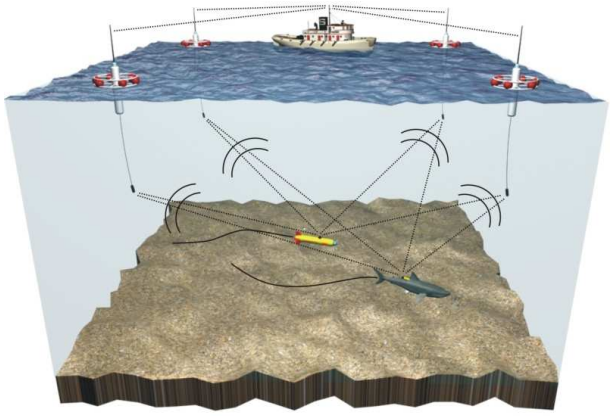
UNIVERSIDADE TÉCNICA DE LISBOA

INSTITUTO SUPERIOR TÉCNICO



Positioning and Navigation Systems for Robotic Underwater Vehicles

Alex Alcocer Peñas
(Licenciado)

Dissertação para a obtenção do Grau de Doutor
em Engenharia Electrotécnica e Computadores

Orientador:	Doutor Paulo Jorge Coelho Ramalho Oliveira
Co-Orientador:	Doutor António Manuel dos Santos Pascoal

Júri:

Presidente:	Doutor José António Marinho Brandão Faria
Vogais:	Doutor Jérôme Jouffroy
	Doutor António Manuel dos Santos Pascoal
	Doutor Aníbal Castilho Coimbra de Matos
	Doutor Fernando Duarte Nunes
	Doutor Paulo Jorge Coelho Ramalho Oliveira
	Doutor João Manuel de Freitas Xavier

Lisboa, Novembro de 2009

Abstract

This thesis addresses the problem of underwater navigation of robotic vehicles using acoustic positioning systems. Several estimation problems are considered that are based on Range-Only measurements obtained from the Times of Arrival of acoustic signals. First, the Range-Only localization problem is addressed, which consists of determining the position of a vehicle given ranges to a set of landmarks with known locations. This problem arises in acoustic positioning systems such as GIB (GPS Intelligent Buoys). Several solutions based on Least Squares, Maximum Likelihood, and Extended Kalman filtering are presented and applied to real experimental data obtained during sea trials. Special attention is given to performance issues and practical problems related to acoustic positioning systems such as sound speed estimation and multipath mitigation.

Second, the problem of pose estimation with Range-Only measurements is addressed, in which the vehicle is equipped with an array of beacons with known relative position and uses range measurements to a set of Earth fixed landmarks. A Maximum Likelihood estimator is derived that requires solving a constrained minimization problem on the Special Euclidean group $SE(3)$. Borrowing tools from optimization on Riemannian manifolds, generalized gradient and Newton methods are derived to solve this problem. An alternative solution is derived in a system-theoretic setting by adopting a suitable Lyapunov function that is a function of range measurements only, yielding convergence conditions.

Finally, the thesis addresses the post-processing of acoustic positioning data. An extension of diffusion-based trajectory observers is derived that incorporates measurement error information.

Key words: Underwater Navigation, Acoustic Positioning Systems, Range-Only measurements, Localization, Pose estimation, Maximum Likelihood estimation

Resumo

Esta tese aborda o problema da navegação de veículos robóticos submarinos usando sistemas de posicionamento acústicos. Consideram-se vários problemas de estimação baseados em medidas de distância, obtidas através da medição dos tempos de chegada de sinais acústicos. Em primeiro lugar aborda-se o problema da localização com medidas de distância, que consiste em determinar a posição de um veículo dadas medidas de distância a um conjunto de balizas com posição conhecida. Este problema pode ser encontrado em sistemas de posicionamento acústico tais como o GIB (GPS Intelligent Buoys). Varias soluções baseadas em mínimos quadrados, máxima verosimilhança e filtros estendidos de Kalman são apresentados e validados com dados experimentais obtidos durante testes de mar. É dada especial atenção às questões de desempenho e a problemas práticos relacionados com os sistemas de posicionamento acústicos, tais como a estimação da velocidade do som e a identificação de múltiplos ecos. Em segundo lugar, é abordado o problema da estimação conjunta de orientação e posição, usando medidas de distância. Neste caso, equipa-se o veículo com um conjunto de balizas com posições relativas conhecidas, e efectuam-se medidas de distância a um conjunto de balizas fixas no ambiente, com posição conhecida. Para este mesmo problema, propõe-se também um estimador de máxima verosimilhança, que requer a resolução de um problema de optimização com restrições no grupo especial Euclideano SE(3). Recorrendo a ferramentas de optimização em variedades Riemanianas, são derivados algoritmos generalizados de gradiente e Newton para resolver o problema em causa. Uma solução alternativa é derivada no contexto de teoria de sistemas, ao considerar uma função de Lyapunov, que depende só das medidas de distância, e permite obter resultados de convergência. Finalmente, a tese aborda o pós-processamento dos dados de um sistema de posicionamento acústico. Para esse fim, usa-se uma extensão dos observadores de trajectória baseados em processos de difusão, onde se incorpora de forma explícita informação sobre a incerteza nas medidas.

Palavras chave: Navegação submarina, sistemas de posicionamento submarinos, medidas de distância, localização, estimação de orientação e posição, estimação de máxima verosimilhança

Acknowledgements

I would like to thank a large number of people that in many different ways have helped to make the present document possible.

In the first place, I wish to thank the thesis supervisor Professor Paulo Oliveira for his constant help and support. He has always provided excellent scientific guidance as well as personal and financial support. I am also indebted to the thesis co-supervisor Professor Antonio Pascoal, who first welcomed me in Lisbon, and has been an invaluable support both from a scientific and a personal point of view. Thanks for sharing the enthusiasm in science and the values of friendship, honesty, and good research. I have to thank both supervisors for their efforts to keep the thesis on track and provide an excellent working environment together with countless exciting research and professional opportunities.

During all my stay at ISR I had the chance to assist to PhD courses and learn from several persons that have helped to improve the content and scope of the thesis. I am specially obliged to Professor João Xavier, who kindly agreed to participate in my thesis research and PhD commitee, for his numerous advices and inspiration. It was during one of his courses that the work from chapter 4 originated. I am obliged to Professor Pedro Aguiar and Professor Carlos Silvestre for providing good advices and many interesting discussions. I also wish to thank Professor Michael Athans. It was a great pleasure, and wonderful motivating beginning of my work at ISR, to be his student during a PhD course. My special gratitude goes also to Professors João Miranda Lemos, João Sentieiro, and Jorge Salvador Marques.

During the Spring of 2006 I had the chance to spend four wonderful months at NTNU, Trondheim, Norway. I am indebted to Professor Thor I. Fossen for making this visit possible and treating me like one of his students. During this time I had the chance to work closely with Professor Jérôme Jouffroy who kindly agreed to participate in my thesis research and later on the PhD thesis committee. It was a great pleasure and I have to deeply thank him for his advice and inspiration through countless interesting discussions. I want also to thank Jørgen, Bjørn, Andrew, Andrea, Micheal, Tristan, Morten, and Shen, for making a pleasant stay at NTNU.

My special gratitude to Professor Isaac Kaminer for his scientific and personal advices and many interesting discussions. I am also obliged to Doctor Elgar Desa and the NIO staff who warmly received me during a short visit in December 2008.

I also thank Professor Anibal Matos for having kindly agreed to participate in my thesis research and final PhD committee, for his numerous comments and suggestions. My appreciation goes also to Professor José António Brandão Faria and Professor Fernando Nunes for his participation in the PhD committee.

I would like to thank all the institutions that supported this work, and in special the FCT (Fundacão para a Ciência e a Tecnologia) for providing me with a PhD grant [1]. I would also like to thank the ISR administrative staff, Filomena, both Ana's, and specially to Loïc, who have many times helped me with the most diverse administrative tasks.

My special thanks to the DSOR lab friends, Luis, Joao, Rufino, Rita, Bruno C., Bruno G., with which I shared so many lunches, *lunches*, great laughs, and endless discussions on the most diverse topics.

I specially want to express my gratitude to a great number of persons, that with their friendship helped to make my stay in Lisbon more pleasant: Reza, Pramod, Charu, Dictino, Narcis, Vahid, Maryam, Karim, Andrea D., David, Andreas, Andre O., Behzad, Nelson, Jose Maria, Marco, Paulo R., Cabecinhas, Ricardo, Manel, Francisco, Rodolfo, Pedro Gomes, Guillerme, Paulino, Neveena, Ram Kiram, Lionel, Dejan, Sebastian, Danilo, Ettore, Eduarda, Rodrigo, Oli, Martha, Hesiodo, Torto, Andre G., Caseiro, Aguiam, Tiago, Luis S., Luis M., Bernardo, Paula and Miguel.

I am also obliged to my colleagues at Ictineu, Carme, Pere, and Gian Piero, for their support in the last stage of the thesis.

My deepest gratitude is to my parents Montserrat and Carmelo, and the rest of my family for their constant support without which this thesis would not have been possible.

A very special and sincere thanks goes to Gisela, for her unconditional support, for showing and accompanying me to the beautiful Gulbenkian library where an important part of the thesis was written. I know It has not been easy, and I am truly indebted to her.

At last, I would like to dedicate this thesis to my grandfather Esteban, who always motivated and supported me towards the pursue of knowledge.

[1] Research supported by project RUMOS of the FCT (PT), project GREX / CECIST (Contract No. 035223), Venus (Contract No. 034924),,and FREESUB (HPRN-CT-2000-00032) of the EC, and the FCT-ISR/IST plurianual funding program (through the POS-Conhecimento Program initiative, in cooperation with FEDER).

Contents

Nomenclature

List of Symbols

$\{\mathcal{I}\}$	Inertial Reference Frame
$\{\mathcal{B}\}$	Body fixed Reference Frame
$\mathbb{R}$	Set of real numbers
$\mathbb{N}$	Set of positive integers
$\mathbb{R}^n$	Set of column vectors with real entries of dimension n
$\mathbb{R}^{n\times m}$	Set of matrices with real entries of dimension $n \times m$
$\|\cdot\|$	Euclidean norm
$\|\cdot\|_F$	Frobenious matrix norm
$SO(3)$	Special Orthogonal group of dimension 3
$SE(3) = SO(3) \times \mathbb{R}^n$	Special Euclidean group of dimension 3
$\mathbf{A}$	Capital case boldface symbols represent Matrices
$\mathbf{a}$	Lower case boldface symbols represent column vectors
$\mathbf{p} \in \mathbb{R}^n$	Vehicle position
$\mathcal{R} \in SO(3)$	Vehicle attitude
$(\mathcal{R}, \mathbf{p}) \in SE(3)$	Vehicle pose
$\mathbf{I}_n$	Identity matrix of dimension $n \times n$
$\mathbf{1}_n$	Column vector of ones of dimension $n \times 1$
$\mathbf{0}_{n\times m}$	Matrix of zeros dimension $n \times m$
$\det(\cdot)$	Matrix determinant
$\delta(\mathbf{A})$	Column vector containing the diagonal of square matrix $\mathbf{A}$
$\delta(\mathbf{a})$	Square matrix with $\mathbf{a}$ on its diagonal and zeros elsewhere
$\mathbf{A}^T$	Matrix transpose
$\mathbf{A}^{-1}$	Matrix inverse
$\mathbf{A}^\dagger$	Matrix pseudoinverse
$\mathrm{vec}(\cdot)$	Matrix vectorization operator
$\mathrm{tr}\,(\cdot)$	Matrix trace operator
$\otimes$	Kronecker product of matrices
$\odot$	Hadamard element-wise product of matrices
$\mathrm{E}\{\cdot\}$	Expectation
$n \in \{2, 3\}$	Dimension of estimation problem

$m \in \mathbb{N}$	Number of landmarks
$p \in \mathbb{N}$	Number of beacons
$r \in \mathbb{R}$	Range
$\bar{r} \in \mathbb{R}$	Measured range
$w \in \mathbb{R}$	Range measurement error
$d \in \mathbb{R}$	Squared range
$\bar{d} \in \mathbb{R}$	Measured squared range
$\xi \in \mathbb{R}$	Squared range measurement error
$\mathbf{r}$	Vector of ranges
$\bar{\mathbf{r}}$	Vector of measured ranges
$\mathbf{w}$	Vector of range measurement errors
$\mathbf{d}$	Vector of squared ranges
$\bar{\mathbf{d}}$	Vector of measured squared ranges
$\boldsymbol{\xi}$	Vector of squared range measurement errors
$\mathbf{R}$	Range measurement error covariance matrix
$\boldsymbol{\Sigma}_{\boldsymbol{\xi}}$	Squared ranges measurement error covariance
$\mathbf{p}_i \in \mathbb{R}^n$	Landmark coordinate
$\mathbf{P} \in \mathbb{R}^{n \times m}$	Matrix of landmark coordinates
$\mathbf{b}_i \in \mathbb{R}^p$	Beacon coordinate
$\mathbf{B} \in \mathbb{R}^{n \times p}$	Matrix of beacon coordinates
$\mathbf{M} \in \mathbb{R}^{m \times m}$	Centering projection matrix
$\mathbf{P}_c = \mathbf{PM} \in \mathbb{R}^{n \times m}$	Matrix of centered landmark coordinates

List of Acronyms

TOA	Times of Arrival
TDOA	Time Differences of Arrival
RD's	Range Differences
GPS	Global Positioning System
INS	Inertial Navigation System
DR	Dead Reckoning
DVL	Doppler Velocity Log
LBL	Long Baseline
SBL	Short Baseline
USBL	Ultra Short Short Baseline
GIB	GPS Intelligent Buoys
CRB	Cramer Rao Bound
IVLB	Intrinsic Variance Lower Bound
ISS	Input to State Stability
ML	Maximum Likelihood

LS	Least Squares
EKF	Extended Kalman Filter
KF	Kalman Filter
RMSE	Root Mean Square Error
SVD	Singular Value Decomposition
AUV	Autonomous Underwater Vehicle
ROV	Remotely Operated Vehicle

Chapter 1

Introduction

A fundamental question that arises in the operation of robotic underwater vehicles is that of determining their relative position and orientation with respect to some inertial reference frame. Linear and angular position information is crucial, for instance, as an input to control systems, to correctly georeference scientific data, for safety purposes, and mission supervision. In the robotics literature, estimating the position and attitude of a vehicle is commonly referred as *Navigation*. Closely related terms are *Guidance*, which deals with the generation of spatial (and/or temporal) reference paths (ususally dependent on the data generated by the Navigation) and *Control*, which deals with the generation of low-level actuator reference signals in order to follow these paths.

The English term navigation derives from the Latin *navigare*, where *navis* means ship and *agere* means to lead, or to drive. The term also has a similar root in Sanskrit *navgathi*. Both Latin and Sanskrit languages belong to the Indo European family which explains the similitude between the two roots and corroborates the ancient origin of the term. The science of navigation has a vast and rich history, and has been naturally linked to the exploration of our planet through the oceans, human migrations, and the development of commerce and civilizations since ancient times. Although originally the term Navigation was almost limited to the maritime world, nowadays it is also used in many other different fields related to avionics, mobile robotics, space exploration, and the principal subject of this thesis: underwater robotics.

Over the last years there has been a surge of interest and fast paced development of Inertial Navigation Systems (INS) and Global Positioning System (GPS) related technologies. Nowadays, some of these technologies have become common in consumer products, and it is not unusual to find GPS receivers, compasses, and accelerometers in mobile phones, cars, or computer gaming controls. Given these developments, and specially the popularization of GPS receivers, one might be tempted to think that the Navigation problem is almost solved, except for a steady search of more precise, reduced size, cost, and power effective solutions. However, in many

practical scenarios where GPS signals are not available the navigation problem is still quite challenging. Typical environments on which GPS is unavailable, or not reliable, include

- **Indoor**: Inside buildings, where GPS signals are blocked by concrete walls [102], [142].
- **Urban**: GPS signals are not reliable and usually blocked by surrounding buildings or totally lost in tunnels [50].
- **Caves**: Either on land or underwater, they are a major challenging environment for any navigation task [60].
- **Space**: Although there has been some experimental work on using GPS signals on extra-terrestrial navigation, GPS signals are usually not available in space [127].
- **Underwater**: Either in lakes, rivers, or oceans, where GPS signals are blocked by the water surface [111], [97]

One of the simplest approaches to navigation is to rely on the estimates of the velocity of a moving platform and of the time of travel in order to predict the future position with respect to a previous position fix. This technique is known as *Dead Reckoning* (DR) Navigation. There is some discussion about the etymological origin of the term. Although some authors claim that this term originates from an abbreviation of *deduced reckoning* there is no apparent historical evidence that support this, and the most provable origin has to do with computing the relative position with respect to a *dead*, or fixed, point. There are many different ways of doing dead reckoning navigation, depending on the kind of measuring instruments, all having in common the use of speed and direction information, and the absence of absolute position measurements.

The history of the evolution of instruments and devices used for DR navigation is quite rich and filled with great inventions from the ancient compass and *ship log* to measure direction of travel and speed over water, to the nowadays sophisticated Inertial Navigation Systems (INS) and Doppler Velocity Logs (DVL). One of the intrinsic problems of dead reckoning navigation is that, due to the uncertainty associated with any kind of measurement, the position estimation errors increase with the time and distance travelled. In the absence of position measurements, dead reckoning relies on the integration in time of velocity and/or acceleration measurements which generates a random walk kind of error with increasing covariance. This phenomena is often referred as *drift*. The problem can be minimized by using better and more sophisticated sensors, together with improved navigation algorithms, but due to the stochastic nature of the measurements drift it can not be completely eliminated. Moreover, by means of DR navigation, one can only perform relative navigation with respect to an initial position fix. Velocity and heading measurements contain no information about absolute position and are invariant to pure transla-

tions. This means that, strictly speaking, by DR one can answer *how much have I moved* but not *where am I*.

In order to solve the problems associated with DR one must resort to some external measurements containing position information. The most obvious are absolute position measurements such as that given by GPS receivers, but any measurement of variables that are correlated with the position of the vehicle can be used for this purpose. For instance, bearing and range with respect to points with known positions, terrain elevation, temperature, chemical and geophysical features whenever a feature map is available. In striking contrast to what happened with velocity and orientation measurements, the above data are not invariant to pure translations and can be used to mitigate the drift of the DR system. This thesis will focus on the use of range measurements between the vehicle and landmarks with known inertial coordinates.

It is common to use DR together with other techniques to perform precise absolute navigation. DR and Inertial Navigation usually provide measurements with good short-time accuracy at high sampling rates whereas the instruments that provide absolute position information usually provide measurements with good long term accuracy, but with larger errors at much lower sampling rates. The art of using all the available measurements in the best possible way so as to exploit the benefits of each individual sensor and minimizing their weaknesses is the subject of sensor fusion, which is a fundamental ingredient of advanced Navigation systems. It is natural to envision that in the future navigation systems will make use of many different information sources, with very different temporal and spatial scales and associated errors, thus making sensor fusion a vital issue.

It is worth pointing out that there is a navigation methodology that is somehow in between pure DR and navigation with external position measurements consisting of the use of *on the fly* generated feature maps to help eliminate the drift of the DR navigation. This technique is usually called Simultaneous Localization and Mapping (SLAM) or less commonly Concurrent Mapping and Localization (CML). The absolute position of the vehicle remains unknown but it is now possible to navigate with respect to a relative frame eliminating most of the DR associated problems. This method has become popular among the mobile robotics community, and has a great potential for long range exploration kinds of missions.

It is important to mention that Navigation is one of the scientific fields in which some animals outperform state of the art technology. Some insects, birds, fish and marine mammals are able to navigate during long distance migrations and reach certain locations with a precision that is still nowadays hard to explain by scientists and researchers worldwide: *"How animals navigate through apparently featureless environments with pinpoint accuracy, often over very long distances, remains one of the great, unsolved mysteries in biology"* [185]. The field of animal navigation is extremely exciting and will undoubtedly provide inspiration and solutions for future

long distance navigation of autonomous robotic vehicles where GPS is unavailable [30].

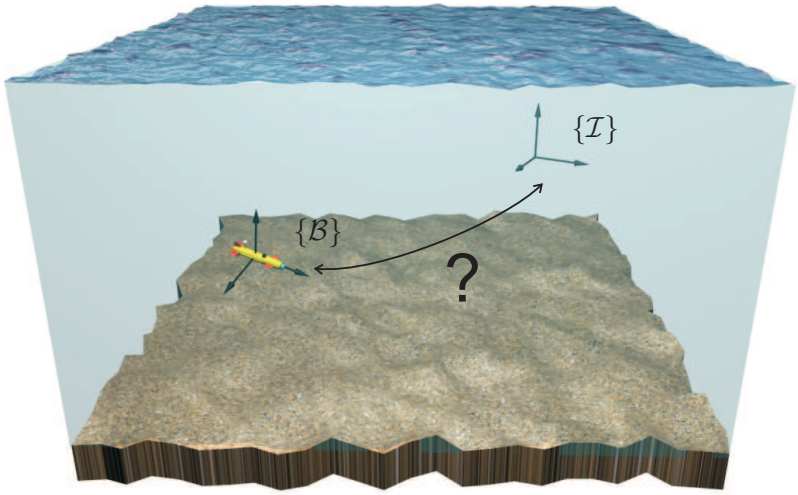


Figure 1.1: Navigation Problem

The Navigation problem can be stated in a quite simple way: determine the position and the relative orientation of an object (vehicle) with respect to some reference frame. In order to state this more formally, define a reference frame attached to the vehicle $\{\mathcal{B}\}$ and a reference frame $\{\mathcal{I}\}$ with respect to which one wishes to determine position and relative orientation, see Fig. 1. It is common to make distinction between absolute (or global) and relative (or local) navigation. From a mathematical point of view there is little difference among them. In the first case, the absolute position of $\{\mathcal{I}\}$ is known, say its coordinates in some geodetic reference system are given, whereas in relative navigation the absolute position of $\{\mathcal{I}\}$ is either unknown or irrelevant. In both cases, the position of $\{\mathcal{B}\}$ with respect to $\{\mathcal{I}\}$ expressed in $\{\mathcal{I}\}$ can be represented by vector ${}^{\mathcal{I}}\mathbf{p}_B \in \mathbb{R}^3$ and the relative orientation of $\{\mathcal{B}\}$ with respect to $\{\mathcal{I}\}$ can be represented by the rotation matrix ${}^{\mathcal{I}}_{\mathcal{B}}\mathcal{R} \in SO(3)$. Rotation matrices belong to the Special Orthogonal group defined as

$$SO(3) = \left\{\mathcal{R} \in \mathbb{R}^{3\times 3} \,:\, \mathcal{R}^T\mathcal{R} = \mathbf{I}_3,\ \det(\mathcal{R}) = 1\right\} \tag{1.1}$$

where $\mathbf{I}_3$ is the 3-by-3 identity matrix, and $\det(\cdot)$ is the matrix determinant operator [130]. The position and relative orientation of the vehicle can then be identified with the pair $({}^{\mathcal{I}}_{\mathcal{B}}\mathcal{R}, {}^{\mathcal{I}}\mathbf{p}_B)$, an element of the Special Euclidean group $SE(3) = SO(3) \times \mathbb{R}^3$. The Navigation problem can then be formulated as an estimation problem in $SE(3)$ where the nature of the observations varies depending on the particular application. In some cases one is also interested in estimating the time derivatives of the position

and relative orientation, either because they are needed or as a means of obtaining better and smoother position estimates.

There are many kinds of sensors and information sources that can be used in order to solve the navigation problem. Among them, the present thesis will focus on a particular kind of observations that are relevant due to their extensive use in underwater robotics applications: Range-Only measurements. These kinds of measurements, which are naturally obtained from the times of arrival of acoustic signals, are the basis of acoustic positioning systems. In what follows we give an overview of underwater navigation techniques and acoustic positioning systems.

1.1 Underwater Navigation

The last decade has witnessed the emergence of Ocean Robotics as a major field of research. Remotely operated vehicles (ROVs) and, more recently, autonomous underwater vehicles (AUVs) have shown to be extremely important instruments in the study and exploration of the oceans. Free from the constraints of an umbilical cable, AUVs are steadily becoming the tool par excellence to acquire marine data on an unprecedented scale and, in the future, to carry out interventions in undersea structures. Central to the operation of these vehicles is the availability of accurate navigation and positioning systems. Due to the fact that GPS signals are not available below the water surface, underwater navigation is a particularly challenging task.

The topic of underwater navigation is still far from being solved. Partial solutions exists, mainly for local or short range navigation, but the quest for long range navigation is still active. Inertial Navigation systems and DVL sensors are becoming more popular, affordable, and decreasing in size, although they are not sufficient for some particular missions. Acoustic navigation is widely employed but has many drawbacks including high operational costs and relatively short ranges. It is reasonable to predict that future navigation systems will make use of a high number of sensor sources and techniques suited for different time and space scales. For an introduction to and a survey on this topic see [111], [97], [180], [181], [125], [85], [71], [82], [64], [44], [123].

1.1.1 Dead Reckoning and Inertial Navigation

One of the most common solutions to underwater navigation is Dead Reckoning. In the absence of absolute position measurements, DR relies on the time integration of velocity and/or acceleration measurements to estimate the vehicle displacement from an initial position fix. There are several types of sensors and methodologies to perform DR navigation, varying in complexity and their long time accuracy. Inertial Navigation Systems (INS) are one of the most widely employed. They are typically composed of accelerometers, rate gyros, and magnetometers, and can

provide position and attitude information with good short time accuracy and high update rates. However their cost remains prohibitive.

Another sensor that is commonly used in underwater navigation is the Doppler Velocity Log (DVL). It is an acoustic sensor based on the Doppler effect and it measures the relative velocity between the sensor unit and the water column (*water lock*) or between the unit and the sea bottom (*bottom lock*). When the vehicle altitude above the sea bed is sufficiently small, typically no more than 100-200m, depending on the unit working frequency and specifications, the velocity with respect to the sea bottom can be measured. This mode of operation is preferred from a navigation point of view because, unlike the measurements of velocity with respect to the water column, it is not affected by water currents [98], [187].

A key issue in both INS and DVL systems is the availability of accurate attitude estimates. Typically, these sensors provide velocity measurements in a reference frame solidary to the sensor unit. Therefore, in order to use them for vehicle navigation, it is necessary to convert the measurements from sensor to inertial reference frame. To do so, it is crucial to know what is the actual attitude of the sensor unit with respect to the inertial reference frame, and also how the sensor unit is mounted with respect to the vehicle reference frame. The importance of the latter information, referred to as the vehicle-sensor alignment, is often underestimated and is actually one of the most notorious sources of error of DR navigation using INS and DVL sensors [99][187]. Extremely small attitude errors, when integrated for a long time can produce large position errors.

Due to the fact that DR errors increase with the distance and time of travel, one approach is to resurface periodically to acquire GPS fixes which is sometimes referred as Yo-Yo maneuvers. This unsuitable for long range and deep water operations, and whenever accurate absolute position measurements are required.

1.1.2 Acoustic Navigation

Due to the good propagation characteristics of sound in water, acoustic positioning systems have been widely employed to reduce the errors associated with DR navigation. There is a great diversity of acoustic positioning systems, suited for different tasks and navigation accuracies, most of them relying on the computation of ranges or bearings to acoustic sources with known positions based on measurements of the times of arrival (TOA) or time differences of arrival (TDOA) of acoustic signals [125], [97], [180], [12], [10].

LBL

Long Baseline systems are one of the most classical and widely used class of underwater acoustic positioning systems. They rely on a set of beacons/transponders fixed to the sea bottom with know positions Fig. 1.1.2. Typically, an underwater vehicle

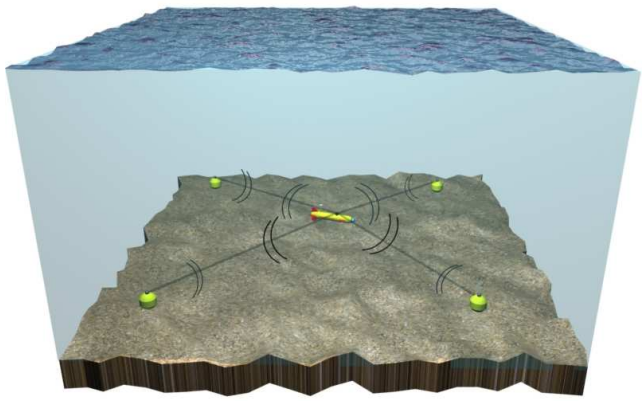


Figure 1.2: LBL

carries a transponder that can interrogate the beacons. Each beacon replies and the two way travel times are used to compute the distances between the vehicle and the beacons. The resulting range measurements can then be merged in the navigation system using a filtering algorithm such as the Extended Kalman Filter (EKF), or collected in a batch until enough distances are measured, from which a trilateration position fix can be obtained. The navigation system can automatically determine the sequence of beacon interrogation so as to optimize the estimation error. The area of operations that is possible with LBL is ususally not bigger than a few kilometers, depending on the working frequency of the system and its characteristics. The term baseline refers to the distance between the beacons which in this case is considered to be long. It is typical to have slow interrogation cycles so that update rates of 10 seconds or longer are common. Typically, LBL systems are used for relatively long range and wide area coverage navigation, and generate errors in the order of few meters. There are, however, some high frequency LBL systems employed for short range precision positioning such as the EXACT [192] [187] that claim to provide centimetric accuracy. These high frequency systems have been used for drilling operations and very accurate archaeological mapping.

The fact that the beacon positions should be known accurately introduces the need for a calibration procedure prior to operations. Usually, the beacons are deployed from a surface ship, and by interrogating the beacons several times, range measurements to the ship from different locations are acquired that can be used to estimate the beacons positions [85], [37]. Any error in the calibration will have direct effects on the vehicle navigation performance. At the end of the operations, the beacons can be recovered by using an acoustic release. The operational costs of a mission involving an LBL system are considerable, including the deployment, calibration and recovery of the beacons, which stresses the need for improved un-

derwater navigation solutions.

SBL

Short baseline systems use baselines on the order of 1-100 meters and are usually composed of a set of receiver hydrophones mounted on the hull of a support vessel or in a rigid structure [157]. Usually, SBL systems are used to track an underwater vehicle from the surface, so that its position is computed and known at a surface ship and not at the vehicle Fig. 1.1.2. In this case the system provides only relative position estimates between the SBL hydrophones and the vehicle. If absolute position estimates are required, it is necessary to accurately know the absolute position and orientation of the structure (usually a support vessel) where the hydrophones are mounted.

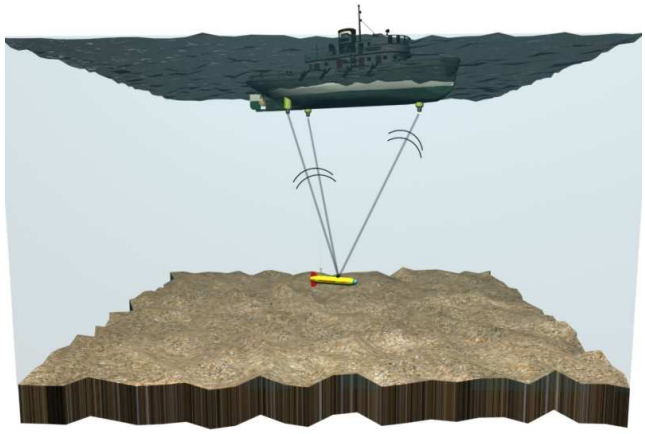


Figure 1.3: SBL

USBL

Ultra Short Baseline (USBL) systems are one of the most widely employed underwater tracking systems. In this case, the baseline, the typical length between the receiving elements, is less than 1m. The system is composed by an USBL transducer head that contains an array of acoustic transponders (typically 4) mounted in a convenient geometry and a pinger installed on-board the vehicle one wishes to track, as depicted in Fig. 1.1.2. Using the times of arrival and the time difference of arrivals in the receiving array it is possible to determine the direction and range between the USBL head and the pinger on-board the vehicle. Due to the extremely short baselines it is usually required to use advanced signal processing techniques. In order to determine the absolute position of the vehicle, it is necessary to accurately know the actual position and attitude of the USBL transducer head. In order to do this, typically these units contain an INS and take the input of a GPS receiver,

whose antenna position with respect to the transducer head is known. Any error on the estimated position and attitude of the USBL sensor head is automatically translated into vehicle position estimation errors, so that system calibration prior operations is of utmost importance. The calibration procedure that is common to most INS systems aims at identifying and compensating the ferromagnetic signature of the vessel, as well as identifying other system parameters [137].

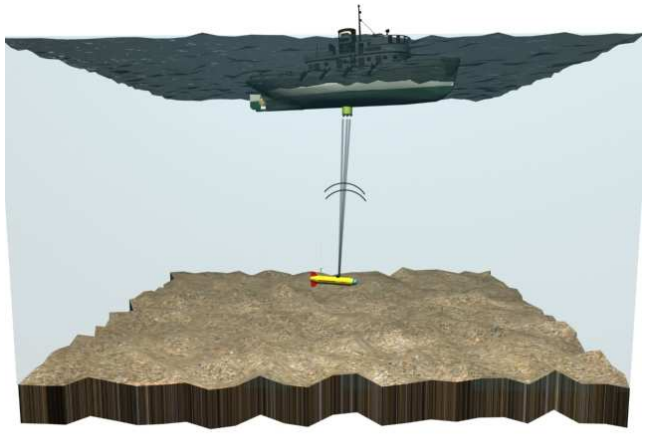


Figure 1.4: USBL

USBL are commonly used to track underwater vehicles, but can not be directly used for vehicle navigation, as the position measurements are only known on the support ship carrying the USBL transducer. Some commercial units incorporate an acoustic modem capability so that position information can then be sent back to the vehicle through the acoustic channel. There is an alternative configuration called inverted USBL that allows for vehicle navigation. In this case, the vehicle carries the USBL transducer head, and navigates by using an acoustic pinger with known position, for instace fixed on the sea bottom [128] [181].

USBL systems are widely used because they are simple to operate and have relatively moderate prices as compared to other systems. The resulting position estimation errors are usually greater that in other longer baseline systems, very sensible to attitude errors on the transducer head, and increase with the slant range. One can achieve relatively good relative navigation and repeatability, for instance when several pingers are tracked simultaneously or if one pinger is used as a homing reference for the other, but the system can generate large absolute position errors.

GIB - Inverted LBL

The quest for a GPS-like system underwater is still very active. GPS has become the workhorse of terrain navigation because of its wide area coverage, the capability to provide navigation data seamlessly to multiple vehicles, relatively low power

requirements, miniaturization of receivers, and for being environmental friendly in the sense that its signals do not interfere significantly with the ecosystem. Typical acoustic underwater positioning systems are quite the opposite: they exhibit reduced area coverage, do not usually scale well so as to serve for multiple vehicle navigation, have high power requirements, and show moderated to high impact on the environment in terms of acoustic pollution.

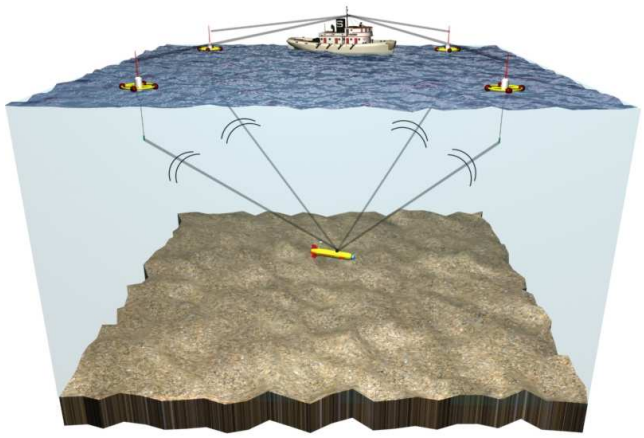


Figure 1.5: GIB

The concept of having a set of surface buoys equipped with GPS receivers broadcasting underwater GPS-like signals was introduced in [193]. Given the problems and difficulties associated with underwater acoustic communications this concept has not been yet fully realized. However, there has been an increasing number of systems using sets of surface buoys equipped with GPS receivers and acoustic communication capabilities [169] [65] [12]. The GPS Intelligent Buoy's (GIB) system is a clear example, and is commercially available. The system consists of a set of surface buoys with GPS receiver, submerged hydrophones, and radio modems. The times of arrival of a pinger installed on-board the vehicle, which is synchronized with GPS time prior to system deployment, are recorded by the buoys and sent in real time through the radio link to a control unit, typically on-board a support vessel, where the data is processed and a position fix is computed [10], see Fig. 1.1.2. Note that, unlike in a LBL system, the position information is only available at the control unit and therefore the system can not be directly applied for vehicle navigation. The GIB and alike systems are basically used to track underwater platforms. If one wishes to use them as a real-time underwater vehicle navigation aid, the need arises to use an acoustic modem to inform the vehicle about its own position. This type of systems is also referred as an inverted LBL since, in this case, the sea bottom fixed transponders have been replaced by surface buoys and the information flow is somehow opposite as compared to a classic LBL. The advantage of these systems is

that the operational costs are reduced because they eliminate the need to deploy, calibrate and recover a set of sea bottom transponders, while providing good accuracy on the order of a few meters. Typically, the surface buoys are free drifting or moored, but there are also systems with self propelled buoys which allow for the execution of basic station keeping and underwater platform tracking without the need of a mooring line or when operating at high depths.

1.1.3 Geophysical and Feature based Navigation

It was mentioned before that any environmental features that show rich spatial diversity can eventually be used to extract position information. There are many diverse types of features that can be used for underwater navigation. The most common are bathymetric, geomagnetic and gravimetric features, although other have been used and proposed such as light intensity and its diurnal variation, as well as the concentration of certain chemical components.

Bathymetric data has been used for long time as an external aid to underwater navigation of manned submarines. By using an echo-sounding sonar it is possible to measure the altitude over the sea bottom, which coupled with depth readings from a pressure sensor, allows for the measurement of the local bathymetry. If a bathymetric map of the area is available it is possible to use diverse map-correlation and filtering techniques to improve vehicle navigation [28], [94], [93], [132], [166]. The use of geophysical features for underwater navigation has been the subject of study for some years but has not yet realized its full potential as a fundamental part of underwater navigation systems. As biologists and scientists worldwide try to study and understand how animals can to perform long range underwater navigation, the robotics community is trying to apply these ideas to autonomous underwater vehicles [185], [30]. For an introduction to this fascinating topic and further references see also [171], [172], [174], [173], [146], [133], [167].

1.2 Thesis outline and related publications

In what follows we give a brief overview of the topics addressed in this thesis, the contents of its chapters, and a list of the papers that contain some of the results obtained. The reader is referred to the introduction at the beginning of each chapter for more details, bibliographical references, and rigorous problem formulations.

1.2.1 Range-Only localization

The Range-Only localization problem consists of estimating the position of a vehicle by using the rages (or distance) measurements with respect to a set of points with known inertial locations. This problem arises frequently in passive radar and sonar systems, underwater acoustic positioning systems, and geophysics. The problem is

introduced and formulated in Chapter 2, which contains an overview of a number of closed form Least Squares algorithms that vary in complexity and estimation errors. Whenever possible, analytical expressions for the bias and covariances of the resulting localization errors are derived. A Maximum Likelihood estimator is formulated and Newton descent iterative minimization algorithms proposed to solve the range-only localization problem. The performance of the derived LS and iterative algorithms is compared against the Cramér- Rao Bound (CRB) for the problem.

Finally, the case in which the vehicle is not static but moves along a certain unknown spatial trajectory is considered. Filtering architectures such as Kalman Filtering (KF) plus trilateration and Extended Kalman Filtering (EKF) are discussed. A nonlinear continuous time observer using squared range measurements is derived which exhibits an Input to State Stability (ISS) property with respect to state disturbances and measurement errors.

In Chapter 3, some of the previous algorithms are applied to experimental data obtained with a GIB acoustic positioning system during sea trials. A detailed description of the GIB system architecture is done. The nature of the experimental times of arrival measurements and their associated problems, such as multipaths and outliers, is analyzed in detail. An Extended Kalman Filter which takes into account the delays inherent to the time of arrival measurements is derived and tested with experimental data.

Some of the results in these chapters appear in the authors publications [11], [10], and [12].

1.2.2 Range-only pose estimation

Range-only measurements can also be used to provide attitude information about a rigid body. When distances between points in the rigid body and points fixed in the ambient space are known, it is possible to extract relative orientation information. Although in this case it is common to use differences of times of arrival instead of range measurements, there are situations in which the latter might be more convenient. For instance, when dealing with relatively short baselines in which a planar wavefront approximation does not hold, or when fine precision is required. This approach is already being used to provide accurate (and insensitive to magnetic disturbances) GPS based heading measurements by using multiple GPS antennas. As the cost of GPS receivers is reduced, this might become a viable and cost-effective alternative (or complement) to more expensive Inertial Navigation Systems above water. Another situation in which these techniques might prove valuable are underwater and indoor applications. Typically, the attitude and positioning problem is addressed resorting to some particular parametrization of the rotation matrices such as Euler angles or quaternions. The use of a particular parametrization generates some problems such as the need for normalization schemes and singularities. Moreover, the attitude and positioning problem is often not addressed simultaneously

which compromises accuracy.

Chapter 4 addresses the range-only attitude and positioning problem from a Maximum Likelihood point of view, and proposes geometric descent algorithms on the Special Euclidean group $SE(3)$. The algorithms are of a Newton nature, converge relatively fast, and exhibit performances very close to the Intrinsic Variance Lower Bound (IVLB). However there are local minima that can eventually prevent the algorithms from converging to the correct solution. In Chapter 5, and in order to try and overcome this, a different solution is derived. To this effect, a continuous time dynamic observer is obtained based on a conveniently defined Lyapunov function of the range measurements that exhibits asymptotic convergence.

These results appear in the authors publications [14], [9], [15], and [13].

1.2.3 Smoothing of acoustic positioning data

Chapter 6 addresses the problem of off-line vehicle trajectory estimation. This problem arises for instance when post-processing position fixes from an acoustic positioning system. The chapter contains an extension of diffusion-based trajectory observers in order to incorporate variance constraints. The resulting observers are shown to be asymptotically stable and their performance is evaluated with experimental data from sea trials. The results appear in the authors publication [8].

1.3 Thesis main contributions

- Chapter 2: Range-Only localization.
  - Overview of Least Squares TOA algorithms. Statistic characterization of squared range measurements.
  - Derivation of Newton algorithms to determine Maximum Likelihood estimates with general measurement error covariances. Conditions to ensure convexity of the ML-SR cost function.
  - Derivation of a continuous time nonlinear observer with ISS properties with respect to state disturbances and range measurement errors.
- Chapter 3: Application to a GIB underwater positioning system,
  - Characterization of acoustic TOA data from a GIB system.
  - Extended Kalman Filter design incorporating delayed TOA measurements.
- Chapter 4: Range-Only Maximum Likelihood pose estimation.
  - Derivation of generalized gradient and Newton descent algorithms in $SE(3)$ to determine Maximum Likelihood estimates.

  - Derivation of a simple 3-step closed form initialization algorithm.
  - Derivation of intrinsic performance bounds.

- Chapter 5: Range-Only Dynamic pose estimation.
  - Derivation of a dynamic estimator which exhibits asymptotic stability.

- Chapter 6: Diffusion-based smoothing
  - Derivation of a diffusion-based trajectory observer incorporating variance constraints with convergence warranties.

Chapter 2

Range-Only Localization

The problem of estimating the position of an object using only the distances to a set of points with known coordinates appears in many engineering and scientific applications. Typically, the distance or range measurements are obtained from the Times of Arrival (TOA) of acoustic or electromagnetic signals given that the speed of propagation of signals in the medium is known. Because of the nonlinear nonsmooth nature of the range measurements there is no straightforward solution to the problem. The Maximum Likelihood (ML) estimator requires the minimization of a nonconvex nonsmooth cost function, where iterative descent algorithms might fail to determine a global solution. At the expense of increased estimation errors and bias, many simple closed form Least Squares (LS) algorithms have been proposed in the literature. In this chapter we review some of these closed form algorithms and illustrate their performance as compared against the Cramér Rao Bound (CRB) that specifies the best possible performance attainable with any estimator. We also describe two iterative algorithms based on the minimization of the ML function of the range measurements (ML-R) and a related cost function that uses the square of the range measurements (ML-SR). Closed form expressions for the gradient and Hessian of the ML-R and ML-SR functions are derived that allow direct implementation of fast Newton descent algorithms. The numerical complexity of the resulting iterative descent algorithms is relatively moderate as compared to some of the closed form LS estimators. We also show that under certain conditions, that can be directly checked from the available information, the ML-SR cost function is convex and therefore the convergence of the descent algorithm to a unique global solution can be warranted.

When the object whose position we want to determine is not static but moving along a certain trajectory, better performance may be achieved by using filtering or dynamic algorithms that exploit some knowledge on the dynamics of the object. We describe two algorithms based on a Kalman Filter (KF) plus closed form LS trilateration fixes, and an Extended Kalman Filter that directly fuses the Range-Only measurements. We also derive a continuous time non-linear observer which exhibits Input to State Stability (ISS) with respect to disturbances and range measurement

errors.

2.1 Introduction

In many engineering and science applications, one is faced with the problem of determining the position of an object by using only the distances between the object and a set of points with known locations. The distance, or range, measurements are often obtained from the Times of Arrival (TOA) of acoustic or electromagnetic signals. If the speed of propagation of the signals in the medium is known, and assumed constant, a simple linear relationship allows for the computation of the distances from the TOA measurements. This problem is often referred as Range-Only localization and *trilateration*. Although the term *triangulation* is also sometimes employed, it is more correct to use it to denote the position estimation using angle measurements. Some common applications of trilateration include underwater positioning systems, where one is interested in locating an underwater platform by measuring the TOA of acoustic signals propagating between the platform and a set of hydrophones/transponders with known coordinates [107, 82, 37, 37, 145, 35, 12, 10]. In indoor positioning systems, and wireless sensor network localization, one uses the TOA of electromagnetic signals between a device and a set of RF receivers/transmiters with known location [42, 142, 151].

A closely related problem is that of localization using Range Differences (RD's) or Time Differences of Arrival (TDOA). In passive sonar and radar systems and geophysics, one is often interested in determining the location of an emitting source by measuring the signals at a set of receivers with known coordinates. In these situations neither the time of emission nor the nature of the signals is known. Therefore it is common to adopt a two step solution: first, estimate the vector of time delays by analyzing the correlation between pairs of signals at different receivers [78], [66] and then use the resulting delays as an input for TDOA localization algorithms such as [156, 67, 40, 84]. Note that, by assuming a constant signal speed of propagation it is straightforward to transform time delays into ranges. From a mathematical point of view one can always transform a set of TOA measurements into a set of TDOA but the opposite is not possible. This means that all of the TDOA and RD's algorithms can be eventually used to solve a TOA localization problem. However this is done at the expense of increasing estimation errors. The TDOA localization problem is often refereed as Hyperbolic localization, given that in the two dimensional problem each Range Difference equation defines an hyperbola where the position of the source might lie. This denomination is opposed to Spherical or Circular localization with TOA or range measurements where each observable defines a sphere of possible locations.

Typically, range measurements are obtained by measuring the time of travel of acoustic or electromagnetic signals between an emitter and a receiver given that

the speed of propagation in the medium is known. There are several mechanisms to do this, most of them including precise timing and clock synchronization issues. If the emitter and the receiver are synchronized, the time of flight can be directly determined from the time of arrival at the receiver using one way communication. If the emitter/receiver are not synchronized then the problem becomes more subtle. A common approach is to use an interrogation procedure from which the two way travel time of signals can be used. Otherwise, one is left with an extra degree of uncertainty that accounts for the unknown clock synchronization error, giving rise to what are commonly referred as pseudorange observations. These are the observables found in GPS systems, where the clock on the receiver has a bias with respect to the atomic clocks on-board the GPS satellites [75]. Pseudorange localization problems are addressed in [22], [3], [39], [38].

Because of the nonlinearity and relative complexity of the range equations and its numerous applications, there has been a long history of solutions and methods to solve the Range-Only localization problem. One of the most naive approaches considers direct solutions resulting from algebraic manipulation of the range equations. Closed form solutions can be obtained in some cases that have few concern about measurement noise sensitivity and the resulting estimation errors [76], [141], [107], [82], [37], [120].

The Range Difference or TDOA localization problem has received much more attention in the literature than the TOA equivalent. The complete problem is even more complex than the TOA localization and no simple optimal solutions exist. The Maximum Likelihood estimator requires the minimization of a nonconvex and nonsmooth cost function where descent algorithms might fail to find a global solution. This, together with the increased numerical and computational complexity of iterative algorithms motivates the use of simple Least Squares (LS) closed form algorithms that can be implemented in simple electronic devices or used as initial fixes for iterative methods. Historically, the first of these algorithms was the plane intersection presented in [153]. In [152], the authors proposed the Spherical Intersection (SX) algorithm which consisted in the resolution of a second order polynomial and a simple linear Least Squares problem. Inspired by the SX algorithm, Abel and Smith introduced the celebrated Spherical Interpolation (SI) algorithm [4, 156, 155]. The SI algorithm is quite simple and behaves apparently well under moderate noise conditions and sensor geometry. In [40], a further refinement of the SI algorithm is presented that is sometimes referred as Quadratic-Correction Least-Squares algorithm. The authors show (analytically) that their algorithm is efficient (it achieves the CRB) under some assumptions, involving many simplifications and approximations. A further improvement of the SI method was presented in [84] and named Linear Correction Least Squares approach. It resorts to Lagrange multipliers and the computation of the roots of a 6-degree polynomial. Most of the previous methods rely on a set of transformations on the TDOA equations involving squaring. The

noise impact of squaring the observations and how under some realistic assumptions the measurement noise remained Gaussian are discussed in [5],[40], [84]. The mutually dependent noise model (as compared to the independent Gaussian model) is discussed in [84].

As previously mentioned, all the TDOA LS closed form algorithms can eventually be used to solve a TOA localization problem. Most of them, however, can be modified so as to deal explicitly with the TOA observations. In [42], a modification of the method presented in [84] is derived for the TOA problem. The method requires finding all the roots of a 5-degree polynomial. In [25], the authors present a simpler method based on Generalized Trust Region Subproblems that does not require all of the roots but only one which can be efficiently determined. The advantage of LS solutions is that they are computationally simpler as compared to iterative minimization algorithms. The price to pay is a degradation of performance. That is, an increase in the sizes of the estimation error variances and biases. By introducing several correction methods and sophistication's as in [40, 84, 25] one may be able to partially overcome some of this drawbacks but never completely overcome them. Moreover, some of these improved LS algorithms include finding the roots of high order polynomials and other numerically involved processes that might approach some TOA iterative algorithms in complexity.

In theory, one would like to determine the Maximum Likelihood (ML) estimate for the problem. However, this implies the minimization of a nonconvex nonsmooth cost function. Recent publications have started revisiting the nature of ML cost function and related iterative algorithms [26]. Here, the authors study two cost functions based on the ML of the range measurements and on the square of the ranges, that we will call ML-R and ML-SR functions, respectively. The authors further derive simple iterative algorithms and prove their convergence (eventually to a local minimum), resorting to Trust Region optimization algorithms. The algorithms are of a gradient nature, and thus exhibit poor linear convergence rates. In this chapter we will derive simple closed form expressions for the gradients and Hessians of the ML-R and ML-SR functions using Matrix Differential Calculus. This allows for simple implementation of fast Newton descent algorithms. Moreover, it will be shown that under certain conditions, which basically require that the unknown object position lies relatively close to the sensors centroid, the ML-SR function is in fact convex. Under these conditions, iterative descent algorithms can be shown to converge to a unique global minimum. The resulting algorithms are numerically relatively simple, and yield convergence in few iterations, thus being suitable for real-time implementations.

Performance bounds are a central tool in order to compare and benchmark different estimators. Cramér Rao Bound (CRB) sets a lower bound on the Root Mean Squared Error (RMSE) of any unbiased estimator [170, 95]. When an algorithm attains the CRB it is said to be *efficient*. The CRB can also be used in order to

optimize the sensor design. In [6, 2] a graphical interpretation of the Fisher Information Matrix is derived which provides an intuitive way of optimizing sensor design. In [54], an interesting and simple comparison between the CRB for the TOA and TDOA problem is given. We will derive the CRB for the TOA problem and compare it with numerical simulations of the TOA localization algorithms. Simulation results suggest that both the ML-R and ML-SR iterative algorithms are efficient and attain the CRB.

For the sake of completeness we should also briefly mention some other methods of very different nature and motivation that have been propposed to solve the Range-Only localization problem. The problem of sensor network localization has received an increasing attention in the last years. In this problem, one wants to determine the relative position of a set of nodes that measure distances between them and a set of nodes with known locations, called anchors. Several solutions have been proposed based on semidefinite relaxation (SDR) and Euclidean Distance Matrix (EDM) completion problems [56], [160], [53], [43], [25]. Those formulations are mostly suited for problems with a high number of nodes and require the use of relatively complex Semidefinite programming (SDP) toolbox such as the SeDuMi. These kind of algorithms require considerable computational resources and are not specifically suited to be implemented in real time on autonomous robotic vehicles. Other approaches include [1], where a *divide and conquer* approach is presented. In [35], a *set-membership* algorithm is presented which allows to deal with non constant sound velocity profiles. In [168, 36] methods based on Cayley-Manger determinants are considered, and in [77] a particle filter solution is presented.

2.2 Problem Formulation

In this section we formulate the Range-Only localization problem. We are interested in determining the absolute position of a vehicle with respect to some fixed inertial reference frame by using noisy distances or range measurements between the vehicle and a set of landmarks with known positions, see Fig. 2.1.

Let $\{\mathcal{I}\}$ be some fixed local inertial reference frame. The origin of $\{\mathcal{I}\}$ is located at an arbitrary point, selected by the user, with respect to which we will formulate the localization problem. Typically, the origin of $\{\mathcal{I}\}$ is chosen near the area of operation, in order to avoid large numbers that could generate numerical conditioning problems, see appendix D for more details. Let $\mathbf{p} \in \mathbb{R}^n$ be the vector representing the position of the vehicle with respect to $\{\mathcal{I}\}$. The dimension of the space in which the estimation problem is formulated is denoted by n and can be either 2 or 3. Consider a set of m points, or *landmarks*, with known positions with respect to $\{\mathcal{I}\}$ denoted by $\mathbf{p}_i \in \mathbb{R}^n$ where $i \in \{1, 2, \ldots, m\}$. These points could be, for example, acoustic transponders fixed on the sea bed or surface buoys with submerged hydrophones and will be referred in the sequel as *landmarks*.

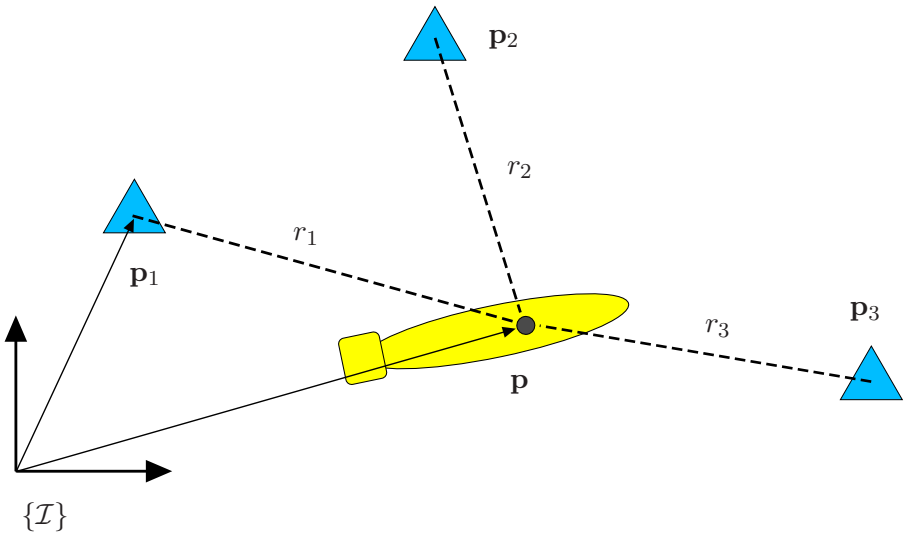


Figure 2.1: Range-Only localization problem.

Suppose that one measures the distance, or range, between i'th landmark and the vehicle as

$$\bar{r}_i = \|\mathbf{p} - \mathbf{p}_i\| + w_i, \quad i \in \{1, 2, \ldots, m\}, \tag{2.1}$$

where $\|\mathbf{p} - \mathbf{p}_i\| = r_i$ is the actual range, and w_i is some zero mean Gaussian stationary measurement error. Define vectors $\bar{\mathbf{r}} = [\bar{r}_1, \ldots, \bar{r}_m]^T$, $\mathbf{r} = [r_1, \ldots, r_m]^T$, and $\mathbf{w} = [w_1, \ldots, w_m]^T$. Let $\mathbf{R} = \mathrm{E}\left\{\mathbf{w}\mathbf{w}^T\right\} \in \mathbb{R}^{m \times m}$ denote the measurement error covariance matrix. The full set of range measurements can then be written as $\bar{\mathbf{r}} = \mathbf{r} + \mathbf{w}$. Note that at this point we don't make any assumption on the structure of the covariance matrix $\mathbf{R}$. The nature of the ranging errors can be quite complex and hard to model. To assume that they are Gaussian, zero-mean, and uncorrelated without further justification seems quite unfair. Although in practice we need to make this kind of assumption in order to formulate problems and make them treatable, one should always keep in mind what the impact such simplifications will have. Typically, the estimation errors originate from converting a time of flight quantity into a distance by multiplying it by an assumed constant and known signal speed of propagation.

In the underwater environment, the constant sound speed model is far from being rigorous, as the sound velocity profile in the ocean varies with depth, temperature, and salinity [176][116]. Depending on the region of operation and environmental conditions this can be a mild simplification or a recipe for failure. In the presence of severe thermoclines, deep water operation, and fast temporal variations, such costant speed models must be considered carefully. Even if the sound speed were constant, one would still need to estimate it and unavoidably make some errors while doing so. Consistent errors in the value of the speed of sound will generate biased range

estimates with non zero-mean errors. If the sound speed is over-estimated, so will be the range measurements and vice versa. Another often important nuisance is the existence of multipath detections and other sort of outliers. If multipaths are not properly identified and isolated, they generate disturbances that are highly biased and non Gaussian as discussed in [135]. For a more detailed discussion on these important issues the reader is referred to the latter section 3.3.

We will assume that the measurement error $\mathbf{w} \in \mathbb{R}^m$ is zero mean, that is, $\mathrm{E}\{\mathbf{w}\} = 0$, and is Gaussian with covariance

$$\mathrm{E}\left\{\mathbf{w}\mathbf{w}^T\right\} = \mathbf{R} := \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \dots & \sigma_{1m}^2 \\ \sigma_{12}^2 & \sigma_{22}^2 & \dots & \sigma_{2m}^2 \\ \vdots & & \ddots & \vdots \\ \sigma_{1m}^2 & \dots & \dots & \sigma_{mm}^2 \end{bmatrix} \in \mathbb{R}^{m\times m}, \tag{2.2}$$

which is not necessarily diagonal. By doing this, it is possible to incorporate different measurement error models. The simplest case is to consider independent and equal standard deviation σ for all the landmark measurements, i.e., $\mathbf{R} = \sigma^2\mathbf{I}_m$, or independent with different standard deviations (for instance as a function of range) $\mathbf{R} = \mathrm{diag}(\sigma_1^2, \dots, \sigma_m^2)$. In practice, however, it is often the case when the errors are not independent and the covariance is not diagonal. Another common error model is the *mutually dependent* noise, as discussed in [84], in which case the errors have a common and an independent component. For instance this can be modelled by taking the covariance matrix as $\mathbf{R} = \sigma^2(\mathbf{I}_m + k\mathbf{1}_m\mathbf{1}_m^T)$ for some positive scalar $k > 0$.

The Range-Only localization problem can now be stated as follows

Problem 2.2.1 (Range-Only Localization)**.** *Let $\mathbf{p} \in \mathbb{R}^n$ be the position of a vehicle and $\mathbf{p}_i \in \mathbb{R}^n; i \in \{1, \dots, m\}$ be the positions of a set of landmarks. Further, let $r_i = \|\mathbf{p} - \mathbf{p}_i\|$ denote the distance between the vehicle and landmark i. Define $\mathbf{r} = [r_1, \dots, r_m]^T$. Compute an estimate $\widehat{\mathbf{p}} \in \mathbb{R}^n$ of $\mathbf{p}$ based on a set of range measurements $\bar{\mathbf{r}} = \mathbf{r} + \mathbf{w} \in \mathbb{R}^m$, where $\mathbf{w} \in \mathbb{R}^m$ is a zero mean Gaussian disturbance vector with covariance $\mathbf{R} \in \mathbb{R}^{m\times m}$.*

The Range-Only localization problem assumes that the vehicle is static. The vehicle dynamics are not considered and the problem is formulated as a static parameter estimation. When the vehicle is not static but moves along a certain spatial trajectory, instead another kind of problem can be formulated. Consider that the vehicle moves along a certain smooth curve $\mathbf{p} : [0, T] \subset \mathbb{R} \to \mathbb{R}^n$, where the position of the vehicle at time $t \in [0, T]$ is given by $\mathbf{p}(t) \in \mathbb{R}^n$. The landmark positions might also change with time according to $\mathbf{p}_i(t)$. Consider a set of range measurements $\bar{r}_i(t) = \|\mathbf{p}(t) - \mathbf{p}_i(t)\| + w_i(t)$ at discrete times $t \in \{t_1, \dots, t_N\}$ with $t_k \in [0, T]$ and $w_i(t)$ a zero mean Gaussian variable.

Problem 2.2.2 (Range-Only Target Tracking)**.** *Obtain an estimate of the vehicle trajectory $\mathbf{p}(t) \in \mathbb{R}^n$, $n \in \{2, 3\}$ given a set of range observations $\bar{\mathbf{r}}(t) =$*

$\mathbf{r}(t) + \mathbf{w}(t) \in \mathbb{R}^m$ *corresponding to times* $t \in \{t_1, \ldots, t_N\}; t_k \in [0, T]$ *consisting of the distances between the vehicle and a set of* m *landmarks with positions* $\mathbf{p}_i(t) \in \mathbb{R}^n; i \in \{1, \ldots, m\}$ *corrupted by a zero mean Gaussian disturbance* $\mathbf{w}(t) \in \mathbb{R}^m$ *with covariance* $\mathbf{R}(t) = E\left\{\mathbf{w}(t)\mathbf{w}(t)^T\right\} \in \mathbb{R}^{m \times m}$.

The Range-Only localization problem, as opposed to the Range-Only target tracking problem, considers a set of range measurements from a single instant of time, or epoch. Past time information, as well as vehicle dynamics are neglected. The algorithms used to solve this problem are interesting because of a number of reasons. First, because of their simplicity, it is possible to implement them on platforms with low computational and power resources such as microcontrollers. Second, they do not require any information about vehicle dynamics. Moreover, since they do not depend on past information, and only on the current set of measurements they are more robust to outliers and sensor failures, since they can only affect the current estimate but not the future estimates. However, the performance and estimation errors achievable with such kind of algorithms are worse than those of more sophisticated filtering and dynamic algorithms used to solve the Range-Only target tracking problem.

We now consider the resolution of the Range-Only localization problem using closed form and iterative algorithms. First, we will introduce a set of simple Least Squares algorithms with different complexities and performances. Next we will introduce some iterative iterative descent algorithms to solve the Maximum Likelihood problem that, as will be shown, achieve better performance at the expense of greater numerical complexity.

2.3 Closed form Least Squares (LS) algorithms

Finding the optimal solution to the Range-Only localization problem from a Maximum Likelihood point of view, involves the resolution of a nonconvex nonsmooth optimization problem. Iterative descent algorithms might be employed to solve the problem but with a lack of convergence warranties and a relatively high computational cost. In order to overcome these problems, several approximation Least Squares algorithms have been proposed in the literature. In this section we will introduce some of these algorithms and illustrate their performance.

The Range-Only localization problem is closely related to the Range Differences localization problem. As was already mentioned, it is straight forward to obtain Range differences from Range measurements (or similarly TDOA from TOA measurements) but the opposite is not true. Hence, all the TDOA and Range Difference localization methods can also be used to solve the Range-Only localization problem. In fact most of the Least Squares TOA algorithms were inspired by their TDOA counterparts, as the second kind of problems has received more attention in the literature.

In this section we will introduce some Least Squares algorithms to solve the Range-Only localization problem. The first will be called Unconstrained Least Squares (LS-U) and Unconstrained Weighted Least Squares (LS-UW). The algorithms consist of manipulating the squared Range measurements to obtain a linear equation in the unknown position $\mathbf{p}$ and its square norm $\|\mathbf{p}\|^2$. Neglecting their dependency, an unconstrained linear LS can be solved. This is one of the simplest algorithms and is inspired by their TDOA equivalents [152], [155], and formally introduced in [42], [25]. Instead, one can eliminate the term in $\|\mathbf{p}\|^2$ from the equations by using an appropriate projection matrix. This yields a simple linear equation in the unknown $\mathbf{p}$ which can be solved by Least Squares. The resulting algorithms will be referred as Centered Least Squares (LS-C) and Centered Weighted Least Squares (LS-CW) algorithms, and are inspired by the work in [4], [82], [67]. Analytical expressions for the bias and covariance of these algorithms will be derived. It will be shown that, in fact, the LS-C and LS-U yield identical solutions. However, the expressions for the mean and covariance of the LS-C are easier to interpret from a geometrical point of view.

