily available through satellite-based navigation systems.

#### 3.1.3.B Depth measurements

Like position measurements, depth measurements do not involve other UVs. The model is given by $y = h(\mathbf{x}) = \mathbf{p}_i^z(t_k)$ and its Jacobian is then simply given by

$$\frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & \mathbf{0}_{1\times(6N-6)} \end{bmatrix}.$$

While these measurements are simple, geometrically, they remove a degree of freedom from the possible positions of the $i^{\text{th}}$ UV, namely in the $z$ direction. As such, coupling a depth measurement with a bearing measurement to another UV will fix the possible positions of this agent to a single point, given the position of the $j^{\text{th}}$ UV, and given that they do not lie on the same horizontal plane. Due to the simplicity and cheap cost of depth gauges that provide this type of measurements, it is often be assumed that UVs are equipped with this kind of sensor.

#### 3.1.3.C Bearing measurements

If the $i^{\text{th}}$ UV takes a bearing measurements about another UV with index $j$, this measurement is modeled using

$$\mathbf{h}_{ij}(\mathbf{x}) = \begin{bmatrix} \theta(\mathbf{p}_i, \mathbf{p}_j) \\ \phi(\mathbf{p}_i, \mathbf{p}_j) \end{bmatrix} = \begin{bmatrix} \operatorname{atan2}\left(\mathbf{p}_j^z - \mathbf{p}_i^z, \sqrt{(\mathbf{p}_j^x - \mathbf{p}_i^x)^2 + (\mathbf{p}_j^y - \mathbf{p}_i^y)^2}\right) \\ \operatorname{atan2}\left(\mathbf{p}_j^y - \mathbf{p}_i^y, \mathbf{p}_j^x - \mathbf{p}_i^x\right) \end{bmatrix},$$

where the angles $\theta$ and $\phi$ are represented in the inertial frame. Bearing measurements are usually measured in the UV's body frame, however, it will be assumed that the UV knows its orientation and, as such, can rotate the bearing measurement so that it is represented in $\{I\}$. For this measurement, one has

$$\frac{\partial \mathbf{h}_{ij}}{\partial \mathbf{x}_i}(\mathbf{x}_i, \mathbf{x}_j) = \begin{bmatrix} \dfrac{(\mathbf{p}_j^z - \mathbf{p}_i^z)(\mathbf{p}_j^x - \mathbf{p}_i^x)}{L_{xy} L_{xyz}^2} & \dfrac{(\mathbf{p}_j^z - \mathbf{p}_i^z)(\mathbf{p}_j^y - \mathbf{p}_i^y)}{L_{xy} L_{xyz}^2} & -\dfrac{L_{xy}}{L_{xyz}^2} & \mathbf{0}_{1\times 3} \\ \dfrac{\mathbf{p}_j^y - \mathbf{p}_i^y}{L_{xy}^2} & \dfrac{\mathbf{p}_j^x - \mathbf{p}_i^x}{L_{xy}^2} & 0 & \mathbf{0}_{1\times 3} \end{bmatrix},$$

with

$$\begin{cases} L_{xy} = \sqrt{(\mathbf{p}_j^x - \mathbf{p}_i^x)^2 + (\mathbf{p}_j^y - \mathbf{p}_i^y)^2} \\ L_{xyz} = \sqrt{(\mathbf{p}_j^x - \mathbf{p}_i^x)^2 + (\mathbf{p}_j^y - \mathbf{p}_i^y)^2 + (\mathbf{p}_j^z - \mathbf{p}_i^z)^2} \end{cases},$$

and

$$\frac{\partial \mathbf{h}_{ij}}{\partial \mathbf{x}_j}(\mathbf{x}_i, \mathbf{x}_j) = -\frac{\partial \mathbf{h}_{ij}}{\partial \mathbf{x}_i}(\mathbf{x}_i, \mathbf{x}_j).$$

These measurements remove two degrees of freedom from the possible positions of the measuring agent, constraining it to a line connecting both participating agents.

## 3.2 Decentralized extended Kalman filter

As mentioned in section 1.2, this algorithm is presented in depth in [18]. The main idea behind this approach is that, rather than attempt to derive a decentralized estimator that is equivalent to the centralized Kalman filter presented in section 3.1, the decentralized extended Kalman filter (DEKF) tries to approximate its centralized counterpart as best as possible while keeping communication to a minimum.

### 3.2.1 Motion updates

Upon receiving measurements that concern more than one agent, that is, relative measurements, the CEKF introduces coupling between the different state estimates, such that the second equation of (3.2) cannot be distributed among the UVs without extensive communication. One way around this issue would be to simply ignore the coupling between the UVs, but this would lead to an overconfident estimation, which could cause the filter to ignore updates and diverge. However, while the covariance prediction equation does affect the cross-covariance terms, the predicted covariance does not have any effect on the predicted state estimate. The ultimate goal of having the cross-covariance terms, is to update the state estimates upon the capture of measurements. So, as long as the UVs can keep track of the correct cross-covariance terms to other UVs between measurement updates, there is no difference in the prediction behavior of the algorithm.

Consider the state of the $i^{\text{th}}$ agent, $\mathbf{x}_i$, defined as in (2.3), and denote its filtered estimate and covariance by $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{\Sigma}}_{ii}$, respectively. Note that the DEKF approximates the CEKF, thus, the covariances of each agent and their cross-covariances to other agents will not be exact, hence the chosen hat notation. Consider then the factorization of the cross-covariance between agents $i$ and $j$, $\hat{\mathbf{\Sigma}}_{ij}$, such that

$$\hat{\mathbf{\Sigma}}_{ij}[k] = \hat{\mathbf{\Phi}}_{ij}[k]\hat{\mathbf{\Phi}}_{ji}^T[k]. \tag{3.4}$$

Let each agent carry its estimated belief, $\mathcal{B}_i \coloneqq \{\hat{\mathbf{x}}_i, \hat{\mathbf{\Sigma}}_{ii}\}$, and cross-covariance factor, $\hat{\mathbf{\Phi}}_{ij}$, between

itself and other agents it has knowledge of, i.e. $\hat{\mathbf{\Phi}}_{ij}$ for all $j \in \mathcal{N}_i$. The corresponding CEKF prediction equations for agent $i$, which account for its motion, are given by

$$\begin{cases} \hat{\mathbf{x}}_i[k+1|k] = \mathbf{A}\hat{\mathbf{x}}_i[k|k] + \mathbf{B}\mathbf{u}_i[k] \\ \mathbf{\Sigma}_{ii}[k+1|k] = \mathbf{A}\mathbf{\Sigma}_{ii}[k|k]\mathbf{A}^T + \mathbf{Q}_i \\ \mathbf{\Sigma}_{ij}[k+1|k] = \mathbf{A}\mathbf{\Sigma}_{ij}[k|k] \end{cases}, \tag{3.5}$$

leaving the remaining terms $\hat{\mathbf{x}}_j, \mathbf{\Sigma}_{jj}$ for all $j \neq i$, unchanged. So, following [18], by letting UV $i$ update its cross-covariance factor to another UV $j$ through

$$\hat{\mathbf{\Phi}}_{ij}[k+1|k] = \mathbf{A}\hat{\mathbf{\Phi}}_{ij}[k|k] \tag{3.6}$$

when performing prediction steps, when they meet, their reconstructed cross-covariance is given by

$$\begin{aligned} \hat{\mathbf{\Sigma}}_{ij}[k+1|k] &= \mathbf{A}\hat{\mathbf{\Phi}}_{ij}[k|k]\hat{\mathbf{\Phi}}_{ji}[k|k]^T \\ &= \mathbf{A}\hat{\mathbf{\Sigma}}_{ij}[k|k] \\ &= \mathbf{\Sigma}_{ij}[k+1|k], \end{aligned}$$

if it holds that $\hat{\mathbf{\Sigma}}_{ij}[k|k] = \mathbf{\Sigma}_{ij}[k|k]$. In general, $\hat{\mathbf{\Sigma}}_{ij}[k|k] \neq \mathbf{\Sigma}_{ij}[k|k]$, however, what is important is that, since all terms are available, the prediction step of the CEKF can be reproduced exactly at each agent in a decentralized way while requiring no communication, thus resulting in no loss of estimation capabilities with respect to this step. All UVs then predict their beliefs and cross-covariance factors to other agents according to the first two equations of (3.5) and to (3.6), substituting $\mathbf{x}_i$ and $\mathbf{\Sigma}_{ii}$ by their estimated state and covariance matrix, $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{\Sigma}}_{ii}$.

### 3.2.2 Measurement updates

In order to simplify the solution and make it more flexible, an asynchronous behavior in which each UV makes measurements independently is considered. This would be equivalent to considering only a measurement at a time in the CEKF measurement model and performing several successive updates at each time step. The DEKF makes a distinction between *private* measurements, those that only concern one agent at a time, and *relative* measurements, which concern the participating UVs.

#### 3.2.2.A Private measurements

Consider the centralized update equations. If, at time $t = t_{k+1}$, the $i^{\text{th}}$ UV makes a private measurement, $\mathbf{y}_i[k+1] = \mathbf{h}_i(\mathbf{x}_i[k+1])$, its update equations are given by

$$\begin{cases} \hat{\mathbf{x}}_i[k+1|k+1] = \hat{\mathbf{x}}_i[k+1|k] + \mathbf{K}_i \left(\mathbf{y}_i[k+1] - \mathbf{h}_i\left(\hat{\mathbf{x}}_i[k+1|k]\right)\right) \\ \mathbf{\Sigma}_{ii}[k+1|k+1] = \left(\mathbf{I}_6 - \mathbf{K}_i \hat{\mathbf{J}}_i\right) \mathbf{\Sigma}_{ii}[k+1|k] \end{cases}. \tag{3.7}$$

The other UVs would update their state and covariance matrices according to

$$\begin{cases} \hat{\mathbf{x}}_j[k+1|k+1] = \hat{\mathbf{x}}_j[k+1|k] + \mathbf{K}_j \left(\mathbf{y}_i[k+1] - \mathbf{h}_i(\hat{\mathbf{x}}_i[k+1|k])\right) \\ \mathbf{\Sigma}_{jj}[k+1|k+1] = \mathbf{\Sigma}_{jj}[k+1|k] - \mathbf{K}_j \hat{\mathbf{J}}_i \mathbf{\Sigma}_{ij}[k+1|k] \end{cases}, \tag{3.8}$$

and their cross-covariance to other UVs update according to

$$\mathbf{\Sigma}_{ij}[k+1|k+1] = \left(\mathbf{I} - \mathbf{K}_i \hat{\mathbf{J}}_i\right) \mathbf{\Sigma}_{ij}[k+1|k], \tag{3.9}$$

with

$$\begin{cases} \hat{\mathbf{J}}_i = \dfrac{\partial \mathbf{h}_i}{\partial \mathbf{x}_i}\left(\hat{\mathbf{x}}_i[k+1|k]\right) \\ \mathbf{K}_i = \mathbf{\Sigma}_{ii}[k+1|k] \hat{\mathbf{J}}_i^T (\hat{\mathbf{J}}_i \mathbf{\Sigma}_{ii}[k+1|k] \hat{\mathbf{J}}_i^T + \mathbf{R}_i)^{-1} \\ \mathbf{K}_j = \mathbf{\Sigma}_{ji}[k+1|k] \hat{\mathbf{J}}_i^T (\hat{\mathbf{J}}_i \mathbf{\Sigma}_{ii}[k+1|k] \hat{\mathbf{J}}_i^T + \mathbf{R}_i)^{-1} \end{cases},$$

where $\mathbf{R}_i$ is measurement noise covariance matrix of UV $i$. The update (3.7) can be computed at the $i^{\text{th}}$ UV using only local information. However, (3.8) can only be computed if one allows communication between agents, since it would require cross-covariance reconstruction. Since this is not desirable, (3.8) is not performed and the values remain unchanged.

In order to replicate the update (3.9), following a similar reasoning to the one presented in Section 3.2.1, and as done in [18], the $i^{\text{th}}$ UV can update its cross-covariance factors to other UVs using

$$\hat{\mathbf{\Phi}}_{ij}[k+1|k+1] = \left(\mathbf{I} - \mathbf{K}_i \hat{\mathbf{J}}_i\right) \hat{\mathbf{\Phi}}_{ij}[k+1|k]. \tag{3.10}$$

When the $i^{\text{th}}$ and $j^{\text{th}}$ UVs need to reproduce their associated cross-covariance, they can simply use (3.4), resulting in

$$\begin{aligned} \hat{\mathbf{\Sigma}}_{ij}[k+1|k+1] &= \left(\mathbf{I} - \mathbf{K}_i \hat{\mathbf{J}}_i\right) \hat{\mathbf{\Phi}}_{ij}[k+1|k] \hat{\mathbf{\Phi}}_{ij}^T[k+1|k] \\ &= \left(\mathbf{I} - \mathbf{K}_i \hat{\mathbf{J}}_i\right) \hat{\mathbf{\Sigma}}_{ij}[k+1|k], \end{aligned}$$

which is equivalent to (3.9), if $\hat{\mathbf{\Sigma}}_{ij}[k+1|k] = \mathbf{\Sigma}_{ij}[k+1|k]$. Again, this is generally not the case, however, it shows that the centralized step can be reproduced in a decentralized way using no communication.

Summing up, the update equations for when a UV makes a private measurement are given by (3.7) and (3.10), using their estimated belief in place of $\mathbf{x}_i$ and $\mathbf{\Sigma}_{ii}$ for computation purposes.

#### 3.2.2.B Relative measurements

Here, measurements that involve two UVs are considered. Following the derivation presented by the authors of [18], consider that, at time $t_{k+1}$, the $i^{\text{th}}$ UV makes a measurement that includes the state of the $j^{\text{th}}$ UV,

$$\mathbf{y}_i[k+1] = \mathbf{h}_i(\mathbf{x}_i[k+1], \mathbf{x}_j[k+1]) \in \mathbb{R}^m.$$

Then, without loss of generality, let $\hat{\mathbf{x}}_a[k] \coloneqq \begin{bmatrix}\hat{\mathbf{x}}_i^T[k] & \hat{\mathbf{x}}_j^T[k]\end{bmatrix}^T$ and $\hat{\mathbf{x}}_b[k] \coloneqq \begin{bmatrix}\hat{\mathbf{x}}_k^T[k] & \cdots & \hat{\mathbf{x}}_N^T[k]\end{bmatrix}^T$, such that the complete state estimation vector is given by $\hat{\mathbf{x}}[k] \coloneqq \begin{bmatrix}\hat{\mathbf{x}}_a^T[k] & \hat{\mathbf{x}}_b^T[k]\end{bmatrix}^T$. The Kalman gain for the joint system is given by

$$\mathbf{K} = \hat{\mathbf{\Sigma}}[k+1|k]\hat{\mathbf{J}}^T\left(\hat{\mathbf{J}}\hat{\mathbf{\Sigma}}[k+1|k]\hat{\mathbf{J}} + \mathbf{R}_i\right)^{-1},$$

where

$$\begin{aligned}\hat{\mathbf{J}} &= \begin{bmatrix}\dfrac{\partial \mathbf{h}_i}{\partial \mathbf{x}_i}(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j) & \dfrac{\partial \mathbf{h}_i}{\partial \mathbf{x}_j}(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j) & \mathbf{0}_{m\times 6} & \cdots & \mathbf{0}_{m\times 6}\end{bmatrix} \\ &= \begin{bmatrix}\hat{\mathbf{J}}_a & \mathbf{0}_{m\times 6} & \cdots & \mathbf{0}_{m\times 6}\end{bmatrix},\end{aligned}$$

computed using the predicted state estimates, and $\mathbf{R}_i$ is the measurement noise covariance matrix. Letting

$$\hat{\mathbf{\Sigma}}[k] \coloneqq \begin{bmatrix}\hat{\mathbf{\Sigma}}_{aa}[k] & \hat{\mathbf{\Sigma}}_{ab}[k] \\ \hat{\mathbf{\Sigma}}_{ba}[k] & \hat{\mathbf{\Sigma}}_{bb}[k]\end{bmatrix},$$

the Kalman gain can be decomposed into

$$\mathbf{K} \coloneqq \begin{bmatrix}\mathbf{K}_a \\ \mathbf{K}_b\end{bmatrix},$$

where

$$\begin{cases}\mathbf{K}_a = \hat{\mathbf{\Sigma}}_{aa}[k+1|k]\hat{\mathbf{J}}_a^T\left(\mathbf{J}_a\hat{\mathbf{\Sigma}}_{aa}[k+1|k]\hat{\mathbf{J}}_a^T + \mathbf{R}_i\right)^{-1} \\ \mathbf{K}_b = \hat{\mathbf{\Sigma}}_{ba}[k+1|k]\hat{\mathbf{J}}_a^T\left(\mathbf{J}_a\hat{\mathbf{\Sigma}}_{aa}[k+1|k]\hat{\mathbf{J}}_a^T + \mathbf{R}_i\right)^{-1}\end{cases}.$$

