Vehicle Network Localization Based on Ranges and Bearings with Packet Loss

Adriana Maria Tavares Pereira da Silva

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisors: Prof. Cláudia Alexandra Magalhães Soares
Prof. João Pedro Castilho Pereira Santos Gomes

Examination Committee
Chairperson: Prof. Fernando José Parracho Lau
Supervisor: Prof. Cláudia Alexandra Magalhães Soares
Member of the Committee: Prof. Marko Beko

July 2019
Acknowledgments

I would like to show my sincerest gratitude to the Professors who accompanied me throughout this journey, Professor Cláudia Soares and Professor João Pedro Gomes. Their guidance, helpfulness and constant motivation are one of the main reasons I was able to successfully complete this work.

I also take this opportunity to thank my family, for having supported me through all these years, and also for their never ending support and encouragement.

And finally, to my friends, for always being there, in the ups and downs of this adventure.
Resumo

Num contexto real, vários sistemas necessitam de estimar a sua posição no espaço de modo a desempenharem as suas funções. No entanto, em áreas em que o GPS não é uma opção, como dentro de um edifício ou em ambiente subaquático, as técnicas para produzir uma boa estimativa de posição estão sujeitas a ruído, e a outros fenómenos como medidas com atraso e perda de pacotes. Esta tese aborda este problema utilizando métodos estatísticos em que a perda de pacotes corresponde a dados em falta numa sequência temporal de uma base de dados.

Inicialmente, o problema de localização é formulado através do estimador de posição de Máxima Verosimilhança para uma rede genérica de agentes com acesso a medidas de distância e ângulo entre eles, sem perdas de informação. Esta abordagem gera um problema de otimização não convexo, sendo aplicada uma relaxação convexa. Posteriormente, para lidar com a perda de pacotes, é incorporada no problema, que já se encontra numa perspetiva temporal, a probabilidade a priori das medidas em falta, obtendo-se o estimador de Máxima Probabilidade a posteriori. Finalmente, é apresentado um algoritmo de otimização para obter a solução do problema. O método desenvolvido é robusto, convexo e não requer uma configuração específica da rede de agentes. Para além disso, atinge precisões semelhantes ou melhores do que o estado da arte.

Palavras-chave: Perda de pacotes, Estimador de Máxima Verosimilhança, Estimador Máximo a Posteriori, Otimização convexa, Aprendizagem estatística
Abstract

In a real-world scenario, numerous systems need to estimate their position in space to perform their assigned tasks. However, for cases where GPS tracking is not an option (like indoor and in underwater environments), the techniques to produce a reliable position estimate are subject to noise and other phenomena, like random transmission delays and packet loss. The present thesis aims to address this problem resorting to a statistical algorithm, in which lost packages correspond to missing entries in a temporal sequence of a database.

Initially, the localization problem is formulated through a maximum likelihood estimator for a generic network of nodes that have access to the distance and angle measurements between them, with no loss of information of any kind. This approach results in a non-convex optimization problem, and a convex relaxation is used. The next step, to deal with package loss, faces this problem from a temporal perspective and incorporates in it the prior probability of the missing measurements, which results in a maximum a posteriori estimator for a generic network. A distributed optimization algorithm is presented to obtain the solution of the convex problem. The method developed is robust, convex and does not require any particular anchor configuration. It improves accuracy in one order of magnitude, when compared with a state-of-the-art method.

Keywords: Missing data, Maximum Likelihood Estimator, Maximum a Posteriori Estimator, Convex Optimization, Statistical Learning
Contents

Acknowledgments ................................................................. iii
Resumo ................................................................. v
Abstract .............................................................. vii
List of Tables .............................................................. xiii
List of Figures ............................................................... xv
Notation ............................................................... xvii
Glossary .............................................................. xix

1 Introduction 1
   1.1 Motivation ......................................................... 1
   1.2 Objectives and Contributions ........................................ 2
   1.3 Thesis Outline ....................................................... 2

2 Background 4
   2.1 Missing Data Problem .............................................. 4
   2.1.1 Distributed Fusion Kalman Filter ................................ 4
   2.1.2 Variance-constrained Approach to Recursive State Estimation ........................................... 5
   2.1.3 Asynchronous Gradient-based Localization ................................................................. 6
   2.1.4 Robust Kalman Filter Design .................................. 6
   2.1.5 Event-Based Variance-Constrained $H_{\infty}$ Filter ......................................................... 7
   2.1.6 Matrix Completion .............................................. 7
   2.2 Wireless Sensor Network ............................................ 8
   2.2.1 Graph Concepts .................................................. 8
   2.2.2 Network Localization ........................................... 9
   2.3 Estimation Theory Overview ....................................... 12
   2.3.1 Maximum Likelihood Estimator ................................ 12
   2.3.2 Maximum a Posteriori Estimator .............................. 13
   2.4 Performance Evaluation ............................................ 14
   2.4.1 Scalability ....................................................... 15
   2.4.2 Accuracy ....................................................... 15
   2.4.3 Resilience to Noise and Errors ............................... 16
List of Tables

5.1 MAE of MLE with noiseless measurements. ................................................. 54
5.2 MPE of MLE with noisy measurements. ...................................................... 55
5.3 MPE of the MAP estimator with missing measurements. ............................ 58
5.4 RMSE of the distances estimated $\Omega$ and $\Upsilon$ averaged over 50 MC trials. 58
5.5 MPE of the estimated positions and RMSE of the estimated distances $\Omega$ and $\Upsilon$ when vehicles do not travel in formation. ......................................................... 62
5.6 MPE of the estimated positions and RMSE of the estimated distances $\Omega$ and $\Upsilon$ in the helix trajectory. ................................................................. 64
5.7 Equivalence between concentration parameter and equivalent standard deviation. 64
5.8 Variation of the MPE with distance standard deviation. .............................. 65
5.9 Variation of the MPE with angle standard deviation. .................................... 65
5.10 Variation of the MPE with the percentage of missing measurements. ............ 66
5.11 MPE and RMSE of the MAP estimator and the state-of-the-art method throughout the linear trajectory. ................................................................. 68
5.12 MPE and RMSE of the MAP estimator and state-of-the-art method throughout the spiral trajectory. ................................................................. 69
A.1 MPE and RMSE of the numerical example of the state-of-the-art method. ........ 79
List of Figures

2.1 Representation of the architecture of a Wireless Sensor Network. 8
2.2 Illustrative example of the position of a node relative to the position of an anchor. 10
2.3 Illustrative example of the localization of a node in a 2D space. 10
2.4 Image of an AUV, the The Autosub6000. 10
2.5 Illustrative example of an Underwater Network. 10
2.6 Representation of the Maximum Likelihood Estimator. 13

3.1 Graph of a convex function. 24
3.2 Representation of function $f = (|| x || - d)^2$ and its convexity analysis. 25
3.3 Representation of function $f = \frac{x}{|| x ||}$ and its convexity analysis. 25
3.4 Representation of the original function $f(x)$ and its relaxation $f(x)_+$. 26
3.5 Representation of the relaxation of distance terms. 27
3.6 Representation of the relaxation. 28
3.7 Illustrative example of how the MAP estimator handles missing measurements. 32

4.1 Illustrative representation of the virtual network defined by the time window. 39
4.2 Representation of the quadratic lower bound of a strongly convex function. 43
4.3 Representation of the quadratic upper bound of $f$ with Lipschitz continuous gradient. 44

5.1 Representation of a linear trajectory. 50
5.2 Representation of a spiral trajectory. 50
5.3 Representation of a linear trajectory where the vehicles do not travel in formation. 51
5.4 Representation of a helix trajectory. 51
5.5 Estimated positions with noiseless distance measurements. 52
5.6 Estimated positions with noiseless distance and angle measurements. 52
5.7 Estimated positions with noiseless distance measurements in a spiral trajectory. 53
5.8 Estimated positions with noiseless distance and angle measurements in a spiral trajectory. 53
5.9 Estimated positions in a linear trajectory with noisy distance and angle measurements. 55
5.10 Estimated positions in a spiral trajectory with noisy distance and angle measurements. 55
5.11 Estimated positions in a linear trajectory with 5% missing measurements. 56
5.12 Estimated positions in a linear trajectory with 50% missing measurements. 57
5.13 Estimated positions in a spiral trajectory with 5% missing measurements. 57
5.14 Estimated positions in a spiral trajectory with 50% missing measurements. 58
5.15 Estimated positions with the vehicles not traveling in formation, without missing measurements. 60
5.16 Estimated positions with the vehicles not traveling in formation, with 5% missing measurements. 61
5.17 Estimated positions with the vehicles not traveling in formation, with 50% missing measurements. 61
5.18 Estimated positions in a helix trajectory, without missing measurements. 62
5.19 Estimated positions in a helix trajectory, with 5% missing measurements. 63
5.20 Estimated positions in a helix trajectory, with 50% missing measurements. 63
5.21 Variation of MPE with distance standard deviation. 65
5.22 Variation of MPE with angle standard deviation. 65
5.23 Variation of MPE with missing data percentage. 66
5.24 MNE of MAP estimator and state-of-the-art method throughout the linear trajectory. 67
5.25 Empirical CDF of MAP estimator and state-of-the-art method throughout the linear trajectory. 67
5.26 Empirical CDF and hypothetical CDF of the error in the linear trajectory with 5% missing data. 68
5.27 MNE of MAP estimator and state-of-the-art method throughout the spiral trajectory. 69
5.28 Empirical CDF of MAP estimator and state-of-the-art method throughout the spiral trajectory. 69
5.29 Empirical CDF and hypothetical CDF of the error in the spiral trajectory with 5% missing data. 70
A.1 Actual states and state estimates of node 1. 78
A.2 Actual states and state estimates of node 2. 78
A.3 Actual states and state estimates of node 3. 78
A.4 Actual states and state estimates of node 4. 79
Notation

\( \mathcal{A} \) Set of anchors
\( \mathcal{E} \) Set of edges
\( \mathcal{G} \) Graph representation of a networks
\( \mathcal{V} \) Set of nodes
\( a \) Anchor position vector
\( d \) Noisy distance measurement between two nodes
\( q \) Unit vector representation of noisy bearing measurement between one node and one anchor
\( r \) Noisy distance measurement between one node and one anchor
\( u \) Unit vector representation of noisy bearing measurement between two nodes
\( x \) Node position vector

Greek symbols

\( \beta \) Position parameter
\( \epsilon \) Measurement Error
\( \kappa \) Concentration Parameter (of von Mises-Fisher distribution)
\( \mu \) Mean (parameter of Gaussian distribution)
\( \Omega \) Missing distance measurement between two nodes
\( \rho \) Distance parameter
\( \sigma \) Variance (parameter of Gaussian distribution)
\( \Upsilon \) Missing distance measurement between between one node and one anchor
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUV</td>
<td>Autonomous Underwater Vehicle</td>
</tr>
<tr>
<td>CDF</td>
<td>Cumulative Distribution Function</td>
</tr>
<tr>
<td>DFKF</td>
<td>Distributed Fusion Kalman Filter</td>
</tr>
<tr>
<td>EDM</td>
<td>Euclidean Distance Matrix</td>
</tr>
<tr>
<td>GNSS</td>
<td>Global Navigation Satellite System</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>IID</td>
<td>Independent and Identically Distributed</td>
</tr>
<tr>
<td>MAE</td>
<td>Mean Absolute Error</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum a Posteriori</td>
</tr>
<tr>
<td>MC</td>
<td>Monte Carlo</td>
</tr>
<tr>
<td>MDS</td>
<td>Multidimensional Scaling</td>
</tr>
<tr>
<td>MLE</td>
<td>Maximum Likelihood Estimator</td>
</tr>
<tr>
<td>MNE</td>
<td>Mean Navigation Error</td>
</tr>
<tr>
<td>MPE</td>
<td>Mean Positioning Error</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square Error</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Density Function</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root Mean Square Error</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
<tr>
<td>vMF</td>
<td>von Mises-Fisher</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation

Learning and estimation are some of the most exciting and promising technological fields of our time, offering countless possibilities. It is a tool that has seen great dissemination in the last few years and has therefore grown in applications, one of the wider goes towards real-world implementation, estimation under missing data. This thesis approaches an instance of estimation with incomplete data.

Our world grows increasingly connected. As the amount of data generated by communications technology increases, so too does the amount of missing data, defined as the data value that is not stored for a variable in the observation of interest. There can be a multitude of reasons why this occurs, including human errors during data entry, incorrect sensor readings, and software bugs. For example, when a person is asked to fill out a survey, unanswered questions are considered a source of missing data. In the case of a Wireless Sensor Network (WSN), loss of information packets occurs when communication fails between agents.

This work is focused on the topic of prediction under lost data, but it is applied to a very specific case, that of Autonomous Underwater Vehicles (AUVs) localization and the sensor measurements that get lost in transmission. Underwater communication is difficult due to the high conductivity of water that severely disrupts radio wave propagation, restricting the use of Global Navigation Satellite Systems (GNSS) to obtain the position of vehicles in this environment. Since there is no GNSS information available, it is necessary to find other ways to track the vehicle’s position. One of these methods, perhaps the most common one, is to use a swarm of vehicles that communicate between themselves and constantly feed each other information about their relative distances. It is then possible to use this information and, resorting to triangulation techniques, achieve a very good approximation of the position of each node (or vehicle, for the case at hand) in the grid.

Such a method is fairly reliable, and one might take measures to ensure further fidelity like increasing the number of nodes or resorting to better sensors. However, this is not always an option, and there will always be situations where data is lost in transmission. It is, therefore, of great importance to be able to harness the information of lost packets and to do so with the maximum precision possible. Furthermore,
methods of this type rely on statistical estimators, and seldom deal with missing data, further reinforcing the idea that there is the need for algorithms that can handle this kind of adversity.

This thesis aims to address that exact problem, and to do so resorting to estimation. In short, its objective is to create a model that can deal with a major data loss scenario that AUVs might suffer, and still be able to locate them accurately. It is worth mentioning that even though this method is developed here with a very specific task in mind it does not mean that it cannot find any uses besides its original application. Such an algorithm could prove very useful when dealing with any kind of swarm of autonomous vehicles, not just those that travel underwater. Flying and terrestrial vehicles could benefit from such an algorithm as well, for example, but we have chosen to focus on AUV because in this particular case missing data prediction is very relevant, given the nature of these vehicles and how often these episodes occur.

1.2 Objectives and Contributions

In this thesis, we propose a localization algorithm for GNSS-denied environments, under the presence of missing measurements. The aim is to develop a distributed and robust method directed at generic networks with noisy measurements of ranges and bearings between its nodes, capable of handling packet losses, without requiring initialization near the true position.

Under a convex framework, an approximate maximum likelihood estimator (MLE) is proposed, accounting for range and bearing measurements. We employed a convex relaxation developed in [1]. Afterward, from the maximum likelihood estimator, we develop a new maximum a posteriori (MAP) estimator to handle packet losses, accounting for previous positions and previous measurements of the missing packets. The aim is to present a new approach to the missing data problem in vehicle tracking, not only autonomous underwater vehicles, but also applicable to terrestrial and flying vehicles.

1.3 Thesis Outline

In the present Chapter 1, it is presented a brief introduction to the work developed in this thesis. After introducing the motivating reasons to develop a new algorithm, the objectives and contributions of this thesis are proposed.

In Chapter 2, it is presented an overview of the background of the problem in study. First, some state-of-the-art methods to handle missing measurements in state estimation are introduced. Then, some concepts regarding Wireless Sensor Networks and network localization are presented, along with the problem of underwater localization. Finally, since this work is based on estimation theory and statistical processes, some theoretical concepts regarding this topic are also provided, including some background on performance evaluation.

In Chapter 3, a convex formulation of the problem is presented. First, a maximum likelihood estimator is formulated suited for mobile networks, assuming that noisy measurements related to ranges and bearings are acquired. However, this estimator does not take into account missing measurements.
Therefore, a new maximum a posteriori Estimator is formulated, adding the previous values and positions of the missing variables to the maximum likelihood estimator.

In Chapter 4, an explicit solution to the optimization problem is presented. First, the problem is reformulated to be implemented, then the problem is mathematically analyzed and a method is developed based on some properties of the problem.

The results obtained for the problem in study are presented in Chapter 5, for three two-dimensional trajectories and one three-dimensional trajectory. The results are presented both with and without missing measurements. Then, the method is evaluated regarding noise influence, the percentage of missing packets and compared to a state-of-the-art-method.

Finally, Chapter 6 discusses the results, as well as possible areas for further study and development of this work.
Chapter 2

Background

Before deriving the proposed estimator, several concepts have to be understood. First, some state-of-the-art methods are presented, followed by an introduction to Wireless Sensor Networks and graph theory. Then, the estimation theory, which will be the basis for the work developed, will be summarized. Finally, some criteria regarding the performance evaluation algorithms will be described.

2.1 Missing Data Problem

Complex networks, whose structure is irregular, complex and dynamically evolving in time, are composed by a group of nodes interconnected under certain topological structures. The analysis of complex networks has drawn research interest due to its potential applications in real-world networks, such as computer, social and biological networks.

The state estimation problem for complex networks has recently received particular research interest, because, in reality, system states are not always available due to physical, technological or financial constraints.

The measurements available are employed to estimate the states of the nodes of the network. However, in a real environment and due to the complexity of large-scale networks, transmission delays and packet dropouts between nodes are inevitable due to several causes, such as accidental loss of collected data, sensors aging, sensor failures, and network congestion.

When designing a state estimator, it is important to take into account the missing data phenomenon, to improve the estimation performance. The state estimation problem with missing measurements has attracted research interest in the last few years and several state-of-the-art methods have been developed to deal with missing data in a network. This section provides a summary of these methods, describing their background, advantages, and shortcomings.

2.1.1 Distributed Fusion Kalman Filter

Reference [2] is concerned with the distributed Kalman filtering problem for a class of networked multisensor fusion systems with missing sensor measurements, random transmission delays, and packet
losses. The authors propose a stochastic model, described by multiple binary random variables with known probabilities, to model transmission delays and packet drop-outs. The authors consider that each sensor is bound to the measurement missing phenomenon, where a binary stochastic variable is used to model missing data on the sensor measurements. Subsequently, a local Kalman filter is applied to the output information of each sensor, which is transmitted to the fusion center of the communication network. In this process, the transmission delays and packet drop-outs are taken into account.

The objective of this method is to find a group of optimally weighted matrices, computed using the local error covariance, such that the mean square error (MSE) of the designed DFKF is optimal, that is, minimized, since the DFKF itself was determined using a group of optimally weighted matrices. It is aimed to prove that the MSE of the designed DFKF does not depend on the choice of the initial values. The results obtained here imply that the effect of the initial conditions is forgotten as more data is processed and this is important since the initial values are often poorly known or just set arbitrarily. The proposed method has advantages when compared with other methods. However, even though the authors claim that this is a distributed algorithm, all the information is gathered in the fusion center. This is undesirable since when the number of sensors increases, it has the potential for a critical failure point in the central node.

This method resembles our method in the employment of a MAP estimator since the Kalman filter is a MAP method. The prior probabilities used in the DFKF are the occurrence probabilities of delays, which are known a priori through statistical tests. Also, in the occurrence of packet dropouts, the prior position of the nodes is used.

2.1.2 Variance-constrained Approach to Recursive State Estimation

Reference [3] investigates the recursive state estimation problem is for an array of discrete time-varying coupled stochastic complex networks with missing measurements. The phenomenon of missing measurements is characterized by introducing a set of random variables satisfying certain probabilistic distributions, where each sensor can have an individual missing probability. These random variables are introduced in the output of each node of the network.

For the addressed time-varying complex networks, the measurement outputs may have missing data and measurement noises and it is difficult to obtain the ideal output. The purpose of the addressed state estimation problem is to design a time-varying state estimator such that, in the presence of the missing measurements and the random disturbances, an upper bound of the estimation error covariance can be guaranteed. The upper bound is minimized by properly designing an estimator parameter at each sampling instant. The estimator parameter is characterized by the solutions to two Riccati-like difference equations. For more information on Riccati equations see reference [4].

The exact value of the estimation error covariance is difficult to obtain due to the existence of unknown terms, linearization errors, and missing measurements. The authors employ an alternative way to find an upper bound of the estimation error covariance, using the Riccati-like difference equations.

As far as we know, this paper is one of the first attempts to investigate the recursive state estimation
problem for an array of discrete time-varying stochastic complex networks with missing measurements and random disturbances. The results obtained are not ideal, because the estimation error obtained is considerably high.

The state estimator resembles a Kalman filter, which is considered a MAP method, like the one we will use in our method. In each estimation, the previously estimated position of the nodes is employed.

2.1.3 Asynchronous Gradient-based Localization

The estimation of measurements in a multi-agent network, when in the presence of varying values of distance between the nodes, is a topic that is addressed in [5]. The approach taken in this work could be considered as a backward propagation of the values observed during a normal working period of the vehicle, to then reconstruct as closely as possible its past trajectory. To achieve this, the authors resort to GPS measurements and compasses, which provide, respectively, low precision measurements for absolute orientation and position, and high precision measurements for estimating the formation’s shape.

The work in question also aims to provide a solution that is distributed, asynchronous and robust to communication losses and random delays. A fast distributed and asynchronous linear least-squares algorithm is proposed to solve an approximated version of the non-linear Maximum Likelihood problem. If the range and bearing errors are sufficiently small, the localization problem is linearized, achieving a performance which is very close to the exact maximum likelihood solution.