In order to improve the simple (LS-U) algorithms, several refinement methods have been proposed in the TDOA literature to incorporate the dependency between $\mathbf{p}$ and $\|\mathbf{p}\|^2$. These include the celebrated Spherical Interpolation method [155], [156], and the methods presented in [40] and [84]. In [42], an extension to the Lagrange multiplier correction approach of [84] is presented for the TOA problem. The method involves finding the roots of a 5-degree polynomial and a trial and error procedure to determine the correct solution. However, in the recent paper [25] the authors show a simpler way of solving the problem based on Generalized Trust Region Subproblems (GTRS) which will be called GTRS Least Squares (LS-GTRS).

Central to all the Least Squares algorithms is the technique of squaring the Range measurements. In the presence of measurement errors, this will introduce artificial disturbance terms, including the square of the errors, that should be carefully taken into account. Assuming that the measurement error vector $\mathbf{w}$ is Gaussian, what will the effect of the squaring transformation be? Will the resulting squared range measurement errors still be Gaussian and zero mean? Well, in fact neither of them, but fortunately, under some realistic conditions the zero-mean Gaussian approximation remains valid.

2.3.1 On the Squared Range measurements

In what follows we will discuss the nature and statistics of the squared range measurements. Suppose that one has access to a set of noisy range measurements $\bar{r}_i = r_i + w_i; i \in \{1, \ldots, m\}$, written in compact form as $\bar{\mathbf{r}} = \mathbf{r} + \mathbf{w}$, where $\mathbf{w}$ is

Gaussian with covariance $\mathbf{R}$. Define the squared range measurement as

$$\begin{aligned}\bar{d}_i &:= \bar{r}_i^2 = (r_i + w_i)^2 = r_i^2 + 2r_i w_i + w_i^2 = d_i + 2r_i w_i + w_i^2 \\ &= d_i + \xi_i,\end{aligned} \tag{2.3}$$

where $d_i := r_i^2$ and $\xi_i := 2r_i w_i + w_i^2$. Define vectors $\bar{\mathbf{d}} = [\bar{d}_1, \ldots, \bar{d}_m]^T \in \mathbb{R}^m$, $\mathbf{d} = [d_1, \ldots, d_m]^T \in \mathbb{R}^m$, and $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_m]^T \in \mathbb{R}^m$. The squared range measurements can be written in compact form as

$$\bar{\mathbf{d}} = \mathbf{d} + 2\mathbf{r} \odot \mathbf{w} + \mathbf{w} \odot \mathbf{w} = \mathbf{d} + \boldsymbol{\xi}, \tag{2.4}$$

where $\boldsymbol{\xi} = 2\mathbf{r} \odot \mathbf{w} + \mathbf{w} \odot \mathbf{w}$ and, given two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times m}$, $\mathbf{A} \odot \mathbf{B}$ denotes their Hadamard element-wise product [115] defined as

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} a_{11} & \ldots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \ldots & a_{nm} \end{bmatrix} \odot \begin{bmatrix} b_{11} & \ldots & b_{1m} \\ \vdots & \ddots & \vdots \\ b_{n1} & \ldots & b_{nm} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & \ldots & a_{1m}b_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1}b_{n1} & \ldots & a_{nm}b_{nm} \end{bmatrix}. \tag{2.5}$$

A first question to ask is how to compare the original measurement error $\mathbf{w}$ with the artificial error result of squaring the ranges $\boldsymbol{\xi}$. Will it still be zero mean Gaussian? The next proposition characterizes the first two moments of $\boldsymbol{\xi}$:

Proposition 2.3.1 (Characterization of $\boldsymbol{\xi}$)**.** *The mean and covariance of the artificial error $\boldsymbol{\xi}$ are given by*

$$\boldsymbol{\mu}_\xi := E\{\boldsymbol{\xi}\} = \delta(\mathbf{R}), \tag{2.6}$$

$$\boldsymbol{\Sigma}_\xi := E\left\{(\boldsymbol{\xi} - \boldsymbol{\mu}_\xi)(\boldsymbol{\xi} - \boldsymbol{\mu}_\xi)^T\right\} = 4\delta(\mathbf{r})\mathbf{R}\delta(\mathbf{r}) + 2\mathbf{R} \odot \mathbf{R}, \tag{2.7}$$

where given a matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{n \times n}$ and a vector $\mathbf{b} \in \mathbb{R}^n$, the operator $\delta(\cdot)$ is defined as

$$\delta(\mathbf{A}) = \delta\left(\begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \ldots & a_{nn} \end{bmatrix}\right) = \begin{bmatrix} a_{11} \\ \vdots \\ a_{nn} \end{bmatrix} \in \mathbb{R}^n, \tag{2.8}$$

$$\delta(\mathbf{b}) = \delta\left(\begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}\right) = \begin{bmatrix} b_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & b_n \end{bmatrix} \in \mathbb{R}^{n \times n}. \tag{2.9}$$

Note that the operator $\delta(\cdot)$ behaves in a different way if the input is a matrix or a vector and we expect not to cause confusion (it is the *diag* command in MATLAB).

Proof. The computation of the mean yields

$$\mathrm{E}\{\boldsymbol{\xi}\} = \mathrm{E}\{2\mathbf{r}\odot\mathbf{w}+\mathbf{w}\odot\mathbf{w}\} = 2\mathbf{r}\odot\mathrm{E}\{\mathbf{w}\}+\mathrm{E}\{\mathbf{w}\odot\mathbf{w}\} \tag{2.10}$$

$$= \mathrm{E}\left\{\begin{bmatrix} w_1^2 \\ \vdots \\ w_m^2 \end{bmatrix}\right\} = \begin{bmatrix} \sigma_{11}^2 \\ \vdots \\ \sigma_{mm}^2 \end{bmatrix} = \delta(\mathbf{R}), \tag{2.11}$$

because since $\mathrm{E}\{\mathbf{w}\}=0$. The computation of the covariance unfolds in two steps. First, $\mathrm{E}\left\{\boldsymbol{\xi}\boldsymbol{\xi}^T\right\}$ is computed. To this effect, since for any pair of vectors $\mathbf{a},\mathbf{b}\in\mathbb{R}^n$ the identity $\mathbf{a}\odot\mathbf{b}=\delta(\mathbf{a})\mathbf{b}$ applies, then $\boldsymbol{\xi}=2\delta(\mathbf{r})\mathbf{w}+\mathbf{w}\odot\mathbf{w}$. Thus,

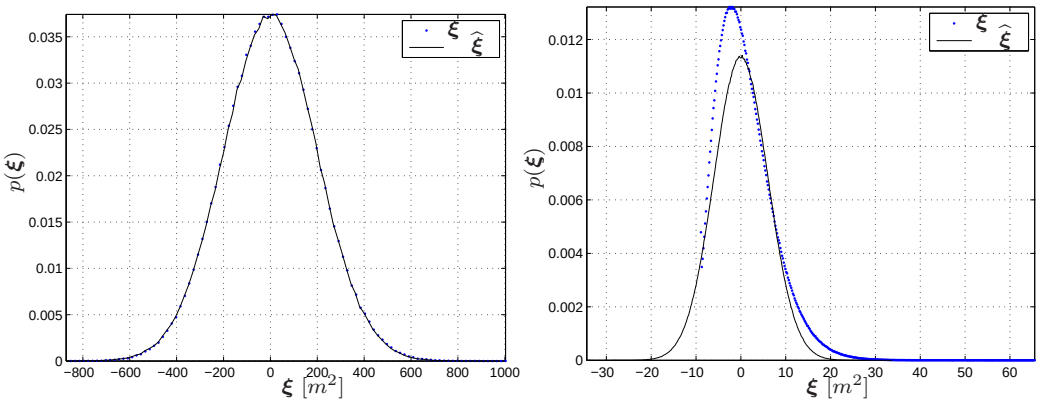
$$\begin{aligned}
\mathrm{E}\left\{\boldsymbol{\xi}\boldsymbol{\xi}^T\right\} &= 4\delta(\mathbf{r})\mathrm{E}\left\{\mathbf{w}\mathbf{w}^T\right\}\delta(\mathbf{r}) + \mathrm{E}\left\{(\mathbf{w}\odot\mathbf{w})(\mathbf{w}\odot\mathbf{w})^T\right\} \\
&\quad + 2\delta(\mathbf{r})\mathrm{E}\left\{\mathbf{w}(\mathbf{w}\odot\mathbf{w})^T\right\} + 2\mathrm{E}\left\{(\mathbf{w}\odot\mathbf{w})\mathbf{w}^T\right\}\delta(\mathbf{r}) \\
&= 4\delta(\mathbf{r})\mathbf{R}\delta(\mathbf{r}) + \mathrm{E}\left\{\begin{bmatrix} w_1^4 & \dots & w_1^2w_m^2 \\ \vdots & \ddots & \vdots \\ w_m^2w_1^2 & \dots & w_m^4 \end{bmatrix}\right\} \\
&= 4\delta(\mathbf{r})\mathbf{R}\delta(\mathbf{r}) + \begin{bmatrix} 3\sigma_{11}^4 & \dots & 2\sigma_{1m}^4+\sigma_{11}^2\sigma_{mm}^2 \\ \vdots & \ddots & \vdots \\ 2\sigma_{m1}^4+\sigma_{11}^2\sigma_{mm}^2 & \dots & 3\sigma_{mm}^4 \end{bmatrix} \\
&= 4\delta(\mathbf{r})\mathbf{R}\delta(\mathbf{r}) + 2\mathbf{R}\odot\mathbf{R} + \delta(\mathbf{R})\delta(\mathbf{R})^T.
\end{aligned} \tag{2.12}$$

It now follows that

$$\begin{aligned}
\boldsymbol{\Sigma}_\xi &= \mathrm{E}\left\{(\boldsymbol{\xi}-\boldsymbol{\mu}_\xi)(\boldsymbol{\xi}-\boldsymbol{\mu}_\xi)^T\right\} = \mathrm{E}\left\{\boldsymbol{\xi}\boldsymbol{\xi}^T\right\} - \boldsymbol{\mu}_\xi\mathrm{E}\left\{\boldsymbol{\xi}^T\right\} - \mathrm{E}\left\{\boldsymbol{\xi}\right\}\boldsymbol{\mu}_\xi^T + \boldsymbol{\mu}_\xi\boldsymbol{\mu}_\xi^T \\
&= \mathrm{E}\left\{\boldsymbol{\xi}\boldsymbol{\xi}^T\right\} - \boldsymbol{\mu}_\xi\boldsymbol{\mu}_\xi^T = \mathrm{E}\left\{\boldsymbol{\xi}\boldsymbol{\xi}^T\right\} - \delta(\mathbf{R})\delta(\mathbf{R})^T \\
&= 4\delta(\mathbf{r})\mathbf{R}\delta(\mathbf{r}) + 2\mathbf{R}\odot\mathbf{R}.
\end{aligned} \tag{2.13}$$

□

Strictly speaking this shows that $\boldsymbol{\xi}$ is neither a zero mean nor a Gaussian random variable. However, in many realistic scenarios it is actually very close to a zero mean Gaussian variable. When the magnitude of the ranges is much bigger than the standard deviation of the measurement errors, that is, when $r_i >> \sigma_{ii}$, we can make the approximation $\boldsymbol{\xi}\approx 2\mathbf{r}\odot\mathbf{w}$ which is a zero mean Gaussian variable. This is actually a quite mild assumption since, if the measurement errors were of the same order of magnitude as the measurements themselves, then the measurements would be of little or no interest at all. Nonetheless, in the case when the vehicle is very close to one of the landmarks, this condition fails and a careful analysis is required. See [5], [40] and [42] for some comments on this issue. In Fig. 2.2 the numerically estimated probability density functions of the actual $\boldsymbol{\xi}=2\mathbf{r}\odot\mathbf{w}+\mathbf{w}\odot\mathbf{w}$ and approximated $\widehat{\boldsymbol{\xi}}:=2\mathbf{r}\odot\mathbf{w}$ are shown for two different scenarios. In the left plot,



(a) $\mathbf{r} = 100\text{m}$, $\sigma = 1\text{m}$ (b) $\mathbf{r} = 3\text{m}$, $\sigma = 1\text{m}$

Figure 2.2: Estimated probability density functions of the actual $\boldsymbol{\xi}$ and approximated $\widehat{\boldsymbol{\xi}}$ squared ranges measurement error for a one dimensional case. The zero mean Gaussian approximation is valid when $\mathbf{r} >> \sigma$.

the assumptions are fulfilled and the actual and approximated artificial errors are almost the same, zero mean, and Gaussian. In the right plot, the range of 3m is very close to the standard deviation of the error, $\sigma = 1\text{m}$, and the approximation is not very accurate, resulting in a biased and non Gaussian error distribution.

2.3.2 Unconstrained Least Squares (LS-U and LS-WU)

Consider the vector of squared range measurements $\bar{\mathbf{d}} = \mathbf{d} + \boldsymbol{\xi} \in \mathbb{R}^m$. Define the matrix $\mathbf{P} \in \mathbb{R}^{n \times m}$ as

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1 & \dots & \mathbf{p}_m \end{bmatrix} \in \mathbb{R}^{n \times m}, \tag{2.14}$$

which contains the landmark coordinates as columns. Each entry of the noise free squared range vector $\mathbf{d}$ can be written as

$$d_i = r_i^2 = \|\mathbf{p} - \mathbf{p}_i\|^2 = (\mathbf{p} - \mathbf{p}_i)^T(\mathbf{p} - \mathbf{p}_i) = \mathbf{p}^T\mathbf{p} - 2\mathbf{p}_i^T\mathbf{p} + \mathbf{p}_i^T\mathbf{p}_i, \tag{2.15}$$

and therefore

$$\begin{aligned} \mathbf{d} = \begin{bmatrix} d_1 \\ \vdots \\ d_m \end{bmatrix} = \begin{bmatrix} \mathbf{p}^T\mathbf{p} - 2\mathbf{p}_1^T\mathbf{p} + \mathbf{p}_1^T\mathbf{p}_1 \\ \vdots \\ \mathbf{p}^T\mathbf{p} - 2\mathbf{p}_m^T\mathbf{p} + \mathbf{p}_m^T\mathbf{p}_m \end{bmatrix} &= \|\mathbf{p}\|^2 \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} - 2 \begin{bmatrix} \mathbf{p}_1^T \\ \vdots \\ \mathbf{p}_m^T \end{bmatrix} \mathbf{p} + \begin{bmatrix} \mathbf{p}_1^T\mathbf{p}_1 \\ \vdots \\ \mathbf{p}_m^T\mathbf{p}_m \end{bmatrix} \\ &= \|\mathbf{p}\|^2\mathbf{1}_m - 2\mathbf{P}^T\mathbf{p} + \delta(\mathbf{P}^T\mathbf{P}). \end{aligned} \tag{2.16}$$

Re-organizing the terms in $\bar{\mathbf{d}} = \mathbf{d} + \boldsymbol{\xi}$ yields

$$2\mathbf{P}^T\mathbf{p} - \|\mathbf{p}\|^2\mathbf{1}_m = \delta(\mathbf{P}^T\mathbf{P}) - \bar{\mathbf{d}} + \boldsymbol{\xi}, \tag{2.17}$$

which can be written as a linear system of the form

$$\mathbf{A}\boldsymbol{\theta} = \mathbf{b} + \boldsymbol{\xi}, \tag{2.18}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n+1}$, $\boldsymbol{\theta} \in \mathbb{R}^{n+1}$, and $\mathbf{b} \in \mathbb{R}^m$ are given by

$$\mathbf{A} = \begin{bmatrix} 2\mathbf{P}^T & -\mathbf{1}_m \end{bmatrix} = \begin{bmatrix} 2\mathbf{p}_1^T & -1 \\ \vdots & \vdots \\ 2\mathbf{p}_1^T & -1 \end{bmatrix}, \quad \boldsymbol{\theta} = \begin{bmatrix} \mathbf{p} \\ \|\mathbf{p}\|^2 \end{bmatrix}, \tag{2.19}$$

$$\mathbf{b} = \delta(\mathbf{P}^T\mathbf{P}) - \bar{\mathbf{d}} = \begin{bmatrix} \|\mathbf{p}_1\|^2 - \bar{d}_1 \\ \vdots \\ \|\mathbf{p}_m\|^2 - \bar{d}_m \end{bmatrix}. \tag{2.20}$$

Note that all the unknowns are contained in the new variable $\boldsymbol{\theta}$. Define the matrix $\mathbf{N} = [\mathbf{I}_n \ \ 0] \in \mathbb{R}^{n\times n+1}$ in such a way that $\mathbf{p} = \mathbf{N}\boldsymbol{\theta}$. One can attempt at solving the previous linear system from a Least Squares point of view neglecting the relationship between the elements in $\boldsymbol{\theta}$, $\mathbf{p}$ and $\|\mathbf{p}\|^2$. This gives rise to the unconstrained minimization problem

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^{n+1}} \|\mathbf{A}\boldsymbol{\theta} - \mathbf{b}\|^2, \tag{2.21}$$

with solution $\boldsymbol{\theta}^* = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$. The vehicle position estimate, obtained by extracting the first n components of $\boldsymbol{\theta}^*$, is given by

$$\boxed{\widehat{\mathbf{p}}_{\text{LS-U}} = \mathbf{N}\boldsymbol{\theta}^* = \mathbf{N}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}.} \tag{2.22}$$

and will be called the *Unconstrained Least Squares* (LS-U) solution to the Range-Only localization problem [42], [25]. Note a similar result is the starting point of of the SX and SI methods for TDOA localization in [152] and [156, 155] .

Instead of using the previous LS solution one can add a weighting matrix $\mathbf{W}$ that takes into account some prior knowledge on the nature of the measurement errors. This is ususally referred as Generalized or Weighted Least Squares estimation [7] [18]. Consider a positive definite weighting matrix $\mathbf{W} \in \mathbb{R}^{m\times m}$ and the unconstrained minimization problem

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^{n+1}} (\mathbf{A}\boldsymbol{\theta} - \mathbf{b})^T\mathbf{W}(\mathbf{A}\boldsymbol{\theta} - \mathbf{b}), \tag{2.23}$$

with solution $\boldsymbol{\theta}^* = (\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}\mathbf{b}$. The vehicle position estimate obtained by extracting the first n components of $\boldsymbol{\theta}^*$ is given by

$$\boxed{\widehat{\mathbf{p}}_{\text{LS-UW}} = \mathbf{N}(\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}\mathbf{b},} \tag{2.24}$$

and will be called the *Unconstrained Weighted Least Squares* (LS-UW) solution to the Range-Only localization problem. The most natural choice for the weighting matrix, which approximates the Maximum Likelihood criteria, is $\mathbf{W} = \mathbf{\Sigma}_{\xi}^{-1}$ where $\mathbf{\Sigma}_{\xi}$ is the covariance matrix of the squared measurement error $\boldsymbol{\xi}$ in (2.7). Since the actual ranges $\mathbf{r}$ are unknown, it is common to use the approximation $\mathbf{\Sigma}_{\xi} \approx 4\delta(\bar{\mathbf{r}})\mathbf{R}\delta(\bar{\mathbf{r}})$ as in [42].

LS-U and LS-UW estimation error characterization

It is possible to derive expressions for the mean and covariance of the position estimation errors associated to the LS-U and LS-UW solutions. We will discuss only the weighted solution, as the unweighted one can then be directly derived by setting $\mathbf{W} = \mathbf{I}_m$. Define the estimation error as $\widetilde{\mathbf{p}}_{\text{LS-UW}} = \mathbf{p} - \widehat{\mathbf{p}}_{\text{LS-UW}}$. where $\widehat{\mathbf{p}}_{\text{LS-UW}}$ is the LS-UW estimate. It can be shown that the estimation error can be written as $\widetilde{\mathbf{p}}_{\text{LS-UW}} = \mathbf{N}(\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}\boldsymbol{\xi}$ which has mean

$$\boldsymbol{\mu}_{\text{LS-UW}} = \mathrm{E}\left\{\widetilde{\mathbf{p}}_{\text{LS-UW}}\right\} = \mathbf{N}(\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}\delta(\mathbf{R}), \tag{2.25}$$

and covariance approximately given by

$$\begin{aligned}\mathbf{\Sigma}_{\text{LS-UW}} &= \mathrm{E}\left\{(\widetilde{\mathbf{p}}_{\text{LS-UW}} - \boldsymbol{\mu}_{\text{LS-UW}})(\widetilde{\mathbf{p}}_{\text{LS-UW}} - \boldsymbol{\mu}_{\text{LS-UW}})^T\right\} \approx \mathrm{E}\left\{\widetilde{\mathbf{p}}_{\text{LS-UW}}\widetilde{\mathbf{p}}_{\text{LS-UW}}^T\right\} \\ &= \mathbf{N}(\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}\mathrm{E}\left\{\boldsymbol{\xi}\boldsymbol{\xi}^T\right\}\mathbf{W}\mathbf{A}^T(\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{N}^T \\ &= \mathbf{N}(\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}\mathbf{\Sigma}_{\boldsymbol{\xi}}\mathbf{W}\mathbf{A}(\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{N}^T,\end{aligned} \tag{2.26}$$

which further simplifies to $\mathbf{\Sigma}_{\text{LS-UW}} \approx \mathbf{N}(\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{N}^T$ when the weighting matrix is taken as $\mathbf{W} = (4\delta(\bar{\mathbf{r}})\mathbf{R}\delta(\bar{\mathbf{r}}))^{-1} \approx \mathbf{\Sigma}_{\boldsymbol{\xi}}^{-1}$.

2.3.3 Centered Least Squares (LS-C and LS-CW)

Another approach to solve the Range-Only localization problem consists of eliminating the unknown term in $\|\mathbf{p}\|^2$ from the squared range equations by using an appropriate projection matrix. This method will be referred as Centered Least Squares (LS-C) and Centered Weighted Least Squares (LS-C) and is inspired by the work in [82] and the TDOA methods proposed in [2] and [67]. Expressions for the bias and error covariance of these algorithms will be derived that can be easily interpreted from a geometrical point of view. Moreover, it will be shown that the solutions of these algorithms are identical to the previous LS-U and LS-UW.

The starting point of the Least Squares algorithms is the linear equation in (2.17) which contained an unknown term $\|\mathbf{p}\|^2\mathbf{1}_m$ on the left hand side. Now, instead of solving for both $\mathbf{p}$ and $\|\mathbf{p}\|^2$ we will try to get rid of the latter. This can be done by using a matrix $\mathbf{M} \in \mathbb{R}^{m\times m}$ which has $\mathbf{1}_m$ on its null space. Matrix $\mathbf{M}$ can be chosen

for instance as

$$\mathbf{M} = \mathbf{I}_m - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^T = \frac{1}{m}\begin{bmatrix} m-1 & -1 & \dots & -1 \\ -1 & m-1 & & \vdots \\ \vdots & & \ddots & -1 \\ -1 & \dots & -1 & m-1 \end{bmatrix} \in \mathbb{R}^{m\times m}. \tag{2.27}$$

Note that, in this case, $\mathbf{M}$ is a projection matrix and satisfies $\mathbf{M}\mathbf{1}_m = 0$, $\mathbf{M} = \mathbf{M}^T$, and $\mathbf{M}\mathbf{M} = \mathbf{M}$. The term centered is used because the projection matrix $\mathbf{M}$ is a centering operator. For instance, the matrix $\mathbf{P}_c := \mathbf{M}\mathbf{P}$ contains the centered landmark coordinates. That is, its columns contain the landmark coordinates expressed with respect to a reference frame with the same orientation as $\{\mathcal{I}\}$ and with origin at the landmarks centroid. In order to show this, let $\boldsymbol{\chi} \in \mathbb{R}^n$ be the vector containing the coordinates of the centroid of the landmarks, that is,

$$\boldsymbol{\chi} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{p}_i = \frac{1}{m}\mathbf{P}\mathbf{1}_m, \tag{2.28}$$

and note that

$$\begin{aligned}\mathbf{P}_c = \mathbf{P}\mathbf{M} &= \mathbf{P}(\mathbf{I}_m - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^T) = \mathbf{P} - \frac{1}{m}\mathbf{P}\mathbf{1}_m\mathbf{1}_m^T = \mathbf{P} - \boldsymbol{\chi}\mathbf{1}_m^T \\ &= \begin{bmatrix}\mathbf{p}_1 & \dots & \mathbf{p}_m\end{bmatrix} - \begin{bmatrix}\boldsymbol{\chi} & \cdots & \boldsymbol{\chi}\end{bmatrix} = \begin{bmatrix}\mathbf{p}_1 - \boldsymbol{\chi} & \cdots & \mathbf{p}_m - \boldsymbol{\chi}\end{bmatrix}.\end{aligned} \tag{2.29}$$

Multiplying both sides of (2.17) by the centering matrix $\mathbf{M}$ on the left and reordering yields

$$2\mathbf{M}\mathbf{P}^T\mathbf{p} = \mathbf{M}(\delta(\mathbf{P}^T\mathbf{P}) - \bar{\mathbf{d}}) + \mathbf{M}\boldsymbol{\xi}. \tag{2.30}$$

This equation can be written as a linear system of the form $\mathbf{A}\mathbf{p} = \mathbf{b} + \boldsymbol{\epsilon}$, where

$$\begin{aligned}\mathbf{A} &= 2\mathbf{M}\mathbf{P}^T \in \mathbb{R}^{m\times n}, \\ \mathbf{b} &= \mathbf{M}(\delta(\mathbf{P}^T\mathbf{P}) - \bar{\mathbf{d}}) \in \mathbb{R}^m, \\ \boldsymbol{\epsilon} &= \mathbf{M}\boldsymbol{\xi} \in \mathbb{R}^m.\end{aligned} \tag{2.31}$$

The system can be solved from a Least Squares point of view as $\widehat{\mathbf{p}}_{\text{LS-C}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$ which can be also written as

$$\boxed{\widehat{\mathbf{p}}_{\text{LS-C}} = \boldsymbol{\Theta}(\delta(\mathbf{P}^T\mathbf{P}) - \bar{\mathbf{d}}), \quad \boldsymbol{\Theta} = \frac{1}{2}(\mathbf{P}_c\mathbf{P}_c^T)^{-1}\mathbf{P}_c.} \tag{2.32}$$

This will be called the *Centered Least Squares* (LS-C) solution to the Range-Only localization problem. This solution is inspired in [4],[82], and [67]. Note that in the previous expression we used the property that $\mathbf{M}^T\mathbf{M} = \mathbf{M}$ to simplify $\boldsymbol{\Theta}\mathbf{M} = \boldsymbol{\Theta}$.

At this point, an important question arises: Is it possible to determine under which conditions the system has a unique solution? It is also interesting to find out if the conditions derived can be easily interpreted from a geometric point of view, in terms of the number of landmarks required and their relative geometry. In order for the LS-C solution to be well defined, one needs to show that $\mathbf{P}_c^T\mathbf{P}_c$ is invertible. The following is a well known result:

Proposition 2.3.2. *The LS-C solution is well defined if and only if there is a set of at least $n+1$ landmarks which do not lie on an affine lower dimension subspace of $\mathbb{R}^n$. In the two dimensional case ($n=2$) a set of at least 3 non collinear landmarks are required and in the three dimensional case ($n=3$) a set of at least 4 non coplanar landmarks are required.*

Proof. Matrix $\mathbf{P}_c\mathbf{P}_c^T \in \mathbb{R}^{n\times n}$ is invertible if and only if matrix $\mathbf{P}_c \in \mathbb{R}^{n\times m}$ has full column rank n. Hence, at least one must have $m \geq n$. If all the landmarks lie on an affine subspace of dimension $k < n$, then their centered version with respect to their centroid $\boldsymbol{\chi}$ yields a set of linearly dependent vectors and matrix $\mathbf{P}_c$ would have rank $k < n$. Moreover, a set of n points in $\mathbb{R}^n$ always define an affine proper subspace of dimension $n-1$. Hence, in order to have a rank n matrix $\mathbf{P}_c$ it is necessary to have at least a set of $n+1$ landmarks which do not lie on an affine proper subspace. □

Mimicking what was done in the LS-UW solution, it is desirable to incorporate a weighting matrix $\mathbf{W}$ for better performance. However, the natural choice of using the inverse of the covariance of the error $\boldsymbol{\epsilon}$ can not be applied here since in this case the covariance matrix is singular. Note from (2.30), that the equation error becomes $\boldsymbol{\epsilon} = \mathbf{M}\boldsymbol{\xi}$ and its covariance $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} = \mathrm{E}\left\{\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T\right\} = \mathbf{M}\mathrm{E}\left\{\boldsymbol{\xi}\boldsymbol{\xi}^T\right\}\mathbf{M} = \mathbf{M}\boldsymbol{\Sigma}_{\boldsymbol{\xi}}\mathbf{M}$ (2.7). Under the assumption that the ranges satisfy $r_i >> \sigma_{ii}$ one can make the approximation $\boldsymbol{\Sigma}_{\boldsymbol{\xi}} \approx 4\delta(\mathbf{r})\mathbf{R}\delta(\mathbf{r})$ such that $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} \approx 4\mathbf{M}\delta(\mathbf{r})\mathbf{R}\delta(\mathbf{r})\mathbf{M}$. The true range vector $\mathbf{r}$ is unknown, and only the measured one $\bar{\mathbf{r}}$ is available. Define the matrix

$$\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\epsilon}} := 4\mathbf{M}\delta(\bar{\mathbf{r}})\mathbf{R}\delta(\bar{\mathbf{r}})\mathbf{M}, \tag{2.33}$$

which, under the assumption $r_i >> \sigma_{ii}$, satisfies $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\epsilon}} \approx \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}$. Note that since $\mathbf{M}$ is rank deficient (it has rank $m-1$), the matrix $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\epsilon}}$ is singular and positive semidefinite. This means that one cannot choose directly the weighting matrix as the inverse of $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\epsilon}}$, since it does not exist. Instead, it is possible to use its pseudoinverse $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\epsilon}}^{\dagger}$, defined below.

Definition 2.3.1 (Pseudoinverse of a square matrix)**.** Given a square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, its pseudoinverse is the unique matrix $\mathbf{A}^{\dagger} \in \mathbb{R}^{n\times n}$ that satisfies $\mathbf{A}\mathbf{A}^{\dagger}\mathbf{A} = \mathbf{A}$ and $\mathbf{A}^{\dagger}\mathbf{A}\mathbf{A}^{\dagger} = \mathbf{A}^{\dagger}$ and can be computed as follows. Let $\mathbf{A}$ admit the eigenvalue

decomposition

$$\mathbf{A} = \mathbf{V} \begin{bmatrix} \lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_n \end{bmatrix} \mathbf{V}^{-1}. \tag{2.34}$$

Then, its pseudoinverse is defined by

$$\mathbf{A}^\dagger = \mathbf{V} \begin{bmatrix} \theta_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \theta_n \end{bmatrix} \mathbf{V}^{-1}, \quad \theta_i = \begin{cases} 1/\lambda_i & \text{if } \lambda_i \neq 0, \\ 0 & \text{if } \lambda_i = 0. \end{cases} \tag{2.35}$$

We can now introduce the *Centered Weighted Least Squares* solution of the Range-Only localization problem as follows:

$$\boxed{\widehat{\mathbf{p}}_{\text{LS-CW}} = \mathbf{\Theta}_W(\delta(\mathbf{P}^T\mathbf{P}) - \bar{\mathbf{d}}),} \tag{2.36}$$

where

$$\mathbf{\Theta}_W = \frac{1}{2}(\mathbf{P}_c\widehat{\mathbf{\Sigma}}_{\boldsymbol{\epsilon}}^\dagger\mathbf{P}_c^T)^{-1}\mathbf{P}_c\widehat{\mathbf{\Sigma}}_{\boldsymbol{\epsilon}}^\dagger, \tag{2.37}$$

$$\widehat{\mathbf{\Sigma}}_{\boldsymbol{\epsilon}} = 4\mathbf{M}\delta(\bar{\mathbf{r}})\mathbf{R}\delta(\bar{\mathbf{r}})\mathbf{M}. \tag{2.38}$$

To derive the previous expression we used the fact that $\mathbf{\Theta}_W\mathbf{M} = \mathbf{\Theta}_W$. In order for the LS-CW solution to be defined it is necessary that the matrix $(\mathbf{P}_c\widehat{\mathbf{\Sigma}}_{\boldsymbol{\epsilon}}^\dagger\mathbf{P}_c^T)$ be invertible. Conditions under which this is true are given next:

Proposition 2.3.3. *Suppose that $\mathbf{P}_c$ is full column rank n and that none of the measured ranges $\bar{r}_i$ are zero. Then, the matrix $(\mathbf{P}_c\widehat{\mathbf{\Sigma}}_{\boldsymbol{\epsilon}}^\dagger\mathbf{P}_c^T) \in \mathbb{R}^{n\times n}$ is invertible.*

Proof. Note that if all the entries of $\bar{\mathbf{r}}$ are non zero, then $\delta(\bar{\mathbf{r}})\mathbf{R}\delta(\bar{\mathbf{r}})$ is positive definite. Furthermore, matrix $\mathbf{M} \in \mathbb{R}^{m\times m}$ has rank $m-1$ and has the vector of ones $\mathbf{1}_m$ in its null space. By definition, $\widehat{\mathbf{\Sigma}}_{\boldsymbol{\epsilon}}$ is symmetric and also has $\mathbf{1}_m$ in its null space. Therefore, matrix $\widehat{\mathbf{\Sigma}}_{\boldsymbol{\epsilon}}$ admits the decomposition

$$\begin{aligned} \widehat{\mathbf{\Sigma}}_{\boldsymbol{\epsilon}} := \mathbf{V}\mathbf{D}\mathbf{V}^T &= \mathbf{V} \begin{bmatrix} \lambda_1 & \dots & \dots & 0 \\ \vdots & \ddots & & \vdots \\ \vdots & & \lambda_{m-1} & \vdots \\ 0 & \dots & \dots & 0 \end{bmatrix} \mathbf{V}^T := \begin{bmatrix} \bar{\mathbf{V}} & \mathbf{1}_m \end{bmatrix} \begin{bmatrix} \bar{\mathbf{D}} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{\mathbf{V}}^T \\ \mathbf{1}_m^T \end{bmatrix} \\ &= \bar{\mathbf{V}}\bar{\mathbf{D}}\bar{\mathbf{V}}^T, \end{aligned} \tag{2.39}$$

where $\lambda_1 \geq \dots \geq \lambda_{m-1} > 0$. The columns of $\bar{\mathbf{V}} \in \mathbb{R}^{m\times m-1}$ form an orthonormal basis for the orthogonal complement $\mathbf{1}_m^\perp$ such that $\bar{\mathbf{V}}^T\bar{\mathbf{V}} = \mathbf{I}_{m-1}$ and $\bar{\mathbf{V}}^T\mathbf{1}_m = 0$.

The pseudoinverse of $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\epsilon}}$ can be written as

$$\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\epsilon}}^{\dagger} = \mathbf{V} \begin{bmatrix} \frac{1}{\lambda_1} & \dots & \dots & 0 \\ \vdots & \ddots & & \vdots \\ \vdots & & \frac{1}{\lambda_{m-1}} & \vdots \\ 0 & \dots & \dots & 0 \end{bmatrix} \mathbf{V}^T = \begin{bmatrix} \bar{\mathbf{V}} & \mathbf{1}_m \end{bmatrix} \begin{bmatrix} \bar{\mathbf{D}}^{-1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{\mathbf{V}}^T \\ \mathbf{1}_m^T \end{bmatrix} \tag{2.40}$$

$$= \bar{\mathbf{V}} \bar{\mathbf{D}}^{-1} \bar{\mathbf{V}}^T. \tag{2.41}$$

Moreover, since $\mathbf{P}_c \mathbf{1}_m = 0$, the columns of $\mathbf{P}_c^T$ belong to the orthogonal complement $\mathbf{1}_m^{\perp}$, and can be written as a linear combination of the columns of $\bar{\mathbf{V}}$. That is, one can write $\mathbf{P}_c^T = \bar{\mathbf{V}}\mathbf{B}$ where $\mathbf{B} \in \mathbb{R}^{m-1 \times n}$ is full rank if $\mathbf{P}_c$ is full rank n. Therefore,

$$\mathbf{P}_c \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\epsilon}}^{\dagger} \mathbf{P}_c^T = \mathbf{B}^T \bar{\mathbf{V}}^T \bar{\mathbf{V}} \bar{\mathbf{D}}^{-1} \bar{\mathbf{V}}^T \bar{\mathbf{V}} \mathbf{B} = \mathbf{B}^T \bar{\mathbf{D}}^{-1} \mathbf{B}, \tag{2.42}$$

is positive definite and nonsingular since $\mathbf{B}$ is full column rank and $\bar{\mathbf{D}}^{-1}$ is positive definite. □

Proposition 2.3.4. *Suppose that $\mathbf{P}_c \in \mathbb{R}^{n \times m}$ has full column rank n and that there are exactly $m = n+1$ landmarks. Then the LS-C and the LS-CW estimates of (2.32) and (2.36) are equivalent.*

Proof. The LS-C and LS-CW estimates are defined as $\widehat{\mathbf{p}}_{\text{LS-C}} = \boldsymbol{\Theta}(\delta(\mathbf{P}^T\mathbf{P}) - \bar{\mathbf{d}})$ and $\widehat{\mathbf{p}}_{\text{LS-CW}} = \boldsymbol{\Theta}_W(\delta(\mathbf{P}^T\mathbf{P}) - \bar{\mathbf{d}})$, respectively. It is enough to show that when $m = n+1$ and $\mathbf{P}_c$ is full rank $\boldsymbol{\Theta} = \boldsymbol{\Theta}_W$, where

$$\boldsymbol{\Theta} = \frac{1}{2}(\mathbf{P}_c\mathbf{P}_c)^{-1}\mathbf{P}_c, \tag{2.43}$$

$$\boldsymbol{\Theta}_W = \frac{1}{2}(\mathbf{P}_c \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\epsilon}}^{\dagger} \mathbf{P}_c^T)^{-1} \mathbf{P}_c \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\epsilon}}^{\dagger}. \tag{2.44}$$

Consider the decomposition $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\epsilon}}^{\dagger} = \bar{\mathbf{V}} \bar{\mathbf{D}}^{-1} \bar{\mathbf{V}}^T$ of (2.41) where the columns of $\bar{\mathbf{V}} \in \mathbb{R}^{n+1 \times n}$ form an orthonormal basis for the orthogonal complement to $\mathbf{1}_m$. Note that $\mathbf{P}_c = \mathbf{B}^T \bar{\mathbf{V}}^T$ where now $\mathbf{B} \in \mathbb{R}^{n \times n}$ is square and nonsingular. It then follows that

$$\begin{aligned} (\mathbf{P}_c\mathbf{P}_c)^{-1}\mathbf{P}_c &= (\mathbf{B}^T\bar{\mathbf{V}}^T\bar{\mathbf{V}}\mathbf{B})^{-1}\mathbf{B}^T\bar{\mathbf{V}}^T = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\bar{\mathbf{V}}^T \\ &= \mathbf{B}^{-1}\mathbf{B}^{-T}\mathbf{B}^T\bar{\mathbf{V}}^T = \mathbf{B}^{-1}\bar{\mathbf{V}}^T, \end{aligned} \tag{2.45}$$

and

$$\begin{aligned} (\mathbf{P}_c \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\epsilon}}^{\dagger} \mathbf{P}_c^T)^{-1} \mathbf{P}_c \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{\epsilon}}^{\dagger} &= (\mathbf{B}^T \bar{\mathbf{V}}^T \bar{\mathbf{V}} \bar{\mathbf{D}}^{-1} \bar{\mathbf{V}}^T \bar{\mathbf{V}} \mathbf{B})^{-1} \mathbf{B}^T \bar{\mathbf{V}}^T \bar{\mathbf{V}} \bar{\mathbf{D}}^{-1} \bar{\mathbf{V}}^T \\ &= (\mathbf{B}^T \bar{\mathbf{D}}^{-1} \mathbf{B})^{-1} \mathbf{B}^T \bar{\mathbf{D}}^{-1} \bar{\mathbf{V}}^T \\ &= \mathbf{B}^{-1} \bar{\mathbf{D}} \mathbf{B}^{-T} \mathbf{B}^T \bar{\mathbf{D}}^{-1} \bar{\mathbf{V}}^T \\ &= \mathbf{B}^{-1} \bar{\mathbf{V}}^T, \end{aligned} \tag{2.46}$$

which shows that $\boldsymbol{\Theta} = \boldsymbol{\Theta}_W = \frac{1}{2}\mathbf{B}^{-1}\bar{\mathbf{V}}^T$ and the equivalence of the LS-C and LS-CW estimators. □

LS-C and LS-CW estimation error characterization

We next characterize the mean and covariance of the estimation errors associated with the LS-C and LS-CW solutions $\widetilde{\mathbf{p}}_{\text{LS-C}} := \mathbf{p} - \widehat{\mathbf{p}}_{\text{LS-C}}$ and $\widetilde{\mathbf{p}}_{\text{LS-CW}} := \mathbf{p} - \widehat{\mathbf{p}}_{\text{LS-CW}}$ respectively.

The LS-C estimation error has mean

$$\boldsymbol{\mu}_{\text{LS-C}} := \mathrm{E}\left\{\widetilde{\mathbf{p}}_{\text{LS-C}}\right\} = \mathrm{E}\left\{\boldsymbol{\Theta}\boldsymbol{\epsilon}\right\} = \boldsymbol{\Theta}\mathrm{E}\left\{\mathbf{M}\boldsymbol{\xi}\right\} = \boldsymbol{\Theta}\mathbf{M}\delta(\mathbf{R}) = \boldsymbol{\Theta}\delta(\mathbf{R}), \tag{2.47}$$

and covariance

$$\begin{aligned}
\boldsymbol{\Sigma}_{\text{LS-C}} &:= \mathrm{E}\left\{(\widetilde{\mathbf{p}}_{\text{LS-C}} - \boldsymbol{\mu}_{\text{LS-C}})(\widetilde{\mathbf{p}}_{\text{LS-C}} - \boldsymbol{\mu}_{\text{LS-C}})^T\right\} = \mathrm{E}\left\{\widetilde{\mathbf{p}}_{\text{LS-C}}\widetilde{\mathbf{p}}_{\text{LS-C}}^T\right\} - \boldsymbol{\mu}_{\text{LS-C}}\boldsymbol{\mu}_{\text{LS-C}}^T \\
&= \mathrm{E}\left\{\boldsymbol{\Theta}\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T\boldsymbol{\Theta}^T\right\} + \boldsymbol{\Theta}\delta(\mathbf{R})\delta(\mathbf{R})^T\boldsymbol{\Theta}^T = \boldsymbol{\Theta}\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\boldsymbol{\Theta}^T + \boldsymbol{\Theta}\delta(\mathbf{R})^2\boldsymbol{\Theta}^T \\
&= \boldsymbol{\Theta}\mathbf{M}\boldsymbol{\Sigma}_{\boldsymbol{\xi}}\mathbf{M}\boldsymbol{\Theta}^T + \boldsymbol{\Theta}\delta(\mathbf{R})^2\boldsymbol{\Theta}^T = \boldsymbol{\Theta}(\boldsymbol{\Sigma}_{\boldsymbol{\xi}} + \delta(\mathbf{R})^2)\boldsymbol{\Theta}^T, \\
&= \boldsymbol{\Theta}\left(4\delta(\mathbf{r})\mathbf{R}\delta(\mathbf{r}) + 2\mathbf{R}\odot\mathbf{R} + 2\delta(\mathbf{R})^2\right)\boldsymbol{\Theta}^T
\end{aligned} \tag{2.48}$$

which can be approximated by $\boldsymbol{\Sigma}_{\text{LS-C}} \approx 4\boldsymbol{\Theta}\delta(\mathbf{r})\mathbf{R}\delta(\mathbf{r})\boldsymbol{\Theta}^T$ under the assumption that $r_i >> \sigma_{ii}$, i.e., loosely speaking when the entries of $\mathbf{r}$ are much bigger than those of $\mathbf{R}$. Using these results it is possible to estimate a priori what will be the expected error covariance and bias associated with the LS-C algorithm. The size of the bias will be strongly determined by the singular values of matrix $\boldsymbol{\Theta} = \frac{1}{2}(\mathbf{P}_c\mathbf{P}_c^T)^{-1}\mathbf{P}_c$, which in turn depend on the singular values of the matrix $\mathbf{P}_c$. In fact, from (2.47) one has that

$$\|\boldsymbol{\mu}_{\text{LS-C}}\| \leq \|\boldsymbol{\Theta}\|\|\delta(\mathbf{R})\| = \frac{1}{2\underline{\sigma}(\mathbf{P}_c)}\|\delta(\mathbf{R})\|, \tag{2.49}$$

where $\underline{\sigma}(\mathbf{P}_c)$ denotes the minimum singular value of $\mathbf{P}_c$. This matrix, which describes the relative geometry of the landmarks, has a crucial role in the trilateration estimation performance which will be discussed on a later section.

The bias and covariance associated with the LS-CW solution can be directly obtained from the previous results by replacing $\boldsymbol{\Theta}$ with $\boldsymbol{\Theta}_W$. That is,

$$\boldsymbol{\mu}_{\text{LS-CW}} = \boldsymbol{\Theta}_W\delta(\mathbf{R}), \tag{2.50}$$

$$\begin{aligned}
\boldsymbol{\Sigma}_{\text{LS-CW}} &= \boldsymbol{\Theta}_W\left(4\delta(\mathbf{r})\mathbf{R}\delta(\mathbf{r}) + 2\mathbf{R}\odot\mathbf{R} + 2\delta(\mathbf{R})^2\right)\boldsymbol{\Theta}_W^T \\
&\approx 4\boldsymbol{\Theta}_W\delta(\mathbf{r})\mathbf{R}\delta(\mathbf{r})\boldsymbol{\Theta}_W^T.
\end{aligned} \tag{2.51}$$

The Centered Least Squares (LS-C) algorithm has some features that make it interesting. It is computationally simple, thus making it suitable for implementation with low computational power hardware. When the landmarks are fixed so that $\mathbf{P}_c$

is constant over time, $\mathbf{\Theta}$ remains also constant and can be computed a priori. The computation of a trilateration solution can then be done in an efficient manner involving basically only one matrix multiplication and a few additions. As will be discussed later, the price to pay for this simplicity is poorer performance as compared to that achieved with more complex algorithms. The LS-CW algorithm is slightly more complex than the simpler non weighted version LS-C, but as will be shown later its performance its far superior being comparable to other much more complex algorithms. Unlike in the unweighted case, now the matrix $\mathbf{\Theta}_W$ depends on the current set of range measurements and cannot therefore be computed a priori.

Proposition 2.3.5. *The LS-U and LS-C solutions to the Range-Only localization problem are equivalent.*

Proof. We need to show that $\mathbf{N}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A} = \mathbf{\Theta}$, where $\mathbf{A} = [2\mathbf{P}^T - \mathbf{1}_m]$, $\mathbf{N} = [\mathbf{I}_n\ \mathbf{0}_{n\times 1}]$, and $\mathbf{\Theta} = \frac{1}{2}(\mathbf{P}_c\mathbf{P}_c^T)\mathbf{P}_c = \frac{1}{2}(\mathbf{P}\mathbf{M}\mathbf{P}^T)\mathbf{P}\mathbf{M}$. Note that

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} 2\mathbf{P} \\ -\mathbf{1}_m^T \end{bmatrix}\begin{bmatrix} 2\mathbf{P}^T & -\mathbf{1}_m^T \end{bmatrix} = \begin{bmatrix} 4\mathbf{P}\mathbf{P}^T & -2\mathbf{P}\mathbf{1}_m \\ -2\mathbf{1}_m^T\mathbf{P}^T & m \end{bmatrix}. \tag{2.52}$$

A well known property of the inverse of a partitioned matrix is that [115]

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} (\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & -(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}\mathbf{B}\mathbf{D}^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & (\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{bmatrix}, \tag{2.53}$$

whenever $\mathbf{D}$ and $\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C}$ are invertible. Using this and the fact that

$$(\mathbf{P}\mathbf{P}^T - \mathbf{P}\mathbf{1}_m\frac{1}{m}\mathbf{1}_m^T\mathbf{P}^T) = \mathbf{P}(\mathbf{I}_m - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^T)\mathbf{P}^T = \mathbf{P}\mathbf{M}\mathbf{P}^T = \mathbf{P}_c\mathbf{P}_c^T, \tag{2.54}$$

we have that

$$\begin{bmatrix} 4\mathbf{P}\mathbf{P}^T & -2\mathbf{P}\mathbf{1}_m \\ -2\mathbf{1}_m^T\mathbf{P}^T & m \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{4}(\mathbf{P}_c\mathbf{P}_c^T)^{-1} & \frac{1}{2m}(\mathbf{P}_c\mathbf{P}_c^T)^{-1}\mathbf{P}\mathbf{1}_m \\ \frac{1}{2m}\mathbf{1}_m^T\mathbf{P}^T(\mathbf{P}_c\mathbf{P}_c^T)^{-1} & (m-\mathbf{1}_m^T\mathbf{P}^T(\mathbf{P}\mathbf{P}^T)^{-1}\mathbf{P}\mathbf{1}_m)^{-1} \end{bmatrix}, \tag{2.55}$$

and

$$\mathbf{N}(\mathbf{A}^T\mathbf{A})^{-1} = \begin{bmatrix} \frac{1}{4}(\mathbf{P}_c\mathbf{P}_c^T)^{-1} & \frac{1}{2m}(\mathbf{P}_c\mathbf{P}_c^T)^{-1}\mathbf{P}\mathbf{1}_m \end{bmatrix}. \tag{2.56}$$

Therefore,

$$\begin{aligned}\mathbf{N}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T &= \begin{bmatrix}\frac{1}{4}(\mathbf{P}_c\mathbf{P}_c^T)^{-1} & \frac{1}{2m}(\mathbf{P}_c\mathbf{P}_c^T)^{-1}\mathbf{P}\mathbf{1}_m\end{bmatrix}\begin{bmatrix}2\mathbf{P}\\ -\mathbf{1}_m^T\end{bmatrix}\\ &= \frac{1}{2}(\mathbf{P}_c\mathbf{P}_c^T)^{-1}\mathbf{P} - \frac{1}{2m}(\mathbf{P}_c\mathbf{P}_c^T)^{-1}\mathbf{P}\mathbf{1}_m\mathbf{1}_m^T\\ &= \frac{1}{2}(\mathbf{P}_c\mathbf{P}_c^T)^{-1}\mathbf{P}(\mathbf{I}_m - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^T)\\ &= \frac{1}{2}(\mathbf{P}_c\mathbf{P}_c^T)^{-1}\mathbf{P}_c = \mathbf{\Theta}. \end{aligned} \tag{2.57}$$

thus concluding the proof. □

2.3.4 GTRS Least Squares (LS-GTRS)

The Unconstrained Least Squares (LS-U) solution was based on manipulating the squared range equations to obtain a linear system of equations in $\mathbf{p}$ and its square norm $\|\mathbf{p}\|^2$. Ignoring the relationship between them, an unconstrained minimization problem was formulated in the new variable $\boldsymbol{\theta} = [\mathbf{p}^T \ \ \|\mathbf{p}\|^2]^T \in \mathbb{R}^{n+1}$. The position estimate was then given by the first n elements of its solution $\boldsymbol{\theta}^* = [\widehat{\mathbf{p}}^T \ c]^T$ in (2.21). It is then natural to expect that the resulting solution will not satisfy the constraint $c = \|\widehat{\mathbf{p}}\|^2$.

Several methods have been proposed in the literature which, starting from an unconstrained LS-U solution, consider an additional correction step in order to take the constraint into account. In the closely related TDOA localization literature, the Spherical Intersection (SX) [152] and the Spherical Intersection (SI) [4, 155, 156] were the first to use this approach. Later in [40] and [84] more elaborate and numerically involved correction steps were proposed. The Lagrange multiplier correction method of [84] was extended to the TOA problem in [42]. Recently, the authors in [25] proposed an equivalent much simpler method in terms of Generalized Trust Region Subproblems (GTRS) which is next briefly summarized.

Define the matrices

$$\mathbf{A} = \begin{bmatrix}2\mathbf{P}^T & -\mathbf{1}_m\end{bmatrix} = \begin{bmatrix}2\mathbf{p}_1^T & -1\\ \vdots & \vdots\\ 2\mathbf{p}_1^T & -1\end{bmatrix}, \quad \boldsymbol{\theta} = \begin{bmatrix}\mathbf{p}\\ \|\mathbf{p}\|^2\end{bmatrix}, \tag{2.58}$$

$$\mathbf{b} = \delta(\mathbf{P}^T\mathbf{P}) - \bar{\mathbf{d}} = \begin{bmatrix}\|\mathbf{p}_1\|^2 - \bar{d}_1\\ \vdots\\ \|\mathbf{p}_m\|^2 - \bar{d}_m\end{bmatrix}, \tag{2.59}$$

$$\mathbf{N} = \begin{bmatrix}\mathbf{I}_n & \mathbf{0}_{n\times 1}\end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix}\mathbf{0}_{n\times 1}\\ -1\end{bmatrix}. \tag{2.60}$$

The dependency between the components of $\boldsymbol{\theta}$ can be written as $\|\mathbf{N}\boldsymbol{\theta}\|^2 + \mathbf{f}^T\boldsymbol{\theta} =$

0, or equivalently, $\boldsymbol{\theta}^T\mathbf{N}^T\mathbf{N}\boldsymbol{\theta} + \mathbf{f}^T\boldsymbol{\theta} = 0$. Then one can consider the constrained minimization problem

$$\begin{aligned}\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}\in\mathbb{R}^{n+1}} & \|\mathbf{A}\boldsymbol{\theta} - \mathbf{b}\|^2 \\ \text{s.t} \quad & \boldsymbol{\theta}^T\mathbf{N}^T\mathbf{N}\boldsymbol{\theta} + \mathbf{f}^T\boldsymbol{\theta} = 0,\end{aligned} \tag{2.61}$$

which consists in minimizing a quadratic cost function subject to a single quadratic constrain. This problem belong to the class of Generalized Trust Region Subproblems (GTRS). According to [25] the solution to (2.61) is given by

$$\widehat{\boldsymbol{\theta}}(\lambda) = (\mathbf{A}^T\mathbf{A} + \lambda\mathbf{N}^T\mathbf{N})^{-1}(\mathbf{A}^T\mathbf{b} - \frac{\lambda}{2}\mathbf{f}), \tag{2.62}$$

where $\lambda \in \mathbb{R}$ is the unique solution of $\varphi(\lambda) = 0$ in the interval $\lambda \in I \subset \mathbb{R}$, where

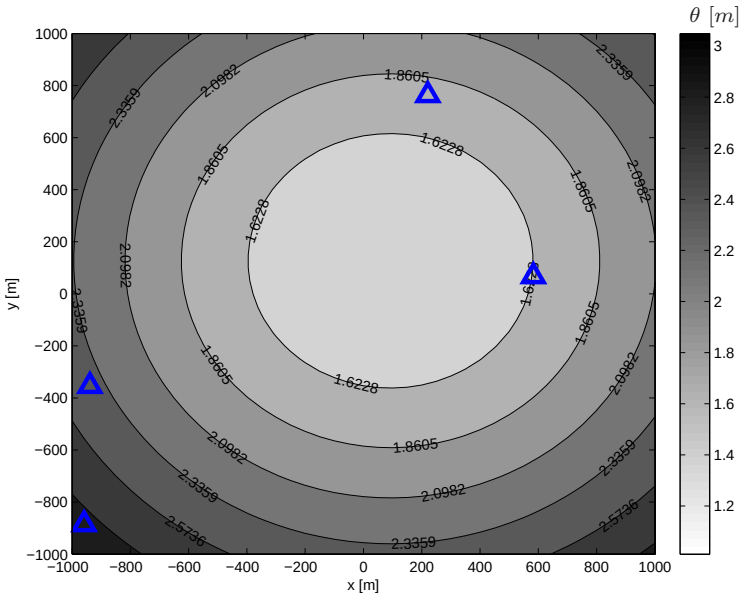
$$\varphi(\lambda) := \widehat{\boldsymbol{\theta}}(\lambda)^T\mathbf{N}^T\mathbf{N}\widehat{\boldsymbol{\theta}}(\lambda) + \mathbf{f}^T\widehat{\boldsymbol{\theta}}(\lambda) \tag{2.63}$$

and the interval I consists of all λ for which the matrix $\mathbf{A}^T\mathbf{A} + \lambda\mathbf{N}^T\mathbf{N}$ is positive definite. That is, $I = (-\frac{1}{\alpha}, \infty)$ where α is the largest eigenvalue of matrix $(\mathbf{A}^T\mathbf{A})^{-\frac{1}{2}}\mathbf{N}^T\mathbf{N}(\mathbf{A}^T\mathbf{A})^{-\frac{1}{2}}$. The authors claim that φ is strictly decreasing in the interval I and therefore a simple bisection algorithm can be employed to determine its unique zero in the interval. Instead, the work in [84] proposed expressing φ as a 5-degree polynomial and determined all of its roots. The solution that yielded the smallest error was then selected, see [25] for more details.

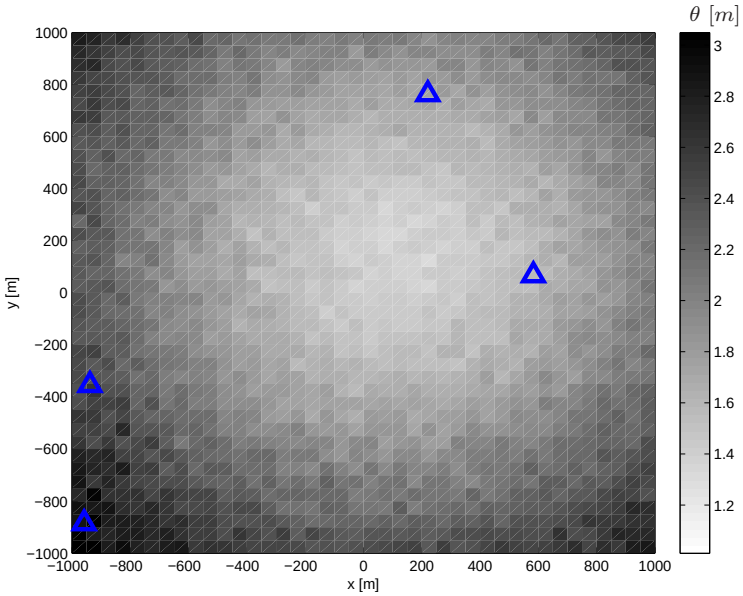
The previously described LS-GTRS algorithm is not strictly a closed form algorithm since some iterative of numerical algorithms must be used to determine the optimal λ. Its numerical complexity is then much higher than the simpler LS-C and LS-CW solutions. However, as shown later, its performance is far superior to that obtained with simple methods. Some numerical simulations of the LS-C, LS-CW and LS-GTRS were performed to illustrate its behavior. A set of $m = 4$ landmarks were located at random locations. Fig. 2.3 shows the theoretical predicted Root Mean Square Error (RMSE) of the LS-C algorithm, given by (2.48), as compared to a numerically derived RMSE. The numerical results were obtained by generating a set of $N = 500$ noisy range measurements at each grid point with $\mathbf{R} = \mathbf{I}_4$ (that is, $\sigma_{ii} = 1$m) with which the algorithm was run. Fig. 2.4 shows the same information for the LS-CW algorithm. Fig. 2.5 shows only the experimental RMSE of the LS-GTRS algorithm, obtained in a similar way.

2.3.5 3D Range-Only localization with known z

In some applications one is interested in solving a 3D Range-Only localization problem in which the z-coordinate of the vehicle is already known. For instance, in underwater applications pressure sensors are a simple and cost effective solution to

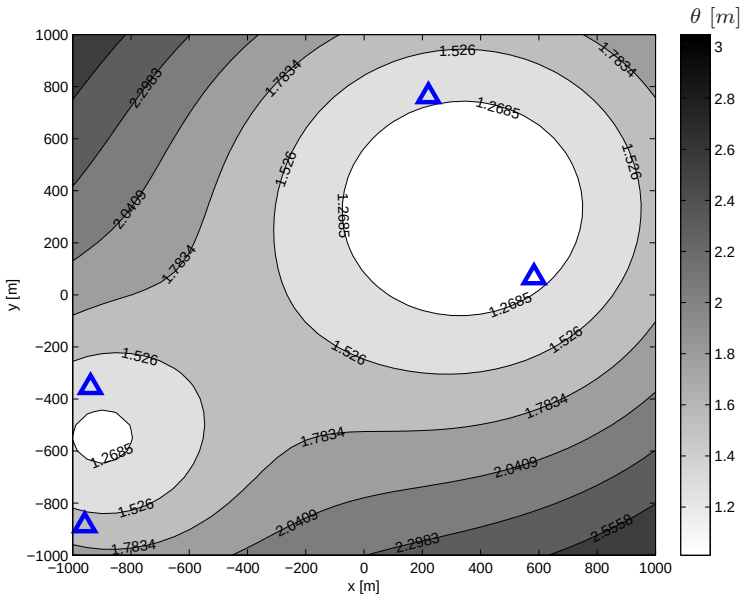


(a) LS-C (Theoretical RMSE)

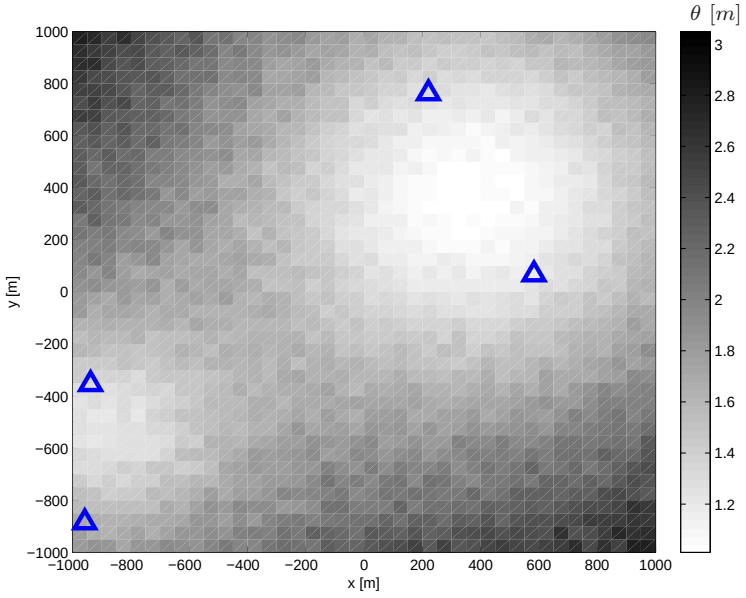


(b) LS-C (Experimental RMSE)

Figure 2.3: Numerical validation of the theoretical Root Mean Square Error (RMSE) of the LS-C algorithm. The study was done with four landmarks ($m = 4$) in a random geometry and the covariance of the range measurements was set to $\mathbf{R} = \mathbf{I}_3$ in [m^2], which corresponds to a standard deviation $\sigma_{ii} = 1$m. The value of the theoretical RMSE $\theta = \sqrt{\text{tr}\left(\mathbf{\Sigma}_{\text{LS-C}}\right)}$ is shown where $\mathbf{\Sigma}_{\text{LS-C}}$ is the error covariance in (2.48). The experimental values were obtained through 500 Monte Carlo simulations at each grid point.

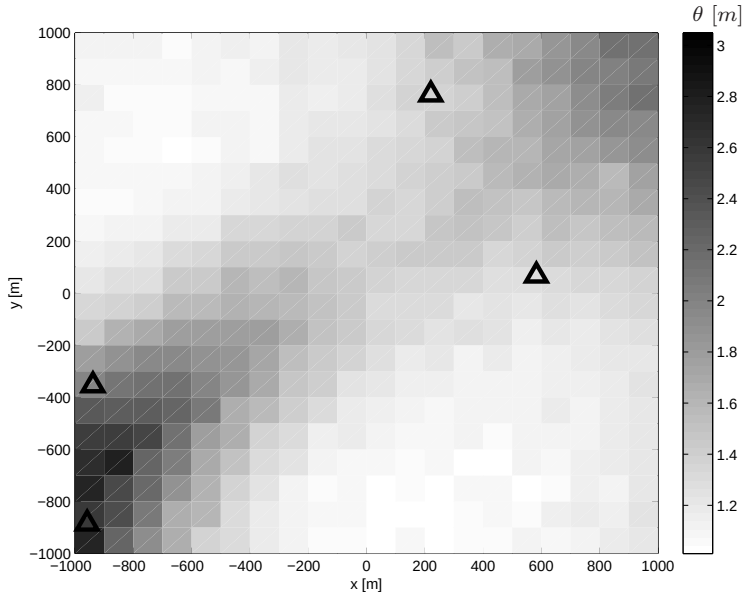


(a) LS-CW (Theoretical RMSE)



(b) LS-CW (Experimental RMSE)

Figure 2.4: Numerical validation of the theoretical RMSE of the LS-CW algorithm. The value of the theoretical RMSE $\theta = \sqrt{\mathrm{tr}\left(\mathbf{\Sigma}_{\text{LS-CW}}\right)}$ is shown where $\mathbf{\Sigma}_{\text{LS-CW}}$ is the error covariance in (2.51). The experimental values were obtained through 500 Monte Carlo simulations at each grid point.



(a) LS-GTRS (Experimental RMSE)

Figure 2.5: Numerical validation of the RMSE of the LS-GTRS algorithm. The experimental values were obtained through 500 Monte Carlo simulations at each grid point.

provide depth measurements. When possible, one should avoid to use acoustics to estimate depth as they are much more complex and expensive. A quite common situation is to use acoustics to determine the two dimensional position (x, y) and use a depth sensor to provide the z coordinate. External acoustic tracking systems as the GPS Intelligent Buoys (GIB) (see the Introduction or next chapter for more details) do not have direct access to the depth information on-board the vehicle, but this data is sent through the acoustic channel using a simple telemetry trick: The vehicle depth is coded in the delay of two consecutive acoustic pulses sent at each emission. It is then interesting in practice to specialize some of the Range-Only localization algorithms to the case when the z coordinate is known.