The decomposed update equations for $\hat{\mathbf{x}}_a$ and $\hat{\mathbf{x}}_b$ are then

$$\begin{cases}\hat{\mathbf{x}}_a[k+1|k+1] = \hat{\mathbf{x}}_a[k+1|k] + \mathbf{K}_a(\mathbf{y}_i[k+1] - \hat{\mathbf{y}}_i[k+1]) & (3.11) \\ \hat{\mathbf{x}}_b[k+1|k+1] = \hat{\mathbf{x}}_b[k+1|k] + \mathbf{K}_b(\mathbf{y}_i[k+1] - \hat{\mathbf{y}}_i[k+1]) & (3.12)\end{cases},$$

and the covariances update as

$$\begin{cases} \hat{\mathbf{\Sigma}}_{aa}[k+1|k+1] = \left(\mathbf{I} - \mathbf{K}_a \hat{\mathbf{J}}_a\right) \hat{\mathbf{\Sigma}}_{aa}[k+1|k] & (3.13) \\ \hat{\mathbf{\Sigma}}_{ab}[k+1|k+1] = \left(\mathbf{I} - \mathbf{K}_a \hat{\mathbf{J}}_a\right) \hat{\mathbf{\Sigma}}_{ab}[k+1|k] & (3.14) \\ \hat{\mathbf{\Sigma}}_{bb}[k+1|k+1] = \hat{\mathbf{\Sigma}}_{bb}[k+1|k] - \mathbf{K}_b \hat{\mathbf{J}}_a \hat{\mathbf{\Sigma}}_{ab}[k+1|k] & (3.15) \end{cases} .$$

As before, the beliefs of UVs that do not participate in the measurement remain unchanged, since this would require communication between participating and non-participating agents, thus updates (3.12) and (3.15) are not performed. Letting $\mathbf{K}_a = \begin{bmatrix} \mathbf{K}_i \\ \mathbf{K}_j \end{bmatrix}$ and $\hat{\mathbf{J}}_a = \begin{bmatrix} \hat{\mathbf{J}}_i & \hat{\mathbf{J}}_j \end{bmatrix}$, further decomposing of equations (3.11) and (3.13) results in

$$\begin{cases} \hat{\mathbf{x}}_i[k+1|k+1] = \hat{\mathbf{x}}_i[k+1|k] + \mathbf{K}_i(\mathbf{y}_i[k+1] - \hat{\mathbf{y}}_i[k+1]) \\ \hat{\mathbf{x}}_j[k+1|k+1] = \hat{\mathbf{x}}_j[k+1|k] + \mathbf{K}_j(\mathbf{y}_i[k+1] - \hat{\mathbf{y}}_i[k+1]) \end{cases},$$

and

$$\begin{cases} \hat{\mathbf{\Sigma}}_{ii}[k+1|k+1] & = \left(\mathbf{I} - \mathbf{K}_i \hat{\mathbf{J}}_i\right) \hat{\mathbf{\Sigma}}_{ii}[k+1|k] - \mathbf{K}_i \hat{\mathbf{J}}_j \hat{\mathbf{\Sigma}}_{ji}[k+1|k] \\ \hat{\mathbf{\Sigma}}_{ij}[k+1|k+1] & = \left(\mathbf{I} - \mathbf{K}_j \hat{\mathbf{J}}_j\right) \hat{\mathbf{\Sigma}}_{ij}[k+1|k] - \mathbf{K}_i \hat{\mathbf{J}}_j \hat{\mathbf{\Sigma}}_{jj}[k+1|k] \\ \hat{\mathbf{\Sigma}}_{jj}[k+1|k+1] & = \left(\mathbf{I} - \mathbf{K}_j \hat{\mathbf{J}}_j\right) \hat{\mathbf{\Sigma}}_{jj}[k+1|k] - \mathbf{K}_j \hat{\mathbf{J}}_i \hat{\mathbf{\Sigma}}_{ij}[k+1|k] \end{cases} .$$

Thus, to make the above equations implementable, when the $i^{\text{th}}$ agent takes a measurement about the $j^{\text{th}}$ one, the latter communicates to the former its belief, $\mathcal{B}_j$, and its cross-covariance factor between the two, $\hat{\mathbf{\Phi}}_{ji}$. Then, the updates (3.11) and (3.13) can be performed locally, at the $i^{\text{th}}$ UV, by reconstructing the cross-covariance term, $\hat{\mathbf{\Sigma}}_{ij}$, according to (3.4). Afterwards, UV $i$ can send UV $j$ its updated belief, extracted from $\hat{\mathbf{x}}_{aa}[k+1|k+1]$ and $\hat{\mathbf{\Sigma}}_{aa}[k+1|k+1]$. The updated cross-covariance between the $i^{\text{th}}$ and $j^{\text{th}}$ agents can be tracked without communication by agreeing upon a factorization beforehand, as done in [18], by letting

$$\begin{cases} \hat{\mathbf{\Phi}}_{ij}[k+1|k+1] = \hat{\mathbf{\Sigma}}_{ij}[k+1|k+1] & (3.16) \\ \hat{\mathbf{\Phi}}_{ji}[k+1|k+1] = \mathbf{I}_6 & (3.17) \end{cases},$$

thus guaranteeing that $\hat{\mathbf{\Phi}}_{ij}[k+1|k+1]\hat{\mathbf{\Phi}}_{ji}^T[k+1|k+1] = \hat{\mathbf{\Sigma}}_{ij}[k+1|k+1]$.

The only part of the update that is missing is (3.14). Further decomposing of that equation results in

$$\begin{cases} \hat{\mathbf{\Sigma}}_{ik}[k+1|k+1] = (\mathbf{I} - \mathbf{K}_i \mathbf{J}_i)\, \hat{\mathbf{\Sigma}}_{ik}[k+1|k] - \mathbf{K}_i \hat{\mathbf{J}}_j \hat{\mathbf{\Sigma}}_{jk}[k+1|k] \\ \hat{\mathbf{\Sigma}}_{jk}[k+1|k+1] = (\mathbf{I} - \mathbf{K}_j \mathbf{J}_j)\, \hat{\mathbf{\Sigma}}_{jk}[k+1|k] - \mathbf{K}_j \hat{\mathbf{J}}_i \hat{\mathbf{\Sigma}}_{ik}[k+1|k] \end{cases},$$

which correspond to the update equations for the cross-covariance between participating agents and non-participating ones. Since computation of these equations would require communication between participating and non-participating UVs in order to reconstruct the associated cross-covariances, they

are instead approximated by

$$\begin{cases} \hat{\mathbf{\Sigma}}_{ik}[k+1|k+1] \approx \hat{\mathbf{\Sigma}}_{ii}[k+1|k+1]\hat{\mathbf{\Sigma}}_{ii}^{-1}[k+1|k]\hat{\mathbf{\Sigma}}_{ik}[k+1|k] \\ \hat{\mathbf{\Sigma}}_{jk}[k+1|k+1] \approx \hat{\mathbf{\Sigma}}_{jj}[k+1|k+1]\hat{\mathbf{\Sigma}}_{jj}^{-1}[k+1|k]\hat{\mathbf{\Sigma}}_{jk}[k+1|k] \end{cases},$$

which are implemented on each UV through their cross-covariance factors as

$$\begin{cases} \hat{\mathbf{\Phi}}_{ik}[k+1|k+1] = \hat{\mathbf{\Sigma}}_{ii}[k+1|k+1]\hat{\mathbf{\Sigma}}_{ii}^{-1}[k+1|k]\hat{\mathbf{\Phi}}_{ik}[k+1|k] & (3.18) \\ \hat{\mathbf{\Phi}}_{jk}[k+1|k+1] = \hat{\mathbf{\Sigma}}_{jj}[k+1|k+1]\hat{\mathbf{\Sigma}}_{jj}^{-1}[k+1|k]\hat{\mathbf{\Phi}}_{jk}[k+1|k] & (3.19) \end{cases}.$$

This approximation is presented as the main contribution of the authors of this approach and the interested reader is referred to [18] for details, though it relies on the fact that it is a good approximation if there is a high correlation between participating and non-participating agents.

In summary, when the $i^{\text{th}}$ UV makes a measurement about a neighbor UV, the latter communicates $\hat{\mathbf{x}}_j$, $\hat{\mathbf{\Sigma}}_{jj}$ and $\hat{\mathbf{\Phi}}_{ji}$ to the former. The $i^{\text{th}}$ UV then performs the updates given by (3.11), (3.13), (3.16) and (3.18) and transmits the updated $\hat{\mathbf{x}}_j$ and $\hat{\mathbf{\Sigma}}_{jj}$ back to its neighbor, who then performs the updates (3.17) and (3.19). Note that (3.18) and (3.19) must be performed for every UV that is known by the updating agent $i$ or $j$.

# 4

# Observers Based on Artificial Measurements

Here, navigation algorithms based on state observers that feature artificial quantities are presented. These quantities can be observation matrices, measurement vectors or both, and are designed such that the resulting system is linear, either time-varying or not. This makes it so the complete system has the potential to present globally convergent error dynamics, given an appropriate measurement topology and spatial formation. Unlike the EKF-based approaches presented previously, if the formation configuration is appropriate, these algorithms might not require that the initial navigation estimates be close to their real values, which eliminates the need for an initial setup of each UV, thus increasing the time efficiency of missions. The approaches presented are largely based on the projection matrix discussed in Section 4.1, which was presented previously in [19].

## 4.1 The bearing projection matrix

An essential feature of each of these estimators is the matrix constructed from the direction vector that points from agent $i$ to agent $j$ [19],

$$\mathbf{D}_{ij}(t_k) \coloneqq \mathbf{d}_{ij}(t_k)\mathbf{d}_{ij}^T(t_k), \tag{4.1}$$

with

$$\mathbf{d}_{ij}(t_k) = \frac{\mathbf{p}_j(t_k) - \mathbf{p}_i(t_k)}{||\mathbf{p}_j(t_k) - \mathbf{p}_i(t_k)||} \in \mathbb{R}^3.$$

$\mathbf{D}_{ij}$ is a projection matrix which, when applied to a vector $\mathbf{x}$, gives its projection along the line passing through the origin with direction $\mathbf{d}_{ij}$. In order to make this fact clear, let $\mathbf{x} \in \mathbb{R}^3$ be any non-zero vector and $\mathbf{D} \in \mathbb{R}^{3\times3}$ a matrix constructed from a direction vector, $\mathbf{d}$, as in (4.1). Then, letting $\mathbf{x}_d = \frac{\mathbf{x}}{||\mathbf{x}||}$,

$$\begin{aligned}\mathbf{D}\mathbf{x} &= \mathbf{d}\mathbf{d}^T\mathbf{x} \\ &= \mathbf{d}\mathbf{d}^T\mathbf{x}_d||\mathbf{x}|| \\ &= \mathbf{d}||\mathbf{x}||\cos(\alpha),\end{aligned}$$

that is, $\mathbf{D}\mathbf{x}$ is a vector with direction $\mathbf{d}$, and magnitude $||\mathbf{D}\mathbf{x}|| = ||\mathbf{x}||\cos(\alpha)$, where $\alpha = \arccos(\mathbf{d}^T\mathbf{x}_d) \in \mathbb{R}$ is the angle between the vectors $\mathbf{d}$ and $\mathbf{x}$.

Letting $\mathbf{x}$ be a position difference between two agents, $\mathbf{x} = \mathbf{p}_i - \mathbf{p}_j$ such that $\mathbf{p}_i \neq \mathbf{p}_j$, and letting $\mathbf{d}_{ij}$ be the direction vector from agent $i$ to agent $j$, then

$$\mathbf{D}_{ij}(\mathbf{p}_i - \mathbf{p}_j) = \mathbf{p}_i - \mathbf{p}_j,$$

or, equivalently,

$$\bar{\mathbf{D}}_{ij}(\mathbf{p}_i - \mathbf{p}_j) = \mathbf{0}_{3\times1},$$

where

$$\bar{\mathbf{D}}_{ij}(t_k) \coloneqq \mathbf{I}_3 - \mathbf{d}_{ij}(t_k)\mathbf{d}_{ij}^T(t_k) \tag{4.2}$$

is the orthogonal complement of the projection matrix $\mathbf{D}_{ij}$. Considering now that one has access to position estimates at time $t = t_k$, that is, $\hat{\mathbf{p}}_i(t_k)$ and $\hat{\mathbf{p}}_j(t_k)$, the projection of these estimates by the matrix $\mathbf{D}_{ij}(t_k)$, dropping the explicit time-dependence for readability, is given by

$$\begin{aligned}\mathbf{D}_{ij}(\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j) &= \mathbf{D}_{ij}(\mathbf{p}_i - \mathbf{p}_j) + \mathbf{D}_{ij}(\mathbf{e}_i - \mathbf{e}_j) \\ &= \mathbf{p}_i - \mathbf{p}_j + \mathbf{D}_{ij}(\mathbf{e}_i - \mathbf{e}_j),\end{aligned}$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$ are the unknown error terms associated with the position estimates. The projected

position difference estimate error is then given by

$$\mathbf{D}_{ij}(\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j) - (\mathbf{p}_i - \mathbf{p}_j) = \mathbf{D}_{ij}(\mathbf{e}_i - \mathbf{e}_j)$$

and, since

$$\mathbf{D}_{ij}(\mathbf{e}_i - \mathbf{e}_j) = \mathbf{d}_{ij}||\mathbf{e}_i - \mathbf{e}_j|| \cos(\alpha),$$

the projected position difference estimate vector is never a worse approximation of $\mathbf{p}_i - \mathbf{p}_j$ than the original vector (given noiseless direction measurements), such that

$$||\mathbf{p}_i - \mathbf{p}_j - \mathbf{D}_{ij}(\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j)|| \leq ||\mathbf{p}_i - \mathbf{p}_j - (\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j)||.$$

### 4.1.1 Direction vector error bias

Consider that the agent with index $i$ takes a bearing measurement about the $j^{\text{th}}$ UV in agent $i$'s body frame. Then, the direction vector, $\mathbf{d}_{ij}(t_k)$, can be built using

$$\mathbf{d}_{ij}(t_k) = \mathbf{R}_i(t_k) \begin{bmatrix} \cos\theta_{ij}(t_k)\cos\phi_{ij}(t_k) \\ \cos\theta_{ij}(t_k)\sin\phi_{ij}(t_k) \\ \sin\theta_{ij}(t_k) \end{bmatrix}. \tag{4.3}$$