Regarding the presence of packet losses and random delays, the algorithm proposed converges to the optimal solution. However, it must be noted that the higher the losses the slower the convergence. Although this approach is proven to be robust to packet losses and random delays, it does not deal directly with missing measurements or random delays that occur in a network. One of the drawbacks of this method is that, in a real set-up, different nodes could wake up and update their estimate at the same time, increasing the probability of communication collision.

This method formulates the localization problem as a maximum likelihood estimation problem using GPS position readings, range, and bearing measurements. This resembles our estimator since we will formulate an MLE using distance and angle measurements as well.

2.1.4 Robust Kalman Filter Design

Reference [6] is concerned with the robust Kalman filter design for linear uncertain systems subject to randomly varying delay and missing measurements.

When parameter uncertainties, which inevitably exist, are considered in the model of the addressed system, however, the performance of traditional Kalman and Extended Kalman filters can be severely degraded and sometimes even become unacceptable.

In the algorithm presented, Bernoulli random variables are introduced to account for the phenomena of random delay and missing measurements. Merging the stochastic uncertainties into the process noise and utilizing state augmentation, the original system is transformed into a stochastic parametrized
uncertain system. For the augmented system, a robust filter is proposed. Based on the discrete Riccati difference equation approach, an optimized upper bound on the filtering error variance is derived for all uncertainties, which is then minimized by determining suitable filtering parameters.

It is proven that this algorithm is reasonable since it is observed that actual estimation error variances stay below their upper bounds. Moreover, this filter has its advantages when compared with other filtering algorithms presented before, and thus has a satisfactory performance.

### 2.1.5 Event-Based Variance-Constrained $\mathcal{H}_\infty$ Filter

Reference [7] is concerned with the distributed $\mathcal{H}_\infty$ filtering problem [8] for a class of discrete time-varying stochastic parameter systems with error variance constraints over a sensor network where the sensor outputs are subject to successive missing measurements.

The phenomenon of the successive missing measurements for each sensor is modeled by a sequence of mutually independent random variables that obey the Bernoulli binary distribution law. To reduce the frequency of unnecessary data transmission and alleviate the communication burden, an event-triggered mechanism is introduced for the sensor node such that only some important data is transmitted to its neighboring sensors when specific events occur.

The objective of the problem addressed is to design a time-varying filter such that both the $\mathcal{H}_\infty$ requirements and the error variance constraints are guaranteed over a given finite-horizon against successive missing measurements and stochastic noises.

To avoid unnecessary data transmissions between the adjacent sensor nodes, an event-triggered communication mechanism is employed to determine whether the currently estimated states need to be delivered to its neighbors or not. This means that when there are missing measurements between neighbor sensors, the estimations do not change, and therefore a data transmission event is not triggered.

### 2.1.6 Matrix Completion

A simple approach to solving the missing measurement problem in the state estimation of a network is the employment of the Euclidean Distance Matrix Completion. To obtain the state estimate a Classical Multidimensional Scaling (MDS) algorithm can be employed. In [9] it is possible to find methods to obtain the state estimate through MDS.

A Euclidean Distance Matrix (EDM) is a symmetric, zero diagonal matrix, with non-negative elements, in which each entry is the square of the Euclidean distance between two vectors of some configuration of a network [10]. In a time horizon framework, for each sampling time, there is one respective EDM. When a measurement is missing, so is its respective entry of the EDM. Several methods to solve the EDM problem and to obtain the missing entries can be found in [10], [11] or in [12].
2.2 Wireless Sensor Network

Wireless sensor networks are collections of spatially distributed, compact-size, relatively inexpensive computational sensor nodes that measure local environmental conditions or other parameters, and cooperatively forward their data through the network to a central node. An example of the architecture of a WSN is shown in Figure 2.1. WSNs support a wide range of useful applications, from military surveillance to health care monitoring and water quality monitoring [13].

The nodes have various real-time applications, performing tasks like smart detection, neighbor node discovery, data collection, processing and storage, target tracking, monitoring and control, and also node localization. A critical resource constraint in WSNs is energy since they are powered by batteries. Low power consumption is a key factor in ensuring long operating horizons for non-power-fed systems.

![Figure 2.1: Architecture of a Wireless Sensor Network [14].](image)

A WSN is frequently represented as a graph $G(V, E)$, where the vertices $V$ correspond to sensor nodes and the edges $E$ to the distances (or angles) between them. To clarify this, some concepts of Graph Theory will be reviewed.

2.2.1 Graph Concepts

A graph $G$ consists of two sets, such that $V$ is a non-empty finite set of elements $V = \{v_1, ..., v_n\}$ designated as nodes (or vertices), and $E$ is a set of pairs of elements of $V$ designated as edges (or links) [15].

When two nodes are connected by an edge they are adjacent to each other and incident with the edge. Two edges with a common node are also called adjacent. Additionally, the degree of a node is defined by the number of its incident edges. This is a fundamental concept regarding WSN localization since it translates to the number of measurements each sensor has available.

A graph can be classified as either directed or undirected. The former has edges with direction, meaning that the order of the two nodes that form an edge is important. In this case, $(i, j)$ stands for a link from $i$ to $j$ and not the other way around. The latter graphs have edges without direction, and thus the order of the nodes that form an edge is not important [16].

From now on, only simple graphs will be considered, where each edge connects distinct node elements and there is at most one edge joining a given pair of vertices.
It is often useful to represent a graph using matrix notation. A graph with \( n \) nodes can be completely described by its adjacency (or connectivity) matrix \( A \), a \( n \times n \) square matrix whose entry \( a_{ij} (i, j = 1, ..., n) \) is equal to 1 when the edge \((i, j)\) exists, and zero otherwise. The diagonal of the adjacency matrix contains zeros. For undirected graphs, this matrix is symmetric. An alternative possibility is the incidence matrix \( M \), a \( n \times m \) matrix, considering \( m \) labeled edges, where element \( m_{ij} \) is equal to 1 if node \( i \) is incident with edge \( j \) and 0 otherwise. In the context of WSN localization, this is a possible representation of which sensors have available measurements between them.

2.2.2 Network Localization

The nodes of a WSN are classified as target, anchor and sensor nodes. Target nodes are defined as the nodes that do not provide their position and will not collaborate for the localization algorithm. Anchor nodes have a known position because they may be placed at known positions or have access to GNSS. Finally, sensor nodes are defined as the nodes that started with an unknown location and could estimate it through collaboration [17].

The Network Localization problem, also known as graph realization, can be roughly defined as finding the positions of unknown nodes, so that edge constraints between them are respected. It comprises a set of techniques and mechanisms that allow a sensor to estimate its location based on information gathered from the environment. While GPS is undoubtedly the most well-known location-sensing system, it is not accessible in all environments and may not be feasible in terms of costs for all WSNs.

There are two possible ways to express the measurements of a sensor, either in an absolute or a relative way. Resources like GPS help us determine the first type, by providing the absolute distance values inside a global reference frame. Relative metrics, on the other hand, rely on arbitrarily defined sets of axis, so by definition, they differ greatly from absolute values. Using this paradigm, for example, one could express the distance between a group of sensors using only the reference values between themselves, effectively eliminating the need for any outside positional cue. In the specific case of a WSN, it is likely infeasible for all nodes to be aware of every other agents’ position at every single instant in time, so some networks rely on a combination of these two types of networks. A common approach is to use a subset of global sensors, named anchors, which provide an absolute position reference to all other nodes on the network. Methods that are based on this sort of approach are called anchor-based localization techniques, contrarily to the ones who don’t, referred to as anchor-free methods. We could also classify networks based on their use of measurement estimations: when they use range measurements, meaning relative distances, they are called range-based localization techniques. It is worth mentioning, however, that these methods require sensors to monitor measurable characteristics, like for example the strength of the received signal or the difference in time between ultrasound pulses [13].

The position of a node can be obtained based on measured distances between itself and several anchor points with known positions. Given the location of an anchor and the distance from the node to the anchor, it is known that the node must be positioned somewhere along the circumference of a
circle centered at the position of the anchor with a radius equal to the sensor–anchor distance. This is illustrated in Figure 2.2. In two-dimensional space, distance measurements from at least three non-collinear anchors are required to obtain a unique location. This concept is illustrated in Figure 2.3. In general, unambiguous localization demands $n + 1$ anchors in a $n$–dimensional space.

Figure 2.2: The node is positioned somewhere along the circumference of a circle centered at the position of the anchor with a radius equal to the sensor–anchor distance.

Figure 2.3: In a 2D space, to obtain a unique location of a node, distance measurements from at least three non-collinear anchors are required.

**Underwater Localization**

Autonomous Underwater Vehicles (AUVs) are unmanned, self-propelled and most often torpedo-shaped vehicles. Commonly deployed from a surface vessel, they can operate independently of that vessel for periods of a few hours to several days [18]. An example of an AUV is shown in Figure 2.4. Multiple AUVs form an Underwater Network and are deployed to perform collaborative monitoring tasks over a given area, as it can be observed in the illustration in Figure 2.5.

Figure 2.4: The *Autosub6000*, an AUV used for under ice operations in the Arctic and Antarctic [18].

AUVs follow a preprogrammed course and can navigate using arrays of acoustic beacons and inertial navigation. Unlike submarine gliders, which have an undulating trajectory, AUVs can maintain a linear trajectory through the water and are therefore well suited to geoscience applications requiring constant
altitude. These vehicles operate at speeds from 0.5 m/s to 5 m/s, with most AUVs operating at a cruising speed of about 1.5 m/s [20].

The high conductivity of seawater severely disrupts radio wave propagation, hence it is not possible to rely on GNSS for localization. Therefore, to communicate, AUVs use acoustic modems that enable communication, thus allowing a cooperative navigation network. Each vehicle is assumed to have a modem providing noisy distance measurements between itself and other vehicles within a certain range. Moreover, AUVs can also be equipped with a vector sensor producing bearing measurements between vehicles.

**Missing Data Problem in Underwater Localization**

In an aquatic environment, acoustic communication is limited. Through the air, high-frequency radio waves transmit data at near light speeds, but underwater, communication uses acoustic waves instead of electromagnetic waves, which are less efficient. The waves can propagate at long distances through seawater at extra-low frequencies (30 – 300 Hz) and require large antennae and high transmission power [21].

The signal propagation is affected by many factors. Some of the main challenges of underwater communication are presented in this section, such as attenuation, uncertainties, multi-path propagation, and propagation speed.

Attenuation is the gradual loss of intensity as a signal moves through a medium, which is greater in a liquid when compared to air. It is caused by absorption due to the conversion of acoustic energy into heat.

The differences in temperature or density in the water may also cause the scattering, reverberation, refraction, and dispersion of the signals, further contributing to this loss.

Uncertainties have a major influence in underwater estimates. They can be caused by man-made noise, such as power plants, shipping activity, or ambient noise, such as currents, storms, wind, rain, and seismic activity.

Multi-path propagation may be responsible for severe degradation of the acoustic communication signal, since it generates Inter-Symbol Interference, a form of distortion of a signal in which one symbol interferes with subsequent symbols, making the communication less reliable.

The propagation speed in underwater channels is considerably low, which can cause propagation delays of the acoustic wave and affect the estimation accuracy. The delay of sound waves is five orders of magnitude longer than the delay of the radio-frequency signals, however, since radio waves are attenuated so strongly in salt water, their use in communication becomes impractical, except at very short ranges [22].

Most of the factors mentioned above are caused by the chemical and physical properties of the water medium such as temperature, salinity and density, and by spatiotemporal variations.

Besides, underwater sensors are prone to failure because of fouling and corrosion. Further difficulties arise with energy sources. Acoustic communication consumes more energy than radio communication, and batteries have limited power and are usually not easy to recharge. Lastly, communication can be
compromised by obstacles, such as underwater structures.

Considering all the mentioned factors, the probability of packet losses and random transmission delays occurring during the propagation of acoustic waves is considerably high. The motivation for our work is supported by the challenges in underwater communication and a method to handle missing sensor measurements and transmission delays are presented.

2.3 Estimation Theory Overview

Estimation theory is a branch of statistics that is concerned with the problem of parameter estimation. Many of the basic ideas of the theory of statistics are based on a collection of a data set of measured random variables \( x = \{X_1, \ldots, X_n\} \) with probability model \( f(x; \theta) \). The parameter \( \theta \) in the probability model is unknown, and the information in the random sample is used for estimating and testing claims about \( \theta \), or to define an estimator \( \hat{\theta} \). The formulas derived for estimating \( \theta \) are often based on transformations of the random variables in the data set [23].

Therefore, an estimator attempts to approximate the unknown parameters using the measured data. To determine good estimators, the first step is to mathematically model the data. Since the data is inherently random, it is described by its probability density function (PDF). The PDF is parametrized by the unknown parameter \( \theta \) and there exists a class of PDFs each one defined by a different value of \( \theta \) [24].

There are different methods to estimate the unknown parameter. This section is focused on two methods: the maximum likelihood estimator and the maximum a posteriori estimator.

2.3.1 Maximum Likelihood Estimator

The maximum likelihood estimator, which is based on the maximum likelihood principle, is the most popular approach to obtain practical estimators.

Maximum likelihood estimation is a method that determines values for the parameters of a model. The parameter values are found, such that they maximize the likelihood that the process described by the model produces the data observed.

The MLE for a scalar parameter is defined to be the value of \( \theta \) that maximizes the PDF \( p(x; \theta) \), for a fixed \( x \), where \( p(x; \theta) \) denotes the likelihood function. The logical basis for the MLE revolves around the observation that \( p(x; \theta) \) gives the probability of observing \( x \) for a given \( \theta \). The value of \( p(x = x_0; \theta) \) represents the probability of observing \( x = x_0 \) for a given value of \( \theta \). To clarify, it can be observed in Figure 2.6 that, if \( x = x_0 \) had indeed been observed, then inferring that \( \theta = \theta_1 \) would be unreasonable. Because if \( \theta = \theta_1 \), the probability of actually observing \( x = x_0 \) would be small. It is more likely that \( \theta = \theta_1 \) is the true value. Hence, \( \hat{\theta} = \theta_1 \) is the estimate chosen and is the value that maximizes \( p(x = x_0; \theta) \) over the allowable range of \( \theta \).

The objective is to calculate the total probability of observing all of the data, i.e. the joint probability distribution. To calculate the joint probability, it would be necessary to calculate conditional probabilities,
which can be difficult to obtain. To avoid this, it is assumed that each data point is generated independently of the others. If the processes that generate the data are independent, then the total probability of observing all of the data is the product of observing each data point individually. Therefore the likelihood function is given by

\[ L(\theta, x) = \prod_{i=1}^{n} p_i(x_i; \theta) \]  

(2.1)

where \( n \) is the total number of samples and \( p_i(x_i; \theta) \) is the PDF of the underlying distribution.

Figure 2.6: The PDF is evaluated for \( x = x_0 \) and plotted versus \( \theta \). The value of \( p(x = x_0; \theta) \) for each \( \theta \) gives the probability of observing \( x \) in the region centered around \( x_0 \), assuming the given value of \( \theta \). If \( \theta = \theta_2 \), the probability of observing \( x = x_0 \) would be small. It is more likely that \( \theta = \theta_1 \) is the true value.

It is usually more convenient to work with a logarithmic transformation of the likelihood function, known as the log-likelihood function \( \ln(L(\theta, x)) \). The natural logarithm is a monotonically increasing function, and the maximum value of the log-likelihood function occurs at the same point as the original probability function. Therefore, simpler log-likelihood can be used instead of the original likelihood.

Maximum-likelihood estimation has the advantages of being asymptotically efficient since it attains the minimum variance possible, meaning that the estimate gets better with more samples, it is easier to compute and the estimation is obtained without any prior information. However, in a real-world scenario, there is always some prior information about the parameter to be estimated, which is not taken into account. Another drawback is that for noisy data, noisy estimates will be given since the estimation closely agrees with data.

2.3.2 Maximum a Posteriori Estimator

The maximum a posteriori estimator is an approach used to obtain the estimate of an unknown quantity that equals the mode of the posterior distribution. If there is some prior knowledge available about the parameter to be estimated, that information can be incorporated into the estimator. In this approach, it is required to assume that \( \theta \) is a random variable with a given prior PDF.

Consequently, following the MAP estimation method the value of \( \theta \) that maximizes the posterior PDF \( p(\theta|x) \) is obtained. In finding the maximum of \( p(\theta|x) \), an equivalent expression given by the Bayes’
Bayes’ Theorem

In probability theory, Bayes’ theorem (also known as Bayes’ law or Bayes’ rule) describes the probability of an event, based on prior knowledge of conditions that might be related to the event.

The expression that describes Bayes’s theorem is stated as

\[ p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}, \]  

(2.2)

where \( p(\theta|x) \) is the likelihood of event \( \theta \) occurring given that \( x \) is true, \( p(x|\theta) \) is the likelihood of event \( x \) occurring given that \( \theta \) is true and \( P(\theta) \) and \( P(x) \) are the probabilities of observing \( \theta \) and \( x \) independently of each other.

As observed in (2.2), the maximization of \( p(\theta|x) \) is equivalent to the maximization of \( p(x|\theta)p(\theta) \). The value of \( p(x) \) does not depend on \( \theta \), and therefore the function to be maximized is not affected. Hence, the MAP estimator is

\[ \hat{\theta} = \arg \max_{\theta} p(x|\theta)p(\theta). \]  

(2.3)

Similarly to the MLE case, to avoid the differentiation of 2.3, the natural logarithm of the posterior distribution is maximized instead of the function itself.

The posterior distribution \( p(x|\theta) \) corresponds to the likelihood function \( p(x;\theta) \). Subsequently, the MAP estimate is reminiscent of the MLE with the presence of the prior PDF.

If the parameter assumes a uniform distribution, this means that all possible values of \( \theta \) have the same constant weight. Being constant, it can be ignored from the MAP equation, as it will not contribute to the maximization. Therefore, the MLE can be a special case of MAP estimate.

Even though the MAP estimate includes the prior information of the parameters, it has some limitations as well. The most obvious drawback is that it does not provide any measure of uncertainty since modeling errors can be incorporated in the prior information [25].

2.4 Performance Evaluation

Evaluating the performance of an algorithm is important, either when validating a new method against state-of-the-art techniques or when choosing an existing algorithm for a WSN application with certain requirements. Various criteria may be employed focusing on different issues such as scalability, accuracy, resilience to noise, geometric characteristics or cost. Different applications will have different needs and deciding which performance criteria to use is important for the success of the resulting implementation [26].

The intuitive measure of performance of a localization algorithm may be to compare the estimated position of the nodes with the known ground truth. However, localization algorithms are also subjected to other constraints and a broader set of evaluation criteria are used, such as accuracy, cost, robustness,
and scalability. These criteria reflect computational limitations, power constraints, unit cost, and network scalability. Some performance criteria and evaluation metrics are presented in this section.

2.4.1 Scalability

In the context of localization, the property of scalability described how a WSN handles a growing amount of nodes. Two types of algorithms can be distinguished, centralized and distributed. In a centralized approach one central node receives necessary measurements from all nodes and computes their position, which is at the end sent back to each of them. By contrast, distributed techniques allow for each node to locally compute its position, merely relying on information from neighbor nodes and anchors.

Centralized approaches can make use of all information in the network, but are susceptible to failure of an entire network if the central processing unit fails. They are also more prone to congestion near to the central node when the number of nodes in the network increases. For the specific case of mobile large networks, a distributed algorithm is sometimes preferred, as the time it takes for all nodes to send their information to be centrally processed and receive the position estimate can be considerable, and highly dependent on the current network topology.

2.4.2 Accuracy

Localization accuracy is a measure of how well the estimated positions relate to the true ones. This is an important consideration since it relates to the primary objective of any localization algorithm. Accuracy of an algorithm is most likely connected to the measurement noise, bias, accuracy, and precision of input data.

Mean Absolute Error

The simplest way to describe the localization performance is to determine the Mean Absolute Error (MAE), where the error between the true and estimated position of each node is computed and averaged amongst all nodes in the network. The MAE is determined by

\[
MAE(\hat{x}) = \frac{1}{N} \sum_{n=1}^{N} ||\hat{x}_n - x_n||_2,
\]

(2.4)

where \(\hat{x}\) is an estimate of \(x\) and \(N\) is the number of nodes in the network.

Mean Navigation Error

The definition of MAE can be extended to a trajectory, where the positions are estimated over time. It is possible to conduct \(M\) Monte Carlo (MC) trials, which will be explained later, and obtain the Mean Navigation Error (MNE) in a specific time instant \(t\), defined by

\[
MNE(\hat{x}, t) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{n=1}^{N} ||\hat{x}_n^i(t) - x_n^i(t)||_2.
\]

(2.5)
Mean Positioning Error

It is possible to summarize the MNE for a certain time interval. The Mean Positioning Error is defined by

\[
MPE(\hat{x}) = \frac{1}{MNT} \sum_{i=1}^{M} \sum_{n=1}^{N} \sum_{t=1}^{T} ||\hat{x}_{ni}^{n}(t) - x_{ni}^{n}(t)||^2
\]

(2.6)

where \( T \) defines the number of steps in a time interval.

Root Mean Square Error

The root-mean-square error (RMSE) is a measure of accuracy which represents the quadratic mean of the differences between values predicted by an estimator and the values observed. It corresponds to the standard deviation of the predicted values. The RMSE is determined by

\[
RMSE(\hat{x}) = \sqrt{\frac{1}{MNT} \sum_{i=1}^{M} \sum_{n=1}^{N} \sum_{t=1}^{T} (\hat{x}_{ni}^{n}(t) - x_{ni}^{n}(t))^2}.
\]

(2.7)

Monte Carlo Trials

Monte Carlo simulations rely on repeated random sampling to obtain numerical results that depend on random variables, to model the probability of different outcomes [27].