The 2D Range-Only localization problem can be formulated using a transformation of the 3D measured ranges. Let the 3D vectors $\mathbf{p}$ and $\mathbf{p}_i$ have coordinates

$$\mathbf{p} := \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}, \quad \mathbf{p}_i := \begin{bmatrix} p_{ix} \\ p_{iy} \\ p_{iz} \end{bmatrix}, \tag{2.64}$$

and define their 2D versions

$$\mathbf{p}' := \begin{bmatrix} p_x \\ p_y \end{bmatrix}, \quad \mathbf{p}_i' := \begin{bmatrix} p_{ix} \\ p_{iy} \end{bmatrix}. \tag{2.65}$$

Let $\delta_{iz} \in \mathbb{R}$ denote the difference between the depth of the vehicle and the z coordi-

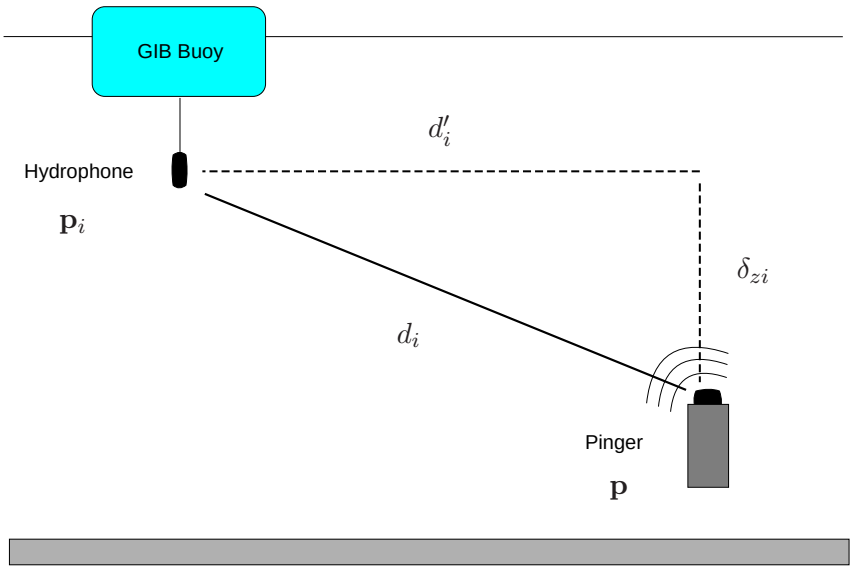


Figure 2.6: 2D range measurements from 3D ranges

nate of the i-th landmark, i.e., $\delta_{iz} = p_z - p_{iz}$. Then one can obtain the 2D squared ranges from their 3d versions by

$$d'_i := \|\mathbf{p}' - \mathbf{p}'_i\|^2 = d_i - \delta_{iz}^2, \tag{2.66}$$

and formulate the trilateration problem entirely in 2D, see Fig.2.6.

Even more interesting is the case in which the vehicle depth is not known but all the landmarks have the same z-coordinate. We will see that in this case it is also possible to formulate an equivalent 2D Range-Only localization problem. This is quite convenient since, three (not collinear) landmarks are enough to uniquely determine a 2D fix whereas at least 4 landmarks are required to obtain a 3D fix. If all the landmarks have the same z coordinate $p_{1z} = p_{2z} = \ldots = p_{mz}$, then the difference in z coordinate between the vehicle and all the landmarks is the same, that is, $\delta_z := \delta_{1z} = \delta_{2z} = \ldots = \delta_{mz}$. In this case, one can write

$$\begin{aligned} d_i &= \|\mathbf{p} - \mathbf{p}_i\|^2 = (p_x - p_{ix})^2 + (p_y - p_{iy})^2 + (p_z - p_{iz})^2 && (2.67)\\ &= \|\mathbf{p}' - \mathbf{p}'_i\|^2 + \delta_z^2 = \mathbf{p}'^T_i \mathbf{p}'_i - 2\mathbf{p}'^T_i \mathbf{p}' + \|\mathbf{p}'\|^2 + \delta_z^2, && (2.68) \end{aligned}$$

and

$$\mathbf{d} = \delta(\mathbf{P}'^T \mathbf{P}') - 2\mathbf{P}'^T \mathbf{p}' + (\|\mathbf{p}'\|^2 + \delta_z^2)\mathbf{1}_m. \tag{2.69}$$

The unknown scalar terms $\|\mathbf{p}'\|^2 + \delta_z^2$ appear multiplying the vector of ones $\mathbf{1}_m$. Define the vector of 2D squared ranges $\mathbf{d}' = [d'_1, \ldots, d'_m]^T$. Note that the unknown term containing $\mathbf{1}_m$ can be deleted by multiplying both sides of the equation by

matrix $\mathbf{M}$, which has $\mathbf{1}_m$ on its null space, to obtain

$$\mathbf{Md} = \mathbf{M}\delta(\mathbf{P}'^T\mathbf{P}') - 2\mathbf{M}\mathbf{P}'^T\mathbf{p}' = \mathbf{Md}'. \tag{2.70}$$

This shows that after multiplying by matrix $\mathbf{M}$, the 3D squared ranges are the same as the 2D ranges and an equivalent 2D Range-Only localization problem can be formulated. Now the previous LS algorithms can be applied to solve the equivalent (and much simpler) 2D Range-Only localization problem.

2.4 Iterative minimization algorithms

The previous Least Squares algorithms were derived in order to obtain numerically simple solutions to the Range-Only localization problem. The solutions were derived by certain algebraic manipulations and in general are not optimal. Given a set of noisy observations of an unknown parameter, and assuming that the distribution of the observation errors is known, it is natural to ask what is the value of the parameter that with most probability could have generated the observations. This parameter estimate, determined by solving an optimization problem that in general does not admit a closed form solution, is the Maximum Likelihood (ML) estimate [170], [95].

In this section we will derive gradient and Newton descent iterative algorithms to minimize certain cost functions related to the Maximum Likelihood estimate of the range measurements. First, we will give a brief summary of these algorithms including a popular line search procedure. We will then introduce the Maximum Likelihood function with ranges (ML-R) and an alternative cost function based on the squared range measurements that is inspired in the ML criteria (ML-SR). In a recent paper [26], the authors present and analyze some simple iterative algorithms to minimize the ML-R and ML-SR cost functions. The algorithms are of a gradient nature and thus exhibit poor linear convergence rates. Moreover, the algorithms consider only the case of diagonal measurement error covariances, which do not include, for instance, mutually dependent measurement error models. The authors show that under certain conditions their algorithms converge to a stationary point, which does not exclude local minima.

In this section we will show that under certain conditions, that can be directly checked from the available data, the ML-SR function is convex and therefore iterative descent algorithms can be efficiently used to find a global minimum. We will derive Newton based algorithms that exhibit quadratic convergence near the solution. The algorithms consider a general measurement error covariance $\mathbf{R}$, that is not necessarily diagonal, and can then therefore capture mutually dependent errors. Moreover, the resulting algorithms are of a relatively moderate numerical complexity, which will be empirically shown to be of the same order of magnitude as that of the LS-GTRS

algorithm. A third cost function will be discussed, denoted ML-CSR, that is based on the centered squared ranges inspired by the LS-C algorithm. It will be shown that the ML-CSR is always convex and that its global minimum coincides with the solution of the LS-C algorithm.

A fundamental requirement of the iterative minimization algorithms to be presented is the need for an initial position estimate. The behavior of the iterative algorithms might be very different depending on the initial estimate. In some cases, in the presence of local minimums, the final solution of the algorithms might depend on the initial estimate. Unfortunately, in general there is no a priori knowledge of the vehicle position such that an initialization scheme must be derived. It is common to use the previous closed form localization algorithms to determine an initial position estimate and then use iterative minimization algorithms to refine the solution.

2.4.1 Gradient and Newton descent algorithms

For the sake of completeness, in this subsection we will give a brief overview of the gradient and Newton descent algorithms. There are several other optimization algorithms that could be employed such as the conjugate gradient descent, or quasi-Newton methods. The reader is referred to the excellent textbooks by Luenberger [114], Bertsekas [29], and Boyd [32] for more details. In the present work we will focus on the gradient and Newton algorithms. As it will be shown later, it is relatively simple to obtain expressions for the gradient and Hessians of the cost functions of interest, which allows for straight forward implementation of the above mentioned algorithms.

Suppose one wants to determine the solution of

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}), \tag{2.71}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth function. Starting from an initial estimate $\mathbf{x}_0$, the gradient and Newton descent algorithms are based on the recursion

$$\mathbf{x}_{k+1} = \mathbf{x}_k + s_k h(\mathbf{x}_k), \tag{2.72}$$

where s_k is the step size and $h(\mathbf{x}_k)$, with $h : \mathbb{R}^n \to \mathbb{R}^n$ smooth, is the search direction at iteration k. The gradient descent algorithm uses a negative gradient direction $h(\mathbf{x}) = -\nabla f(\mathbf{x})$, and the Newton algorithm uses $h(\mathbf{x}) = -(\nabla^2 f(\mathbf{x}))^{-1}\nabla f(\mathbf{x})$ whenever the Hessian $\nabla^2 f(\mathbf{x})$ is invertible.

There are several methodologies to choose the step size s_k at each iteration. The simplest one is to make $s_k = s$, a constant. However, this particular choice yields very poor performance and the number of iterations required to achieve a certain level of convergence may become large. Alternatively, one can determine the optimal

step size s_k^* at each iteration, that is, the step size that solves

$$s_k^* = \arg\min_{s\in\mathbb{R}} f(\mathbf{x}_k + s\, h(\mathbf{x}_k)), \tag{2.73}$$

where f is the function we want to minimize, and $\mathbf{x}_k$ is the value of the parameter at iteration k. In practice, this procedure can be extremely time consuming, depending on the complexity of the cost function, being some times as hard as the original optimization problem. It is common to use a stepsize selection rule that is a compromise between the simplicity of the fixed step and the complexity of the optimal solution. The Armijo rule is one of the most commonly used (see for instance [29, p.29]). It consists of selecting the step size

$$s_k = s\beta^{m_i}, \tag{2.74}$$

where $m_i \in \{0, 1, 2, \ldots\}$ is the first integer that satisfies

$$f(\mathbf{x}_k + s\beta^{m_i} h(\mathbf{x}_k)) \leq f(\mathbf{x}_k) + \sigma s\beta^{m_i}\, h(\mathbf{x}_k)^T \nabla f(\mathbf{x}_k), \tag{2.75}$$

for some constants $s > 0$, and $\beta, \sigma \in (0, 1)$.

The gradient and Newton algorithms can then be summarized as follow:

1. *Start at initial estimate* $\mathbf{x}_0 \in \mathbb{R}^n$. *Set* $k = 0$.
2. *Determine a search direction:*
   (a) *Gradient descent:* $h(\mathbf{x}) = -\nabla f(\mathbf{x})$,
   (b) *Newton descent:* $h(\mathbf{x}) = -(\nabla^2 f(\mathbf{x}))^{-1}\nabla f(\mathbf{x})$.
3. *Determine step size* s_k *using Armijo rule (2.74).*
4. *Update estimate:* $\mathbf{x}_{k+1} = \mathbf{x}_k + s_k h(\mathbf{x}_k)$. *Set* $k = k + 1$.
5. *If* $\|\nabla f(\mathbf{x}_k)\| \leq \epsilon$ *or* $k \geq k_{max}$, *stop. Otherwise return to 2.*

2.4.2 Maximum Likelihood with Ranges (ML-R)

The range measurement vector between the vehicle at position $\mathbf{x} \in \mathbb{R}^n$ and the landmarks can be written as $\bar{\mathbf{r}} = \mathbf{r}(\mathbf{x}) + \mathbf{w} \in \mathbb{R}^m$, where the entries of $\mathbf{r}(\mathbf{x}) = [r_1(\mathbf{x}), \ldots, r_m(\mathbf{x})]^T \in \mathbb{R}^m$ contain the true ranges $r_i(\mathbf{x}) = \|\mathbf{x} - \mathbf{p}_i\|$ and the observation error $\mathbf{w} \in \mathbb{R}^m$ is taken to be a zero mean Gaussian vector with covariance $\mathbf{R}$. Given a vector of observations $\bar{\mathbf{r}} \in \mathbb{R}^m$, the function $\mathcal{L} : \mathbb{R}^n \to [0, 1] \subset \mathbb{R}$ which for any vehicle position $\mathbf{x} \in \mathbb{R}^n$ yields the probability $p(\bar{\mathbf{r}}|\mathbf{x})$, is referred to as the

likelihood function, given by

$$\mathcal{L}(\mathbf{x}) := p(\bar{\mathbf{r}}|\mathbf{x}) = \frac{1}{(2\pi)^{\frac{m}{2}}\,|\mathbf{R}|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\bar{\mathbf{r}}-\mathbf{r}(\mathbf{x}))^T\mathbf{R}^{-1}(\bar{\mathbf{r}}-\mathbf{r}(\mathbf{x}))\right\}. \tag{2.76}$$

The Maximum Likelihood (ML) estimator is defined as

$$\widehat{\mathbf{p}}_{ML} = \arg\max_{\mathbf{x}\in\mathbb{R}^n} \mathcal{L}(\mathbf{x}). \tag{2.77}$$

A common practice in Maximum Likelihood Estimation is to work with the *log-likelihood function*. Since the logarithm is a strictly increasing function, and $\mathcal{L}(\mathbf{x})$ is strictly positive, maximizing the *likelihood* and the *log-likelihood* are equivalent. In our case we have

$$\begin{aligned}\log\mathcal{L}(\mathbf{x}) &= -\frac{m}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{R}| - \frac{1}{2}(\bar{\mathbf{r}}-\mathbf{r}(\mathbf{x}))^T\mathbf{R}^{-1}(\bar{\mathbf{r}}-\mathbf{r}(\mathbf{x}))\\ &= K - \frac{1}{2}(\bar{\mathbf{r}}-\mathbf{r}(\mathbf{x}))^T\mathbf{R}^{-1}(\bar{\mathbf{r}}-\mathbf{r}(\mathbf{x})).\end{aligned} \tag{2.78}$$

Now neglecting constant terms, the ML estimator can be found by solving the optimization problem

$$\widehat{\mathbf{p}}_{ML} = \arg\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}), \tag{2.79}$$

where $f:\mathbb{R}^n\to\mathbb{R}$ is given by

$$f(\mathbf{x}) := \frac{1}{2}(\bar{\mathbf{r}}-\mathbf{r}(\mathbf{x}))^T\mathbf{R}^{-1}(\bar{\mathbf{r}}-\mathbf{r}(\mathbf{x})). \tag{2.80}$$

In the sequel, we will refer to this function as the Maximum Likelihood with Ranges (ML-R) cost function. In general, there is no closed form solution to the previous optimization problem. The ML-R is relatively complex, nonlinear and even not differentiable at some points because of the square root that defines the range measurements. Nevertheless, one can try to solve the ML-R minimization problem by using some iterative optimization algorithm such as the negative gradient descent or Newton methods [32], [114], [29].

In order to implement gradient and Newton descent algorithms to minimize the ML-R function it is necessary to have expressions for its gradient and Hessian. This can be done resorting to Matrix Differential Calculus, as detailed in appendix A. We have the following result:

Proposition 2.4.1 (Gradient and Hessian of the ML-R cost function)**.** *Define the matrix*

$$\mathbf{C} := \mathbf{x}\mathbf{1}_m^T - \mathbf{P} = \begin{bmatrix}\mathbf{x}-\mathbf{p}_1 & \dots & \mathbf{x}-\mathbf{p}_m\end{bmatrix} \in \mathbb{R}^{n\times m}, \tag{2.81}$$

and the vector $\boldsymbol{\alpha} = \mathbf{R}^{-1}(\bar{\mathbf{r}}-\mathbf{r})$*. The gradient of the ML-R cost function at* $\mathbf{x}\in\mathbb{R}^n$

is given by

$$\nabla f(\mathbf{x}) = -\mathbf{C}\delta(\mathbf{r})^{-1}\mathbf{R}^{-1}(\bar{\mathbf{r}} - \mathbf{r}), \tag{2.82}$$

and its Hessian by

$$\nabla^2 f(\mathbf{x}) = \mathbf{C}\delta(\mathbf{r})^{-1}\Big(\mathbf{R}^{-1} - \delta(\mathbf{r})^{-1}\delta(\boldsymbol{\alpha})\Big)\delta(\mathbf{r})^{-1}\mathbf{C}^T + \boldsymbol{\alpha}^T\delta(\mathbf{r})^{-1}\mathbf{1}_m\mathbf{I}_n, \tag{2.83}$$

where $\mathbf{C} = \mathbf{C}(\mathbf{x})$ *and* $\mathbf{r} = \mathbf{r}(\mathbf{x})$. *In particular, when the measurement error covariance is diagonal with structure*

$$\mathbf{R} = \begin{bmatrix} \sigma_{11}^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_{mm}^2 \end{bmatrix}, \tag{2.84}$$

the Hessian becomes

$$\nabla^2 f(\mathbf{x}) = \mathbf{C}\delta(\mathbf{r})^{-3}\mathbf{R}^{-1}\delta(2\mathbf{r} - \bar{\mathbf{r}})\mathbf{C}^T + \boldsymbol{\alpha}^T\delta(\mathbf{r})^{-1}\mathbf{1}_m\mathbf{I}_n. \tag{2.85}$$

Proof. See Appendix A.2.1 □

Care should be taken when using the gradient and Newton iterative algorithms in the ML-R function. The ML-R cost function exhibits numerous local minima, that may eventually attract the iterative solutions. Numerical analysis of the cost function reveals the existence of saddle points located in the proximity of the landmarks. Moreover, local minima appeared which severity was greater when the true vehicle position was located far away from the landmarks centroid. These non desired critical points, specially the local minima, could prevent the algorithms from converging to the actual vehicle position so particular attention is required when running the algorithms in an unsupervised manner, or the initial estimate $\mathbf{x}_0$ is far from the actual vehicle position.

2.4.3 Maximum Likelihood with Squared Ranges (ML-SR)

It is apparent that most of the difficulties encountered when minimizing the ML-R cost function are related to the square roots that define the range measurements. Using the square of the range measurements simplifies the problem in a great manner and this is indeed the main motivation for the Least Squares algorithms in the previous section. It is interesting to try and reformulate the original ML problem using the squared ranges as the observations.

It has been shown under the assumption that the ranges are much bigger than the standard deviation of the measurement errors ($r_i >> \sigma_{ii}$), that the squared range measurements can be modelled as $\bar{\mathbf{d}} = \mathbf{d} + \boldsymbol{\xi}$, where $\boldsymbol{\xi}$ is a zero mean Gaussian vector with covariance $\boldsymbol{\Sigma}_\xi = 4\delta(\mathbf{r})\mathbf{R}\delta(\mathbf{r})$, see (2.7). Given a set of range observations

$\bar{\mathbf{r}} \in \mathbb{R}^m$, consider the following cost function $f : \mathbb{R}^n \to \mathbb{R}$,

$$f(\mathbf{x}) := \frac{1}{2}(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x}))^T \boldsymbol{\Sigma}^{-1}(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x})), \tag{2.86}$$

where $\boldsymbol{\Sigma}$ is a constant positive definite matrix, $\bar{\mathbf{d}}$ is the vector of measured squared range measurements, and $\mathbf{d}(\mathbf{x})$ contains the squared ranges between $\mathbf{x}$ and the landmarks i.e., it has entries $d_i(\mathbf{x}) = \|\mathbf{x} - \mathbf{p}_i\|^2$. This function will be referred as the Maximum Likelihood with Squared Ranges (ML-SR) cost function. The ML-SR cost function has some nice properties that make it interesting over the ML-R function. It will be shown that when the true vehicle position is, roughly speaking, near the centroid of the landmarks then the ML-SR is a convex function and a unique global minimum exists. Hence, in this case, one can ensure that by minimizing the ML-SR function with a gradient descent or Newton algorithm, the unique solution can be found with prescribed rates of convergence.

Strictly speaking, the minimizer of the ML-SR function is not a Maximum Likelihood estimate. However, there is a choice of covariance matrix $\boldsymbol{\Sigma}$ that makes the ML-SR most coherent with the Maximum Likelihood function of the squared range observations, that is, $\boldsymbol{\Sigma} = 4\delta(\bar{\mathbf{r}})\mathbf{R}\delta(\bar{\mathbf{r}}) \approx \boldsymbol{\Sigma}_{\xi}$. Unlike the ML-R cost function, we will see that under certain conditions the ML-SR function is convex. The conditions basically require that the true vehicle position be near the centroid of the landmarks. To better illustrate this fact we will first analyze the convexity of the ML-SR function with diagonal covariance $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$. Later, we will analyze the more realistic case in which $\boldsymbol{\Sigma} = 4\delta(\bar{\mathbf{r}})\mathbf{R}\delta(\bar{\mathbf{r}})$ with $\mathbf{R} = \sigma^2\mathbf{I}_m$.

The gradient and the Hessian of the ML-SR function can be computed in a simple compact form as given by the following result:

Proposition 2.4.2 (Gradient and Hessian of the ML-SR cost function)**.** *The gradient of the ML-SR cost function at a point $\mathbf{x} \in \mathbb{R}^n$ can be written as*

$$\nabla f(\mathbf{x}) = -2\mathbf{C}(\mathbf{x})\boldsymbol{\Sigma}^{-1}(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x})), \tag{2.87}$$

and its Hessian as

$$\nabla^2 f(\mathbf{x}) = 4\mathbf{C}(\mathbf{x})\boldsymbol{\Sigma}^{-1}\mathbf{C}^T(\mathbf{x}) - 2(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x}))^T\boldsymbol{\Sigma}^{-1}\mathbf{1}_m\mathbf{I}_n. \tag{2.88}$$

Proof. See Appendix A.2.2. □

Let $\boldsymbol{\chi} = \frac{1}{m}\mathbf{P}\mathbf{1}_m \in \mathbb{R}^n$ denote the centroid of the landmarks in $\mathbf{P}$ and let $\mathbf{p} \in \mathbb{R}^n$ be the actual vehicle position. We now derive a set of results that will prove useful.

Property 2.4.3. *The following identity holds*

$$(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x}))^T\mathbf{1}_m = m(\|\mathbf{p} - \boldsymbol{\chi}\|^2 - \|\mathbf{x} - \boldsymbol{\chi}\|^2) + \boldsymbol{\xi}^T\mathbf{1}_m. \tag{2.89}$$

Proof. Using the fact that

$$\bar{\mathbf{d}} = \mathbf{d}(\mathbf{p}) + \boldsymbol{\xi} = \delta(\mathbf{P}^T\mathbf{P}) - 2\mathbf{P}^T\mathbf{p} + \|\mathbf{p}\|^2\mathbf{1}_m + \boldsymbol{\xi} \tag{2.90}$$

$$\mathbf{d}(\mathbf{x}) = \delta(\mathbf{P}^T\mathbf{P}) - 2\mathbf{P}^T\mathbf{x} + \|\mathbf{x}\|^2\mathbf{1}_m, \tag{2.91}$$

simple computations yield

$$\begin{aligned}(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x}))^T\mathbf{1}_m &= m(\mathbf{p}^T\mathbf{p} - 2\boldsymbol{\chi}^T\mathbf{p} - \mathbf{x}^T\mathbf{x} + 2\boldsymbol{\chi}^T\mathbf{x}) + \boldsymbol{\xi}^T\mathbf{1}_m \\ &= m(\mathbf{p}^T\mathbf{p} - 2\boldsymbol{\chi}^T\mathbf{p} + \boldsymbol{\chi}^T\boldsymbol{\chi}^T) \\ &\quad - m(\mathbf{x}^T\mathbf{x} - 2\boldsymbol{\chi}^T\mathbf{x} + \boldsymbol{\chi}^T\boldsymbol{\chi}) + \boldsymbol{\xi}^T\mathbf{1}_m \\ &= m(\|\mathbf{p} - \boldsymbol{\chi}\|^2 - \|\mathbf{x} - \boldsymbol{\chi}\|^2) + \boldsymbol{\xi}^T\mathbf{1}_m. \end{aligned} \tag{2.92}$$

□

Theorem 2.4.4 (Convexity of the ML-SR cost function with $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}_m$)**.** *The ML-SR cost function defined in (2.87) with diagonal covariance $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}_m$ is convex if and only if*

$$\|\mathbf{p} - \boldsymbol{\chi}\|^2 \leq \frac{2}{m}\lambda_{min}(\mathbf{P}_c\mathbf{P}_c^T) - \frac{1}{m}\boldsymbol{\xi}^T\mathbf{1}_m, \tag{2.93}$$

where $\lambda_{min}(\cdot)$ is the function that extracts the minimum eigenvalue of a matrix.

Proof. A function is convex if and only if its Hessian is positive semidefinite for all $\mathbf{x} \in \mathbb{R}^n$ [29], [32]. Note that the ML-SR cost function is invariant with respect to changes in the inertial reference frame $\{\mathcal{I}\}$ (the same property applies to the ML-R cost function). This is because its terms depend only on distances, which are invariant with respect to isometries. Using this fact, let us consider, without loss of generality, that $\{\mathcal{I}\}$ is centered at $\boldsymbol{\chi}$, the centroid of $\mathbf{P}$. That is, consider the cost function in the transformed variables $\mathbf{p}_c = \mathbf{p} - \boldsymbol{\chi}$, $\mathbf{x}_c = \mathbf{x} - \boldsymbol{\chi}$ and $\mathbf{P}_c = \mathbf{PM} = \mathbf{P} - \boldsymbol{\chi}\mathbf{1}_m^T$. The Hessian now becomes

$$\begin{aligned}\nabla^2 f(\mathbf{x}_c) &= 4(\mathbf{x}_c\mathbf{1}_m^T - \mathbf{P}_c)\boldsymbol{\Sigma}^{-1}(\mathbf{x}_c\mathbf{1}_m^T - \mathbf{P}_c) - 2(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x_c}))^T\boldsymbol{\Sigma}^{-1}\mathbf{1}_m\mathbf{I}_n. \\ &= \frac{4m}{\sigma^2}\mathbf{x}_c\mathbf{x}_c^T + \frac{4}{\sigma^2}\mathbf{P}_c\mathbf{P}_c^T + \frac{2m}{\sigma^2}(\|\mathbf{x}_c\|^2 - \|\mathbf{p}_c\|^2)\mathbf{I}_n - \frac{2}{\sigma^2}\boldsymbol{\xi}^T\mathbf{1}_m\mathbf{I}_n, \end{aligned} \tag{2.94}$$

where property 2.4.3 and the fact that $\mathbf{P}_c\mathbf{1}_m = 0$ have been used. Because $\mathbf{x}_c\mathbf{x}_c^T \succeq 0$ and $\|\mathbf{x}_c\|^2\mathbf{I}_n \succeq 0$, we have that

$$\begin{aligned}\nabla^2 f(\mathbf{x}_c) &\succeq \frac{4}{\sigma^2}\mathbf{P}_c\mathbf{P}_c^T - \frac{2m}{\sigma^2}\|\mathbf{p}_c\|^2\mathbf{I}_n - \frac{2}{\sigma^2}\boldsymbol{\xi}^T\mathbf{1}_m\mathbf{I}_n \\ &\succeq \frac{2}{\sigma^2}(2\lambda_{\min}(\mathbf{P}_c\mathbf{P}_c^T) - m\|\mathbf{p}_c\|^2 - \boldsymbol{\xi}^T\mathbf{1}_m)\mathbf{I}_n. \end{aligned} \tag{2.95}$$

Hence, if

$$\|\mathbf{p}_c\|^2 = \|\mathbf{p} - \boldsymbol{\chi}\|^2 \leq \frac{2}{m}\lambda_{\min}(\mathbf{P}_c\mathbf{P}_c^T) - \frac{1}{m}\boldsymbol{\xi}^T\mathbf{1}_m, \tag{2.96}$$

then $\nabla^2 f(\mathbf{x}_c) \succeq 0$ for all $\mathbf{x}_c \in \mathbb{R}^n$ and f is convex. This proves sufficiency.

Necessity will be shown by contradiction. Assume that f is convex and the condition is not satisfied, i.e.,

$$\|\mathbf{p}_c\|^2 = \|\mathbf{p} - \boldsymbol{\chi}\|^2 > \frac{2}{m}\lambda_{\min}(\mathbf{P}_c\mathbf{P}_c^T) - \frac{1}{m}\boldsymbol{\xi}^T\mathbf{1}_m. \tag{2.97}$$

Since f is convex, its Hessian is positive semidefinite which means that $\mathbf{v}^T\nabla^2 f(\mathbf{x}_c)\mathbf{v} \geq 0$ for all vectors $\mathbf{v}, \mathbf{x}_c \in \mathbb{R}^n$. Now, choose $\mathbf{x}_c = 0$ and $\mathbf{v}$ as the unitary eigenvector of $\mathbf{P}_c\mathbf{P}_c^T$ associated with its minimum eigenvalue $\lambda_{\min}(\mathbf{P}_c\mathbf{P}_c^T)$. We obtain

$$\begin{aligned}
\mathbf{v}^T\nabla^2 f(0)\mathbf{v} &= \frac{4}{\sigma^2}\mathbf{v}^T\mathbf{P}_c\mathbf{P}_c^T\mathbf{v} - \frac{2m}{\sigma^2}\|\mathbf{p}_c\|^2\mathbf{v}^T\mathbf{I}_n\mathbf{v} - \frac{2}{\sigma^2}\boldsymbol{\xi}^T\mathbf{1}_m\mathbf{v}^T\mathbf{v} \\
&= \frac{4}{\sigma^2}\lambda_{\min}(\mathbf{P}_c\mathbf{P}_c^T)\|\mathbf{v}\|^2 - \frac{2m}{\sigma^2}\|\mathbf{p}_c\|^2\|\mathbf{v}\|^2 - \frac{2}{\sigma^2}\boldsymbol{\xi}^T\mathbf{1}_m\|\mathbf{v}\|^2 \\
&= \frac{2m}{\sigma^2}\left(\frac{2}{m}\lambda_{\min}(\mathbf{P}_c\mathbf{P}_c^T) - \|\mathbf{p}_c\|^2 - \frac{1}{m}\boldsymbol{\xi}^T\mathbf{1}_m\right) < 0,
\end{aligned} \tag{2.98}$$

which is a contradiction. □

Proposition 2.4.5 (Sufficient condition for the convexity of the ML-SR cost function with $\boldsymbol{\Sigma} = 4\delta(\bar{\mathbf{r}})\mathbf{R}\delta(\bar{\mathbf{r}})$ and $\mathbf{R} = \sigma^2\mathbf{I}_m$)**.** *Suppose the following condition is satisfied:*

$$2\lambda_{min}(\mathbf{P}_c\delta(\bar{\mathbf{d}})^{-1}\mathbf{P}_c^T) + \beta_m \geq m, \tag{2.99}$$

where

$$\beta_m := \beta(\boldsymbol{\theta}) = \mathbf{d}(\boldsymbol{\theta})^T\delta(\bar{\mathbf{d}})^{-1}\mathbf{1}_m = \sum_{i=1}^{m}\frac{\|\boldsymbol{\theta} - \mathbf{p}_i\|^2}{\bar{d}_i}, \tag{2.100}$$

$$\boldsymbol{\theta} := \frac{1}{\alpha}\mathbf{P}_c\delta(\bar{\mathbf{d}})^{-1}\mathbf{1}_m, \tag{2.101}$$

$$\alpha := \mathbf{1}_m^T\delta(\bar{\mathbf{d}})^{-1}\mathbf{1}_m = \sum_{i=1}^{m}\frac{1}{\bar{d}_i}. \tag{2.102}$$

Then, the ML-SR cost function defined in (2.87) with $\boldsymbol{\Sigma} = 4\delta(\bar{\mathbf{r}})\mathbf{R}\delta(\bar{\mathbf{r}})$ *and* $\mathbf{R} = \sigma^2\mathbf{I}_m$ *is convex.*

Proof. Consider the cost function in the transformed variables $\mathbf{p}_c = \mathbf{p} - \boldsymbol{\chi}$, $\mathbf{x}_c = \mathbf{x} - \boldsymbol{\chi}$, and $\mathbf{P}_c = \mathbf{PM} = \mathbf{P} - \boldsymbol{\chi}\mathbf{1}_m^T$. The covariance matrix $\boldsymbol{\Sigma}$ can be written as

$\mathbf{\Sigma} = 4\delta(\bar{\mathbf{r}})\sigma^2\mathbf{I}_m\delta(\bar{\mathbf{r}}) = 4\sigma^2\delta(\bar{\mathbf{d}})$. The Hessian becomes

$$\begin{aligned}\nabla^2 f(\mathbf{x}_c) &= 4(\mathbf{x}_c\mathbf{1}_m^T - \mathbf{P}_c)\mathbf{\Sigma}^{-1}(\mathbf{x}_c\mathbf{1}_m^T - \mathbf{P}_c)^T - 2(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x_c}))^T\mathbf{\Sigma}^{-1}\mathbf{1}_m\mathbf{I}_n \\ &= 4\mathbf{x}_c\mathbf{1}_m^T\mathbf{\Sigma}^{-1}\mathbf{1}_m\mathbf{x}_c^T + 4\mathbf{P}_c\mathbf{\Sigma}^{-1}\mathbf{P}_c^T - \frac{1}{2\sigma^2}(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x}_c))^T\delta(\bar{\mathbf{d}})^{-1}\mathbf{1}_m\mathbf{I}_n \\ &= \frac{\alpha}{\sigma^2}\mathbf{x}_c\mathbf{x}_c^T + \frac{1}{\sigma^2}\mathbf{P}_c\delta(\bar{\mathbf{d}})^{-1}\mathbf{P}_c^T + \frac{1}{2\sigma^2}(\beta(\mathbf{x}_c) - m)\mathbf{I}_n, \end{aligned} \tag{2.103}$$

where

$$\alpha := \mathbf{1}_m^T\delta(\bar{\mathbf{d}})^{-1}\mathbf{1}_m = \sum_{i=1}^{m}\frac{1}{\bar{d}_i} > 0 \tag{2.104}$$

$$\beta(\mathbf{x}_c) := \mathbf{d}(\mathbf{x}_c)^T\delta(\bar{\mathbf{d}})^{-1}\mathbf{1}_m = \sum_{i=1}^{m}\frac{d_i(\mathbf{x}_c)}{\bar{d}_i} = \sum_{i=1}^{m}\frac{\|\mathbf{x}_c - \mathbf{p}_i\|^2}{\bar{d}_i} > 0. \tag{2.105}$$

The function $\beta(\cdot)$ is always positive, and it can be shown that it is lower bounded by a positive constant $\beta_m := \beta(\boldsymbol{\theta})$ where

$$\boldsymbol{\theta} := \frac{1}{\alpha}\mathbf{P}_c\delta(\bar{\mathbf{d}})^{-1}\mathbf{1}_m = \frac{1}{\alpha}\sum_{i=1}^{m}\frac{\mathbf{p}_i}{\bar{d}_i}. \tag{2.106}$$

Because $\alpha\mathbf{x}_c\mathbf{x}_c^T \succeq 0$ and $\beta(\mathbf{x}_c) \geq \beta_m > 0$, we have that

$$\nabla^2 f(\mathbf{x}_c) \succeq \frac{1}{\sigma^2}\mathbf{P}_c\delta(\bar{\mathbf{d}})^{-1}\mathbf{P}_c^T + \frac{\beta_m - m}{2\sigma^2}\mathbf{I}_n \tag{2.107}$$

$$\succeq \frac{1}{\sigma^2}\left(\lambda_{\min}(\mathbf{P}_c\delta(\bar{\mathbf{d}})^{-1}\mathbf{P}_c^T) + \frac{\beta_m - m}{2}\right)\mathbf{I}_n. \tag{2.108}$$

Hence, if

$$2\lambda_{\min}(\mathbf{P}_c\delta(\bar{\mathbf{d}})^{-1}\mathbf{P}_c^T) + \beta_m \geq m, \tag{2.109}$$

then $\nabla^2 f(\mathbf{x}_c) \succeq 0$ and f is convex. □

Note that the above sufficient condition can be checked a priori without knowing the actual position of the vehicle, since (2.99) only depends on the matrix of landmark coordinates $\mathbf{P}_c$ and the actual squared range measurements $\bar{\mathbf{d}}$. If the condition is satisfied, then one can be sure that using a Newton or gradient descent algorithms to minimize the ML-SR function a unique global minimizer will be reached with prescribed degrees of convergence [32],[29]. If, on the other hand, the condition is not satisfied, extra care is required when using the aforementioned iterative minimization algorithms since they might get stuck at non desired local minima.

2.4.4 Maximum Likelihood with Centered Squared Ranges (ML-CSR)

Consider the cost function $f : \mathbb{R}^n \to \mathbb{R}$ given by

$$f(\mathbf{x}) := \frac{1}{2}(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x}))^T \mathbf{M}\boldsymbol{\Sigma}\mathbf{M}(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x})), \tag{2.110}$$

where $\boldsymbol{\Sigma} \in \mathbb{R}^{m\times m}$ is a positive semidefinite matrix and $\mathbf{M} = \mathbf{I}_m - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^T$ is the centering operator. This cost function will be referred as the Maximum Likelihood with Centered Square Ranges (ML-CSR). It has the nice property of being always convex.

Proposition 2.4.6 (Convexity of the ML-CSR cost function)**.** *The ML-CSR cost function is convex.*

Proof. Since

$$\bar{\mathbf{d}} = \mathbf{d}(\mathbf{p}) + \boldsymbol{\xi} = \delta(\mathbf{P}^T\mathbf{P}) - 2\mathbf{P}^T\mathbf{p} + \|\mathbf{p}\|^2\mathbf{1}_m + \boldsymbol{\xi} \tag{2.111}$$

$$\mathbf{d}(\mathbf{x}) = \delta(\mathbf{P}^T\mathbf{P}) - 2\mathbf{P}^T\mathbf{x} + \|\mathbf{x}\|^2\mathbf{1}_m, \tag{2.112}$$

we have that

$$\mathbf{M}(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x})) = -2\mathbf{P}_c^T(\mathbf{p} - \mathbf{x}) + \mathbf{M}\boldsymbol{\xi}, \tag{2.113}$$

and the ML-CSR cost function can be written as

$$f(\mathbf{x}) = 2(\mathbf{p} - \mathbf{x})^T\mathbf{P}_c\boldsymbol{\Sigma}\mathbf{P}_c^T(\mathbf{p} - \mathbf{x}) - 2(\mathbf{p} - \mathbf{x})^T\mathbf{P}_c\boldsymbol{\Sigma}\mathbf{M}\boldsymbol{\xi} + \frac{1}{2}\boldsymbol{\xi}^T\mathbf{M}\boldsymbol{\xi}. \tag{2.114}$$

Now it is simpler to determine its gradient

$$\nabla f(\mathbf{x}) = -4\mathbf{P}_c\boldsymbol{\Sigma}\mathbf{P}_c^T(\mathbf{p} - \mathbf{x}) + 2\mathbf{P}_c\boldsymbol{\Sigma}\mathbf{M}\boldsymbol{\xi}, \tag{2.115}$$

and its Hessian

$$\nabla^2 f(\mathbf{x}) = 4\mathbf{P}_c\boldsymbol{\Sigma}\mathbf{P}_c^T \succeq 0. \tag{2.116}$$

Since $\boldsymbol{\Sigma}$ is positive definite, the Hessian is positive semidefinite, and this shows convexity of the ML-CSR. Moreover, if $\mathbf{P}_c$ is full column rank and $\mathbf{P}_c\boldsymbol{\Sigma}\mathbf{P}_c^T$ is positive definite, then the ML-CSR function is strictly convex. □

The ML-CSR cost function does not have the problems encountered when minimizing the ML-R and the ML-SR cost functions. In this case, the function is always convex, and usually strictly convex so that a unique global minimum exists. In fact, it does not make sense to try to iteratively minimize the ML-CSR function since it has a closed form solution. It can be shown that when $\boldsymbol{\Sigma} = \mathbf{I}_m$, the minimizer of the

ML-CSR cost function is the LS-C estimate described in (2.32). Moreover, when $\boldsymbol{\Sigma}$ is chosen as $\boldsymbol{\Sigma} = \widehat{\boldsymbol{\Sigma}}_e^{\dagger}$ as defined in (2.33) then the minimizer of the ML-CSR function is the LS-CW estimate in (2.36).

2.4.5 Numerical simulations

In order to illustrate the characteristics of the different cost functions previously described, numerical simulations were done. The contours of the ML-R, ML-SR and ML-CSR cost functions are depicted in Figs. 2.7 - 2.10 where the geometry of the landmarks and the position of the actual vehicle were varied. In all the simulations, for simplicity, the covariance matrices were chosen as $\mathbf{R} = \mathbf{I}_m$ and $\boldsymbol{\Sigma} = \mathbf{I}_m$, were m is the number of landmarks of each simulation. This corresponds to a standard deviation of 1m in both the ranges and the square ranges. The cost functions considered were as follows:

ML-R	$f(\mathbf{x}) = \frac{1}{2}\|\bar{\mathbf{r}} - \mathbf{r}(\mathbf{x})\|^2$
ML-SR	$f(\mathbf{x}) = \frac{1}{2}\|\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x})\|^2$
ML-CSR	$f(\mathbf{x}) = \frac{1}{2}\|\mathbf{M}(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x}))\|^2$

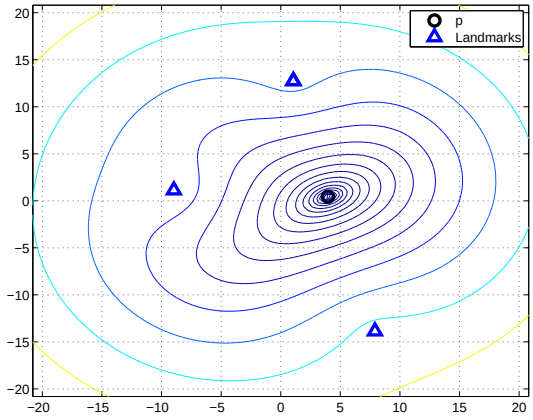
The values of the different cost functions were evaluated in a grid after which their level curves were determined and plotted using MATLAB. For each simulation, first the true vehicle position and the landmark coordinates were set. Then, the true ranges and square ranges between the landmarks and the vehicle $\bar{\mathbf{r}}$ and $\bar{\mathbf{d}}$ were computed without noise ($\boldsymbol{\xi} = \mathbf{w} = 0$). At every grid point $\mathbf{x}$, the ranges and square ranges $\bar{\mathbf{r}}(\mathbf{x})$ and $\bar{\mathbf{d}}(\mathbf{x})$ were computed and the cost functions evaluated. In all of the ML-SR figures, the region that satisfies the condition (2.93) is delimitted by a red circle. According to proposition 2.4.4, if the actual vehicle position $\mathbf{p}$ is inside this region, then the ML-SR function is convex and a unique global minimum exists.

Figure 2.7 shows the case of a good landmark geometry with $m = 3$ (almost an equilateral triangle), where the actual vehicle position $\mathbf{p}$ is inside the red circle. The ML-R function seems well behaved but closer examination reveals the existence of saddle points near the landmarks. Instead, the ML-SR and ML-CSR functions are convex. If with the same favorable geometry the actual vehicle position $\mathbf{p}$ is carried away from the landmark centroid, outside the region delimited by the red circle, the flaws of the ML-R and ML-SR cost functions become more evident. Figure 2.8 shows that both the ML-R and the ML-SR function are not convex and several other critical points appear. However, the ML-CSR is still well behaved, and a unique local minimum appears on top of the actual vehicle position.

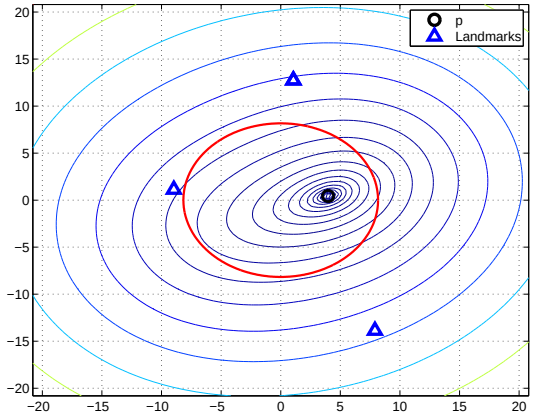
Figures 2.9 and 2.10 illustrate the same phenomenon with a poorer landmark geometry composed of $m = 5$ landmark in random locations. When the true vehicle is inside the red circle (Figure 2.9), all the functions seem well behaved although theory tells us about the existence of nondesired critical points in the ML-R case. When the vehicle is far from the landmark centroid (Figure 2.10), then several

local minimum, saddle points and local maximums appear in the ML-R and ML-SR functions. The ML-CSR function remains convex however.

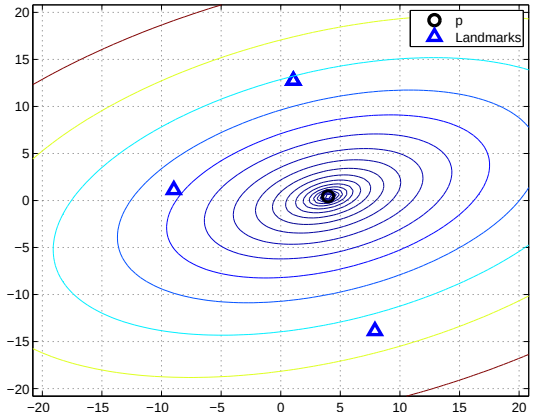
Figures 2.11-2.14 illustrate the behavior of the ML-R and ML-SR Newton iterative minimization algorithms for different landmark geometries and initialization points. Figures 2.11-2.12 show the results corresponding to a landmark geometry equal to that depicted in Figure 2.7. There is a good landmark geometry and the vehicle position is near the landmark centroid. Several instances of the algorithms were run for different initial conditions, as depicted by the black circles. The value of the gradient of the costs functions as well as the estimation error at different iterations is also shown. For all the initial conditions the solutions converge to the desired critical point. On the other hand, Figures 2.13-2.14 show the behavior of the algorithms in the case of poor landmark geometry and vehicle far from landmark centroid as in 2.7. For some of the initial conditions, the solutions of the algorithms converge to non desired local minima.



(a) ML-R

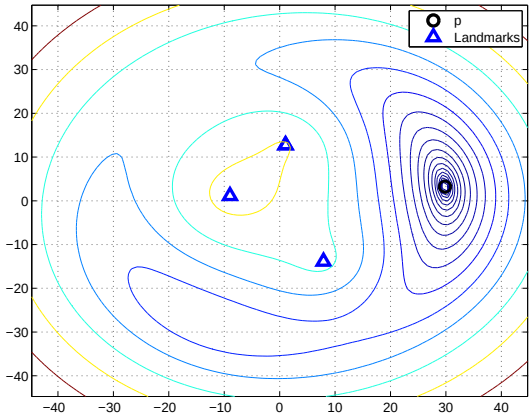


(b) ML-SR

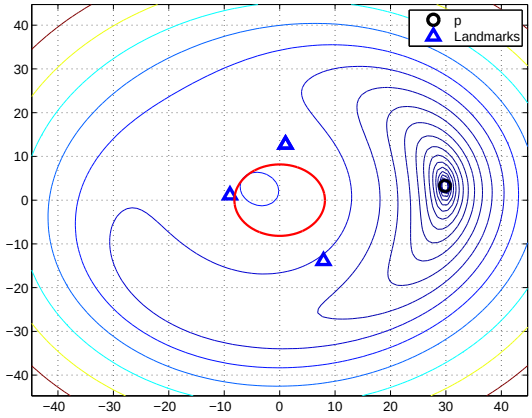


(c) ML-CSR

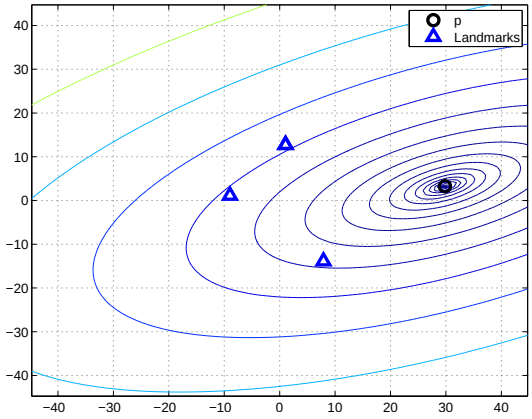
Figure 2.7: Level curves of the ML cost functions with $\mathbf{R} = \mathbf{I}_m$, good landmark geometry, and true vehicle near landmark centroid such that condition (2.93) is satisfied.



(a) ML-R

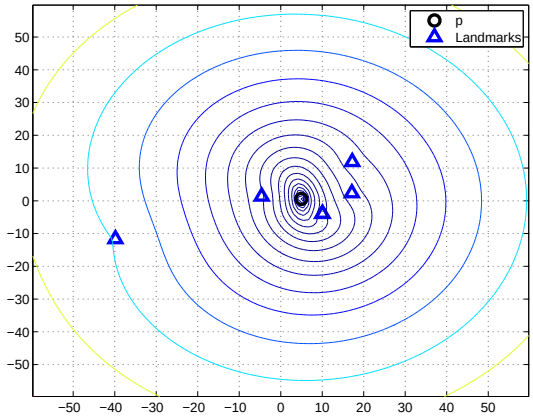


(b) ML-SR

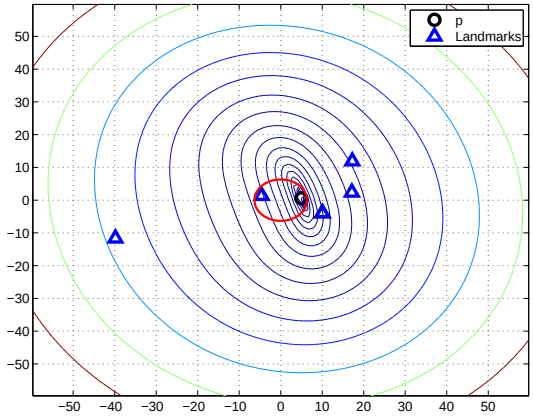


(c) ML-CSR

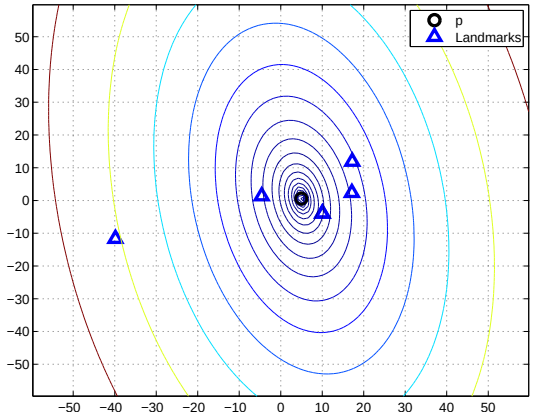
Figure 2.8: Level curves of the ML cost functions with $\mathbf{R} = \mathbf{I}_m$, good landmark geometry, and true vehicle far from landmark centroid such that condition (2.93) is not satisfied.



(a) ML-R

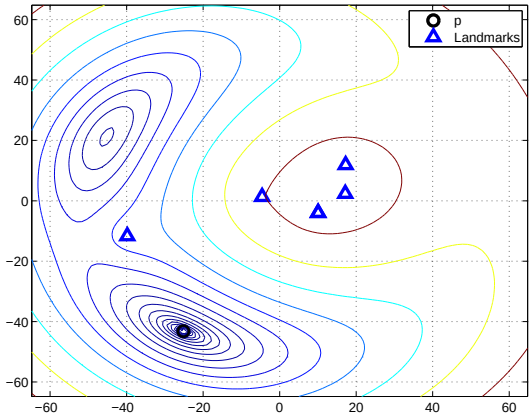


(b) ML-SR

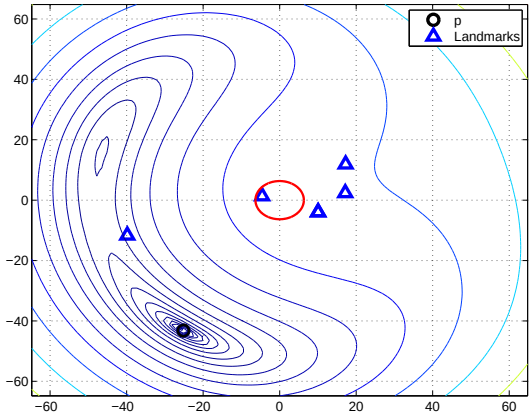


(c) ML-CSR

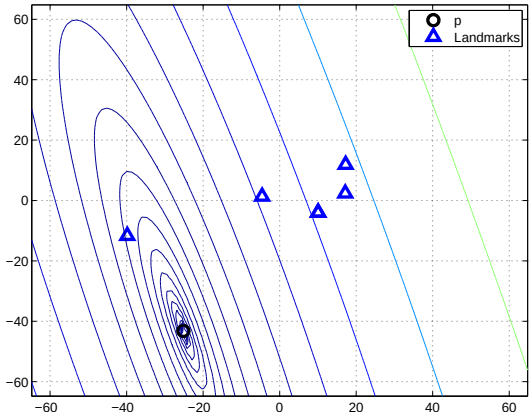
Figure 2.9: Level curves of the ML cost functions with $\mathbf{R} = \mathbf{I}_m$, poor landmark geometry, and true vehicle near landmark centroid such that condition (2.93) is satisfied.



(a) ML-R

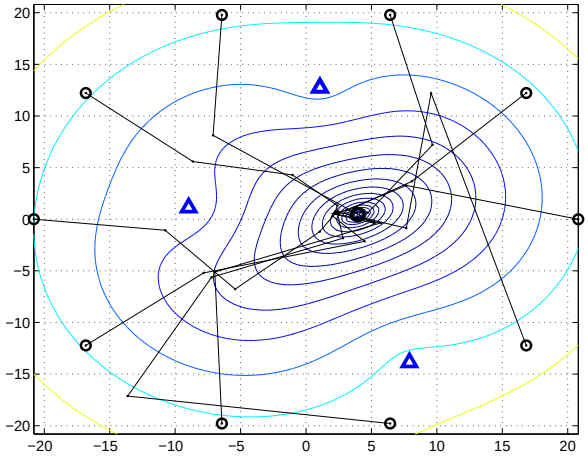


(b) ML-SR

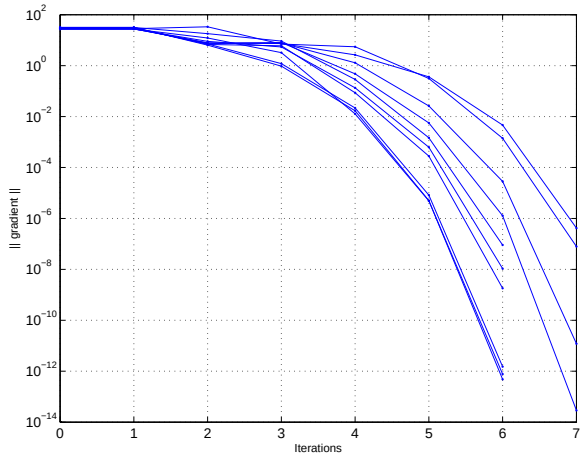


(c) ML-CSR

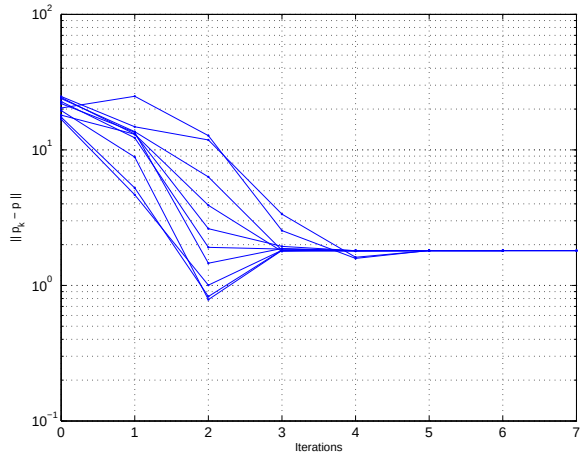
Figure 2.10: Level curves of the ML cost functions with $\mathbf{R} = \mathbf{I}_m$, poor landmark geometry, and true vehicle far from landmark centroid such that condition (2.93) is not satisfied.



(a) ML-R Iterations

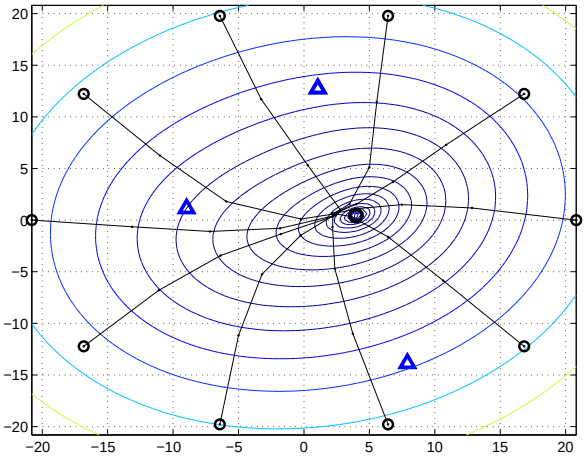


(b) ML-R $\|\nabla f(\widehat{\mathbf{p}})_k\|$

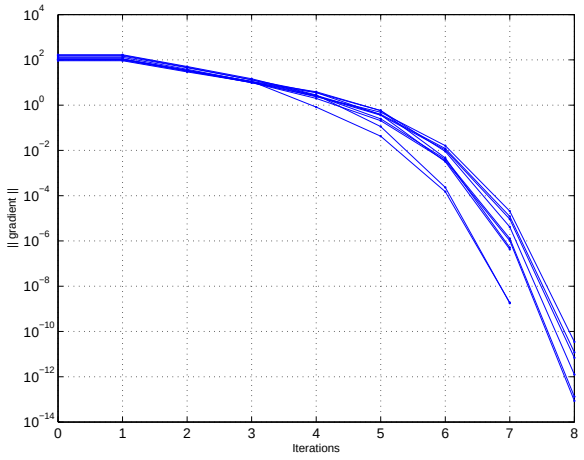


(c) ML-R $\|\widehat{\mathbf{p}}_k - \mathbf{p}\|$

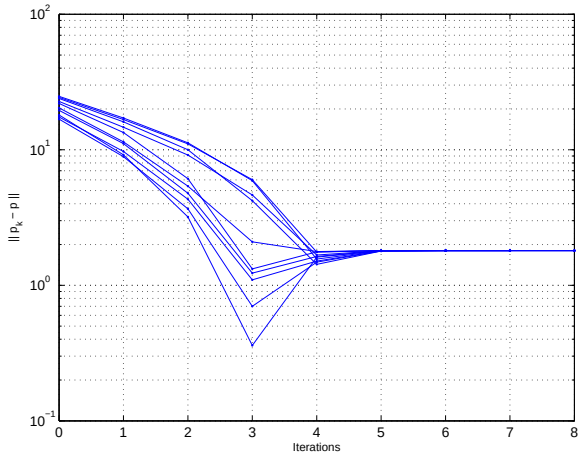
Figure 2.11: Numerical simulations of ML-R Newton minimization for different initial positions. Good landmark geometry and true vehicle near landmark centroid. All the solutions converge to the true value (although there are non desired critical points).



(a) ML-SR Montecarlo Simulations

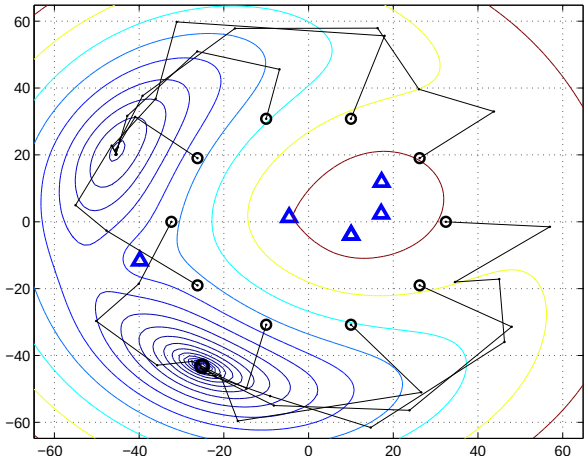


(b) ML-SR $\|\nabla f(\widehat{\mathbf{p}})_k\|$

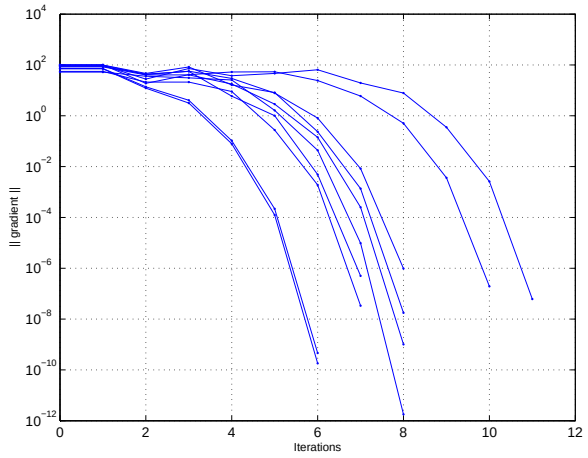


(c) ML-SR $\|\widehat{\mathbf{p}}_k - \mathbf{p}\|$

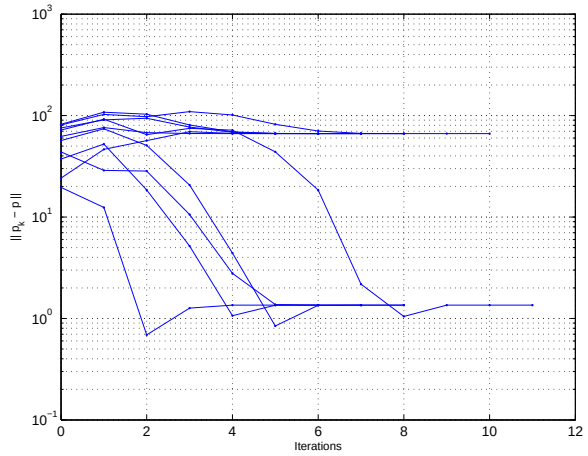
Figure 2.12: Numerical simulations of ML-SR Newton minimization for different initial positions. Good landmark geometry and true vehicle near landmark centroid. ML-SR is convex, and all the solutions converge to the true value.



(a) ML-R Montecarlo Simulations

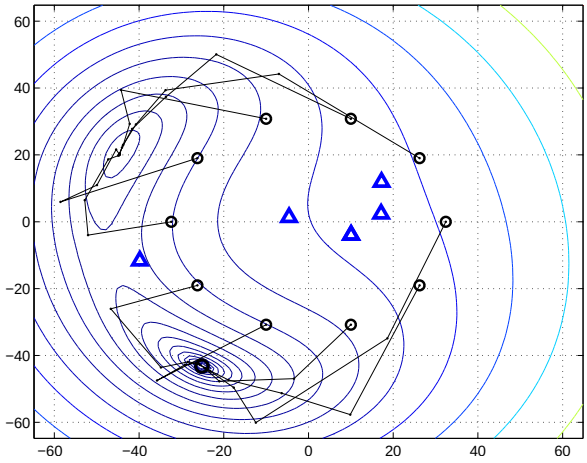


(b) ML-R $\|\nabla f(\widehat{\mathbf{p}})_k\|$

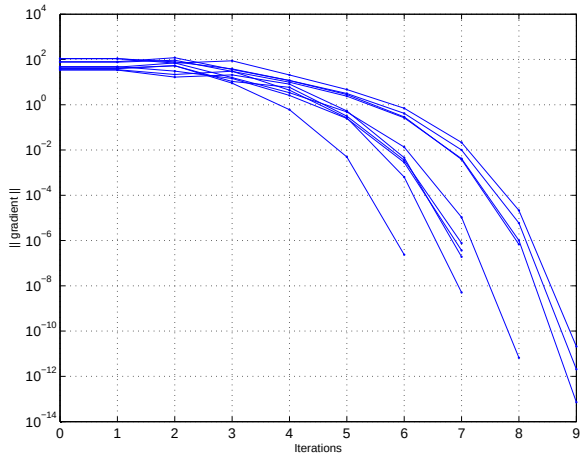


(c) ML-R $\|\widehat{\mathbf{p}}_k - \mathbf{p}\|$

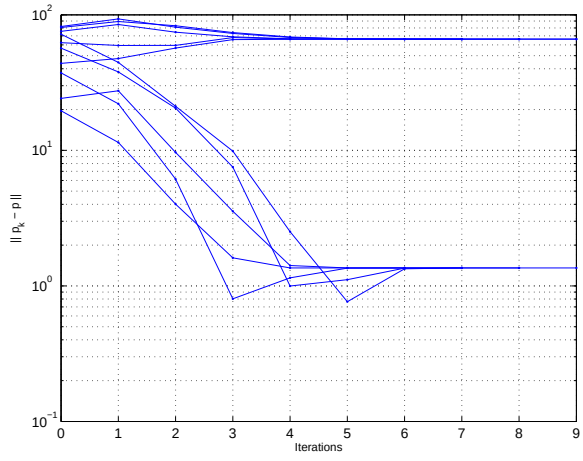
Figure 2.13: Numerical simulations of ML-R Newton minimization for different initial positions. Poor landmark geometry and true vehicle far away from landmark centroid. Some of the solutions converge to a local minimum.



(a) ML-SR Montecarlo Simulations



(b) ML-SR $\|\nabla f(\widehat{\mathbf{p}})_k\|$



(c) ML-SR $\|\widehat{\mathbf{p}}_k - \mathbf{p}\|$

Figure 2.14: Numerical simulations of ML-SR Newton minimization for different initial positions. Poor landmark geometry and true vehicle far away from landmark centroid. ML-SR is not convex and some of the solutions converge to a local minimum.