An issue with building this direction vector, however, is that the unbiased noise characteristics of the original bearing angles, $\theta_{ij}$ and $\phi_{ij}$, are lost. Consider the normally distributed random variable $X \sim \mathcal{N}(\mu, \sigma^2)$ and the transformation $Y = e^{jX}$. Considering $Z \sim \mathcal{N}(0, 1)$, one has that $X = \mu + \sigma Z$. Taking the expected value of $Y$, one has

$$E[Y] = E\left[e^{jX}\right] = E\left[e^{j\mu + j\sigma Z}\right] = E\left[e^{j\mu}e^{j\sigma Z}\right] = e^{j\mu}E\left[e^{j\sigma Z}\right].$$

Letting $\alpha = j\sigma$ and $f(z)$ be the probability density function of the normally distributed variable, $Z$, one

has

$$\begin{aligned} E\left[e^{\alpha Z}\right] &= \int_{-\infty}^{+\infty} e^{\alpha z} f(z) dz = \\ &= \int_{-\infty}^{+\infty} e^{\alpha z} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz = \\ &= \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{\frac{1}{2}(-z^2+2\alpha)} dz = \\ &= \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(z^2-2\alpha+\alpha^2)} e^{\alpha^2/2} dz = \\ &= e^{\alpha^2/2} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-(z-\alpha)^2/2} dz = \\ &= e^{\alpha^2/2}, \end{aligned}$$

where the following equality was used, considering using $u = z - \alpha$,

$$\int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-u^2/2} du = 1.$$

Finally, it can be concluded that

$$E\left[Y\right] = e^{j\mu} e^{(j\sigma)^2/2} = e^{j\mu} e^{-\sigma^2/2} = e^{-\sigma^2/2} \left(\cos\mu + j\sin\mu\right).$$

Since $e^{jX} = \cos X + j \sin X$, it holds that

$$E\left[\cos X + j\sin X\right] = e^{-\sigma^2/2} \left(\cos\mu + j\sin\mu\right),$$

or, equivalently,

$$E\left[\cos X\right] + jE\left[\sin X\right] = e^{-\sigma^2/2}\cos\mu + je^{-\sigma^2/2}\sin\mu,$$

from which it is possible to conclude that

$$\begin{cases} E\left[\cos X\right] = e^{-\sigma^2/2}\cos\mu \\ E\left[\sin X\right] = e^{-\sigma^2/2}\sin\mu \end{cases}. \tag{4.4}$$

Consider the two independent and normally distributed random variables $\Theta \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2)$ and $\Phi \sim \mathcal{N}(\mu_\phi, \sigma_\phi^2)$, modeling the bearing measurements at each time-step, and consider also the transformation

$$\mathbf{D} = \begin{bmatrix} \cos\Theta \sin\Phi \\ \cos\Theta\cos\Phi \\ \sin\Theta \end{bmatrix}.$$

Since $\Theta$ and $\Phi$ are independent, then, according to (4.4), one has

$$E\left[\mathbf{D}\right] = \begin{bmatrix} e^{-\frac{1}{2}(\sigma_\theta^2+\sigma_\phi^2)}\cos\mu_\theta\cos\mu_\phi \\ e^{-\frac{1}{2}(\sigma_\theta^2+\sigma_\phi^2)}\cos\mu_\theta\sin\mu_\phi \\ e^{-\sigma_\theta^2/2}\sin\mu_\theta \end{bmatrix},$$

which, when compared with the nominal direction vector, $\mathbf{d}$, gives

$$\begin{aligned}\mathbf{d} - E\left[\mathbf{D}\right] &= \begin{bmatrix} \cos\mu_\theta\cos\mu_\phi \\ \cos\mu_\theta\sin\mu_\phi \\ \sin\mu_\theta \end{bmatrix} - \begin{bmatrix} e^{-\frac{1}{2}(\sigma_\theta^2+\sigma_\phi^2)}\cos\mu_\theta\cos\mu_\phi \\ e^{-\frac{1}{2}(\sigma_\theta^2+\sigma_\phi^2)}\cos\mu_\theta\sin\mu_\phi \\ e^{-\sigma_\theta^2/2}\sin\mu_\theta \end{bmatrix} \\ &= \begin{bmatrix} \left(1-e^{-\frac{1}{2}(\sigma_\theta^2+\sigma_\phi^2)}\right)\cos\mu_\theta\cos\mu_\phi \\ \left(1-e^{-\frac{1}{2}(\sigma_\theta^2+\sigma_\phi^2)}\right)\cos\mu_\theta\sin\mu_\phi \\ \left(1-e^{-\sigma_\theta^2/2}\right)\sin\mu_\theta \end{bmatrix} \neq \mathbf{0}_{3\times 1}.\end{aligned}$$

Thus, on average, the constructed direction vectors will not correspond to the nominal direction vector. If this artificial direction vector is used, the Kalman filter state estimate error will generally not be unbiased.

## 4.2 Independently connected Kalman filters

In this section, the decentralized approach presented in [19] is described, whereby an artificial output, using depth information and the bearing projection matrix, is created and used to ensure that each UV sub-system is modeled as a globally observable linear time-variant system. Each agent is then responsible for carrying a Kalman filter based on this system model, and the estimates produced by each filter are fed to the other observers as ”true” information, interconnecting them in such a way that the complete system presents globally convergent observer error dynamics. This can be considered the least cooperative of the considered observers, since each agent has its own estimator which does not take into account cross-measurement activity, unlike the other approaches which will be studied, which use the measurement graph information to couple the estimates of each filter. However, besides the approach presented in Section 4.5, it is the one which requires the least amount of communication.

Let $i$ be a follower UV with state $\mathbf{x}_i$ and neighbor set $\mathcal{N}_i = \{1, \ldots, |\mathcal{N}_i|\}$, where, for ease of representation, it is assumed that the neighbors of the $i^{\text{th}}$ UV have indices $1$ to $|\mathcal{N}_i|$. From

$$\bar{\mathbf{D}}_{ij}(t_k)(\mathbf{p}_i(t_k) - \mathbf{p}_j(t_k)) = \mathbf{0}_{3\times 1},$$

with $\bar{\mathbf{D}}_{ij}(t_k)$ defined as in (4.2), an artificial output relating the position of the $i^{\text{th}}$ UV with the position of

an agent $j \in \mathcal{N}_i$ can be written as

$$\bar{\mathbf{D}}_{ij}(t_k)\mathbf{p}_i(t_k) = \bar{\mathbf{D}}_{ij}(t_k)\mathbf{p}_j(t_k) =: \mathbf{y}_{ij}[k].$$

If agent $i$ has more than one neighbor, the artificial outputs can be concatenated, and the complete measurement vector of UV $i$ is given by

$$\mathbf{y}_i[k] \coloneqq \begin{bmatrix} \mathbf{y}_{i1}[k] \\ \vdots \\ \mathbf{y}_{i|\mathcal{N}_i|}[k] \\ z_i(t_k) \end{bmatrix} = \mathbf{C}_i(t_k)\mathbf{x}[k],$$

where $z_i$ is the depth measurement obtained by the UV and

$$\mathbf{C}_i(t_k) = \begin{bmatrix} \bar{\mathbf{D}}_{i1}(t_k) & \mathbf{0}_3 \\ \vdots & \vdots \\ \bar{\mathbf{D}}_{i|\mathcal{N}_i|}(t_k) & \mathbf{0}_3 \\ \mathbf{e}_z & \mathbf{0}_{1\times 3} \end{bmatrix},$$

with $\mathbf{e}_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$.

A Kalman filter can then be independently defined for each agent using the artificial output, $\mathbf{y}_i$. The prediction equations are given by

$$\begin{cases} \hat{\mathbf{x}}_i[k+1|k] = \mathbf{A}\hat{\mathbf{x}}[k|k] + \mathbf{B}\mathbf{u}[k] \\ \mathbf{\Sigma}_{ii}[k+1|k] = \mathbf{A}\mathbf{\Sigma}_{ii}[k|k]\mathbf{A}^T + \mathbf{Q}_i \end{cases},$$

where $\mathbf{Q}_i$ is the agent's process noise covariance matrix, and the update equations for this filter are given by

$$\begin{cases} \hat{\mathbf{x}}_i[k+1|k+1] = \hat{\mathbf{x}}_i[k+1|k] + \mathbf{K}_i(\mathbf{y}_i[k+1] - \mathbf{C}_i(t_{k+1})\hat{\mathbf{x}}_i[k+1|k]) \\ \mathbf{\Sigma}_{ii}[k+1|k+1] = (\mathbf{I}_6 - \mathbf{K}_i\mathbf{C}_i(t_{k+1}))\,\mathbf{\Sigma}_{ii}[k+1|k] \end{cases},$$

where $\mathbf{K}_i$ is the Kalman gain, computed according to

$$\mathbf{K}_i = \mathbf{\Sigma}_i[k+1|k]\mathbf{C}_i^T(t_{k+1})\left(\mathbf{C}_i(t_{k+1})\mathbf{\Sigma}_i[k+1|k]\mathbf{C}_i^T(t_{k+1}) + \mathbf{R}_i\right)^{-1},$$

and $\mathbf{R}_i$ is the measurement noise covariance matrix. Since the actual position vector of UV $j$, $\mathbf{p}_j(t_{k+1})$, is unknown, the current position estimate of the $j^{\text{th}}$ agent, $\hat{\mathbf{p}}_j(t_{k+1}|t_k)$, is extracted from the predicted

estimate of UV $j$, $\hat{\mathbf{x}}_j[k+1|k]$, and used instead to build the measurement vector, $\mathbf{y}_i[k+1]$, such that

$$\mathbf{y}_i[k+1] = \begin{bmatrix} \bar{\mathbf{D}}_{i1}(t_{k+1})\hat{\mathbf{p}}_1(t_{k+1}|t_k) \\ \vdots \\ \bar{\mathbf{D}}_{i|\mathcal{N}_i|}(t_{k+1})\hat{\mathbf{p}}_{|\mathcal{N}_i|}(t_{k+1}|t_k) \\ z_i(t_{k+1}) \end{bmatrix}.$$

By doing this, an error is introduced into the estimate of $\mathbf{p}_i$, however, provided that the error in $\hat{\mathbf{p}}_j$ is decaying, the error in $\hat{\mathbf{p}}_i$ will decay as well, which is easily shown to be the case for acyclical measurement topologies [19, 20].

In [19], the approach is considered for tiered formations, in which there are no loops in the measurement graph, and thus no reintroduction of estimation errors into the update equations. For cyclical measurement topologies, however, it is not as straightforward to show that the interconnected observers will have globally stable error dynamics. The convergence of this algorithm for under both types of measurement topologies is studied via Monte Carlo analysis in Section 5.2.

## 4.3 Centralized Kalman filter

The centralized version of the observers based on bearing and depth measurements is presented in this section. Again, while a centralized approach might not be feasible in practice, it still serves as a baseline for comparison with the decentralized approaches based on bearing-based artificial outputs.

Let the state of the centralized system be defined as

$$\mathbf{x}[k] \coloneqq \begin{bmatrix} \mathbf{x}_1[k] \\ \vdots \\ \mathbf{x}_N[k] \end{bmatrix} \in \mathbb{R}^{6N},$$

and let $\hat{\mathbf{x}}$ and $\mathbf{\Sigma}$ be its state estimate and covariance matrix, respectively. The motion model of this approach is the same as that of the CEKF, i.e., upon receiving the control signals, the agents' estimates are predicted according to

$$\begin{cases} \hat{\mathbf{x}}[k+1|k] = \mathbf{A}_c\hat{\mathbf{x}}[k|k] + \mathbf{B}_c\mathbf{u}[k] \\ \mathbf{\Sigma}[k+1|k] = \mathbf{A}_c\mathbf{\Sigma}[k|k]\mathbf{A}_c^T + \mathbf{Q}_c \end{cases},$$

where $\mathbf{A}_c$ and $\mathbf{B}_c$ are defined as in (3.1) and $\mathbf{Q}_c$ is the centralized process noise covariance matrix.

Regarding the update step, for the case of leader agents, $\mathbf{y}_i$ is a position measurement and $\mathbf{C}_i = \begin{bmatrix} \cdots & \mathbf{I}_3 & \mathbf{0}_3 & \cdots \end{bmatrix}$. As for follower UVs, let $\mathbf{C}_c$ be the centralized observation matrix, such that $\mathbf{y} = \mathbf{C}_c\mathbf{x}$,

and let $\mathbf{y}$ and $\mathbf{C}_c$ be defined as

$$\mathbf{y}[k] \coloneqq \begin{bmatrix} \mathbf{y}_1[k] \\ \vdots \\ \mathbf{y}_N[k] \end{bmatrix}, \qquad \mathbf{C}_c(t_k) \coloneqq \begin{bmatrix} \mathbf{C}_1(t_k) \\ \vdots \\ \mathbf{C}_N(t_k) \end{bmatrix},$$

where $\mathbf{y}_i$ is the measurement vector captured by the follower agent $i$. Considering the relationship presented in [19],

$$\bar{\mathbf{D}}_{ij}(t_k)(\mathbf{p}_i(t_k) - \mathbf{p}_j(t_k)) = \mathbf{0}_{3\times 1},$$

and that the updating agent has access to depth measurements, then

$$\mathbf{y}_i[k] = \begin{bmatrix} \mathbf{0}_{3|\mathcal{N}_i|\times 1} \\ z_i(t_k) \end{bmatrix},$$

where $z_i$ is the depth measurement. Each $\mathbf{C}_i$ relates the measurements captured by the $i^{\text{th}}$ agent with the total state vector, using the orthogonal complement of the bearing projection matrix and depth information, as shown below, in Example 2. The total state estimate is then corrected according to the standard Kalman filter update equations

$$\begin{cases} \hat{\mathbf{x}}[k+1|k+1] = \hat{\mathbf{x}}[k+1|k] + \mathbf{K}(\mathbf{y}[k+1] - \mathbf{C}_c(t_{k+1})\hat{\mathbf{x}}[k+1|k]) \\ \mathbf{\Sigma}[k+1|k+1] = (\mathbf{I} - \mathbf{K}\mathbf{C}_c(t_{k+1}))\,\mathbf{\Sigma}[k+1|k] \end{cases},$$

where $\mathbf{K} = \mathbf{\Sigma}[k+1|k]\mathbf{C}_c^T(t_{k+1})\left(\mathbf{C}_c(t_{k+1})\mathbf{\Sigma}[k+1|k]\mathbf{C}_c^T(t_{k+1}) + \mathbf{R}_c\right)^{-1}$ is the Kalman gain, with $\mathbf{R}_c$ a suitable measurement noise covariance matrix.