When faced with significant uncertainty in the process of making an estimation, rather than just replacing the uncertain variable with a single average number, the Monte Carlo Simulation might prove to be a better solution because it relies on repeated random sampling.

The MC trials frequently follow the steps presented below [28].

1. Determine the statistical properties of possible inputs;
2. Generate, repeatedly, sets of possible inputs which follow these properties;
3. Perform a deterministic calculation with these sets;
4. Analyse statistically the results.

Roughly speaking, the main idea of MC simulations is to repeat an experiment many times to obtain several quantities of interest and observe the possible outcomes of these experiments.

The advantage of a Monte Carlo simulation is that it can indicate the likelihood of the resulting outcomes. It is essential for the simulation of real-life random systems, to understand the behavior of those systems.

2.4.3 Resilience to Noise and Errors

The resilience to noise and error measures how well a localization algorithm will perform without an accurate or a full set of input data. This evaluation criterion indicates how measurement noise and bias in the input data affects the performance of the algorithm, and also indicates the number of nodes that can be localized due to the occurrence of missing measurements.
2.4.4 Implementation Costs

Cost metrics refer to how expensive a given method is to implement, either in terms of money, time or required technology. The costs can be measured in terms of power consumption required to obtain the location of the nodes, time taken to localize a node, amount of communication (number of transmitted packets) required, and number of anchors needed since anchors usually require more expensive sensors to obtain an accurate position. Sometimes, a less accurate algorithm may be preferred if it is less expensive to conduct.
Chapter 3

Estimation with Missing Measurements

This chapter aims to formulate the model of the estimator proposed in this work. After stating the problem, the maximum likelihood estimator is proposed together with the statistical models of the distance and angle measurements, followed by the derivation of the maximum a posteriori estimator. The MAP estimator handles missing data with the introduction of statistical models for the missing measurements based on their previous values.

3.1 Problem Formulation

This work presents an algorithm to estimate the localization of a network with a variable number of AUVs (nodes) under the presence of missing measurements. The anchors are assumed to be vehicles with access to an accurate positioning system.

Each vehicle is assumed to have a modem providing noisy distance measurements between itself and other vehicles within a certain range. Each vehicle might also be equipped with a vector sensor producing bearing measurements between vehicles. To obtain a good performance, it is sufficient to have approximately 50% of nodes with bearing capability [1]. This angular information is assumed to be already in a common frame of reference, which could be obtained resorting to compass measurements, so that all angles are expressed with respect to the North direction, for instance.

It is further considered that if node 1, for example, has measurements concerning node 2, then the reverse is also true. All vehicles are mobile and can follow any trajectory, including anchors.

3.1.1 Problem Statement and Notation

The aim of this work is to determine the position of each vehicle of a network in the presence of missing sensor data, during a certain time interval. In order to achieve our goal, we first define our network as an undirected connected graph $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}(t))$, where each different graph corresponds to a given time.
instant. The vertices of the graph \( V(t) \subseteq \{1, \ldots, n\} \) correspond to the vehicles (nodes) and the edges \( E(t) = \bigcup_v E^v(t) \) correspond to the measurement of type \( v \) available between node \( i \) and node \( j \), at time \( t \).

The anchors, which are the known sensor positions, are defined by \( A(t) \subseteq \{1, \ldots, m\} \). Moreover, the subset \( A^v_i(t) \subseteq A(t) \) contains the anchors relative to which node \( i \) has an available measurement of type \( v \) at time \( t \). In particular, \( A^v_i(t) = \{ k \in A(t) : \exists r_{ik}(t) \} \) refers to distance measurements and \( A^\theta_i(t) = \{ k \in A(t) : \exists \theta_{ik}(t) \} \) to angle measurements.

In this work, we consider that each node has measurements available with respect to all its neighbors within a certain distance. At time \( t \), the noisy distance measurement between nodes \( i \) and \( j \) is represented by \( d_{ij}(t) = d_{ji}(t) \) and the noisy distance measurements between node \( i \) and anchor \( k \) is represented by \( r_{ik}(t) = r_{ki}(t) \). We employ a Noisy Disk Model [26], stating that a distance measurement \( d_{ij} \) between nodes \( i \) and \( j \) is only available if both nodes are separated by less than certain distance \( d_{\text{max}} \).

The noisy angle measurements \( \theta_{ij}(t) \) between nodes \( i \) and \( j \), at time \( t \), is represented as a unit-norm vector in the correspondent direction \( u_{ij}(t) \). Similarly, the noisy angle measurement \( \gamma_{ik}(t) \) between node \( i \) and anchor \( k \) is given by the unit-norm vector \( q_{ik}(t) \). We will consider two types of edges, \( E^d(t) = \{ i \sim j : 3d_{ij}(t) \} \) which corresponds to distance measurements and \( E^\theta(t) = \{ i \sim j : 3u_{ij}(t) \} \) which corresponds to angle measurements.

The position of sensor \( i \) at time \( t \) is defined by \( x_i(t) \in \mathbb{R}^p \) and the position of anchor \( k \) is defined by \( a_k(t) \in \mathbb{R}^p \). \( \mathbb{R}^p \) represents the Euclidean space of dimension \( p \) where the estimation algorithm is developed. In this work we consider the case where \( p = 2 \) (2D) and \( p = 3 \) (3D).

At the current time instant \( t \), the distances observed between nodes are \( d_o(t) = \{ d_{ij}(t) : i \sim j \in E(t) \} \). However, due to data packet losses or transmission delays there might be unknown distance measurements. These unknown distances between nodes \( i \) and \( j \) are defined as \( d_u(t) = \{ d_{ij}(t) : i \sim j \notin E(t) \} \). Moreover, the edges that correspond to missing measurements are given by \( E_u(t) = \{ i \sim j : i \sim j \notin E(t) \} \). When the missing distance measurements occur between one node and one anchor, node \( i \) has a distance measurement relative to anchor \( k \) at time \( t-1 \) available. There is, however, no measurement available at time \( t \). The subset containing these anchors is defined by \( A^v_i(t) = \{ k : k \in A_i(t-1), k \notin A_i(t) \} \). Furthermore, regardless of the type of measurement that is lost, the subset \( V_u(t) \subseteq V(t) \) contains all the nodes that are related to each missing measurement.

It is assumed that when the packet is lost at time \( t \), both the distance measurement and the angle measurement relative to that packet are considered missing data. It is also assumed that when a random delay occurs in the transmission, the respective information packet is treated as missing data.

The problem is to estimate the unknown sensor positions and as nuisance parameters, the random missing measurements. The node position \( x(t) = \{ x_i(t) : i \in V(t) \} \) is estimated given measurements \( \{ d_{ij}(t) : i \sim j \in E^d(t) \} \cup \{ r_{ik}(t) : k \in A^v_i(t), i \in V(t) \} \cup \{ u_{ij}(t) : i \sim j \in E^\theta(t) \} \cup \{ q_{ik}(t) : k \in A^\theta_i(t), i \in V(t) \} \). The random missing measurements are estimated from the previous measurement and node position estimate, given by \( \{ x_i(t-1) : i \in V_u(t) \} \cup \{ d_{ij}(t-1) : i \sim j \in E_u(t) \} \) when a packet between
two nodes is lost or \( \{ x_i(t-1) : i \in V_u(t) \} \cup \{ r_{ik}(t-1) : k \in A^u_v(t), i \in V(t) \} \) when a packet between one anchor and a node is lost.

### 3.2 Maximum Likelihood Estimator for Static Network Localization with Complete Data

In this section, we present the first part of the model, which does not take into account any missing measurements, developed in [29] and [1].

The MLE formulation, which considers ranges and bearing measurements, is concerned with the localization of the network in a scenario where there are no packet losses and delays. The problem is formulated as an optimization problem and an existing relaxation is presented.

For the sake of convenience, the time dependence notation will not be used in this section. The notation will be reintroduced later when the dynamic formulation is presented.

First, it is necessary to model the relation between the obtained measurements and the actual features of the network.

#### 3.2.1 Data Model for Ranges

The distance between node \( i \) and node \( j \) is a noisy measurement given by

\[
    d_{ij} = ||x_i - x_j|| + \epsilon_{ij}^d,
\]

where \( ||x_i - x_j|| \) is the true Euclidean distance between node \( i \) and node \( j \) and \( \epsilon_{ij}^d \) is the error associated with such measurement. The distance between two nodes is known by both nodes, therefore \( d_{ij} = d_{ji} \).

The physical distance that separates two nodes is obtained based on signal propagation time, which in this case is Acoustic Time of Arrival.

The Euclidean norm \( ||x|| \) is defined as

\[
    ||x|| = \sqrt{x \cdot x} = \sqrt{\sum_{i=1}^{n} x_i^2}.
\]

For underwater ranges, the noise is commonly assumed to be Gaussian white noise [30]. Hence, the error can be modeled with a Gaussian distribution of zero mean \( \mu \) and fixed variance \( \sigma^2 \). Therefore, it is assumed that \( \epsilon_{ij}^d \) is normally distributed with zero mean and variance \( \sigma^2_{ij} \), that is

\[
    \epsilon_{ij}^d \sim \mathcal{N}(0, \sigma^2_{ij}).
\]

The distance measurement \( d_{ij} \) is a linear transformation of \( \epsilon_{ij}^d \) with mean \( ||x_i - x_j|| \) and variance \( \sigma^2_{ij} \), that is

\[
    d_{ij} \sim \mathcal{N}(||x_i - x_j||, \sigma^2_{ij}).
\]
The probability function of the Gaussian distribution with mean $||x_i - x_j||$ and variance $\sigma_{ij}^2$ is defined as

$$p(d_{ij}; \sigma_{ij}) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} \exp\left(-\frac{(||x_i - x_j|| - d_{ij})^2}{2\sigma_{ij}^2}\right).$$  \hfill (3.5)

Similarly, the distance measurement between node $i$ and anchor $k$ is defined as

$$r_{ik} = ||x_i - a_k|| + \epsilon_{ik}^r.$$  \hfill (3.6)

The respective distributions for the error $\epsilon_{ik}^r$ and the measurement $r_{ik}$ are given by

$$\epsilon_{ik}^r \sim \mathcal{N}(0, \varsigma_{ik}^2)$$  \hfill (3.7)

and

$$r_{ik} \sim \mathcal{N}(||x_i - a_k||, \varsigma_{ik}^2).$$  \hfill (3.8)

Finally, the probability function is

$$p(r_{ik}; \varsigma_{ik}) = \frac{1}{\sqrt{2\pi\varsigma_{ik}^2}} \exp\left(-\frac{(||x_i - a_k|| - r_{ik})^2}{2\varsigma_{ik}^2}\right).$$  \hfill (3.9)

### 3.2.2 Data Model for Bearings

The bearing between node $i$ and node $j$ is a noisy measurement of $\angle(x_i, x_j)$, modeled as

$$\theta_{ij} = \angle(x_i, x_j) + \epsilon_{ij}^\theta.$$  \hfill (3.10)

Similarly, bearing measurements between nodes and anchors are given by

$$\gamma_{ik} = \angle(x_i, a_k) + \epsilon_{ik}^\gamma.$$  \hfill (3.11)

Unlike distances, bearings refer to circular data, which requires special consideration. Specific statistics for directional data are required and the bearing errors will be modeled with the analog of the Gaussian distribution for this kind of data: the von Mises-Fisher (vMF) distribution.

### Von Mises-Fisher Distribution

The probability density function of the von Mises–Fisher distribution for the random $p$–dimensional unit vector $x$, is given by:

$$f(x; \mu, \kappa) = C_p(\kappa) \exp(\kappa \mu^T x)$$  \hfill (3.12)

where $\mu$ is the mean direction (with $||\mu|| = 1$) and $\kappa \geq 0$ is the concentration parameter. $C_p(\kappa)$ is a constant dependent on $\kappa$ and dimension $p$, given as

$$C_p(\kappa) = \frac{\kappa^{p/2-1}}{(2\pi)^{p/2} I_{p/2-1}(\kappa)}$$  \hfill (3.13)
where $I_s(\kappa)$ denotes the modified Bessel function of the first kind with order $s$.

Parameter $\mu$ is the mean direction of vector $x$, whereas $\kappa$ is a measure of the distribution concentration around $\mu$. For $\kappa = 0$, the distribution is uniform and, as it increases, the variables are more concentrated around the mean direction. The vMF parameters $\mu$ and $1/\kappa$ are analogous to $\mu$ and $\sigma^2$ (mean and variance) in the normal distribution.

Angular measurements are from now on considered as their correspondent unit-vectors. Noisy angle measurements $\theta_{ij}$ between two nodes are represented as a unit-norm vector in the correspondent direction $u_{ij}$. Similarly, noisy angle measurements $\gamma_{ik}$ between one node and one anchor are taken as the unit-norm vector $q_{ik}$. Their respective distributions are

$$u_{ij} \sim vMF \left( \frac{x_i - x_j}{||x_i - x_j||}, \kappa_{ij} \right)$$

and

$$q_{ik} \sim vMF \left( \frac{x_i - a_k}{||x_i - a_k||}, \lambda_{ik} \right)$$

where $\frac{x_i - x_j}{||x_i - x_j||}$ and $\frac{x_i - a_k}{||x_i - a_k||}$ are their mean directions, and $\kappa_{ij}$ and $\lambda_{ik}$ are the concentration parameters.

The correspondent probability functions are, then, given as

$$p(u_{ij}; \kappa_{ij}) = C_p(\kappa_{ij}) \exp \left( \kappa_{ij} u_{ij}^T \frac{x_i - x_j}{||x_i - x_j||} \right)$$

and

$$p(q_{ik}; \lambda_{ik}) = C_p(\lambda_{ik}) \exp \left( \lambda_{ik} q_{ik}^T \frac{x_i - a_k}{||x_i - a_k||} \right).$$

### 3.2.3 Likelihood Function

When the probability density function of $x$ is viewed as a function of the unknown parameter $\theta$, with fixed $x$, it is termed the likelihood function and is represented by $p(x; \theta)$. [24]

Two or more random variables are said to be independent and identically distributed (IID) if they are mutually independent and each random variable has the same probability distribution as the others.

Under the assumption that measurement noises are independent and identically distributed across time and sensors, the likelihood function for $p(d_{ij}, r_{ik}, u_{ij}, q_{ik}; x_i)$ is given as

$$L(d_{ij}, r_{ik}, u_{ij}, q_{ik}; x_i) = \prod_{i \sim j \in E^d} p(d_{ij}; x_i) \prod_{i \in V} \prod_{k \in A^j} p(r_{ik}; x_i) \prod_{i \sim j \in E^a} p(u_{ij}; x_i) \prod_{i \in V} \prod_{k \in A^a} p(q_{ik}; x_i)$$

As mentioned in chapter 2, it is often convenient to work with the natural logarithm of the likelihood function, called the log-likelihood, because it is easier to handle. The logarithm is a strictly increasing function, so the logarithm of a function achieves its maximum at the same point as the function itself, so it is equivalent to maximize either one.
The log-likelihood function is

$$
\ln \mathcal{L}(d, r, u, q; x) = \sum_{i \sim j \in E} \ln p(d_{ij}; x_i) + \sum_{i \in V} \sum_{k \in A^v} \ln p(r_{ik}; x_i) + \sum_{i \sim j \in E} \ln p(u_{ij}; x_i) + \sum_{i \in V} \sum_{k \in A^v} \ln p(q_{ik}; x_i),
$$

(3.19)

with

$$
\ln p(d_{ij}; x_i) = Z_d - \frac{1}{2 \sigma_{ij}^2} (||x_i - x_j|| - d_{ij})^2,
$$

(3.20)

$$
\ln p(r_{ik}; x_i) = Z_r - \frac{1}{2 \varsigma_{ik}^2} (||x_i - a_k|| - r_{ik})^2,
$$

(3.21)

$$
\ln p(u_{ij}; x_i) = Z_u + \kappa_{ij} u_{ij}^T \frac{x_i - x_j}{||x_i - x_j||},
$$

(3.22)

$$
\ln p(q_{ik}; x_i) = Z_q + \lambda_{ik} q_{ik}^T \frac{x_i - a_k}{||x_i - a_k||},
$$

(3.23)

where $Z_d$, $Z_r$, $Z_u$ and $Z_q$ are independent of the parameters $x_i$.

### 3.2.4 Optimization Problem

#### Background on Convex Optimization

A mathematical optimization problem has the form [10]

$$
\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq b_i, \quad i = 1, \ldots, m.
\end{align*}
$$

(3.24)

where $x = (x_1, \ldots, x_n)$ is the optimization variable, $f_0 : \mathbb{R} \to \mathbb{R}$ is the objective function, also called cost function, and the functions $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$ are the constraint functions limited by the constants $b_1, \ldots, b_m$.

A vector $x^*$ is called optimal, or a solution to the problem if, of all vectors respecting the constraints in (3.24), it attains the minimal value for $f_0(x)$.

A convex optimization problem takes the form of Problem (3.24) but the functions $f_0, f_1, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}$ are convex.

Now, we will clarify the notion of convexity, first regarding sets and then functions.

A set $C$ is convex if the line segment between any two points in $C$ lies in $C$, i.e., if for any $x_1, x_2 \in C$ and any $\theta$ with $0 \leq \theta \leq 1$, we have

$$
\theta x_1 + (1 - \theta)x_2 \in C.
$$

(3.25)

A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if dom$f$ is a convex set and $\forall x, y \in \text{dom}f$ and $\theta$ with $0 \leq \theta \leq 1$ we have

$$
f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y).
$$

(3.26)
Geometrically, this inequality means that the line segment between \((x, f(x))\) and \((y, f(y))\) lies above the graph of \(f\). This is visualized in Figure 3.1.

Figure 3.1: Graph of a convex function. The line segment between any two points on the graph lies above the graph [10].

**Formulation of the Optimization Problem**

When there is no missing data, the MLE is defined as the values of \(x_i, d_{ij}\) and \(r_{ik}\) that maximize the likelihood function. By maximizing (3.19) over \(x, d\) and \(r\), the following unconstrained optimization problem is obtained

\[
\begin{align*}
\text{maximize} \quad & - \sum_{i \sim j \in \mathcal{E}_a} \frac{1}{2\sigma_{ij}^2} (||x_i - x_j|| - d_{ij})^2 - \sum_{i \in V} \sum_{k \in \mathcal{A}_i^a} \frac{1}{2\varsigma_{ik}} (||x_i - a_k|| - r_{ik})^2 \\
& + \sum_{i \sim j \in \mathcal{E}_a} \kappa_{ij} u_j^T \frac{x_i - x_j}{||x_i - x_j||} + \sum_{i \in V} \sum_{k \in \mathcal{A}_i} \lambda_{ik} q_k^T \frac{x_i - a_k}{||x_i - a_k||}.
\end{align*}
\]

(3.27)

The problem in (3.27) is clearly equivalent to the formulation

\[
\begin{align*}
\text{minimize} \quad & \sum_{i \sim j \in \mathcal{E}_a} \frac{1}{2\sigma_{ij}^2} (||x_i - x_j|| - d_{ij})^2 + \sum_{i \in V} \sum_{k \in \mathcal{A}_i^a} \frac{1}{2\varsigma_{ik}} (||x_i - a_k|| - r_{ik})^2 \\
& - \sum_{i \sim j \in \mathcal{E}_a} \kappa_{ij} u_j^T \frac{x_i - x_j}{||x_i - x_j||} - \sum_{i \in V} \sum_{k \in \mathcal{A}_i} \lambda_{ik} q_k^T \frac{x_i - a_k}{||x_i - a_k||}.
\end{align*}
\]

(3.28)

For completeness, we now summarize the analysis from [29] and [1].

**Convexity Analysis**

The optimization problem in (3.28) is non-convex, both in the distance terms and in the angle terms. An analysis of both cases will demonstrate this.

First, regarding the distance terms, let us consider a function

\[
f(x, d) = (||x|| - d)^2
\]

(3.29)

with two variables, \(x\) and \(d\). This is equivalent to the distance terms in (3.28), considering an anchor or a node situated at the origin.

The convexity of function (3.29) will be analyzed for each variable separately.

By definition, a function is convex if the line segment between any two points of the function lies above it. [24] The graph of \(f\) is represented in Figure 3.2, for both cases, when \(x\) is the variable and \(d\) is
fixed and when \( d \) is the variable and \( x \) is fixed.

![Graph](image1)

Figure 3.2: Representation of \((|x| - d)^2\). On the left, for fixed \( x = 4 \), the function \((4 - d)^2\) is convex, because a line segment between any two points of the graph will lie above the graph. On the right, for fixed \( d = 4 \), the function \((|x| - 4)^2\) is non-convex, because a line segment between any two points of the graph may not lie above the graph.

The function \( f(d) \) for fixed \( x \) is convex, yet, the function \( f(x) \) for fixed \( x \) is non-convex. Due to the joint convexity property, \( f(x, d) \) is non-convex.

In the angle terms we have one variable, \( x \), and as an equivalent of the angle terms we consider the function

\[
f(x) = \frac{x}{|x|}
\]

(3.30)

The function \( f(x) \) is obviously non-convex. The graph of the function is represented in Figure 3.3.

![Graph](image2)

Figure 3.3: Representation of \( f(x) = \frac{x}{|x|} \). The line segment between any two points of the graph does not lie above the graph.

To deal with non-convex terms, a common approximation is to employ convex relaxations and turn the problem into a convex one. In this present section we explain the relaxations employed to the distance terms and to the angle terms.