2.5 Performace bounds and landmark geometry

When faced with an estimation problem it is of extremely practical and theoretical importance to determine the best performance that can possibly be achieved with any estimator. In fact, whenever a set of noisy observations are used to estimate a certain parameter of interest, it is intuitive to think that there are some limitations on the minimum size of the estimation errors that can be obtained dictated by the stochastic nature of the observations. Even if infinite computing resources were available or the most fancy estimator were used, it would not be possible to overcome certain fundamental limitations. These performance bounds, thus obtained can be useful in many different ways. For instance, in order to asses if certain specifications on the estimation errors can be met, or as a reference against to which benchmark different estimators. Another application in which these bounds turn out to be valuable is in sensor allocation and estimator design. Because they help to identify what variables have a greater impact on the estimation error, and what sensor configurations lead to better estimation results. In the Range-Only trilateration problem that we are studying, performance bounds can be used to analyze and determine which landmark configurations are more favorable for a given problem.

As a side note, recall that it is very easy to derive lower bounds on the performance of estimators. For instance, zero is a lower bound to the performance of any estimator. However, it is a completely useless bound. What it is important is to derive tight lower bounds, that is, bounds which capture as close as possible the fundamental limitations inherent to the estimation problem. Among these bounds, one of the most widely used is the Cramér-Rao Bound (CRB) [143], [170], [95]. The CRB sets a lower bound on the performance of unbiased estimators that use observations according to a certain probability density function.

2.5.1 The Cramér-Rao Bound (CRB)

To begin, let us introduce some notation and terminology that is common to estimation theory. What follows is mainly taken from [190]. In the standard setup, one is interested in estimating an unknown parameter $\boldsymbol{\theta} \in \mathbb{R}^n$ using a set of noisy observations $\mathbf{y} \in \mathbb{R}^m$. Moreover, we are given a parametric family $\{p_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \mathbb{R}^n\}$ of positive probability density functions of $\mathbb{R}^m$. That is, for each possible value of the parameter $\boldsymbol{\theta} \in \mathbb{R}^n$, the function $p_{\boldsymbol{\theta}} : \mathbb{R}^m \to \mathbb{R}$, $\mathbf{y} \mapsto p_{\boldsymbol{\theta}}(\mathbf{y})$ is the probability of obtaining the observation $\mathbf{y}$ given that the true parameter is $\boldsymbol{\theta}$. Recall that this is a more general and precise way of defining the *likelihood* function than that introduced in section 2.4. There, we used $p(\mathbf{y}|\boldsymbol{\theta})$ for what we now denote as $p_{\boldsymbol{\theta}}(\mathbf{y})$. The second notation stresses the fact that $\boldsymbol{\theta}$ is fixed and we expect that this will not cause confusion. An estimator $\widehat{\boldsymbol{\theta}}$ is a mapping between the observation and the parameter spaces $\widehat{\boldsymbol{\theta}} : \mathbb{R}^m \to \mathbb{R}^n$, written $\mathbf{y} \mapsto \widehat{\boldsymbol{\theta}}(\mathbf{y})$. The estimator is said to be unbiased if $\mathrm{E}\left\{\widehat{\boldsymbol{\theta}}(\mathbf{y})\right\} = \boldsymbol{\theta}$ for all $\boldsymbol{\theta} \in \mathbb{R}^n$.

Let $\widehat{\boldsymbol{\theta}} : \mathbb{R}^m \to \mathbb{R}^n$ be an unbiased estimator. Under some regularity conditions on the probability density functions $p_{\boldsymbol{\theta}}(\cdot)$, the CRB theorem states that

$$\boxed{\mathsf{Cov}\left\{\widehat{\boldsymbol{\theta}}\right\} \succeq \mathbf{I}(\boldsymbol{\theta})^{-1},} \tag{2.117}$$

where

$$\mathsf{Cov}\left\{\widehat{\boldsymbol{\theta}}\right\} := \mathrm{E}\left\{(\widehat{\boldsymbol{\theta}}(\mathbf{y}) - \boldsymbol{\theta})(\widehat{\boldsymbol{\theta}}(\mathbf{y}) - \boldsymbol{\theta})^T\right\}, \tag{2.118}$$

and $\mathbf{I}(\boldsymbol{\theta})$ denotes the Fisher Information Matrix (FIM) defined as

$$\boxed{\mathbf{I}(\boldsymbol{\theta}) = \mathrm{E}\left\{(\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{y}))(\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{y}))^T\right\}.} \tag{2.119}$$

The symbol $\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{y})$ stands for the gradient of the (assumed differentiable) function $\mathbb{R}^n \to \mathbb{R}$, $\mathbf{x} \mapsto \log p_{\mathbf{x}}(\mathbf{y})$ evaluated at $\mathbf{x} \in \mathbb{R}^n$. Relating this to the material in section 2.4, it can be seen that $\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{y})$ denotes the gradient of the *log likelihood* function with respect to the unknown parameters. Moreover, for symmetric matrices $\mathbf{A}$ and $\mathbf{B}$, the notation $\mathbf{A} \succeq \mathbf{B}$ means that the difference $\mathbf{A} - \mathbf{B}$ is positive semi-definite. The inequality in (2.117) is usually refereed as the CRB covariance inequality.

Taking the trace of both sides at the covariance inequality we obtain a new inequality, usually refereed as the accuracy inequality, which sets a fundamental lower bound on the mean-square error of any unbiased estimator, given by

$$\boxed{\mathsf{var}\left\{\widehat{\boldsymbol{\theta}}\right\} := \mathrm{tr}\left(\mathsf{Cov}\left\{\widehat{\boldsymbol{\theta}}\right\}\right) = \mathrm{E}\left\{\|\widehat{\boldsymbol{\theta}}(\mathbf{y}) - \boldsymbol{\theta}\|^2\right\} \geq \mathrm{tr}\left(\mathbf{I}(\boldsymbol{\theta})^{-1}\right).} \tag{2.120}$$

Example: Additive zero-mean Gaussian measurement errors

Suppose one wants to estimate a parameter $\boldsymbol{\theta} \in \mathbb{R}^n$ from a vector of noisy observations $\mathbf{y} \in \mathbb{R}^m$ given by

$$\mathbf{y} = h(\boldsymbol{\theta}) + \mathbf{w}, \tag{2.121}$$

where $h : \mathbb{R}^n \to \mathbb{R}^m$ is some smooth nonlinear mapping, and $\mathbf{w} \in \mathbb{R}^m$ is a Gaussian error vector with zero mean and covariance $\mathrm{E}\left\{\mathbf{w}\mathbf{w}^T\right\} = \mathbf{R} \in \mathbb{R}^{m \times m}$. The *likelihood* function $p_{\boldsymbol{\theta}} : \mathbb{R}^m \to \mathbb{R}$ can be written as

$$p_{\boldsymbol{\theta}}(\mathbf{y}) := \frac{1}{(2\pi)^{\frac{m}{2}} |\mathbf{R}|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{y} - h(\boldsymbol{\theta}))^T \mathbf{R}^{-1} (\mathbf{y} - h(\boldsymbol{\theta}))\right\}, \tag{2.122}$$

and the *log likelihood* function as

$$\log p_{\boldsymbol{\theta}}(\mathbf{y}) = C - \frac{1}{2}(\mathbf{y} - h(\boldsymbol{\theta}))^T \mathbf{R}^{-1} (\mathbf{y} - h(\boldsymbol{\theta})), \tag{2.123}$$

where $C = -\log(2\pi)^{\frac{m}{2}} |\mathbf{R}|^{\frac{1}{2}}$ is a constant term. The gradient of the *log likelihood* with respect to $\boldsymbol{\theta}$ can be computed as

$$\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{y}) = \nabla h(\boldsymbol{\theta})^T \nabla h(\boldsymbol{\theta})^T \mathbf{R}^{-1}(\mathbf{y} - h(\boldsymbol{\theta})), \tag{2.124}$$

and the Fisher Information Matrix as

$$\begin{aligned}
\mathbf{I}(\boldsymbol{\theta}) &= \mathrm{E}\left\{(\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{y}))(\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{y}))^T\right\} \\
&= \mathrm{E}\left\{\nabla h(\boldsymbol{\theta})^T \mathbf{R}^{-1}(\mathbf{y} - h(\boldsymbol{\theta}))(\mathbf{y} - h(\boldsymbol{\theta}))^T \mathbf{R}^{-1} \nabla h(\boldsymbol{\theta})\right\} \\
&= \nabla h(\boldsymbol{\theta})^T \mathbf{R}^{-1} \mathrm{E}\left\{(\mathbf{y} - h(\boldsymbol{\theta}))(\mathbf{y} - h(\boldsymbol{\theta}))^T\right\} \mathbf{R}^{-1} \nabla h(\boldsymbol{\theta}) \\
&= \nabla h(\boldsymbol{\theta})^T \mathbf{R}^{-1} \mathrm{E}\left\{\mathbf{w}\mathbf{w}^T\right\} \mathbf{R}^{-1} \nabla h(\boldsymbol{\theta}) \\
&= \nabla h(\boldsymbol{\theta})^T \mathbf{R}^{-1} \mathbf{R} \mathbf{R}^{-1} \nabla h(\boldsymbol{\theta}) \\
&= \nabla h(\boldsymbol{\theta})^T \mathbf{R}^{-1} \nabla h(\boldsymbol{\theta}).
\end{aligned} \tag{2.125}$$

Now it is easy to compute the covariance and accuracy CRB inequalities. The zero mean Gaussian additive noise is quite common because of its simplicity. It is the same error model that was considered in the Range-Only localization problem.

2.5.2 The CRB for the trilateration problem

It is now simple to derive the CRB bounds for the Range-Only positioning problem. Suppose one wants to estimate the vehicle position $\mathbf{p} \in \mathbb{R}^n$ from a vector of range measurements $\bar{\mathbf{r}} = \mathbf{r}(\mathbf{p}) + \mathbf{w} \in \mathbb{R}^m$ where $\mathbf{w}$ is zero mean Gaussian with covariance $\mathbf{R}$. Each entry of vector $\mathbf{r}(\mathbf{p})$ has the form

$$r_i(\mathbf{p}) = \|\mathbf{p} - \mathbf{p}_i\| = (\mathbf{p}^T\mathbf{p} - 2\mathbf{p}_i^T\mathbf{p} + \mathbf{p}_i^T\mathbf{p}_i)^{\frac{1}{2}}. \tag{2.126}$$

Now we can compute

$$\nabla r_i(\mathbf{p}) = \frac{1}{2}(\mathbf{p}^T\mathbf{p} - 2\mathbf{p}_i^T\mathbf{p} + \mathbf{p}_i^T\mathbf{p}_i)^{-\frac{1}{2}}(2\mathbf{p}^T - 2\mathbf{p}_i^T) \tag{2.127}$$

$$= \frac{1}{r_i(\mathbf{p})}(\mathbf{p} - \mathbf{p}_i)^T, \tag{2.128}$$

and

$$\nabla r(\mathbf{p}) = \begin{bmatrix} \nabla r_1(\mathbf{p}) \\ \vdots \\ \nabla r_m(\mathbf{p}) \end{bmatrix} = \begin{bmatrix} \frac{1}{r_1(\mathbf{p})} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{r_m(\mathbf{p})} \end{bmatrix} \begin{bmatrix} (\mathbf{p} - \mathbf{p}_1)^T \\ \vdots \\ (\mathbf{p} - \mathbf{p}_m)^T \end{bmatrix} \tag{2.129}$$

$$= \delta(\mathbf{r})^{-1}(\mathbf{p}\mathbf{1}_m^T - \mathbf{P})^T = \delta(\mathbf{r})^{-1}\mathbf{C}^T, \tag{2.130}$$

where $\mathbf{C} = (\mathbf{p}\mathbf{1}_m^T - \mathbf{P}) \in \mathbb{R}^{n\times m}$ is the same matrix that was already defined in (2.81). According to (2.125), the Fisher Information Matrix is given by

$$\mathbf{I}(\mathbf{p}) = \nabla r(\mathbf{p})^T \mathbf{R}^{-1} \nabla r(\mathbf{p}) = \mathbf{C}\delta(\mathbf{r})^{-1}\mathbf{R}^{-1}\delta(\mathbf{r})^{-1}\mathbf{C}^T. \tag{2.131}$$

The CRB inequalities can be written as follow. Suppose that $\widehat{\mathbf{p}} : \mathbb{R}^m \to \mathbb{R}^n$ is some unbiased estimator of the vehicle position that uses as observations the noisy range measurements $\bar{\mathbf{r}}$. Then

$$\boxed{\mathrm{E}\left\{(\widehat{\mathbf{p}}(\bar{\mathbf{r}}) - \mathbf{p})(\widehat{\mathbf{p}}(\bar{\mathbf{r}}) - \mathbf{p})^T\right\} \succeq \left(\mathbf{C}\delta(\mathbf{r})^{-1}\mathbf{R}^{-1}\delta(\mathbf{r})^{-1}\mathbf{C}^T\right)^{-1},} \tag{2.132}$$

and

$$\boxed{\mathsf{var}\left\{\widehat{\mathbf{p}}\right\} = \mathrm{E}\left\{\|\widehat{\mathbf{p}}(\bar{\mathbf{r}}) - \mathbf{p}\|^2\right\} \geq \mathrm{tr}\left(\left(\mathbf{C}\delta(\mathbf{r})^{-1}\mathbf{R}^{-1}\delta(\mathbf{r})^{-1}\mathbf{C}^T\right)^{-1}\right).} \tag{2.133}$$

When an estimator attains the CRB it is said that the estimator is *efficient* [170],[95]. This is the best that one can expect and it is not warranted that an efficient estimator will exist. Under certain conditions, the Maximum Likelihood estimator can be shown to be efficient, but this is not true in general. Numerical simulations suggest that this is true for the trilateration problem, so that the estimation errors produced by the the ML-R and ML-SR algorithms are always very close to the CRB. However, it will be shown that the LS-C and LS-CW algorithms are not efficient, and the errors they produce depart from the CRB.

The CRB accuracy inequality bound is represented in Figures 2.15 and 2.16 for different number of landmarks and geometries. The figures were obtained by computing the value of the CRB bound (2.133) at every point of an equally spaced grid and using MATLAB to represent the contours of the values. The measurement error covariance was always set to $\mathbf{R} = \mathbf{I}_m$ where m was 3 or 4 depending on the number of landmarks of each figure. This corresponds to a standard deviation of $\sigma_{ii} = 1\text{m}$ in the range measurements. Note that by using these types of plots, one can estimate a priori what will be the minimum errors achievable for a given noise covariance and landmark geometry. In practice one does not known the exact value of the error covariance, but still this plots can serve to provide rough estimates of the RMSE expected at each regions. Moreover, they can prove very valuable when determining the optimal location of the landmarks to minimize the estimations errors in a particular are of operation. It can be seen from the plots that the estimation errors are smaller near the centroid of the landmarks and increase greatly as one moves away from it.

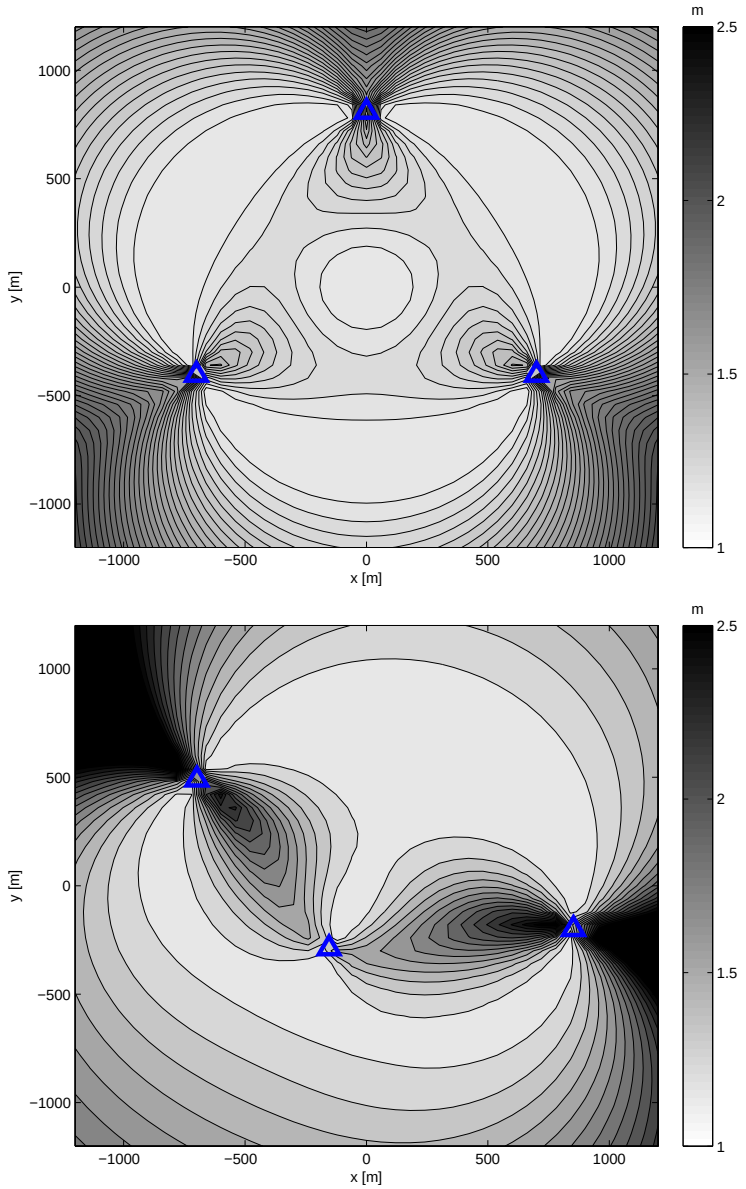


Figure 2.15: CRB root mean square error theoretical bound (accuracy inequality) for different landmark geometries and $\mathbf{R} = \mathbf{I}_m$.

2.5.3 Trilateration algorithms performance analysis

We have now introduced all the tools needed to investigate the performance of the proposed Range-Only localization algorithms. In this section we will analyze through numerical Monte Carlo simulations the estimation errors produced by the proposed algorithms. The algorithms that are the subject of study will be the Centered Least Squares (LS-C), the Centered Weighted Least Squares (LS-CW), the Generalized Trust Region Subproblem Least Squares (LS-GTRS), the Newton minimization of the Maximum likelihood with Ranges (ML-R), and the Newton minimization of the Maximum likelihood with Squared Ranges (ML-SR). In all the situations the

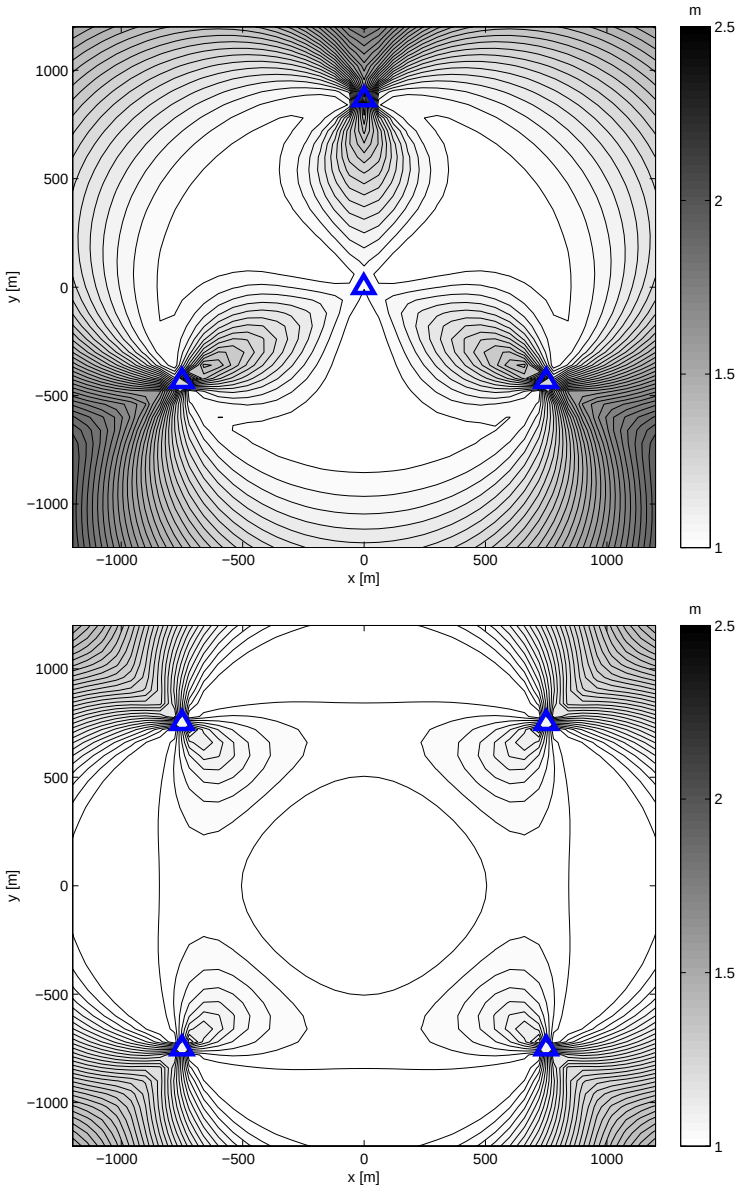


Figure 2.16: CRB root mean square error theoretical bound (accuracy inequality) for different landmark geometries and $\mathbf{R} = \mathbf{I}_m$.

range measurement error covariance was set to the identity matrix $\mathbf{I}_m$, where m is the number of landmarks in each case. Note that technically, the LS-C and LS-CW algorithms are not unbiased, so that strictly speaking they do not fulfill the requirements of the CRB. We will see that in practice this has little influence on the numerical results.

In order to illustrate the performance of the algorithms, three case study vehicle positions and a typical landmark geometry were chosen. Figure 2.17 shows the setup for the simulation scenario. Four landmarks were located in the corners of a deformed square of approximately 1000m side. Three points were considered with

$\widehat{\mathbf{p}}_0$	$\{\mathbf{p}_A, \mathbf{p}_B, \mathbf{p}_C\}$	Initial estimate is taken as the true position
ϵ	$1e-8$	Threshold on the gradient $\nabla f(\widehat{\mathbf{p}}_k)$
k_{max}	25	Maximum number of iterations allowed
s	1	Armijo line search parameter
β	0.5	Armijo line search parameter
σ	0.1	Armijo line search parameter

Table 2.1: Newton algorithm parameters for the ML-R and ML-SR minimization algorithms used in the Monte Carlo simulations.

coordinates $\mathbf{p}_A = (200, -100)$m, $\mathbf{p}_B = (-300, 300)$m, and $\mathbf{p}_C = (-1000, 1000)$m. Figure 2.17(a) shows the CRB RMSE bound together with the landmark coordinates (blue triangles), and the position of the three case study vehicle positions (green circles). Figures 2.17(b) and 2.17(c) show the predicted RMSE of the LS-C and LS-CW trilateration algorithms, respectively, as given by expressions (2.48) and (2.51). At first sight, one can see that both the LS and WLS algorithms present bigger predicted RMSE's when far from the landmark centroid, but pretty close to the CRB in the center region of the landmarks. The LS-CW performs slightly better than the LS-C, with a greater region where the error remains close to the CRB. The three case study vehicle positions are taken from regions with apparently different behaviors.

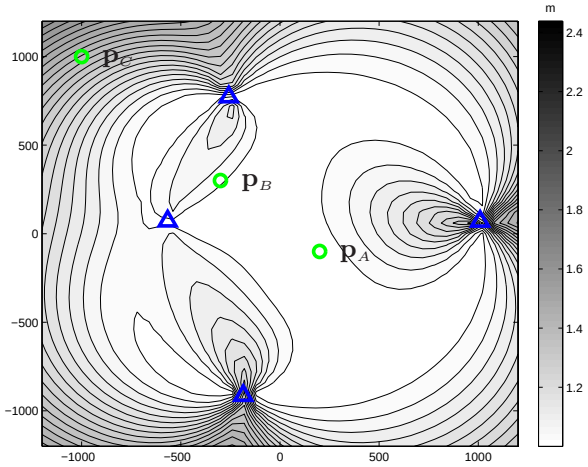
Monte Carlo simulations were performed to analyze the performance of the algorithms on the three points. Figures 2.18 - 2.20 illustrate the results. At each of the points, $N = 1000$ samples of noisy range measurement vectors with covariance $\mathbf{R} = \mathbf{I}_m$ were generated. These measurements were then used to generate N trilateration position estimates using the different algorithms. The ML-R and ML-SR cost functions were minimized using the Newton algorithm with parameters described in Table 2.5.3. These algorithms were initialized at the true vehicle positions. Note that we were only interested in the final value produced by the algorithms after convergence, so this had little influence. Figures 2.18 - 2.20 illustrate the scatter plots of the different trilateration algorithms. The results also show the 3σ confidence ellipsoids of the data computed experimentally together with the 3σ ellipsoid given by the CRB. In the LS-C and LS-CW case the plots also show the 3σ ellipsoids of the predicted theoretical covariances from (2.48) and (2.51).

In the first case, Figure 2.18, the vehicle position $\mathbf{p}_A$ is located near the center of the landmark centroid. The errors generated by the different algorithms are very similar, and are very close to the CRB. That is, the experimental 3σ ellipsoid of the data covariance almost coincides with the 3σ ellipsoid of the CRB. Moreover, the predicted 3σ ellipsoid for the LS-C and LS-CW algorithms fit the experimental data quite well. This suggests that near the landmark centroid the simple LS and LS-CW algorithms perform similar to the more sophisticated ML-R and ML-SR algorithms and seem to achieve the CRB.

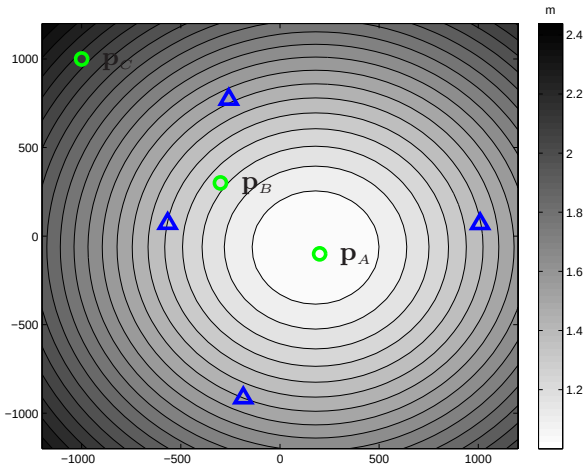
In the second case, Figure 2.19, the vehicle position $\mathbf{p}_B$ is located a little far off the center of the landmark centroid. The LS-C and LS-CW generate bigger errors than the ML-R and ML-SR, and depart slightly from the CRB. This is much more severe in the case of the LS-C (Figure 2.19(a)), while the LS-CW still remains acceptably close to the CRB (Figure 2.19(b)). The LS-GTRS does not perform very well here, and this is probably due to the fact that it does not exploit the knowledge on the covariance of the squared range measurements. The ML-R and ML-SR still produce errors close to the CRB (Figures 2.19(c) and 2.19(d)).

In the third simulation case, Figure 2.20, the vehicle position $\mathbf{p}_C$ is located far away from the center of the landmark centroid. In this situation both the LS-C and LS-CW produce estimation errors much bigger then the corresponding ML-R and ML-SR algorithms. The LS-CW still performs slightly better than the simpler LS-C algorithm. In this case the LS-GTRS outperforms both the LS-C and LS-CW algorithms, being much closer to the CRB. The ML-R and ML-SR produce errors very close to the CRB. From these simulations we conclude that apparently, the ML-R and ML-SR trilateration algorithms are efficient, in the sense that they achieve the CRB for the problem. The severity of the mismatch between the CRB and the errors of the simple LS-C and LS-CW algorithms become more important as the vehicle position moves far away from the landmark centroid. Moreover, when the vehicle position is near the centroid,all the algorithms seem to perform equally well and achieve the CRB.

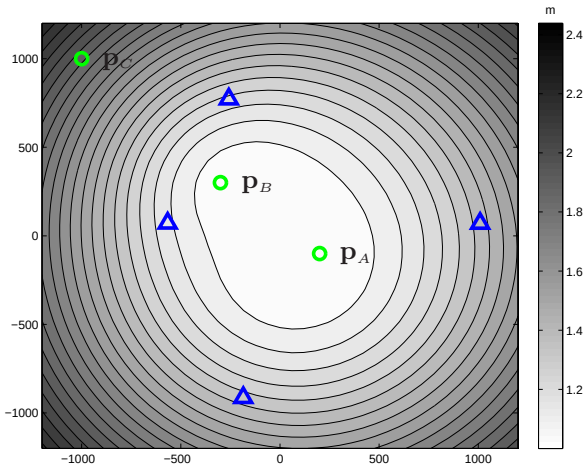
In Figures 2.21, 2.22, and 2.23 the value of the CRB and experimental RMSE obtained with the different trilateration algorithms and different measurement errors with standard deviations between $\sigma = 0.1$m and $\sigma = 10$m. The results were computed at the three different points $\mathbf{p}_A$, $\mathbf{p}_B$, and $\mathbf{p}_c$ from $N = 100$ Montecarlo simulations at each standard deviation.



(a) CRB

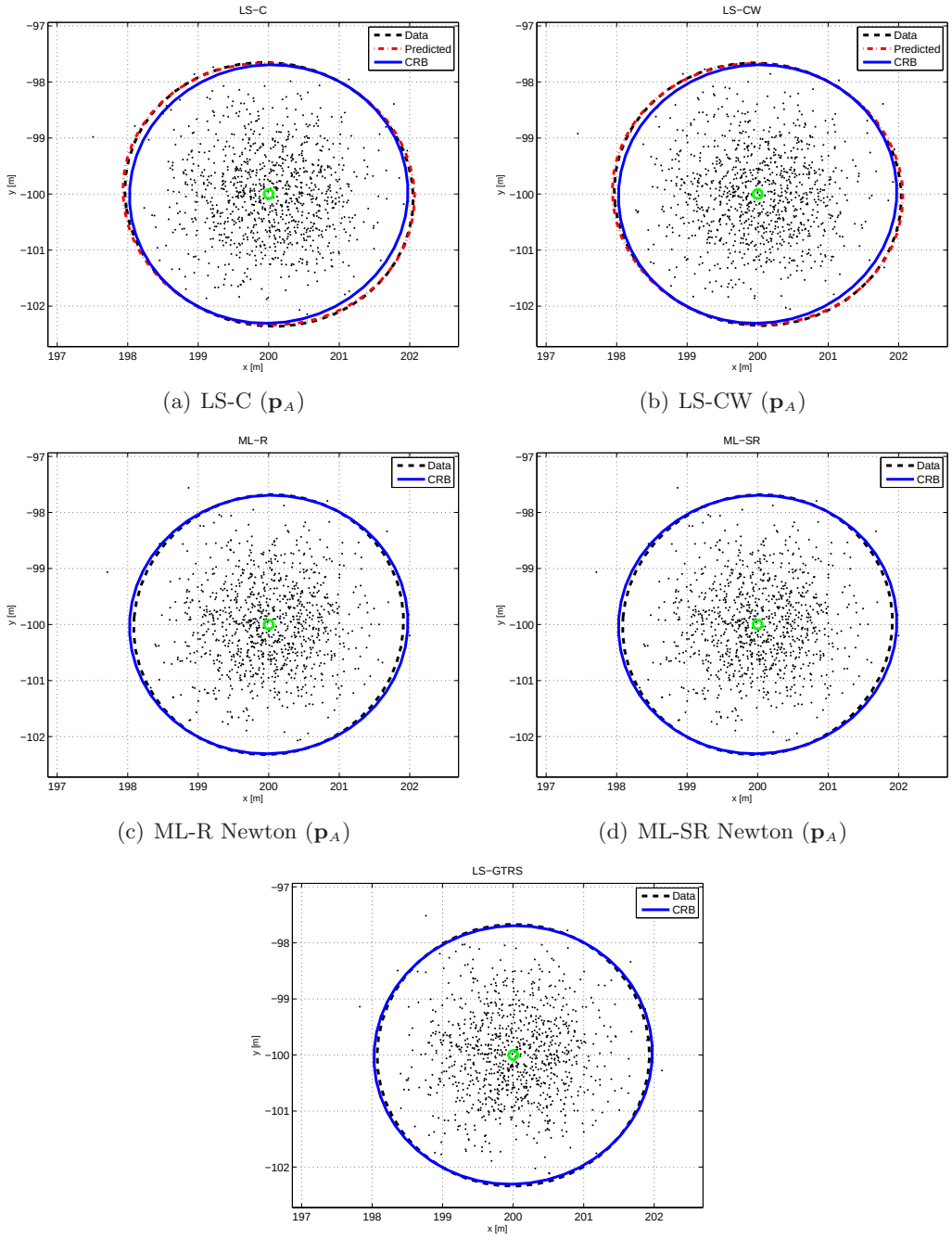


(b) LS-C



(c) LS-CW

Figure 2.17: CRB root mean square error (RMSE) theoretical bound (accuracy inequality) compared to predicted LS-C and LS-CW RMSE for $\mathbf{R} = \mathbf{I}_m$. Points $\mathbf{p}_A$, $\mathbf{p}_B$, and $\mathbf{p}_C$ of the Monte Carlo Simulations are shown on top.



(a) LS-C ($\mathbf{p}_A$)

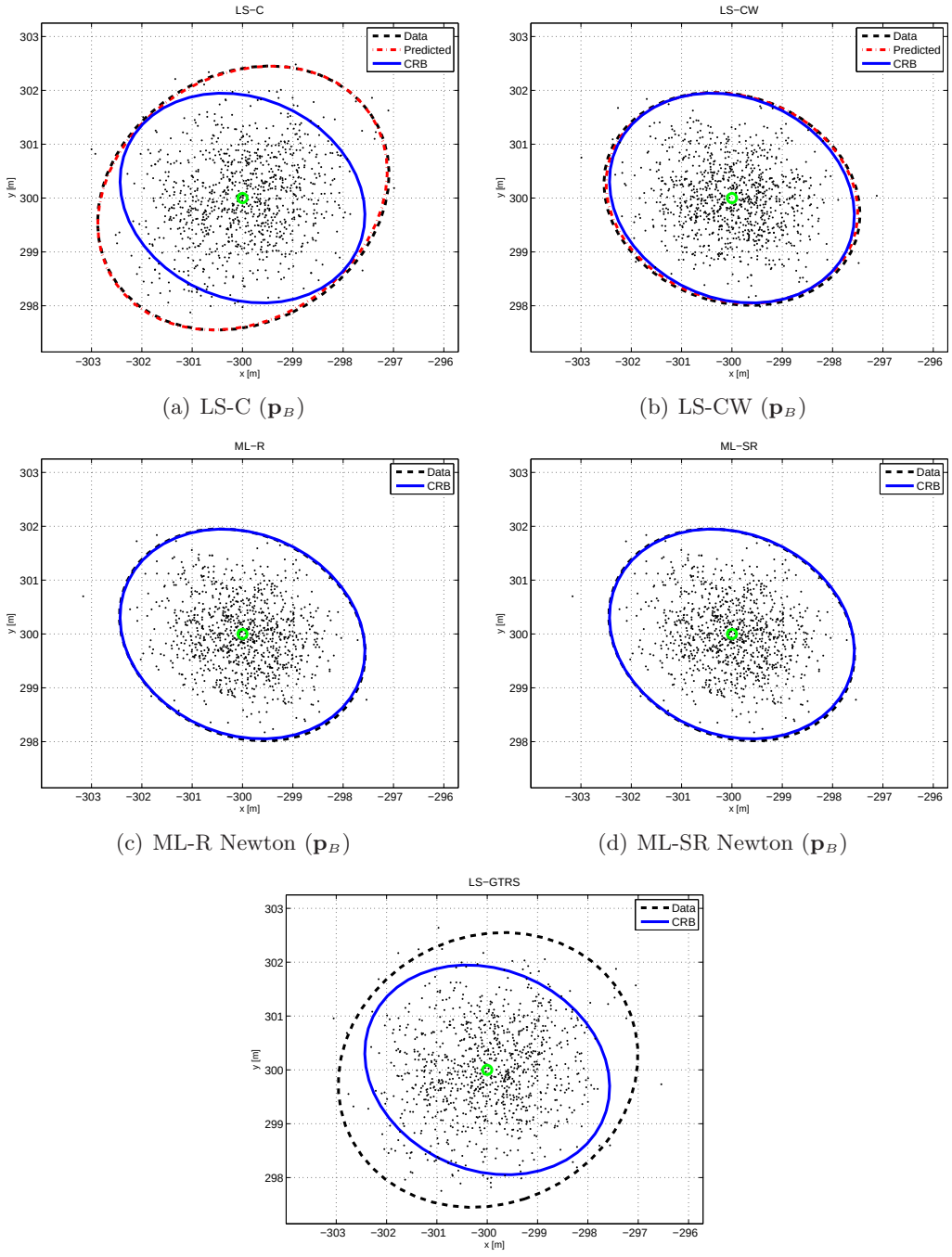
(b) LS-CW ($\mathbf{p}_A$)

(c) ML-R Newton ($\mathbf{p}_A$)

(d) ML-SR Newton ($\mathbf{p}_A$)

(e) LS-GTRS ($\mathbf{p}_A$)

Figure 2.18: Comparison of experimental estimation errors produced by LS-C, LS-CW, ML-R, ML-SR, and LS-GTRS algorithms for point $\mathbf{p}_A = (200, -100)$m, near the centroid of the landmarks. The plots also show the 3σ ellipsoids for the experimental data, the CRB, and the predicted (only for the LS-C and LS-CW case).



(a) LS-C ($\mathbf{p}_B$)

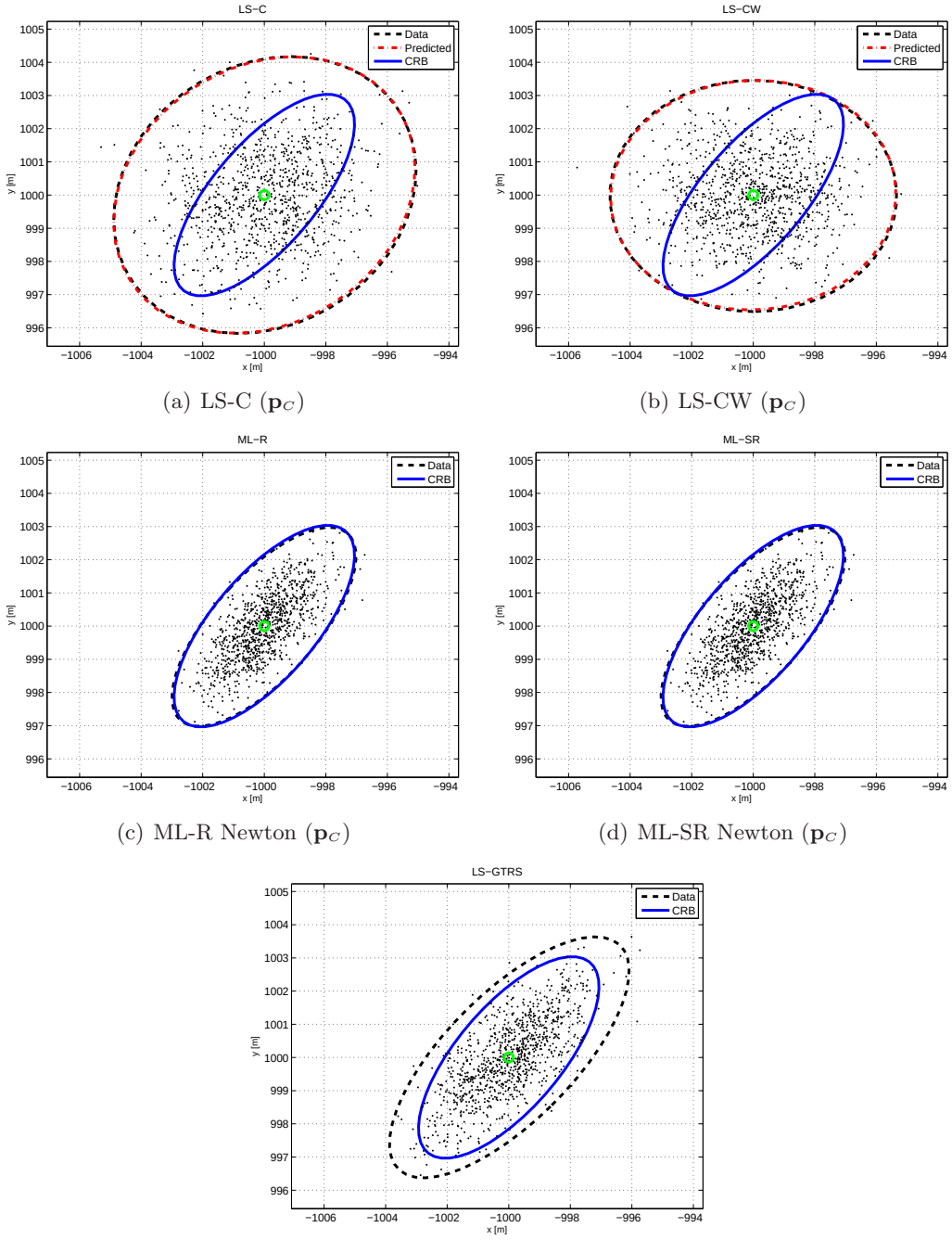
(b) LS-CW ($\mathbf{p}_B$)

(c) ML-R Newton ($\mathbf{p}_B$)

(d) ML-SR Newton ($\mathbf{p}_B$)

(e) LS-GTRS ($\mathbf{p}_B$)

Figure 2.19: Comparison of experimental estimation errors produced by LS-C, LS-CW, ML-R, ML-SR, and LS-GTRS algorithms for point $\mathbf{p}_B = (-300, 300)$m. The plots also show the 3σ ellipsoids for the experimental data, the CRB, and the predicted (only for the LS-C and LS-CW case).



(a) LS-C ($\mathbf{p}_C$)

(b) LS-CW ($\mathbf{p}_C$)

(c) ML-R Newton ($\mathbf{p}_C$)

(d) ML-SR Newton ($\mathbf{p}_C$)

(e) LS-GTRS ($\mathbf{p}_C$)

Figure 2.20: Comparison of experimental estimation errors produced by LS-C, LS-CW, ML-R, ML-SR, and LS-GTRS algorithms for point $\mathbf{p}_C = (-1000, 1000)$m. The plots also show the 3σ ellipsoids for the experimental data, the CRB, and the predicted (only for the LS-C and LS-CW case).

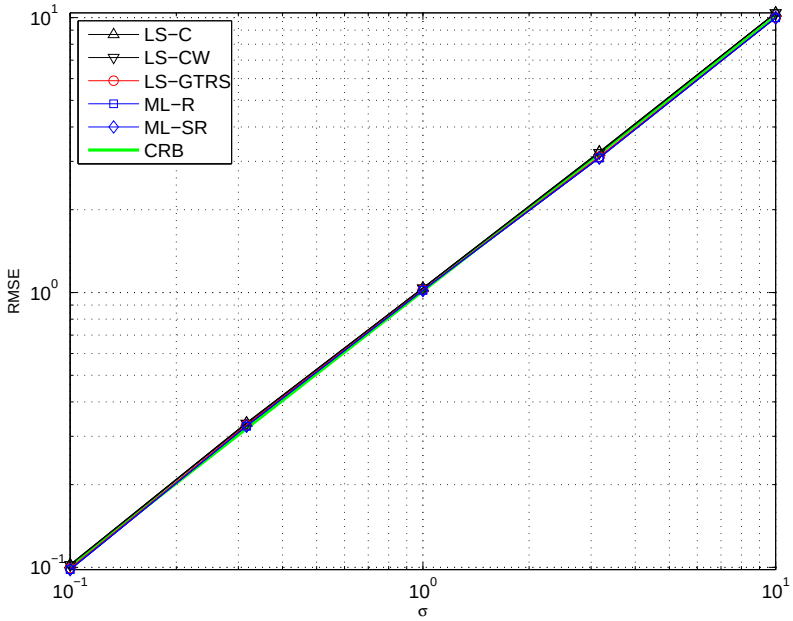


Figure 2.21: RMSE (in meters) of different trilateration algorithms at point $\mathbf{p}_A = (200, -100)$ for different measurement error covariances $\mathbf{R} = \sigma^2\mathbf{I}_m$, where σ varies between 0.1m and 10m.

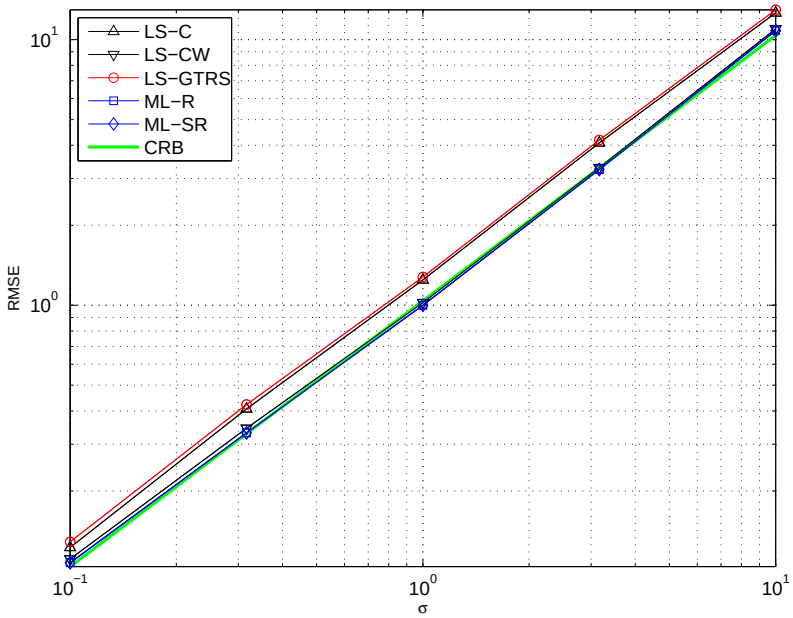


Figure 2.22: RMSE (in meters) of different trilateration algorithms at point $\mathbf{p}_B = (-300, 300)$ for different measurement error covariances $\mathbf{R} = \sigma^2\mathbf{I}_m$, where σ varies between 0.1m and 10m.

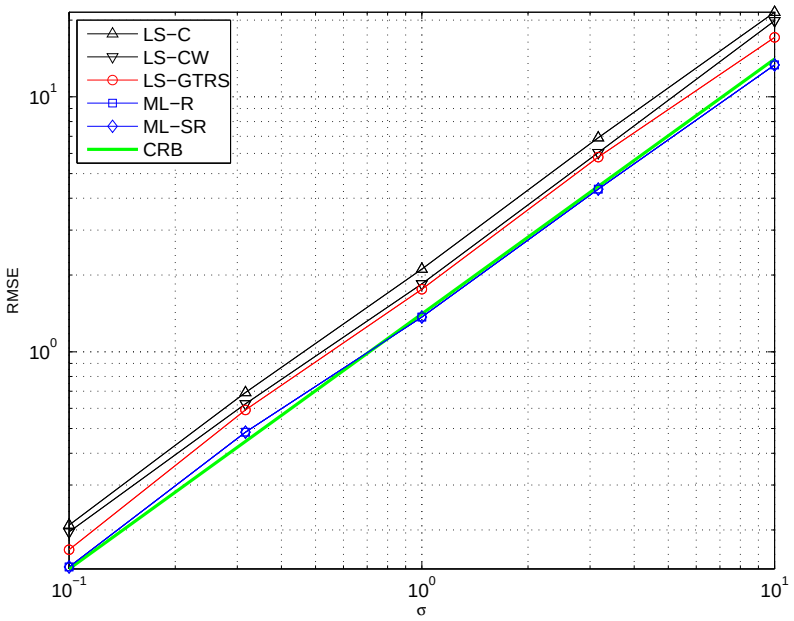


Figure 2.23: RMSE (in meters) of different trilateration algorithms at point $\mathbf{p}_C = (-1000, 1000)$ for different measurement error covariances $\mathbf{R} = \sigma^2 \mathbf{I}_m$, where σ varies between 0.1m and 10m.

2.6 Filtering and dynamic algorithms

So far, we have considered the Range-Only positioning problem in the static case. That is, the vehicle position was fixed and we wanted to determine its position from a vector of range measurements corresponding to an acoustic emission at a single instant of time. In practice, the vehicle is seldom static and describes some arbitrary trajectory in space. Moreover, the range measurements are obtained as the vehicle moves and are available at a certain measurement rate.

By exploiting the temporal diversity of the problem and knowledge about the kinematics or dynamics of the vehicle, better performances can be obtained in terms of estimation errors as compared to the case where a fixed trilateration problem is solved at each time instant. One should make a clear distinction between two different classes of systems henceforth referred to as Tracking and Navigation systems. Tracking systems aim at obtaining a position estimate of a vehicle that is tracked externally using range measurements, so that little information about the vehicle internal states (velocities, accelerations, attitude, depth) is available. In Navigation systems, one has access to some of the vehicle internal states (measured by some other sensors) and wishes to integrate and fuse them with the range measurements in order to obtain an improved position estimate.

Most of the difficulties of estimation with range-only measurements come from the fact that the measurements are naturally nonlinear. This prevents direct application of classic linear filtering and estimation techniques. The most naive and simple solution is to determine at each instant of time a position fix, using for instance some of the trilateration algorithms previously presented, and use this position fixes to feed some linear filter such as a Kalman Filter (KF). This has the advantage of being quite simple and reliable. Moreover, the KF measurement error covariances matrices can be easily obtained a priori: In the case of LS-C and LS-CW one can use (2.48) and (2.51) respectively, and in the case of Maximum Likelihood estimators one can use the CRB covariance (2.117).

However this naive approach has some major disadvantages. It is necessary that at each time instant a full set of range measurements, enough to uniquely determine a position fix, be available. That is, 4 ranges (from non coplanar landmarks) in the 3D case and 3 ranges (from non collinear landmarks) in the 2D case. Moreover, those ranges should correspond to the same instant of time. In practice, this meets with some difficulties. First, it may happen that at some instants of time some of the range measurements are not available, say due to communication failures or because some of them turn out to be outliers. In this case, if not enough valid range measurements are available to uniquely compute a position, the remaining measurements can not be used. However, it is clear that although these remaining ranges might not be enough to uniquely determine a position fix, they certainly contain valuable information that one should be able to use.

A more sophisticated solution consists of using each individual range measurement in the position estimation process by resorting to an Extended Kalman Filter structure. This type of filters can extract information from less number of ranges (than those required to uniquely determine a position) but raise some important issues about observability and convergence. Convergence is compromised by the nonlinear nature of the measurements and it is common that only local convergence can be guaranteed.

2.6.1 Target Tracking Maneuvering models

This section will focus on the tracking subproblem. Depending on the degree of knowledge about the vehicle states and type of trajectories, there are different vehicle models that can be used [24] [113].

For vehicle maneuvering model we define some smooth mapping $f : \mathbb{R}^N \times \mathbb{R}^q \times \mathbb{N} \rightarrow \mathbb{R}^N$ such that

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \boldsymbol{\xi}_k, k), \tag{2.134}$$

where $\mathbf{x}$ is the state, $\boldsymbol{\xi}$ is some vector of disturbance parameters. One of the simplest and most useful models is the Random Walk with Constant Velocity linear model (RWCV) with state $\mathbf{x} = [\mathbf{p}^T \ \mathbf{v}^T]^T \in \mathbb{R}^{2n}$, given by

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{p}_{k+1} \\ \mathbf{v}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_k + h\mathbf{v}_k + \boldsymbol{\xi}_{pk} \\ \mathbf{v}_k + \boldsymbol{\xi}_{vk} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_n & h\mathbf{I}_n \\ 0 & \mathbf{I}_n \end{bmatrix} \begin{bmatrix} \mathbf{p}_k \\ \mathbf{v}_k \end{bmatrix} + \begin{bmatrix} \boldsymbol{\xi}_{pk} \\ \boldsymbol{\xi}_{vk} \end{bmatrix} \tag{2.135}$$

$$= \mathbf{A}\mathbf{x}_k + \boldsymbol{\xi}_k, \tag{2.136}$$

where $\boldsymbol{\xi}_k \in \mathbb{R}^{2n}$ is a Gaussian variable that models how much the vehicle trajectory deviates from a straight line. By modifying the covariance of $\boldsymbol{\xi}_k$, $\mathbf{Q}_k = \mathrm{E}\left\{\boldsymbol{\xi}_k \boldsymbol{\xi}_k^T\right\}$, very different behaviors can be represented. Small covariances represent vehicles with slow dynamics, almost straight line trajectories with large radii of curvature, while high covariances capture fast vehicle dynamics, with fast turns of small radii of curvature.

Another common model is the Random Walk with Constant Acceleration (RWCA) linear model with state $\mathbf{x} = [\mathbf{p}^T \ \mathbf{v}^T \ \mathbf{a}^T]^T \in \mathbb{R}^{3n}$, given by

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{p}_{k+1} \\ \mathbf{v}_{k+1} \\ \mathbf{a}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_k + h\mathbf{v}_k + \boldsymbol{\xi}_{pk} \\ \mathbf{v}_k + h\mathbf{a}_k + \boldsymbol{\xi}_{vk} \\ \mathbf{a}_k + + \boldsymbol{\xi}_{ak} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_n & h\mathbf{I}_n & 0 \\ 0 & \mathbf{I}_n & h\mathbf{I}_n \\ 0 & 0 & \mathbf{I}_n \end{bmatrix} \begin{bmatrix} \mathbf{p}_k \\ \mathbf{v}_k \\ \mathbf{a}_k \end{bmatrix} + \begin{bmatrix} \boldsymbol{\xi}_{pk} \\ \boldsymbol{\xi}_{vk} \\ \boldsymbol{\xi}_{ak} \end{bmatrix} \tag{2.137}$$

$$= \mathbf{A}\mathbf{x}_k + \boldsymbol{\xi}. \tag{2.138}$$

A different kind of models use a speed/direction (V, ψ) representation of vehicle velocity vector that can be used to to model vehicle turns of constant angular velocity

r. Defining the state $\mathbf{x} = [x\ y\ V\ \psi\ r] \in \mathbb{R}^5$ one can write the Random Walk with Constant Turning Rate (RWCTR) model

$$\mathbf{x}_{k+1} = \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ V_{k+1} \\ \psi_{k+1} \\ r_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + hV_k \cos(\psi_k) + \xi_{xk} \\ y_k + hV_k \sin(\psi_k) + \xi_{yk} \\ V_k + \xi_{vk} \\ \psi_k + r_k + \xi_{\psi k} \\ r_k + \xi_{rk} \end{bmatrix} \tag{2.139}$$

$$= \begin{bmatrix} 1 & 0 & h\cos(\psi_k) & 0 & 0 \\ 0 & 1 & h\sin(\psi_k) & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1\ h \\ 0 & 0 & 0 & 0 & 0\ 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ V_k \\ \psi_k \\ r_k \end{bmatrix} + \begin{bmatrix} \xi_{xk} \\ \xi_{yk} \\ \xi_{Vk} \\ \xi_{\psi k} \\ \xi_{rk} \end{bmatrix} = \mathbf{A}(\psi_k)\mathbf{x}_k + \boldsymbol{\xi}_k, \tag{2.140}$$

which has the form of a linear parametrically varying linear model.

2.6.2 KF with trilateration position fixes

This is the simplest filtering scheme that one can consider. At each iteration, a position fix is computed using some trilateration algorithm and the position fixes are used as linear measurements for a standard Kalman Filter (KF).

Consider a general linear maneuvering model of the form

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \boldsymbol{\xi}_k, k) = \mathbf{A}_k\mathbf{x}_k + \mathbf{L}_k\boldsymbol{\xi}_k, \tag{2.141}$$

where $\mathbf{x}_k \in \mathbb{R}^N$, $\mathbf{A}_k \in \mathbb{R}^{N\times N}$, $\mathbf{L}_k \in \mathbb{R}^{N\times q}$, and $\boldsymbol{\xi}_k \in \mathbb{R}^q$. Note that $\mathbf{L}_k = \mathbf{I}_N$ for both the RWCV and RWCA models. Let

$$\mathrm{E}\{\mathbf{x}_0\} = \boldsymbol{\mu}_0 \in \mathbb{R}^N, \tag{2.142}$$

$$\mathrm{E}\left\{(\mathbf{x}_0 - \boldsymbol{\mu}_0)(\mathbf{x}_0 - \boldsymbol{\mu}_0)^T\right\} = \boldsymbol{\Sigma}_0 \in \mathbb{R}^{N\times N}, \tag{2.143}$$

$$\mathrm{E}\{\boldsymbol{\xi}_k\} = 0, \tag{2.144}$$

$$\mathrm{E}\left\{\boldsymbol{\xi}_k\boldsymbol{\xi}_s^T\right\} = 0\ \forall k \neq s, \tag{2.145}$$

$$\mathrm{E}\left\{\boldsymbol{\xi}_k\boldsymbol{\xi}_k^T\right\} = \mathbf{Q}_k \in \mathbb{R}^{q\times q}, \tag{2.146}$$

where $\boldsymbol{\Sigma}_0$ and $\mathbf{Q}_k$ are positive definite matrices. Suppose that at each iteration, one measures a set of $m_k \in \mathbb{N}$ range measurements $\bar{\mathbf{r}}_k = \mathbf{r}_k + \mathbf{w}_k \in \mathbb{R}^{m_k}$, with $\bar{r}_{ik} = r_{ik} + w_{ik} = \|\mathbf{p}_k - \mathbf{p}_{ik}\| + w_{ik},\ i \in \{1, \ldots, m_k\}$ where $\mathbf{p}_k$ is the position of the vehicle at iteration k, $\mathbf{p}_{ik}$ is the position of the i'th landmark at iteration k, and w_{ik} is a Gaussian measurement error. The measurement error vector has covariance $\mathrm{E}\left\{\mathbf{w}_k\mathbf{w}_k^T\right\} = \mathbf{R}_k \in \mathbb{R}^{m_k\times m_k}$.

Furthermore, suppose that at iteration k there are enough range measurements to compute a trilateration position fix (four in the 3D case and three in the 2D

case). Assume that the expected error covariance of the trilateration algorithm used is known. For instance, as in the LS (2.48), WLS (2.51) or Maximum Likelihood estimators ML-R and ML-SR the CRB (2.117). In this situation we can consider the trilateration position fixes as being linear measurements of the state, given as

$$\mathbf{y}_k = \mathbf{C}_k \mathbf{x}_k + \boldsymbol{\theta}_k = \mathbf{p}_k + \boldsymbol{\theta}_k, \in \mathbb{R}^n \tag{2.147}$$

corrupted by some zero mean, uncorrelated, Gaussian disturbance $\boldsymbol{\theta}_k \in \mathbb{R}^n$ with covariance $\mathrm{E}\left\{\boldsymbol{\theta}_k \boldsymbol{\theta}_k^T\right\} = \mathbf{S}_k \in \mathbb{R}^{n \times n}$. For instance, if the simple LS trilateration algorithm is used, then one can chose from (2.48)

$$\mathbf{S}_k = (\mathbf{P}_k \mathbf{M} \mathbf{P}_k^T)^{-1} \mathbf{P}_k \mathbf{M} \delta(\bar{\mathbf{r}}_k) \mathbf{R}_k \delta(\bar{\mathbf{r}}_k) \mathbf{M} \mathbf{P}_k^T (\mathbf{P}_k \mathbf{M} \mathbf{P}_k^T)^{-1}, \tag{2.148}$$

where $\mathbf{P}_k = [\mathbf{p}_{1k} \ \ldots \ \mathbf{p}_{mk}] \in \mathbb{R}^{n \times m_k}$ contains the coordinates of the landmarks at iteration k. Note that $\mathbf{C}_k = [\mathbf{I}_n \ 0] \in \mathbb{R}^{n \times 2n}$ for the RWCV maneuvering model and $\mathbf{C}_k = [\mathbf{I}_n \ 0 \ 0] \in \mathbb{R}^{n \times 3n}$ for the RWCA maneuvering model. Now one can apply the standard KF predict and update cycles [24] [17]

$$\text{Predict cycle:} \quad \begin{cases} \widehat{\boldsymbol{\Sigma}}_{k+1} = \mathbf{A}_k \widehat{\boldsymbol{\Sigma}}_k^+ \mathbf{A}_k^T + \mathbf{L}_k \mathbf{Q}_k \mathbf{L}_k^T, \\ \widehat{\mathbf{x}}_{k+1} = \mathbf{A}_k \widehat{\mathbf{x}}_k^+, \end{cases} \tag{2.149}$$

$$\text{Update cycle:} \quad \begin{cases} \widehat{\boldsymbol{\Sigma}}_k^+ = \widehat{\boldsymbol{\Sigma}}_k - \widehat{\boldsymbol{\Sigma}}_k \mathbf{C}_k^T (\mathbf{C}_k \widehat{\boldsymbol{\Sigma}}_k \mathbf{C}_k^T + \mathbf{S}_k)^{-1} \mathbf{C}_k \widehat{\boldsymbol{\Sigma}}_k, \\ \mathbf{K}_k = \widehat{\boldsymbol{\Sigma}}_k^+ \mathbf{C}_k^T \mathbf{S}_k^{-1}, \\ \widehat{\mathbf{x}}_k^+ = \mathbf{K}_k (\mathbf{y}_k - \mathbf{C}_k \widehat{\mathbf{x}}_k), \end{cases} \tag{2.150}$$

initialized at $\widehat{\mathbf{x}}_0 = \boldsymbol{\mu}_0$ and $\widehat{\boldsymbol{\Sigma}}_0 = \boldsymbol{\Sigma}_0$.

2.6.3 EKF with Range-Only measurements

The major drawback of the previous simple KF plus trilateration approach is that a minimum of $m_k = n+1$ range measurements are needed at every iteration. If at some iteration there are only $m_k < n+1$ measurements available, then the trilateration problem does not have a unique solution and a position fix can not be computed. This means that one has to *throw away* those m_k range measurements and wait until some future iteration where there are enough number of measurements. This is quite regretable because valuable information is not being used. To overcome this problem, this section describes how to use the nonlinear range measurements directly as an input for an Extended Kalman Filter which do not require to compute a trilateration fix.

Suppose one considers a general maneuvering model $f : \mathbb{R}^N \times \mathbb{R}^q \times \mathbb{N} \to \mathbb{R}^N$

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \boldsymbol{\xi}_k, k), \tag{2.151}$$

where $\mathbf{x}_k \in \mathbb{R}^N$, $\boldsymbol{\xi}_k \in \mathbb{R}^q$, and $k \in \mathbb{N}$. Moreover let the true vehicle state and disturbance satisfy

$$\mathrm{E}\{\mathbf{x}_0\} = \boldsymbol{\mu}_0 \in \mathbb{R}^N, \tag{2.152}$$

$$\mathrm{E}\left\{(\mathbf{x}_0 - \boldsymbol{\mu}_0)(\mathbf{x}_0 - \boldsymbol{\mu}_0)^T\right\} = \boldsymbol{\Sigma}_0 \in \mathbb{R}^{N\times N}, \tag{2.153}$$

$$\mathrm{E}\{\boldsymbol{\xi}_k\} = 0, \tag{2.154}$$

$$\mathrm{E}\left\{\boldsymbol{\xi}_k\boldsymbol{\xi}_s^T\right\} = 0 \ \forall k \neq s, \tag{2.155}$$

$$\mathrm{E}\left\{\boldsymbol{\xi}_k\boldsymbol{\xi}_k^T\right\} = \mathbf{Q}_k \in \mathbb{R}^{q\times q}, \tag{2.156}$$

where $\boldsymbol{\Sigma}_0$ and $\mathbf{Q}_k$ are positive definite matrices.

Now suppose that at discrete time k, there are m_k range measurements available $\bar{\mathbf{r}}_k = h_k(\mathbf{x}_k, \mathbf{w}_k) = \mathbf{r}_k + \mathbf{w}_k$ where $\bar{\mathbf{r}}_k, \mathbf{r}_k, \mathbf{w}_k \in \mathbb{R}^{m_k}$ and $h_k : \mathbb{R}^N \times \mathbb{R}^{m_k} \to \mathbb{R}^{m_k}$ is regarded as a mapping that computes the ranges between the vehicle and the landmarks available at discrete time k, given the full vehicle maneuvering state $\mathbf{x}_k \in \mathbb{R}^N$ (which contains the vehicle position $\mathbf{p}_k \in \mathbb{R}^n$). Moreover,the measurement error covariance is defined as $\mathrm{E}\{\mathbf{w}_k\mathbf{w}_k\} = \mathbf{R}_k \in \mathbb{R}^{m_k\times m_k}$.