**Example 2.** Consider the formation presented in Fig. 2.1, such that $\mathcal{V} = \{1,2,3\}$, $\mathcal{V}_L = \{1\}$, and $\mathcal{V}_F = \{2,3\}$. The edge set is given by $\mathcal{E} = \{(1,2),(1,3),(2,3),(3,2)\}$. The measurement vector, in this example, is given by

$$\mathbf{y}[k] = \begin{bmatrix} \mathbf{y}_1[k] \\ \mathbf{0}_{6\times 1} \\ z_2(t_k) \\ \mathbf{0}_{6\times 1} \\ z_3(t_k) \end{bmatrix},$$

where $\mathbf{y}_1$ is the position measurement captured by agent $1$. Omitting the explicit time-dependence, one

has

$$\begin{cases} \mathbf{C}_1 &= \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \\ \mathbf{C}_2(t_k) &= \begin{bmatrix} -\bar{\mathbf{D}}_{12}(t_k) & \mathbf{0}_3 & \bar{\mathbf{D}}_{12}(t_k) & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \bar{\mathbf{D}}_{32}(t_k) & \mathbf{0}_3 & -\bar{\mathbf{D}}_{32}(t_k) & \mathbf{0}_3 \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{e}_z^T & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \end{bmatrix} \\ \mathbf{C}_3 &= \begin{bmatrix} -\bar{\mathbf{D}}_{13}(t_k) & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \bar{\mathbf{D}}_{13}(t_k) & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & -\bar{\mathbf{D}}_{23}(t_k) & \mathbf{0}_3 & \bar{\mathbf{D}}_{23}(t_k) & \mathbf{0}_3 \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{e}_z^T & \mathbf{0}_{1\times 3} \end{bmatrix} \end{cases},$$

from which the centralized observation matrix becomes

$$\mathbf{C}_c(t_k) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ -\bar{\mathbf{D}}_{21}(t_k) & \mathbf{0}_3 & \bar{\mathbf{D}}_{21}(t_k) & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \bar{\mathbf{D}}_{23}(t_k) & \mathbf{0}_3 & -\bar{\mathbf{D}}_{23}(t_k) & \mathbf{0}_3 \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{e}_z^T & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \\ -\bar{\mathbf{D}}_{31}(t_k) & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \bar{\mathbf{D}}_{31}(t_k) & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & -\bar{\mathbf{D}}_{32}(t_k) & \mathbf{0}_3 & \bar{\mathbf{D}}_{32}(t_k) & \mathbf{0}_3 \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{e}_z^T & \mathbf{0}_{1\times 3} \end{bmatrix}.$$

## 4.4 Distributed Kalman filter with covariance factorization

The algorithm presented in this section is based on a slight extension of the distributed filter presented in [18], allowing for asynchronous updates involving more than two agents at a time. Similarly to the DEKF, each agent carries its own estimated belief, $\mathcal{B}_i = \{\hat{\mathbf{x}}_i, \hat{\mathbf{\Sigma}}_{ii}\}$, and cross-covariance factors to other UVs, $\hat{\mathbf{\Phi}}_{ij}$. The prediction equations for these quantities are performed as in the DEKF. In fact, the only difference between these two approaches is the update step, which is not restricted to pairwise communication and, instead, uses the orthogonal complement of the projection matrix, presented in section 4.1, to construct an observer with linear dynamics.

Again, let agent $i$ take bearing measurements about its neighbors, which will be assumed, without loss of generality, to have indices $j \in \mathcal{N}_i = \{1, \ldots, |\mathcal{N}_i|\}$. Additionally, the $i^{\text{th}}$ UV might have access to depth measurements. Using the relationship presented in [19] once more, $\bar{\mathbf{D}}_{ij}(t_k)(\mathbf{p}_i(t_k) - \mathbf{p}_j(t_k)) = \mathbf{0}_3$, then, considering the joint state vector

$$\mathbf{x}_a[k] \coloneqq \begin{bmatrix} \mathbf{x}_i[k] \\ \mathbf{x}_1[k] \\ \vdots \\ \mathbf{x}_{|\mathcal{N}_i|}[k] \end{bmatrix},$$

where $\mathbf{x}_i[k]$ is the current state of the measuring agent at time $k$, one has

$$\mathbf{C}_a(t_k)\mathbf{x}_a[k] = \begin{bmatrix} \mathbf{0}_{3|\mathcal{N}_i|\times 1} \\ z_i(t_k) \end{bmatrix} =: \mathbf{y}_a[k],$$

where

$$\mathbf{C}_a(t_k) = \begin{bmatrix} \bar{\mathbf{D}}_{i1}(t_k) & \mathbf{0}_3 & -\bar{\mathbf{D}}_{i1}(t_k) & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \cdots & \mathbf{0}_3 & \mathbf{0}_3 \\ \bar{\mathbf{D}}_{i2}(t_k) & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\bar{\mathbf{D}}_{i2}(t_k) & \mathbf{0}_3 & \cdots & \mathbf{0}_3 & \mathbf{0}_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \bar{\mathbf{D}}_{i|\mathcal{N}_i|}(t_k) & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \cdots & -\bar{\mathbf{D}}_{i|\mathcal{N}_i|}(t_k) & \mathbf{0}_3 \\ \mathbf{e}_z^T & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \cdots & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \end{bmatrix},$$

and $\mathbf{e}_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$.

Let the joint system state estimate be denoted as $\hat{\mathbf{x}}_a$, and its associated covariance matrix estimate as

$$\hat{\mathbf{\Sigma}}_{aa}[k] = \begin{bmatrix} \hat{\mathbf{\Sigma}}_{ii}[k] & \hat{\mathbf{\Sigma}}_{i1}[k] & \cdots & \hat{\mathbf{\Sigma}}_{i|\mathcal{N}_i|}[k] \\ \hat{\mathbf{\Sigma}}_{1i}[k] & \hat{\mathbf{\Sigma}}_{11}[k] & \cdots & \hat{\mathbf{\Sigma}}_{1|\mathcal{N}_i|}[k] \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathbf{\Sigma}}_{|\mathcal{N}_i|i}[k] & \hat{\mathbf{\Sigma}}_{|\mathcal{N}_i|1}[k] & \cdots & \hat{\mathbf{\Sigma}}_{|\mathcal{N}_i||\mathcal{N}_i|}[k] \end{bmatrix}. \tag{4.5}$$

In order to reduce the required amount of communication, the cross-covariance terms between UVs that agent $i$ makes measurements about can be ignored, such that

$$\hat{\mathbf{\Sigma}}_{aa}[k] \approx \begin{bmatrix} \hat{\mathbf{\Sigma}}_{ii}[k] & \hat{\mathbf{\Sigma}}_{i1}[k] & \cdots & \hat{\mathbf{\Sigma}}_{i|\mathcal{N}_i|}[k] \\ \hat{\mathbf{\Sigma}}_{1i}[k] & \hat{\mathbf{\Sigma}}_{11}[k] & \cdots & \mathbf{0}_6 \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathbf{\Sigma}}_{|\mathcal{N}_i|i}[k] & \mathbf{0}_6 & \cdots & \hat{\mathbf{\Sigma}}_{|\mathcal{N}_i||\mathcal{N}_i|}[k] \end{bmatrix}. \tag{4.6}$$

If the communication restrictions are not as strict, the cross-covariance terms $\hat{\mathbf{\Sigma}}_{jk}$, for $j, k \in \mathcal{N}_i$, can be obtained from the cross-covariance factors that the participating agents $j$ and $k$ carry. By letting them communicate these quantities to agent $i$, it can then reconstruct the cross-covariance terms and place them into $\hat{\mathbf{\Sigma}}_{aa}$. The update equations are then performed as in the DEKF approach. The Kalman gain is computed for the joint system using the reconstructed covariance matrix, $\hat{\mathbf{\Sigma}}_{aa}$, and the new beliefs are computed and communicated to the participating agents, which then also perform updates to their cross-covariance factors according to the approximation presented in [18], i.e., each agent that participates in the update performs

$$\hat{\mathbf{\Phi}}_{ik}[k+1|k+1] = \hat{\mathbf{\Sigma}}_{ii}[k+1|k+1]\hat{\mathbf{\Sigma}}_{ii}^{-1}[k+1|k]\hat{\mathbf{\Phi}}_{ik}[k+1|k]$$

for every non-participating agent $k$ it has knowledge of. In case the full covariance matrix was used, the new cross-covariance terms between the participating agents can be factorized and distributed in a way

that does not double count information. A possible rule for distributing the cross-covariance terms could be, for example

$$\hat{\mathbf{\Phi}}_{ij}[k+1] = \begin{cases} \hat{\mathbf{\Sigma}}_{ij}[k+1] & \text{if } i < j \\ \mathbf{I}_6 & \text{if } i > j \end{cases},$$

though it is not necessarily the one which minimizes the amount of communication.

Two versions of this algorithm were implemented. The first relies on minimal communication which scales with the number of neighbors at each agent, ignoring all cross-covariance entries between the neighbors of the measuring agent, as in (4.6), and is labeled as the decentralized Kalman filter with partial covariance sharing (DKF-PCS); and the second one reproduces the full joint covariance matrix, thus requiring an amount of communication which scales with $|\mathcal{N}_i|^2$ at each agent. This latter approach is labeled as the decentralized Kalman filter with full covariance sharing (DKF-FCS).

## 4.5 Static-gain observer

In this section, a technique for computing steady-state observer gains for agents that can acquire relative position measurements to their neighbors, presented in [26], is briefly described. Local observers for each follower agent are then designed, coupling these gains with an artificial relative position outputs built from bearing measurements and depth differences between agents. Each observer has a prediction step and an update step, as with regular Kalman filters, though a covariance matrix is not maintained. Unlike the algorithms presented so far, this is the only one which requires the agents to have access to depth measurements. This is due to a geometrical limitation when it comes to obtaining position differences between two agents. This quantity can be extracted using a combination of range and depth measurements, or using a combination of bearing and depth measurements of different value, however, it is not possible to extract a position difference between two agents considering only bearing measurements.

All agents predict their estimate according to

$$\hat{\mathbf{x}}_i[k+1|k] = \mathbf{A}\hat{\mathbf{x}}_i[k|k] + \mathbf{B}\mathbf{u}_i[k],$$

where $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{u}_i[k]$ are defined as in (2.4), (2.5), and (2.2), respectively. Agent $i$ updates its estimate upon receiving measurements, according to

$$\hat{\mathbf{x}}_i[k+1|k+1] = \begin{cases} \hat{\mathbf{x}}_i[k+1|k] + \mathbf{K}_i(\mathbf{y}_i[k+1] - \hat{\mathbf{x}}_i[k+1|k]), & \text{if } i \in \mathcal{V}_L \\ \hat{\mathbf{x}}_i[k+1|k] + \mathbf{K}_i(\mathbf{m}_i[k+1] - \Delta\hat{\mathbf{x}}_i[k+1|k]), & \text{if } i \in \mathcal{V}_F \end{cases},$$

where

$$\Delta\hat{\mathbf{x}}_i[k] \coloneqq \begin{bmatrix} \hat{\mathbf{x}}_i[k] - \hat{\mathbf{x}}_1[k] \\ \vdots \\ \hat{\mathbf{x}}_i[k] - \hat{\mathbf{x}}_{|\mathcal{N}_i|}[k] \end{bmatrix},$$

$\mathbf{y}_i$ is an absolute position measurement, and $\mathbf{m}_i$ is a vector containing the captured depth measurement and relative position measurements between the measuring agent and its neighbors, assumed to be $\mathcal{N}_i = \{1, \ldots, |\mathcal{N}_i|\}$. The formation gains, $\mathbf{K}_i$, are computed by propagating the centralized system's covariance prediction and update equations using a gain matrix computed subject to a certain sparsity constraint, which, in this case, constrains the total system gain matrix, $\mathbf{K}$, to be block diagonal. Upon computing the formation gains, each block of $\mathbf{K}$ is extracted and set as $\mathbf{K}_i$ accordingly.

The centralized system's motion model is identical to that of the CEKF, presented in Section 3.1.1, such that $\mathbf{A}_c = \mathbf{I}_N \otimes \mathbf{A}$. As for the observation model, whereas leader agents can capture measurements of their own position, follower agents can only capture relative position and depth measurements, that is

$$\mathbf{m}_i[k] = \begin{bmatrix} \mathbf{p}_i(t_k) - \mathbf{p}_1(t_k) \\ \vdots \\ \mathbf{p}_i(t_k) - \mathbf{p}_{|\mathcal{N}_i|}(t_k) \\ z_i(t_k) \end{bmatrix}.$$

Let $\mathbf{C}_c$ be centralized system's observation matrix, containing matrices $\mathbf{C}_L = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \end{bmatrix}$ for leader measurement entries. For follower measurement entries, the measuring agent's entry is modeled with $\mathbf{C}_L$, whereas the entry corresponding to the agent whose measurement is taken about is modeled with $-\mathbf{C}_L$. Additionally, the depth measurements taken by follower agents are modeled using the vector $\mathbf{e}_z = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$. This construction is exemplified in Example 3.

Following the results derived in [26], the centralized gain subject to a sparsity constraint is then computed by propagating

$$\mathbf{\Sigma}[k+1|k] = \mathbf{A}_c \mathbf{\Sigma}[k|k] \mathbf{A}_c^T + \mathbf{Q}_c, \tag{4.7}$$

and

$$\mathbf{\Sigma}[k+1|k+1] = (\mathbf{I}_{6N} - \mathbf{K}[k]\mathbf{C}_c)\mathbf{\Sigma}[k+1|k](\mathbf{I}_{6N} - \mathbf{K}[k]\mathbf{C}_c)^T + \mathbf{K}[k]\mathbf{R}_c\mathbf{K}[k]^T \tag{4.8}$$

until the trace of $\mathbf{\Sigma}[k+1|k+1]$ reaches a steady-state value. Define $\mathbf{l}_i \in \mathbb{R}^{6N}$ as the unit vector such that all entries are zero except the $i^{\text{th}}$ one and let $\mathbf{L}_i \coloneqq \operatorname{diag}(\mathbf{l}_i)$. In the above equations, $\mathbf{Q}_c = \operatorname{diag}(\mathbf{Q}_1, \ldots, \mathbf{Q}_N)$ is the centralized process noise covariance matrix, $\mathbf{R}_c$ is the centralized observation model noise covariance matrix, and $\mathbf{K}[k]$ is given by

$$\mathbf{K}[k] = \sum_{i=1}^{6N} \mathbf{L}_i \mathbf{\Sigma}[k+1|k] \mathbf{C}_c^T \mathbf{M}_i \left(\mathbf{I}_{6N} - \mathbf{M}_i + \mathbf{M}_i \mathbf{S} \mathbf{M}_i\right)^{-1},$$

where

$$\mathbf{S} = \mathbf{C}_c \boldsymbol{\Sigma}[k+1|k]\mathbf{C}_c^T + \mathbf{R}_c.$$

The sparsity constraint is imposed by the matrix $\mathbf{M}_i$, which is built to encode the measurements each agent has access to. Letting $\mathbf{s}_i$ be a vector such that

$$\begin{cases} \mathbf{s}_i^j = 1, & \text{if } \mathbf{E}^{ij} = 1, \\ \mathbf{s}_i^j = 0, & \text{otherwise}, \end{cases}$$

where $\mathbf{E} \in \mathbb{R}^{6N \times m}$ is the sparsity pattern matrix, with $m$ the total measurement vector length, $\mathbf{M}_i$ is then built as $\mathbf{M}_i = \mathrm{diag}(\mathbf{s}_i)$. For a derivation of these equations the reader is referred to [26].

**Example 3.** Consider again the formation presented in Fig. 2.1. Following the measurement ordering adopted in Example 2, the centralized observation matrix, $\mathbf{C}_c$, is given by

$$\mathbf{C}_c = \begin{bmatrix} \mathbf{C}_L & \mathbf{0}_{3\times 6} & \mathbf{0}_{3\times 6} \\ -\mathbf{C}_L & \mathbf{C}_L & \mathbf{0}_{3\times 6} \\ \mathbf{0}_{3\times 6} & \mathbf{C}_L & -\mathbf{C}_L \\ \mathbf{0}_{1\times 6} & \mathbf{e}_z & \mathbf{0}_{1\times 6} \\ -\mathbf{C}_L & \mathbf{0}_{3\times 6} & \mathbf{C}_L \\ \mathbf{0}_{3\times 6} & -\mathbf{C}_L & \mathbf{C}_L \\ \mathbf{0}_{1\times 6} & \mathbf{0}_{1\times 6} & \mathbf{e}_z \end{bmatrix},$$

and the sparsity pattern matrix is defined as

$$\mathbf{E} = \begin{bmatrix} \mathbf{1}_{6\times 3} & \mathbf{0}_{6\times 7} & \mathbf{0}_{6\times 7} \\ \mathbf{0}_{6\times 3} & \mathbf{1}_{6\times 7} & \mathbf{0}_{6\times 7} \\ \mathbf{0}_{6\times 3} & \mathbf{0}_{6\times 7} & \mathbf{1}_{6\times 7} \end{bmatrix},$$

where $\mathbf{1}_{p\times q} \in \mathbb{R}^{p\times q}$ is a $p$ by $q$ matrix filled with ones. The sparsity matrix defines that UV $1$ has access to the first 3 entries of the total measurement vector, agent $2$ to the following 7 entries, and agent $3$ has access to the remaining ones. Note that, upon setting a measurement ordering for the centralized observation matrix, that same ordering must be kept when building the measurement vector of each agent.