**Relaxation of distance terms**

Here we will present a known relaxation from [1] and [29], that will enable the treatment of missing data.
The positive part of a function is defined by

\[ f_+ = \max\{0, f\} = \begin{cases} f, & \text{if } f > 0 \\ 0, & \text{otherwise}. \end{cases} \] (3.31)

This definition can be applied to the distance terms in order to obtain convexity. The distance term between two nodes, becomes

\[ (||x_i - x_j|| - d_{ij})_+ = \max\{0, ||x_i - x_j|| - d_{ij}\}. \] (3.32)

To better understand the relaxation applied to this term, the graphs of the function in (3.29) before and after truncation are represented in Figure 3.4. While the original function has two minima for \( x = -4 \) and \( x = 4 \), the relaxation allows for solutions in the interval \([-4, 4]\). This illustrates both the advantages and drawbacks of relaxations. On one hand, the value of \( x \) minimizing the cost function is not necessarily the same, with any value in the interval \([-4, 4]\) producing the same value as the intended solution \( x = \pm 4 \). On the other hand, it is possible to employ convex methods to find the global minimum.

![Figure 3.4: Representation of the original function \( f(x) \) and its relaxation \( f(x)_+ \). The difference is observed in the interval \([-4, 4]\) where the non-convexity is removed. The minimum is now verified for that entire interval, instead of the previous solutions occurring at \( x = -4 \) and \( x = 4 \) only.](image)

The corresponding term of the optimization problem in (3.32) is defined by

\[ (||x_i - x_j|| - d_{ij})^2_+ = \begin{cases} (||x_i - x_j|| - d_{ij})^2, & \text{if } ||x_i - x_j|| - d_{ij} > 0 \\ 0, & \text{otherwise}. \end{cases} \] (3.33)

The relaxed term in (3.32) translates to finding the values of \( x_i \) and \( x_j \) with distance \( ||x_i - x_j|| \) as close as possible to the measured \( d_{ij} \). However, an equivalent formulation is to minimize the distance of \( ||x_i - x_j|| \) to a set of points \( y_{ij} \), with norm \( d_{ij} \).

The equivalent formulation is then

\[ (||x_i - x_j|| - d_{ij})^2_+ = \inf_{||y_{ij}|| \leq d_{ij}} ||x_i - x_j - y_{ij}||^2 \] (3.34)
where the constraint $||y_{ij}|| \leq d_{ij}$ corresponds to the norm ball defined as $\{x_i : ||x_i - x_j|| \leq d_{ij}\}$.

Norm balls are convex sets, but their surface alone is not. Before the relaxation, the optimization of the non-convex distance term was equivalent to minimize the distance of $x$ to a set of points $y$, with norm $d$, with the constraint $||y|| = d$, which corresponds to the surface of a norm ball. After the relaxation, the constraint is the norm ball and not only its surface. An illustrative example of this formulation is depicted in Figure 3.5, where the role of variable $y$ is evident. The aim is to minimize the distance between the variables $y$ and $x$. The solution is the same in both cases, showing the effectiveness of the relaxation employed.

![Figure 3.5](image.png)

Figure 3.5: Representation of the minimization of distance terms. On the figure on the left, variable $y$ is restricted to the circle of radius $d$, while on the figure on the right, after the relaxation, $y$ is restricted to the norm ball of radius $d$.

The same relaxation is applied to the distance term between one node and one anchor. The truncated expression and the equivalent formulation have the form

$$((||x_i - a_k|| - r_{ik})^2 = \inf_{||w_{ik}|| \leq r_{ik}} ||x_i - a_k - w_{ik}||^2) \quad (3.35)$$

where the constraint $||w_{ik}|| \leq r_{ik}$ corresponds to the norm ball defined as $\{x_i : ||x_i - a_k|| \leq r_{ik}\}$.

The optimization problem with only range measurements with the convex relaxation is given by

$$\min_{x,y,w} \sum_{i,j \in E} \frac{1}{2}\sigma_{ij}^2 ||x_i - x_j - y_{ij}||^2 + \sum_{i \in V} \sum_{k \in A} \frac{1}{2}\varsigma_{ik}^2 ||x_i - a_k - w_{ik}||^2$$

subject to $||y_{ij}|| \leq d_{ij}$, $||w_{ik}|| \leq r_{ik}$. \quad (3.36)

Relaxation of angle terms

From the relaxation of the distance terms, the optimal case generates $x_i - x_j = y_{ij}$ and $||x_i - x_j|| = ||y_{ij}|| = d_{ij}$.

The relaxation of the angle terms with the optimal values from the distance terms minimization is proposed, using the optimal terms of $x$ and $y$ to approximate the bearing terms. Applying these results to the bearing term between two nodes, $x_i - x_j$ is approximated by $y_{ij}$ and $||x_i - x_j||$ is approximated...
by $d_{ij}$. In Figure 3.5 it is shown that variable $y$ is now being forced to the direction given by vector $u$.

The bearing term between two nodes in the optimization problem becomes

$$\kappa_{ij} y_{ij}^T \frac{x_i - x_j}{||x_i - x_j||} \approx \frac{\kappa_{ij} y_{ij}^T d_{ij}}{d_{ij}}.$$  \hspace{1cm} (3.37)

Similarly, the bearing term between one node and one anchor in the optimization problem becomes

$$\lambda_{ik} q_{ik}^T \frac{x_i - a_k}{||x_i - a_k||} \approx \frac{\lambda_{ik} q_{ik}^T w_{ik}}{r_{iw}}.$$  \hspace{1cm} (3.38)

The final expression for this convex approximation with ranges and bearings terms is

$$\minimize_{x,y,w} \left( \sum_{i \sim j \in E} \frac{1}{2\sigma_{ij}^2} ||x_i - x_j - y_{ij}||^2 + \sum_{i \in V} \sum_{k \in A_i} \frac{1}{2\sigma_{ik}^2} ||x_i - a_k - w_{ik}||^2 \right) - \sum_{i \sim j \in E} \kappa_{ij} y_{ij}^T d_{ij} - \sum_{i \in V} \sum_{k \in A_i} \lambda_{ik} q_{ik}^T w_{ik}$$  \hspace{1cm} (3.39)

subject to $||y_{ij}|| \leq d_{ij}$, $||w_{ik}|| \leq r_{ik}$.

### 3.2.5 Dynamic Formulation

We further develop the formulation to be suitable for mobile networks, as a network of AUVs, for instance. When vehicles follow a trajectory, it is beneficial to take into account different consecutive positions to obtain a better estimate. Each position at time $t$ relates to the previous and next one through a dynamic model. The dynamic model is essential to describe the behavior of the system over time.

#### Formulation with time window

It is advantageous to take into account the successive different positions in time during the trajectory of a vehicle to obtain a more accurate estimate. Introducing a time window will correspond to minimizing the cost function over consecutive time instants.
By considering a time window of size $T_0 + 1$ in the MLE formulation of (3.39), the optimization problem becomes

\[
\begin{align*}
\text{minimize} & \quad \sum_{\tau=t-T_0}^t \left( \sum_{i \sim j \in E_{\tau}(\tau)} \frac{1}{2\sigma_{ij}^2} \| x_i(\tau) - x_j(\tau) - y_{ij}(\tau) \|^2 + \sum_{i \in V(\tau)} \sum_{k \in A_i^\tau(\tau)} \frac{1}{2\varsigma_{ik}^2} \| x_i(\tau) - a_k(\tau) - w_{ik}(\tau) \|^2 \right. \\
& \quad \left. - \sum_{i \sim j \in E_{\tau}(\tau)} \kappa_{ij} u_{ij}^T(\tau) \frac{y_{ij}(\tau)}{d_{ij}(\tau)} - \sum_{i \in V(\tau)} \sum_{k \in A_i^\tau(\tau)} \lambda_{ik} q_{ik}^T(\tau) \frac{w_{ik}(\tau)}{r_{ik}(\tau)} \right) \\
\text{subject to} & \quad \| y_{ij}(\tau) \| \leq d_{ij}(\tau), \| w_{ik}(\tau) \| \leq r_{ik}(\tau).
\end{align*}
\]

(3.40)

With the introduction of the time window, the variables $x, y$ and $w$ are optimized, so that the trajectory best matches the set of observed ranges and bearings, for each time instant. In the MLE formulation, the addition of the time window is redundant, since the time instants are not coupled to each other. However, the time window is introduced now, because, in the next section, the variables of the MAP estimator will depend on previous values, meaning that the time instants will be coupled.

### 3.3 Maximum a Posteriori Estimator with Missing Data

In this section, we present the novelty part of our method. We introduce a MAP estimate formulation to take into account packet losses and random delays.

In the MAP estimator approach, the posterior probability density function of a parameter is maximized. As mentioned in Chapter 2, this approach resembles the maximum likelihood estimator, with the presence of the likelihood function, and also, the prior PDF is introduced.

First, the prior probability function is defined for each one of the variables that are related to the missing measurements, the distance between nodes $d_{ij}$, the distance between a node and an anchor $r_{ik}$ and the positions of the nodes $x_i$. Then, the optimization problem is formulated. Finally, it is analyzed the possibility of the occurrence of consecutive missing measurements.

#### 3.3.1 Prior Data Model for Positions

At a given time instant $t$, it is assumed that the nodes of our network traveled in average a certain distance during the time interval. The distance traveled is limited by a maximum value, which is obtained considering the maximum distance that an AUV can travel in a certain amount of time, by taking into account the speed of the vehicle.

The position of each node of our network is assumed to be related to the position of the previous time instant $t - 1$, by

\[
x_i(t) = x_i(t-1) + \epsilon_i^x \tag{3.41}
\]

where $\epsilon_i^x$ is the associated unknown displacement.

The error associated with 3.41 is modeled by a Gaussian distribution with zero mean and variance
\[ \beta^2 I_p, \] that is

\[ \epsilon_i^x \sim \mathcal{N}(0, \beta_i^2 I_p). \]  

(3.42)

The standard deviation \( \beta_i \) corresponds to the maximum distance that a node can travel in a time instant and \( I_p \) is the identity matrix with appropriate dimension. The position of the node \( i \) is a linear transformation of \( \epsilon_i^x \) with mean \( x_i(t-1) \) and variance \( \beta_i^2 \), that is

\[ x_i(t) \sim \mathcal{N}(x_i(t-1), \beta_i^2 I_p). \]  

(3.43)

The respective probability function is

\[ p(x_i(t); \beta_i) = \frac{1}{\sqrt{2\pi\beta_i^2}} \exp\left(-\frac{|x_i(t) - x_i(t-1)|^2}{2\beta_i^2}\right). \]  

(3.44)

To better understand this data model, the reasoning behind it is clarified. Between time instants, a vehicle can travel a distance \( \beta_i \) with a probability of 68%, since in a normal distribution about 68% of values fall within one standard deviation of the mean. \( \beta_i \) is defined by the speed of the vehicle and the sampling time. Figure 3.7 shows a scheme where this reasoning is illustrated. Two nodes travel at certain speeds and a packet loss between the two nodes occurs at instant \( t \). To estimate the position of the nodes at time \( t \), it is assumed that node 1 traveled \( \beta_1 \) from the previous position at \( t-1 \) and node 2 traveled \( \beta_2 \) as well.

### 3.3.2 Prior Data Model for Ranges

Similarly to the data model of the positions, in the presence of a missing measurement at instant \( t \), it is assumed that the missing distance measurement \( d_{ij}(t) \) is related to the distance measurement of the previous time instant \( d_{ij}(t-1) \). When a packet is lost, we assume that the distance measured between instants cannot change significantly, because AUVs commonly travel in formation and \( \Delta t \), the amount of time between time samples is so considerably small that the vehicles do not travel very far in the meantime.

Thereby, in the case of a missing distance measurement between two nodes, \( d_{ij}(t) \) will be, from now on, defined as \( \Omega_{ij}(t) \). That same distance is estimated by

\[ \Omega_{ij}(t) = d_{ij}(t-1) + \epsilon_{ij}^d \]  

(3.45)

where \( d_{ij}(t-1) \) is the value of the range measurement between nodes \( i \) and \( j \) in the previous time instant \( t-1 \), and therefore we can consider it as a constant when estimating \( \Omega_{ij}(t) \). \( \epsilon_{ij}^d \) is the associated error, which is modeled with a Gaussian distribution with zero mean and variance \( \rho^2 \), that is

\[ \epsilon_{ij}^d \sim \mathcal{N}(0, \rho_{ij}^2). \]  

(3.46)

where the standard deviation \( \rho \) corresponds to the standard deviation of the missing distance measure-
ment error. We use the simplifying assumption that $\epsilon_{ij}^d$ is independent of $\epsilon_i^x$ and $\epsilon_j^x$.

Therefore, the distribution of the missing distance measurement has the form

$$\Omega_{ij}(t) \sim \mathcal{N}(d_{ij}(t-1), \rho_{ij}^2).$$

(3.47)

The respective PDF is

$$p(\Omega_{ij}(t); \rho_{ij}) = \frac{1}{\sqrt{2\pi\rho_{ij}^2}} \exp\left(-\frac{(\Omega_{ij}(t) - d_{ij}(t-1))^2}{2\rho_{ij}^2}\right).$$

(3.48)

In the same way, the missing measurement of distance between one node and one anchor, from now on defined as $\Upsilon_{ik}$, is estimated by

$$\Upsilon_{ik}(t) = r_{ik}(t-1) + \epsilon_{ik}^r.$$

(3.49)

Again, we assume that $\epsilon_{ik}^r$ is independent of $\epsilon_i^x$.

The respective distribution and PDF are given by

$$\Upsilon_{ik}(t) \sim \mathcal{N}(r_{ik}(t-1), \delta_{ik}^2)$$

(3.50)

and

$$p(\Upsilon_{ik}(t); \delta_{ik}) = \frac{1}{\sqrt{2\pi\delta_{ik}^2}} \exp\left(-\frac{(\Upsilon_{ik}(t) - r_{ik}(t-1))^2}{2\delta_{ik}^2}\right).$$

(3.51)

In Figure 3.7 is illustrated the reasoning behind this data model. When there is a missing measurement between nodes 1 and 2 at time $t$, the distance $\Omega_{12}(t)$ is estimated based on the previous distance measurement $d_{12}(t-1)$ plus a parameter $\rho_{12}$. This parameter is related to the position parameters $\beta_1$ and $\beta_2$, in such that $-(\beta_1 + \beta_2) \geq \rho_{12} \leq \beta_1 + \beta_2$. The maximum distance $\Omega_{12}(t)$ for one standard deviation corresponds to $d_{12}(t-1) + (\beta_1 + \beta_2)$ and the minimum $d_{12}(t)$ is given by $d_{12}(t-1) - (\beta_1 + \beta_2)$, with high probability.
Figure 3.7: Illustrative example of how the MAP estimator handles missing measurements. When nodes 1 and 2 are traveling at certain speeds, the maximum distance possible to be traveled between \( t-1 \) and \( t \) is limited by \( \beta_1 \) and \( \beta_2 \), respectively. The distance \( \Omega_{12} \) between nodes at \( t \) is obtained by \( d_{12} \) at \( t-1 \) plus the parameter \( \rho_{12} \), which is bounded by \( -(\beta_1 + \beta_2) \geq \rho_{12} \leq \beta_1 + \beta_2 \).

### 3.3.3 Posterior Distribution

The posterior distribution is given by the product of the likelihood function with the prior distribution. In our problem, with the time window of size \( T_0 + 1 \), this results in

\[
\mathcal{P}(x_i(\tau), \Omega_{ij}(\tau), \Upsilon_{ik}(\tau)) = \prod_{\tau=t-T_0}^t \left( \mathcal{L}(d_{ij}(\tau), r_{ik}(\tau), u_{ij}(\tau), q_{ik}(t); x_i(\tau)) \times \right.
\]

\[
\prod_{i \in \mathcal{V}(\tau)} p(x_i(\tau); \beta_i) \prod_{i \sim j \in \mathcal{E}_i(\tau)} p(\Omega_{ij}(\tau); \rho_{ij}) \prod_{i \in \mathcal{V}(\tau)} \left. \prod_{k \in \mathcal{A}_{i}(\tau)} p(\Upsilon_{ik}(\tau); \delta_{ik}) \right). \tag{3.52}
\]

As in Section 3.2, it will be more convenient to maximize the log of the MAP function, called the log-a-posteriori, which gives us the final form for MAP estimation of parameters.

The log-a-posteriori function is

\[
\ln \mathcal{P}(x_i(\tau), \Omega_{ij}(\tau), \Upsilon_{ik}(\tau)) = \ln \mathcal{L}(d_{ij}(\tau), r_{ik}(\tau), u_{ij}(\tau), q_{ik}(t); x_i(\tau)) + \sum_{i \in \mathcal{V}(\tau)} \ln p(x_i(\tau); \beta_i) + \sum_{i \sim j \in \mathcal{E}_i(\tau)} \ln p(\Omega_{ij}(\tau); \rho_{ij}) + \sum_{i \in \mathcal{V}(\tau)} \sum_{k \in \mathcal{A}_{i}(\tau)} \ln p(\Upsilon_{ik}(\tau); \delta_{ik}), \tag{3.53}
\]

with

\[
\ln p(x_i(\tau); \beta_i) = Z^p_{\beta_i} - \frac{1}{2 \beta_i^2} ||x_i(\tau) - x_i(\tau - 1)||^2, \tag{3.54}
\]

\[
\ln p(\Omega_{ij}(\tau); \rho_{ij}) = Z^p_{\rho_{ij}} - \frac{1}{2 \rho_{ij}^2} (\Omega_{ij}(\tau) - d_{ij}(\tau - 1))^2, \tag{3.55}
\]

and

\[
\ln p(\Upsilon_{ik}(\tau); \delta_{ik}) = Z^p_{\delta_{ik}} - \frac{1}{2 \delta_{ik}^2} (\Upsilon_{ik}(\tau) - r_{ik}(\tau - 1))^2, \tag{3.56}
\]
where $Z^p_x$, $Z^p_d$ and $Z^p_r$ are constants.

### 3.3.4 Optimization Problem

As mentioned previously in the formulation of the optimization problem for the MLE, the maximization of the log-a-posteriori function is equivalent to the minimization of its respective negative function.

The terms of the optimization problem can be separated into distance measurements, angle measurements and prior terms. These are defined as

$$h_{\text{dist}}(\tau) = \sum_{i \sim j \in \mathcal{E}_x(\tau)} \frac{1}{2\sigma^2_{ij}} ((|x_i(\tau) - x_j(\tau)| - d_{ij}(\tau))^2 + \sum_{i \in \mathcal{V}(\tau) \ k \in \mathcal{A}^p_x(\tau)} \frac{1}{2\varsigma^2_{ik}} ((|x_i(\tau) - a_k(\tau)| - r_{ik}(\tau))^2), \quad (3.57)$$

$$h_{\text{angle}}(\tau) = - \sum_{i \sim j \in \mathcal{E}_a(\tau)} \kappa_{ij} \sum_{T} ((x_i(\tau) - x_j(\tau)) - \sum_{i \in \mathcal{V}(\tau) \ k \in \mathcal{A}^p_a(\tau)} \lambda_{ik} q_{ik}^T (x_i(\tau) - a_k(\tau))^2 \quad (3.58)$$

and

$$h_{\text{prior}}(\tau) = \sum_{i \in \mathcal{V}(\tau)} \frac{1}{2\beta^2_i} ||x_i(\tau) - x_i(\tau - 1)||^2 + \sum_{i \sim j \in \mathcal{E}_a(\tau)} \frac{1}{2\rho^2_{ij}} ((\Omega_{ij}(\tau) - d_{ij}(\tau - 1))^2$$

$$+ \sum_{i \in \mathcal{V}(\tau) \ k \in \mathcal{A}^p_a(\tau)} \frac{1}{2\delta^2_{ik}} ((\Upsilon_{ik}(\tau) - r_{ik}(\tau - 1))^2, \quad (3.59)$$

respectively.

The optimization problem has the following form

$$\min_{x(\tau), \Omega(\tau), \Upsilon(\tau)} \sum_{\tau = t - T_0}^t (h_{\text{dist}}(\tau) + h_{\text{angle}}(\tau) + h_{\text{prior}}(\tau)). \quad (3.60)$$

The distance and angle terms of Problem (3.60) correspond to the optimization problem of the MLE (3.28) and are expressed without the convex relaxation in (3.57) and (3.58). Since a convexity analysis and a subsequent convex relaxation of those terms was already performed, we will now proceed to the convex analysis of the new terms resulting from the prior distribution.

### Convexity Analysis

The new terms (3.59) that derived from the prior distribution have the quadratic form

$$f(x) = (x - c)^2 \quad (3.61)$$

where $c$ is a constant.

In the convex analysis in section 3.2, a function of the same form of (3.61) was considered. As can be observed in Figure (3.2), a quadratic function, like the one in study, is convex. Hence, (3.59) is convex. Meaning that the terms regarding the prior distribution introduced in the MAP estimation are convex.