According to standard EKF theory [17] we need to compute the Jacobians

$$\mathbf{A}_k(\cdot) = \frac{\partial f(\mathbf{x}, \boldsymbol{\xi})}{\partial \mathbf{x}} : \mathbb{R}^N \to \mathbb{R}^{N\times N}, \tag{2.157}$$

$$\mathbf{L}_k(\cdot) = \frac{\partial f(\mathbf{x}, \boldsymbol{\xi})}{\partial \boldsymbol{\xi}} : \mathbb{R}^q \to \mathbb{R}^{q\times N}, \tag{2.158}$$

$$\mathbf{C}_k(\cdot) = \frac{\partial h_k(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}} : \mathbb{R}^N \to \mathbb{R}^{m_k\times N}, \tag{2.159}$$

$$\mathbf{D}_k(\cdot) = \frac{\partial h_k(\mathbf{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} : \mathbb{R}^{m_k} \to \mathbb{R}^{m_k\times m_k}, \tag{2.160}$$

which assume different forms for the different models as follows

$$\text{RWCV:} \quad \begin{cases} \mathbf{A}_k(\mathbf{x}) = \mathbf{A}_k = \begin{bmatrix} \mathbf{I}_n & h\mathbf{I}_n \\ 0 & \mathbf{I}_n \end{bmatrix} \in \mathbb{R}^{2n\times 2n} \\ \mathbf{C}_k(\mathbf{x}) = \left[\delta(\mathbf{r}_k(\mathbf{p}))^{-1}(\mathbf{p}\mathbf{1}_{m_k}^T - \mathbf{P}_k)^T \ 0\right] \in \mathbb{R}^{m_k\times N} \\ \mathbf{x} = [\mathbf{p}^T \ \mathbf{v}^T]^T \in \mathbb{R}^{2n} \\ \mathbf{L}_k = \mathbf{I}_N, \ \mathbf{D}_k = \mathbf{I}_{m_k} \end{cases} \tag{2.161}$$

$$\text{RWCA:} \quad \begin{cases} \mathbf{A}_k(\mathbf{x}) = \mathbf{A}_k = \begin{bmatrix} \mathbf{I}_n & h\mathbf{I}_n & 0 \\ 0 & \mathbf{I}_n & h\mathbf{I}_n \\ 0 & 0 & \mathbf{I}_n \end{bmatrix} \in \mathbb{R}^{3n\times 3n} \\ \mathbf{C}_k(\mathbf{x}) = \left[\delta(\mathbf{r}_k(\mathbf{p}))^{-1}(\mathbf{p}\mathbf{1}_{m_k}^T - \mathbf{P}_k)^T \ 0 \ 0\right] \in \mathbb{R}^{m_k\times N} \\ \mathbf{x} = [\mathbf{p}^T \ \mathbf{v}^T \ \mathbf{a}^T]^T \in \mathbb{R}^{3n} \\ \mathbf{L}_k = \mathbf{I}_N, \ \mathbf{D}_k = \mathbf{I}_{m_k} \end{cases} \tag{2.162}$$

$$\text{RWCTR:} \quad \begin{cases} \mathbf{A}_k(\mathbf{x}) = \begin{bmatrix} 1 & 0 & h\cos(\psi) & -hV\sin(\psi) & 0 \\ 0 & 1 & h\sin(\psi) & hV_k\cos(\psi) & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1\ h \\ 0 & 0 & 0 & 0 & 0\ 1 \end{bmatrix} \in \mathbb{R}^{5\times 5} \\ \mathbf{C}_k(\mathbf{x}) = \left[\delta(\mathbf{r}_k(\mathbf{p}))^{-1}(\mathbf{p}\mathbf{1}_{m_k}^T - \mathbf{P}_k)^T \ 0 \ 0 \ 0\right] \in \mathbb{R}^{m_k\times 5} \\ \mathbf{x} = [\mathbf{p}^T \ V \ \psi \ r]^T \in \mathbb{R}^5 \\ \mathbf{L}_k = \mathbf{I}_5, \ \mathbf{D}_k = \mathbf{I}_{m_k} \end{cases} \tag{2.163}$$

We can now apply the standard EKF predict and update cycles [24] [17]

$$\text{Predict cycle:} \quad \begin{cases} \widehat{\mathbf{\Sigma}}_{k+1} = \mathbf{A}_k(\widehat{\mathbf{x}}_k^+)\widehat{\mathbf{\Sigma}}_k^+\mathbf{A}_k(\widehat{\mathbf{x}}_k^+)^T + \mathbf{L}_k\mathbf{Q}_k\mathbf{L}_k^T, \\ \widehat{\mathbf{x}}_{k+1} = f(\widehat{\mathbf{x}}_k^+, 0), \end{cases} \tag{2.164}$$

$$\text{Update cycle:} \quad \begin{cases} \widehat{\mathbf{\Sigma}}_k^+ = \widehat{\mathbf{\Sigma}}_k - \widehat{\mathbf{\Sigma}}_k\mathbf{C}_k(\widehat{\mathbf{x}}_k)^T\Big(\mathbf{C}_k(\widehat{\mathbf{x}}_k)\widehat{\mathbf{\Sigma}}_k\mathbf{C}_k(\widehat{\mathbf{x}}_k)^T + \mathbf{R}_k\Big)^{-1}\mathbf{C}_k(\widehat{\mathbf{x}}_k)\widehat{\mathbf{\Sigma}}_k, \\ \mathbf{K}_k = \widehat{\mathbf{\Sigma}}_k^+\mathbf{C}_k(\widehat{\mathbf{x}}_k)^T\mathbf{R}_k^{-1}, \\ \widehat{\mathbf{x}}_k^+ = \mathbf{K}_k(\bar{\mathbf{r}}_k - h_k(\widehat{\mathbf{x}}_k, 0)), \end{cases} \tag{2.165}$$

initialized at $\widehat{\mathbf{x}}_0 = \boldsymbol{\mu}_0$ and $\widehat{\mathbf{\Sigma}}_0 = \mathbf{\Sigma}_0$.

2.6.4 Nonlinear observer with continuous squared range measurements

In this section we follow a different approach to the range-only target tracking problem. We will consider a simple continuous time vehicle kinematic model and continuous squared range measurements. Under the assumption that there are enough landmarks with a good relative geometry, a continuous time nonlinear observer will be presented with some convergence warranties.

Consider a simple continuous time linear vehicle kinematic model without dis-

turbances

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \tag{2.166}$$

with state $\mathbf{x} \in \mathbb{R}^N$ and system matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$. Assume that the position of the vehicle $\mathbf{p} \in \mathbb{R}^n$ is a linear function of the state vector and can be written as $\mathbf{p} = \mathbf{C}\mathbf{x}$, where $\mathbf{C} \in \mathbb{R}^{n\times N}$ is a constant matrix. For instance, in the RWCV model, we have $\mathbf{x} = [\mathbf{p}^T\ \mathbf{v}^T]^T \in \mathbb{R}^{2n}$ and $\mathbf{C} = [\mathbf{I}_n\ 0] \in \mathbb{R}^{n\times 2n}$.

Let $\mathbf{d} = [d_1\ \ldots\ d_m]^T \in \mathbb{R}^m$ denote the continuous squared noise free range measurements between the vehicle and the landmarks, that is, $d_i = \|\mathbf{p}-\mathbf{p}_i\|^2$, which can be written as

$$\mathbf{d} = \delta(\mathbf{P}^T\mathbf{P}) + \|\mathbf{p}\|^2\mathbf{1}_m - 2\mathbf{P}^T\mathbf{p}. \tag{2.167}$$

Proposition 2.6.1. *Consider the nonlinear observer*

$$\dot{\widehat{\mathbf{x}}} = \mathbf{A}\widehat{\mathbf{x}} - \frac{1}{2}\mathbf{K}\mathbf{P}\mathbf{M}(\mathbf{d} - \widehat{\mathbf{d}}), \tag{2.168}$$

where $\widehat{\mathbf{x}} = [\widehat{\mathbf{p}}^T\ \widehat{\mathbf{v}}^T]^T \in \mathbb{R}^{2n}$ *is the vehicle estimated state,* $\mathbf{P}$ *is the matrix of landmark coordinates,* $\mathbf{K} \in \mathbb{R}^{2n\times n}$ *is a constant matrix gain,* $\mathbf{M} = \mathbf{I}_m - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^T$ *is the usual centering operator, and* $\widehat{\mathbf{d}} = \widehat{\mathbf{d}}(\widehat{\mathbf{x}})$ *is the vector of estimated squared ranges*

$$\widehat{\mathbf{d}} = \delta(\mathbf{P}^T\mathbf{P}) + \|\widehat{\mathbf{p}}\|^2\mathbf{1}_m - 2\mathbf{P}^T\widehat{\mathbf{p}}. \tag{2.169}$$

Then, if the matrix $\bar{\mathbf{A}} := \mathbf{A} - \mathbf{K}\mathbf{P}_c\mathbf{P}_c^T\mathbf{C}$ *is stable, for all initial conditions* $\widehat{\mathbf{x}}(0)$ *the solutions of (2.168) satisfy* $lim_{t\to\infty}\widehat{\mathbf{x}}(t) = \mathbf{x}(t)$.

Proof. Define the estimation error $\widetilde{\mathbf{x}} = \mathbf{x}-\widetilde{\mathbf{x}}$. First note that $\mathbf{M}(\mathbf{d}-\widehat{\mathbf{d}}) = -2\mathbf{M}\mathbf{P}^T(\mathbf{p}-\widehat{\mathbf{p}}) = -2\mathbf{P}_c^T\widetilde{\mathbf{p}} = -2\mathbf{P}_c^T\mathbf{C}\widetilde{\mathbf{x}}$. Then

$$\begin{aligned}\dot{\widetilde{\mathbf{x}}} &= \dot{\mathbf{x}} - \dot{\widehat{\mathbf{x}}} = \mathbf{A}\mathbf{x} - \mathbf{A}\widehat{\mathbf{x}} + \mathbf{K}\mathbf{P}\mathbf{M}(\mathbf{d} - \widehat{\mathbf{d}})\\ &= \mathbf{A}\widetilde{\mathbf{x}} - 2\mathbf{K}\mathbf{P}_c\mathbf{P}_c^T\widetilde{\mathbf{p}}\\ &= \mathbf{A}\widetilde{\mathbf{x}} - 2\mathbf{K}\mathbf{P}_c\mathbf{P}_c^T\mathbf{C}\widetilde{\mathbf{x}}\\ &= (\mathbf{A} - 2\mathbf{K}\mathbf{P}_c\mathbf{P}_c^T\mathbf{C})\widetilde{\mathbf{x}}.\end{aligned} \tag{2.170}$$

□

The previous vehicle model did not consider disturbances, so that only was able to capture limited type of trajectories. For instance in the RWCV only straight lines would be correctly modelled. Moreover, no measurement noise was considered in the squared range measurements. In practice, a better vehicle model is that subject to some disturbances the squared range measurements are always corrupted by some

noise. Consider a vehicle model

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}, \tag{2.171}$$

where $\mathbf{b} \in \mathbb{R}^N$ is some deterministic bounded disturbance. Moreover, let

$$\bar{\mathbf{d}} = \mathbf{d} + \boldsymbol{\xi}, \tag{2.172}$$

be the measured squared ranges vector, where $\boldsymbol{\xi} \in \mathbb{R}^m$ is some bounded deterministic disturbance vector.

Proposition 2.6.2. *Consider the nonlinear observer (2.168) together with the vehicle model (2.171) and the noisy squared range measurements (2.172). Suppose that the matrix* $\bar{\mathbf{A}} = \mathbf{A} - \mathbf{K}\mathbf{P}_c\mathbf{P}_c^T\mathbf{C}$ *is stable. Then the system is Input to State Stable with respect to the inputs* $\mathbf{b}$, $\boldsymbol{\xi}$ *and state* $\widetilde{\mathbf{x}}$.

Proof. The estimation error in this case satisfies

$$\dot{\widetilde{\mathbf{x}}} = = (\mathbf{A} - 2\mathbf{K}\mathbf{P}_c\mathbf{P}_c^T\mathbf{C})\widetilde{\mathbf{x}} + \mathbf{M}\boldsymbol{\xi} + \mathbf{b} \tag{2.173}$$

$$= \bar{\mathbf{A}}\widetilde{\mathbf{x}} + \mathbf{M}\boldsymbol{\xi} + \mathbf{b}, \tag{2.174}$$

and

$$\widetilde{\mathbf{x}}(t) = e^{t\bar{\mathbf{A}}}\widetilde{\mathbf{x}}(0) + \int_0^t e^{(t-\tau)\bar{\mathbf{A}}}(\mathbf{M}\boldsymbol{\xi}(\tau) + \mathbf{b}(\tau))d\tau. \tag{2.175}$$

Now, taking norms

$$\begin{aligned}
\|\widetilde{\mathbf{x}}(t)\| &\leq \|e^{t\bar{\mathbf{A}}}\|\,\|\widetilde{\mathbf{x}}(0)\| + \| \int_0^t e^{(t-\tau)\bar{\mathbf{A}}}\big(\mathbf{M}\boldsymbol{\xi}(\tau) + \mathbf{b}(\tau)\big)\|d\tau \\
&\leq ke^{-\lambda t}\|\widetilde{\mathbf{x}}(0)\| + \sup_{0\leq\tau\leq t}\Big(\|\mathbf{M}\boldsymbol{\xi}(\tau)\| + \|\mathbf{b}(\tau)\|\Big)\int_0^t ke^{-\lambda(t-\tau)}d\tau \\
&= ke^{-\lambda t}\|\widetilde{\mathbf{x}}(0)\| + \sup_{0\leq\tau\leq t}\Big(\|\boldsymbol{\xi}(\tau)\| + \|\mathbf{b}(\tau)\|\Big)\frac{k}{\lambda}(1 - e^{-\lambda t}) \\
&\leq ke^{-\lambda t}\|\widetilde{\mathbf{x}}(0)\| + \frac{k}{\lambda}\sup_{0\leq\tau\leq t}\Big(\|\boldsymbol{\xi}(\tau)\| + \|\mathbf{b}(\tau)\|\Big),
\end{aligned} \tag{2.176}$$

where we used that since $\bar{\mathbf{A}}$ is Hurwitz, we have $\|e^{t\bar{\mathbf{A}}}\| \leq ke^{-\lambda t}$ for some $\lambda, k > 0$ [96, p.174], [96, p.221], [96, p.202]. □

Proposition 2.6.3. *Consider the vehicle model (2.166) with state* $\mathbf{x} = [\mathbf{p}^T\ \mathbf{v}^T]^T \in \mathbb{R}^{2n}$, *and system matrix* $\mathbf{A} = \left[\begin{smallmatrix} 0 & \mathbf{I}_n \\ 0 & 0 \end{smallmatrix}\right] \in \mathbb{R}^{2n\times 2n}$ *corresponding to a simple constant velocity model. Suppose that* $\mathbf{P}_c$ *is full column rank and that the matrix gain is*

chosen as

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_1 \\ \mathbf{K}_2 \end{bmatrix} = \begin{bmatrix} \gamma_1 (\mathbf{P}_c \mathbf{P}_c^T)^{-1} \\ \gamma_2 (\mathbf{P}_c \mathbf{P}_c^T)^{-1} \end{bmatrix} \in \mathbb{R}^{2n \times n}, \tag{2.177}$$

for some constants $\gamma_1, \gamma_2 > 0$. *Then the error system matrix* $\bar{\mathbf{A}} = \mathbf{A} - \mathbf{K}\mathbf{P}_c\mathbf{P}_c^T\mathbf{C}$ *has the form*

$$\bar{\mathbf{A}} = \begin{bmatrix} -\gamma_1 & 1 \\ -\gamma_2 & 0 \end{bmatrix} \otimes \mathbf{I}_n, \tag{2.178}$$

which eigenvalues have all negative real part and have the form

$$\lambda_i = -\frac{\gamma_1}{2} \pm \frac{1}{2}\sqrt{\gamma_1^2 - 4\gamma_2}. \tag{2.179}$$

Proof. Follows from direct substitution. □

For instance taking $\gamma_2 = \frac{1}{4}\gamma_1^2$ we obtain that all the eigenvalues of are equal to $\lambda_1 = \ldots = \lambda_{2n} = -\frac{1}{2}\gamma_1$.

Figures 2.24-2.27 show simulation results of the derived nonlinear observer. Three landmarks were located with a relatively good relative geometry, and a baseline of approximately 500m. A lawn mower type of trajectory was simulated with the vehicle moving at a nominal speed of 5m/s. Continuous range measurements corrupted with a Gaussian disturbance of standard deviation $\sigma = 1$m were generated and then squared in order to run the observer. Figure 2.24 shows the simulation setup, together with the actual and estimated vehicle trajectories. The initial estimated vehicle position was located roughly 200m away from the true vehicle position. The nonlinear observer differential equations were implemented and solved using MATLAB SIMULINK. The observer gains were set according to (2.177) with $\gamma_1 = 1$ and $\gamma_2 = \frac{1}{4}\gamma_1^2 = \frac{1}{4}$. Figures 2.25-2.26 show the actual and estimated vehicle position and velocity as a function of time. Note how using the derived nonlinear observer, the velocity of the vehicle can be estimated reasonably well event though there are only position measurements. Figure 2.27 shows the vehicle position and velocity estimation errors.

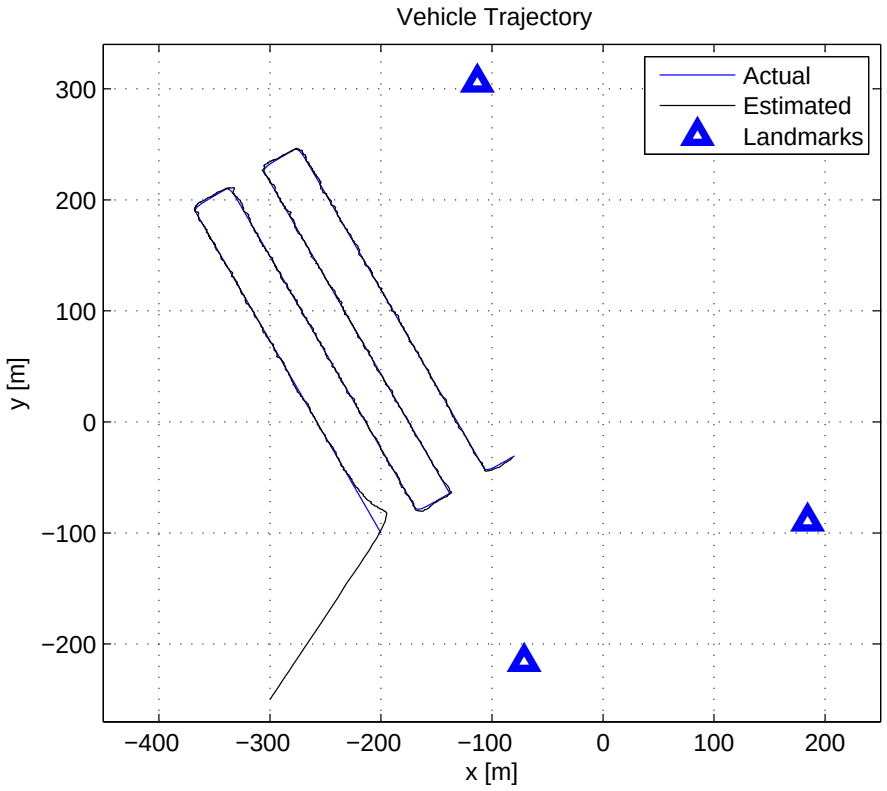


Figure 2.24: Nonlinear observer simulation setup.

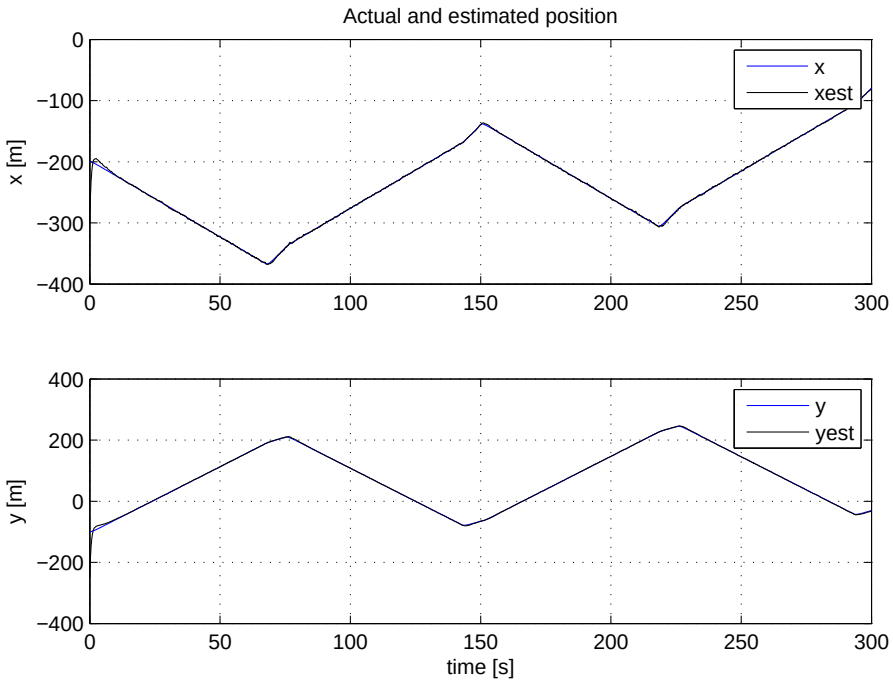


Figure 2.25: Actual and estimated vehicle position.

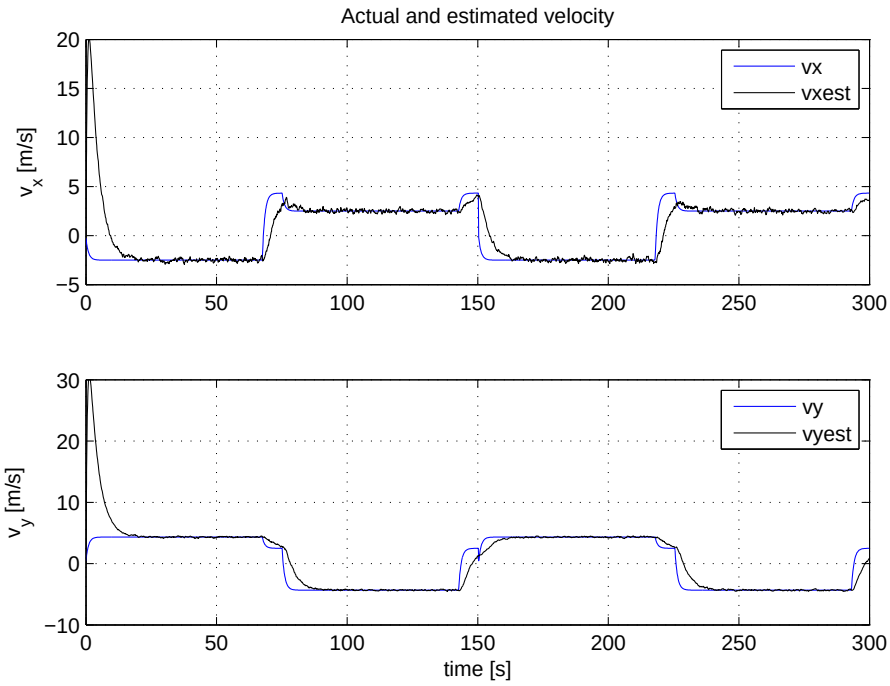


Figure 2.26: Actual and estimated vehicle velocity.

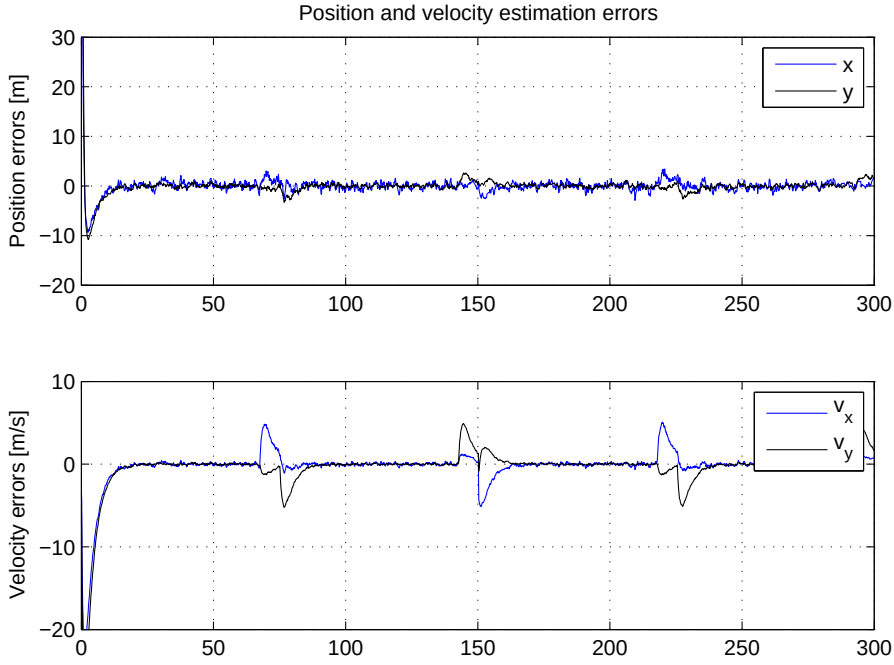


Figure 2.27: Position and velocity estimation errors.

2.7 Summary

This chapter has addressed different solutions to the range only localization and target tracking problems which arise frequently in underwater acoustic positioning systems. Despite its apparent simplicity, the range-only localization problem is not completely understood and is still subject of numerous research.

Several closed form Least Squares algorithms and Maximum Likelihood iterative algorithms were presented. These algorithms are relatively simple and convenient to be implemented in real time on autonomous robotic vehicles. A detailed analysis of the associated errors and a thorough comparison with the Cramér Rao Bound (CRB) for the problem at hand has been done. In order to perform this analysis a Gaussian assumption was done on the nature of the range measurement errors. This is a useful approximation as it allows to obtain neat analytic expressions and simple to interpret results. However, some criticism might be raised against the Gaussian assumption. Range measurements, when obtained through the times of arrival of acoustic signals, might include disturbances due to a large number of physical phenomena that need to be considered. In the next chapter we will analyze in detail the nature of these measurement errors for a real underwater positioning system and will discuss some important practical issues that need to be considered when implementing these algorithms.

The chapter has also addressed some dynamic and filtering algorithms to solve the range-only target tracking problem. A simple Kalman filter solution using position fixes from a closed form trilateration algorithm and an extended Kalman filter using the direct range measurements has been considered. Moreover, a continuous time nonlinear observer has also been analyzed. In order to implement the previously mentioned filtering algorithms in real-time it is necessary to address several practical and theoretical issues. One of the most important is the requirement for a robust outlier identification and rejection algorithm. The next chapter addresses some of these issues and describes a real implementation on a real underwater acoustic positioning system.

Chapter 3

Application to a GIB underwater positioning system

3.1 Introduction

There is a long history of techniques and devices used by man in order to navigate underwater. Acoustic positioning systems have been widely used, first in military applications, and later in industry and scientific applications [125], [180], [112]. Still, nowadays most of the available acoustic positioning systems are not very scalable, operate over relatively reduced areas, and have high costs of operation. Recent advances in GPS technologies have made this technology available to the great public. GPS has many advantages, it scales very well, supporting unlimited number of simultaneous receivers, has a great area of operation, covering almost the entire Earth, and it is relatively cheap. The miniaturization of GPS receivers, their popularization and large scale production, has enabled a fast growing number of applications and technologies that were unthinkable in the past. GPS receivers are nowadays found in cars, mobile phones, personal computers, and watches.

The scenario is far more *prehistoric* in the underwater world [180], [181]. For many years, researchers and entrepreneurs, have worked in order to replicate the GPS concept in underwater applications. However, there are technical difficulties that have prevented such system to appear. GPS signals do not penetrate with sufficient energy below the sea surface. Sound waves propagate well in the water but with much higher wavelengths and power requirements than their electromagnetic equivalents. Acoustic communications, and acoustic modems, are several orders of magnitude behind in terms of achievable bandwidths, when compared with electromagnetic based communications.

In [193], a system that replicates the GPS concept underwater was pattented. The system consisted of a set of surface buoys with GPS receivers that broadcasted GPS-like signals underwater using acoustics. Such a concept has not been yet materialized, due to the difficulties inherent to underwater acoustic communications.

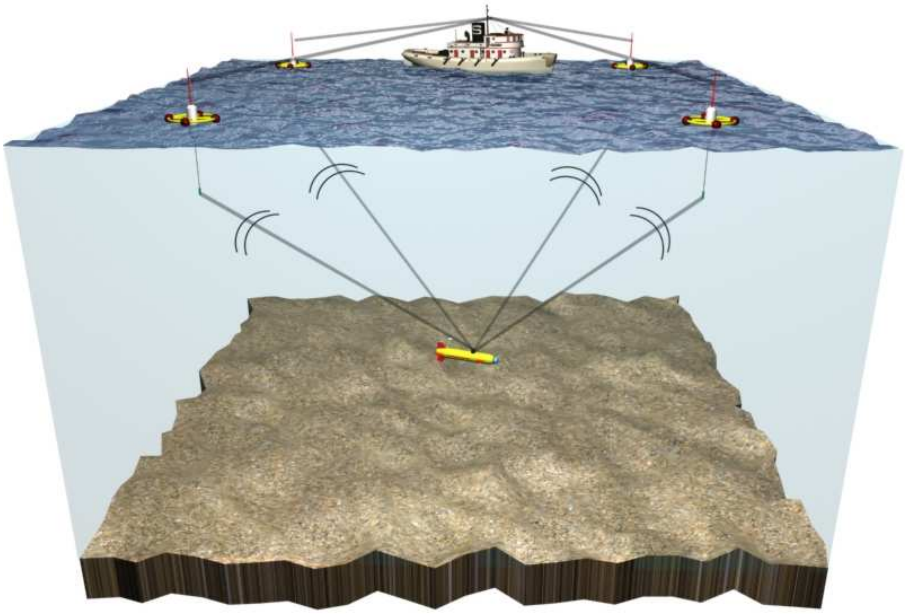


Figure 3.1: GIB

Other systems based on buoys equipped with GPS receivers have appeared [169] [65], and the RATS VEMCO system, specialized to the tracking of tagged fishes. The GIB (GPS Intelligent Buoys) system is commercialized by ACSA [169]. It consists of a set of surface buoys equipped with GPS receivers and submerged hydrophones that compute the times of arrival of the acoustic signals emitted by a synchronized pinger. The times of arrival are then sent through a radio link to a control station, typically on-board a support vessel, where the position of the pinger is computed and displayed, see Figure 3.1. As opposed to the *ideal* underwater GPS concept, the position of the pinger is only known at the control unit, and thus can not be directly used for underwater vehicle navigation. In order to do this, one could use an acoustic modem to transmit the computed position back to the underwater vehicle.

The GIB system presents some advantages when compared to other classic acoustic positioning systems such as Long Baselione (LBL) or Ultra-Short Baseline (USBL). It is significantly easier to operate than a LBL system, without the need to deploy, calibrate and recover a set of transponders. It does not require calibration in order to determine the beacons position (as in LBL) since the position is given by GPS receivers on each buoy. This features turn out valuable when operating at different areas and moving the system from place to place. If, on the other hand, one has to operate in the same area for a long time, an LBL system has some advantages. However, note that LBL can be used directly as a vehicle navigation aiding system, whereas GIB is only suited to tracking an underwater target from the surface. When compared to an USBL system, GIB is not as simple and compact,

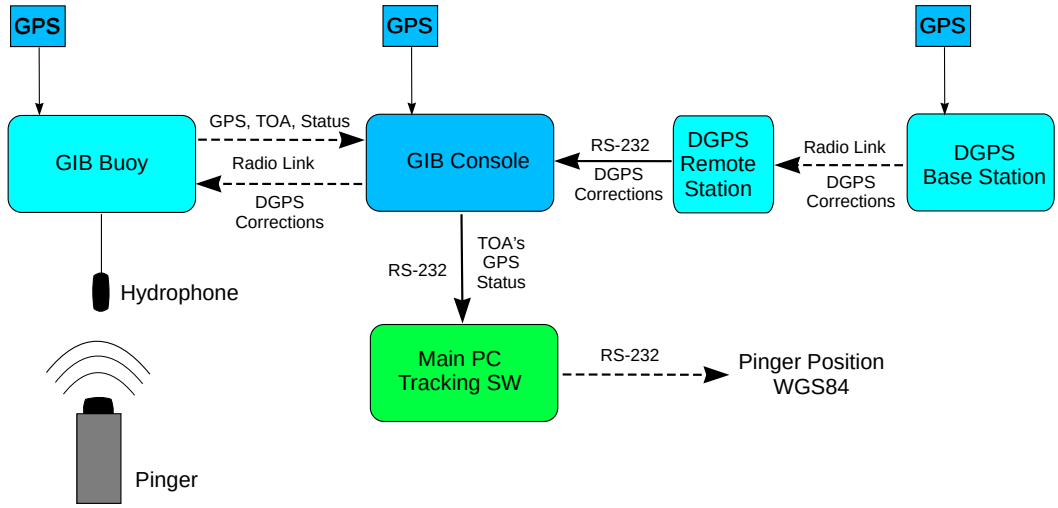


Figure 3.2: GIB System Architecture

and it requires significantly more time to deploy (possibly moore) and recover the buoys. The USBL system, instead, is quite compact, contains all of its receiving elements in a single transponder head, though it requires some careful system installation calibration not required by GIB. A fundamental difference is the sizes of the baselines, which are of the order of centimeters for an USBL system and in the order of 500m-2000m for a GIB system. By having much bigger baselines, the GIB is able to provide in principle better absolute positioning accuracies.

3.2 GIB system architecture

The GIB system architecture is depicted in Figure 3.2. The GIB system is conceived to track the position of an underwater acoustic pinger from the surface. The pinger can be attached to different sort of vehicles such as ROVs, AUVs, divers, manned submersibles, towfish. The pinger sends consecutive acoustic pulses at a typical rate of 1s (although higher rates of 2s or 4s can be used) at times synchronized with GPS. The times of arrival of these acoustic pulses are detected by submerged hydrophones at each of the surface buoys (typically 4) and sent through a radio link to a control unit. The control unit receives the times of arrival of all the buoys plus their GPS positions and some other status information such as battery level, acoustic channel quality, total number of acoustic detections. This in formation is sent through a serial link to a main PC running a dedicated software where the pinger position is computed and displayed.

Next we will describe all of the system elements with some detail:

Figure 3.3: GIB pinger at the Lab.

3.2.1 Pinger

The GIB pinger emits acoustic pulses at a nominal frequency of 12kHz. It weights approximately 2 kg and carries electronic circuitry with a rechargeable battery giving it an autonomy of about 15h. The pulses are of a sinusoidal nature and of a duration of approximately 10ms. The pinger can be configured to send acoustic pulses at periods of 1s, 2s and 4s, depending on the user needs. Slower emission rates increase system range up to a few kilometers and battery life, whereas high emission rates are typically used for a maximum range (maximum distance between pinger and any of the buoys) of about 1km and reduced battery life. The pinger can then be classified according to [180] as a *free-running pinger*, i.e., there is no interrogation cycle, neither any communication from the buoys to the pinger using the acoustic link. The pinger is equipped with a high precision crystal oscillator, which is calibrated with GPS prior system operation.

This is a very important feature the GIB system. The GPS clock synchronization allows to use direct one way travel time of the acoustic signals in order to extract ranging information. Since all the buoys are equipped with a GPS receiver, they are provided with the same time reference as the pinger (neglecting clock drifts). It is then not necessary to resort to time difference of arrival algorithms, or to estimate the unknown clock bias as is the case for instance in GPS systems. The quality of the oscillators in the pinger are of utmost importance if one wishes to keep a good positioning accuracy over long periods of time. The pinger must be calibrated, by connecting it with a cable to the GIB console which has a GPS receiver to provide PPS signals. After calibration the pinger can be used in missions that can take as long as 20h. Drifts in the pinger clock will directly translate into ranging errors, to sum up with all other sources of error, so having a precise clock is important. The drift of the pinger was roughly evaluated during experiments at the laboratory. The

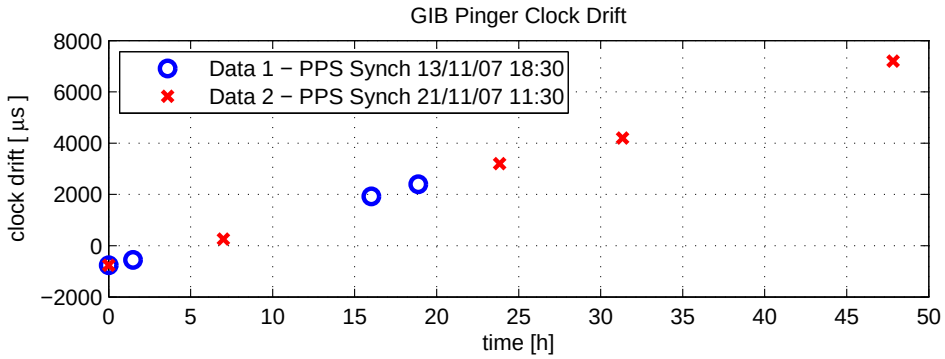


Figure 3.4: Pinger clock drift experimental evaluation in the laboratory.

pinger clock was synchronized with GPS, and readings of the clock bias, using an oscilloscope and a GPS PPS signal were recorder at different intervals. It was found that under these particular lab conditions, the pinger had a typical drift of about 0.046 ppm, which corresponds to approximately 1ms drift every 5 hours, see Figure 3.4. In terms of ranging accuracy, this would correspond to an approximate error of 1.5m after 5 hours of pinger calibration, only because of clock drift. Of course this results are only a rough estimation since clock drift is of a stochastic nature, highly influenced by temperature and other environmental parameters.

The pinger is also equipped with a pressure sensor. This allows to obtain depth measurements which are later sent through the acoustic channel using a basic telemetry. At each emission cycle, two consecutive pulses are sent separated by a time delay that is a linear function of the pinger depth. That is, the delay between the two consecutive pulses is equal to $\Delta t = c_0 + c_1 z$ where $c_0, c_1 > 0$ are two user configurable constants and z is the pinger depth. Typical values of these parameters are $c_0 = 250$[ms] and $c_1 = 10$[ms/m].

3.2.2 Buoys

The GIB buoys are the most innovative part of the system. Each buoy is equipped with a GPS receiver, a radio modem, and an hydrophone. The buoys weight approximately 15kg and are depicted in Figure 3.5. The hydrophone is connected to the buoy through a long cable which sets the hydrophone at a depth of approximately 10m. The buoys are equipped with a set of batteries that provide an autonomy of about 20h.

The buoys can be moored or free drifting. Figure 3.5 shows a set of buoys with their respective hydrophones and anchor lines.

(a) (b)

Figure 3.5: GIB buoys before deployment (a) and ready for transportation to the operation site (b).

Figure 3.6: D-GPSfixed station. Figures shows the GPS and radio antennas at the top of a building in line of sight to the operations area.

3.2.3 DGPS base station

In order to provide better positioning accuracy, the GIB systems allows the use of Differential GPS corrections. The buoys and console GPS receivers accept D-GPS corrections in the standard RTCM format. The corrections need to be provided through a serial link to the console station which broadcasts those corrections via radio link to all the buoys. In order to obtain the corrections it is possible to use a fixed DGPS station near the site of operation with a fixed GPS antenna and a radio link to the console/ control unit site.

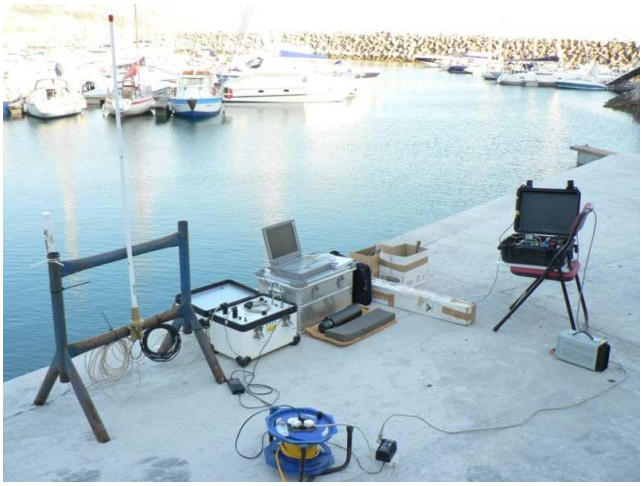


Figure 3.7: GIB control unit and main PC. Preliminary system tests from shore.

3.2.4 Control Unit

The control unit or GIB console is the main component of the system. It receives all the data from the buoys and sends it through a serial link to the main PC running the tracking software. It is equipped with a GPS receiver, in order to monitor the position of the console while on operations, and to provide the PPS signals for pinger clock synchronization. It also receives the DGPS RTCM correction from the DGPS remote station through a serial port. The control unit, the DGPS remote station and the main tracking PC are shown in Figure 3.7. The console is also equipped with a lead acid battery that provides autonomy for some hours.

The main PC runs the tracking software and receives all the buoy information through a serial port. The software computes the position of the pinger based on the times of arrival and the GPS positions of the buoys. The computed pinger position is displayed and stored for further analysis in the PC screen.

3.3 Ranges from Time Of Arrival measurements

In order to extract the distances between pinger and buoys from the times of arrival some considerations need to be made. We will next discuss some important practical issues that are found when dealing with experimental time of arrival data.

In order to continue we have first to understand how are the times of arrival measurements obtained. Every second, the GIB pinger emits two acoustic pulses. Each of the pulses have a duration of approximately 10ms. The first pulse is sent at the start of every second as given by the high precision crystal oscillator that was synchronized with GPS before operation. The second pulse is sent after a delay that is a linear function of pinger depth. Those acoustic signals propagate through the water almost omnidirectionally. At each emission cycle the buoys analyze the

signals coming from the hydrophones and store a set of possible acoustic detections. A detection is considered basically when a capacitor threshold on the hydrophone transducer is achieved. As will be discussed later, those acoustic detections can be of very different natures: direct paths, multipaths, outliers, environmental noise,... Typically, in a noise friendly environment, the first detection corresponds to the direct path of the first pinger pulse and the following detections correspond to the second pulse, and multipaths. Among all the acoustic detections found locally at each the buoy, only three are selected at each instant of time to be broadcasted via radio to the control unit. Figure 3.8 illustrates the three available times of arrival from one GIB buoy as received by the control unit during a period of 800s.

In an ideal scenario the process of computing pinger depth and range would be quite simple. The first TOA would correspond to the direct acoustic path between the pinger and buoy. Assuming a constant sound speed medium, this path would be a straight line between pinger and buoy. Multiplying the TOA by the value of the sound speed will give us the direct range in meters. Computing the difference between the first and second TOAs and using a simple linear relation will give us the pinger depth. In practice, there are some important issues that make the whole process a little more complicated. In the first place, the existence of outliers and multipaths, makes it difficult some times to determine if a given TOA corresponds to the direct path (first pinger pulse), to depth information (second pinger pulse), or whether it is an outlier. In second place, the constant sound speed model is far from real. The sound velocity profile depends on a number of factors including temperature, depth, and salinity. The sound speed variation is responsible for an acoustic ray bending, according to *Snell's law*, the severity of which can turn the simplistic straight line ray model unusable.

3.3.1 Experimental TOA data

We will next present and discuss a series of experimental times of arrival data obtained during sea trials. We will show examples of outliers, environmental noise interferences, multipaths and other characteristic effects that should be taken into account.

Figure 3.9 shows an example of Times of Arrival data with good characteristics. There are almost no outliers, and the first TOA corresponds to the first pinger pulse in the majority of cases. Even in this acoustic friendly environment, we still can notice the existence of few isolated outliers between times 200-300s. Figure 3.10 shows times of arrival data in a bad acoustic environment. There are several outliers, hard to differentiate from the actual TOAs of interest. The first TOA does not correspond most of the times with the direct path but with an outlier.

Figure 3.11 shows a typical environmental noise interference problem. During the time when the data was recorded, there was a nearby ship using an echo sounder sonar operating in a similar frequency as the GIB pinger (12kHz). The buoy acoustic

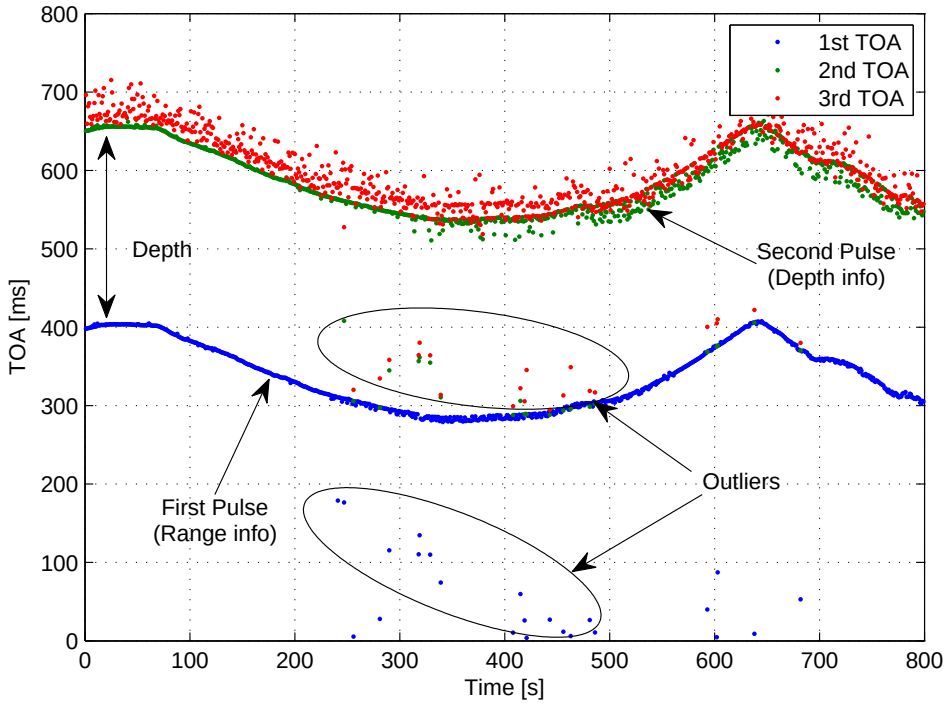


Figure 3.8: Typical Times of Arrival (TOA) data from one of the GIB buoys.

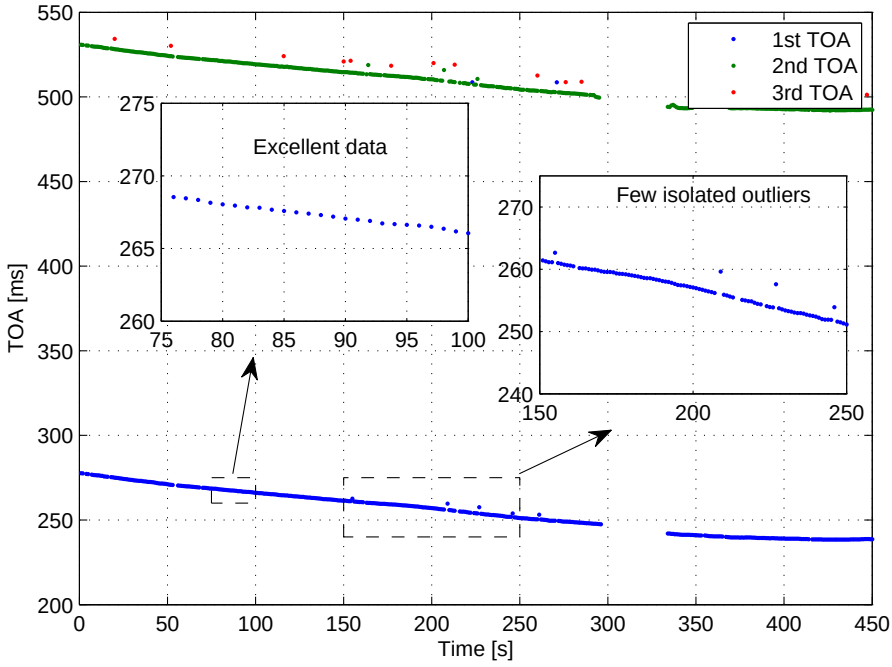


Figure 3.9: Times of arrival with good acoustic conditions.

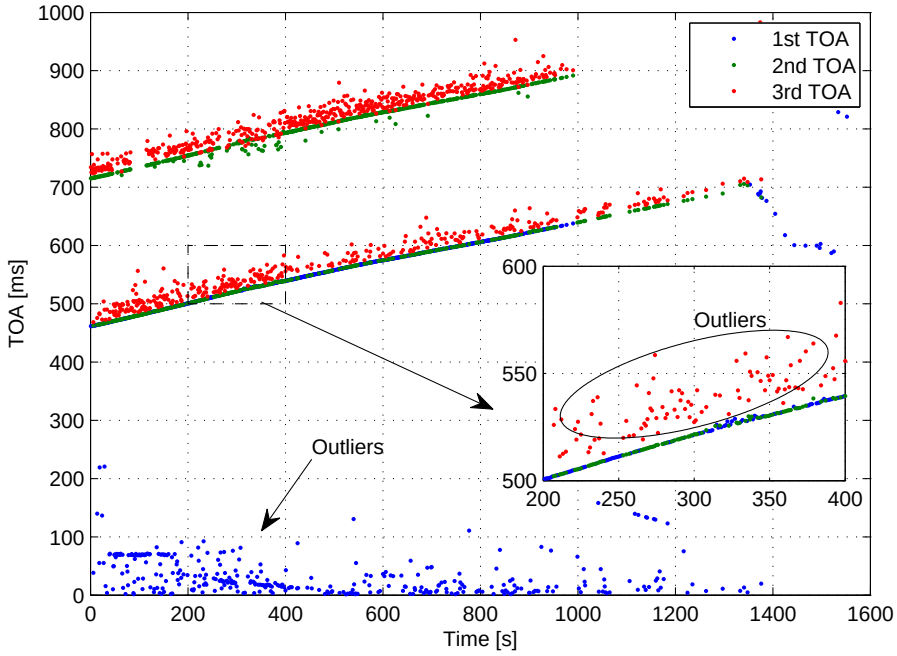


Figure 3.10: Times of arrival with bad acoustic conditions.

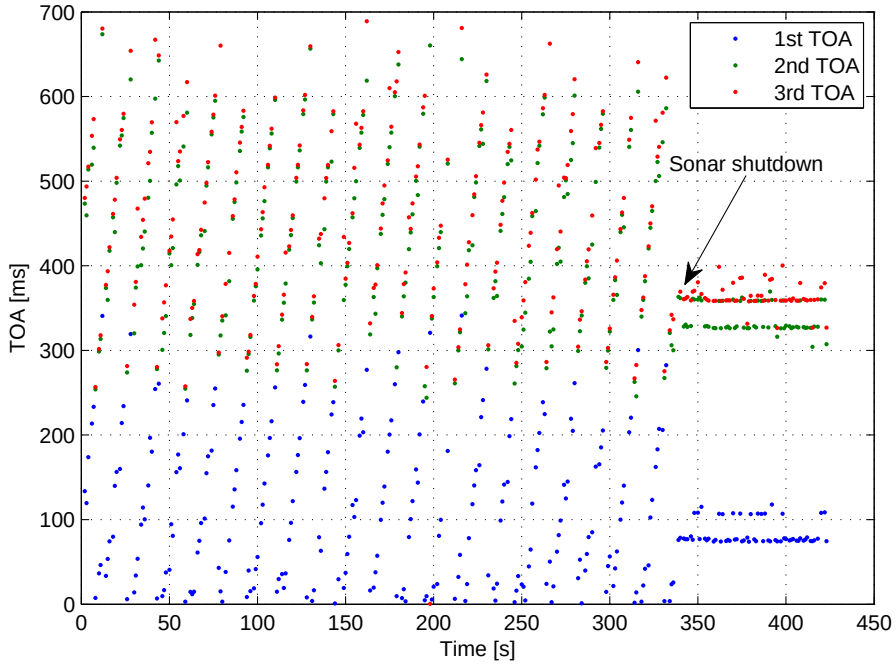


Figure 3.11: Times of arrival. Interferences with a nearby ship using a sonar in a similar working frequency as GIB pinger. At time 340s the interfering sonar system was shut down.

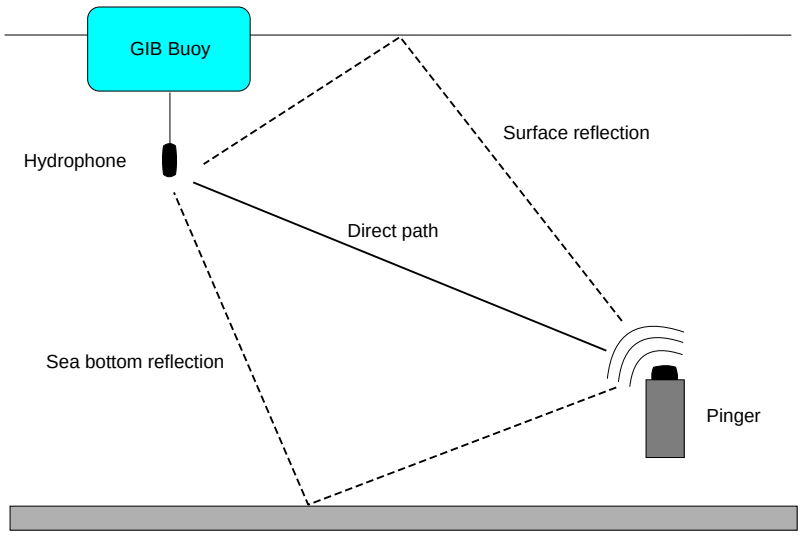


Figure 3.12: Direct path and most common Multipaths.

detection was totally messed up with the interference until the sonar system was shut down at time approximately 340s. Figures 3.13 and 3.14 show another important practical issue when operating acoustic positioning systems: Multipaths. As the sound waves propagate omnidirectionally from the pinger there acoustic rays other that the direct path that can reach the buoys hydrophones with sufficient energy to produce a detection. The most typical are seabed and surface reflections but we can also encounter reflections on vessels hulls and other sea structures as well as some marine life and air bubbles, see Figure 3.12. Multipaths are particularly dangerous if not properly identified because they usually mimic perfectly the direct acoustic paths producing consistent detections. Figure 3.13 shows a clear example of multipaths. When taking a closer look at the TOA data (Right figure) it is possible to see two clear patterns of detects. Both patterns are separated by a delay of approximately 5ms which corresponds roughly to a 7.5m distance (assuming 1500m/s sound speed). During the experiments, the pinger was maneuvered at a small distance below the surface (approximately 2m), and it is probable that this is an example of surface multipath detection.

Figure 3.14 shows an even more dramatic example of multipaths. In the previous case (Figure 3.13) both the direct and multipath were detected (although some times not simultaneously). This fact made it easier to eventually detect and isolate the multipaths. However in Figure 3.14 there is a long sequence of time were only the multipath is being detected (between times 400-600s). This is could happen if for some reason the direct path is not detected, for instace, because it is being blocked by some obstacle. In this case the delay between the direct and multipaths is about 15ms which corresponds to roughly 23m in travelled distance. During the experiments, the pinger was maneuvered near the surface using a 15m support ship

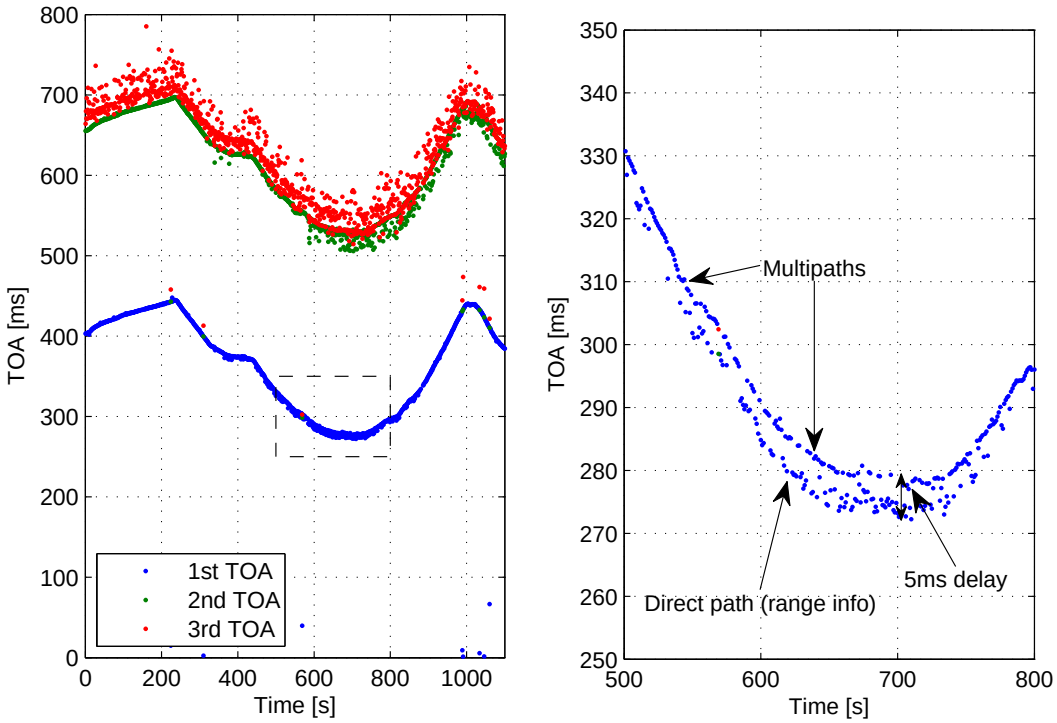


Figure 3.13: Times of arrival with clear multipaths.

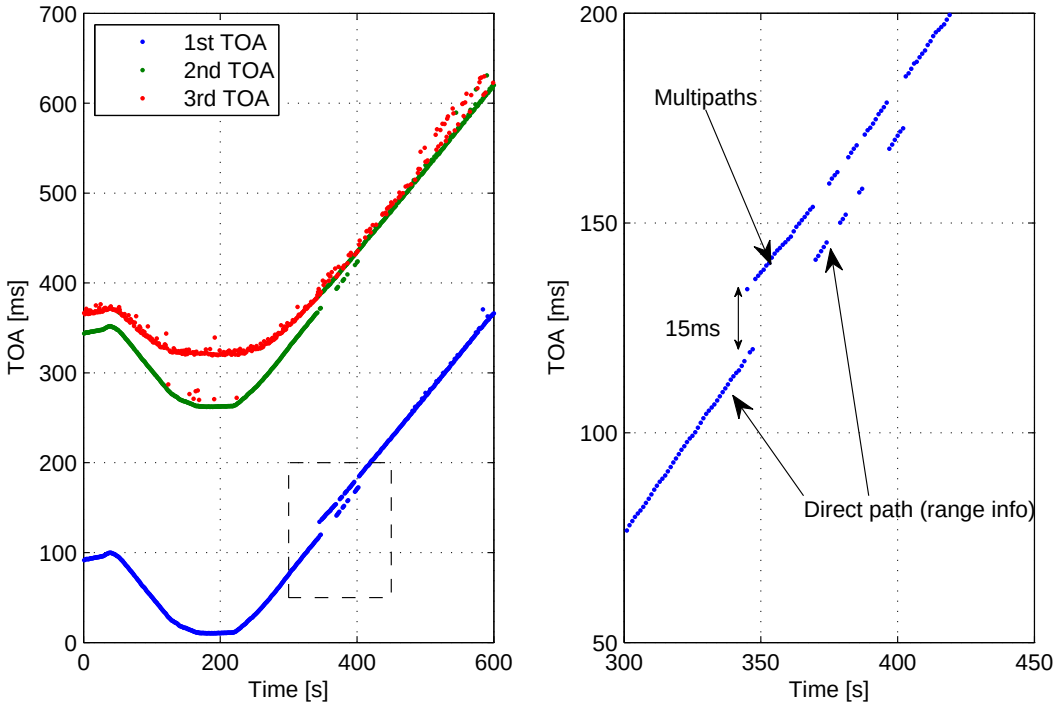


Figure 3.14: Times of arrival with clear multipaths.

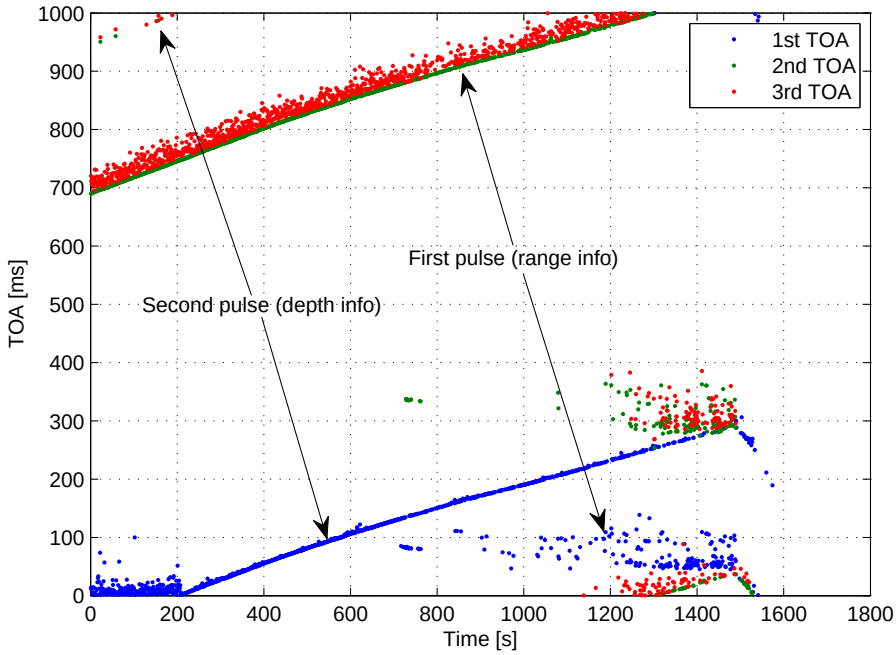


Figure 3.15: Times of arrival with pinger too far from buoy: TOA emission cycle overlap.

and the sea depth at the area was about 30m. It is possible that in this case the direct path was being blocked by the ship hull and the detected signal was the seabed reflection.

As it was previously mentioned, the pinger can be configured to run at periods of 1s, 2s, or 4s. During the experiments the period was set to 1s. Assuming a typical sound speed value of 1500m/s, an acoustic signal can travel 1500m every second. Suppose that the distance between the pinger and one of the buoys is 1650m. Then the first pinger pulse will take 1650/1500=1.1s to cover the distance. Since the buoys are programmed to listen at intervals of 1s, the pulse will only be detected at the next 1 second cycle, with a reading of 0.1s which would correspond to an incorrect 150m distance. This overlapping process is more subject to happen with the second pinger pulse. Suppose that the distance between the pinger and one of the buoys is 1200m. This means the first pulse will take 1200/1500=0.8s to travel this distance. So at time 0.8s the buoy will detect the first pulse. Suppose that the second pulse (corresponding to the depth information) is sent after a delay of $\Delta t = c_0 + c_1 z = 300ms$. Then the second pulse will be received at the buoy after 0.8 + 0.3=1.1s delay which corresponds to the next reception cycle. That is, the second pulse will appear as a detection at 0.1s at the next reception cycle which could be wrongly interpreted as a first pinger pulse at a distance of 150m. This overlapping phenomenon is shown in Figure 3.15. At time approximately 200s, the second pulse overlaps and starts appearing in the next reception cycle. At time approximately

1300s, the same happens with the first pulse. That is, at this point the distance between pinger and buoy reaches 1500m and the first pulse is only detected in the next reception cycle.

All the previous issues with experimental TOA data must be taken into account if one wants to determine the ranges between the pinger and all of the buoys. If by error a multipath, an outlier, or an overlapping signal is mistaken by the first pulse, this can produce big range and positioning errors. We have seen that the simplistic approach of taking the first TOA detect as the first pulse is not robust and can lead to big errors. It is then necessary to implement signal analysis techniques in order to automatically classify and validate the TOA data before it can be used for range computation. The TOA validation and outlier detection processes are of utmost importance and will be further discussed in a later section.

3.3.2 Synchronization issues

As previously mentioned, the pinger is equipped with a high precision crystal oscillator with a drift of approximately 0.02ppm. Previous to each system operation, the pinger clock must be synchronized with GPS time. After synchronization, the system is ready to be used, and for some hours the effect of the pinger clock drift in the positioning accuracy is neglected (a ranging error of 1.5m is expected after 5h of operation due only to clock drift).

The synchronization procedure is quite inconvenient. Before every operation, the pinger needs to be on surface and connected by wire to the GIB console to make the synchronization. Other ranging systems use an interrogation protocol in order to avoid the synchronization. In GPS systems, the receiver clock drift is estimated together with the receiver position. This could also have been done in the GIB system but then 4 buoys would be required all the time to provide a unique position and clock drift estimate. In the current configuration, the system is able to provide unique position estimates with only 3 buoys, which is quite convenient in case one of the four buoys fail.

3.3.3 Sound Speed estimation

In order to extract distance information from time of travel of acoustic signals it is necessary to have an estimate of the speed of propagation of sound in water. The sound speed in water v_s is a function of several parameters, most important being temperature, depth, and salinity [176] with typical values being between 1470 and 1520 meters per second. The so called Mackenzie formula [116] is commonly used to predict its value given a temperature depth and salinity parameters.

Let t_i be the time of arrival of the acoustic signal between pinger and buoy i. Strictly speaking, from the time of arrival one can only measure the length of the

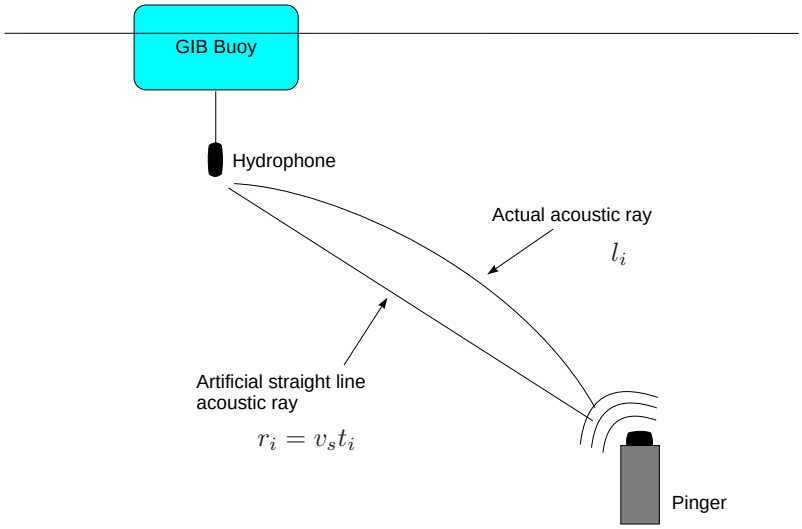


Figure 3.16: Sound speed estimation

path described by the acoustic ray which is not in general a straight line,

$$l_i = \int_0^{t_i} \phi_s(\gamma_i(t))\mathrm{d}t \tag{3.1}$$

where $\gamma_i : [0, t_i] \subset \mathbb{R} \to \mathbb{R}^3$ represents the acoustic ray path coordinates between pinger and buoy i, and $\phi_s : \mathbb{R}^3 \to \mathbb{R}$ is the sound velocity field, that is, given a point coordinates it gives the value of the sound speed at that point, see Figure 3.16.