### 4.5.1 Artificial relative position output

Consider the vector $\mathbf{y}_{ij}(t_k) \coloneqq \begin{bmatrix}\mathbf{0}_{1\times 3} & z_i(t_k) - z_j(t_k)\end{bmatrix}^T$, where $z_i(t_k)$ and $z_j(t_k)$ are the depth measurements obtained by UVs with indices $i$ and $j$, respectively. Letting $\boldsymbol{\Delta}_{ij}(t_k) = \mathbf{p}_i(t_k) - \mathbf{p}_j(t_k)$ and

$$\mathbf{P}_{ij}(t_k) = \begin{bmatrix} & \bar{\mathbf{D}}_{ij}(t_k) & \\ 0 & 0 & 1 \end{bmatrix},$$

with $\bar{\mathbf{D}}_{ij}$ defined as in (4.2), one then has that $\mathbf{P}_{ij}(t_k)\boldsymbol{\Delta}_{ij}(t_k) = \mathbf{y}_{ij}(t_k)$, from which it is possible to recover $\boldsymbol{\Delta}_{ij}(t_k)$ as

$$\boldsymbol{\Delta}_{ij}(t_k) = \left(\mathbf{P}_{ij}^T(t_k)\mathbf{P}_{ij}(t_k)\right)^{-1}\mathbf{P}_{ij}^T(t_k)\mathbf{y}_{ij}(t_k),$$

provided that $\mathbf{P}_{ij}^T(t_k)\mathbf{P}_{ij}(t_k)$ is invertible, which is the case if $z_i(t_k) - z_j(t_k) \neq 0$.

Rather than actual relative position measurements, the vector entries, $\mathbf{m}_{ij}[k] \in \mathbb{R}^3$, of $\mathbf{m}_i[k]$, are instead taken as

$$\mathbf{m}_{ij}[k+1] = \alpha_{ij}\boldsymbol{\Delta}_{ij}(t_{k+1}) + (1-\alpha_{ij})\mathbf{D}_{ij}(t_{k+1})(\hat{\mathbf{p}}_i(t_{k+1}|t_k) - \hat{\mathbf{p}}_j(t_{k+1}|t_k)),$$

with $\mathbf{D}_{ij}$ defined in (4.1). Each artificial relative position measurement is given by a weighted sum of the extracted position difference, $\boldsymbol{\Delta}_{ij}$, and the projected estimated position difference, $\mathbf{D}_{ij}(\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j)$, where $\hat{\mathbf{p}}_i$ and $\hat{\mathbf{p}}_j$ can be extracted from $\hat{\mathbf{x}}_i[k+1|k]$ and $\hat{\mathbf{x}}_j[k+1|k]$, respectively. This projection is done using $\mathbf{D}_{ij}$, which is built using the measured bearing angles via $\mathbf{d}_{ij}$, according to (4.1). Since the matrix $\mathbf{P}_{ij}^T\mathbf{P}_{ij}$ is close to singular if the height difference between the agents is close to zero, causing numerical instability, the weights are chosen as $\alpha_{ij} = |\mathbf{d}_{ij}^z|$. This ensures that when the extracted position difference, $\boldsymbol{\Delta}_{ij}$, is unreliable, the bearing measurement information can still be used.

# 5

# Simulation Results

In this section, the performance of each of the presented estimators is investigated. Firstly, in Section 5.1, a simple formation configuration, composed of only $5$ agents, performs a mock patrolling mission, whereby they have to visit the neighborhood of a predetermined set of waypoints, resulting in the trajectory presented in Fig. 5.1. Afterwards, in Section 5.2, the performance of the algorithms was then compared via Monte Carlo simulations, where the mean and root-mean-squared-error (RMSE) of the produced estimates was computed. For this comparison, a larger team, composed of $10$ agents performs a different mission, which requires diving deep underwater and changing its formation before doing so. The resulting trajectory for this mission is presented in Fig. 5.2.

## 5.1 Sample run

The patrolling mission was simulated and some results regarding the estimates obtained by each of the estimators are presented in this section. The spatial distribution of the agents remains constant throughout the whole mission, and their initial positions are depicted in Fig. 5.3. The nominal mission trajectory is depicted in Fig. 5.1. In the following section, the simulation and filter parameters are specified.

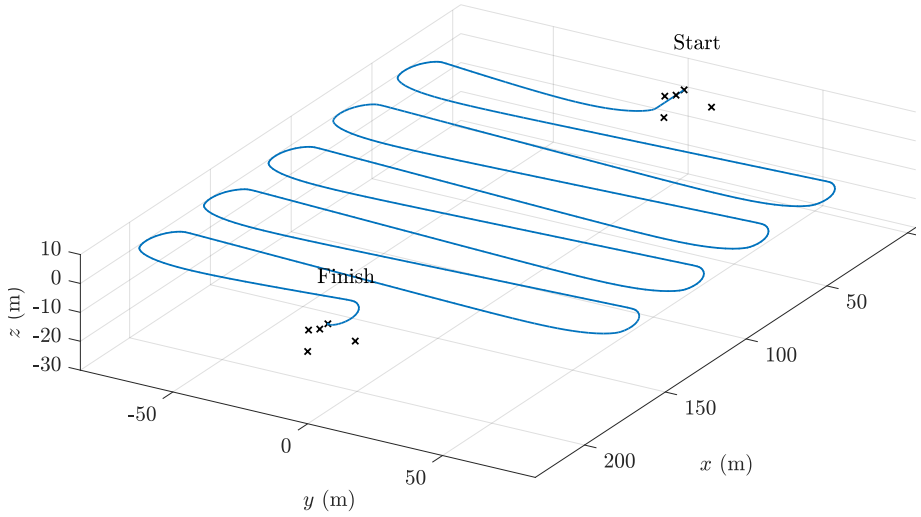



**Figure 5.1:** Nominal trajectory of the leader agent with index 1 for the patrolling mission.

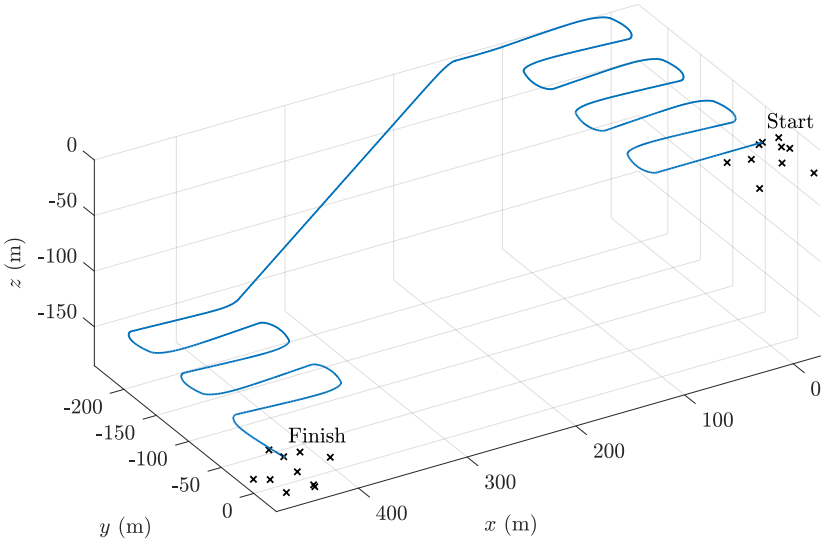



**Figure 5.2:** Nominal trajectory of the leader agent with index 1 for the Monte Carlo study.

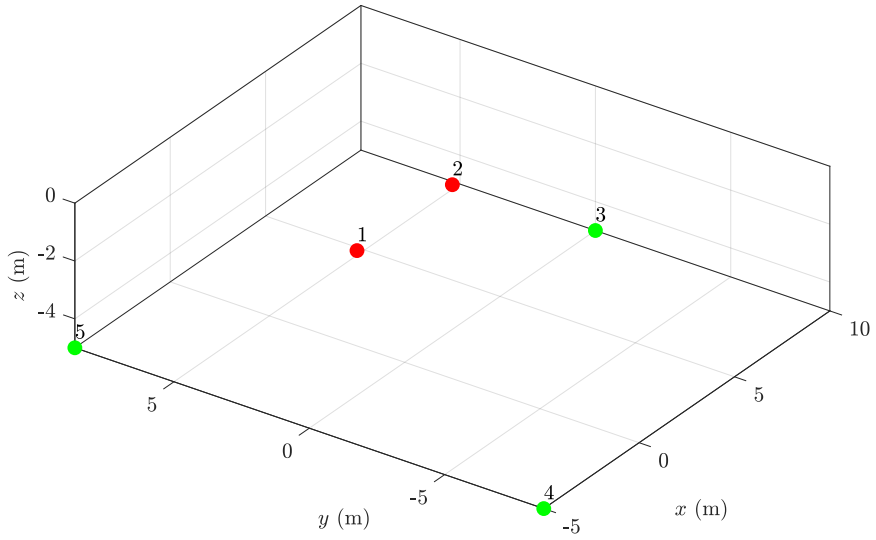



**Figure 5.3:** Spatial formation considered for the example mission.

### 5.1.1 Setup

The relative positions of the agents remain constant throughout the whole mission and the UVs are constrained to the measurement topology depicted in Fig. 5.4. This topology was chosen in order to allow for the most correlation to be created between the agents, so that the behavior of the algorithms is more differentiated, depending on how much cross-measurement information they can make use of. Additionally, since agents $3$, $4$, and $5$ are at the same depth, it serves to analyze how the static-gain observer with the artificial relative position measurements deals with the fact that the $\mathbf{P}_{ij}$ matrices, used to extract the relative position measurements, are close to singular.

The fluid velocity was assumed to be constant throughout the whole operating space, such that $\mathbf{v}_{f_i}(t) = \begin{bmatrix} 0.1 & -0.2 & 0 \end{bmatrix}^T (\mathrm{m\,s^{-1}})$ for all $i \in \mathcal{V}$, where $\mathcal{V} = \{1, 2, 3, 4, 5\}$ is the set of UVs. Each agent has access to $\mathbf{v}_{r_i}$ at a rate of $50\,\mathrm{Hz}$, corrupted by zero-mean white Gaussian noise, with covariance matrix $\mathbf{\Sigma}_u = 0.01^2\mathbf{I}_3$. The agents have access to their orientation, parameterized by roll, pitch and yaw Euler angles. These are also corrupted with independent zero-mean white Gaussian noise, with standard deviation of $0.05^\circ$ for the roll and pitch angles, and $0.3^\circ$ for the yaw angle. The control signal of each agent, $\mathbf{u}_i[k]$, is obtained through trapezoidal integration of $\mathbf{R}_i(t)\mathbf{v}_{r_i}(t)$ between measurement time steps, approximating (2.2).

UVs capture measurements every $T = 1\,\mathrm{s}$. The leader, agent $1$, can capture measurements of its position corrupted by zero mean white Gaussian noise, with covariance matrix $\mathbf{\Sigma}_{\text{pos}} = 0.1^2\mathbf{I}_3$. The depth measurements of the follower UVs are corrupted by zero mean white Gaussian noise, with standard deviation of $0.1\,\mathrm{m}$, and the measured bearing angles, $\theta$ and $\phi$, captured in the measuring agent's body frame, are corrupted by independent zero mean Gaussian noise, with standard deviation of $1^\circ$.

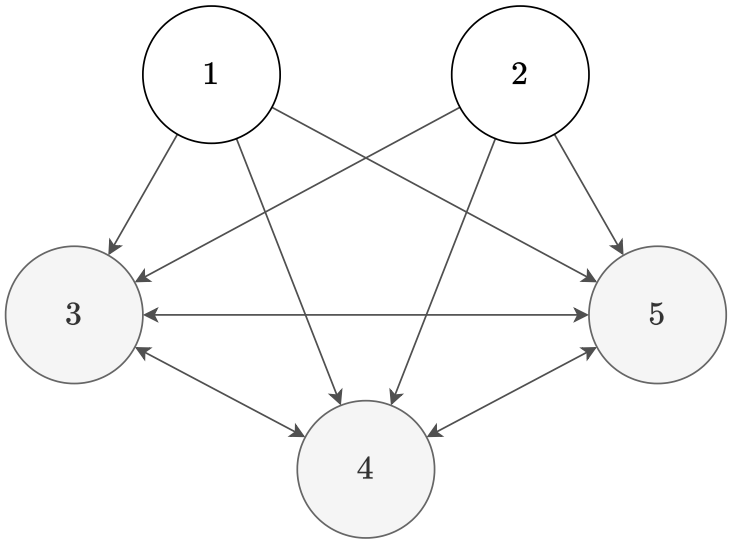



**Figure 5.4:** Measurement topology for example mission.

As for the filter parameters, all UVs have their process noise modeled using the covariance matrix $\mathbf{Q}_i = \mathrm{diag}(0.1^2\mathbf{I}_3, 0.01^2\mathbf{I}_3)$. The position measurements taken by the leader agent are modeled with the observation noise covariance matrix $\mathbf{R}_{\text{GPS}} = 0.1^2\mathbf{I}_3$, and the depth measurements taken by follower agents are modeled with standard deviation of $\sigma_d = 0.1\,\mathrm{m}$. For EKF-based approaches, the bearing measurements are modeled using the covariance matrix $\mathbf{R}_{\text{bear}} = 0.2^2\mathbf{I_2}$, and for the observers based on artificial linear systems, the relative measurements are modeled with the covariance matrix $\mathbf{R}_{\text{rel}} = \mathbf{I}_3$.

Since EKF-based approaches are not globally convergent, the initial state estimates of the UVs were sampled from a small neighborhood around their true initial state. For this effect, $\hat{\mathbf{x}}_i[0]$ was sampled from a normal distribution, centered at $\mathbf{x}_i[0]$, with covariance matrix $\mathbf{\Sigma}_{0_{\text{EKF}}} = \mathrm{diag}(10^2\mathbf{I}_3, 0.1^2\mathbf{I}_3)$. For the other approaches, the initial state estimates were sampled from a normal distribution centered at $\mathbf{x}_i[0]$, with covariance matrix $\mathbf{\Sigma}_{0_{\text{LTS}}} = \mathrm{diag}(25^2\mathbf{I}_3, 3^2\mathbf{I}_3)$.

At $k = 0$, all agents are assumed to be completely uncorrelated, such that $\mathbf{\Sigma}_{ij}[0] = \hat{\mathbf{\Phi}}_{ij}[0] = \mathbf{0}_6$ for all $i, j \in \mathcal{V}$. For all filters, the initial covariance matrix of each agent was set as $\mathbf{\Sigma}_{ii}[0] = \hat{\mathbf{\Sigma}}_{ii}[0] = \mathrm{diag}(0.1^2\mathbf{I}_3, 0.1^2\mathbf{I}_3)$ for visualization purposes, so that the algorithms do not converge too fast. For each estimator, the norm of the total state estimation error, $||\mathbf{x}(t) - \hat{\mathbf{x}}(t)||$, is presented, along with the state estimation results for agent 3.