33
Reformulation with Relaxed Terms

In Section 3.2, a relaxation of the non-convex terms was proposed. The expressions for distance measurements and angle measurements with the convex relaxation applied are

\[
\hat{h}_{\text{dist}}(\tau) = \sum_{i \sim j \in E^s(\tau)} \frac{1}{2\sigma_{ij}^2} ||x_i(\tau) - x_j(\tau) - y_{ij}(\tau)||^2 + \sum_{i \in V(\tau)} \sum_{k \in A^o_i(\tau)} \frac{1}{2\sigma_{ik}^2} ||x_i(\tau) - a_k(\tau) - w_{ik}(\tau)||^2 \tag{3.62}
\]

and

\[
\hat{h}_{\text{angle}}(\tau) = - \sum_{i \sim j \in E^s(\tau)} \kappa_{ij} u_{ij}^T(\tau) \frac{y_{ij}(\tau)}{d_{ij}(\tau)} - \sum_{i \in V(\tau)} \sum_{k \in A^o_i(\tau)} \lambda_{ik} q_{ik}^T(\tau) \frac{w_{ik}(\tau)}{r_{ik}(\tau)} \tag{3.63}
\]

The terms of the prior distribution remain equal to (3.59) considering that they are convex.

The final expression for the optimization problem with the convex relaxation is defined as

\[
\begin{align*}
\text{minimize} & \quad \sum_{\tau = t \sim T_0} \left( \hat{h}_{\text{dist}}(\tau) + \hat{h}_{\text{angle}}(\tau) + h_{\text{prior}}(\tau) \right) \\
\text{subject to} & \quad ||y_{ij}(\tau)|| \leq d_{ij}(\tau), ||w_{ik}(\tau)|| \leq r_{ik}(\tau) \\
& \quad \text{if} \quad i \sim j \in E_u(\tau) : ||y_{ij}(\tau)|| \leq \Omega_{ij}(\tau) \\
& \quad \text{if} \quad k \in A_u(\tau), i \in V(\tau) : ||w_{ik}(\tau)|| \leq \Upsilon_{ik}(\tau).
\end{align*}
\tag{3.64}
\]

3.4 Interpretation of the MAP estimator

At this point, the final optimization problem is defined. In this section, we proceed to describe the missing measurement phenomenon and how the model proposed handles the packet loss.

Data drop-outs in the transmission of information in a network are inevitable and they occur randomly. The stochastic missing measurement phenomenon is described by a series of mutually independent random variables obeying a certain Bernoulli distribution.

When a random packet loss occurs, the MAP estimator engages the prior distribution of the position of the nodes and the prior distribution of the missing measurement. The terms of the posterior distribution, which correspond to the terms of the MLE, are no longer observed data, but random variables to be estimated. To better explain this, a simple example is presented. Considering a small network with only 3 nodes connected between them, meaning that there are 3 edges, and for simplicity, no anchors are considered. At a particular time \( t \) when all measurements are available, the problem formulation is

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2\sigma_{12}^2} ||x_1(t) - x_2(t) - y_{12}(t)||^2 + \frac{1}{2\sigma_{13}^2} ||x_1(t) - x_3(t) - y_{13}(t)||^2 \\
& \quad + \frac{1}{2\sigma_{23}^2} ||x_2(t) - x_3(t) - y_{23}(t)||^2 - \kappa_{12} u^T_{12}(t) \frac{y_{12}(t)}{d_{12}(t)} \\
& \quad - \kappa_{13} u^T_{13}(t) \frac{y_{13}(t)}{d_{13}(t)} - \kappa_{23} u^T_{23}(t) \frac{y_{23}(t)}{d_{23}(t)} \\
\text{subject to} & \quad ||y_{12}(t)|| \leq d_{12}(t), ||y_{13}(t)|| \leq d_{13}(t), ||y_{23}(t)|| \leq d_{23}(t).
\end{align*}
\tag{3.65}
\]

This formulation corresponds to the MLE. If at the time instant \( t \), a random packet drop-out between
nodes 1 and 2 occurs, the distance and angle measurements are missing. To deal with the packet loss, our estimator employs the prior distributions. The optimization problem at that time instant becomes

$$\min_{x(t),y(t),\Omega(t)} \frac{1}{2\sigma^2_1}||x_1(t) - x_1(t-1)||^2 + \frac{1}{2\sigma^2_2}||x_2(t) - x_2(t-1)||^2 + (\Omega_{12}(t) - d_{12}(t-1))^2$$
$$+ \frac{1}{2\sigma^2_3}||x_1(t) - x_3(t) - y_{13}(t)||^2 + \frac{1}{2\sigma^2_3}||x_2(t) - x_3(t) - y_{23}(t)||^2$$
$$- \kappa_{13} u_{13}'(t) \frac{y_{13}(t)}{d_{13}(t)} - \kappa_{23} u_{23}'(t) \frac{y_{23}(t)}{d_{23}(t)}$$

subject to

$$||y_{13}(t)|| \leq d_{13}(t), ||y_{23}(t)|| \leq d_{23}(t)$$
$$||y_{12}(\tau)|| \leq \Omega_{12}(\tau).$$

(3.66)

The range term $\frac{1}{2\sigma^2_2}||x_1(t) - x_1(t-1)||^2$ from the MLE was substituted by $\frac{1}{2\sigma^2_1}||x_1(t) - x_1(t-1)||^2 + \frac{1}{2\sigma^2_2}||x_2(t) - x_2(t-1)||^2 + (\Omega_{12}(t) - d_{12}(t-1))^2$. This corresponds to the MAP estimator. The angle term $-\kappa_{13} u_{13}'(t) \frac{y_{13}(t)}{d_{13}(t)}$ was also removed from the optimization problem. However, the prior distribution of the angles is not used, because when an angle measurement is missing, the respective term becomes non-convex again. As a first approach, it was preferred to handle only missing distance measurements. The same reasoning is applied in the case of a packet drop-out between one anchor and one node.

### 3.4.1 Consecutive missing measurements

Given the statistical and random nature of the problem, it is necessary to analyze the case when multiple missing measurements occur, before proceeding to solve the optimization problem. The packet loss is a random event and in a large network, the probability of occurring two consecutive missing measurements in one edge is low.

**Assumption:** There are no two consecutive missing measurements of the same type in time.

This means that, if at time instant $t$ there is a missing distance measurement $\Omega_{ij}$ between node $i$ and node $j$, there cannot exist a missing distance measurement $\Omega_{ij}$ between the same nodes $i$ and $j$ at the time instant $t - 1$. The same reasoning is applied to missing distance measurements $T_{ik}$ between nodes and anchors.

To explain this assumption, it is first stated that when two random variables are independent, they are uncorrelated and the covariance between them is zero because there is no linear relationship between them. Hence, the the variance of the sum of two uncorrelated variables $X$ and $Y$ is given by $\text{var}(X+Y) = \text{var}(X) + \text{var}(Y)$.

We assume that the error associated with each range measurement is independent over time. The same reasoning is applied for both types of measurements, node-node, and node-anchor.

When there is one missing distance measurement, that measurement is estimated by (3.45). When there are two consecutive missing range measurements, we have

$$\Omega_{ij}(t) = d_{ij}(t-2) + \epsilon^d_{ij}(t-1)\epsilon^d_{ij}(t-2),$$

(3.67)
and in the case when there are \( n \) consecutive missing measurements, we have

\[
\Omega_{ij}(t) = d_{ij}(t - n) + \sum_{\tau=1}^{n} \rho_{ij}(t - \tau).
\] (3.68)

As mentioned before, the error has a Gaussian distribution with zero mean and variance \( \rho^2 \), that is \( \rho_{ij}(t - \tau) \sim \mathcal{N}(0, \rho^2) \). Therefore, the distribution of \( d_{ij} \) is given by

\[
\Omega_{ij}(t) \sim \mathcal{N}(d_{ij}(t - n), \Xi),
\] (3.69)

where \( \Xi = \text{var}(\sum_{\tau=1}^{n} \rho_{ij}(t - \tau)) \) is the variance of the sum of \( n \) error terms.

The \( n \) zero mean random variables \( \rho_{ij}(t - \tau), \tau = 1, \ldots, n \) are independent. Given the aforementioned property, the variance of the sum of \( n \) error terms is defined by

\[
\text{var}\left(\sum_{\tau=1}^{n} \rho_{ij}(t - \tau)\right) = \text{var}(\rho_{ij}(t - 1)) + \ldots + \text{var}(\rho_{ij}(t - n)) = n\rho^2_{ij},
\] (3.70)

and therefore the variance of the distance term to be estimated is defined by

\[
\text{var}(\Omega_{ij}(t)) = n\rho^2_{ij}.
\] (3.71)

The variance of \( \Omega_{ij}(t) \) obtained in (3.71) can be interpreted to have a linear dependence with \( n \), the number of consecutive missing measurements. With increasing \( n \), the variance of \( \Omega_{ij} \) increases, meaning that the distribution of the values obtained for \( \Omega_{ij} \) is more dispersed. However, in a real-world scenario, the occurrence of missing measurements is a random event and in the presence of a large data set, the loss of two consecutive data packages is not likely to happen.

The term that includes the missing data in the optimization problem would then become

\[
\sum_{i \sim j \in \mathcal{E}_i(t)} \frac{1}{n\rho^2_{ij}} (\Omega_{ij}(t) - d_{ij}(t - n))^2.
\] (3.72)

Similarly, in the case of \( m \) missing distance measurements between node \( i \) and anchor \( k \), the term in the optimization problem would become

\[
\sum_{i \in \mathcal{V}_i(t)} \sum_{k \in \mathcal{A}^*_k(t)} \frac{1}{m_{\delta_{ik}}^2} (\Upsilon_{ik}(t) - r_{ik}(t - m))^2.
\] (3.73)

When this is the case, the weight of the range terms in the cost function will be \( \frac{1}{n\rho^2_{ij}}, n \geq 1 \) and \( \frac{1}{m_{\delta_{ik}}}, m \geq 1 \). If \( n \) or \( m \) are greater than 1, it means that we have at least two consecutive missing measurements and the weight of these terms in the cost function will decrease, which leads to the loss of importance of these terms in the optimization problem. The assumption that two consecutive missing measurements cannot occur is made to avoid that the weight of the respective terms in the optimization problem does not decrease.

At this point, the final optimization problem obtained from the MAP estimator, which includes the
MLE, is defined and the case of consecutive packet loss has been analyzed. The next logical and necessary step is to solve the optimization problem and obtain the estimates of the unknown variables $x$, $\Omega$ and $\Upsilon$. This topic will be covered in the next chapter.

It is also worth mentioning that with this assumption, we are assuming that the occurrence of a missing measurement, at some instant, implies that in the previous instant that measurement was observed. This means that the graph will not be fully connected since the edges corresponding to the lost packets do not exist. In a connected graph, there are no unreachable vertices, but with this assumption, there might exist unreachable vertices.
Chapter 4

Optimization Algorithm

In this chapter, a method to solve the optimization problem obtained in Chapter 3 is presented. It is an accelerated proximal gradient method for a strongly convex function with Lipschitz continuous gradient. First, the optimization problem will be reformulated and then these conditions will be proved. Finally, the optimization method will be presented.

4.1 Problem Reformulation for Implementation

The optimization problem in (3.64) will be rewritten in matrix notation under the considerations presented in this section, to be later implemented.

The arc-node incidence matrix $C$ is defined by

$$C_{ji} = \begin{cases} 
-1 & \text{if edge } j \text{ leaves node } i \\
1 & \text{if edge } j \text{ enters node } i \\
0 & \text{otherwise.} 
\end{cases} \tag{4.1}$$

Matrix $A$ is defined as the Kronecker product of the identity matrix of dimension $T_0$ with the Kronecker product of $C$ with the identity matrix of dimension $p$, that is

$$A = I_{T_0} \otimes (C \otimes I_p) = (I_{T_0} \otimes C) \otimes I_p. \tag{4.2}$$

This can be interpreted as matrix $C$ being extended along the dimension of the problem $p$ and time window $T_0$, assuming the set of edges remains constant over $T_0$. The result of $I_{T_0} \otimes C$ may be interpreted as a "virtual" arc-node incidence matrix with $T_0 \times n$ nodes as exemplified in Figure 4.1.

The direction of the edges is chosen arbitrarily (the choice of signs can be arbitrary as long as one is positive and the other is negative), therefore the entries in $C$ and, consequently, the entries in $A$ corresponding to each node will have a positive or negative sign depending on the chosen edge direction. Then, it is possible to write $\sum_{i \sim j \in E(\tau)} ||x_i(\tau) - x_j(\tau)||^2$ as $||Ax||^2$, where $x$ is a vertical concatenation of all position vectors over time. Therefore, $x(\tau) = \{x_i(\tau)\}_{i \in V(\tau)}$ and $x = \{x(\tau)\}_{t-T_0 \leq \tau \leq t}$.
Figure 4.1: Example of a graph characterized by a network of 4 nodes, where each color represents a
time instant. The lines represent the edges, where the dashed lines are the edge with missing measure-
ments to be estimated along with the position of the nodes. In this case, matrix $A$ is the graph-incidence
matrix of a network with 16 nodes and $C$ is the matrix of each time instant network with 4 nodes.

Vectors $y_{ij}(\tau)$ are also vertically concatenated, first in the same time instant as $y(\tau) = \{y_{ij}(\tau)\}_{i \sim j}$
and then over the time window as $y = \{y(\tau)\}_{t-T_0 \leq \tau \leq t}$.

The distance term, concerning measurements between nodes, is reformulated as

$$\sum_{\tau=t-T_0}^{t} \sum_{i \sim j \in E(\tau)} \frac{1}{2\sigma_{ij}^2} ||x_i(\tau) - x_j(\tau) - y_{ij}(\tau)||^2 = \frac{1}{2} ||\Sigma_N B A x - \Sigma_N B y||^2$$

(4.3)

where $\Sigma_N$ is the diagonal matrix of $\frac{1}{\sigma_{ij}^2}$. Matrix $B$ is an indicator of which terms concerning this type are
missing and removes them from the optimization problem.

The anchor terms follow a similar reasoning, except it is also necessary to concatenate along the
different nodes, other than the edges and the time window. Therefore, $w_i(\tau) = \{w_{ik}(\tau)\}_{k \in A_i(\tau)}$, $w(\tau) =
\{w_i(\tau)\}_{i \in V(\tau)}$ and $w = \{w(\tau)\}_{t-T_0 \leq \tau \leq t}$. Additionally, $\alpha_i(\tau) = \{a_{ik}(\tau)\}_{k \in A_i(\tau)}$, $\alpha(\tau) = \{\alpha_i(\tau)\}_{i \in V(\tau)}$ and $\alpha = \{\alpha(\tau)\}_{t-T_0 \leq \tau \leq t}$.

The distance term, concerning node-anchor measurements, is reformulated as

$$\sum_{\tau=t-T_0}^{t} \sum_{i \in V(\tau)} \sum_{k \in A_i(\tau)} \frac{1}{2\zeta_{ik}^2} ||x_i(\tau) - a_k(\tau) - w_{ik}(\tau)||^2 = \frac{1}{2} ||\Sigma_A D E x - \Sigma_A D A - \Sigma_A D w||^2$$

(4.4)

where $E$ is a selector matrix, indicating which node has a measurement relative to each anchor and $\Sigma_A$
is the diagonal matrix of $\frac{1}{\zeta_{ik}^2}$. Matrix $D$ is an indicator for the terms concerning this type of edge that are
missing, in order to remove them from the optimization problem.

Looking at the angle measurements in (3.63) we define

$$\tilde{u}_{ij}(\tau) = \kappa_{ij} \frac{u_{ij}(\tau)}{d_{ij}(\tau)}$$

(4.5)

and

$$\tilde{q}_{ik}(\tau) = \lambda_{ik} \frac{q_{ik}(\tau)}{r_{ik}(\tau)}.$$  

(4.6)

Following the same reasoning as before, $u(\tau) = \{\tilde{u}_{ij}(\tau)\}_{i \sim j \in E(\tau)}$ and $u = \{u(\tau)\}_{t-T_0 \leq \tau \leq t}$. Similarly,
$q_i(\tau) = \{q_{ik}(\tau)\}_{k \in A_i^+(\tau)}$, $q(\tau) = \{q_i(\tau)\}_{i \in V(\tau)}$ and $q = \{q(\tau)\}_{t-T_0 \leq \tau \leq t}$.

The angle term concerning measurements between nodes may be reformulated as

$$- \sum_{\tau=t-T_0}^{t} \sum_{i \sim j \in E^-(\tau)} \kappa_{ij} u^T_{ij}(\tau) \frac{y_{ij}(\tau)}{d_{ij}(\tau)} = -Bu^T \gamma \quad (4.7)$$

and the angle term concerning measurements between nodes and anchors may be reformulated as

$$- \sum_{\tau=t-T_0}^{t} \sum_{i \in V(\tau)} \sum_{k \in A_i^+(\tau)} \lambda_{ik} q^T_{ik}(\tau) \frac{w_{ik}(\tau)}{r_{ik}(\tau)} = -Dq^T \omega. \quad (4.8)$$

In the presence of a missing measurement, the respective angle term is eliminated from the optimization problem. The angle measurement $u_{ij}$ is removed by the corresponding zero entry from matrix $B$ and the angle measurement $q_{ik}$ is removed by the corresponding zero entry from matrix $D$.

The prior terms follow the same considerations. The prior term concerning the positions of the nodes is reformulated as

$$\sum_{\tau=t-T_0}^{t} \sum_{i \in V(\tau)} \frac{1}{2\beta_i^2} (\|x_i(\tau) - x_i(\tau - 1)\|^2 = \frac{1}{2} \|\Sigma_X x - \Sigma_X x_{\text{prior}}\|^2 \quad (4.9)$$

where $x_{\text{prior}}$ is a vertical concatenation of all previous positions $t-1$ of vector $x$, that is $x_{\text{prior}}(\tau) = \{x_i(\tau - 1)\}_{i \in V(\tau)}$ and $x_{\text{prior}} = \{x_{\text{prior}}(\tau)\}_{t-T_0 \leq \tau \leq t}$, and $\Sigma_X$ is the diagonal matrix of $\frac{1}{\beta_i}$.

The reformulation of the prior term concerning node-node distance measurements follows the same logic as the prior term of position and is defined as

$$\sum_{\tau=t-T_0}^{t} \sum_{i \sim j \in E(\tau)} \frac{1}{2\gamma_{ij}} (\Omega_{ij}(\tau) - d_{ij}(\tau - 1))^2 = \frac{1}{2} \|\Sigma_{\Omega} K \Omega - \Sigma_{\Omega} K d_{\text{prior}}\|^2 \quad (4.10)$$

where $K$ is a selector matrix that indicates which distance measurements of type $d_{ij}$ are missing and consequently will be estimated. Matrix $B$ in (4.3) equals $B = (I - K)$. The concatenated vectors containing the missing variables are composed by $\Omega(\tau) = \{\Omega_{ij}(\tau)\}_{i \sim j \in E(\tau)}$ and $\Omega = \{\Omega(\tau)\}_{t-T_0 \leq \tau \leq t}$.

The concatenated vectors containing the prior values are $d_{\text{prior}}(\tau) = \{d_{ij}(\tau - 1)\}_{i \sim j \in E(\tau)}$ and $d_{\text{prior}} = \{d_{\text{prior}}(\tau)\}_{t-T_0 \leq \tau \leq t}$. $\Sigma_{\Omega}$ is the diagonal matrix of $\frac{1}{\gamma_{ij}}$.

Lastly, the prior term concerning node-anchor distance measurements is reformulated as

$$\sum_{\tau=t-T_0}^{t} \sum_{i \in V(\tau)} \sum_{k \in A_i^+(\tau)} \frac{1}{2\gamma_{ik}^2} (T_{ik}(\tau) - r_{ik}(\tau - 1))^2 = \frac{1}{2} \|\Sigma_T N T - \Sigma_T N r_{\text{prior}}\|^2 \quad (4.11)$$

where $N$ is a selector matrix that indicates which distance measurements of type $r_{ik}$ are missing.

Matrix $D$ in (4.4) equals $D = (I - N)$. The concatenated vectors with the missing variable are $T_i(\tau) = \{T_{ik}(\tau)\}_{k \in A_i^+(\tau)}$, $Y(\tau) = \{Y_i(\tau)\}_{i \in V(\tau)}$, and $Y = \{Y(\tau)\}_{t-T_0 \leq \tau \leq t}$ and with the prior values are $r_{\text{prior}}(\tau) = \{r_{ik}(\tau - 1)\}_{k \in A^+(\tau)}$, $r_{\text{prior}}(\tau) = \{r_{\text{prior}}(\tau)\}_{i \in V(\tau)}$, and $r_{\text{prior}} = \{r_{\text{prior}}(\tau)\}_{t-T_0 \leq \tau \leq t}$ and $\Sigma_T$ is the diagonal matrix of $\frac{1}{\gamma_{ik}}$. 
Thus, the optimization problem in (3.64) may be rewritten as

$$\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| \Sigma_N B A - \Sigma_N B \| z + \frac{1}{2} \| \Sigma_A D E x - \Sigma_A D \| z - \Sigma_A D \| \| \| - B u^T y - D q^T w \\
& + \frac{1}{2} \| \Sigma_N x - \Sigma_N x_{prior} \| + \frac{1}{2} \| \Sigma_N K \| z - \Sigma_N K \| d_{prior} \| z + \frac{1}{2} \| \Sigma_N N T - \Sigma_N N r_{prior} \| \| z \\
\text{subject to} & \quad || y_{ij} (\tau) || \leq d_{ij} (\tau), || w_{ik} (\tau) || \leq r_{ik} (\tau) \\
& \text{if} \quad i \sim j \in E_u (\tau): || y_{ij} (\tau) || \leq \Omega_{ij} (\tau) \\
& \text{if} \quad k \in A_u (\tau), i \in V (\tau): || w_{ik} (\tau) || \leq \Upsilon_{ik} (\tau).
\end{align*}$$

(4.12)

In order to proceed with the reformulation of our problem, we introduce the variable $z = (x, y, w, \Omega, \Upsilon)$ and define $Z = \{ z : || y_{ij} (\tau) || \leq d_{ij} (\tau), i \sim j \in E^d (\tau), || w_{ik} (\tau) || \leq r_{ik} (\tau), i \in V (\tau), k \in A^d (\tau), t - T_0 \leq \tau \leq t \}$.