However the length of the actual acoustic ray is not very useful if one wants to compute the pinger position. Instead one would like to measure the distance between the pinger and the buoy, that is the length of the straight line segment between them. The most simple approach is to consider an artificial constant sound speed v_s that when multiplied by the time of arrival gives a rough estimate of the actual distance, that is

$$r_i = v_s t_i. \tag{3.2}$$

In some cases, the value of v_s is taken as the actual value of sound speed at the pinger depth, or at the surface, or an average between them. However, one is not really interested in the actual value of the sound speed. It is more desirable to determine the constant value v_s that at the end produces better position estimates. We will discuss two simple methodologies for estimating a convenient value of sound speed. If there are $m > n + 1$ range measurements (functioning buoys) it is possible to formulate a linear problem, similar to what was done in the LS and WLS trilateration, where the unknown variables are both the pinger position and the sound speed. If there are not enough measurements a two stage procedure is presented that tries to minimize the estimation error residuals.

Joint sound speed and position LS estimation

The nature of the acoustic rays and the variation of sound speed in water is a quite cumbersome topic on it self. Most simple acoustic positioning algorithms are based on trilateration, that is, on using the distances between a set of landmarks with known location and the unknown vehicle.

Strictly speaking, the available measurements are the times of arrival, which then are usually transformed into ranges by some simple rule. The most general problem would be to estimate both the position and the a constant sound speed value that when multiplied by the times of arrival most resembles the true distances. This can be done in simple closed form in a similar fashion to the LS trilateration.

Suppose there are $m > n+1$ times of arrival measurements t_i with $i \in \{1, \dots, m\}$. Then one can formulate

$$d_i = r_i^2 = \|\mathbf{p} - \mathbf{p}_i\|^2 = v_s^2 t_i^2 \tag{3.3}$$

which in vector form reads

$$\mathbf{1}_m \|\mathbf{p}\|^2 + \delta(\mathbf{P}^T\mathbf{P}) - 2\mathbf{P}^T\mathbf{p} = v_s^2 \mathbf{t}^2 \tag{3.4}$$

Multiplying by matrix $\mathbf{M}$ which has $\mathbf{1}_m$ on its null space and re-ordering we obtain the linear system

$$\begin{bmatrix} 2\mathbf{M}\mathbf{P}^T & \mathbf{M}\mathbf{t}^2 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ v_s^2 \end{bmatrix} = \mathbf{M}\delta(\mathbf{P}^T\mathbf{P}) \tag{3.5}$$

where $\mathbf{t}^2 = [t_1^2 \ \dots \ t_m^2]^T$ is the vector containing the squared times of arrival. The previous system has the form $\mathbf{A}\boldsymbol{\theta} = \mathbf{y}$ which has a solution $\widehat{\boldsymbol{\theta}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}$ when matrix $\mathbf{A}^T\mathbf{A}$ is nonsingular. The $m > n+1$ requirement comes from the fact that if $m \leq n+1$ then this matrix is always singular.

Proposition 3.3.1. *If there are $m \leq n+1$ times of arrival measurements, then $\mathbf{A}^T\mathbf{A}$ is singular.*

Proof. The columns of $2\mathbf{M}\mathbf{P}^T = 2\mathbf{P}_c^T$, as well as $\mathbf{M}\mathbf{t}^2$ belong to the orthogonal complement of $\mathbf{1}_m$, which has dimension $m-1 \leq n$ this means that matrix $\mathbf{A} \in \mathbb{R}^{m \times n+1}$ has at most rank n, and $\mathbf{A}^T\mathbf{A} \in \mathbb{R}^{n+1 \times n+1}$ is singular. □

This roughly means that as opposed to the known sound speed problem, if one needs to determine sound speed an extra observation is needed. Remember that it was shown that it was enough to have $m = n+1$ range measurements (from a non coplanar buoy configuration) to uniquely determine a vehicle position with range observations. if we also want to estimate the value of sound speed we need an extra observation $m > n+1$. In the case of 2 dimensional positioning problem (with

known depth) and 4 buoys, the standard GIB setup, this condition is satisfied since one has $m = 4 > n + 1 = 3$.

For better results one could determine a sound speed value using data from a period of time, where its value is supposed to be constant. Suppose one collects data from N instants of time. That is, consider a set of times of arrival vectors $\{\mathbf{t}_1, \ldots, \mathbf{t}_N\}$ where each $\mathbf{t}_k = [t_{1k}, \ldots\ t_{m_k k}] \in \mathbb{R}^{m_k}$ contains the times of arrival between the pinger and the available buoys at time k. Let $\mathbf{P}_k \in \mathbb{R}^{n \times m_k}$ be the matrix containing the buoy hydrophone coordinates that are available at time k. Now one can formulate a linear problem

$$\begin{bmatrix} 2\mathbf{M}\mathbf{P}_1^T & \ldots & 0 & \mathbf{M}\mathbf{t}_1^2 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & 2\mathbf{M}\mathbf{P}_N^T & \mathbf{M}\mathbf{t}_N^2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \\ v_s^2 \end{bmatrix} = \begin{bmatrix} \mathbf{M}\delta(\mathbf{P}_1^T\mathbf{P}_1) \\ \vdots \\ \mathbf{M}\delta(\mathbf{P}_N^T\mathbf{P}_N) \end{bmatrix} \tag{3.6}$$

which again has the form $\mathbf{A}\boldsymbol{\theta} = \mathbf{y}$.

Joint sound speed and position ML estimation

A more complex solution, which does not have a closed form, would be to consider the minimization of a Maximum Likelihood type the cost function

$$f(\mathbf{x}, v_s) = (\mathbf{r}(\mathbf{x}) - v_s\mathbf{t})^T\mathbf{R}^{-1}(\mathbf{r}(\mathbf{x}) - v_s\mathbf{t}) \tag{3.7}$$

where $\mathbf{r}(\mathbf{x}) = [\|\mathbf{x} - \mathbf{p}_1\|, \ldots,\ \|\mathbf{x} - \mathbf{p}_m\|]^T$, is the vector of distances between pinger position $\mathbf{x}$ and buoys and, $\mathbf{t} = [t_1, \ldots\ , t_m]^T$ contains the times of arrival, and $\mathbf{R}$ is some user defined covariance matrix. A similar cost function can be formulated when considering N instants of time

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_N, v_s) = \sum_{k=1}^{N} (\mathbf{r}(\mathbf{x}_k) - v_s\mathbf{t}_k)^T\mathbf{R}_k^{-1}(\mathbf{r}(\mathbf{x}_k) - v_s\mathbf{t}_k) \tag{3.8}$$

where a different covariance matrix $\mathbf{R}_k$ could be taken for each instant of time. The minimum of such functions, can be determined by resorting to some iterative minimization algorithm such as gradient descent or Newton. In fact, it is not hard to determine the closed form expressions for the gradients and the Hessians of the above cost functions using the gradient and Hessian of the MLR function in (2.82) and (2.83).

The previous joint position and sound speed estimation procedures can only be used whenever there are $m > n+1$ times of arrival available at each instant of time. If this is not the case one can still obtain an approximate sound speed value that is useful for obtaining ranges from times of arrival.

Residual TOA minimization sound speed estimation

Suppose that we already computed a sequence of pinger positions $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ using a preliminary value of sound speed v_s. It is then easy to generate the set of expected times of arrival and compare them to the measured ones. Then one can try to determine which is the value of sound speed that minimizes the difference between the expected and measured times of arrival. For instance, at time k, The estimated time of arrival of buoy i given a sound speed v_s is defined as $\hat{t}_{ik} = \|\mathbf{x}_k - \mathbf{p}_{ik}\|/v_s$. Defining the expected times of arrival vectors $\widehat{\mathbf{t}}_k = [\hat{t}_{1k}, \ldots, \hat{t}_{m_k k}] \in \mathbb{R}^{m_k}$ where m_k denotes the number of available buoys at time k. Now it makes sense to question which is the value of v_s that minimizes the difference between the expected times of arrival $\{\widehat{\mathbf{t}}_1, \ldots, \widehat{\mathbf{t}}_N\}$ and the measured $\{\mathbf{t}_1, \ldots, \mathbf{t}_N\}$. This can be done in a simple closed form if we considering the minimization of the residual $\sum_{k=1}^{N} \|\mathbf{M}(\mathbf{t}_k^2 - \widehat{\mathbf{t}}_k^2)\|^2$ which corresponds to solving the system

$$\underbrace{\begin{bmatrix} \mathbf{M}\mathbf{t}_1^2 \\ \vdots \\ \mathbf{M}\mathbf{t}_N^2 \end{bmatrix}}_{\mathbf{a}} v_s^2 = \underbrace{\begin{bmatrix} \mathbf{M}\delta(\mathbf{P}_1^T\mathbf{P}_1) - 2\mathbf{M}\mathbf{P}_1^T\mathbf{x}_1 \\ \vdots \\ \mathbf{M}\delta(\mathbf{P}_N^T\mathbf{P}_N) - 2\mathbf{M}\mathbf{P}_N^T\mathbf{x}_N \end{bmatrix}}_{\mathbf{b}} \tag{3.9}$$

which has the form $\mathbf{a}v_s^2 = \mathbf{b}$. The value of sound speed that minimizes the residual can then be computed as

$$\widehat{v}_s = \frac{1}{\|\mathbf{a}\|}\sqrt{\mathbf{a}^T\mathbf{b}}. \tag{3.10}$$

With this new value of sound speed one could compute a new sequence of pinger positions $\{\mathbf{x}'_1, \ldots, \mathbf{x}'_N\}$and repeat the previous sound estimation procedure. This can be iterated until a satisfactory sound speed value is found.

3.4 EKF Design

The following section will describe the implementation of an Extended Kalman Filter to process the TOA data coming from a GIB system and produce position estimates of a maneuvering underwater platform. In particular it will address the general problem of estimating the position of an underwater target given a set of range measurements from the target to known buoy locations. Classically, this problem has been solved by resorting to fixed point trilateration techniques, which require that at least three range measurements (in the 2 dimensional setting) be available at the end of each acoustic emission-reception cycle. This is hardly feasible in practice, due to unavoidable communication and sensor failures. It is therefore of interest to develop an estimator structure capable of dealing with the case where the number of range measurements available is time-varying. It will be shown how this problem

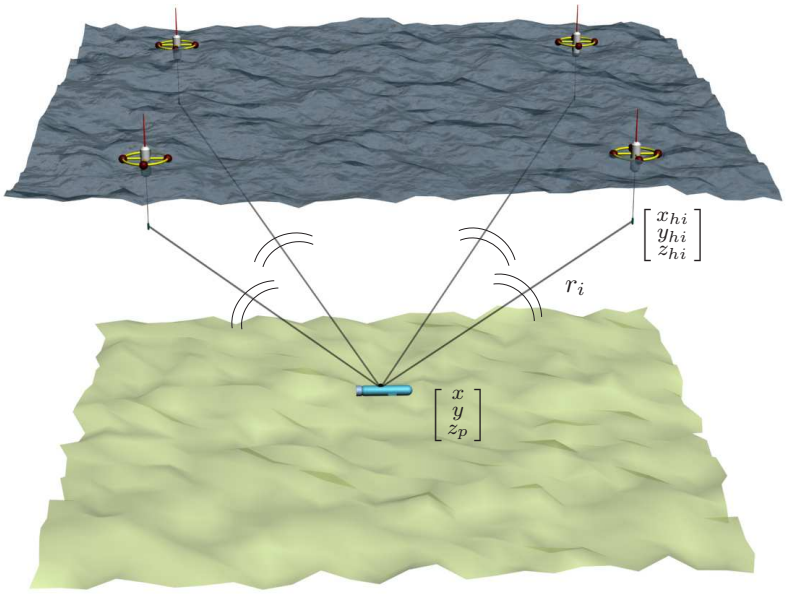


Figure 3.17: The GIB system.

can be tackled in the framework of Extended Kalman Filtering (EKF) whereby four vehicle-to-buoy range measurements drive a filter that relies on a simple kinematic model of the underwater target.

It is important to recall that due to the finite speed of propagation of sound in water, the range measurements are obtained at the buoys with different latencies. To overcome this problem, the section shows how the differently delayed measurements can be merged in an EKF setting by incorporating a *back and forward* fusion approach. Simulation as well as experimental results illustrate the performance of the filter proposed.

Consider an earth fixed reference frame $\{I\}:= \{X_0, Y_0, Z_0\}$ and four (possibly drifting) buoys at the sea surface with submerged hydrophones at positions $[x_{hi}(t)\ y_{hi}(t)\ z_{hi}(t)]^T;\ i = 1, \ldots, 4$ as depicted in Figure 3.17. For simplicity of presentation, we restrict ourselves to the case where the target moves in a plane at a fixed known depth $z_p(t)$. Its position in the earth fixed frame is therefore given by vector $[x(t)\ y(t)\ z_p(t)]^T$. The problem considered in this paper can then be briefly stated as follows: obtain estimates of the two dimensional horizontal target position $\mathbf{p}(t) = [x(t)\ y(t)]^T$ based on information provided by the buoys, which compute the travel time of the acoustic signals emitted periodically by a pinger installed on-board the underwater platform. The solution derived can be easily extended to the case where the target undergoes motions in three dimensional space.

3.4.1 Target model

In what follows we avoid writing explicitly the dynamical equations of the underwater target being tracked and rely on its kinematic equations of motion only. Thus, a general solution for target positioning is obtained that fits different kinds of moving bodies such as AUVs, ROVs, divers, or even marine mammals.

The following notation will be used in the sequel: V is the total velocity of the vehicle in an inertial reference frame $\{I\}$, ψ denotes the angle between vector V and X_0, and r is the derivative of ψ (see Figure 3.18). Note that ψ does not represent the heading of the vehicle, but the angle of the total velocity vector of the vehicle. Only in the absence of sideslip they would be equivalent. Notice that the case where the target moves in three-dimensional space can be cast in the framework adopted in this paper if the depth coordinate can be measured independently. Under these circumstances, V should be re-interpreted as the projection of the total velocity vector on its two first components. Given a continuous-time variable $u(t)$, $u(t_k)$ denotes its values taken at discrete instants of time $t_k = kh; k \in Z_+$, where $h > 0$ denotes the sampling interval. For reasons that will become later it is not convenient to introduce the standard abbreviation $u(k)$ for $u(t_k)$. Simple arguments lead to the Random Walk with Constant Turning Rate (RWCTR) discrete-time kinematic model introduced in (2.140)

$$\begin{cases} x(t_{k+1}) & = x(t_k) + hV(t_k)\cos\psi(t_k) \\ y(t_{k+1}) & = y(t_k) + hV(t_k)\sin\psi(t_k) \\ V(t_{k+1}) & = V(t_k) + \xi_v(t_k) \\ \psi(t_{k+1}) & = \psi(t_k) + hr(t_k) + \xi_\psi(t_k) \\ r(t_{k+1}) & = r(t_k) + \xi_r(t_k), \end{cases} \tag{3.11}$$

where the inclusion of the angular rate equation for $r(t_k)$ captures the fact that the target undergoes motions in ψ that are not measured directly and are thus assumed to be unknown. The process noises $\xi_v(t_k)$, $\xi_\psi(t_k)$, and $\xi_r(t_k)$ are assumed to be stationary, independent, zero-mean, and Gaussian, with constant standard deviations σ_v, σ_ψ, and σ_r respectively. The above model can be written in a compact form as

$$\mathbf{x}(t_{k+1}) = f(\mathbf{x}(t_k), \boldsymbol{\xi}(t_k)) = \bar{\mathbf{A}}(\mathbf{x}(t_k))\mathbf{x}(t_k) + \bar{\mathbf{L}}\boldsymbol{\xi}(t_k) \tag{3.12}$$

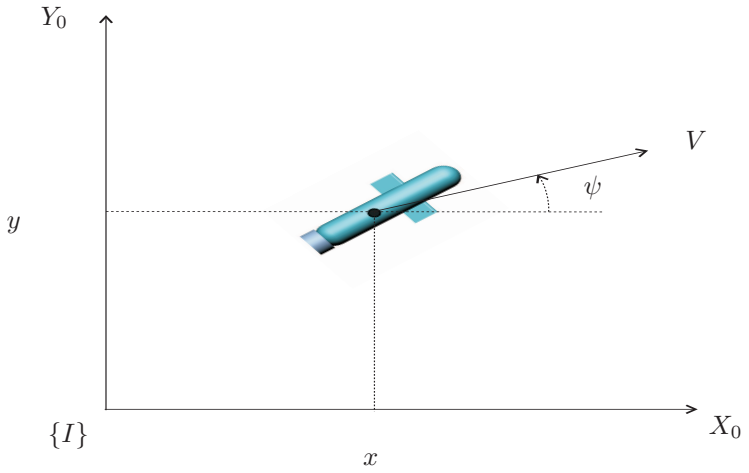


Figure 3.18: Target model. Note that ψ is not the vehicle heading but the angle of the vehicle total velocity.

with state vector $\mathbf{x} = [x\ y\ V\ \psi\ r] \in \mathbb{R}^5$, system matrices

$$\bar{\mathbf{A}} = \begin{bmatrix} 1 & 0 & h\cos(\psi_k) & 0 & 0 \\ 0 & 1 & h\sin(\psi_k) & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1\ h \\ 0 & 0 & 0 & 0 & 0\ 1 \end{bmatrix}, \quad \bar{\mathbf{L}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.13}$$

and disturbance vector $\boldsymbol{\xi} = [\xi_V\ \xi_\psi\ \xi_r]^T \in \mathbb{R}^3$.

3.4.2 Measurement model

In the set-up adopted for vehicle positioning the underwater pinger carries a high precision clock that is synchronized with those of the buoys (and thus with GPS) prior to target deployment. The pinger emits an acoustic signal every T seconds, at precisely known instants of time. To avoid a proliferation of symbols, only one period of interrogation will be examined in detail, the extension to the full time interval being trivial. See Figure 3.19. Let s (at the beginning of an interrogation cycle) denote an arbitrary time at which the pinger emits a signal. In response to this excitation, the buoys i; $i = 1, ..., m$ (where $m = 4$ in a typical GIB setup) compute their distances r_i to the underwater unit at times $s_i \geq s$; $s_i = s + N_i h$, where

$$r_i(s) = \|\mathbf{p}(s) - \mathbf{p}_i(s)\| \tag{3.14}$$

and N_i is the time it takes for the acoustic signal to reach buoy i, modulo the sampling interval h. In the above equation, $\mathbf{p}(s) = [x(s)\ y(s)]^T$ and $\mathbf{p}_i(s) = [x_{hi}(s)\ y_{hi}(s)]^T$ denote the two dimensional target and buoy position respectively,

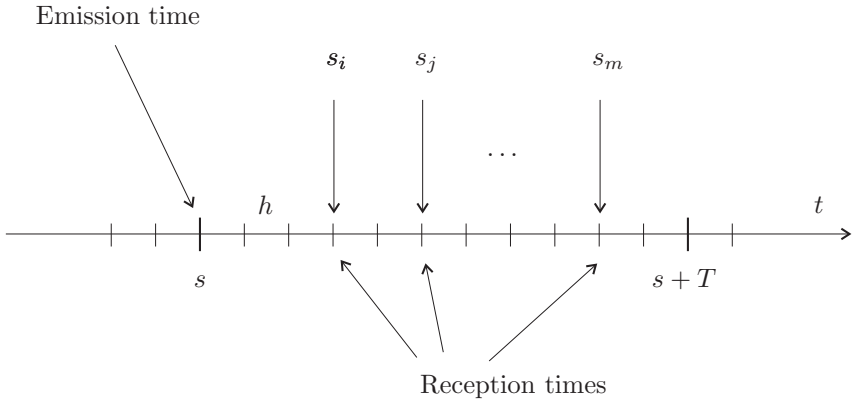


Figure 3.19: Delayed observations. An acoustic pulse is generated at emission time s and observations computed at latter times $s_1, \ldots, s_m$.

at time s. If the buoys are moored and approximately stationary, then the dependence on s vanishes. In what follows it is assumed that $max_i\ N_i h \leq T$, that is, all the receptions at the buoys arrive during an interrogation cycle. The two dimensional range observations r_i are obtained from the times of arrival following the methodology described in section 2.3.5 and a simple constant sound speed model in (3.2). That is, given a time of arrival t_i the corresponding two dimensional range is computed by

$$r_i = \sqrt{t_i^2 v_s^2 - z_p^2} \tag{3.15}$$

where v_s is the sound speed and z_p is the measured pinger depth (as obtained by simple telemetry).

Notice that the N_i's are not necessarily ordered by increasing order of magnitude, since they depend on the distance of each of the buoys to the target. Notice also that even though $z_i = z_i(s)$ refers to time s, its value can only be accessed at time $s_i > s$. It is therefore convenient to define $\bar{z}_i(s_i) = z_i(s_i - N_i h) = z_i(s)$, that is, $\bar{z}_i(s_i)$ is the measurement of $z_i(s)$ obtained at a later time s_i. With the above notation, the model adopted for the noisy measurements $z_i(s)$ of $r_i(s)$ is

$$z_i(s) = r_i(s) + (1 + \eta r_i(s))\, w_i(s), \tag{3.16}$$

where $w_i(s)$ is a stationary, zero-mean, Gaussian white noise process with constant standard deviation σ_i. It is assumed that $w_i(s)$ and $w_j(s)$ are independent for $i \neq j$. The constant parameter η captures the fact that the measurement error increases as the range r_i grows.

The full set of available measurements available over an acoustic emission cycle can vary from 0 to m (where m is typically equal to 4), depending on the conditions of the acoustic channel, and will be denoted by m_k. Mathematically, the set of

$0 \leq m_k \leq m$ measurements can be written as

$$\mathbf{z}^{m_k}(s) \triangleq \mathcal{C}\left[z_1(s) \ \cdots \ z_m(s)\right]^T \tag{3.17}$$

where $\mathcal{C} : \mathcal{R}^m \rightarrow \mathcal{R}^{m_k}$ denotes the operator that extracts the m_k entries in $[z_1(s) \ \cdots \ z_m(s)]^T$ that are actually available and orders them according to the time-sequence at which they are computed at the buoys. Missing entries are simply ignored. Again, it is important to emphasize that even though the information contained in $\mathbf{z}^{m_k}(s)$ refers to time s, it is available in a scattered manner over the interrogation cycle. For clarity of exposition it is convenient to introduce the vector $\mathbf{z}^p(s)$; $p \leq m_k$ that is obtained from $\mathbf{z}^{m_k}(s)$ by keeping its first p components. It is also convenient to define

$$\mathbf{w}^{m_k}(s) \triangleq \mathcal{C}\left[v_1(s) \ \cdots \ v_m(s)\right]^T, \tag{3.18}$$

$$\mathbf{R}^{m_k} \triangleq \mathrm{E}\left\{\mathbf{w}^{m_k}(s)\mathbf{w}^{m_k T}(s)\right\} = \mathrm{diag}\left\{\mathcal{C}\left[\sigma_1 \ \cdots \ \sigma_m\right]^T\right\}. \tag{3.19}$$

In preparation for the development that follows, consider for the time being the "ideal" situation where all or part of the m measurements obtained over an interrogation cycle are available at the corresponding interrogation time s, as condensed in vector $\mathbf{z}^p(s)$; $p \leq m_k$. The procedure adopted to lift this assumption will become clear later. In this case, given the nonlinear process and the observation models given by (3.11) and (3.16), respectively it is simple to derive an EKF structure to provide estimates of positions $x(t_k)$ and $y(t_k)$ based on measurements $\mathbf{z}^p(s)$, where s denotes an arbitrary interrogation time. The details are omitted; see for example [17], [24] and the references therein. Following standard practice, the derivation of an Extended Kalman Filter (EKF) for the design model (3.12) builds on the computation of the following Jacobian matrices about estimated values $\hat{\mathbf{x}}(t_k)$ of the state vector $\mathbf{x}(t_k); t_k = s, s+h, ..., s+T$:

$$\mathbf{A}(\hat{\mathbf{x}}(t_k)) = \frac{\partial f(\mathbf{x},\mathbf{w})}{\partial \mathbf{x}}|_{\hat{\mathbf{x}}(t_k)}, \qquad \mathbf{L}(\hat{\mathbf{x}}(t_k)) = \frac{\partial f(\mathbf{x},\mathbf{w})}{\partial \mathbf{w}}|_{\hat{\mathbf{x}}(t_k)}, \tag{3.20}$$

$$\mathbf{C}(\hat{\mathbf{x}}(s)) = \frac{\partial \mathbf{z}^p}{\partial \mathbf{x}}|_{\hat{\mathbf{x}}(s)}, \qquad \mathbf{D}(\hat{\mathbf{x}}(s)) = \frac{\partial \mathbf{z}^p}{\partial \mathbf{v}}|_{\hat{\mathbf{x}}(s)} \tag{3.21}$$

Notice that the matrices $\mathbf{C}(\hat{\mathbf{x}}(s))$ (abbv. $\mathbf{C}(s)$) and $\mathbf{D}(\hat{\mathbf{x}}(s))$ (abbv. $\mathbf{D}(s)$) are only computed at $t = s$, that is, at the beginning of the interrogation cycle. It is

straightforward to compute

$$\mathbf{A}(\hat{\mathbf{x}}(t_k)) = \begin{bmatrix} 1 & 0 & h\cos(\hat{\psi}(t_k)) & -h\hat{V}(t_k)\sin(\hat{\psi}(t_k)) & 0 \\ 0 & 1 & h\sin(\hat{\psi}(t_k)) & h\hat{V}(k)\cos(\hat{\psi}(t_k)) & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & h \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.22}$$

$$\mathbf{L}(\hat{\mathbf{x}}(t_k)) = \mathbf{L} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.23}$$

Furthermore, by defining

$$\mathbf{C}_i(\hat{\mathbf{x}}(s)) := -\frac{1}{\hat{z}_i(s)} \begin{bmatrix} (\hat{\mathbf{p}}(s) - \mathbf{p}_i(s))^T & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{1\times 5} \tag{3.24}$$

$$\mathbf{D}_i(\hat{\mathbf{x}}(s)) := 1 + \eta\hat{z}_i(s) \in \mathbb{R}, \tag{3.25}$$

where $\hat{z}_i(s) = \|\hat{\mathbf{p}}(s) - \mathbf{p}_i(s)\|$ is the expected rangemeasurement from target to the i'th buoy, it follows that

$$\hat{\mathbf{C}}(\hat{\mathbf{x}}(s)) = \text{stack}^p\{\mathbf{C}_j(\hat{\mathbf{x}}(s))\} \tag{3.26}$$

where $\text{stack}^p(\cdot)$ denotes the operation of stacking p row matrices $\mathbf{C}_j(\hat{\mathbf{x}}(s)); j = 1, ..p$ by re-ordering the sequence of sub-indices j to match the sequence in $\mathbf{z}^p(s)$. For example, if the distances measured by buoys $1, 2$, and 3 are obtained according to the sequence $1, 3, 2$, then

$$\mathbf{C}(\hat{\mathbf{x}}(s)) = \begin{bmatrix} \mathbf{C}_1(\hat{\mathbf{x}}(s)) \\ \mathbf{C}_3(\hat{\mathbf{x}}(s)) \\ \mathbf{C}_2(\hat{\mathbf{x}}(s)) \end{bmatrix}. \tag{3.27}$$

Similarly,

$$\mathbf{D}(\hat{\mathbf{x}}(s)) = \text{diag}^p\{\mathbf{D}_i(\hat{\mathbf{x}}(s)\} \tag{3.28}$$

where the elements of the $p \times p$ diagonal matrix $\hat{\mathbf{D}}(\hat{\mathbf{x}}(s))$ are ordered in an analogous manner. Note that the dimensions of $\hat{\mathbf{C}}(s)$ and $\hat{\mathbf{D}}(s)$ vary according to the number of measurements that are available at time s.

The matrices $\bar{\mathbf{A}}(\hat{\mathbf{x}}(t_k))$ and $\mathbf{A}(\hat{\mathbf{x}}(t_k))$ defined in (3.13) and (3.22) have the following property that will be used later.

Property 3.4.1. *Given any nonzero positive integer N, define*

$$\begin{cases} \alpha_1 = \alpha_1(N,t_k) := \sum_{l=0}^{N} \cos(\hat{\psi}(t_k) + l\hat{r}(t_k)), \\ \alpha_2 = \alpha_2(N,t_k) := \sum_{l=0}^{N} \sin(\hat{\psi}(t_k) + l\hat{r}(t_k)), \\ \beta_1 = \beta_1(N,t_k) := \sum_{l=0}^{N} l\cos(\hat{\psi}(k) + l\hat{r}(k)), \\ \beta_2 = \beta_2(N,t_k) := \sum_{l=0}^{N} l\sin(\hat{\psi}(k) + l\hat{r}(t_k)) \end{cases} \tag{3.29}$$

Then it can be shown that

$$\bar{\mathbf{\Phi}}(t_k + Nh, t_k) := \prod_{l=0}^{N} \bar{\mathbf{A}}(\hat{\mathbf{x}}(t_k + lh)) = \begin{bmatrix} 1 & 0 & h\alpha_1 & 0 & 0 \\ 0 & 1 & h\alpha_2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & hN \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.30}$$

and

$$\mathbf{\Phi}(t_k + Nh, t_k) := \prod_{l=0}^{N} \mathbf{A}(\hat{\mathbf{x}}(t_k + lh)) = \begin{bmatrix} 1 & 0 & h\alpha_1 & -h\hat{V}(t_k)\alpha_2 & -h\hat{V}(t_k)\beta_1 \\ 0 & 1 & h\alpha_2 & h\hat{V}(t_k)\alpha_1 & h\hat{V}(t_k)\beta_2 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & hN \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.31}$$

For $N = 0$,

$$\bar{\mathbf{\Phi}}(t_k, t_k) = \mathbf{\Phi}(t_k, t_k) \triangleq \mathbf{I}. \tag{3.32}$$

3.4.3 Incorporating delayed measurements in the EKF

In the previous section it was assume that all buoy measurements are available at time s, when the interrogation cycle starts. This unrealistic assumption must be lifted in view of the variable time-delay affecting each buoy measurement. The question then arises as to how delayed measurements can be naturally incorporated into an EKF structure. The reader will find in [106] a survey of different methods proposed in the literature to fuse delayed measurements in a linear Kalman Filter structure. In [106], a new method is also presented that relies on "extrapolating" the measurement of a variable obtained with latency to present time, using past and present estimates of the Kalman Filter. The problem tackled in this paper differs from that studied in [106] in two main aspects: the underlying estimation problem is nonlinear, and the components of the output vector that refers to s are accessible with different latencies. As shown below, this problem can be tackled using a *back and forward* fusion approach which recomputes the filter estimates every time a new measurement is available, as depicted in Figure 3.20. The computational complexity

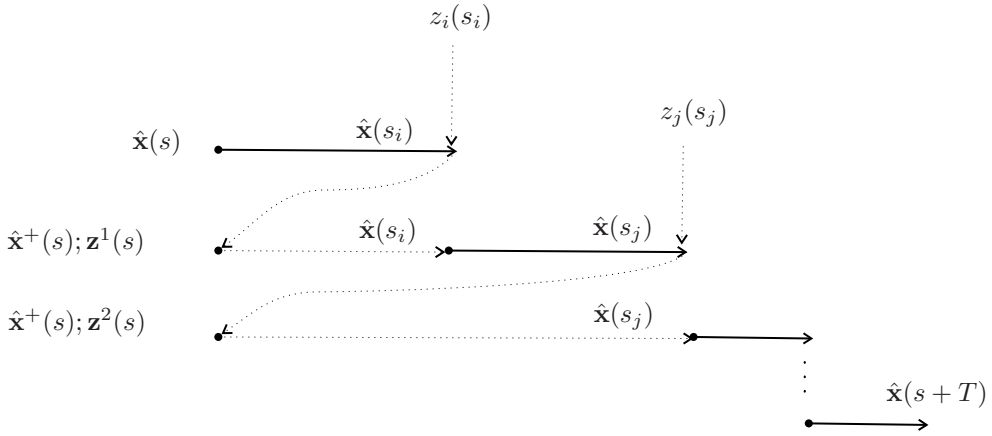


Figure 3.20: *Back and forward* fusion approach. The solid line denotes the availability of real time output filter data.

involved in the algorithm derived is reduced by resorting to Property 3.4.1.

In this work the estimator runs at a sampling period h typically much smaller than T, the interrogation period of the underwater pinger. As before, let s be an arbitrary instant of time at which the underwater pinger emits an acoustic signal and let $i \leq m$ be the buoy that first receives this signal at time $s_i = s + N_i h$. Further let $\bar{z}_i(s_i)$ be the corresponding distance. Up until time s_i no new measurements are available, and a pure state and covariance prediction update are performed using the EKF set-up described before, leading to the *predictor*

$$\hat{\mathbf{x}}(t_{k+1}) = f(\hat{\mathbf{x}}(t_k), 0) = \bar{\mathbf{A}}(\hat{\mathbf{x}}(t_k))\hat{\mathbf{x}}(t_k) \tag{3.33}$$

$$\mathbf{P}(t_{k+1}) = \mathbf{A}(\hat{\mathbf{x}}(t_k))\mathbf{P}(t_k)\mathbf{A}^T(\hat{\mathbf{x}}(t_k)) + \mathbf{L}\mathbf{Q}\mathbf{L}^T \tag{3.34}$$

with $t_k = s, s+h, \ldots, s_i$, where $\hat{\mathbf{x}}(t_k)$ and $\mathbf{P}(t_k)$ are the state prediction and the error prediction covariance, respectively. Upon reception of the first measurement $\bar{z}_i(s_i)$ available during the interrogation cycle, and assuming that the state $\hat{\mathbf{x}}(s)$ and covariance $\mathbf{P}(s)$ at time s have been stored, it is possible to go back to that initial time and perform a *filter state and covariance update* as if the measurement $\bar{z}_i(s_i)$ were in fact available at time s. Using the notation introduced before with $p = 1$ and $\mathbf{z}^p(s) = \bar{z}_i(s_i)$, it is straightforward to compute the update equations

$$\begin{aligned}\mathbf{P}^+(s) = \mathbf{P}(s) - \mathbf{P}(s)\mathbf{C}^T(s) \\ \left[\mathbf{C}(s)\mathbf{P}(s)\mathbf{C}^T(s) + \mathbf{D}(s)\mathbf{R}^p\mathbf{D}^T(s)\right]^{-1}\mathbf{C}(s)\mathbf{P}(s)\end{aligned} \tag{3.35}$$

$$\mathbf{K}(s) = \mathbf{P}^+(s)\hat{\mathbf{C}}^T(s)\left[\hat{\mathbf{D}}(s)\mathbf{R}^p\hat{\mathbf{D}}^T(s)\right]^{-1} \tag{3.36}$$

$$\hat{\mathbf{x}}^+(s) = \hat{\mathbf{x}}(s) + \mathbf{K}(s)\left[\mathbf{z}^p(s) - \hat{\mathbf{z}}^p(s)\right] \tag{3.37}$$

where $\hat{\mathbf{z}}^p(s)$ denotes the estimate of $\bar{\mathbf{z}}^p(s)$ obtained in the previous interrogation cycle. A new prediction cycle can now be done moving forward in time until a new

measurement z_j is available. This is done using (3.33)-(3.34) and starting with the updated states and covariance found in (3.37)-(3.35). Due to property 3.4.1, this prediction can be expressed in a computationally simple form. Let $s_j = s + N_j h$ be the time step at which measurement $\bar{z}_j(s_j)$ is received. Then, the prediction cycle from s to s_j can be computed in closed form as

$$\hat{\mathbf{x}}(s_j) = \bar{\mathbf{\Phi}}(s_j, s)\hat{\mathbf{x}}^+(s) \tag{3.38}$$

$$\mathbf{P}(s_j) = \mathbf{\Phi}(s_j, s)\mathbf{P}^+(s)\mathbf{\Phi}^T(s_j, s) + \sum_{l=0}^{N_j-1} \mathbf{\Phi}(s + lh, s)\mathbf{L}\mathbf{Q}\mathbf{L}^T\mathbf{\Phi}^T(s + lh, s) \tag{3.39}$$

Again, upon computation of measurement $\bar{z}_j(s_j)$, it is possible to go back to time s and perform a filter state and covariance update as if measurements $\bar{z}_i(s_i)$ and $\bar{z}_j(s_j)$ were available at s. This is done using equations (3.37)-(3.35), with the one-dimensional vector $\mathbf{z}^1(s)$ replaced by $\mathbf{z}^2(s) = [\bar{z}_i(s_i), \bar{z}_j(s_j)]^T$ and matrices $\mathbf{C}(s)$,$\mathbf{D}(s)$, and $\mathbf{R}^2$ re-computed accordingly. This *back and forward* structure proceeds until the m_k measurements available over an interrogation cycle (starting at s and ending at $s+T$) are dealt with. The procedure is then repeated for each interrogation cycle. The overall structure of the algorithm proposed is depicted in Figure 3.20.

3.4.4 Measurement Validation and EKF Initialization

In preparation for actual tests of the GIB-based system at sea, this section discusses practical issues that warrant careful consideration. As is well known, the implementation of any acoustic positioning system requires that mechanisms be developed to deal with dropouts and outliers that arise due to acoustic path screening, partial system failure, and multipath effects. See for example [179], [178], [136] and the references therein for an introduction to this circle of ideas and for an interesting application to AUV positioning using a Long Baseline System. In the case of the GIB system, the problem is further complicated because of the mechanism that is used to transmit the depth of the target. In fact, the pinger on-board the vehicle emits two successive acoustic pulses during each emission cycle, the time delay between the two pulses being proportional to the pinger depth. Ideally, the data received at each buoy during each emission cycle consists of two successive pulses only. In practice, a number of pulses may be detected (even though the GIB system only provides 3) depending on the "quality" of the acoustic channel. For example, the data received may correspond to a number of situations that include the following or a combination thereof: i) only the first pulse is received - a valid range measurement is acquired but the depth info is not updated, ii) only the second pulse is received - data contains erroneous information, and iii) a single pulse is received as a consequence of multipath effects - data may be discarded or taken into consideration if a model for multipath propagation is available.

In the present case, following the general strategy outlined in [179], a two stage

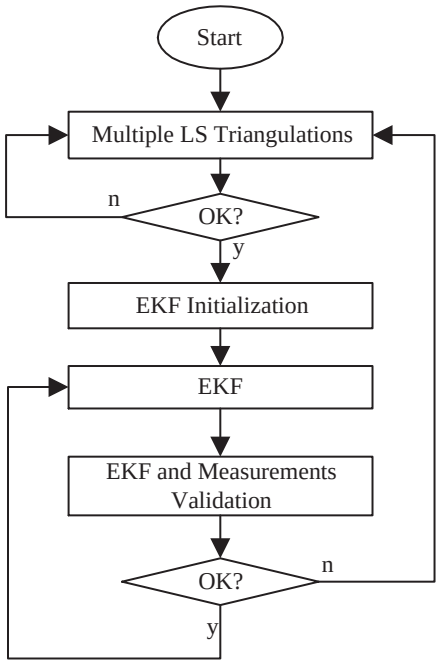


Figure 3.21: Measurement validation and Initialization procedures.

procedure was adopted that includes a time-domain as well as a spatial-domain validation. Time-domain validation is done naturally in an EKF setting by examining the residuals associated with the measurements (i.e., the difference between predicted and measured values as they arrive), and discarding the measurements with residuals that exceed a certain threshold. In this work the threshold is fixed and chosen before system deployment, according to the quality of the acoustic channel. We remark that there is a great potential for the inclusion of a time-varying threshold based on a chi square test on the residuals and the filter covariance, to improve the performance and robustness of the time-domain validation strategy. See for example [24], [124], and the references therein for a lucid presentation of the circle of ideas that can be exploited in future work.

During system initialization, or when the tracker is not driven by valid measurements over an extended period of time, a spatial-domain validation is performed to overcome the fact that the estimate of the target position may become highly inaccurate. This is done via an initialization algorithm that performs multiple Least Squares (LS) triangulations based on all possible scenarios compatible with the set of measurements received and selects the solution that produces the smallest residuals.

The diagram in Figure 3.21 depicts the complete procedure for measurement validation. In an initialization scenario, or whenever a filter reset occurs, the multiple triangulation algorithm is performed until a valid solution is obtained, that is, until the residuals of the resulting set of measurements are less than a certain threshold. Once a valid position fix is obtained, the EKF is initialized and a procedure that

$\mathbf{x}(0)^T$	$\begin{bmatrix}500 & 400 & 1 & \pi/4 & 0\end{bmatrix}$
$\hat{\mathbf{x}}(0)^T$	$\begin{bmatrix}520 & 380 & 0.5 & \pi/2 & 0\end{bmatrix}^T$
$\mathbf{P}(0)$	$\text{diag}\{[(20)^2\ (20)^2\ (0.5)^2\ (0.05)^2\ (0.005)^2]\}$
σ_v	0.001
σ_ψ	0.005
σ_r	0.02
σ_i	$0.1, \qquad i = 1, \ldots, 4$
η	0.001

Table 3.1: Simulation filter parameters.

relies on the EKF estimates and *a priori* information about the vehicle's maximum speed and noise characteristics selects the valid measurements. The EKF will be reset if the residuals become bigger than a threshold or if the duration of a pure prediction phase (that is, the time window during which no validated measurements are available) lasts too long.

3.4.5 Simulations

This section describes the results of simulations aimed at assessing the efficacy of the algorithms derived. In the simulations, four buoys were placed at the corners of a square with a 1 Km side. The depth of the hydrophones z_{hi} was set to 5 m for all the buoys. The target was assumed to move at 1m/s speed along segments of straight lines and circumferences with a 15 m diameter; see Figure 3.22. The motion of the target was restricted to the horizontal plane, at a constant depth $z_p = 50$m. The range measurements were generated every $T = 1$s and corrupted by Gaussian measurement noise as in (3.16) with 0.1m^2 standard deviation. The EKF was run at a sampling period of $h = 0.1$s. The actual and estimated initial states, as well as the process and measurement noise intensities, are shown in Table 3.1.

Figure 3.22 shows a simulation of actual and estimated $2D$ target trajectories and the details of a turning maneuver. Figure 3.23 shows actual and estimated $\psi(t_k)$ as well as the details of actual and estimated $x(t_k)$. Notice the 'jump' in the estimates whenever a new measurement is available. Notice, however how the heading estimates change slowly in the course of a turning maneuver because the positioning system incorporates an internal model for the evolution of yaw rate r. At this point, it is also important to recall that all estimates of the target motion are computed using acoustic range measurements only. In spite of this, the performance of the filter is quite good.

3.5 Experimental Results

In the next section we will present two sets of experimental results obtained during sea trials with a GIB underwater positioning system. The first experimental results

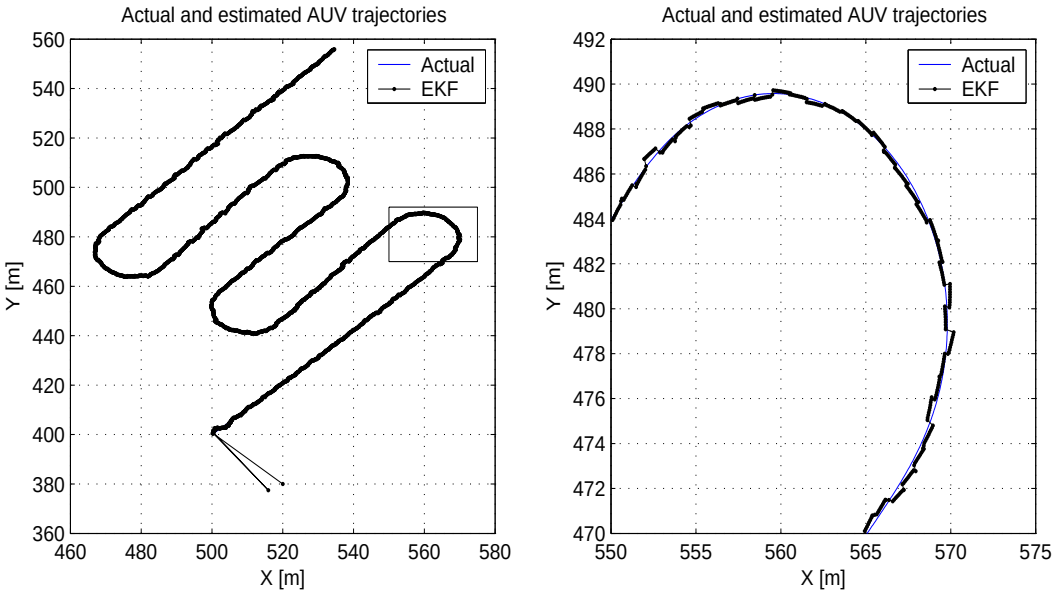


Figure 3.22: Left: simulated and estimated AUV trajectories. Right: idem, zoom-in on boxed area.

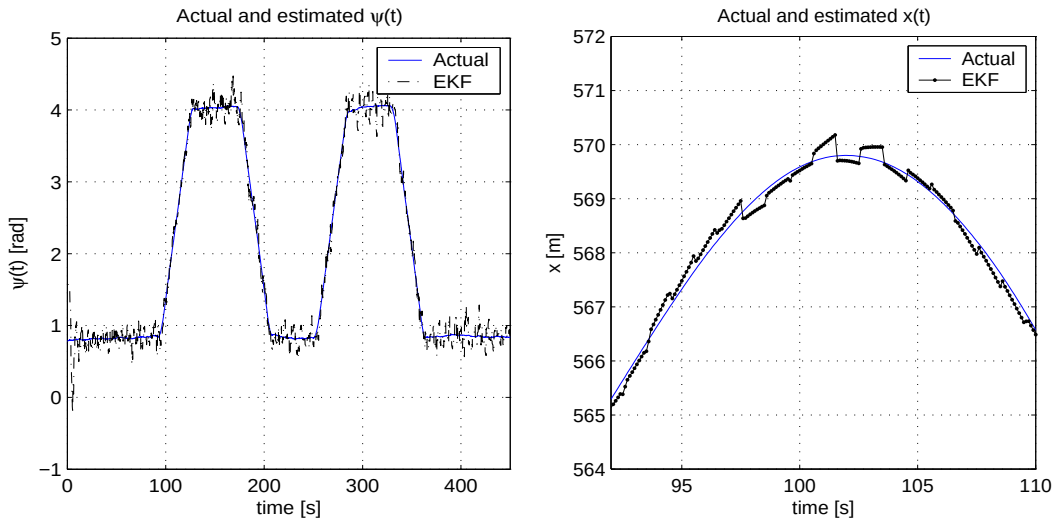


Figure 3.23: Left: simulated and estimated $\psi(t)$. Right: detail of simulated and estimated $x(t)$.

were obtained during a series of sea trials in Sesimbra, Portugal, in June 2004. The objective of the trials was to test and evaluate the performance of the GIB positioning system. This was done by maneuvering the GIB pinger from a ship with a oceanographic surveying platform that provided position measurements (in post processing mode) with an accuracy of 10cm. These post-processed position measurements served as a ground truth for the evaluation of the system. The results of an EKF incorporating delayed measurements described in the previous section was compared to a LS closed form trilateration algorithm and the ground truth.

The second set of experiments was obtained during an underwater archaeological sea trials within the European project VENUS. The experiments were carried out in November 2007 also in Sesimbra. The objective of the trials was to provide position measurements of a Remotely Operated Vehicle (ROV) that performed photometric surveys of the archaeological site *Barco da Telha*. Moreover, the GIB system was used to provide absolute geographical coordinates of the site with respect to which the photometric maps were represented.

3.5.1 Sesimbra sea trials 2004

Evaluating the performance of an underwater positioning system is not an easy task due to the absence of simple procedures capable of yielding very accurate data against which to gage the precision of the position estimates. In some cases, reference positions are produced by off-line smoothing of actual observations, which is clearly far from ideal. Another possible strategy is to use a high frequency LBL to perform such an evaluation [187]. In this work, a different set-up was adopted by mounting the GIB underwater pinger on the IRIS surveying tool seen in Figure 3.24, developed in the scope of the MEDIRES project for automated inspection of both the emerged and submerged parts of rubble-mound breakwaters [154]. The IRIS tool consists of a Inertial Measurement Unit (Seatex MRU-6), two GPS receivers (Ashtech DG14) with the respective antennas carefully installed and calibrated at the bow and stern of the survey vessel, and an underwater body carrying a mechanical scanning pencilbeam sonar (Tritech SeaKing). A GPS receiver (Ashtech DG14) is also installed inshore and calibrated to provide corrections in the post-processing phase. Based on the GPS data acquired, both on-board and inshore, and using commercially available post-processing tools, the location of the underwater unit can be obtained with an accuracy in the horizontal better than 10 centimeters .

For our purposes, the scanning sonar was simply replaced by the GIB pinger. Experimental raw data were acquired using a commercially available GIB system in Sines, Portugal, 24th June 2004, during a campaign of the MEDIRES project at the Sines West breakwater. Four buoys with submerged hydrophones at a nominal depths of 10 meters were moored in a square configuration with approximately 500 meter on the side. Figure 3.25 shows two GIB buoys together with their hydrophones and the GIB pinger attached to a modification of the IRIS structure.

Figure 3.24: The IRIS platform with its arm retracted. The yellow unit can be easily replaced with a support for the GIB pinger. (Courtesy of L. Sebastião).

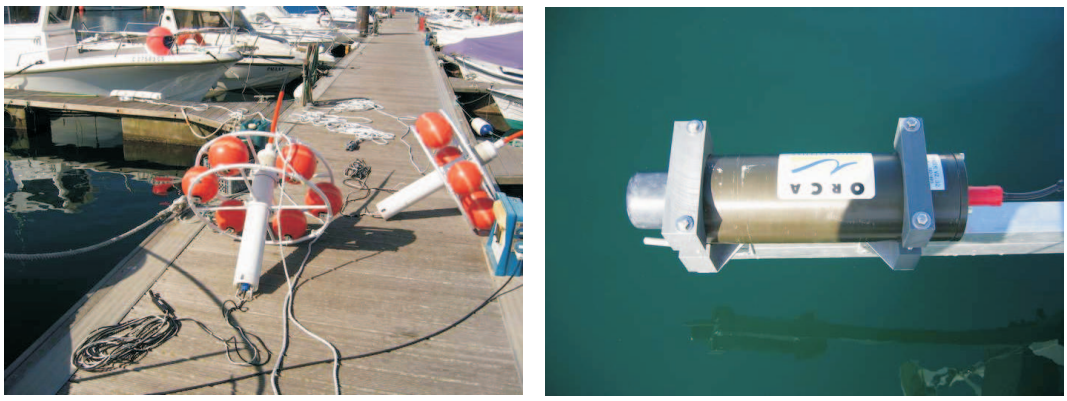


Figure 3.25: GIB buoys (left) and GIB pinger (right) (Courtesy of J. Alves).

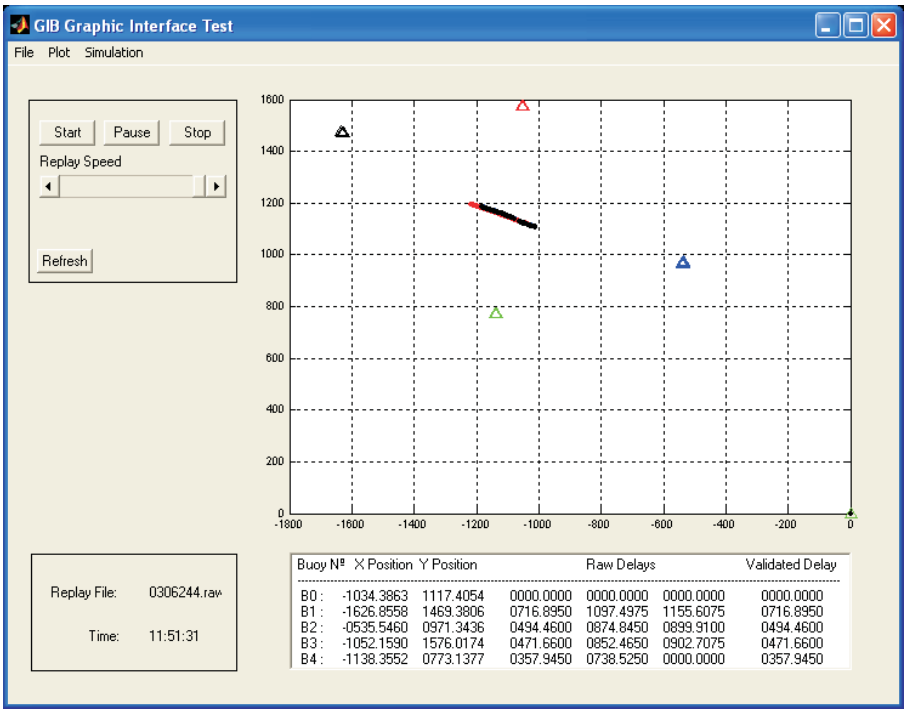


Figure 3.26: Screenshot of positioning system / user interface.

A Matlab based software application was implemented with functions to read and process the raw data from the GIB system log files. These files contain the times of arrival of the sound waves at each of the hydrophones and the buoy positions given by their respective GPS receivers. Because the rate at which data are acquired and processed is low, the positioning algorithm runs on a simple PC using Matlab. This has proved also sufficient for real-time processing. Figure 3.26 is a screenshot of the graphical interface developed to report the status of the proposed algorithm and track the pinger underwater. Table 3.2 shows the filter parameters that were used in the mechanization of the positioning algorithm described in the paper. The interrogation cycle T was set to 1 sec. The sampling time h for the filter was 0.1 sec.

The actual experimental trajectory of the pinger and its estimates can be seen in Figure 3.27. Details are shown in Figure 3.28. In the figures, EKF stands for

$\mathbf{P}(0)$	$\text{diag}\{[(25)^2\ (25)^2\ (1)^2\ (1)^2\ (0.005)^2]\}$
σ_v	$1.58e-2$
σ_ψ	$1.58e-2$
σ_r	$2.5e-10$
σ_i	4 if DGPS, 10 if GPS, $\quad i=1,\ldots,4$
η	$1e-3$

Table 3.2: Experimental filter parameters

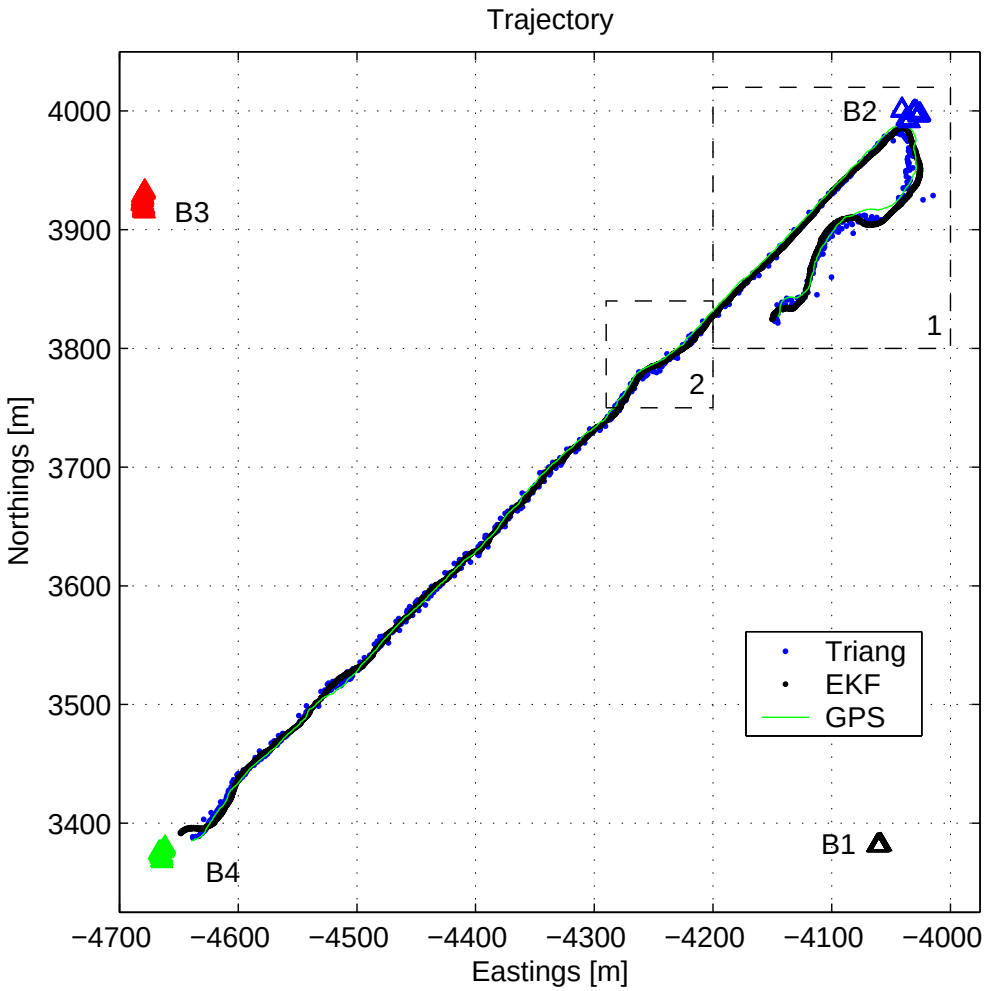


Figure 3.27: Experimental Trajectory.

data obtained with the positioning algorithm proposed, Triang stands for Triangulation fixes, and GPS is the post-processed position reference obtained with the IRIS surveying tool. Note that Triangulation fixes are only computed when 3 or more validated observations are available.

Figure 3.29 shows the positions of the buoys given by GPS. Notice that there is no data from buoy 2 until $t \approx 550s$ into this segment of the mission. Notice also the large jumps in position on the order of $10m$ which are reflected into errors of the underwater positioning system.

Figure 3.30 shows the times of arrival for all buoys. Notice that buoy 2 started transmitting data only at the end of the experiment. Events of this kind can and will occur during real operations at sea. Later, it will be seen how even in the presence of such a dramatic failure the positioning system can still provide good estimates of the underwater target position. Figure 3.31 is the enlargement of regions 1 and 2 in Figure 3.30, that is, of the times of arrival for buoys 1 and 4. The upper and lower

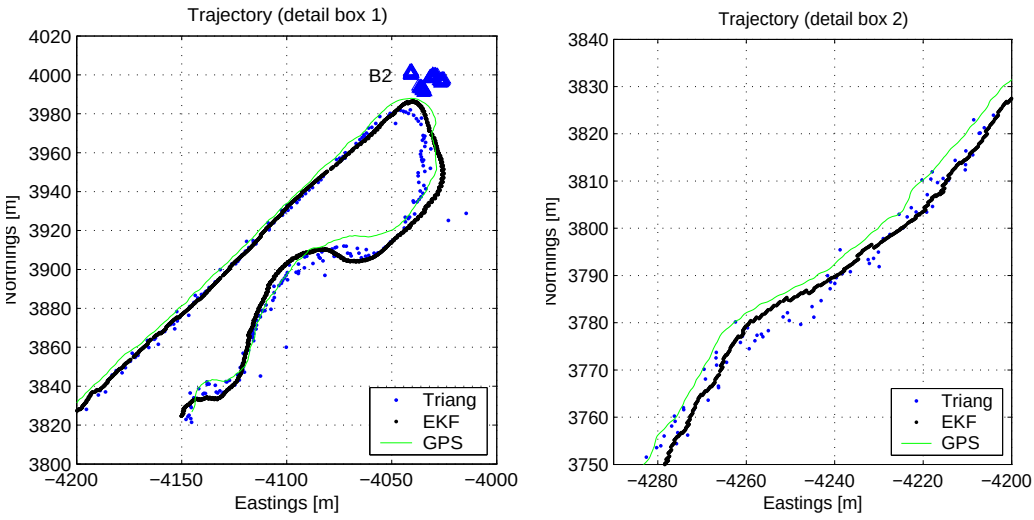


Figure 3.28: Experimental Trajectory (detail).

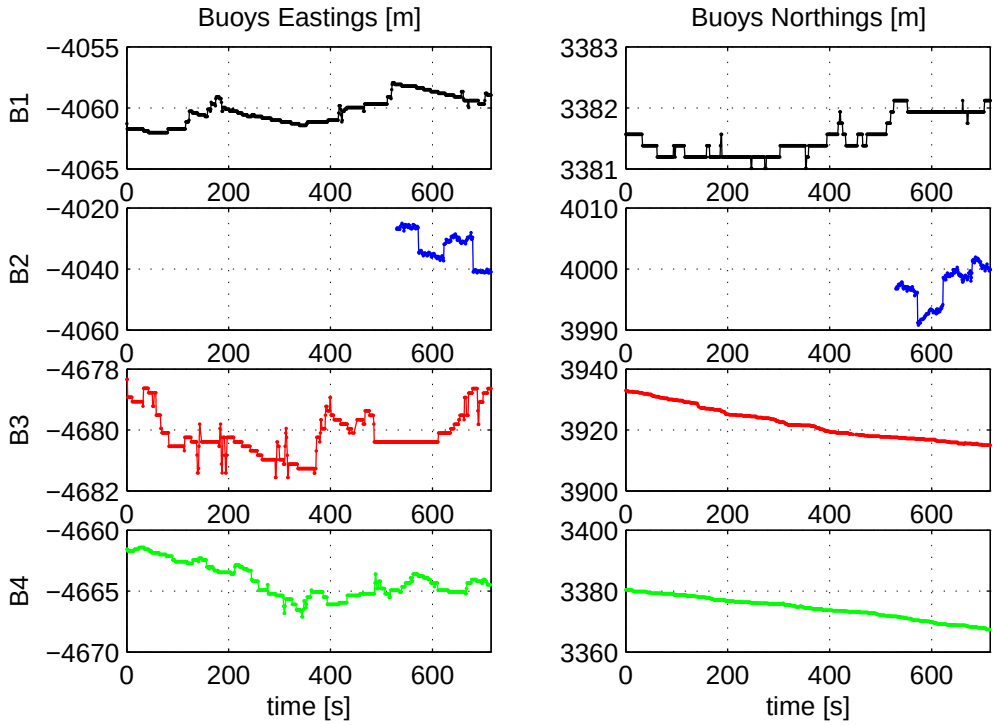


Figure 3.29: Buoy positions given by GPS.

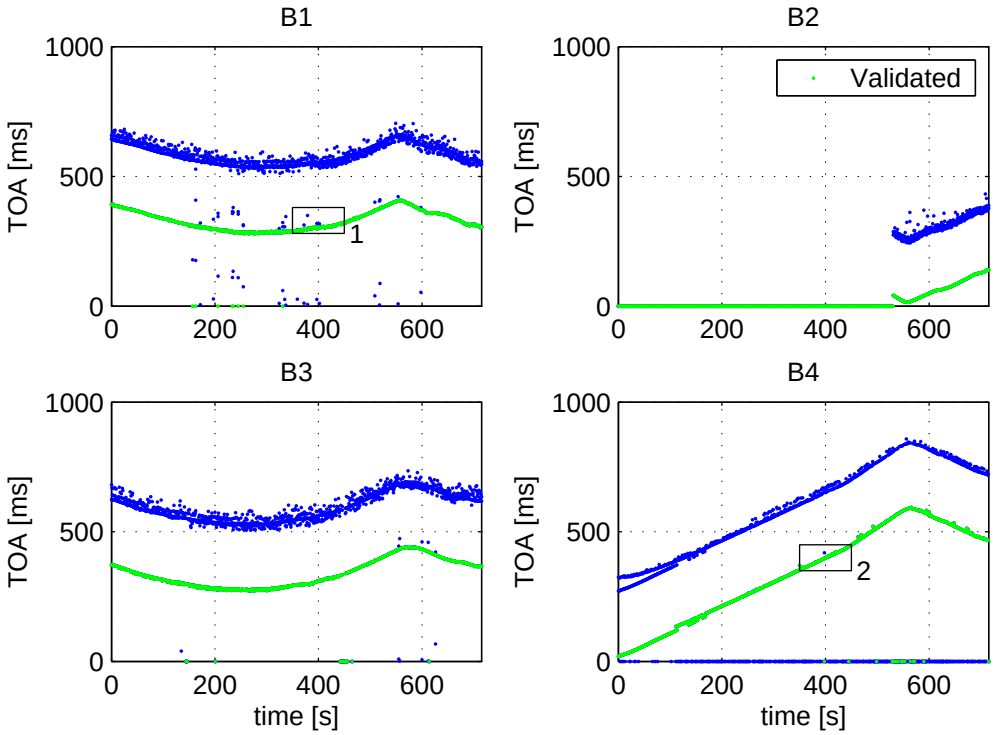


Figure 3.30: Times of arrival at each of the buoys. Validated data in green.

figures correspond to the same time interval and are shown on equal scales. Notice that the times of arrival for buoy 1 are much more noisier than those for buoy 2. Careful analysis suggests that multiple acoustic paths are being detected.