### 5.1.2 Extended Kalman filter based approaches

Here, the results relating the EKF-based approaches are presented. The total state estimation error norm, $||\mathbf{x}[k] - \hat{\mathbf{x}}[k|k]||$, is presented in Fig. 5.5. There is little difference in the behavior of either approach, with the CEKF converging faster and both achieving similar steady-state performance. In

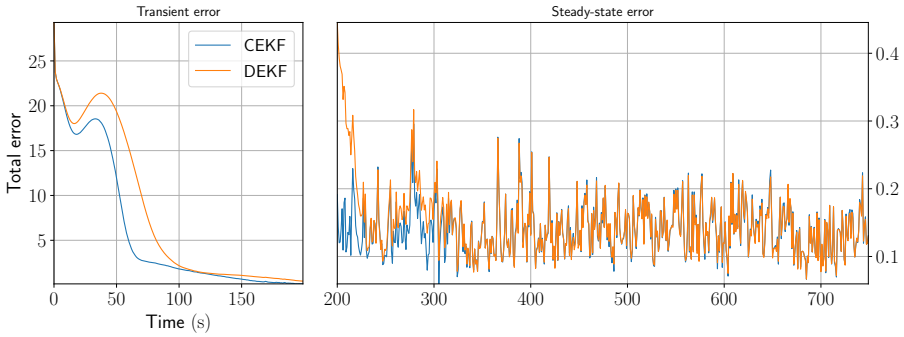
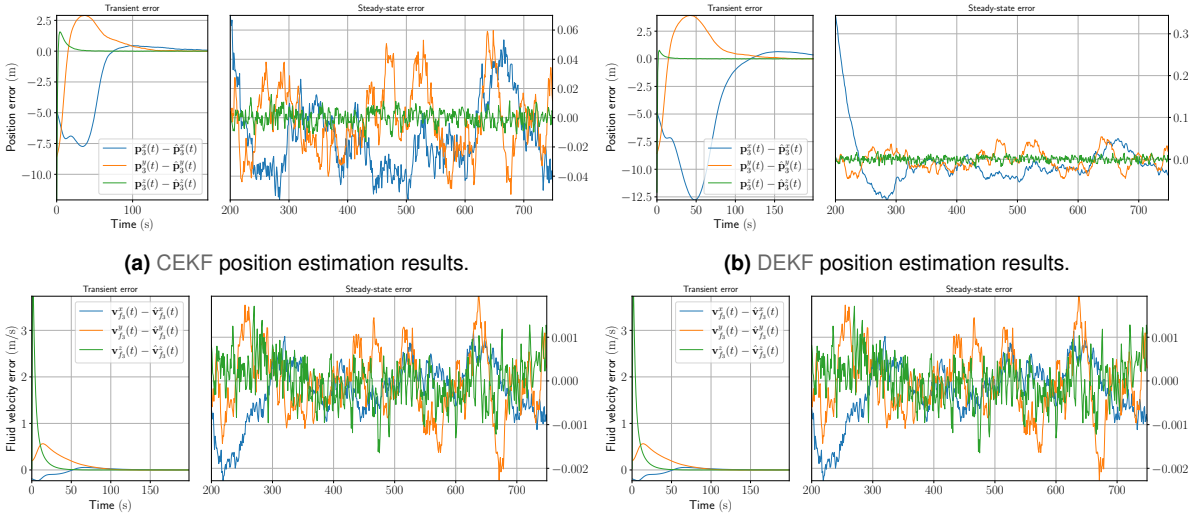



**Figure 5.5:** Total state estimation error norm for the EKF-based approaches.




**(a)** CEKF position estimation results.




**(b)** DEKF position estimation results.




**(c)** CEKF fluid velocity estimation results.




**(d)** CEKF fluid velocity estimation results.

**Figure 5.6:** Position and fluid velocity estimation results of agent 3 using the EKF-based estimators.

Fig. 5.6, the estimated position and fluid velocity coordinates of agent 3 is shown for both approaches. There is little difference in their estimation results, with the only noticeable one being the amplitude of the position coordinate estimation errors, which is larger than the CEKF, even though the measurement graph presents high connectivity. For visualization purposes, the estimation behavior was animated and made available online. The reader can scan the QR-codes presented in Fig. 5.7, or, alternatively, can follow the provided URL.

### 5.1.3 Linear Kalman filter approaches

Here, the results regarding the remaining estimators are presented. The algorithms are labeled as follows: the independently interconnected Kalman fitler (IKF) corresponds to the algorithm presented in Section 4.2, the centralized Kalman filter (CKF) to the one presented in Section 4.3; the DKF-FCS and the DKF-PCS correspond to the two variants of the algorithm presented in Section 4.4. The DKF-FCS

**(a)** CEKF animation URL: https://youtu.be/Ybpzp5qD2so. **(b)** DEKF animation URL: https://youtu.be/5oJN7BwBoAQ.

**Figure 5.7:** Position estimation animation QR-codes for the EKF-based approaches.

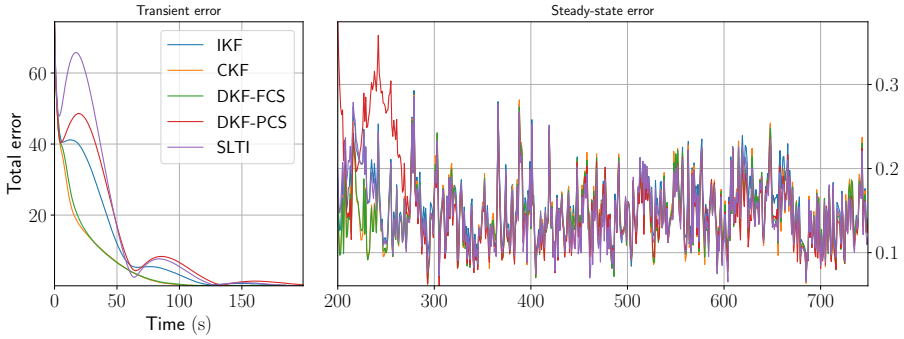
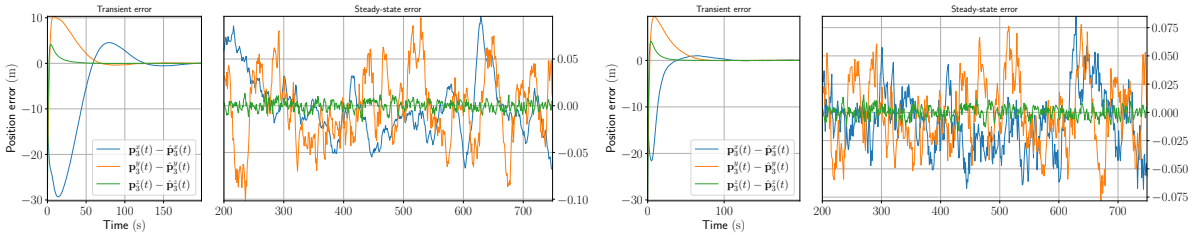



**Figure 5.8:** Total state estimation error norm for the linear Kalman filter approaches.

allows for the agents to communicate the necessary quantities to fill the joint state covariance matrix estimate (4.5), whereas the DKF-PCS ignores all the cross-covariance terms between neighbors of the measuring agent. Finally, the static-gain observer, presented in Section 4.5, is labeled as the static-gain estimator (SLTI).

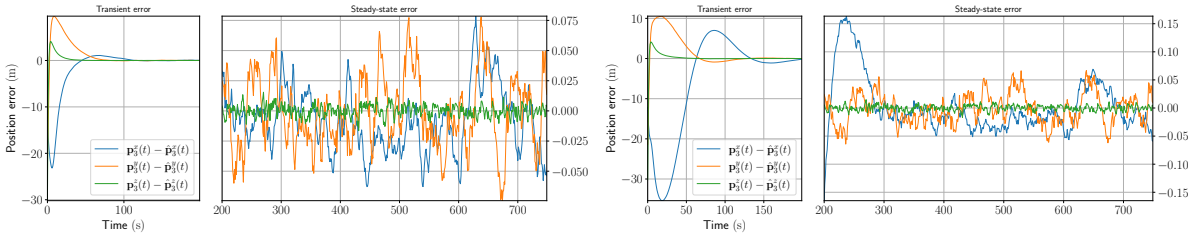
The total error norm of the estimators is presented in Fig. 5.8. Due to the high connectivity of the measurement graph, the performances of the centralized approach, CKF, and the decentralized approach which shares the most cross-measurement information, DKF-FCS, are very similar. Both the IKF and the DKF-PCS, however, converge more slowly than the former other two approaches, presenting a less dampened behavior. This is depicted in Figs. 5.9 and 5.10, where the position and fluid velocity coordinate estimation errors are presented, respectively. Even though the SLTI contains plenty of cross-measurement information in the form of the computed static-gains, it still presents an estimation behavior similar to the IKF and the DKF-PCS. However, this is due to the fact that this estimator is designed to perform in steady-state, where the specific static-gain coupling does present benefits. Similarly to the EKF-based approaches, the estimation results were also animated and the respective URLs can be reached via the QR-codes presented in Fig. 5.11.




**(a)** IKF position estimation results.



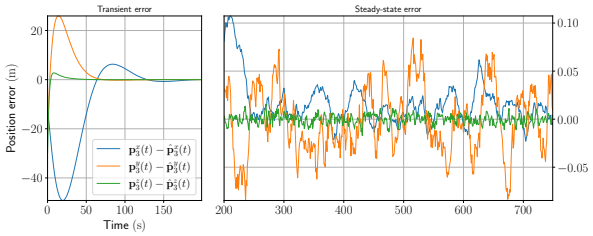

**(b)** CKF position estimation results.




**(c)** DKF-FCS position estimation results.

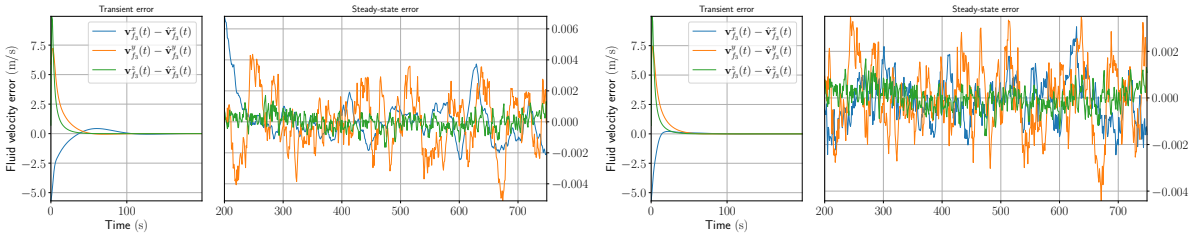



**(d)** DKF-PCS position estimation results.




**(e)** SLTI position estimation results.

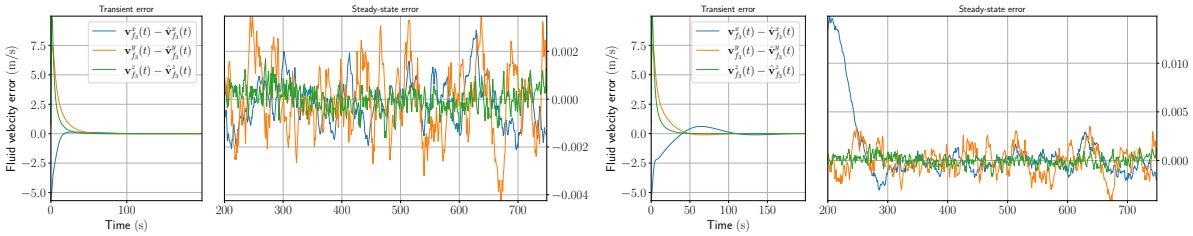
**Figure 5.9:** Position estimation results of agent 3 using the linear Kalman filter approaches.




**(a)** CEKF position estimation results.



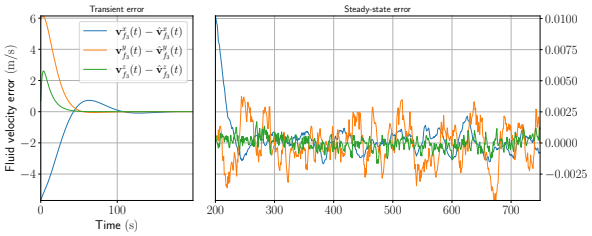

**(b)** DEKF position estimation results.




**(c)** CEKF fluid velocity estimation results.




**(d)** CEKF fluid velocity estimation results.




**(e)** CEKF fluid velocity estimation results.

**Figure 5.10:** Position and fluid velocity estimation results of agent 3 using the EKF-based estimators.

**(a)** IKF animation URL: https://youtu.be/yAb-D6UPALQ.

**(b)** CKF animation URL: https://youtu.be/2Pszre_FS7g.

**(c)** DKF-FCS animation URL: https://youtu.be/E4pTrPjiy-8.

**(d)** DKF-PCS animation URL: https://youtu.be/4PLmRnvSdQY.

**(e)** SLTI animation URL: https://youtu.be/Ve1OOII8haA.

**Figure 5.11:** Position estimation animation QR-codes for the linear Kalman filter approaches.

## 5.2 Monte Carlo results

Due to the high sensitivity of EKF-based approaches to the initial conditions of the problem, as well as to ascertain which algorithm gives the best estimates overall, a second mission was considered. This mission was simulated $N = 500$ times under different noise conditions in each run, and the results of this analysis are presented in this Section.

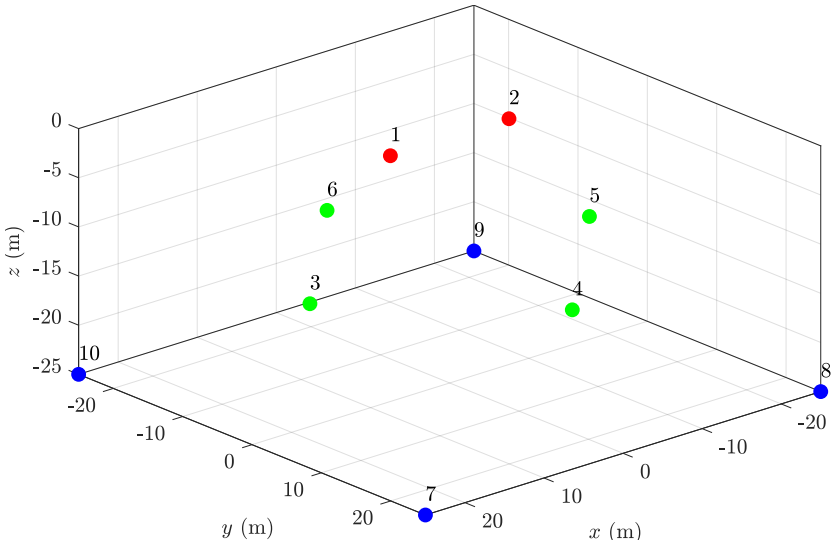
### 5.2.1 Setup

The setup considered for simulation analysis consists of a set of $10$ UVs performing a mission, whereby the agents must visit a set of waypoints while maintaining a certain formation. The agents start with the spatial distribution presented in Fig. 5.12a, and maintain this formation for a portion of their mission. Then, they change to a different spatial distribution, as represented in Fig. 5.12b before diving deeper underwater. They accomplish this by stopping until all have reached their waypoints, and then moving towards their next location in the formation until they reach it. Once all agents are in their respective locations, they move on to the next waypoint. Fig. 5.2 shows the nominal trajectory of agent 1.