Problem (4.12) is transformed into

$$\begin{align*}
\text{minimize} & \quad \frac{1}{2} \left[ \begin{array}{cccc} \Sigma_N B A & -\Sigma_N B & 0 & 0 & 0 \\
\Sigma_N x & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \Sigma_N K & 0 \\
0 & 0 & 0 & \Sigma_N N & 0 \end{array} \right] z - \Sigma_A D \alpha || z \| z \\
& + \frac{1}{2} \left[ \begin{array}{cccc} \Sigma_N x & 0 & 0 & 0 \end{array} \right] z - \Sigma_N x_{prior} \| + \frac{1}{2} \left[ \begin{array}{cccc} 0 & 0 & 0 & \Sigma_N K \end{array} \right] \| z - \Sigma_N K \| d_{prior} \| z + \frac{1}{2} \left[ \begin{array}{cccc} \Sigma_N N & 0 & 0 & \Sigma_N N \end{array} \right] \| z - \Sigma_N N r_{prior} \| z \\
\text{subject to} & \quad z \in Z \\
& \text{if} \quad i \sim j \in E_u (\tau): || y_{ij} (\tau) || - \Omega_{ij} (\tau) \leq 0 \\
& \text{if} \quad k \in A_u (\tau), i \in V (\tau): || w_{ik} (\tau) || - \Upsilon_{ik} (\tau) \leq 0.
\end{align*}$$

(4.13)

The formulation in (4.13), can be expressed in a quadratic form

$$\begin{align*}
\text{minimize} & \quad \frac{1}{2} z^T M z - b^T z + S \\
\text{subject to} & \quad z \in Z \\
& \text{if} \quad i \sim j \in E_u (\tau): || y_{ij} (\tau) || - \Omega_{ij} (\tau) \leq 0 \\
& \text{if} \quad k \in A_u (\tau), i \in V (\tau): || w_{ik} (\tau) || - \Upsilon_{ik} (\tau) \leq 0.
\end{align*}$$

(4.14)

Since $S$ is a constant, it does not influence the minimization of the cost function in (4.14), and therefore the problem is equivalent to

$$\begin{align*}
\text{minimize} & \quad \frac{1}{2} z^T M z - b^T z \\
\text{subject to} & \quad z \in Z \\
& \text{if} \quad i \sim j \in E_u (\tau): || y_{ij} (\tau) || - \Omega_{ij} (\tau) \leq 0 \\
& \text{if} \quad k \in A_u (\tau), i \in V (\tau): || w_{ik} (\tau) || - \Upsilon_{ik} (\tau) \leq 0.
\end{align*}$$

(4.15)

To obtain the quadratic form present in (4.15), some algebraic manipulation of the terms in Problem (4.13) is necessary.
The matrix \( M \) that we obtain is defined by \( M = M_1 + M_2 + M_3 + M_4 + M_5 \), where

\[
M_1 = \begin{bmatrix}
A^T B \Sigma_N & -\Sigma_N B & 0 & 0 & 0 \\
\Sigma_N B A & -\Sigma_N B & 0 & 0 & 0
\end{bmatrix},
\]

(4.16)

\[
M_2 = \begin{bmatrix}
E^T D \Sigma_A & 0 & -\Sigma_A D & 0 & 0 \\
\Sigma_A D E & 0 & -\Sigma_A D & 0 & 0
\end{bmatrix},
\]

(4.17)

\[
M_3 = \begin{bmatrix}
\Sigma_X & 0 & 0 & 0 \\
\Sigma_X & 0 & 0 & 0
\end{bmatrix},
\]

(4.18)

\[
M_4 = \begin{bmatrix}
0 & 0 & 0 & K^T \Sigma_\Omega \\
0 & 0 & 0 & \Sigma_\Omega K
\end{bmatrix},
\]

(4.19)

and

\[
M_5 = \begin{bmatrix}
0 & 0 & 0 & N^T \Sigma_Y \\
0 & 0 & 0 & \Sigma_Y N
\end{bmatrix},
\]

(4.20)

The vector \( b \) that we obtain is defined by \( b = b_1 + b_2 + b_3 + b_4 + b_5 \), where

\[
b_1 = \begin{bmatrix}
0 & Bu & Dq & 0 & 0
\end{bmatrix}^T,
\]

(4.21)

\[
b_2 = \begin{bmatrix}
E^T D \Sigma_A & 0 & -\Sigma_A D & 0 & 0
\end{bmatrix}^T \Sigma_A D \alpha,
\]

(4.22)

\[
b_3 = \begin{bmatrix}
\Sigma_X & 0 & 0 & 0 \\
\Sigma_X & 0 & 0 & 0
\end{bmatrix} \Sigma_X x_{prior},
\]

(4.23)

\[
b_4 = \begin{bmatrix}
0 & 0 & 0 & K^T \Sigma_\Omega \\
0 & 0 & 0 & \Sigma_\Omega K
\end{bmatrix} d_{prior},
\]

(4.24)

and

\[
b_5 = \begin{bmatrix}
0 & 0 & 0 & N^T \Sigma_Y \\
0 & 0 & 0 & \Sigma_Y N
\end{bmatrix} r_{prior},
\]

(4.25)

### 4.2 Nesterov’s Method

#### 4.2.1 Background on Strong Convexity

A function is strongly convex if there exists a constant \( m > 0 \) such that

\[
\nabla^2 f(x) \succeq m I
\]

(4.26)

for all \( x \). [10]

The second-order Taylor approximation of \( f \) near \( x \) is given by

\[
f(y) = f(x) + \nabla f(x)^T (y - x) + \frac{1}{2} (y - x)^T \nabla^2 f(z) (y - x)
\]

(4.27)

for all \( x, y \) and some \( z \) on the line segment \([x, y]\). With the strong convexity assumption in (4.26), the Taylor series approximation induces the inequality

\[
f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{m}{2} \|y - x\|^2
\]

(4.28)
for all \(x, y\). When \(m = 0\), we obtain the inequality equivalent to the first-order condition that characterizes a convex function. When \(m > 0\), we obtain a better lower bound on \(f(y)\). In Figure 4.2 it is possible to observe the difference between the lower bound of a convex function (being \(f(y)\) a convex function) and the lower bound of a strongly convex function (being \(f(y)\) a strongly convex function).

Intuitively speaking, strong convexity means that there exists a quadratic lower bound on the growth of the function. This directly implies that a strongly convex function is strictly convex, since the quadratic lower bound growth is, of course, strictly greater than the linear growth.

Moreover, as an equivalent condition for strong convexity, for any convex function \(f\), the function \(g(x) = f(x) + \frac{m}{2}||x||^2\) is strongly convex. [31]

![Figure 4.2: Representation of the quadratic lower bound of a strongly convex function](image)

\[
f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{m}{2}||y - x||^2
\]

and the linear lower bound of a convex function
\[
f(y) \geq f(x) + \nabla f(x)^T (y - x)
\]

[32].

4.2.2 Background on Lipschitz Continuity

A differentiable function \(f\) is said to have a Lipschitz continuous gradient if for some \(L > 0\) the following condition is true

\[
||\nabla f(x) - \nabla f(y)||_2 \leq L||x - y||_2 \tag{4.29}
\]

for all \(x, y \in \text{dom } f\). However, this definition does not assume the convexity of \(f\).

Essentially, this states that the gradient is limited in how fast it can change. Besides, the condition is satisfied if \(f\) is twice differentiable and the Hessian is bounded by \(L\) over \(\mathbb{R}^n\), for some \(L \geq 0\).

The Lipschitz constant \(L\), which can be interpreted as a bound on the second derivative of \(f\), measures how well \(f\) can be approximated by a quadratic model.

If \(\nabla f\) is Lipschitz continuous with parameter \(L\) and \(\text{dom } f\) is convex, then (4.29) is equivalent to

\[
f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{L}{2}||y - x||^2 \tag{4.30}
\]

which can be interpreted as a quadratic upper bound on \(f\) [32]. In Figure 4.3 we can observe the quadratic upper bound on \(f\).
Quadratic functions, like the one present in Problem (4.15), have a Lipschitz continuous gradient.

### 4.2.3 Nesterov’s Method for a strongly convex function

The method chosen to minimize Problem (4.15) is *Nesterov’s simplest method*, which is derived from the gradient method, a classical solution for minimization of continuously differentiable functions. Reference [33] explicitly presents an overview of this method.

The gradient method consists of the minimization of a convex differentiable function $f$ by following the negative direction of the function’s gradient. The algorithm used to minimize $f$ over $x \in \mathbb{R}^n$, choosing an initial point $x_0 \in \mathbb{R}^n$, is defined by

$$x_{k+1} = x_k - t_k \nabla f(x_k) \quad (4.31)$$

where $t_k > 0$ is the step size, which is constant or otherwise determined by line search, and defines “how much should be traveled along the gradient” in each iteration [32].

Nesterov’s method follows along the same lines with some modifications. It is an accelerated proximal gradient method for minimizing constrained problems with composite objective functions [34].

The problem is defined as

$$\text{minimize}_{x} \quad f(x) = g(x) + h(x) \quad (4.32)$$

where $g(x) : \mathbb{R}^n \to \mathbb{R}$ is a differentiable function with Lipschitz continuous gradient $L$ and $h(x)$ is closed and convex, so that the prox, operator is well defined. Also, there exist some constants $m \geq 0$ and $L \geq 0$, designated as the strong convexity constant and the Lipschitz constant, respectively, such that the functions $g(x) - \frac{m}{2} x^T x$ and $\frac{L}{2} x^T x - g(x)$ are convex.

Under the aforementioned assumptions, Nesterov’s Method to find a solution for Problem (4.15), where our objective function is strongly convex ($m > 0$), is given by

$$y = x^{(k-1)} + \frac{\sqrt{t_k}}{\sqrt{k-1}} \frac{1 - \sqrt{m/L}}{1 + \sqrt{m/L}} \left( x^{(k-1)} - x^{(k-2)} \right) \quad (4.33)$$

$$x^{(k)} = \text{prox}_{t_k h}(y - t_k \nabla g(y)), \quad (4.34)$$
where the prox-operator (or proximal mapping) of a function \( h(x) \) is defined as
\[
\text{prox}_h(x) = \arg\min_u \left( h(u) + \frac{1}{2} ||u - x||^2 \right).
\] (4.35)

With constant step size \( t_k = 1/L \), \( y \) reduces to
\[
y = x^{(k-1)} + \frac{1 - \sqrt{m/L}}{1 + \sqrt{m/L}} \left( x^{(k-1)} - x^{(k-2)} \right).
\] (4.36)

**Implementation**

In order to solve Problem (4.15) with the method presented in this section, we need to define \( g(z) \), \( h(z) \), \( \text{prox}_h(.) \) and \( L \) and \( m \).

The minimization function in (4.32) is composed by \( g(z) \) and \( h(z) \), where \( g \) is a differentiable function and \( h \) is not differentiable. Since \( f(z) = g(z) + h(z) \) is equivalent to our optimization Problem (4.15), which is differentiable, it is straightforward that \( g \) is equivalent to the optimization problem and \( h \) will produce a 0 value, therefore
\[
g(z) = \frac{1}{2} z^T M z - b^T z
\] (4.37)
and
\[
h(z) = \begin{cases} 
0 & \text{if } z \in \mathcal{Z} \\
\infty & \text{otherwise}.
\end{cases}
\] (4.38)

The function \( h(x) \) is the indicator function of a closed convex set \( C \). The prox-operator \( \text{prox}_h \) is the projection of \( h \) onto \( C \). When \( h(x) = 0 \), the prox-operator is \( \text{prox}_h(x) = x \) [35].

We need to take the constraints in set \( \mathcal{Z} \) into account to solve our problem. The conditions in \( \mathcal{Z} \) are projections onto L2-norm balls centered at the origin, with the general format \( \{ z : ||z|| \leq d \} \).

The definition of the projection of \( z \) onto set \( \mathcal{Z} \) is given by
\[
\mathcal{P}_\mathcal{Z}(z) = \arg\min_{z \in \mathcal{Z}} ||x - z||^2.
\] (4.39)

The projection of \( z \) onto set \( \mathcal{Z} \) is the rescaling of point \( z \) towards the origin when its norm exceeds \( d \), that is
\[
\mathcal{P}_\mathcal{Z}(z) = \begin{cases} 
z, & \text{if } ||z|| \leq d \\
\frac{d}{||z||}z, & \text{if } ||z|| > d,
\end{cases}
\] (4.40)
and the projection onto \( \mathcal{Z} \) is the prox-operator of the indicator function \( h \).

Finally, it is necessary to obtain the values for the Lipschitz constant \( L \) and the strong convexity constant \( m \) of the function \( g(x) = \frac{1}{2} z^T M z - b^T z \). The value of the size step is given by \( t_k = 1/L \).
Strong convexity constant derivation

The strong convexity constant \( m \) is obtained from

\[
f(z) = \hat{f}(z) + \frac{m}{2} ||z - b||^2
\]

(4.41)

where \( \hat{f}(z) \) is a convex function.

The objective function of problem (3.64) can be transformed into the form presented in (4.41). Recalling the variable \( z = (x, y, w, \Omega, \Upsilon) \), the convex function \( \hat{f} \) of problem is given by

\[
\hat{f}(z) = \sum_{\tau=t-T_0}^{t} \left( \hat{h}_{\text{dist}}(\tau) + \hat{h}_{\text{angle}}(\tau) \right),
\]

(4.42)

with \( \hat{h}_{\text{dist}} \) and \( \hat{h}_{\text{angle}} \) being the relaxed terms regarding the distance measurements and angle measurements defined in (3.62) and (3.63), respectively. With the relaxation performed, the function is now convex, as proven in the last chapter.

The term in the optimization problem (3.64) regarding the prior terms \( h_{\text{prior}}(\tau) \), defined in (3.59), will be transformed into \( m||z - b||^2 \). In order to obtain \( h_{\text{prior}} = m||z - b||^2 \) some algebraic manipulation is necessary, given by

\[
h_{\text{prior}}(\tau) = \sum_{i \in V(\tau)} \frac{1}{\beta_i} \left| \left| x_i(\tau) - x_i(\tau - 1) \right| \right|^2 + \sum_{i \sim j \in E_s(\tau)} \frac{1}{\rho_{ij}} \left( \Omega_{ij}(\tau) - d_{ij}(\tau - 1) \right)^2 + \\
+ \sum_{i \in V(\tau)} \sum_{k \in A_s(\tau)} \frac{1}{\delta_{ik}} \left( \Upsilon_{ik}(\tau) - r_{ik}(\tau - 1) \right)^2
\]

(4.43)

Using the definition of the Euclidean norm, \( ||x|| = \sqrt{x_1^2 + \ldots + x_n^2} \), the expression above can be reformulated, resulting in

\[
\frac{1}{2} \left\| \left( \frac{v_1}{\beta_i}, \frac{v_2}{\rho_{ij}}, \frac{v_3}{\delta_{ik}} \right) \right\|^2 = \frac{1}{2} \left\| \begin{bmatrix} \Sigma_X & 0 & 0 \\ 0 & \Sigma_\Omega & 0 \\ 0 & 0 & \Sigma_\Upsilon \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \right\|^2
\]

(4.44)

where \( v_1, v_2 \) and \( v_3 \) are defined by

\[
v_1 = \sum_{i \in V(\tau)} \left| \left| x_i(\tau) - x_i(\tau - 1) \right| \right|,
\]

(4.45)

\[
v_2 = \sum_{i \sim j \in E_s(\tau)} \Omega_{ij}(\tau) - d_{ij}(\tau - 1),
\]

(4.46)
and
\[ v_3 = \sum_{i \in V(\tau)} \sum_{k \in A_i(\tau)} T_{ik}(\tau) - r_{ik}(\tau - 1). \] (4.47)

The norm of the multiplication between a matrix \( A \) and a vector \( x \) satisfies the following subordinance property
\[ ||Ax||_2 \leq ||A||_F ||x||_2, \] (4.48)
where the subscript \( ||.||_F \) denotes the Frobenius norm [36]. The Frobenius norm of a real matrix \( A \in \mathbb{R}^{m \times n} \) is defined as the square root of the sum of the absolute squares of its elements, that is
\[ ||A||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} = \sqrt{\text{tr}(AA^T)}. \] (4.49)

In order to convert (4.44) to the form observed in (4.41), the property presented above is applied and subsequently the strong convexity constant is obtained. The algebraic manipulation to obtain \( m \) is given by
\[ \frac{1}{\sqrt{2}} \left\| \begin{bmatrix} \Sigma_X & 0 & 0 \\ 0 & \Sigma_\Omega & 0 \\ 0 & 0 & \Sigma_\Upsilon \end{bmatrix} \right\|^2 \leq \frac{1}{2} \left\| \begin{bmatrix} \Sigma_X & 0 & 0 \\ 0 & \Sigma_\Omega & 0 \\ 0 & 0 & \Sigma_\Upsilon \end{bmatrix} \right\|_F^2 \]
\[ = (\text{tr}(\Sigma_X^2) + \text{tr}(\Sigma_\Omega^2) + \text{tr}(\Sigma_\Upsilon^2)) \left\| (v_1, v_2, v_3) \right\|^2 \]
\[ = m \left\| z - b \right\|^2, \] (4.50)

with \( z = (x, \Omega, \Upsilon) \) and \( b = (x_{\text{prior}}, d_{\text{prior}}, r_{\text{prior}}) \). The variables \( y \) and \( w \) from the optimization problem are omitted here, since they do not influence the computation of \( m \).

The strong convexity constant chosen is the upper-bound of (4.50), therefore its value is given by
\[ m = \sum_{i \in V(\tau)} \frac{1}{\rho_i} + \sum_{i \sim j \in E(\tau)} \frac{1}{\beta_{ij}} + \sum_{i \in V(\tau)} \sum_{k \in A_i(\tau)} \frac{1}{\varsigma_{ik}}. \]

Lipschitz constant derivation

The Lipschitz constant is obtained by the condition in (4.29) and the gradient difference is manipulated until an expression with the same structure is obtained, that is
\[ ||\nabla f(x) - \nabla f(y)||_2 = L||M(x - y)||_2 \]
\[ = ||M||_2 ||x - y||_2 \] (4.51)

The last step of (4.51) corresponds to the definition of Lipschitz continuity with \( L = ||M|| \), where \( ||M|| \) is the maximum singular value norm of matrix \( M \) [37]. Further manipulations are necessary to obtain a bound for this constant.
\[ L = ||M|| \]

\[
\begin{align*}
&\leq \frac{||M1|| + ||M2|| + ||M3|| + ||M4|| + ||M5||}{1 + \sqrt{m/L}}(z^k - z^{k-1}) \\
&= \frac{\sqrt{\lambda_{\max}(M_1^T M_1) + \sqrt{\lambda_{\max}(M_2^T M_2) + \sqrt{\lambda_{\max}(M_3^T M_3) + \sqrt{\lambda_{\max}(M_4^T M_4) + \sqrt{\lambda_{\max}(M_5^T M_5)}}}}}{1 + \sqrt{m/L}}(z^k - z^{k-1}) \\
&= \lambda_{\max}(M_1) + \lambda_{\max}(M_2) + \lambda_{\max}(M_3) + \lambda_{\max}(M_4) + \lambda_{\max}(M_5)
\end{align*}
\]

(4.52)

The justification of these calculations and the properties applied to obtain the upper bound of \( L \) are presented below.

(a) Triangle inequality property, \( ||A + B|| \leq ||A|| + ||B|| \) for matrices \( A \) and \( B \);

(b) Matrix norm \( ||A|| = \sqrt{\lambda_{\max}(A^T A)} \) where \( \lambda_{\max}(A) \) represents the largest singular value of matrix \( A \);

(c) Matrices \( M_1, M_2, M_3, M_4 \) and \( M_5 \) are symmetric \( (A = A^T) \), verifying that \( \sqrt{\lambda_{\max}(A^T A)} = \lambda_{\max}(A) \).