The number of raw and validated observations during the trajectory is shown in Figure 3.32. Again, notice that buoy 2 starts to broadcast data only during the turn in the trajectory, which implies that the straight line between buoys 4 and 2 was done with a maximum of three observations available. There is an almost constant error of about $2m$ between the positions obtained either by EKF or Triangulation and the ones given by IRIS which can be observed in Figure 3.28. One possible source for this error is the fact that the hydrophones might not have been in the vertical directed along the vertical axis of the buoy, due to currents. It is of extreme importance to have an accurate estimate of the sound speed since it is used in the process of transforming differences in the times of arrival into distances. An estimate of the sound speed was determined using a simple least squares algorithm that minimized the residuals of a set of triangulation fixes at the beginning of the experiments, see section 3.3.3.

Figure 3.33 shows the RMS position estimation error of EKF and Triangulation when compared to the post-processed GPS. Clearly, the algorithm proposed yields far better performance than triangulation. The figure does not give total justice to this fact, because it does not reflect the fact that the triangulation fixes are often

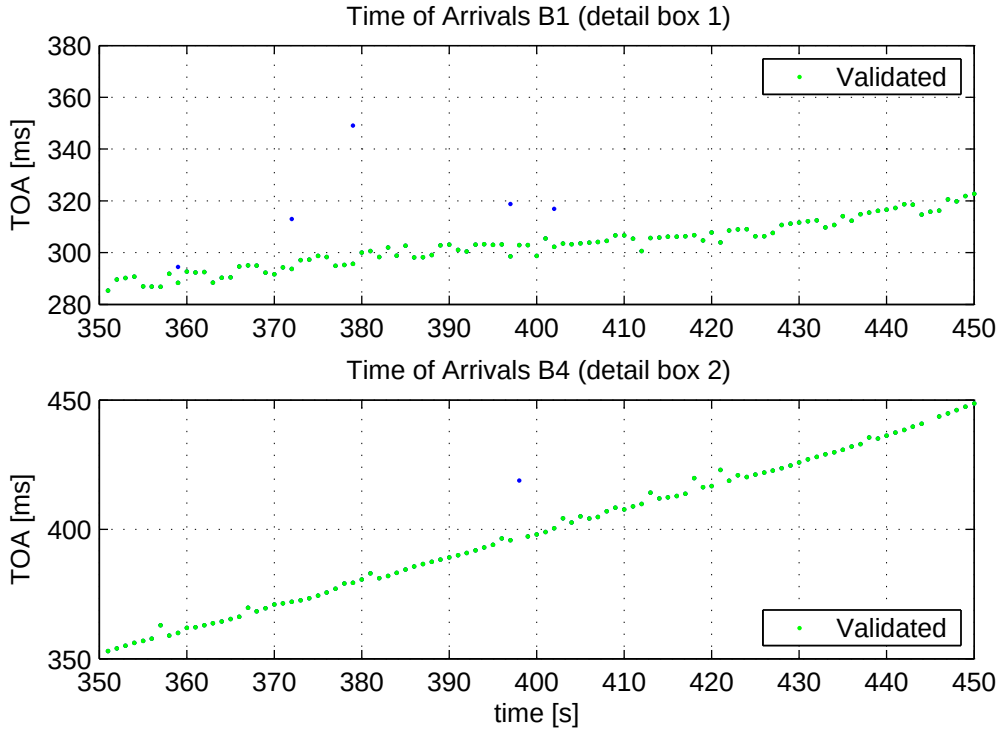


Figure 3.31: Detail of Times of Arrival at buoys 1 and 4.

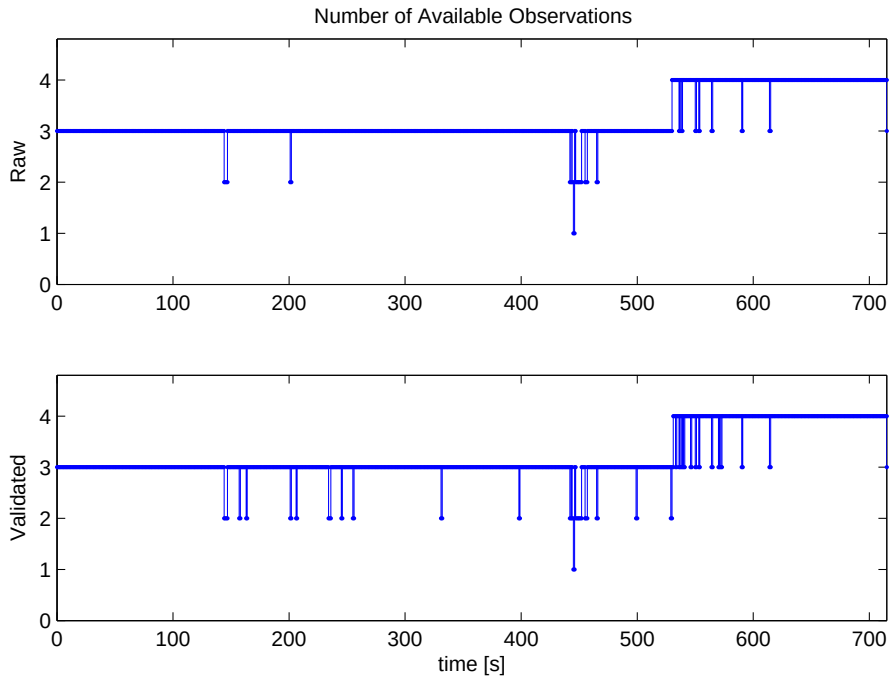


Figure 3.32: Number of available (raw and validated) observations. Only three observations were available until time $t \approx 550s$, when data from buoy 2 started being available.

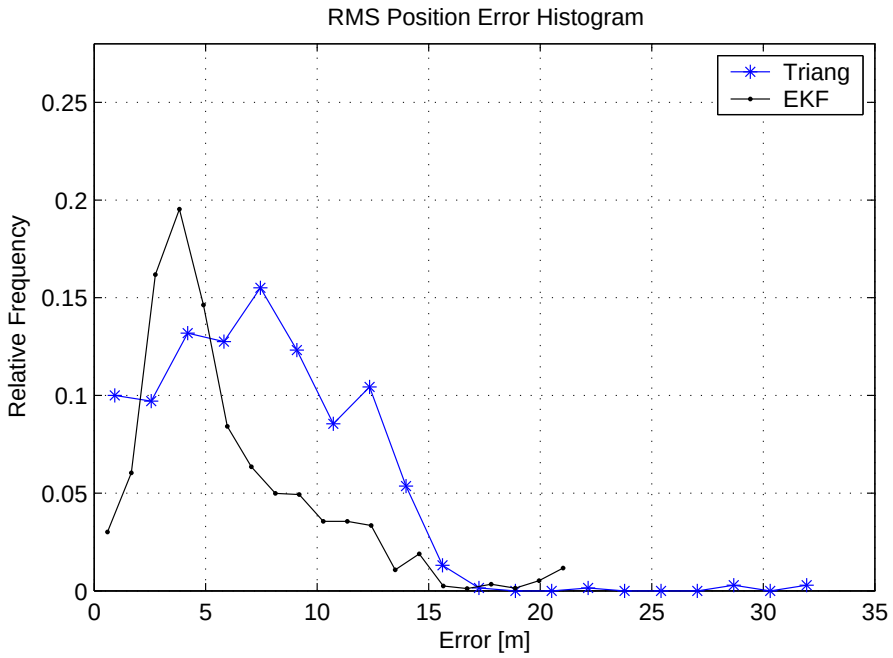


Figure 3.33: RMS of position estimation errors of Triangulation and EKF compared to post-processed GPS.

not available due to bad quality of the data. In fact, only good triangulation data were considered. Furthermore, because the ratio $T/h = 10$ of the EKF filter rate versus the rate of the triangulation updates is large, the samples used to assess the performance of the EKF filter far exceed those used in the triangulation. Again, this fact is not mirrored in Figure 3.33.

3.5.2 Venus sea trials 2007

The second set of experimental data that will be presented correspond to the sea trials of the European project VENUS, which took place in November 2007 in Sesimbra Bay, Portugal. The trials aimed at studying and mapping the recently discovered underwater archaeological site *Barco da Telha*, presumably a XVI wreck carrying a load of ceramic tiles which lied at a sea depth of approximately 60m. The trials included the use of classic and modern archaeological techniques such as photometric surveying by divers and by a Remotely Operated Vehicle (ROV) property of the Italian partner ISME.

The GIB acoustic positioning system was used to provide 3D position estimates of the ROV equipped with a pinger while operating in the site, see Figure 3.36. The ROV was also simultaneously tracked by using an Ultra Short Baseline system, with its transducer installed on the support research vessel AURIGA. The trials also aimed at comparing the performances of both positioning systems.

Figure 3.38 shows the general setup of one of the trials. The figure shows the

Figure 3.34: ISME ROV deployment from research vessel AURIGA.

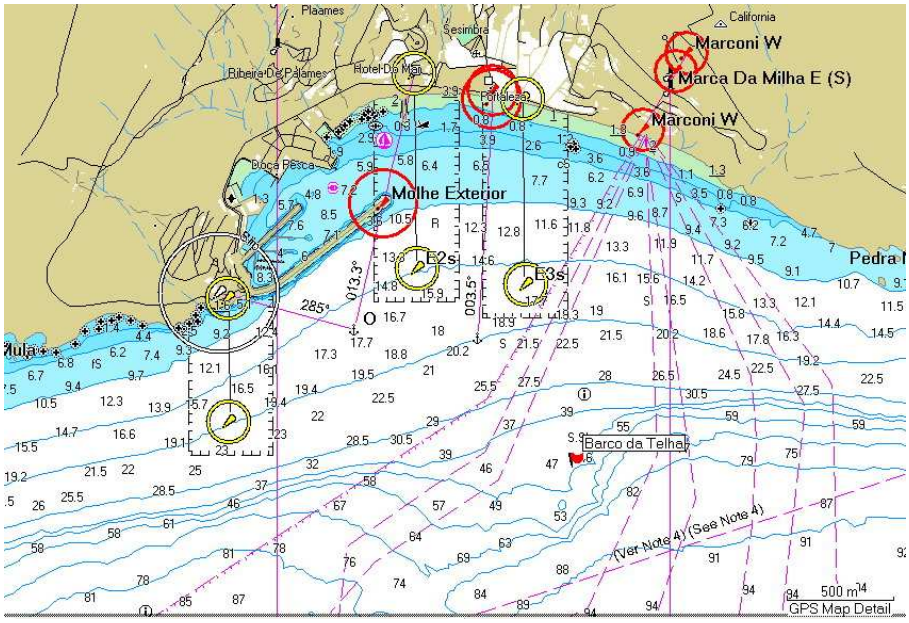
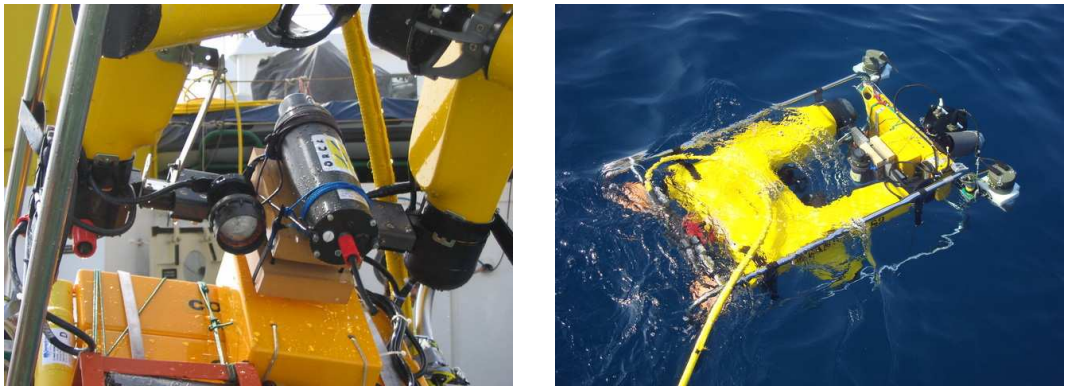


Figure 3.35: Nautical chart of the Sesimbra Bay, Portugal, and the *Barco da Telha* site with bathymetric information that was used in the planing of the mission.

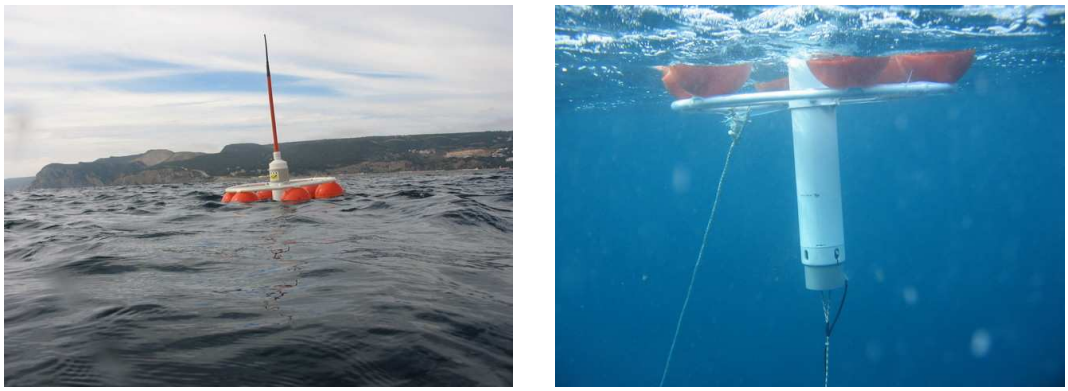


(a) (b)

Figure 3.36: GIB pinger installed on-board ISME ROV.

configuration of the GIB buoys, together with the ROV trajectory as determined using the GIB system, and the position of the support vessel. A local ENU (East North Up) coordinate system was used during the experiments centered at a preliminary site geographical coordinates provided by project member Centro de Nacional de Arqueologia Nautica e Subaquática (CNANS), and is represented with a blue circle in the figure. At the beginning of the trials there was a problem with one of the GIB buoys that could not be fixed on time, so all the experiments had to be carried out with only 3 buoys. The location of the buoys was carefully planned taking into account the local site bathymetry and a favorable geometry. The site was located in a gentle slopping sea bed which reached more than hundred meters few tens of meters away from the wreck, and will make the mooring process a bit more complicated. The final locations of the buoys was set to a triangle shape with the site at its center and in order to minimize the mooring depths. Figure 3.37 shows one of the GIB buoys moored while on operation. The sea state was relatively calm. It is possible to see the beginning of the hydrophone cable (approximately 10m long) in the vertical of the buoy and the mooring line attached to the buoy structure on the left.

The constant sound speed value was determined using the procedure described in 3.3.3 and found $v_s = 1473$m/s. A moderate manual outlier rejection was performed to eliminate some outliers and apparent multipaths from the times of arrival , see Figures 3.39 and 3.40. The ROV depth was computed using a simple algorithm using the fact that at each emission cycle the pinger sends two acoustic pulses separated by a delay proportional to the depth. The resulting depth data was filtered using a simple one dimensional Kalman filter. The the equivalent 2D ranges were then computed using the estimated depth and the times of arrival as described in 2.3.5. The position estimates were obtained by using the MLR cost function minimization and a Newton descent algorithm summarized in 2.4.1. Figure 3.41 shows the com-



(a) (b)

Figure 3.37: Moored GIB buoy on operation.

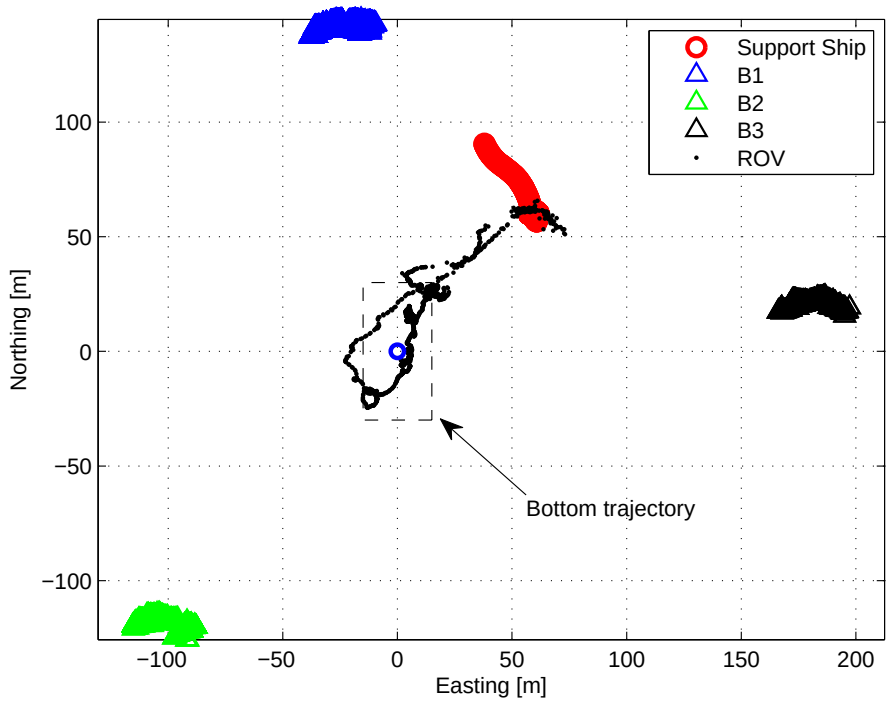


Figure 3.38: Barco da Telha experimental setup.

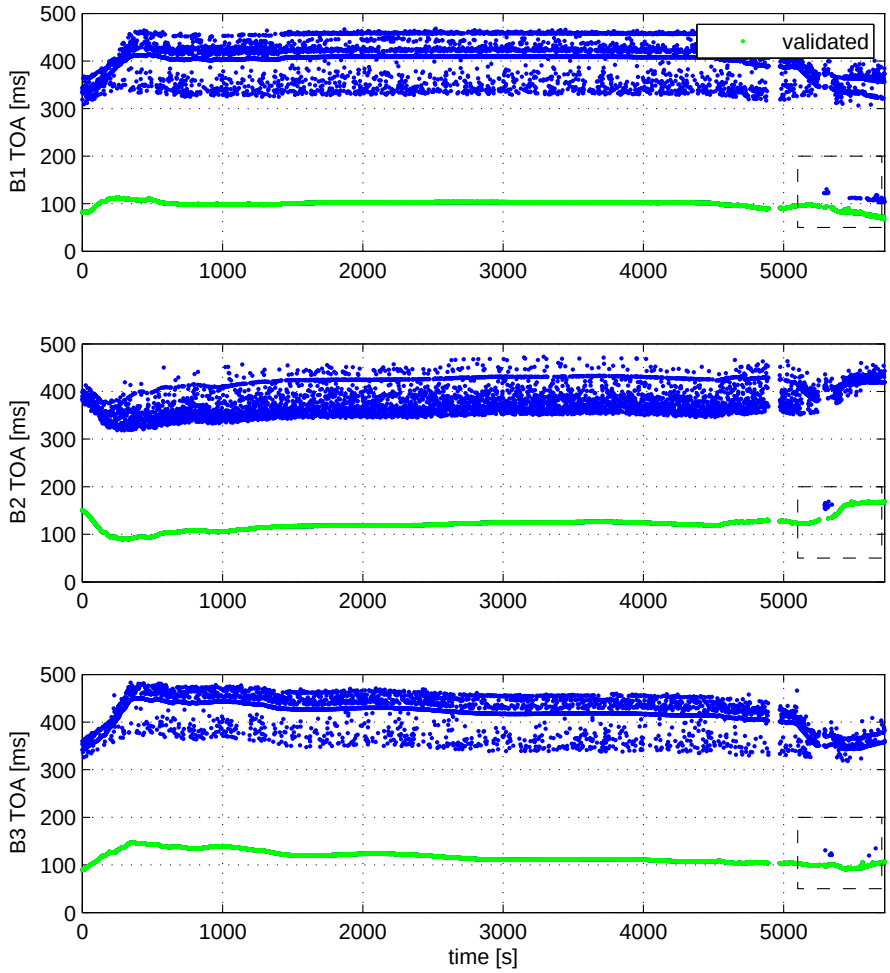


Figure 3.39: Raw and validated TOAs. Good acoustic conditions except for few apparent multipaths at near the end.

puted 3D ROV trajectory during a complete dive. The ROV started at the surface, near the support vessel, then navigated through the surface towards the site where it started the dive. After reaching the sea bed depth, at about -60m, it started a survey of the site area. After that it resurfaced and navigated through the surface back to the support vessel. Figure 3.42 shows a detail of the 2D ROV position fixes near the sea bed (depth $>$ 53m) and Figure 3.43 shows the time evolution of the position components Easting, Northing and depth.

Underwater archaeological site precise localization

The GIB system was used to provide an absolute geographical reference for the local reference system used in the photometric survey. In order to do so, the GIB pinger was moored in a static position near the Barco da Telha site by divers. A set of times of arrival data was recollected during approximately 15 minutes while the pinger remained almost static moored about 50cm above the sea bed. First, a moderate

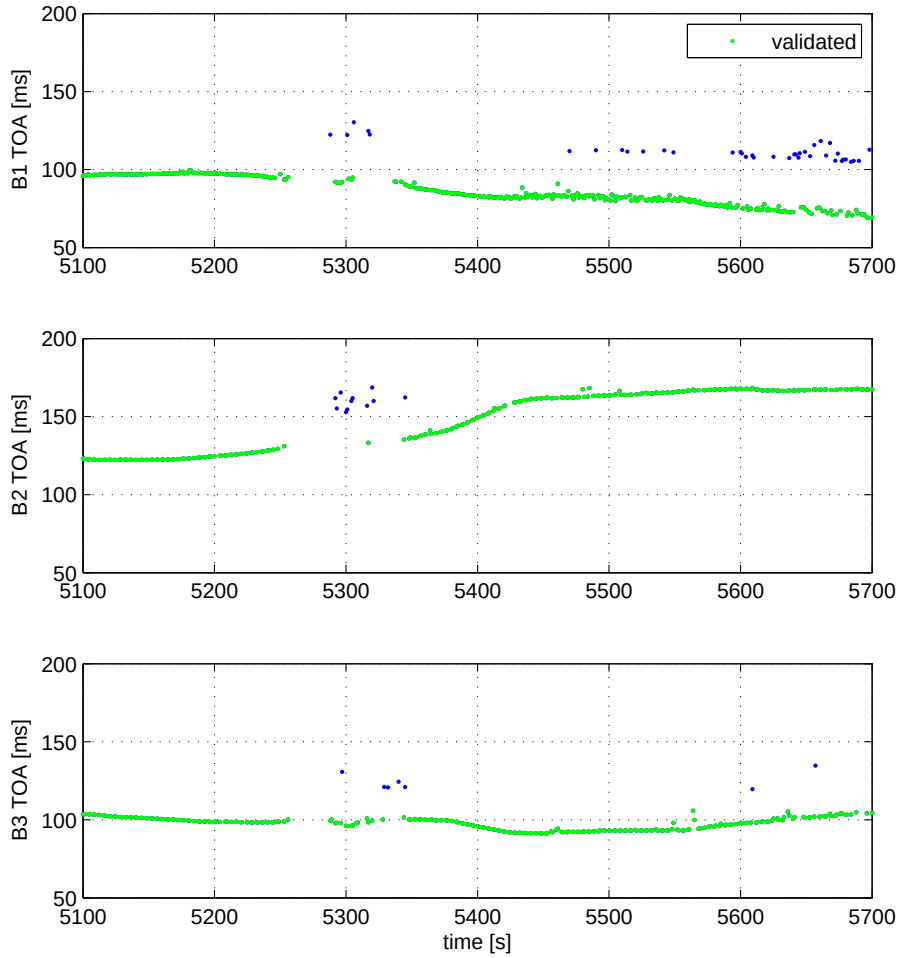


Figure 3.40: Raw and validated TOAs. Apparent multipaths were identified and isolated for later positioning computation.

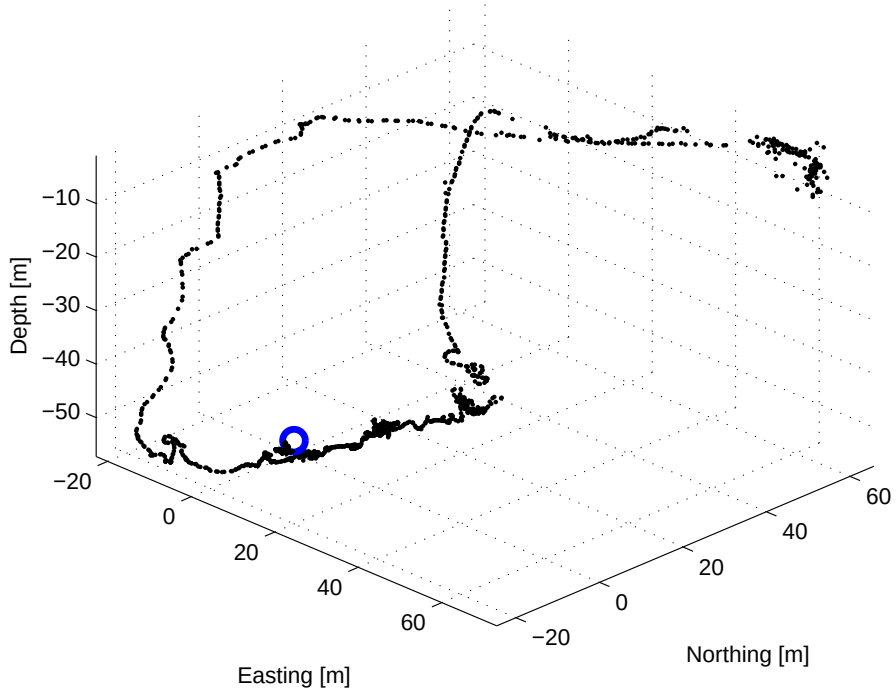


Figure 3.41: ROV 3D trajectory.

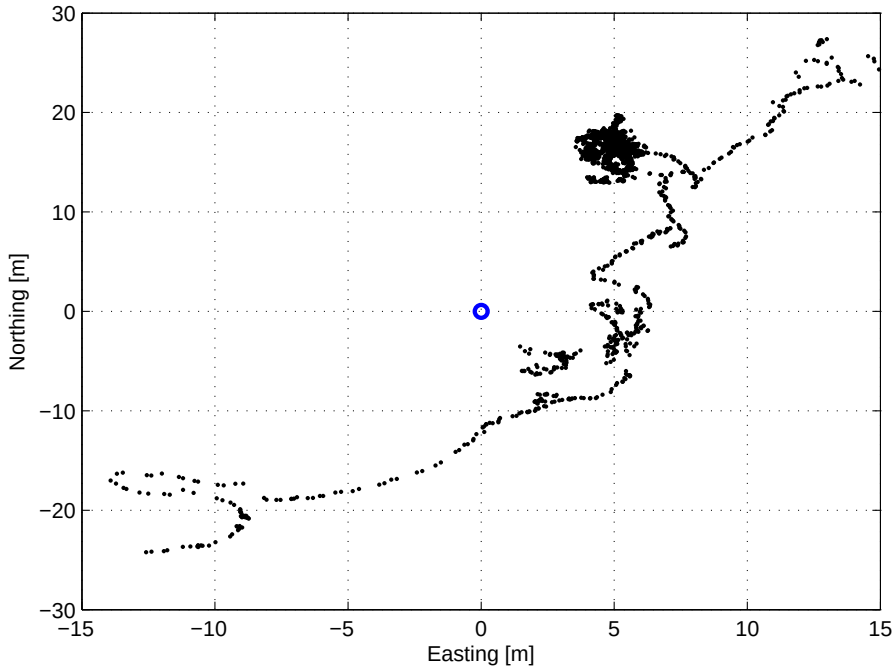


Figure 3.42: ROV bottom trajectory detail.

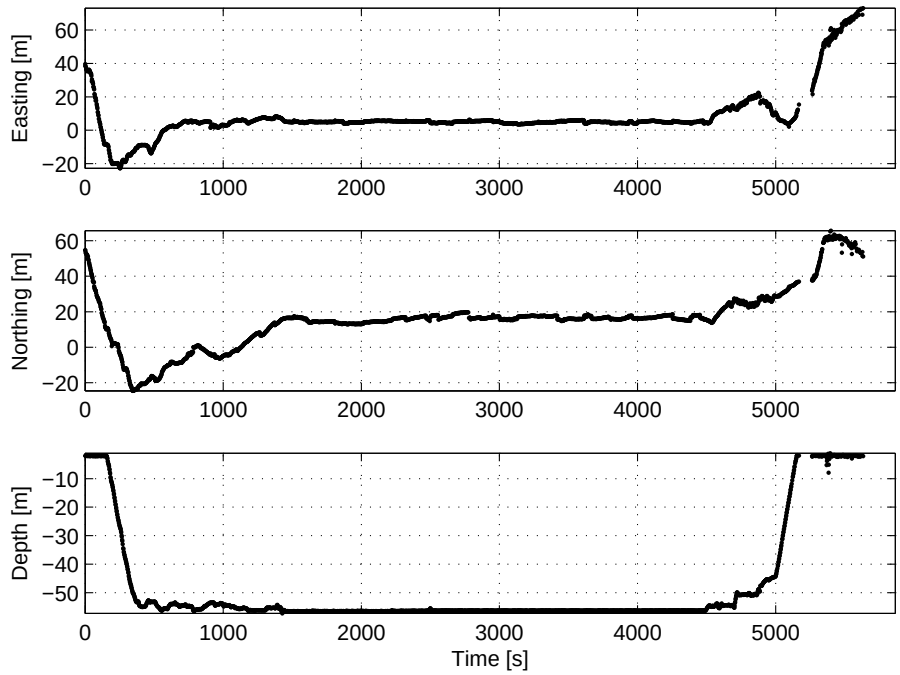


Figure 3.43: ROV trajectory: Easting, Northing and Depth evolution with time.

acoustic outlier rejection was performed on the measured times of arrival, which allowed to remove some apparent multipaths. Then, position fixes were computed from the remaining 919 sets of times of arrival using Newton descent to minimize the ML-R cost function. The value of speed of sound used in the computations was 1482m/s, and was determined initially resorting to a residual minimizing calibration procedure from the available data as described in 2.3.5. Figure 3.44 shows the results of this experiment. The origin of the local coordinate system $(0, 0)$ is shown as a black circle and corresponds to the preliminary site coordinates provided by CNANS. The results of the 919 trilateration position estimates are shown as blue dots, which theoretically correspond to the location of the GIB pinger. The two sigma uncertainty ellipsoid obtained from the experimental data is shown in dashed black line. The average of the trilateration fixes is shown as a black circle, and was chosen as the position of the archaeological site, with respect to which the rest of the scientific data acquired would be georeferenced. Both the local and WGS84 coordinates of the pinger are also shown in the figure. See appendix D for an overview of the transformation between the local and WGS84 coordinate systems.

3.6 Summary and Lessons learned

This chapter has described the application of range only estimation techniques to the operation of a GIB underwater acoustic positioning system. Theoretical as well as real life implementation issues have been introduced and analyzed. An Extended

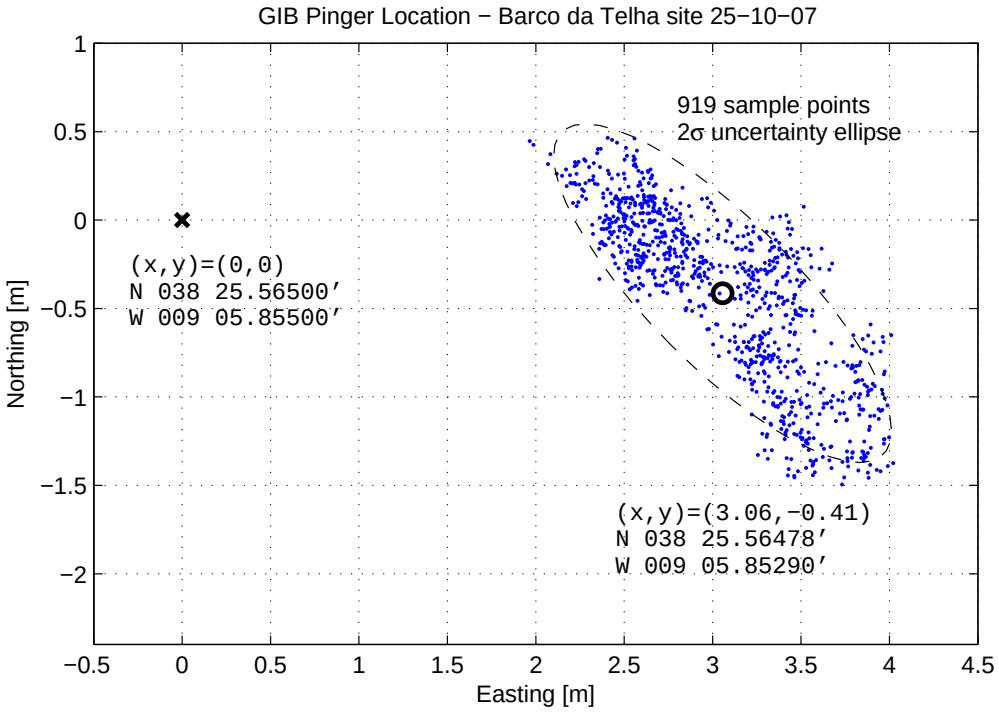


Figure 3.44: GIB pinger static location at Barco da Telha site.

Kalman Filter has been designed and experimentally evaluated that uses a simple vehicle kinematic model and takes into account the inherent delays present in the times of arrival measurements. Experimental results have shown from two different sea trials.

One of the most important lessons learned while operating the system is the theory and practice gap. Dealing with real underwater acoustic times of arrival data required a lot of extra care because of outliers, multipaths, and sound speed estimation. If one wishes to have a completely unsupervised and robust system, the data validation seems one of the most important fundamental blocks of any acoustic positioning system. The importance of the choice of a a particular positioning algorithm has a limited effect as compared to the effects of failing to identify and isolate outliers and multipaths. Moreover, the acoustic conditions are far from a static phenomena. There are temporal and spatial variations that have great effect and need to be considered. The sea temperature varies along the seasons but also varies every day, between night and day. The presence of thermoclines, or local currents with different temperature/salinity may also affect the acoustic conditions. User defined parameters such as filter covariances, thresholds and sound speed values that worked well in one day may produce completely unexpected results in the exact same place in another day with different environmental conditions.

Chapter 4

Range-Only Maximum Likelihood Pose Estimation

This chapter addresses the problem of estimating the attitude and the position of a rigid body when the available measurements consist of ranges between a set of beacons attached to the rigid body and a set of Earth fixed landmarks. This problem arises for instance in some underwater acoustic positioning systems and Indoor tracking systems.

Two cost functions are proposed that are based on the Maximum Likelihood of the range measurements (ML-R) and on a related function of the squared range measurements (ML-SR). The cost functions are naturally defined on the Special Euclidean group $SE(3)$. Borrowing tools from optimization on Riemannian manifolds, intrinsic gradient and Newton-like algorithms are derived to solve the problem. The rigorous mathematical set-up adopted makes the algorithms conceptually simple and elegant. Furthermore, the algorithms do not require the artificial normalization procedures that are recurrent in other estimation schemes formulated in Euclidean space.

Supported by recent results on performance bounds for estimators on Riemannian manifolds, the Intrinsic Variance Lower Bound (IVLB) is derived for the problem at hand. Simulation results are presented to illustrate the performance of the derived algorithms and to validate the tightness of the IVLB in a wide range of signal to noise ratio scenarios.

4.1 Introduction

In the previous chapters we studied the problem of determining the position of a vehicle by using range measurements between some beacon or distance measuring device on-board the vehicle and a set of landmarks with known positions. In some applications the vehicle is equipped with more than one beacon and we are interested in estimating not only its position but also its attitude.

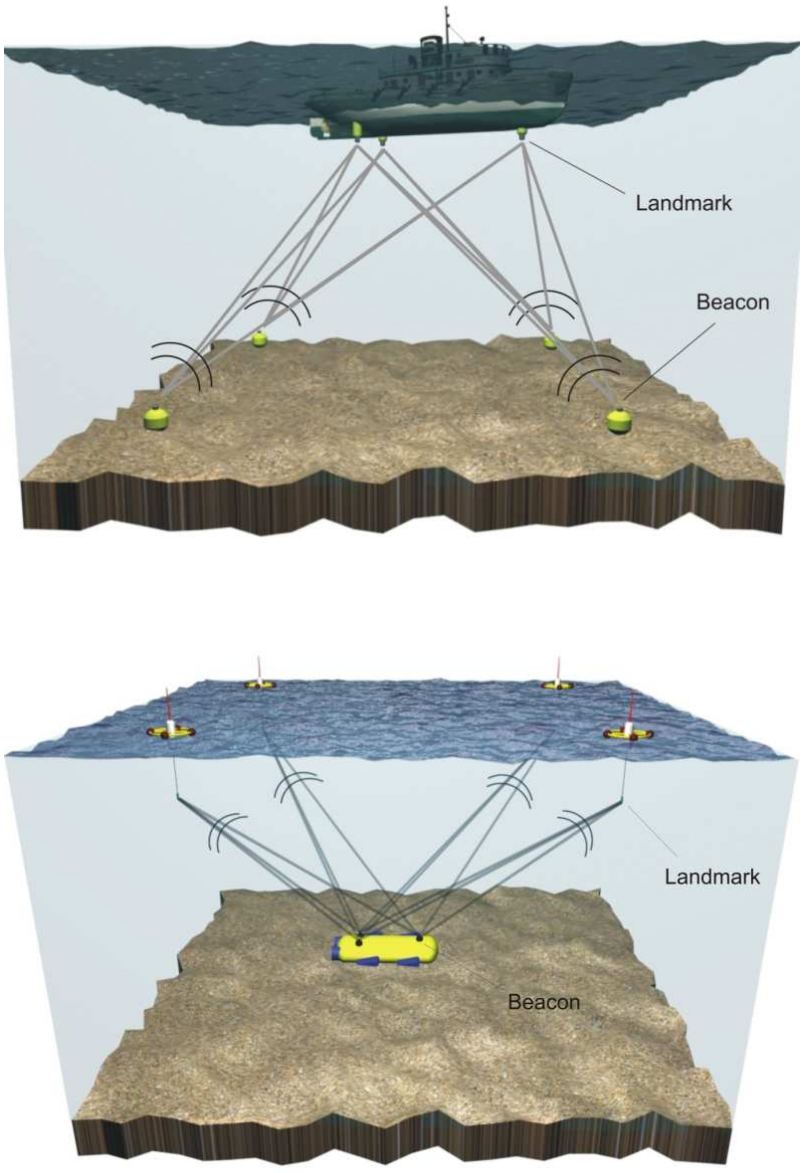


Figure 4.1: Attitude and positioning systems based on underwater acoustics.

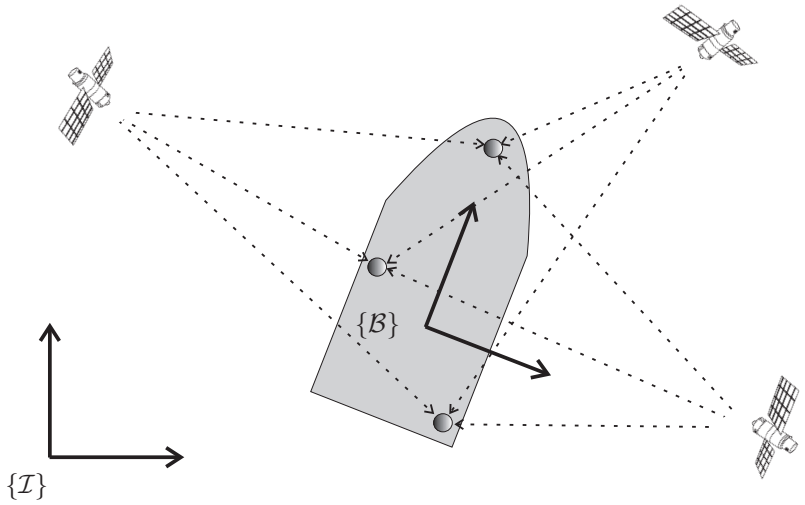


Figure 4.2: Multiple GPS antenna attitude and positioning system.

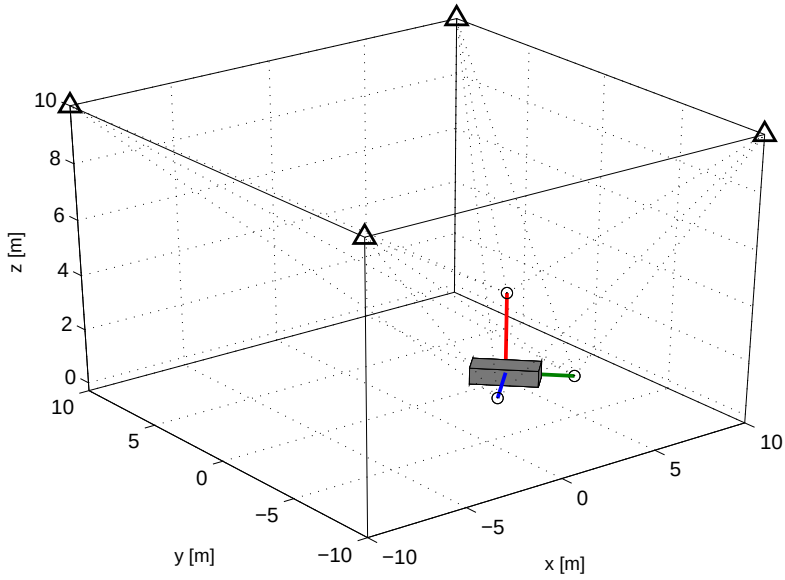


Figure 4.3: Indoor attitude and positioning system.

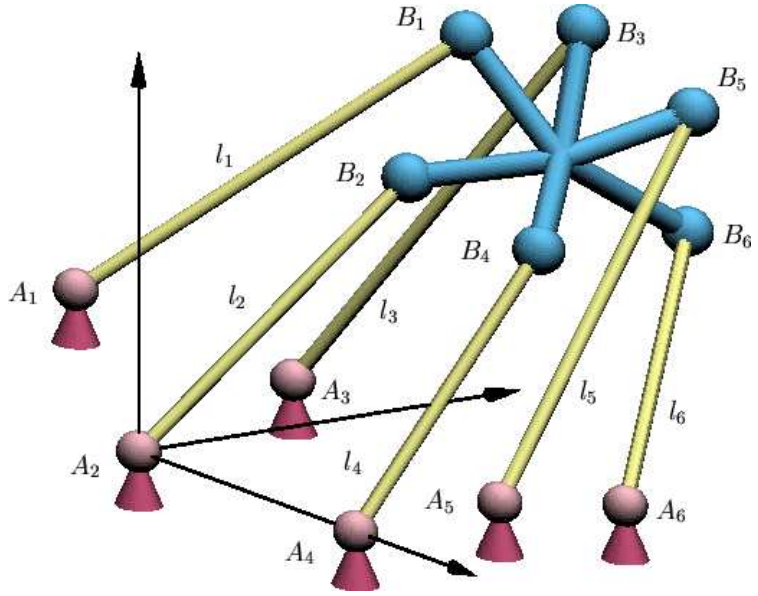


Figure 4.4: Stewart Gough robotic manipulator.

.

The problem of estimating the position of a rigid body given distances (or ranges) between a set of body fixed and earth fixed landmarks with known positions only known in their respective reference frames is common in literature and arises frequently in many practical applications. For instance in GPS and Underwater Acoustic Positioning Systems range measurements are naturally obtained by measuring the time it takes an electromagnetic or acoustic signal to travel between an emitter and a receiver given that the speed of propagation of those signals is assumed to be known.

It is also common to find in the literature work regarding the problem of estimating the attitude (that is the relative orientation of a reference frame with respect to another reference frame) with only range measurements. However, such a problem is often transformed into an equivalent attitude determination problem with vector observations by making the important assumption of planar wavefront. This is a valid assumption when the distances between body fixed landmarks are negligible with respect to the distances between body fixed and earth fixed landmarks. This is the case in multi-antenna GPS applications, where GPS satellites are at distances of about 20000km whereas multiple GPS antennas are separated one from each other usually not more than a few meters. Range observations can be then converted into vector observations when a specific landmark in the rigid body is chosen as a reference, and then range differences with respect to this reference landmark are taken as the new observations. Note that in transforming range into vector observations, the position of the reference landmark is assumed to be known. In practice this is done in a two-step procedure, first we determine the position of the reference landmark and then use this estimate to formulate an equivalent attitude problem with vector

observations for which many methods are available [184], [23], [23] [121], [45], [122] [48], [139], [131], [164]. Considering range differences instead of ranges, has the extra advantage of eliminating any common bias error in the observations. This is actually the case when observations are not ranges but pseudoranges, that is when there is a common bias term in all the measurements, possibly due to an emitter-receiver clock synchronization error.

Attitude/Positioning determination systems based on range only measurements are becoming popular and have received the interest of the engineering community as an alternative to more complex, expensive, and sophisticated Inertial Navigation Systems. A good feature of such systems is that they are drift-less and not sensible to magnetic disturbances. Examples of applications include GPS multi-antenna systems, indoor navigation systems based on wireless networks, and underwater acoustic attitude determination systems based on multiple hydrophone arrays. [45] [139].

Surprisingly enough, the authors are not aware of other works dealing with the problem of attitude and position estimation with range only measurements in an unified manner. However, is intuitive sound to think that both problems are strongly coupled and that some improvements may be achieved by approaching those problems simultaneously. In this chapter we expect to convince the reader that by doing this not only we tackle the problem in a more general and elegant manner but also we are able to derive more efficient estimators. We would like to point out that the way the problem will be formulated does not impose limitations on the roles of the Body fixed and Earth fixed landmarks. Sometimes we refer to them as landmarks and others as distributed sensors. This is simply a way to illustrate the problem at hand. However, it should be clear that their roles can be interchanged arbitrarily and that this does not make any difference from a mathematical point of view. The only assumption we make is that we know their relative distances and their positions in their respective reference frames. Another issue that we should make clear is that we are looking to determine the Attitude and the position of the rigid body given a set of measurements that correspond to a single time instant. The rigid body could be performing a certain trajectory in space but we receive a set of measurements all corresponding to the same time instant. Moreover we assume that no a priory information about past instants of time is available. This is sometimes referred in the literature as instantaneous attitude or positioning problem. Such a class of algorithms may be used in many different ways. For instance, as a way of providing initial estimates for more complex dynamic estimators, as an input for dynamic estimators, or as estimators per se when other algorithms are not available or robustness and memoryless are important features.

The second part of the chapter deals with the problem of deriving performance bounds for Attitude/Position Estimators with range only measurements. Classically this is done by resorting to the Cramér- Rao Bound (CRB) which sets a lower bound

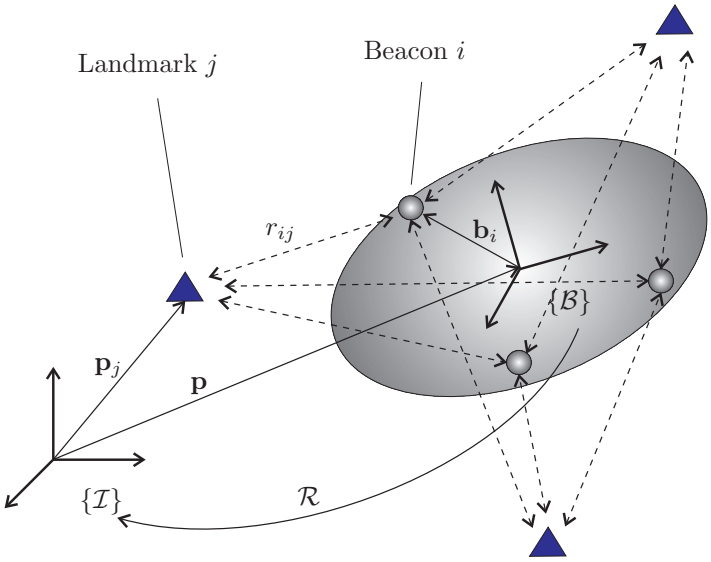


Figure 4.5: Estimation of the attitude and position $(\mathcal{R}, \mathbf{p}) \in SE(3)$ of a rigid body using only relative distance measurements r_{ij}.

for the performance of any unbiased estimator with observations belonging to a given parametric family. A classic reference to the topic is [170], a more modern one with many application examples is [95]. Assuming that the conditional probability of the observations is known it is possible to derive a priory a lower bound for the covariance of any unibiased estimator that uses those observations. This can be of great interest for quite different reasons. For instance, it allows us to benchmark the performance of unbiased estimators. When an estimator attains the CRB we say that it is *efficient* [170]. Another interesting application of performance bounds is observability analysis and sensor geometry planning [175].

The CRB requires, a side from some regularity conditions, the parameter space to be an open subset of $\mathbb{R}^p$ and the estimator to be unbiased. We will see how both requirements are usually violated when the parameter space is a Riemannian Manifold. In our case the parameter space can be identified with the Special Euclidean Group of rigid body motions $SE(3)$. There have been some efforts to define performance bounds or at least qualitative indices for the performance of Attitude Estimators [129] [165]. Recently, new performance bounds have been derived for estimators with parameters on Riemannian Manifolds which take into account the curvature of the parameter space and use the Riemannian distance instead of the usual Euclidean distance. The Intrinsic Variance Lower Bound (IVLB) sets a lower bound for the variance of estimators with parameters on a Riemannian Manifold [190].

4.2 Problem formulation

Suppose that one is interested in estimating the configuration (that is, position and attitude) of a rigid body in space. To this effect, define a Inertial reference frame $\{\mathcal{I}\}$ and a reference frame $\{\mathcal{B}\}$ attached to the rigid body. Let $\mathbf{p} \in \mathbb{R}^3$ denote the position of the origin of $\{\mathcal{B}\}$ with respect to $\{\mathcal{I}\}$ expressed in $\{\mathcal{I}\}$ and let $\mathcal{R} \in SO(3)$ denote the rotation matrix from $\{\mathcal{B}\}$ to $\{\mathcal{I}\}$, where

$$SO(3) = \left\{\mathcal{R} \in \mathbb{R}^{3\times 3} \,:\, \mathcal{R}^T\mathcal{R} = \mathbf{I}_3,\ \det(\mathcal{R}) = 1\right\} \tag{4.1}$$

is the Special Orthogonal group [130]. In the above expression, $\mathbf{I}_3$ stands for the 3×3 identity matrix and $\det(\cdot)$ is the matrix determinant operator. The attitude and position of the rigid body $(\mathcal{R}, \mathbf{p})$ can then be identified with an element of the Special Euclidean group $SE(3) = SO(3) \times \mathbb{R}^3$. There is a vast and rich literature on $SE(3)$, also referred as the group of rigid body motions, a smooth manifold that is not diffeomorphic to $\mathbb{R}^n$ and that can be given the structure of a Lie group [110] [130] [34] [138]. Although they will not be used in the present work, there are several common parametrizations of rotation matrices such as Euler angles, quaternions or angle-axis parametrization [47], [130].

Let us now define what will be our observations. Consider that the rigid body has p beacons and assume that we known where this beacons are located with respect to $\{\mathcal{B}\}$. For instance those could be GPS antennas, or underwater acoustic beacons, arranged with a certain known geometry in the rigid body. Lets further consider that there are m landmarks distributed in the ambient space with known positions. For instance those could be GPS satellites, or acoustic transponders moored in the seabed. Let the positions of the p landmarks in the rigid body be denoted by $\mathbf{b}_i \in \mathbb{R}^3, i \in \{1, \ldots, p\}$ expressed in $\{\mathcal{B}\}$ and the positions of the m Earth fixed landmarks be denoted by $\mathbf{p}_j \in \mathbb{R}^n, j \in \{1, \ldots, m\}$ expressed in $\{\mathcal{I}\}$, see Figure 4.5. Define matrices

$$\mathbf{B} = \begin{bmatrix}\mathbf{b}_1 & \ldots & \mathbf{b}_p\end{bmatrix} \in \mathbb{R}^{3\times p}, \quad \mathbf{P} = \begin{bmatrix}\mathbf{p}_1 & \ldots & \mathbf{p}_m\end{bmatrix} \in \mathbb{R}^{3\times m}, \tag{4.2}$$

which contain the beacon and landmark coordinates in their respective coordinate frames.

The observations are the distances between the m Earth fixed landmarks and the p body fixed beacons corrupted with a Gaussian disturbance. In practice those distances are usually obtained from times of arrival of electromagnetic or acoustic signals, and transformed into distances assuming that the speed of propagation of the signals in the medium is known. Let r_{ij} denote the distance between the i'th rigid body beacon and the j'th Earth fixed landmark. Then,

$$r_{ij} = \|\mathcal{R}\mathbf{b}_i + \mathbf{p} - \mathbf{p}_j\|, \tag{4.3}$$

with $i \in \{1,\dots,p\}$ and $j \in \{1,\dots,m\}$. The observations $\bar{r}_{ij}$ are these distances corrupted with a Gaussian disturbance

$$\bar{r}_{ij} = r_{ij} + w_{ij}. \tag{4.4}$$

Lets stack all the ij values in a more compact form by defining the vectors

$$\mathbf{r} \triangleq \Big[[r_{11} \dots r_{p1}]\ [r_{12} \dots r_{p2}] \dots [r_{1m} \dots r_{pm}]\Big]^T \in \mathbb{R}^{mp}, \tag{4.5}$$

$$\bar{\mathbf{r}} \triangleq \Big[[\bar{r}_{11} \dots \bar{r}_{p1}]\ [\bar{r}_{12} \dots \bar{r}_{p2}] \dots [\bar{r}_{1m} \dots \bar{r}_{pm}]\Big]^T \in \mathbb{R}^{mp}, \tag{4.6}$$

$$\mathbf{w} \triangleq \Big[[w_{11} \dots w_{p1}]\ [w_{12} \dots w_{p2}] \dots [w_{1m} \dots w_{pm}]\Big]^T \in \mathbb{R}^{mp}, \tag{4.7}$$

where this arranging convention follows from considering a matrix with ij elements and applying the vec operator, that is stacking its columns from left to right. With this arrangement, the observations can then be written in a more compact form as

$$\bar{\mathbf{r}} = \mathbf{r} + \mathbf{w}, \quad \mathbf{R} \triangleq \mathrm{E}\left\{\mathbf{w}\mathbf{w}^T\right\} \in \mathbb{R}^{mp \times mp}, \tag{4.8}$$

where $\mathbf{R}$ is the covariance matrix of $\mathbf{w}$ and it should be clear that $\mathbf{r} = \mathbf{r}(\mathcal{R}, \mathbf{p})$. Note that we make no assumption on the structure of $\mathbf{R}$, thus allowing very different kind of disturbance scenarios. For instance, it usually happens that disturbances from equal Earth fixed landmarks are highly correlated. In GPS applications, the signals received at different body fixed antennas from a single GPS satellite have travelled almost through the same propagation channel, thus suffering from very similar disturbances. In that case the covariance matrix $\mathbf{R}$ is block diagonal.

The estimation problem can then be stated as follows:

Problem 4.2.1 (Range-Only Attitude and Positioning problem)**.** *Obtain an estimate of the rigid body position and orientation* $(\widehat{\mathcal{R}}, \widehat{\mathbf{p}}) \in SE(3)$ *given a set of observations* $\bar{r}_{ij}; i \in \{1,\dots,p\}, j \in \{1,\dots,m\}$ *consisting of noisy distances between a set of* p *beacons in the rigid body with positions* $\mathbf{b}_i \in \mathbb{R}^3$ *expressed with respect to* $\{\mathcal{B}\}$ *and a set of* m *Earth fixed landmarks with positions* $\mathbf{p}_j \in \mathbb{R}^3$ *expressed with respect to* $\{\mathcal{I}\}$*.*

4.3 Maximum Likelihood Estimation

Following the methodology introduced in 2.4 we will formulate some constrained optimization problems based on the Maximum Likelihood function of the available range measurements which provide a solution to the Range-Only attitude and positioning problem 4.2.1. The unknown attitude and position pair $(\mathcal{R}, \mathbf{p})$ can be identified with an element of the Special Euclidean group $SE(3)$ which is a smooth manifold not diffeomorphic to $\mathbb{R}^n$. The problem will consist in minimizing some cost functions defined on $SE(3)$. Unfortunately, since the problems do not have a closed

form solution it will be necessary to resort to some iterative optimization algorithms. However, the classic gradient and Newton descent algorithms are only defined when the functions to minimize are defined in an Euclidean space $\mathbb{R}^n$.

In this section we will present generalizations of the classic gradient and Newton descent algorithms that can be used to optimize functions defined in $SE(3)$. The derived tools are based on differential and Riemannian geometry which are the right setup to deal with objects in $SE(3)$. By doing this we will avoid using any local parametrization of rotation matrices, such as quaternions or Euler angles, and avoid some of the problems associated with them [130], [47]. In particular, we will avoid singularities in the representations, and the need for normalization schemes. Although the derivation of the algorithms is relatively complex and require some advanced mathematics, the resulting algorithms are quite simple and can be easily implemented. This section will focus on the results whereas a more advanced treatment and derivations can be found in appendix B.

4.3.1 The ML-R and ML-SR cost functions

Given a rigid body configuration $(\mathcal{R}, \mathbf{p}) \in SE(3)$, it is possible to compute the probability of obtaining a given vector of range observations $\bar{\mathbf{r}}$. This is usually referred as the *likelihood function* and can be computed by

$$p(\bar{\mathbf{r}}|\mathcal{R}, \mathbf{p}) = \frac{1}{(2\pi)^{\frac{mp}{2}} |\mathbf{R}|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\bar{\mathbf{r}} - \mathbf{r}(\mathcal{R}, \mathbf{p}))^T \mathbf{R}^{-1} (\bar{\mathbf{r}} - \mathbf{r}(\mathcal{R}, \mathbf{p}))\right\}. \tag{4.9}$$

The Maximum Likelihood (ML) Estimator is defined as the pair $(\widehat{\mathcal{R}}, \widehat{\mathbf{p}})_{ML}$ that maximizes the *likelihood function* [170] [95]

$$\left(\widehat{\mathcal{R}}, \widehat{\mathbf{p}}\right)_{\text{ML}} = \arg \max_{(\mathcal{R},\mathbf{p}) \in SE(3)} p\left(\bar{\mathbf{r}}|\mathcal{R}, \mathbf{p}\right). \tag{4.10}$$

According to what was done in section 2.4 it is common to work with the *log-likelihood* in order to get rid of the exponential in (4.9) and some constant terms. To this purpose it is convenient to define the Maximum Likelihood with Ranges (ML-R) cost function:

Definition 4.3.1 (ML-R)**.** The Maximum Likelihood with Ranges (ML-R) cost function is defined by $f : SE(3) \to \mathbb{R}$

$$f(\mathcal{R}, \mathbf{p}) = \frac{1}{2}(\bar{\mathbf{r}} - \mathbf{r}(\mathcal{R}, \mathbf{p}))^T \mathbf{R}^{-1} (\bar{\mathbf{r}} - \mathbf{r}(\mathcal{R}, \mathbf{p})), \tag{4.11}$$

where $\bar{\mathbf{r}} \in \mathbb{R}^{mp}$ is a constant vector of range observations, $\mathbf{R} \in \mathbb{R}^{mp \times mp}$ is a positive definite covariance matrix, and $\mathbf{r}(\mathcal{R}, \mathbf{p}) \in \mathbb{R}^{mp}$ can be defined as a mapping that given a rigid body attitude and position it computes the ranges between the body fixed beacons and the Earth fixed landmarks. That is, $\mathbf{r} : SE(3) \to \mathbb{R}^{mp}$, with

$\mathbf{r}(\mathcal{R},\mathbf{p}) = [r_{11}(\mathcal{R},\mathbf{p}) \ \dots \ r_{pm}(\mathcal{R},\mathbf{p})]^T \in \mathbb{R}^{mp}$, and $r_{ij}(\mathcal{R},\mathbf{p}) = \|\mathcal{R}\mathbf{b}_i + \mathbf{p} - \mathbf{p}_j\|$ as defined in (4.3).

The ML estimator can then be found by minimizing the ML-R cost function over the Special Euclidean group $SE(3)$. Because of the square root that defines the range measurements, the ML-R function is not differentiable at the points where the ranges vanish. Similarly to what was done in section 2.4 we will introduce an alternative cost function that uses the squared range measurements and is thus differentiable everywhere.

Define the mapping $\mathbf{d} : SE(3) \to \mathbb{R}^{mp}$ which, given a attitude and position pair, computes the vector of squared range measurements $\mathbf{d}(\mathcal{R},\mathbf{p}) = [d_{11}(\mathcal{R},\mathbf{p}) \ \dots \ d_{pm}(\mathcal{R},\mathbf{p})]^T \in \mathbb{R}^{mp}$, with $d_{ij}(\mathcal{R},\mathbf{p}) = r_{ij}(\mathcal{R},\mathbf{p})^2 = \|\mathcal{R}\mathbf{b}_i + \mathbf{p} - \mathbf{p}_j\|^2$. Define also the vector of squared range measurements $\bar{\mathbf{d}} = [\bar{d}_{11} \ \dots \ \bar{d}_{pm}]^T \in \mathbb{R}^{mp}$ where $\bar{d}_{ij} = \bar{r}_{ij}^2 = (r_{ij} + w_{ij})^2 = r_{ij}^2 + 2r_{ij}w_{ij} + w_{ij}^2 := r_{ij}^2 + \xi_{ij}$. Defining the disturbance vector $\boldsymbol{\xi} = [\xi_{11} \ \dots \ \xi_{pm}]^T = 2\mathbf{r} \odot \mathbf{w} + \mathbf{w} \odot \mathbf{w} \in \mathbb{R}^{mp}$ this can be written in compact form $\bar{\mathbf{d}} = \mathbf{d} + \boldsymbol{\xi}$. As we studied in section 2.3.1, under some realistic assumptions, the squared disturbance error $\boldsymbol{\xi}$ can be approximated by a zero mean Gaussian vector. If the ranges satisfy $r_{ij} >> w_{ij}$, that is if the standard deviation of the range measurements is much smaller than the ranges, then one can approximate $\boldsymbol{\xi} \approx 2\mathbf{r} \odot \mathbf{w}$ which has zero mean and covariance $\boldsymbol{\Sigma}_{\boldsymbol{\xi}} := \mathrm{E}\left\{\boldsymbol{\xi}\boldsymbol{\xi}^T\right\} \approx 4\delta(\mathbf{r})\mathbf{R}\delta(\mathbf{r})$.

Definition 4.3.2 (ML-SR)**.** The Maximum Likelihood with Squared Ranges (ML-SR) cost function is defined by $f : SE(3) \to \mathbb{R}$

$$f(\mathcal{R},\mathbf{p}) = \frac{1}{2}(\bar{\mathbf{d}} - \mathbf{d}(\mathcal{R},\mathbf{p}))^T\boldsymbol{\Sigma}^{-1}(\bar{\mathbf{d}} - \mathbf{d}(\mathcal{R},\mathbf{p})), \tag{4.12}$$

where $\bar{\mathbf{d}} \in \mathbb{R}^{mp}$ is the vector of squared range measurements, $\mathbf{d} : SE(3) \to \mathbb{R}^{mp}$ is the mapping that, given an attitude and position pair, computes the squared ranges between the body fixed beacons and the Earth fixed landmarks, and $\boldsymbol{\Sigma} \in \mathbb{R}^{mp\times mp}$ is a positive definite covariance matrix.

Strictly speaking, the ML-SR cost function is not a Maximum Likelihood function. However, there is a choice of covariance matrix $\boldsymbol{\Sigma}$ that makes it more coherent with the ML criteria. That is, taking $\boldsymbol{\Sigma} = 4\delta(\bar{\mathbf{r}})\mathbf{R}\delta(\bar{\mathbf{r}}) \approx 4\delta(\mathbf{r})\mathbf{R}\delta(\mathbf{r}) \approx \boldsymbol{\Sigma}_{\boldsymbol{\xi}}$ as the covariance of the squared range measurements disturbance vector. Note that since the actual ranges are unknown, we used the range measurements as an approximation for this purpose.

In order to determine the minimizers of the ML-R and ML-SR cost functions we need to solve a constrained optimization problem on the Special Euclidean Group $SE(3) = SO(3) \times \mathbb{R}^3$ (as a set). At this point, it is not obvious how we can continue and effectively solve this constrained optimization problem. The problem does not admit a closed form solution so it is necessary to use some kind of iterative scheme. It turns out that in many cases like the one we are studying, the constrain set can

be shown to be a smooth manifold which, once equipped with a valid Riemannian metric, it becomes a Riemannian manifold. In those situations we are provided with tools and objects from Differential Geometry such as intrinsic gradients, intrinsic Hessians, and geodesics which allow us to solve the constrained optimization problem in an intrinsic manner. What is important about this approach is the fact that for some manifolds as $SE(3)$ these objects can be easily computed, allowing for relatively simple numerical iterative solutions.

4.3.2 Optimization on Riemannian Submanifolds

In this section we will describe a general methodology for optimization on connected embedded Riemannian submanifolds of $\mathbb{R}^n$. The main point will be to show how to generalize the classic gradient and Newton descent methods by defining the intrinsic gradient and Hessian and showing how to perform line searches without leaving the constraint set by following geodesics. The main ideas presented here are taken from [57], and [158], [118]. We will start with a discussion of how a set of smooth constraints can define an embedded submanifold of $\mathbb{R}^n$, which is a common special case of smooth manifold. Then we will briefly introduce some objects and tools from differential geometry that are needed to generalize the gradient and Newton descent methods. There are many good books on the extensive topic of differential geometry. For a good and complete introduction to differential and Riemannian geometry the reader can refer to [109], [110], and [108]. A more advanced treatment can be found in [31], [55], and [150].