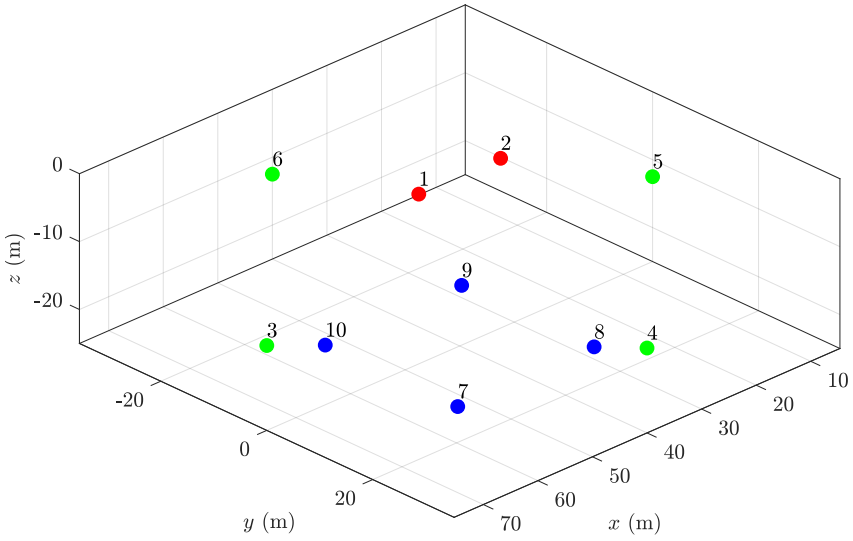
The parameters regarding the fluid velocity of the operating environment, as well as regarding the noise corrupting the measurements and control signals is the same as in Section 5.1. As for the filter parameters, the process noise covariance matrix of each UV is given as before, $\mathbf{Q}_i = \mathrm{diag}\left(0.05^2\mathbf{I}_3, 0.005^2\mathbf{I}_3\right)$. The sampling time is $T = 1\,\mathrm{s}$, the measurement noise covariance matrix for absolute position measurements is given by $\mathbf{R}_{\text{pos}} = 0.1^2\mathbf{I}_3$, and the noise corrupting depth measurements is modeled with a standard deviation of $\sigma_d = 0.1\,\mathrm{m}$. All agents are assumed to be completely uncorrelated at time $t = 0$, such that all the cross-covariance matrices and factors between agents are equal to the zero matrix, $\mathbf{0}_6$. In order to compute the steady-state gains for the static gain observer, (4.7) and (4.8) were propagated until $|\,\mathrm{tr}\,(\boldsymbol{\Sigma}[k+1|k+1]) - \mathrm{tr}\,(\boldsymbol{\Sigma}[k|k])\,| < 0.001$, where $\mathrm{tr}(\cdot)$ is the trace operator. The observers will be studied using two separate sets of tuning parameters, one tuned for convergence, and the other for steady-state behavior. The remaining filter parameters will be specified in each case.

The considered measurement topologies are presented in Figs. 5.13a and 5.13b. The agents are organized by tiers, such that $\mathcal{T}_0 = \{1, 2\} = \mathcal{V}_L$, $\mathcal{T}_1 = \{3, 4, 5, 6\}$ and $\mathcal{T}_2 = \{7, 8, 9, 10\}$ are the sets of agents in tiers 0, 1, and 2, respectively. Note that there are two leaders, agents $1$ and $2$, and they both are at the top of the formation, in tier 0. The cycles were made by flipping some of the edges (highlighted in green in Fig. 5.13b) between tiers 1 and 2 in the acyclical topology, and by introducing the blue edges around each tier of agents.

In order to evaluate the transient response and the steady-state performance of the presented estimators, RMSE of the position and fluid velocity estimates, obtained for each time instant from the

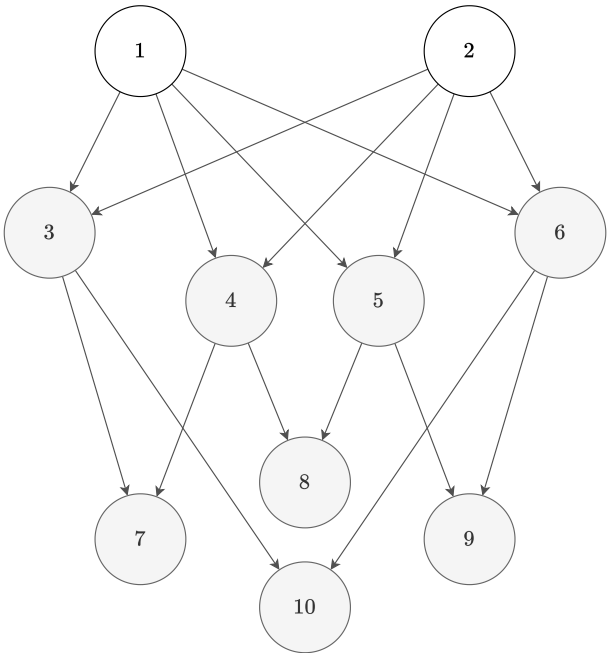



**(a)** Initial spatial configuration, which the agents maintain for the first half of the mission.

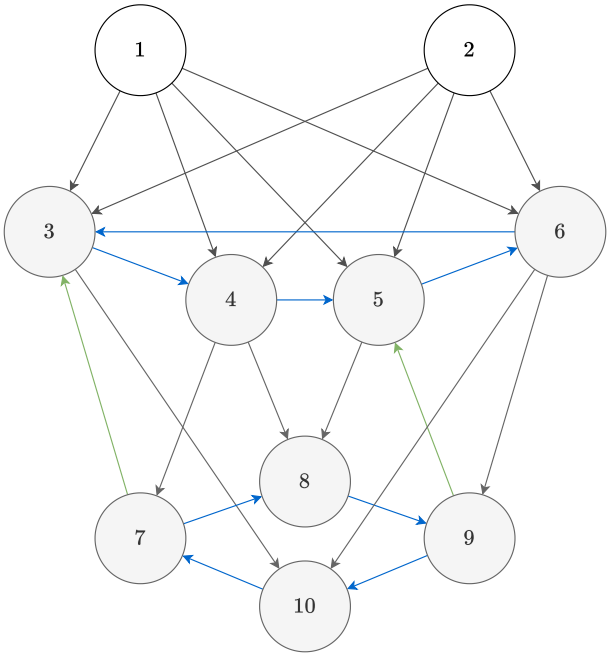



**(b)** Second spatial configuration, which the agents change to at $t = 400\,\mathrm{s}$.

**Figure 5.12:** Spatial configurations maintained by the agents throughout the mission.




**(a)** Acyclical topology measurement graph.




**(b)** Cyclical topology measurement graph.

**Figure 5.13:** Measurement topologies.

collection of Monte Carlo runs, was computed, such that

$$\mathrm{RMSE}(\mathbf{x}[k]) = \sqrt{\frac{\sum_{n=1}^{N} ||\mathbf{x}[k] - \hat{\mathbf{x}}^n[k]||^2}{N}},$$

where $\mathbf{x}[k]$ is the concatenation of the state vectors of all UVs at time $k$, and $\hat{\mathbf{x}}^n[k]$ is its estimate obtained in the $n^{\mathrm{th}}$ Monte Carlo run. Additionally, in order to investigate whether the estimators are biased, the mean error of the estimated quantities, for each time instant, was computed from the collection of Monte Carlo runs, as given by

$$\mathrm{mean}(\mathbf{x}[k]) = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}[k] - \hat{\mathbf{x}}^n[k].$$

Due to the specific tuning parameters used in order to achieve good steady-state performance, the algorithms can take a while to converge. In order to improve the convergence time of these solutions, separate tuning parameters can be used, depending on the phase of the mission. Initially, if the agents are completely unlocalized, they can adopt a certain set of tuning parameters optimized for convergence speed. After a certain amount of time, they can change to parameters optimized for steady-state performance. These parameters were set empirically, as is the case with most nonlinear estimation problems. Taking this into account, the convergence of the algorithms was studied separately from their steady-state performance. Firstly, the convergence behavior of the presented solutions is studied. Then, the steady-state behavior is analyzed by considering small initial state estimation errors.

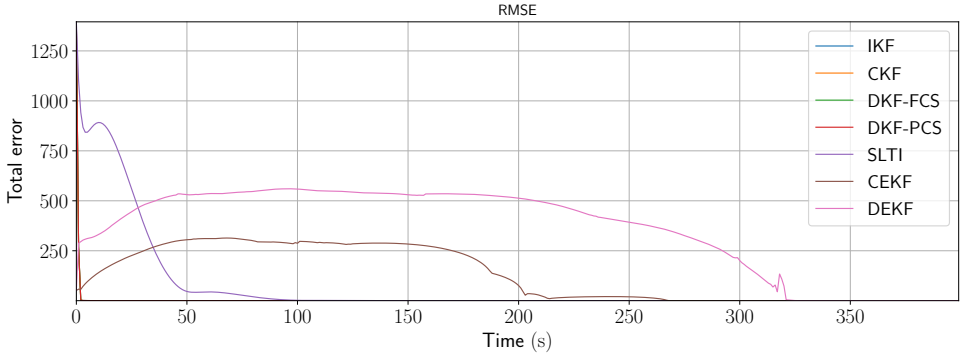
### 5.2.2 Convergence analysis

For the EKF-based approaches, the initial state estimate for each of the Monte Carlo runs was sampled from a Gaussian distribution with mean identical to the true value, and with covariance matrix $\mathbf{\Sigma}_0 = \mathrm{diag}(10^2\mathbf{I}_3, \mathbf{I}_3)$. During previous experiments, it was noted that, for this setting, the linear estimators present globally convergent dynamics, such that their initial state estimates were sampled from a Gaussian distribution with covariance matrix $\mathbf{\Sigma}_0 = \mathrm{diag}(250^2\mathbf{I}_3, 50^2\mathbf{I}_3)$, and mean at the true value. The initial state covariance matrix of each agent was likewise set as $\mathbf{\Sigma}_{ii}[0|0] = \mathrm{diag}(50^2\mathbf{I}_3, 10^2\mathbf{I}_3)$ and the noise affecting the relative measurement between UVs is modeled with the covariance matrix $\mathbf{R}_b = 0.01^2\mathbf{I}_2$ for the EKF-based approaches, and $\mathbf{R}_b = 0.5^2\mathbf{I}_3$ for the remaining ones. The initial 400 seconds of the full mission were simulated $N = 500$ times, with the specified simulation and tuning parameters, considering independent noise vectors for each run.

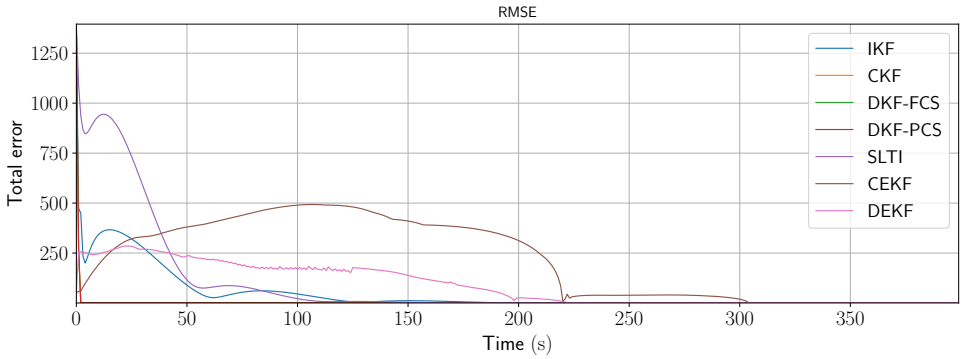
It is well known that EKF-based approaches do not guarantee global convergence and require relatively accurate initial state estimates. How accurate these initial estimates must be depends on both the spatial distribution of the agents in the formation, as well as their measurement topology. The number of convergent runs, under each topology, for both the CEKF and DEKF, are presented in Table 5.1. Simi-

**Table 5.1:** Number of convergent runs (and respective convergence percentage) for each EKF-based estimator under the acyclical and cyclical measurement topologies.

| | Acyclical | Cyclical |
|---|---|---|
| CEKF | 443 $(88.6\,\%)$ | 448 $(89.6\,\%)$ |
| DEKF | 463 $(92.6\,\%)$ | 487 $(97.4\,\%)$ |




**(a)** Acyclical topology RMSE results of observers tuned for convergence speed.




**(b)** Cyclical topology RMSE results of observers tuned for convergence speed.

**Figure 5.14:** RMSE results of algorithms tuned for convergence speed.

lar results were obtained for different formation configurations, and show that the centralized approach is more sensitive to the initial conditions of the agent configuration than its decentralized counterpart, emphasizing the inherent robustness of decentralized approaches. Since the CEKF shares more information, its estimates are also more affected by erroneous initial state estimates, hence the added difficulty in converging. As for the linear estimators, their estimates converged to the true solution on all runs. The RMSE of the convergent estimates obtained by each of the considered estimators, for both measurement topologies, is presented in Fig. 5.14.

As shown in Fig. 5.14a, all the linear observers, except the SLTI, produced estimates that converged to the true solution in just a few time-steps when the measurement topology is acyclical. The SLTI does not achieve this because it is a static-gain observer designed for steady-state performance, which typically involves low gains, thus the slow convergence.

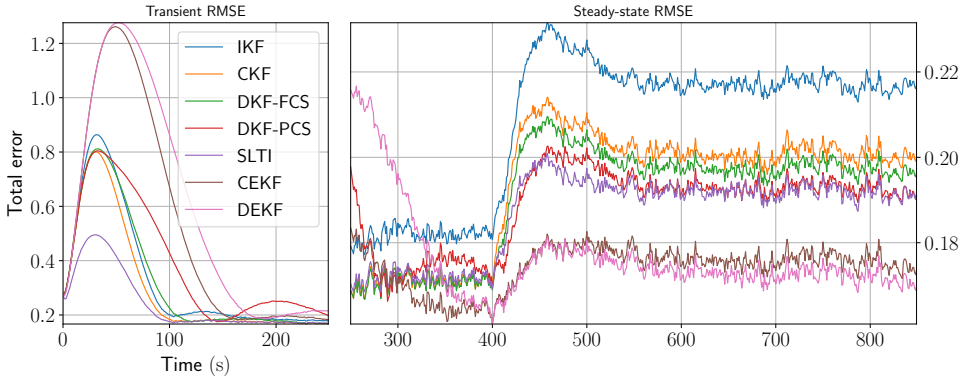
Upon introduction of new edges to form cycles, the convergence speed of IKF is severely affected, as shown in Fig. 5.14b. This is related to the fact that each agent following this observer design produces its estimates independently of the other UVs, disregarding possible cross-measurement information. The other linear time-variant Kalman filter approaches, however, converge to the solution unaffected by the presence of cycles in the measurement graph. Similarly, the SLTI's convergence is only slightly affected by the introduction of these edges. Note that, in both cases, the EKF-based approaches might start diverging at first, or converge to a non-optimal solution. However, the turns present in the trajectory allow for enough perturbation that the algorithm leaves this non-optimal solution and, either diverges, or finds a new equilibrium, which might or not be the true solution. This is the reason why the RMSE curves of the EKF-based approaches look so erratic.

### 5.2.3 Steady-state performance

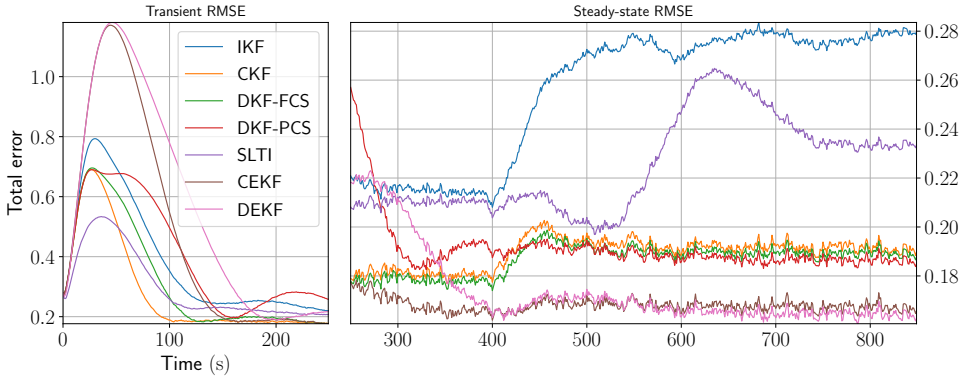
Here, the RMSE of the estimates obtained with each estimator, tuned for steady-state performance, are compared. Since the observers were optimized for steady-state performance, their convergence is quite slow. Thus, the initial state estimates were set very close to their real value by sampling them from a Gaussian distribution, centered at the real state vectors and with covariance matrix $\mathbf{\Sigma}_0 = \mathrm{diag}(0.05^2\mathbf{I}_3, 0.01^2\mathbf{I}_3)$. The initial covariance matrix of each agent was set as $\mathbf{\Sigma}_{ii}[0|0] = \mathbf{\Sigma}_0$ and the artificial measurements' covariance matrix was set as $\mathbf{R}_b = 0.3^2\mathbf{I}_2$ for the EKF-based estimators, and $\mathbf{R}_b = 3^2\mathbf{I}_3$ for the remaining ones. The RMSE results for the acyclical topology are presented in Fig. 5.15a, and for the cyclical measurement topology in Fig. 5.15b.