### 4.2.4 Final Distributed Algorithm

The values of \( m, L \) and the correct gradient are introduced in (4.33) and (4.34) and the explicit formulation of the Nesterov method for distributed application is given by

\[ z^k = z^k + \frac{1 - \sqrt{m/L}}{1 + \sqrt{m/L}}(z^k - z^{k-1}) \]  

(4.53)

and

\[ z^{k+1} = P_Z(z^k - \frac{1}{L}(Mz^k - b)) = P_Z(z^k - \frac{1}{L}Mz^k + \frac{1}{L}b). \]

(4.54)

Developing expression (4.54) for each component of \( z \), the following expressions are obtained

\[ x^{k+1} = \left( I - \frac{1}{L}T \right) \hat{x}^k + \frac{1}{L}A^T\Sigma_N^2 B \hat{y}^k + \frac{1}{L}E^T D\Sigma_A^2(\hat{w}^k + \alpha) \]

(4.55)

\[ Y^{k+1} = P_Y \left( \frac{1}{L} \Sigma_N^2 B A \hat{x}^k + \left( I + \frac{1}{L} \Sigma_N^2 D^2 \right) \hat{y}^k + \frac{1}{L} Bu \right) \]

(4.56)

\[ \omega^{k+1} = P_W \left( \frac{1}{L} \Sigma_N^2 D E \hat{x}^k + \left( I - \frac{1}{L} \Sigma_N^2 D^2 \hat{x}^k + \frac{1}{L} D q - \frac{1}{L} \Sigma_N^2 D \alpha \right) \right) \]

(4.57)

\[ \Omega^{k+1} = \left( I - \frac{1}{L} \Sigma_N^2 K^T K \right) \hat{\Omega}^k + \frac{1}{L} \Sigma_N^2 K^T K d_{\text{prior}} \]

(4.58)

\[ \Sigma^{k+1} = \left( I - \frac{1}{L} \Sigma_N^2 N^T N \right) \hat{\Sigma}^k + \frac{1}{L} \Sigma_N^2 N^T N r_{\text{prior}} \]

(4.59)

where \( T = \Sigma_N^2 A^T A + \Sigma_N^2 E^T E + \Sigma_N^2 \). \( P_Y \) and \( P_W \) are the projections onto the subsets of \( Z \) of the respective variables \( y, w \), defined as \( Y = \{ y(t) : ||y_{ij}(t) \leq d_{ij}, if i \sim j \in \mathcal{E}_u(\tau) : ||y_{ij}(\tau)|| \leq \Omega_{ij}(\tau) \} \) and

48
\[ W = \{ w(t) : \| w_{ik}(t) \| \leq r_{ik}, \text{if } k \in A_\nu(t), i \in V(\tau) : \| w_{ik}(\tau) \| \leq \tau_{ik}(\tau) \}. \] The positions of the nodes \( x_i \) and the missing measurements, \( \Omega_{ij} \) and \( \Upsilon_{ik} \), are not constrained, therefore the update for variables \( x, \Omega \) and \( \Upsilon \) in \( z \) requires no projection. The algorithm containing these expressions is introduced below.

Since the problem formulation is convex, \( z^0 \) can be initialized with any value.

**Algorithm 1** Localization with packet loss

**Input:** 
- \( m; \) \( \text{ // Strong convexity constant} \)
- \( L; \) \( \text{ // Lipschitz constant} \)
- \( z^0 = \{ x^0, y^0, w^0, \Omega^0, \Upsilon^0 \} \) \( \text{ // Initialization} \)
- \( d, r \) \( \text{ // Distance measurements} \)
- \( u, q \) \( \text{ // Angle measurements with distribution parameters} \)
- \( \Sigma_N, \Sigma_A, \Sigma_X, \Sigma_\Omega, \Sigma_\Upsilon \) \( \text{ // Distribution parameters matrices} \)
- \( A, E \) \( \text{ // Network matrices} \)
- \( B, D, K, N \) \( \text{ // Missing data matrices} \)
- \( x_{prior}, d_{prior}, r_{prior} \) \( \text{ // Prior information vectors} \)

**Output:** \( \hat{x} \)

1: for all \( t \) do
2: \( k = 1, z^1 = z^0 \)
3: \( \text{ while some stopping criterion is not met do} \)
4: \( \hat{z}^k = z^k + \frac{1}{\sqrt{m/L}} (z^k - z^{k-1}) \)
5: \( x^{k+1} = (I - \frac{1}{L} T) \hat{x}^k + \frac{1}{L} A^T \Sigma_A B \hat{y}^k + \frac{1}{L} E^T D \Sigma_A (\hat{w}^k + \alpha) \)
6: \( y^{k+1} = P_Y \left( \frac{1}{L} \Sigma_N^2 B A \hat{x}^k + (I + \frac{1}{L} \Sigma_N^2 D^2) \hat{y}^k + \frac{1}{L} B u \right) \)
7: \( w^{k+1} = P_W \left( \frac{1}{L} \Sigma_A^2 D E \hat{x}^k + (I - \frac{1}{L} \Sigma_A^2 D^2) \hat{w}^k + \frac{1}{L} D q - \frac{1}{L} \Sigma_A^2 D \alpha \right) \)
8: \( \Omega^{k+1} = (I - \frac{1}{L} \Sigma_\Omega K^T K) \hat{\Omega}^k + \frac{1}{L} \Sigma_\Omega K^T K d_{prior} \)
9: \( \Upsilon^{k+1} = (I - \frac{1}{L} \Sigma_\Upsilon N^T N) \hat{\Upsilon}^k + \frac{1}{L} \Sigma_\Upsilon N^T N r_{prior} \)
10: \( z^{k+1} = (x^{k+1}, y^{k+1}, w^{k+1}, \Omega^{k+1}, \Upsilon^{k+1}) \)
11: \( k = k + 1 \)
12: \( \text{ end while} \)
13: \( \text{ return } \hat{x} \)
14: \( \text{ end for} \)
Chapter 5

Results

5.1 Considerations on simulation setup

In order to solve the problem, we resorted to CVX for Matlab, a package developed to solve convex optimization problems [38]. The implementation of the Algorithm 1 is work in progress.

The MAP estimator was tested by implementing the optimization problem (4.12) using CVX. It would be equivalent to implement problems (4.13) or (4.14). Our estimator was tested for three 2D trajectories: a linear trajectory and a spiral trajectory of formation vehicles, and a linear trajectory where the vehicles do not travel in formation. Then, the method was also tested for a 3D helix trajectory. These trajectories are presented in Figures 5.1–5.4 and they allow performance evaluation for linear and circular motions.

Figure 5.1: Linear trajectory with three nodes, in blue line, traveling inside of the convex hull of three anchors, in black lines. Both nodes and anchors describe similar trajectories. The start of each trajectory is marked with a star.

Figure 5.2: Spiral trajectory with three nodes, in blue line, traveling inside of the convex hull of three anchors, in black lines. Both nodes and anchors describe similar trajectories. The start of each trajectory is marked with a star.
The linear and spiral trajectories include three nodes and three anchors traveling in formation, where the nodes are inside the convex hull of the anchors. The convex hull is defined by the smallest convex set which contains the positions of the three anchors.

First, the algorithm is tested in the linear and spiral trajectories with noiseless measurements and then the noise is introduced in all measurements. Consequently, all ranges are contaminated by Gaussian noise with zero mean and constant standard deviations, from now on denoted as $\sigma_d$ (for both node-node and node-anchor distance measurements). Bearings are contaminated by vMF noise with zero mean direction and constant concentration parameters, from now on denoted as $\kappa_a$ (for both node-node and node-anchor angle measurements). The standard deviation $\sigma_d$ and concentration parameter $\kappa_a$ are assumed to be known a priori since, in real applications, these parameters can be estimated from previous experiments.

In the linear trajectory, nodes and anchors are considered to be traveling at a constant speed of 1.5 m/s, as this is a regular operation speed for AUVs. In the case of the spiral trajectory, nodes and anchors are traveling at constant angular velocity, where the speed changes in magnitude and in direction at every time step. This is useful to study the influence of variation of speed in our method.

### 5.2 Results without Missing Measurements

The results without missing measurements are the solution to the problem in (3.40) of the MLE formulation. This is equivalent to the problem reformulated in (4.12), only considering distance terms between nodes, distance terms between nodes and anchors and angle terms.
5.2.1 Results with noiseless measurements

The MLE was tested for noiseless measurements, first with only ranges and then with both ranges and bearing. The estimated linear trajectories obtained with CVX are represented in Figure 5.5 with distance measurements only and in 5.6 with distance and angle measurements.

![Figure 5.5: Estimated positions with noiseless distance measurements in a linear trajectory.](image)

![Figure 5.6: Estimated positions with noiseless range and bearing measurements in a linear trajectory.](image)

In Figure 5.5, it is observed a deviation at the ends of the estimated trajectories of the nodes. This deviation occurs only at the ends considering that this is near the boundary of the convex hull formed by the trajectories of the anchors. With the introduction of angle measurements, this deviation is corrected,
since bearings give information about the direction of the nodes at each time instant where this type of measurement is available. This correction is observed in Figure 5.6, where the ends of the estimated trajectories of the nodes do not present any deviation.

The estimated spiral trajectories obtained with CVX are represented in figure 5.7 with distance measurements only and in 5.8 with distance and angle measurements.

![Figure 5.7: Estimated positions with noiseless distance measurements in a spiral trajectory.](image)

![Figure 5.8: Estimated positions with noiseless distance and angle measurements in a spiral trajectory.](image)

Similarly to the linear trajectory, the result of the estimated positions with only distance measurements presents a deviation at the ends of the trajectory. In this case, the ends are in the boundary of the unfolded convex hull (as if the spiral trajectory is unfolded into a linear one) of the trajectory of the
anchor.

The Mean Absolute Error corresponding to the estimation problem with only distance measurements and with distance and angle measurements are listed in Table 5.1.

<table>
<thead>
<tr>
<th></th>
<th>Linear Trajectory MAE [m]</th>
<th>Spiral Trajectory MAE [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only distance measurements</td>
<td>0.101</td>
<td>0.299</td>
</tr>
<tr>
<td>Distance and angle measurements</td>
<td>$3.07 \times 10^{-5}$</td>
<td>$2.13 \times 10^{-7}$</td>
</tr>
</tbody>
</table>

As expected, with the introduction of the angle measurements the mean absolute error of the trajectories decreases, as the deviation at the ends are corrected. The results of the MLE obtained with noiseless measurements are the ones with the lowest error possible since there are not missing measurements or noise contaminating the measurements.

5.2.2 Results with noisy measurements

In a realistic scenario, there are undesired random disturbances in the signals transmitted between sensors. Then, after obtaining the estimation results with noiseless measurements, noise was introduced in all of the measurements available, for both ranges and bearings. The noise values added to each value are IID, meaning that they were drawn independently from the same PDF with zero mean and a constant variance $\sigma^2$. The estimated positions of the nodes with noisy measurements obtained with CVX are presented in Figure 5.9 for a linear trajectory and in Figure 5.10 for a spiral trajectory. The choice of noise parameters was based on standard deviations usually achieved with current instrumentation, in particular $\sigma_d = 0.5 \, m$ and $\kappa_a = 6000$. The respective MPE for each trajectory averaged over 50 MC trials are found in Table 5.2.
Figure 5.9: Estimated positions in a linear trajectory with distance and angle measurements, subjected to noise with $\sigma_d = 0.5 \text{ m}$, $\kappa_a = 6000$.

Figure 5.10: Estimated positions in a spiral trajectory with distance and angle measurements, subjected to noise with $\sigma_d = 0.5 \text{ m}$, $\kappa_a = 6000$.

Table 5.2: Mean positioning error of MLE with noisy measurements averaged over 50 MC trials. Measurements are subject to noise with $\sigma_d = 0.5 \text{ m}$ and $\kappa = 6000$.

<table>
<thead>
<tr>
<th>MPE [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear trajectory</td>
</tr>
<tr>
<td>Spiral trajectory</td>
</tr>
</tbody>
</table>
As expected, in the presence of noise contaminating distance and angle measurements, the estimation error increases in both trajectories and the MPE is very similar in both trajectories, having a difference of 0.006 m.

5.3 Results with Missing Measurements

The next logical step in testing our algorithm is the introduction of random packet losses. These are stochastic events and were simulated taking into account the percentage of missing measurements desired. In short, the missing measurements in every simulation were generated resorting to random permutations, where a sequence of random numbers with the size of the total missing values is created from an array containing all the edges (node-node and node-anchor) available throughout the trajectory.

Our method is limited by the assumption that two consecutive missing measurements cannot occur. In a real scenario, this could happen, however in large data sets with thousands of data entries, the probability of occurring two consecutive packet losses is rather low. Due to this assumption, the maximum percentage of missing measurements that can occur is about 50%.

Given the considerations mentioned above, the simulations were set, first for the linear trajectory with a small percentage of missing measurements of about 5%, which is represented in Figure 5.11 and then for the maximum percentage possible of 50%, which is represented in Figure 5.12. The simulations with the same percentages of packet losses were conducted for the spiral trajectory and the results are illustrated in Figure 5.13 with 5% missing data and in Figure 5.14 with 50% missing data.

Figure 5.11: Estimated positions in a linear trajectory with 5% missing measurements, subjected to noise with $\sigma_d = 0.5$ m, $\kappa_a = 6000$. 

56
Figure 5.12: Estimated positions in a linear trajectory with 50% missing measurements, subjected to noise with $\sigma_d = 0.5 \text{ m}$, $\kappa_a = 6000$.

Figure 5.13: Estimated positions in a spiral trajectory with 5% missing measurements, subjected to noise with $\sigma_d = 0.5 \text{ m}$, $\kappa_a = 6000$. 
Figure 5.14: Estimated positions in a spiral trajectory with 50% missing measurements, subjected to noise with $\sigma_d = 0.5$ m, $\kappa_a = 6000$.

The MPE for the linear and spiral trajectories with different percentages of missing data is presented in Table 5.3.

Table 5.3: Mean positioning error of MAP estimator with missing measurements averaged over 50 MC trials. Measurements are subject to noise with $\sigma_d = 0.5$ m and $\kappa = 6000$.

<table>
<thead>
<tr>
<th></th>
<th>MPE Linear Trajectory [m]</th>
<th>MPE Spiral Trajectory [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>5% packet loss</td>
<td>0.311</td>
<td>0.562</td>
</tr>
<tr>
<td>50% packet loss</td>
<td>1.139</td>
<td>3.107</td>
</tr>
</tbody>
</table>

When the percentage of missing data is higher, the error is intuitively expected to increase compared to a small percentage. This can be observed in the results obtained, where the MPE is larger when the percentage of missing data is higher. Between both trajectories, the MPE is larger in the spiral trajectory since our method does not take velocity changes into account.

In each simulation, not only the position of the nodes is estimated, but also the variables $\Omega$ and $\Upsilon$ are estimated, which correspond to the distance between two nodes and the distance between one node and one anchor, respectively. The RMSE of the estimated $\Omega$ and $\Upsilon$ for both trajectories and for 5% and 50% of missing measurements are depicted in Table 5.4.

Table 5.4: RMSE of the distances estimated $\Omega$ and $\Upsilon$ averaged over 50 MC trials.

<table>
<thead>
<tr>
<th></th>
<th>RMSE Linear Trajectory [m]</th>
<th>RMSE Spiral Trajectory [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>5% packet loss</td>
<td>0.696</td>
<td>0.708</td>
</tr>
<tr>
<td>50% packet loss</td>
<td>0.701</td>
<td>0.716</td>
</tr>
</tbody>
</table>
The RMSE is computed by taking the average of the squared differences between each predicted value and its true value \[39\]. It is simply the square root of the MSE and corresponds to the standard deviation of the predicted values. The results obtained for the RMSE of the distances are consistent with the previous results, the RMSE is larger for a higher percentage of missing data and it is also larger in the spiral trajectory than in the linear trajectory.

The estimation results were obtained having in consideration the parameters of the prior terms for each trajectory. The prior parameter of position is \(\beta_i\), which defines the maximum distance that a node can travel between two-time instants and from now on will be designated simply by \(\beta\). The prior parameters of distance are \(\rho_{ij}\) for a node-node edge and \(\delta_{ik}\) for a node-anchor edge and from now they will be both designated by \(\rho\). This parameter defines the maximum value for the distance between two vehicles when the respective measurement is missing.

The parameters \(\beta\) and \(\rho\) were chosen taking into account the type of trajectory and the velocity of the vehicles. The vehicles in the linear trajectory travel at a constant speed of 1.5 m/s, and in a time window of size \(T_0 = 50\) were the time sample corresponds to 1.36 s, the maximum that a vehicle can travel is 2.04 m between time instants. Therefore, the value chosen for this trajectory is \(\beta = 2.04\) m.

The case of the spiral trajectory is slightly complicated since the vehicles are in a circular motion and travel at constant angular speed. At each time instant, the velocity changes in direction and magnitude and our method does not handle velocity changes. Nonetheless, the prior parameter \(\beta\) can be computed taking into account that information. The approach used consists in calculating the maximum distance traveled between time instants. This is the best-suited approach for simulations where the optimization problem is solved for the whole time window at once. The value chosen for the spiral trajectory was \(\beta = 10\) m having in consideration the maximum velocity attained by the vehicles in the spiral trajectory.

In the case where the optimization problem was solved for each time step at a time, \(\beta\) could be computed differently. At a time \(t - 1\), we have access to the actual estimated position and the previously estimated position at \(t - 2\) and based on that, the distance which the vehicle would travel next to \(t\) is the difference between the actual and the previous positions, meaning

\[
\beta_t = ||x_i(t - 1) - x_i(t - 2)||. \tag{5.1}
\]

This would give a more accurate result since \(\beta\) would be closer to the actual value.

In the linear trajectory, the vehicles travel in formation and the change in distance is small (it depends on the noise). Therefore the change in the distance measurement between time distance will be on average, equal to the standard deviation of the noise introduced in these measurements. In the spiral trajectory, the vehicles travel at constant angular speed and the distance measurement will change in time. However the distance variation is around 1 m, which is considerably small in the given trajectory, so the value chosen for \(\rho\) is kept from the linear trajectory. Therefore, \(\rho = 0.5\) m for both trajectories.

Following the same reasoning as the position parameter \(\beta\), the distance parameter at each instant could be computed based on the known distance measurements at times \(t - 1\) and \(t - 2\). The distance between vehicles at time \(t\) is the difference of the distances between those same vehicles at \(t - 1\) and
This would give a more accurate result of the parameter \( \rho \), since it would be closer to the actual value.

To exemplify this reasoning, a new trajectory is presented, where the vehicles do not travel in formation. For each time instant, the position and distance parameters are computed using (5.1) and (5.2), respectively. The estimated positions and the true positions of the nodes are presented in 5.16 for the case with 5\% missing data and 5.17 for the case with 50\% missing data. The estimation results without missing data in this trajectory are also presented in Figure 5.15, to compare the results obtained with and without packet losses.

In this trajectory, the nodes travel at different speeds. The 2 anchors travel at a constant speed of 1.5 m/s, as well as node 2 (the one with the downward trajectory). Node 1 (the one with the upward trajectory) travels at a constant speed of 1.2 m/s. The MPE of the estimated positions and the RMSE of the estimated distances \( \Omega \) and \( \Upsilon \) for different percentages of missing measurements are depicted in Table 5.5.

![Figure 5.15: Estimated positions in a linear trajectory where the vehicles do not travel in formation, without missing measurements and subjected to noise with \( \sigma_d = 0.5 \) m and \( \kappa_a = 6000 \).](image)
Figure 5.16: Estimated positions in a linear trajectory where the vehicles do not travel in formation, with 5% missing measurements and subjected to noise with $\sigma_d = 0.5$ m and $\kappa_a = 6000$.

Figure 5.17: Estimated positions in a linear trajectory where the vehicles do not travel in formation, with 50% missing measurements and subjected to noise with $\sigma_d = 0.5$ m and $\kappa_a = 6000$. 
Table 5.5: MPE of the estimated positions and RMSE of the estimated distances $\Omega$ and $\Upsilon$ when vehicles do not travel in formation, averaged over 50 MC trials.

<table>
<thead>
<tr>
<th>Packet Loss</th>
<th>MPE [m]</th>
<th>RMSE [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>0.332</td>
<td></td>
</tr>
<tr>
<td>5%</td>
<td>0.546</td>
<td>0.862</td>
</tr>
<tr>
<td>50%</td>
<td>1.223</td>
<td>0.9534</td>
</tr>
</tbody>
</table>

Our method has a good performance in this trajectory, where the nodes do not travel in formation and at different velocities. Although the performance in a trajectory where the vehicles travel in formation in a linear trajectory is better, the maximum MPE obtained is 1.223 m. Nonetheless, the maximum error is smaller than the MPE in the spiral trajectory because our MAP estimator does not handle velocity changes in direction and magnitude. Considering that an AUV has a length of approximately 5 m, our estimator has a good performance.

In the spiral and linear trajectories, the nodes travel inside of the convex hull of the anchors. In this new trajectory and layout of the network, we prove that our method does not require the anchors to be deployed at the boundary of the network. This occurs due to the introduction of bearing terms, that give information about the direction of the nodes. In addition, analyzing the estimated positions shown in Figures 5.15-5.17, it is possible to observe that our method is robust to packet losses. This is also observed in the results obtained in the linear and spiral trajectories.

Lastly, to test our method, we present a 3D trajectory where the vehicles travel in formation, in a circular motion with speed constant in magnitude but changing direction. The estimated positions and the true positions of the nodes obtained with CVX are presented in Figures 5.18 for the case without missing data, in 5.19 for the case with 5% missing data and 5.20 for the case with 50% missing data.

![Figure 5.18: Estimated positions in a helix trajectory, without missing measurements and subjected to noise with $\sigma_d = 0.5m$ and $\kappa_a = 6000$.](image)
The estimator parameters $\beta$ and $\rho$ were chosen similarly to the linear and spiral trajectories, where the vehicles travel in formation. Therefore, the chosen values were $\beta_i = 1 \text{ m}$ and $\rho_{ij} = 0.5 \text{ m}$.

Figure 5.19: Estimated positions in a helix trajectory, with 5% missing measurements and subjected to noise with $\sigma_d = 0.5 \text{ m}$ and $\kappa_a = 6000$.

Figure 5.20: Estimated positions in a helix trajectory, with 50% missing measurements and subjected to noise with $\sigma_d = 0.5 \text{ m}$ and $\kappa_a = 6000$.

The MPE of the estimated positions and the RMSE of the estimated distances $\Omega$ and $\Upsilon$ for different percentages of missing measurements are depicted in Table 5.5.
Table 5.6: MPE of the estimated positions and RMSE of the estimated distances $\Omega$ and $\Upsilon$ in the helix trajectory, averaged over 50 MC trials.

<table>
<thead>
<tr>
<th>Packet Loss</th>
<th>MPE [m]</th>
<th>RMSE [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>0.191</td>
<td></td>
</tr>
<tr>
<td>5%</td>
<td>0.239</td>
<td>0.703</td>
</tr>
<tr>
<td>50%</td>
<td>0.473</td>
<td>0.712</td>
</tr>
</tbody>
</table>

The MPE in this trajectory is lower than the MPE in all the other trajectories, for all percentages of missing data. In this path, the agents travel a bigger distance than in the other trajectories and the total number of points available for estimation is also higher. Also, in this case, the nodes communicate about their relative positions at shorter distances, meaning that they travel approximately 1 m and receive and transmit new data packets, obtaining more accurate results. In the other trajectories, the nodes communicate after traveling at least 2 m at a time. The RMSE (or standard deviation) of the missing variables $\Omega$ and $\Upsilon$ is similar to the other trajectories. These results confirm that our method has a good performance, not only in a two-dimensional space but also in a three-dimensional space.

In summary, we tested the MAP estimator in four trajectories, two linear and two where the agents travel in circular motion. The overall estimation performance of our method is good. The worst case is when the vehicles travel in a circular motion with speed changes in magnitude and direction. The next step is to evaluate our algorithm regarding the influence of noise and do a more detailed study of the influence that the number of packet losses has in our estimator.

## 5.4 Algorithm Performance Evaluation

### 5.4.1 Influence of Noise in Performance

The influence of noise in the algorithm performance is analyzed by the impact of the variables $\sigma_d$ and $\kappa_a$ in the 2D linear and spiral trajectories. The MPE variation with distance noise is presented in Figure 5.21 and is compared for standard deviation values obtained with current instrumentation and for larger values above 1 m. Variation of the MPE with bearing noise is evidenced in Figure 5.22, with $\kappa$ converted to the equivalent standard deviation in degrees. In Table 5.7 is presented the equivalence between the concentration parameter and the equivalent standard deviation in degrees, given by $\sigma_a = 180/(\pi \sqrt{\kappa_a})$. The MPE of our method is not only compared for regular bearing noises below $3^\circ$, but also for a larger value of more than $5^\circ$.

<table>
<thead>
<tr>
<th>$\kappa$</th>
<th>9000</th>
<th>6000</th>
<th>3000</th>
<th>1000</th>
<th>500</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_{eq}[^\circ]$</td>
<td>0.60</td>
<td>0.74</td>
<td>1.05</td>
<td>1.81</td>
<td>2.56</td>
<td>5.73</td>
</tr>
</tbody>
</table>

Table 5.7: Equivalence between concentration parameter and equivalent standard deviation.
Figure 5.21: Variation of MPE with distance standard deviation over linear and spiral trajectories. The results correspond to the MPE averaged over 50 MC trials for each marked value of standard deviation.

Figure 5.22: Variation of MPE with angle standard deviation over linear and spiral trajectories. The results correspond to the MPE averaged over 50 MC trials for each marked value of the equivalent standard deviation.

It is observed that the MPE increases with higher values of noise, as expected. The estimation error is higher in the spiral trajectory than in the linear trajectory, due to the velocities changes which are not taken into account, as mentioned before. The error increases almost in a linear form and at the same pace for both trajectories.

In the linear trajectory, the MPE increases 0.479 m when the distance standard deviation varies from 0 m to 1.5 m and it increases 0.297 m when the angle standard deviation varies from 0° to 5.73°. In the spiral trajectory, the MPE increases 0.425 m when the distance standard deviation varies from 0 m to 1.5 m and it increases 0.3412 m when the angle standard deviation varies from 0° to 5.73°. These variations in percentage form can be observed in Tables 5.8 and 5.9 for distance noise and angle noise, respectively.

<table>
<thead>
<tr>
<th>Table 5.8: Variation of the Mean Positioning Error with distance standard deviation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPE(0m) [m]</td>
</tr>
<tr>
<td>Linear trajectory</td>
</tr>
<tr>
<td>Spiral trajectory</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 5.9: Variation of the Mean Positioning Error with angle standard deviation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPE(0°) [m]</td>
</tr>
<tr>
<td>Linear trajectory</td>
</tr>
<tr>
<td>Spiral trajectory</td>
</tr>
</tbody>
</table>
5.4.2 Influence of Missing Data in Performance

The algorithm is then tested for different amounts of missing data. The percentage of missing measurements was varied between 0% and 50%, which is the maximum percentage of missing data that our method can handle due to its limitations. In Figure 5.23 is presented the behavior of the MPE with the variation of packet losses in the 2D linear and spiral trajectories.

![Graph showing variation of MPE with percentage of missing measurements over linear and spiral trajectories.](image)

Figure 5.23: Variation of MPE with the percentage of missing measurements over linear and spiral trajectories. The results correspond to the MPE averaged over 50 MC trials for each marked value of missing data.

The MPE increases with the percentage of missing data, which is intuitively expected. The lack of measurements leads to an increase in the estimation error. Once again, as observed, the results obtained are consistent with the previous results, the MPE is higher for the spiral trajectory than for the linear trajectory. The increase of the estimation error is almost linear in both trajectories and it increases at a higher rate in the spiral trajectory. The variation of the MPE of the linear trajectory where the vehicles do not travel in formation lies between the MPE of the linear and spiral trajectories.

The variation of the MPE in percentage between packet losses of 5% and 50% is presented in Table 5.10. The initial error MPE(0%) without missing measurements corresponds to the results of the MLE formulation.

<table>
<thead>
<tr>
<th>MPE(0%) [m]</th>
<th>∆MPE_{0%\rightarrow50%} [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear trajectory</td>
<td>0.236</td>
</tr>
<tr>
<td>Spiral trajectory</td>
<td>0.230</td>
</tr>
</tbody>
</table>

Table 5.10: Variation of the Mean Positioning Error with the percentage of missing measurements.
5.5 Comparison with state-of-the-art method

To assess the quality of the method developed in this work, it was compared with a variance-constrained approach [3], which is described in Appendix A. Our algorithm and the state-of-the-art method are evaluated for the same network and the same trajectories, which correspond to the linear and spiral trajectory in study in this chapter.

Firstly, this method was tested for the linear trajectory represented in Figure 5.1. In Figure 5.24 is depicted the Mean Navigation Error comparing our method and the state-of-the-art method. In Figure 5.25 is represented the Empirical Cumulative Distribution Function (CDF) of our method compared to the state-of-the-art method. The Empirical CDF is an estimate of the cumulative distribution function that generated the data observed.

![Figure 5.24: Mean navigation error throughout the linear trajectory with 5% missing data, averaged through 100 MC trials. The vertical axis is in logarithmic scale.](image1)

![Figure 5.25: Empirical CDF for 100 MC trials on the linear trajectory with 5% missing data. The horizontal axis is in logarithmic scale.](image2)
Since the Empirical CDFs resemble vertical lines due to the error gap, the Empirical CDF of our method is presented in Figure 5.26, and is compared with the CDF, which is a hypothetical model of the distribution.

![Empirical CDF and hypothetical CDF](image)

**Figure 5.26:** Empirical CDF and hypothetical CDF of the normal distribution of the error for 100 MC trials on the linear trajectory with 5% missing data.

It is evident that our method has a better performance than the state-of-the-art method in a linear trajectory. In Figure 5.24, the Mean Navigation Error obtained with this method is one order of magnitude higher than the MNE obtained with our method. In Figure 5.25, the error is averaged per vehicle and per number of trajectory points and the Empirical CDF approximates the CDF of the normal distribution, which is the distribution of the random variables in our problem. It is observed that our method outperforms the state-of-the-art method. The Mean Positioning Error and the Root Mean Square Error of the position obtained for this trajectory for both methods are presented in Table 5.11.

Table 5.11: MPE and RMSE of MAP estimator and state-of-the-art method throughout the linear trajectory, with 5% missing measurements averaged over 100 MC trials. Measurements are subject to noise with $\sigma_d = 0.5$ m and $\kappa = 6000$.

<table>
<thead>
<tr>
<th></th>
<th>MPE Linear Trajectory [m]</th>
<th>RMSE Linear Trajectory [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method</td>
<td>0.311</td>
<td>0.372</td>
</tr>
<tr>
<td>state-of-the-art method</td>
<td>20.984</td>
<td>22.501</td>
</tr>
</tbody>
</table>

The results obtained confirm that the estimation error in the state-of-the-art method is one order of magnitude higher than our method, over 20 m.

The state-of-the-art method was then tested in the spiral trajectory represented in Figure 5.2. The MNE comparing our method and this method is depicted in Figure 5.27 and the Empirical CDF of each one is presented in Figure 5.28. The Mean Positioning Error and the Root Mean Square Error of the position obtained for the spiral trajectory for both methods are presented in Table 5.12.
Figure 5.27: Mean navigation error throughout the spiral trajectory with 5% missing data, averaged through 100 Monte Carlo trials. The vertical axis is in logarithmic scale.

Figure 5.28: Empirical CDF for 100 Monte Carlo trials on the spiral trajectory with 5% missing data. The horizontal axis is in logarithmic scale.

Table 5.12: MPE and RMSE of MAP estimator and state-of-the-art method throughout the spiral trajectory, with 5% missing measurements averaged over 100 MC trials. Measurements are subject to noise with $\sigma_d = 0.5$ m and $\kappa = 6000$.

<table>
<thead>
<tr>
<th></th>
<th>MPE Spiral Trajectory [m]</th>
<th>RMSE Spiral Trajectory [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method</td>
<td>0.562</td>
<td>0.806</td>
</tr>
<tr>
<td>state-of-the-art method</td>
<td>41.756</td>
<td>49.440</td>
</tr>
</tbody>
</table>

As in the linear trajectory, in this trajectory, the Empirical CDFs resemble vertical lines due to the error gap. The Empirical CDF of our method is presented in Figure 5.29 and is compared with the CDF.
Furthermore, like in the linear trajectory, our method outperforms by far the state-of-the-art method. The MNE of the state-of-the-art method is one order of magnitude higher than the MNE of our method, as well as the error averaged per vehicles and per number of trajectory points.
Chapter 6

Discussion and Future Work

The present work proposed one of the first approaches to vehicle tracking using a convex approximation to the MAP estimator, while in the presence of missing measurements. To achieve this, first we formulated an MLE problem to address the localization of a generic network, and then a novel MAP estimator was proposed to handle missing distance measurements. The first part of the work, the MLE, includes distance and angle measurements between nodes, as well as measurements between nodes and anchors. The range and bearing terms were subjected to an existing convex relaxation and it was defined that each bearing should have a correspondent distance measurement due to our relaxation, which caused these terms to be dependent on the formulation for the ranges. For the second part of the thesis, as previously mentioned, a new MAP estimator was proposed to handle missing distance measurements. This estimator is based on the MLE formulation, relying on the likelihood function, with the addition of the prior information of distance measurements and nodes' positions. Each data packet includes one distance measurement and one angle measurement, and when a packet is lost, both measurements are considered missing.

The method we present is distributed, convex and does not require initialization. The nodes do not need to be inside of the convex hull of the anchors, due to the introduction of angular information with an appropriate relaxation, which provides greater flexibility to the analysis. Since it is assumed that are not two consecutive missing measurements on the same edge, there is a theoretical maximum percentage of missing data that the MAP estimator can handle, namely 50\%. The increase in range and bearing noise does not seem to have a major influence on accuracy, so much so that such measurements do not need to be taken with high accuracy to obtain a good performance. Even though an increase of missing measurements has a major influence on accuracy, the results obtained for high percentages of missing data are satisfactory, proving that this method is reliable even for adverse operational scenarios.

When our method is applied to formation vehicles in two different trajectories, one linear with constant speed and one spiral with constant angular speed, the comparison shows that it presents better performance in the linear trajectory case. This is explained by the fact that in the spiral trajectory the velocity changes every instant, both in direction and magnitude, and our method was not built to handle changes in speed. Then, comparing our method applied to a trajectory where the vehicles do not
travel in formation and at different speeds, the estimator in the linear formation trajectory still has better performance, but the results are better in this new trajectory than in the spiral trajectory.

Despite all the promising results shown through testing, the method still presents some shortcomings that can serve as a basis for future work. For example, its focus is solely on the estimation of missing distances, due to its greater importance for vehicle tracking, but it will likely benefit from the inclusion of angle measurements. However, this might be a great challenge since the angle terms become non-convex when the measurement is missing, resulting in both range and bearings being considered missing, and discarded from the respective terms in the MAP estimator when a packet is lost, which is far from the ideal scenario. In a future approach, however, this problem could be addressed and an estimator that includes the missing bearing terms might come to existence. It is important, however, to note how the next immediate step should be to address the implementation issue preventing results with the distributed solution.

Another topic that might deserve some attention in the future is the achievement of high fidelity results with a missing data percentage higher than 50%, even though there is a limit to this value since a minimum amount of data to predict the location of the vehicles is always needed, and a velocity component could be fitted in the model, expanding its range of applications. Finally, and given that in the performed simulations it was considered that the algorithm has access to true noise parameters a priori (which in a more realistic scenario would not be a possibility), these parameters could be estimated from the received measurements along with the predicted positions, a very enticing topic for future developments.

In summary, this work resulted in the development of a new method for vehicle tracking in the presence of missing data. The objective was fulfilled, and when compared with a variance-constrained approach to recursive state estimation with missing measurements, one of the very few available options to handle the missing data problem in state estimation, the present algorithm reveals a level of accuracy over one order of magnitude higher than such method, making it one of the most reliable and overall best options for the task at hand.


Appendix A

State-of-the-art method [3]

A complex network with N coupled nodes is composed by a discrete time-varying stochastic array with the form

\[ x_i[k+1] = f(x_i[k]) + \sum_{j=1}^{N} w_{ij} \Gamma x_j[k] + B_i[k]v_i[k] \]  
\[ y_i[k] = \lambda_i[k]C_i[k]x_i[k] + \nu_i[k], \]  

(A.1)

(A.2)

where \( x_i[k] \in \mathbb{R}^n \) is the state vector of the \( i^{th} \) node. \( f \) is a smooth and continuously differentiable nonlinear function describing the dynamic model of the system. \( y_i[k] \in \mathbb{R}^m \) is the measurement output from the \( i^{th} \) node of the complex network. \( v_i[k] \) is additive noise with zero-mean and covariance \( Q_i[k] > 0 \) and \( \nu_i[k] \) is additive noise with zero-mean and covariance \( R_i[k] > 0 \). \( \lambda_i[k] \in \mathbb{R} \) is a random variable which characterize the probabilistic missing phenomena. \( B_i[k] \) and \( C_i[k] \) are known and bounded matrices with appropriate dimensions. \( \Gamma = \text{diag}(\gamma_1, \ldots, \gamma_n) \in \mathbb{R}^{n \times n} \) is the inner-coupling matrix of the network, which describes the way of linking the components in every pair vector of nodes. Matrix \( W = w_{ij} \in \mathbb{R}^{N \times N} \) is the constant coupling configuration matrix which represents the topological structure of the network.

The state estimator developed is given by:

\[ \hat{x}_i[k+1] = f(\hat{x}_i[k]) + \sum_{j=1}^{N} w_{ij} \Gamma \hat{x}_j[k] \]  
\[ \hat{x}_i[k+1] = \hat{x}_i[k+1] + K_i[k+1] (y_i[k+1] - \mu_i[k+1]C_i[k+1]\hat{x}_i[k+1] + \nu_i[k+1]), \]  

(A.3)

(A.4)

where \( \hat{x}_i[k+1] \) is the estimate of \( x_i[k+1] \) at time \( \hat{x}_i[k+1] \). \( x_i[k+1] \) is the one-step prediction at time \( k \). \( y_i[k+1] \) is the actual measurement output from the \( i^{th} \) node of the complex network which is available for the estimator. \( \mu_i[k+1] \) is the mathematical expectation of random variable \( \lambda_i[k+1]. \)

\( K_i[k+1] \) is the estimator parameter to be determined.

The aim is to design a time-varying state estimator such that, in the presence of missing measurements and random disturbances, an upper bound of the estimation error covariance can be guaranteed and the explicit expression of the estimator parameter is given.
From the solutions of Riccati-like difference equations given by
\[
\Xi[k+1\mid k] = (A[k] + W \otimes \Gamma) \left( \Xi[k\mid k]^{-1} - \gamma[k] I \right)^{-1} (A[k] + W \otimes \Gamma)^T \\
+ \gamma[k]^{-1} L[k] L[k]^T + B[k] Q[k] B[k]^T
\] (A.5)
and
\[
\Xi[k+1\mid k+1] = (I - K[k+1] \tilde{\Lambda}[k+1] C[k+1]) \Xi[k+1\mid k] (I - K[k+1] \tilde{\Lambda}[k+1] C[k+1])^T \\
+ K[k+1] \left( \tilde{\Lambda} \circ (C[k+1] \Omega[k+1] C^T[k+1]) \right) K^T[k+1] \\
+ K[k+1] R[k+1] K^T[k+1],
\] (A.6)
an upper bound of the estimation error covariance \( P[k+1\mid k+1] \) is obtained, that is \( P[k+1\mid k+1] \leq \Xi[k+1\mid k+1] \).

The estimator parameter \( K[k+1] \) is designed to minimize the upper bound of the estimation error covariance. It is defined by
\[
K[k+1] = \text{diag}(K_1[k+1], \ldots, K_N[k+1]),
\] (A.7)
with
\[
K_i[k+1] = \Phi_{1,i} \Xi[k+1\mid k] C^T[k+1] \tilde{\Lambda}^T[k+1] \Phi_{2,i}^T \left( \Phi_{2,i} S[k+1] \Phi_{2,i}^T \right)^{-1},
\] (A.8)
where
\[
\Phi_{1,i} = \begin{bmatrix} 0_{n \times (i-1)} & I_{n \times n} & 0_{n \times (N-i)} \end{bmatrix},
\] (A.9)
\[
\Phi_{2,i} = \begin{bmatrix} 0_{m \times (i-1)} & I_{m \times m} & 0_{m \times (N-i)} \end{bmatrix},
\] (A.10)
\[
\Omega[k+1] = (1 + \epsilon[k+1]) \Xi[k+1\mid k] + (1 + \epsilon^{-1}[k+1]) \hat{x}[k+1\mid k] \hat{x}^T[k+1\mid k],
\] (A.11)
and
\[
S[k+1] = \tilde{\Lambda}[k+1] C[k+1] \Xi[k+1\mid k] C^T[k+1] \tilde{\Lambda}^T[k+1] + R[k+1] \\
+ \tilde{\Lambda}[k+1] \circ (C[k+1] \Omega[k+1] C^T[k+1]).
\] (A.12)

In order to obtain the state estimation of all the nodes of the network \( \hat{x} \), the parameter \( K[k+1] \) needs to be determined at each time instant.

### A.0.1 Numerical Example

To illustrate the present estimation method, the results of the numerical example of [3] were reproduced. It is considered a time-varying stochastic complex network with four nodes. The parameters and matrices that characterize the network are presented in Section 4 of [3]. The simulation results are given in Figures A.1-A.4, where the actual states and their estimates are plotted.
Figure A.1: Actual states and state estimates of node 1.

Figure A.2: Actual states and state estimates of node 2.

Figure A.3: Actual states and state estimates of node 3.
From the simulation, it is possible to observe that the designed state estimator does not have a satisfactory tracking performance in the simultaneous presence of missing measurements and stochastic disturbances. In the paper, the only results presented are the plots with the results of the simulation. The estimation error is not analyzed. Therefore, the Mean Positioning Error and the Root Mean Square Error of this numerical example are depicted in Table A.1, averaged over 100 Monte Carlo trials.

Table A.1: MPE and RMSE of the numerical example in [3] averaged over 100 Monte Carlo trials.

<table>
<thead>
<tr>
<th>MPE [m]</th>
<th>RMSE [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.239</td>
<td>8.793</td>
</tr>
</tbody>
</table>

The value of the error is considerably high, given that the trajectories of the nodes are contained in a square window with the side around 3m. With the results obtain we confirm that the method in study does not have a satisfactory tracking performance.

### A.0.2 Implementation

In order to implement this method, the same dynamic behavior as considered in the MLE formulation (without considering missing measurements) has to be employed and adapted. The state estimator presented in this method resembles the Extended Kalman Filter with a variation in order to deal with missing measurements.

The state vector at time $k$ is defined by

$$
x_i[k] = \begin{bmatrix} r_{i,x}[k] \\
r_{i,y}[k] \end{bmatrix},
$$

where $r_{x}[k]$ and $r_{y}[k]$ define the position of node $i$ at time $k$. $x[k] = \{x_i[k]\}$ is the vertical concatenation of all node vectors.
The dynamic model of the system is given by

\[ x_i[k + 1] = x_i[k] + \dot{x}_i[k] \Delta t + \varepsilon, \]

(A.14)

therefore function \( f \) of the estimator is defined as

\[ f(\hat{x}_i[k + 1]) = \hat{x}_i[k] + \dot{\hat{x}}_i[k] \Delta t. \]

(A.15)

The observation model is an adaptation of the data models, given as

\[
\begin{align*}
    d_{ij}[k] &= ||x_i[k] - x_j[k]|| + \delta^d \\
    b_{il}[k] &= ||x_i[k] - a_l[k]|| + \delta^b \\
    w_{ij}[k] &= \frac{x_i[k] - x_j[k]}{||x_i[k] - x_j[k]||} + \delta^u \\
    q_{il}[k] &= \frac{x_i[k] - a_l[k]}{||x_i[k] - a_l[k]||} + \delta^q
\end{align*}
\]

(A.16)

which correspond to the output measurements.