The IKF has the worst performance of the considered estimators. As mentioned before, this is because IKF keeps no cross-measurement information, whereas all the other estimators do some way or the other. Also, contrary to what one would expect, the CKF does not have the best performance out of the estimators in its class, which is due to the presence of a non-zero error bias in the artificial quantities built from the bearing angles. Namely, the expected value of the direction vector $\mathbf{d}_{ij}$, built according to (4.3) considering noisy measurements, is not equal to the nominal direction vector built from noiseless bearing angles. Since the centralized approaches make use of more information to produce their estimates and, in this case, fail to account for biasing errors, this additional information ends up being detrimental to the filter's performance, depending on the tuning parameters. The CEKF and the DEKF have the best performance, which is because they use the bearing angles directly (after rotation to the inertial frame), unlike the other algorithms, and thus are not affected by the measurement error bias originating from the construction of the direction vector. Note, however, that rotating the bearing angles to the inertial frame can still result in the introduction of a bias into the estimation error, resulting in the the DEKF presenting better estimation capabilities than CEKF.

The introduction of cycles was detrimental to the performance of IKF, which, again, is due the fact that




**(a)** Acyclical topology RMSE results of observers tuned for steady-state performance.

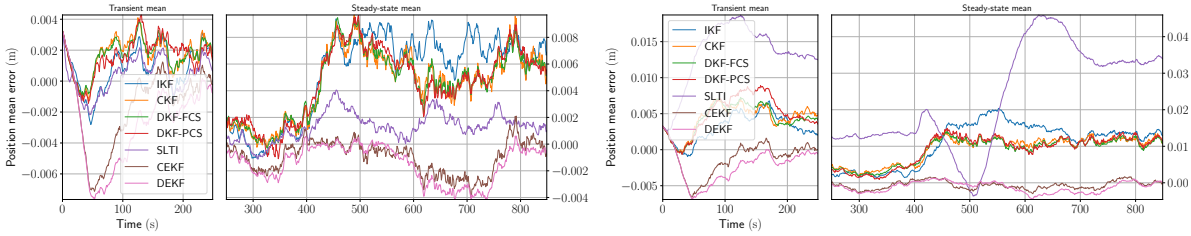



**(b)** Cyclical topology RMSE results of observers tuned for steady-state performance.

**Figure 5.15:** RMSE results of algorithms tuned for steady-state performance.

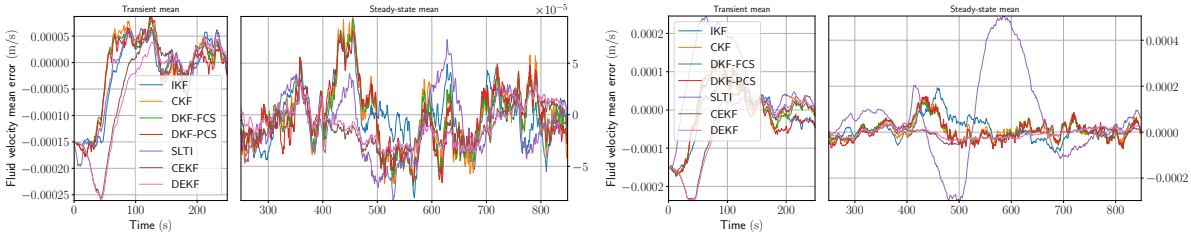
it keeps no cross-measurement information, and thus the estimation errors are re-fed to the estimation algorithm with no regard for where they originated from, resulting in difficulties in converging and higher RMSE. The SLTI has also seen its performance severely decreased. However, this is due to the fact that the blue edges concern agents which are at the same height, and, as mentioned before, the matrices $\mathbf{P}_{ij}$ become singular when this happens, resulting in numerical instability. The CEKF and the DEKF have benefited from the introduction of cycles, since their RMSE is lower. Likewise, the other estimators, the CKF, DKF-FCS and DKF-PCS, saw their performance slightly improved. The introduction of new edges and cycles has allowed the algorithms to work with more information and create better correlation between the agents. However, the biasing errors for the the linear estimators make it so fine-tuning is still required in order to balance measurement information with the amount of biasing error, which depends on both the measurement topology and spatial formation.

In the following, the presence of a non-zero estimation error bias is investigated. For that effect, the mean results for the $x$ coordinate of estimated positions and fluid velocities of UVs 3 and 9 are presented in Figs 5.16 and 5.17. Regardless of the measurement topology, there is a clear non-zero estimation error bias for the linear estimators, though it is very small given these parameters, and the estimators still provide a good enough estimate for most purposes. This bias is dependent on the spatial formation of the agent, and comes mostly from the construction of the artificial direction vectors, $\mathbf{d}_{ij}$. Since the EKF-based approaches use the bearing angles directly (after rotation to the inertial frame), the noise affecting the measurement vector of these approaches is closer to a normally distributed noise than the one affecting the other approaches, hence why there is no noticeable bias in these approaches' results. As for the fluid velocity, there is no clear estimation error bias, though there is no guarantee that there will not be one, since the control input, $\mathbf{u}[k]$, is computed using the noisy rotation matrix.




**(a)** Acyclical topology mean $\mathbf{p}_3^x$ estimation error of observers tuned for steady-state performance.

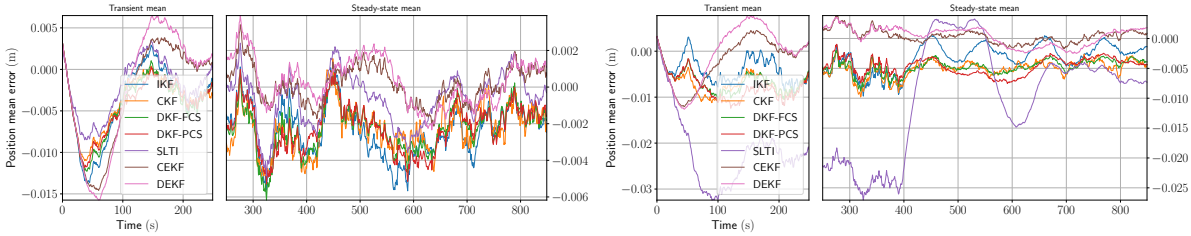
**(b)** Cyclical topology mean $\mathbf{p}_3^x$ estimation error of observers tuned for steady-state performance.




**(c)** Acyclical topology mean $\mathbf{v}_{f_3}^x$ estimation error of observers tuned for steady-state performance.

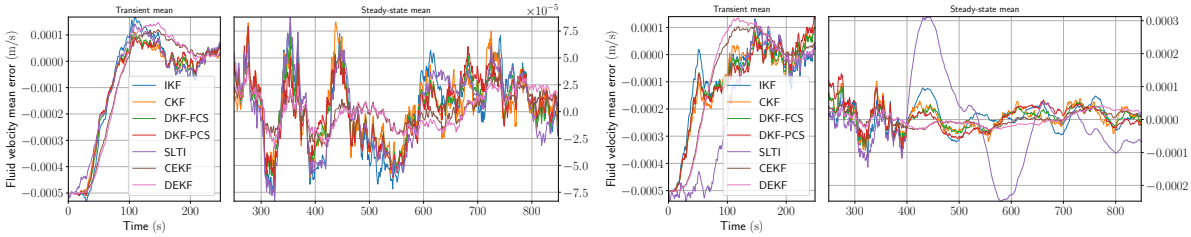
**(d)** Cyclical topology mean $\mathbf{v}_{f_3}^x$ estimation error of observers tuned for steady-state performance.

**Figure 5.16:** Mean $\mathbf{p}_3^x$ and $\mathbf{v}_{f_3}^x$ estimation error of observers tuned for steady-state performance.




**(a)** Acyclical topology mean $\mathbf{p}_9^x$ estimation error of observers tuned for steady-state performance.

**(b)** Cyclical topology mean $\mathbf{p}_9^x$ estimation error of observers tuned for steady-state performance.




**(c)** Acyclical topology mean $\mathbf{v}_{f_9}^x$ estimation error of observers tuned for steady-state performance.

**(d)** Cyclical topology mean $\mathbf{v}_{f_9}^x$ estimation error of observers tuned for steady-state performance.

**Figure 5.17:** Mean $\mathbf{p}_9^x$ and $\mathbf{v}_{f_9}^x$ estimation error of observers tuned for steady-state performance.

# 6

# Conclusions

This work gave a brief overview of the current state of the literature on decentralized navigation, discussing also some relevant work in fields that are similar to it, such as decentralized control and sensor network localization. Some of the current navigation approaches were presented more in depth in this work, namely those in [18, 19, 26], and some algorithms were developed by combining techniques presented in these approaches. In total, seven algorithms to the navigation problem were presented in this thesis. Two of them based on the extended Kalman filter, and the remaining ones based on artificial measurements, which are linear relative to the state of the observer. The effect of the measurement topology on these algorithms was also studied, namely the advantages and disadvantages of considering acyclical or cyclical measurement topologies.

It was found that the cross-measurement information has a dampening-like effect on the produced estimate dynamics, making cross-measurement information, in the form of cycles, valuable for the convergence of the algorithm. On the other hand, this extra information can also be detrimental to the estimation capabilities of the Kalman filters due to possible biased errors, such that fine-tuning is required. While the EKF-based approaches do not present as good convergence properties as the linear estimators, their produced estimates are, generally, of higher quality. The reason for this was investi-

gated and was found to be the presence of non-zero error biases in the artificial measurements. A brief analysis of this biased error was made in Section 4.1.1. The linear observers produced estimates which converged to the true state on all the considered experiments, suggesting that these approaches are globally convergent in the considered scenarios, which makes it so an initial setup of the agents is not necessary, increasing the time efficiency of missions.

An unexpected consequence of the presence the biased errors was the fact that the centralized estimators tended to produce worse estimates, in terms of their RMSE, than their decentralized counterparts. Since centralized approaches share so much information, they are very prone to unexpected errors, such as the biases mentioned. The decentralized approaches, however, are not as affected by this kind of errors, which are always present in practical applications.

Future works include the study of techniques to mitigate this biased error, while maintaining the convergence qualities of the linear estimators. Finally, a formal analysis of the convergence of the algorithms is necessary, in order to assess the conditions under which the estimates converge to the true solution.

# Bibliography


[1] A. Finn and S. Scheding, "Developments and challenges for autonomous unmanned vehicles," *Intelligent Systems Reference Library*, vol. 3, pp. 128–154, 2010.

[2] N. Mohamed, J. Al-Jaroodi, I. Jawhar, A. Idries, and F. Mohammed, "Unmanned aerial vehicles applications in future smart cities," *Technological Forecasting and Social Change*, vol. 153, p. 119293, 2020.

[3] H. Xiang and L. Tian, "Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (uav)," *Biosystems engineering*, vol. 108, no. 2, pp. 174–190, 2011.

[4] D. G. Schmale Iii, B. R. Dingus, and C. Reinholtz, "Development and application of an autonomous unmanned aerial vehicle for precise aerobiological sampling above agricultural fields," *Journal of Field Robotics*, vol. 25, no. 3, pp. 133–147, 2008.

[5] "Flash forest homepage." https://flashforest.ca/. Accessed: 2021-10-12.

[6] Y. Allard and E. Shahbazian, "Unmanned underwater vehicle (uuv) information study," tech. rep., OODA Technologies Inc Montreal, Quebec Canada, 2014.

[7] C. Rodrigues, D. Nunes, D. Clemente, N. Mathias, J. Correia, P. Rosa-Santos, F. Taveira-Pinto, T. Morais, A. Pereira, and J. Ventura, "Emerging triboelectric nanogenerators for ocean wave energy harvesting: state of the art and future perspectives," *Energy & Environmental Science*, vol. 13, no. 9, pp. 2657–2683, 2020.

[8] A. Ryan, M. Zennaro, A. Howell, R. Sengupta, and J. K. Hedrick, "An overview of emerging results in cooperative uav control," in *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, vol. 1, pp. 602–607, IEEE, 2004.

[9] F. Borrelli, T. Keviczky, and G. J. Balas, "Collision-free uav formation flight using decentralized optimization and invariant sets," in *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, vol. 1, pp. 1099–1104, IEEE, 2004.

[10] A. Richards and J. How, "Decentralized model predictive control of cooperating uavs," in *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, vol. 4, pp. 4286–4291, IEEE, 2004.

[11] R. Olfati-Saber and R. M. Murray, "Distributed cooperative control of multiple vehicle formations using structural potential functions," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 495–500, 2002.

[12] F. Schiano, A. Franchi, D. Zelazo, and P. R. Giordano, "A rigidity-based decentralized bearing formation controller for groups of quadrotor uavs," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5099–5106, IEEE, 2016.

[13] B. D. Anderson, B. Fidan, C. Yu, and D. Walle, "Uav formation control: Theory and application," in *Recent advances in learning and control*, pp. 15–33, Springer, 2008.

[14] S. Datta, C. Klinowski, M. Rudafshani, and S. Khaleque, "Distributed localization in static and mobile sensor networks," in *2006 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pp. 69–76, IEEE, 2006.

[15] S. Safavi, U. A. Khan, S. Kar, and J. M. Moura, "Distributed localization: A linear theory," *Proceedings of the IEEE*, vol. 106, no. 7, pp. 1204–1223, 2018.

[16] A. Bahr, J. J. Leonard, and M. F. Fallon, "Cooperative localization for autonomous underwater vehicles," *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 714–728, 2009.

[17] S. S. Kia, S. Rounds, and S. Martinez, "Cooperative localization for mobile agents: A recursive decentralized algorithm based on kalman-filter decoupling," *IEEE Control Systems Magazine*, vol. 36, no. 2, pp. 86–101, 2016.

[18] L. Luft, T. Schubert, S. I. Roumeliotis, and W. Burgard, "Recursive decentralized localization for multi-robot systems with asynchronous pairwise communication," *The International Journal of Robotics Research*, vol. 37, no. 10, pp. 1152–1167, 2018.

[19] D. Santos, P. Batista, P. Oliveira, and C. Silvestre, "Decentralised navigation systems for bearing-based position and velocity estimation in tiered formations," *International Journal of Systems Science*, pp. 1–22, 2021.

[20] D. Viegas, P. Batista, P. Oliveira, and C. Silvestre, "Decentralized state observers for range-based position and velocity estimation in acyclic formations with fixed topologies," *International Journal of Robust and Nonlinear Control*, vol. 26, no. 5, pp. 963–994, 2016.

[21] P. Batista, C. Silvestre, and P. Oliveira, "Navigation systems based on multiple bearing measurements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 4, pp. 2887–2899, 2015.

[22] G. C. Calafiore, L. Carlone, and M. Wei, “A distributed technique for localization of agent formations from relative range measurements,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 42, no. 5, pp. 1065–1076, 2012.

[23] C. Soares, P. Ji, J. Gomes, and A. Pascoal, “Diesel: Distributed self-localization of a network of underwater vehicles,” in *OCEANS 2017-Anchorage*, pp. 1–6, IEEE, 2017.

[24] P. Mendes and P. Batista, “A study on cooperative navigation of auvs based on bearing measurements,” in *proceedings of Global OCEANS 2021: San Diego — Porto*, IEEE, 2021.

[25] J. Reis, M. Morgado, P. Batista, P. Oliveira, and C. Silvestre, “Design and experimental validation of a usbl underwater acoustic positioning system,” *Sensors*, vol. 16, no. 9, p. 1491, 2016.

[26] D. Viegas, P. Batista, P. Oliveira, and C. Silvestre, “Discrete-time distributed kalman filter design for formations of autonomous vehicles,” *Control Engineering Practice*, vol. 75, pp. 55–68, 2018.