Suppose that we want to minimize a smooth function $f : W \subset \mathbb{R}^n \to \mathbb{R}$, where W is an open subset, subject to some constraints. Recall that the function f is smooth if all its partial derivatives of any order exists and are continuous. Furthermore assume that the constraint set $\Theta \subset W$ can be expressed as the level set of a smooth function $h : W \subset \mathbb{R}^n \to \mathbb{R}^{n-d}$. That is, $\Theta = \{\mathbf{x} \in W \, : \, h(\mathbf{x}) = \mathbf{c}\}$ for some constant $\mathbf{c} \in \mathbb{R}^{n-d}$. Lets denote $h(\mathbf{x}) = [h_1(\mathbf{x}), \ldots, h_{n-d}(\mathbf{x})]^T$ and recall that h is smooth if each of the component functions h_i are smooth. Assume that the matrix of gradients

$$\nabla h(\mathbf{x}) = \Big[\nabla h_1(\mathbf{x}) \; \cdots \; \nabla h_{n-d}(\mathbf{x})\Big] \in \mathbb{R}^{n \times n-d} \tag{4.13}$$

has constant rank $n - d$ for all $\mathbf{x} \in \Theta$. In the language of differential geometry, this means that $\mathbf{c}$ is a regular value of h which qualifies the level set Θ as a d-dimensional closed embedded submanifold of $\mathbb{R}^n$ [110, corollary 5.24],[31, ch.3, corollary 5.9].

One of the classic solutions to this constrained optimization problem is the gradient projection method [29] [114]. Assuming we start at some initial estimate $\mathbf{x}_k \in \Theta$ we determine the negative gradient $-\nabla f(\mathbf{x}_k)$ and orthogonally project it onto the constrain set tangent space at $\mathbf{x}_k$. This gives us a feasible direction $\mathcal{D}_k$. Then we perform a line search in the direction $\mathcal{D}_k$ until finding an acceptable point $\widehat{\mathbf{x}}_{k+1}$. Next we project this point orthogonally onto the constraint set to obtain a feasible $\mathbf{x}_{k+1}$, see

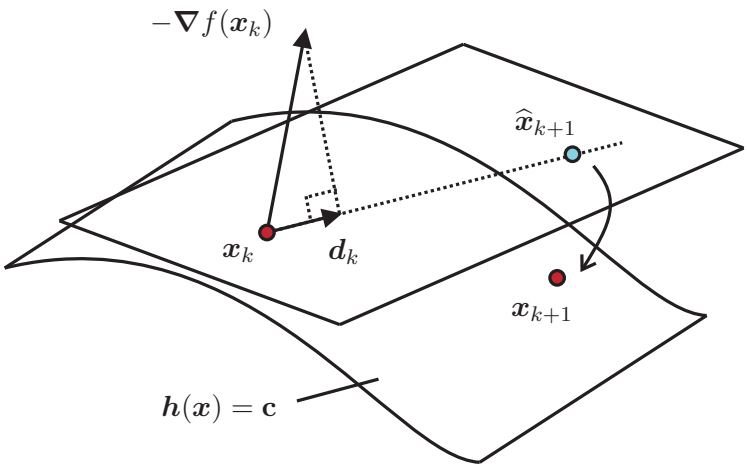


Figure 4.6: Gradient Projection method.

Figure 4.6. Alternatively, if the matrix of second order derivatives $\nabla^2 f(\mathbf{x}_k)$ is invertible, we could have taken $\mathcal{D}_k$ as the orthogonal projection of $-\left[\nabla^2 f(\mathbf{x}_k)\right]^{-1}\nabla f(\mathbf{x}_k)$ defining a Newton like search direction. The gradient projection method has many important drawbacks. First, at each step it involves finding a projection onto the constraint set which can be per se a numerically consuming nonlinear optimization subproblem. Secondly, the line search is performed outside of the constraint set where the function might not even be defined! For instance in the cost functions ML-R and ML-SR defined in (4.11) and (4.12), bizarre results can be expected if the matrix $\mathcal{R}$ is not really a rotation matrix. Some of this issues can be minimized by taking very small step sizes, which in turn affect numerical performance. Now a key question arises: What if we could perform line searches without never leaving the constraint set? This leads us to the concept of geodesic descent and geometric optimization, see Figure 4.7. For the moment just recall that geodesics are the generalization of straight lines on Riemannian manifolds. The gradient projection method and its drawbacks as an alternative to geodesic descent are nicely discussed in [114, ch. 11]. The author treats geodesic descent as an idealization only valid for theoretical analysis due to the impossibility of computing geodesics in a numerically affordable manner. This is actually the case for most Riemannian manifolds, where geodesics have to be computed as the solution of a set of second order nonlinear differential equations. However for some famous manifolds such as $SE(3)$ there are easy closed form expressions to compute them.

Geometric descent optimization provides a generalization of the classic gradient and Newton algorithms. At each iteration a feasible search direction $\mathcal{D}_k$ is computed. This direction can be the intrinsic gradient or intrinsic Newton direction, that will be introduced later. Then a line search along a geodesic is performed until a satisfactory parameter is found. The procedure is repeated iteratively until some stop condition is reached.

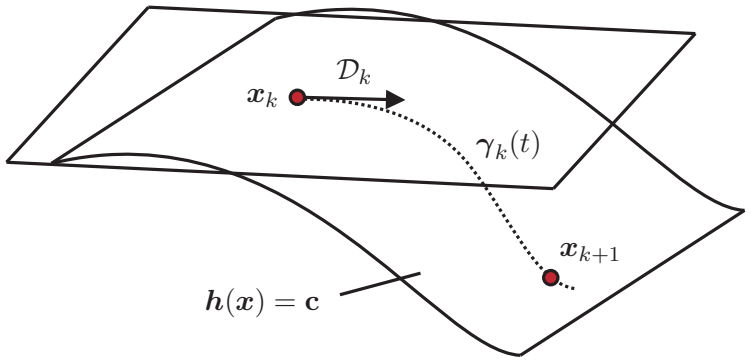


Figure 4.7: Geometric descent optimization.

Next we will introduce the main ingredients of geometric descent optimization in a simple way. A more detailed treatment can be found in appendix B. Let Θ be a d-dimensional closed embedded submanifold of $\mathbb{R}^n$. The most clear example of this kind of manifold is a smooth surface in $\mathbb{R}^3$. The tangent space of a d-dimensional embedded submanifold of $\mathbb{R}^n$ at a point $\mathbf{x} \in \Theta$, denoted by $T_{\mathbf{x}}\Theta$, can be identified with a d-dimensional linear subspace of $\mathbb{R}^n$, given by

$$T_{\mathbf{x}}\Theta = \left\{\Delta \in \mathbb{R}^n \,:\, \nabla h(\mathbf{x})^T \Delta = 0\right\}. \tag{4.14}$$

Recall that we assumed that $\nabla h(\mathbf{x})$ had rank $n - d$ for all $\mathbf{x} \in \Theta$. Hence, $T_{\mathbf{x}}\Theta$, the null space of $\nabla h(\mathbf{x})$, has the same dimension d of the base manifold Θ as we should expect. Tangent vectors can then be identified with very familiar objects, vectors of $\mathbb{R}^n$.

A Riemannian manifold is a smooth manifold that is equipped with a Riemannian metric, that is, a smooth assignment of an inner product to each tangent space. The Euclidean space $\mathbb{R}^n$ has a natural inner product $\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ given by $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \mathbf{y}$. A Riemannian submanifold of $\mathbb{R}^n$ is a smooth submanifold which has been equipped with the canonical metric of its ambient space. That is, given two tangent vectors $\Delta, \Omega \in T_{\mathbf{x}}\Theta \subset \mathbb{R}^n$ its inner product is given by $\langle \Delta, \Omega \rangle = \Delta^T \Omega$.

For each $\mathbf{x} \in \Theta \subset \mathbb{R}^n$, the tangent space $T_{\mathbf{x}}\mathbb{R}^n \simeq \mathbb{R}^n$ can be decomposed into the direct sum $T_{\mathbf{x}}\Theta \oplus N_{\mathbf{x}}\Theta$ where $N_{\mathbf{x}}\Theta$ is the orthogonal complement of the linear subspace $T_{\mathbf{x}}\Theta$ in $\mathbb{R}^n$ [55, p.125].

Intrinsic gradient direction

Let $\widetilde{f} : \mathbb{R}^n \to \mathbb{R}$ be a smooth function and let $f : \Theta \to \mathbb{R}$, be the restriction of $\widetilde{f}$ to Θ, that is, $f = \widetilde{f}|_{\Theta}$. The extrinsic gradient of f at $\mathbf{x} \in \Theta$ is defined as the usual vector of partial derivatives $\nabla \widetilde{f}(\mathbf{x}) = [\frac{\partial}{\partial x_1}\widetilde{f}(\mathbf{x}) \;\cdots\; \frac{\partial}{\partial x_n}\widetilde{f}(\mathbf{x})]^T \in \mathbb{R}^n$. Moreover, according to the above discussion, the extrinsic gradient can be decomposed as the sum of two vectors in $\mathbb{R}^n$ belonging to $T_{\mathbf{x}}\Theta$ and $N_{\mathbf{x}}\Theta$.

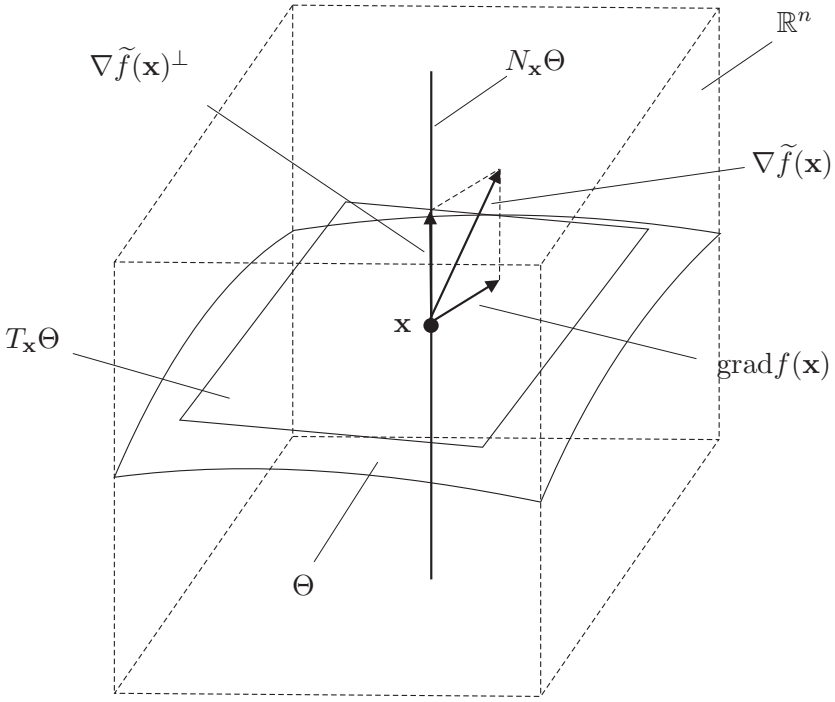


Figure 4.8: Intrinsic gradient.

The intrinsic gradient of f evaluated at $\mathbf{x} \in \Theta$, denoted $\mathrm{grad}f(\mathbf{x})$, is by definition a tangent vector in $T_{\mathbf{x}}\Theta$. It provides a way of locally approximating functions and obtaining generalized (intrinsic) gradient descent search directions. The next proposition summarize fairly well known results, the proofs of which are straightforward exercises in Riemannian geometry:

Proposition 4.3.1 (Intrinsic Gradient)**.** *The intrinsic gradient of f evaluated at $\mathbf{x} \in \Theta$, denoted $\mathrm{grad}f(\mathbf{x}) \in T_{\mathbf{x}}\Theta$, is exactly the orthogonal projection of the extrinsic gradient $\nabla\widetilde{f}(\mathbf{x})$ onto the tangent space $T_{\mathbf{x}}\Theta$. That is, it can be found as the solution of the minimization problem*

$$\mathrm{grad}f(\mathbf{x}) = \arg\min_{\Delta \in T_{\mathbf{x}}\Theta} \|\nabla\widetilde{f}(\mathbf{x}) - \Delta\|. \tag{4.15}$$

Proof. See appendixB. □

The intrinsic gradient descent method at each iteration uses the search direction $\mathcal{D}_k = -\mathrm{grad}f(\mathbf{x}_k)$.

Intrinsic Newton direction

The extrinsic Hessian of f at $\mathbf{x}$ is a map $\mathrm{Hess}\widetilde{f} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ that can be obtained from the usual matrix of second order partial derivatives of $\widetilde{f}$. The intrinsic Hessian of f at $\mathbf{x}$ is defined as a map $\mathrm{Hess}f : T_{\mathbf{x}}\Theta \times T_{\mathbf{x}}\Theta \to \mathbb{R}$ and it will be needed to generalize the Newton direction.

Proposition 4.3.2 (Intrinsic Hessian)**.** *The intrinsic Hessian of the smooth function f at $\mathbf{x} \in \Theta$ satisfies*

$$\mathrm{Hess} f(\Delta_1, \Delta_2) = \mathrm{Hess}\widetilde{f}(\Delta_1, \Delta_2) + \langle II(\Delta_1, \Delta_2), \nabla \widetilde{f}(\mathbf{x}) \rangle \tag{4.16}$$

$$= \Delta_1^T \left(\nabla^2 \widetilde{f}(\mathbf{x}) \right) \Delta_2 + II(\Delta_1, \Delta_2)^T \nabla \widetilde{f}(\mathbf{x}), \tag{4.17}$$

for any tangent vectors $\Delta_1, \Delta_2 \in T_{\mathbf{x}}\Theta$. In the above expression, $II : T_{\mathbf{x}}\Theta \times T_{\mathbf{x}}\Theta \to N_{\mathbf{x}}\Theta$ is the second fundamental form, which relates the geometries of Θ and $\mathbb{R}^n$ [55].

Proof. See appendixB. □

The generalized Newton direction is the unique tangent vector $\mathcal{N}_{\mathbf{x}} \in T_{\mathbf{x}}\Theta$ satisfying $\mathrm{Hess} f(\Delta, \mathcal{N}_{\mathbf{x}}) = -\langle \Delta, \mathrm{grad} f(\mathbf{x}) \rangle$ for all tangent vectors $\Delta \in T_{\mathbf{x}}\Theta$ (assuming that $\mathrm{Hess} f$ is nonsingular at $\mathbf{x}$). The Newton direction can be found by determining the usual matrix of second order partial derivatives and solving a certain linear system of equations. Let $\{E_1, \ldots, E_d\}$, be an orthonormal basis for $T_{\mathbf{x}}\Theta$, that is a set of d mutually orthogonal unit norm vectors of $\mathbb{R}^n$ which span $T_{\mathbf{x}}\Theta$. Since $\mathcal{N}_{\mathbf{x}} \in T_{\mathbf{x}}\Theta$ it can be written as a linear combination of the basis vectors $\mathcal{N}_{\mathbf{x}} = \sum_{i=1}^{d} \alpha_i E_i$ for some scalars $\alpha_1, \ldots, \alpha_d$. The Hessian condition becomes

$$\mathrm{Hess} f(\Delta, \sum_{i=1}^{d} \alpha_i E_i) = \sum_{i=1}^{d} \alpha_i \mathrm{Hess} f(\Delta, E_i) = -\langle \Delta, \mathrm{grad} f(\mathbf{x}) \rangle, \tag{4.18}$$

because of the bilinearity of the Hessian B. Moreover, the above condition must hold for all $\Delta \in T_{\mathbf{x}}\Theta$ and in particular for all the basis vectors $\{E_1, \ldots, E_d\}$ such that

$$\begin{aligned} \alpha_1 \mathrm{Hess} f(E_1, E_1) + \ldots \alpha_d \mathrm{Hess} f(E_1, E_d) &= -\langle E_1, \mathrm{grad} f(\mathbf{x}) \rangle \\ &\vdots \\ \alpha_1 \mathrm{Hess} f(E_d, E_1) + \ldots \alpha_d \mathrm{Hess} f(E_d, E_d) &= -\langle E_d, \mathrm{grad} f(\mathbf{x}) \rangle \end{aligned}$$

which yields the linear system

$$\begin{bmatrix} \mathrm{Hess}(E_1, E_1) & \ldots & \mathrm{Hess}(E_1, E_d) \\ \vdots & \ddots & \vdots \\ \mathrm{Hess}(E_d, E_1) & \ldots & \mathrm{Hess}(E_d, E_d) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_d \end{bmatrix} = - \begin{bmatrix} \langle E_1, \mathrm{grad} f(\mathbf{x}) \rangle \\ \vdots \\ \langle E_d, \mathrm{grad} f(\mathbf{x}) \rangle \end{bmatrix}.$$

Then, assuming that the above system has a unique solution, the Newton descent search direction can be determined according to

$$\mathcal{N}_{\mathbf{x}} = \sum_{i=1}^{d} \alpha_i E_i.$$

The intrinsic Newton descent method uses at each iteration the search direction $\mathcal{D}_k = \mathcal{N}_{\mathbf{x}_k}$.

4.3.3 The geometry of SE(3)

One of the main reasons for us to use tools from differential geometry is that the position and the attitude of a rigid body can be uniquely be identified with an element of the Special Euclidean group $SE(3)$, which is a smooth manifold (and in fact it can be given the structure of a Lie group). In geometric mechanics we would say that $SE(3)$ is the configuration manifold of the rigid body [34], thus the right setup for studying it. The previous and almost all the further development can also be equally applied to the two dimensional case $n = 2$ with minor modifications. The Special Euclidean group also referred as the group of rigid body motions is defined as $SE(3) = SO(3) \times \mathbb{R}^3$, where $SO(3)$ is the Special Orthogonal Group, or the group of rigid body rotations. There are many references in the literature about $SE(3)$ and $SO(3)$. We would like to refer the reader to [27], [31], [126], [130], [138], and [183]. The Special Euclidean group is a 6-dimensional smooth manifold. Moreover, $SE(3)$ can be regarded as being embedded in the Euclidean space $\mathbb{R}^{3\times3} \times \mathbb{R}^3 \simeq \mathbb{R}^{12}$. Elements of $SE(3)$ will be represented as pairs $(\mathcal{R}, \mathbf{p})$ or as vectors $[\text{vec}(\mathcal{R})^T \ \mathbf{p}^T]^T \in \mathbb{R}^{12}$ depending on which representation is more suitable for the computations at hand.

Tangent space and Riemannian metric

Let $\boldsymbol{\theta} = (\mathcal{R}, \mathbf{p})$ be an element of $SE(3)$. The tangent space $T_{\boldsymbol{\theta}}SE(3)$ can be identified with the linear subspace

$$T_{\boldsymbol{\theta}}SE(3) = \left\{(\mathcal{R}\Omega, \mathbf{v}) \ : \ \Omega \in \mathsf{K}(3, \mathbb{R}), \ \mathbf{v} \in \mathbb{R}^3\right\}, \tag{4.19}$$

where $\mathsf{K}(3, \mathbb{R})$ stands for the set of 3×3 skew-symmetric matrices with real entries defined as $\mathsf{K}(3, \mathbb{R}) = \left\{\mathbf{A} \in \mathbb{R}^{3\times3} : \mathbf{A} + \mathbf{A}^T = 0\right\}$. Recall also that sometimes it will be convenient to identify $T_{\boldsymbol{\theta}}SE(3)$ with vectors of the form $\left[\text{vec}(\mathcal{R}\Omega)^T \ \mathbf{v}^T\right]^T$ in $\mathbb{R}^{12}$. Define the map $\mathcal{S} : \mathbb{R}^3 \to \mathsf{K}(3, \mathbb{R})$ and its inverse $\mathcal{S}^{-1} : \mathsf{K}(3, \mathbb{R}) \to \mathbb{R}^3$ by

$$\mathcal{S}\left(\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix}\right) = \begin{bmatrix}0 & -\omega_3 & \omega_2\\ \omega_3 & 0 & -\omega_1\\ -\omega_2 & \omega_1 & 0\end{bmatrix}, \tag{4.20}$$

$$\mathcal{S}^{-1}\left(\begin{bmatrix}0 & -\omega_3 & \omega_2\\ \omega_3 & 0 & -\omega_1\\ -\omega_2 & \omega_1 & 0\end{bmatrix}\right) = \begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix}. \tag{4.21}$$

Note that given a skew symmetric matrix $\Omega \in \mathsf{K}(3, \mathbb{R})$, one can always find a vector $\boldsymbol{\omega} = [\omega_1 \ \omega_2 \ \omega_3]^T \in \mathbb{R}^3$ such that $\Omega = \mathcal{S}(\boldsymbol{\omega})$. Then, the tangent space can also be defined as $T_{\boldsymbol{\theta}}SE(3) = \left\{(\mathcal{R}\mathcal{S}(\boldsymbol{\omega}), \mathbf{v}) \ : \ \boldsymbol{\omega}, \mathbf{v} \in \mathbb{R}^3\right\}$ and identified with vectors of the form $\left[\text{vec}(\mathcal{R}\mathcal{S}(\boldsymbol{\omega}))^T \ \mathbf{v}^T\right]^T \in \mathbb{R}^{12}$.

A Riemannian metric is a smooth assignment of an inner product to each tangent

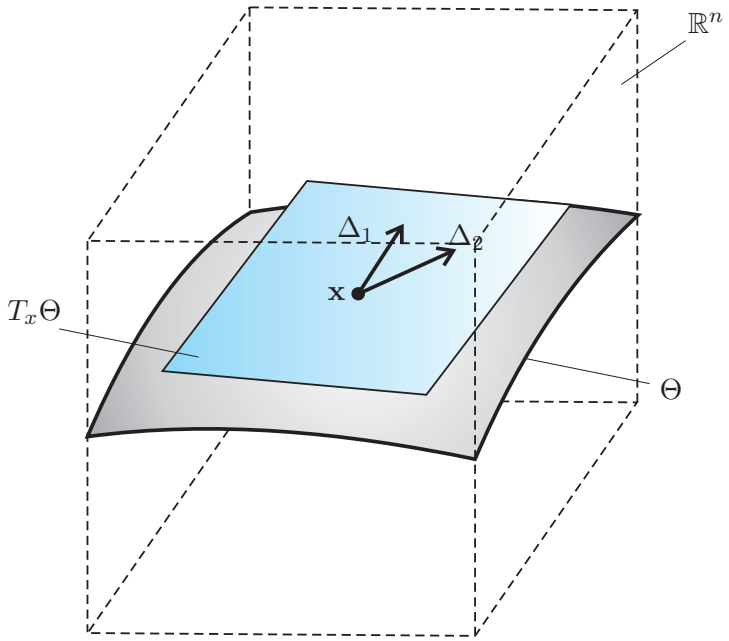


Figure 4.9: Tangent space of a Riemannian submanifold of $\mathbb{R}^n$.

space. The Special Euclidean group can be made an embedded Riemannian submanifold by providing it with the canonical Riemannian metric inherited from its ambient Euclidean space. Let $\Delta_1 = \left[\text{vec}(\mathcal{RS}(\boldsymbol{\omega}_1))^T \ \mathbf{v}_1^T\right]^T$, and $\Delta_2 = \left[\text{vec}(\mathcal{RS}(\boldsymbol{\omega}_1))^T \ \mathbf{v}_2^T\right]^T$ be two tangent vectors in $T_{\boldsymbol{\theta}}SE(3)$. Then the canonical inner product on $SE(3)$ denoted $\langle \Delta_1, \Delta_2 \rangle$ inherited from its Euclidean ambient space $\mathbb{R}^{12}$ becomes

$$\begin{aligned}
\langle \Delta_1, \Delta_2 \rangle = \Delta_1^T \Delta_2 &= \begin{bmatrix} \text{vec}(\mathcal{RS}(\boldsymbol{\omega}_1))^T & \mathbf{v}_1^T \end{bmatrix} \begin{bmatrix} \text{vec}(\mathcal{RS}(\boldsymbol{\omega}_1)) \\ \mathbf{v}_2 \end{bmatrix} \\
&= \text{vec}(\mathcal{RS}(\boldsymbol{\omega}_1))^T \text{vec}(\mathcal{RS}(\boldsymbol{\omega}_2)) + \mathbf{v}_1^T \mathbf{v}_2 \\
&= \text{tr}\left((\mathcal{RS}(\boldsymbol{\omega}_1))^T \mathcal{RS}(\boldsymbol{\omega}_2) \right) + \mathbf{v}_1^T \mathbf{v}_2 \\
&= \text{tr}\left(\mathcal{S}(\boldsymbol{\omega}_1)^T \mathcal{R}^T \mathcal{RS}(\boldsymbol{\omega}_2) \right) + \mathbf{v}_1^T \mathbf{v}_2 \\
&= \text{tr}\left(\mathcal{S}(\boldsymbol{\omega}_1)^T \mathcal{S}(\boldsymbol{\omega}_2) \right) + \mathbf{v}_1^T \mathbf{v}_2 \\
&= 2\boldsymbol{\omega}_1^T \boldsymbol{\omega}_2 + \mathbf{v}_1^T \mathbf{v}_2.
\end{aligned} \tag{4.22}$$

Note that the induced Riemannian metric of (4.22) is in fact equivalent to the *scale-dependent left-invariant metric* of [138] (with $c = 2$, and $d = 1$) and to the canonical product metric on $SO(3) \times \mathbb{R}^3$ (when $SO(3)$ is regarded as a Riemannian submanifold of $\mathbb{R}^{3\times 3}$).

The induced Riemannian distance function on $SO(3)$ is given by

$$d_{SO(3)}(\mathcal{R}_1, \mathcal{R}_2) = \sqrt{2} \arccos\left(\frac{\text{tr}\left(\mathcal{R}_1^T \mathcal{R}_2\right) - 1}{2} \right). \tag{4.23}$$

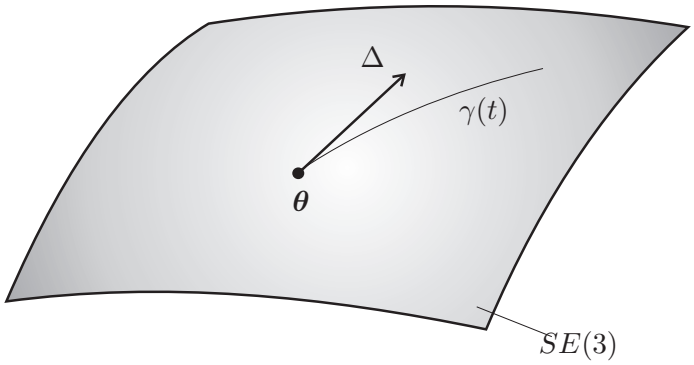


Figure 4.10: Geodesics.

This follows from [126], [138] after removing the multiplicative factor $\frac{1}{\sqrt{2}}$ as argued in [190]. The distance function on $\mathbb{R}^3$ is the usual Euclidean distance $d_{\mathbb{R}^3}(\mathbf{p}_1, \mathbf{p}_2) = \|\mathbf{p}_1 - \mathbf{p}_2\|$. The Riemannian distance on $SE(3)$ induced by the metric (4.22) is as follows. Given $\boldsymbol{\theta}_1 = (\mathcal{R}_1, \mathbf{p}_1) \in SE(3)$ and $\boldsymbol{\theta}_2 = (\mathcal{R}_2, \mathbf{p}_2) \in SE(3)$ the (intrinsic) distance between $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ is given by

$$d_{SE(3)}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \sqrt{d_{SO(3)}(\mathcal{R}_1, \mathcal{R}_2)^2 + \|\mathbf{p}_1 - \mathbf{p}_2\|^2}. \tag{4.24}$$

Geodesics

As discussed previously, the geodesics of $SE(3)$ have a simple closed form expression which can be used to perform computationally affordable intrinsic line searches. If the canonical metric (4.22) is used, the geodesics of $SE(3)$ can be obtained from those of the product space $SO(3) \times \mathbb{R}^3$ as discussed in [138] and [183].

The tangent space of $SO(3)$ at a point $\mathcal{R}$ can be identified with $T_{\mathcal{R}}SO(3) \simeq \mathcal{R}\Omega$ where $\Omega \in \mathsf{K}(3, \mathbb{R})$. It is well known that geodesics emanating from $\mathcal{R} \in SO(3)$ in the direction of $\Delta = \mathcal{R}\Omega$ are given by $\gamma : \mathbb{R} \to SO(3)$,

$$\gamma(t) = \mathcal{R} \exp\left(\mathcal{R}^T \Delta t\right) = \mathcal{R} \exp\left(\Omega t\right), \tag{4.25}$$

where exp is the matrix exponential (see e.g. [31]). It is easily verified that geodesics satisfy $\gamma(0) = \mathcal{R}$ and $\dot{\gamma}(0) = \Omega$. Let us just mention that $SO(3)$ is geodesically complete, which means that geodesics are defined for all $t \in \mathbb{R}$. Note that this is not the usual case, where geodesics are defined only in a neighborhood of the origin. The geodesics of $\mathbb{R}^3$ are the simplest we can imagine: straight lines. The geodesic emanating from $\mathbf{p} \in \mathbb{R}^3$ in the direction of $\mathbf{v} \in \mathbb{R}^3$ is given by $\gamma(t) = \mathbf{p} + t\mathbf{v}$.

Let $\boldsymbol{\theta} = (\mathcal{R}, \mathbf{p}) \in SE(3)$ and $\Delta = (\mathcal{R}\Omega, \mathbf{v}) \in T_{\boldsymbol{\theta}}SE(3)$. The geodesic emanating from $\boldsymbol{\theta}$ in the direction of Δ is given by $\gamma : \mathbb{R} \to SE(3)$,

$$\gamma(t) = (\mathcal{R} \exp\left(\Omega t\right), \mathbf{p} + \mathbf{v}t), \tag{4.26}$$

where it is easy to verify that $\gamma(0) = \boldsymbol{\theta}$ and $\dot{\gamma}(0) = \Delta$.

Second fundamental form

Let $\Delta_1 = (\mathcal{R}\Omega_1, \mathbf{v}_1)$, and $\Delta_2 = (\mathcal{R}\Omega_2, \mathbf{v}_2)$ be tangent vectors at some point $\boldsymbol{\theta} = (\mathcal{R}, \mathbf{p}) \in SE(3)$. The second fundamental form $II : T_{\boldsymbol{\theta}}SE(3) \times T_{\boldsymbol{\theta}}SE(3) \to T_{\boldsymbol{\theta}}SE(3)^{\perp}$ can be found from (B.21) after some computations as

$$II(\Delta_1, \Delta_2) = (-\mathcal{R}\,\mathrm{symm}(\Omega_1^T\Omega_2), 0), \tag{4.27}$$

where symm is the operator that extracts the symmetric part of a matrix, that is $\mathrm{symm}(\mathbf{A}) = \frac{1}{2}(\mathbf{A} + \mathbf{A}^T)$. It can be verified that $II(\Delta_1, \Delta_2)$ is in fact a vector in $T_{\boldsymbol{\theta}}SE(3)^{\perp}$ and is orthogonal to all vectors in $T_{\boldsymbol{\theta}}SE(3)$ with respect to the induced Riemannian metric 4.22. To see this consider a vector $\Delta = (\mathcal{R}\Omega, \mathbf{v}) \in T_{\boldsymbol{\theta}}SE(3)$ and the inner product

$$\begin{aligned}
\langle \Delta, II(\Delta_1, \Delta_2)\rangle &= \begin{bmatrix}\mathrm{vec}(\mathcal{R}\Omega))^T & \mathbf{v}^T\end{bmatrix} \begin{bmatrix}-\mathrm{vec}(\mathcal{R}\,\mathrm{symm}(\Omega_1^T\Omega_2)) \\ 0\end{bmatrix} \\
&= -\mathrm{vec}(\mathcal{R}\Omega)^T \mathrm{vec}(\mathcal{R}\,\mathrm{symm}(\Omega_1^T\Omega_2)) && (4.28)\\
&= -\mathrm{tr}\left(\Omega^T\mathcal{R}^T\mathcal{R}\,\mathrm{symm}(\Omega_1^T\Omega_2)\right) && (4.29)\\
&= -\mathrm{tr}\left(\Omega^T\,\mathrm{symm}(\Omega_1^T\Omega_2)\right) && (4.30)\\
&= 0, && (4.31)
\end{aligned}$$

since given a skew-symmetric matrix Ω and a symmetric matrix S, we have $\mathrm{tr}\left(\Omega^T S\right) = 0$. This can be shown by noticing that

$$\mathrm{tr}\left(\Omega^T S\right) = \mathrm{tr}\left(S\Omega^T\right) = \mathrm{tr}\left(-S\Omega\right) = \mathrm{tr}\left(-\Omega^T S\right) = -\mathrm{tr}\left(\Omega^T S\right), \tag{4.32}$$

since zero is the only number equal to minus itself.

4.3.4 Intrinsic Gradient and Newton algorithms

We now have almost all the tools to derive a generalized gradient and Newton descent methods to minimize the ML-R and ML-SR cost functions on $SE(3)$. The geometric descent optimization algorithms proposed are of iterative nature. At each iteration a gradient or Newton-like search direction $\mathcal{D}_k$ is determined. Then, an intrinsic geodesic line search is performed along $\mathcal{D}_k$ to update the estimated parameter, see Figure 4.7.

Intrinsic gradient direction

According to Proposition 4.3.1, the intrinsic gradient of a smooth function defined on an embedded Riemannian sub-manifold of some Euclidean space can be found by

determining the extrinsic gradient (which is the usual vector of partial derivatives as if the function was defined on the ambient space $\widetilde{f} : \mathbb{R}^{12} \to \mathbb{R}$, instead of $f : SE(3) \to \mathbb{R}$) and projecting it orthogonally onto the tangent space to the constraint surface. We recall that the projection must be done according to the Riemannian metric of the ambient space which is induced on the constraint set. Since in our case we have the usual Euclidean metric on the ambient space, the projection is the usual orthogonal projection.

The derivation of the extrinsic gradients of the ML-R and ML-SR cost functions is a matrix differential calculus exercise and is done with detail in appendix A. We only give here the main results.

Proposition 4.3.3 (Extrinsic gradient of ML-R cost function)**.** *The extrinsic gradient of the ML-R cost function defined in (4.11) evaluated at the point* $\boldsymbol{\theta} = (\mathcal{R}, \mathbf{p}) \in SE(3)$ *is given by*

$$\nabla \widetilde{f}(\boldsymbol{\theta}) := \begin{bmatrix} \mathrm{vec}(\widetilde{G}_{\mathcal{R}}) \\ \widetilde{G}_{\mathbf{p}} \end{bmatrix} := - \begin{bmatrix} \mathbf{F}^T \delta(\mathbf{r}(\boldsymbol{\theta}))^{-1} \mathbf{R}^{-1} \left(\bar{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta}) \right) \\ \mathbf{C}^T \delta(\mathbf{r}(\boldsymbol{\theta}))^{-1} \mathbf{R}^{-1} \left(\bar{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta}) \right) \end{bmatrix} \in \mathbb{R}^{12}, \tag{4.33}$$

where

$$\mathbf{F} := \begin{bmatrix} \mathbf{B}^T \otimes (\mathbf{p} - \mathbf{p}_1)^T \\ \vdots \\ \mathbf{B}^T \otimes (\mathbf{p} - \mathbf{p}_m)^T \end{bmatrix} \in \mathbb{R}^{mp \times 9}, \tag{4.34}$$

$$\mathbf{C} := \mathbf{1}_m \otimes \mathbf{B}^T \mathcal{R} + (\mathbf{1}_m \mathbf{p}^T - \mathbf{P}^T) \otimes \mathbf{1}_p \in \mathbb{R}^{mp \times 3}, \tag{4.35}$$

and where matrices $\mathbf{B} = [\mathbf{b}_1 \dots \mathbf{b}_p] \in \mathbb{R}^{3 \times p}$ *and* $\mathbf{P} = [\mathbf{p}_1 \dots \mathbf{p}_m] \in \mathbb{R}^{3 \times m}$ *contain the beacon and landmark coordinates in their respective reference frames.*

Proof. See Appendix A.3.1. □

Proposition 4.3.4 (Extrinsic gradient of ML-SR cost function)**.** *The extrinsic gradient of the ML-SR cost function defined in (4.12) evaluated at the point* $\boldsymbol{\theta} = (\mathcal{R}, \mathbf{p}) \in SE(3)$ *is given by*

$$\nabla \widetilde{f}(\boldsymbol{\theta}) := \begin{bmatrix} \mathrm{vec}(\widetilde{G}_{\mathcal{R}}) \\ \widetilde{G}_{\mathbf{p}} \end{bmatrix} := -2 \begin{bmatrix} \mathbf{F}^T \boldsymbol{\Sigma}^{-1} \left(\bar{\mathbf{d}} - \mathbf{d}(\boldsymbol{\theta}) \right) \\ \mathbf{C}^T \boldsymbol{\Sigma}^{-1} \left(\bar{\mathbf{d}} - \mathbf{d}(\boldsymbol{\theta}) \right) \end{bmatrix} \in \mathbb{R}^{12}, \tag{4.36}$$

where matrices $\mathbf{F}$ *and* $\mathbf{C}$ *are given by (4.34) - (4.35).*

Proof. See Appendix A.3.2. □

Proposition 4.3.5 (Intrinsic gradient of smooth functions on SE(3))**.** *Let* $f : SE(3) \to \mathbb{R}$ *be a smooth function defined on the Special Euclidean group and* $\widetilde{f} : \mathbb{R}^{12} \to \mathbb{R}$ *its extension to* $\mathbb{R}^{12}$*. Let* $\nabla \widetilde{f}(\boldsymbol{\theta}) = [\mathrm{vec}(\widetilde{G}_{\mathcal{R}})^T \ \widetilde{G}_{\mathbf{p}}^T]^T \in \mathbb{R}^{12}$ *denote the*

extrinsic gradient of f evaluated at $\boldsymbol{\theta} = (\mathcal{R}, \mathbf{p}) \in SE(3)$. The intrinsic gradient of f evaluated at $\boldsymbol{\theta} = (\mathcal{R}, \mathbf{p}) \in SE(3)$ is given by

$$\operatorname{grad} f(\boldsymbol{\theta}) := \begin{bmatrix} \operatorname{vec}(G_{\mathcal{R}}) \\ G_{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\operatorname{vec}(\widetilde{G}_{\mathcal{R}} - \mathcal{R}\widetilde{G}_{\mathcal{R}}^T\mathcal{R}) \\ \widetilde{G}_{\mathbf{p}} \end{bmatrix} \in T_{\boldsymbol{\theta}}SE(3). \tag{4.37}$$

Proof. According to Proposition 4.3.1, the intrinsic gradient can be found by solving the projection problem

$$\begin{aligned} \operatorname{grad} f(\boldsymbol{\theta}) &= \arg\min_{\Delta \in T_{\boldsymbol{\theta}}SE(3)} \langle \nabla \widetilde{f}(\boldsymbol{\theta}) - \Delta, \nabla \widetilde{f}(\boldsymbol{\theta}) - \Delta \rangle \\ &= \arg\min_{\Delta \in T_{\boldsymbol{\theta}}SE(3)} \|\nabla \widetilde{f}(\boldsymbol{\theta}) - \Delta\|^2 \\ &= \arg\min_{\Omega \in \mathsf{K}(3,\mathbb{R}), \mathbf{v} \in \mathbb{R}^3} \| \begin{bmatrix} \operatorname{vec}(\widetilde{G}_{\mathcal{R}}) \\ \widetilde{G}_{\mathbf{p}} \end{bmatrix} - \begin{bmatrix} \operatorname{vec}(\mathcal{R}\Omega) \\ \mathbf{v} \end{bmatrix} \|^2 \\ &= \arg\min_{\Omega \in \mathsf{K}(3,\mathbb{R}), \mathbf{v} \in \mathbb{R}^3} \|\operatorname{vec}(\widetilde{G}_{\mathcal{R}}) - \operatorname{vec}(\mathcal{R}\Omega)\|^2 + \|\widetilde{G}_{\mathbf{p}} - \mathbf{v}\|^2, \end{aligned}$$

where it was used that $\langle , \rangle$ is the induced Riemannian metric on $SE(3)$ inherited from $\mathbb{R}^n$. The tangent vector $\Delta \in T_{\boldsymbol{\theta}}SE(3)$ must have the form $\begin{bmatrix} \operatorname{vec}(\mathcal{R}\Omega)^T & \mathbf{v}^T \end{bmatrix}^T$ for some skew symmetric matrix $\Omega \in \mathsf{K}(3,\mathbb{R})$ and for some vector $\mathbf{v} \in \mathbb{R}^3$. Since $\widetilde{G}_{\mathbf{p}}$ is already a tangent vector to $\mathbb{R}^3$ it is only necessary to project the first component $\widetilde{G}_{\mathcal{R}}$ on the tangent space $T_{\mathcal{R}}SO(3) = \{\mathcal{R}\Omega \,:\, \Omega \in \mathsf{K}(3,\mathbb{R})\}$. This is equivalent to solve

$$\begin{aligned} \Omega^* &= \arg\min_{\Omega \in \mathsf{K}(3,\mathbb{R})} \|\widetilde{G}_{\mathcal{R}} - \mathcal{R}\Omega\|_F^2 \\ &= \arg\min_{\Omega \in \mathsf{K}(3,\mathbb{R})} \|\mathcal{R}^T\widetilde{G}_{\mathcal{R}} - \Omega\|_F^2 \\ &= \operatorname{skew}(\mathcal{R}^T\widetilde{G}_{\mathcal{R}}) := \frac{1}{2}(\mathcal{R}^T\widetilde{G}_{\mathcal{R}} - \widetilde{G}_{\mathcal{R}}^T\mathcal{R}). \end{aligned}$$

Using this yields the desired result $\operatorname{grad} f(\boldsymbol{\theta}) = (G_{\mathcal{R}}, G_{\mathbf{p}})$ where $\widetilde{G}_{\mathbf{p}} = G_{\mathbf{p}}$ and

$$G_{\mathcal{R}} := \mathcal{R}\Omega^* = \frac{1}{2}(\widetilde{G}_{\mathcal{R}} - \mathcal{R}\widetilde{G}_{\mathcal{R}}^T\mathcal{R}).$$

(See [57] for a similar result). □

Intrinsic Newton direction

In this section it is shown how to define an intrinsic Newton search direction from the intrinsic Hessian described in Proposition 4.3.2. The derivation of the extrinsic Hessian (the usual matrix of second order partial derivatives) is done in detail in A.

Proposition 4.3.6 (Extrinsic Hessian of ML-R cost function)**.** *The extrinsic Hessian of the ML-R cost function in (4.11) evaluated at a point $\boldsymbol{\theta} = (\mathcal{R}, \mathbf{p}) \in SE(3)$ is a map $\operatorname{Hess}\widetilde{f} \,:\, T_{\boldsymbol{\theta}}SE(3) \times T_{\boldsymbol{\theta}}SE(3) \to \mathbb{R}$. Given two tangent vectors $\Delta_1, \Delta_2 \in$*

$T_{\boldsymbol{\theta}}SE(3)$ in the form $\Delta_1 = \left[\text{vec}(\mathcal{R}\Omega_1)^T \ \mathbf{v}_1^T\right]^T$ and $\Delta_2 = \left[\text{vec}(\mathcal{R}\Omega_2)^T \ \mathbf{v}_2^T\right]^T$ it can be written as

$$\text{Hess}\widetilde{f}(\Delta_1, \Delta_2) = \Delta_1^T \mathbf{H} \Delta_2, \tag{4.38}$$

where $\mathbf{H} \in \mathbb{R}^{12\times 12}$ is the matrix of second order partial derivatives given by

$$\mathbf{H} = \begin{bmatrix} \mathbf{F}^T\mathbf{W}\mathbf{F} & \mathbf{F}^T\mathbf{W}\mathbf{C} \\ \mathbf{C}^T\mathbf{W}\mathbf{F} & \mathbf{C}^T\mathbf{W}\mathbf{C} \end{bmatrix} - \sum_{i,j=1}^{p,m} \alpha_{ij}\, \mathcal{H}_{ij}. \tag{4.39}$$

In the previous expression, $\mathbf{W} = \delta(\mathbf{r}(\boldsymbol{\theta}))^{-1}\mathbf{R}^{-1}\delta(\mathbf{r}(\boldsymbol{\theta}))^{-1} \in \mathbb{R}^{mp\times mp}$, $\mathbf{F}$ and $\mathbf{C}$ are given by (4.34) - (4.35), and $\alpha_{ij} \in \mathbb{R}$ are the entries of the vector

$$\boldsymbol{\alpha} := [\alpha_{11} \ \cdots \ \alpha_{mp}]^T = \mathbf{R}^{-1}(\bar{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta})) \in \mathbb{R}^{mp}, \tag{4.40}$$

and each of the matrices $\mathcal{H}_{ij}$ has the form

$$\mathcal{H}_{ij} = \begin{bmatrix} \mathcal{H}_{ij}^{11} & \mathcal{H}_{ij}^{21T} \\ \mathcal{H}_{ij}^{21} & \mathcal{H}_{ij}^{22} \end{bmatrix} \in \mathbb{R}^{12\times 12}, \tag{4.41}$$

with

$$\mathcal{H}_{ij}^{11} = -\frac{1}{r_{ij}^3}(\mathbf{b}_i \otimes \mathbf{I}_3)(\mathbf{p} - \mathbf{p}_j)(\mathbf{p} - \mathbf{p}_j)^T(\mathbf{b}_i^T \otimes \mathbf{I}_3) \in \mathbb{R}^{9\times 9}, \tag{4.42}$$

$$\begin{aligned} \mathcal{H}_{ij}^{21} = -\frac{1}{r_{ij}^3}\big\{&(\mathbf{p} - \mathbf{p}_j)(\mathbf{p} - \mathbf{p}_j)^T(\mathbf{b}_i^T \otimes \mathbf{I}_3) \\ &+\mathcal{R}\mathbf{b}_i(\mathbf{p} - \mathbf{p}_j)^T(\mathbf{b}_i^T \otimes \mathbf{I}_3)\big\} + \frac{1}{r_{ij}}(\mathbf{b}_i^T \otimes \mathbf{I}_3) \in \mathbb{R}^{3\times 9}, \end{aligned} \tag{4.43}$$

$$\begin{aligned} \mathcal{H}_{ij}^{22} = -\frac{1}{r_{ij}^3}\big\{&(\mathbf{p} - \mathbf{p}_j)(\mathbf{p} - \mathbf{p}_j)^T + \mathcal{R}\mathbf{b}_i(\mathbf{p} - \mathbf{p}_j)^T \\ &+(\mathbf{p} - \mathbf{p}_j)\mathbf{b}_i^T\mathcal{R}^T + \mathcal{R}\mathbf{b}_i\mathbf{b}_i^T\mathcal{R}^T\big\} + \frac{1}{r_{ij}}\mathbf{I}_3 \in \mathbb{R}^{3\times 3}. \end{aligned} \tag{4.44}$$

Proposition 4.3.7 (Extrinsic Hessian of the ML-SR cost function)**.** *The extrinsic Hessian of the ML-SR cost function in (4.12) evaluated at a point $\boldsymbol{\theta} = (\mathcal{R}, \mathbf{p}) \in SE(3)$ is a map $\text{Hess}\widetilde{f} : T_{\boldsymbol{\theta}}SE(3) \times T_{\boldsymbol{\theta}}SE(3) \to \mathbb{R}$. Given two tangent vectors $\Delta_1, \Delta_2 \in T_{\boldsymbol{\theta}}SE(3)$ in the form $\Delta_1 = \left[\text{vec}(\mathcal{R}\Omega_1)^T \ \mathbf{v}_1^T\right]^T$ and $\Delta_2 = \left[\text{vec}(\mathcal{R}\Omega_2)^T \ \mathbf{v}_2^T\right]^T$ it can be written as*

$$\text{Hess}\widetilde{f}(\Delta_1, \Delta_2) = \Delta_1^T \mathbf{H} \Delta_2, \tag{4.45}$$

where $\mathbf{H} \in \mathbb{R}^{12\times 12}$ is the matrix of second order partial derivatives given by

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{21}^T \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} \in \mathbb{R}^{12\times 12}, \tag{4.46}$$

and

$$\mathbf{H}_{11} = 4\mathbf{F}^T\boldsymbol{\Sigma}^{-1}\mathbf{F}, \tag{4.47}$$

$$\mathbf{H}_{21} = 4\mathbf{C}^T\boldsymbol{\Sigma}^{-1}\mathbf{F} - 2\Big((\bar{\mathbf{d}} - \mathbf{d}(\boldsymbol{\theta}))^T\boldsymbol{\Sigma}^{-1} \otimes \mathbf{I}_3\Big)\Big(\mathbf{1}_m \otimes \mathbf{B}^T \otimes \mathbf{I}_3\Big), \tag{4.48}$$

$$\mathbf{H}_{22} = 4\mathbf{C}^T\boldsymbol{\Sigma}^{-1}\mathbf{C} - 2\Big((\bar{\mathbf{d}} - \mathbf{d}(\boldsymbol{\theta}))^T\boldsymbol{\Sigma}^{-1}\mathbf{1}_{mp}\Big)\mathbf{I}_3. \tag{4.49}$$

Proof. See Appendix A.3.2. □

Proposition 4.3.8 (Intrinsic Hessian of smooth functions defined on $SE(3)$)**.** *Let $f : SE(3) \to \mathbb{R}$ be a smooth function on the Special Euclidean group and $\widetilde{f} : \mathbb{R}^{12} \to \mathbb{R}$ its extension to $\mathbb{R}^{12}$. Let $\nabla\widetilde{f}(\boldsymbol{\theta}) = [\mathrm{vec}(\widetilde{G}_{\mathcal{R}})^T \ \widetilde{G}_{\mathbf{p}}^T]^T \in \mathbb{R}^{12}$ denote the extrinsic gradient of f evaluated at $\boldsymbol{\theta} \in SE(3)$. Let $\mathbf{H} = \in \mathbb{R}^{12\times 12}$ denote the matrix of second order partial derivatives of $\widetilde{f}$ evaluated at $\boldsymbol{\theta} \in SE(3)$. Given two tangent vectors $\Delta_1, \Delta_2 \in T_{\boldsymbol{\theta}}SE(3)$ with the form $\Delta_i = \left[\mathrm{vec}(\mathcal{R}\Omega_i)^T \ \mathbf{v}_i^T\right]^T; i \in \{1, 2\}$ the intrinsic Hessian $\mathrm{Hess}f : T_{\boldsymbol{\theta}}SE(3) \times T_{\boldsymbol{\theta}}SE(3) \to \mathbb{R}$ can be computed as*

$$\mathrm{Hess}f(\Delta_1, \Delta_2) = \Delta_1^T\mathbf{H}\Delta_2 - \mathrm{tr}\left(\widetilde{G}_{\mathcal{R}}^T\mathcal{R}\,\mathrm{symm}(\Omega_1^T\Omega_2)\right). \tag{4.50}$$

Proof. The result follows from Proposition 4.3.2 and the expression for the second fundamental form on $SE(3)$ given by (4.27). □

Once the intrinsic Hessian of the ML-R and ML-SR cost functions has been found, one can determine the generalized intrinsic Newton directions associated to those cost functions. In order to do so we will follow the methodology introduced in section 4.3.2. We first need to determine an orthonormal basis for the tangent space of $SE(3)$ at $\boldsymbol{\theta} = (\mathcal{R}, \mathbf{p})$. That is a set of orthonormal vectors $\{E_1, \ldots, E_6\}$ that span $T_{\boldsymbol{\theta}}SE(3) = \{(\mathcal{R}\Omega, \mathbf{v}) : \Omega \in \mathsf{K}(3, \mathbb{R}), \mathbf{v} \in \mathbb{R}^3\}$. To this purpose, consider an orthonormal basis for the set of skew symmetric matrices $\mathsf{K}(3, \mathbb{R})$ given by

$$\Omega_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \Omega_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \Omega_3 = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \tag{4.51}$$

and choose

$$\begin{gathered} E_1 = (\mathcal{R}\Omega_1, 0), \quad E_2 = (\mathcal{R}\Omega_2, 0), \quad E_3 = (\mathcal{R}\Omega_3, 0), \\ E_4 = (0, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}), \quad E_5 = (0, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}), \quad E_6 = (0, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}), \end{gathered} \tag{4.52}$$

which satisfy $\langle E_i, E_i\rangle = 1$ and $\langle E_i, E_j\rangle = 0$ for $i \neq j$. Assuming that the Hessian is nonsingular, the generalized Newton direction can be found by first solving the

linear system

$$\begin{bmatrix} \text{Hess}(E_1, E_1) & \dots & \text{Hess}(E_1, E_6) \\ \vdots & \ddots & \vdots \\ \text{Hess}(E_d, E_1) & \dots & \text{Hess}(E_d, E_6) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_6 \end{bmatrix} = - \begin{bmatrix} \langle E_1, \text{grad} f(\boldsymbol{\theta}) \rangle \\ \vdots \\ \langle E_6, \text{grad} f(\boldsymbol{\theta}) \rangle \end{bmatrix}. \tag{4.53}$$

Then we obtain the search direction $\mathcal{N}_{\boldsymbol{\theta}} \in T_{\boldsymbol{\theta}} SE(3)$ according to

$$\mathcal{N}_{\boldsymbol{\theta}} = \sum_{i=1}^{6} \alpha_i E_i. \tag{4.54}$$

The intrinsic Newton descent method uses at each iteration the search direction $\mathcal{D}_k = \mathcal{N}_{\boldsymbol{\theta}_k}$.

Intrinsic geodesic line search

Let $\boldsymbol{\theta}_k = (\mathcal{R}_k, \mathbf{p}_k) \in \mathsf{SE}(3)$ be the parameter estimate at iteration k. Let $\mathcal{D}_k = (\mathcal{D}_\mathcal{R}, \mathcal{D}_\mathbf{p}) \in T_{\boldsymbol{\theta}_k} SE(3)$ be the search direction at iteration k, i.e, $\mathcal{D}_k = -\text{grad} f(\boldsymbol{\theta}_k)$ (gradient descent) or $\mathcal{D}_k = \mathcal{N}_{\boldsymbol{\theta}}$ (Newton descent). A line search can be performed along the geodesic

$$\gamma_k(t) = (\mathcal{R} \exp(\mathcal{R}^T \mathcal{D}_\mathcal{R} t), \mathbf{p} + \mathcal{D}_\mathbf{p} t). \tag{4.55}$$

Ideally, the line search procedure aims at finding the optimal stepsize t_k^* satisfying

$$t_k^* = \arg\min_{t \in \mathbb{R}} f(\gamma_k(t)). \tag{4.56}$$

This optimization subproblem can be hard to solve. Alternatively, it is common to obtain an approximate solution to this problem only, by using for instance the Armijo rule [29, p.29]. The Armijo rule selects $t_k = \beta^{m_i} s$, where $m_i \in \{0, 1, 2, \dots\}$ is the first integer satisfying

$$f(\gamma_k(\beta^{m_i} s)) \leq f(\boldsymbol{\theta}_k) + \sigma \beta^{m_i} s \, \langle \text{grad} f(\boldsymbol{\theta}_k), \mathcal{D}_k \rangle, \tag{4.57}$$

for some constants $s > 0$, and $\beta, \sigma \in (0, 1)$.

Algorithms implementation outline

There are some important issues common to Newton like optimization methods that should be addressed. For instance the linear system of (4.53) might be very bad conditioned in such a case we can not rely on its solution. Another important issue is that the obtained search direction N might not be of descent! That is, the cost function might not decrease as we move along the search direction infinitesimally. This can be easily checked since in such a case the inner product of the search direction

1. *Start at initial estimate* $\boldsymbol{\theta}_0 = (\mathcal{R}_0, \mathbf{p}_0) \in SE(3)$. *Set* $k = 0$.

2. *Determine a search direction* $\mathcal{D}_k = (\mathcal{D}_\mathcal{R}, \mathcal{D}_\mathbf{p}) \in T_{\boldsymbol{\theta}_k} SE(3)$.

   (a) *Intrinsic gradient: Determine* $\mathcal{D}_k = -\text{grad} f(\boldsymbol{\theta}_k)$ *using Proposition 4.3.5 and the extrinsic gradients in (4.33) and (4.36).*

   (b) *Intrinsic Newton: Determine the generalized Newton direction* $\mathcal{N}$ *according to Proposition 4.3.8, the ML-R and ML-SR extrinsic Hessians in Propositions 4.3.6 - 4.3.7 and solving the linear system (4.54). If the system (4.53) has a unique solution and* $\langle \mathcal{N}, \text{grad} f(\boldsymbol{\theta}_k) \rangle < 0$ *(descent condition) take* $\mathcal{D}_k = \mathcal{N}$*, otherwise take* $\mathcal{D}_k = -\text{grad} f(\boldsymbol{\theta}_k)$ *(safe gradient descent step).*

3. *Line search along the geodesic*

$$\gamma_k(t) = (\mathcal{R} \exp(\mathcal{R}^T \mathcal{D}_\mathcal{R} t), \mathbf{p} + \mathcal{D}_\mathbf{p} t) \tag{4.58}$$

   using Armijo rule (4.57) to determine step size t_k.

4. *update estimate* $\boldsymbol{\theta}_{k+1} = \gamma_k(t_k)$. *Set* $k = k + 1$.

5. *If* $\|\text{grad} f(\boldsymbol{\theta}_k)\| \leq \epsilon$ *or* $k \geq k_{max}$*, stop. Otherwise return to 2.*

Table 4.1: Geometric descent optimization of ML-R and ML-SR cost functions.

and the intrinsic gradient will not be negative, the descent condition. Whenever one of these problems arises we can force the algorithm to perform a negative gradient descent step.

We are now ready to summarize the generalized gradient and Newton descent methods to minimize the ML-R and ML-SR cost functions on the Special Euclidean group $SE(3)$.

4.3.5 Initialization

The above algorithms need an initial attitude and position estimate $\boldsymbol{\theta}_0 = (\mathcal{R}_0, \mathbf{p}_0) \in SE(3)$ in order to start. Unfortunately there are no convergence warranties since both the ML-R and ML-SR cost functions exhibit local minimums that could attract the iterative algorithms. Numerical simulations revealed that the algorithms did not converge to local minimums when the initial estimates were close to the actual rigid body attitude and position. We will next show a simple three step initialization procedure that allows us to compute an initial estimate quite close to the actual attitude and position.

Determine inertial beacon positions by LS trilateration

Determine the inertial beacon positions (independently) using the range equations. For each beacon $i \in \{1, \dots, p\}$ determine ${}^{\mathcal{I}}\widehat{\mathbf{b}}_i$ which is an estimate of the inertial beacon position ${}^{\mathcal{I}}\mathbf{b}_i = \mathcal{R}\mathbf{b}_i + \mathbf{p}$. This can be done for instance using any Range-Only localization algorithm using the ranges between the i beacon and all the landmarks. This can be done in a simple way using the LS-C trilateration algorithm (2.32) as follows. Consider the matrix containing the square range measurements between the beacons and the landmarks

$$\bar{\mathbf{D}} = \begin{bmatrix} \bar{d}_{11} & \dots & \bar{d}_{1m} \\ \dots & \ddots & \vdots \\ \bar{d}_{p1} & \dots & \bar{d}_{pm} \end{bmatrix} \in \mathbb{R}^{pm}, \tag{4.59}$$

where $\bar{d}_{ij} = \bar{r}_{ij}^2$, $\bar{r}_{ij} = r_{ij} + w_{ij}$, and $r_{ij} = \|{}^{\mathcal{I}}\mathbf{b}_i - \mathbf{p}_j\| = \|\mathcal{R}\mathbf{b}_i + \mathbf{p} - \mathbf{p}_j\|$ as defined in (4.3). Note that by simplicity in the previous sections we omitted the superscripts, and wrote $\mathbf{b}_i, \mathbf{p}_j$ to denote ${}^{\mathcal{B}}\mathbf{b}_i$ and ${}^{\mathcal{I}}\mathbf{p}_j$ respectively. That is, the beacon coordinates were expressed with respect to the body fixed reference frame $\{\mathcal{B}\}$ and the landmark coordinates were expressed with respect to the Earth fixed reference frame $\{\mathcal{I}\}$. Then one can determine in a simple way all the inertial beacon positions using the LS trilateration algorithm by

$${}^{\mathcal{I}}\widehat{\mathbf{B}} := \left[{}^{\mathcal{I}}\widehat{\mathbf{b}}_1 \ \dots \ {}^{\mathcal{I}}\widehat{\mathbf{b}}_p \right] = \boldsymbol{\Theta} \left(\delta(\mathbf{P}^T\mathbf{P})\mathbf{1}_p^T - \bar{\mathbf{D}}, \right) \tag{4.60}$$

where $\boldsymbol{\Theta} = 1/2(\mathbf{P}_c\mathbf{P}_c^T)^{-1}\mathbf{P}_c$ and $\mathbf{P}$ is the matrix containing the landmark inertial coordinates.

Determine attitude by solving orthogonal Procrustes problem

Define the centering matrix $\mathbf{M}_p = \mathbf{I}_p - \frac{1}{p}\mathbf{1}_p\mathbf{1}_p^T \in \mathbb{R}^{p\times p}$, and matrices

$${}^{\mathcal{I}}\widehat{\mathbf{B}}_c = {}^{\mathcal{I}}\widehat{\mathbf{B}}\mathbf{M}_p \in \mathbb{R}^{3\times p}, \quad \mathbf{B}_c = \mathbf{B}\mathbf{M}_p \in \mathbb{R}^{3\times p}. \tag{4.61}$$

Determine the rotation matrix $\widehat{\mathcal{R}}$ that minimizes

$$\widehat{\mathcal{R}} = \arg \min_{\mathcal{R}\in SO(3)} \|{}^{\mathcal{I}}\widehat{\mathbf{B}}_c - \mathcal{R}\mathbf{B}_c\|_F^2. \tag{4.62}$$

This is a classic Orthogonal Procrustes problem, also referred as Wahba's problem, [184], [162], [74], [83]. There is a simple closed form solution to this problem based on SVD decomposition. Define matrix $\mathbf{G} = {}^{\mathcal{I}}\widehat{\mathbf{B}}_c\mathbf{B}_c^T = {}^{\mathcal{I}}\widehat{\mathbf{B}}\mathbf{M}_p\mathbf{B}^T$ which admitsthe SVD decomposition $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$. Then the solution to the orthogonal Procrustes problem is

given by

$$\widehat{\mathcal{R}} = \mathbf{U}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & s\end{bmatrix}\mathbf{V}^T, \quad s = \det(\mathbf{U}\mathbf{V}^T) = \pm 1. \tag{4.63}$$

Determine estimated position

The inertial beacon coordinates satisfy ${}^{\mathcal{I}}\mathbf{b}_i = \mathcal{R}\mathbf{b}_i + \mathbf{p}$ where $(\mathcal{R}, \mathbf{p})$ is the rigid body actual attitude and position. Because of the errors produced in the estimated inertial beacon positions and rigid body, we have that for each beacon, ${}^{\mathcal{I}}\widehat{\mathbf{b}}_i = \widehat{\mathcal{R}}\mathbf{b}_i + \mathbf{p} + \mathbf{e}_i$ where $\mathbf{e}_i$ s some error vector. We can determine the vehicle position that minimizes the sum of the errors

$$\widehat{\mathbf{p}} = \frac{1}{p}\sum_{i=1}^{p}({}^{\mathcal{I}}\widehat{\mathbf{b}}_i - \widehat{\mathcal{R}}\mathbf{b}_i) = \frac{1}{p}({}^{\mathcal{I}}\widehat{\mathbf{B}} - \widehat{\mathcal{R}}\mathbf{B})\mathbf{1}_p. \tag{4.64}$$

Let $\mathbf{P} = [\mathbf{p}_1 \ \dots \mathbf{p}_m] \in \mathbb{R}^{3\times m}$ *be the matrix containing landmark coordinates expressed in the inertial reference frame* $\{\mathcal{I}\}$. *Let* $\mathbf{B} = [\mathbf{b}_1 \ \dots \mathbf{b}_p] \in \mathbb{R}^{3\times p}$ *be the matrix containing the beacon coordinates expressed in the body reference frame* $\{\mathcal{B}\}$. *Define the centering matrices* $\mathbf{M}_k = \mathbf{I}_k - \frac{1}{k}\mathbf{1}_k\mathbf{1}_k^T \in \mathbb{R}^{k\times k}$ *with* $k \in \{m, p\}$. *Let* $\bar{\mathbf{D}} = [\bar{d}_{ij}] \in \mathbb{R}^{m\times p}$ *be the matrix containing the squared range measurements.*

1. *Compute inertial beacon coordinate matrix using LS trilateration* ${}^{\mathcal{I}}\widehat{\mathbf{B}} = \boldsymbol{\Theta}\left(\delta(\mathbf{P}^T\mathbf{P})\mathbf{1}_p^T - \bar{\mathbf{D}}\right)$, *where* $\boldsymbol{\Theta} = 1/2(\mathbf{P}_c\mathbf{P}_c^T)^{-1}\mathbf{P}_c$ *and* $\mathbf{P}_c = \mathbf{P}\mathbf{M}_m$.
2. *Compute estimated rigid body attitude* $\widehat{\mathcal{R}} = \mathbf{U}\left[\begin{smallmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & s\end{smallmatrix}\right]\mathbf{V}^T$, *where* $s = \det(\mathbf{U}\mathbf{V}^T)$ *and* $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ *is the SVD decomposition of* $\mathbf{G} = {}^{\mathcal{I}}\widehat{\mathbf{B}}\mathbf{M}_p\mathbf{B}^T$.
3. *Compute the estimated rigid body position* $\widehat{\mathbf{p}} = \frac{1}{p}({}^{\mathcal{I}}\widehat{\mathbf{B}} - \widehat{\mathcal{R}}\mathbf{B})\mathbf{1}_p$.

Table 4.2: Simple initialization procedure to determine an attitude and position pair $(\widehat{\mathcal{R}}, \widehat{\mathbf{p}})$ based on a squared range measurement matrix $\bar{\mathbf{D}}$.

4.3.6 Simulation Results

In this section we will present simulation results of the geometric descent algorithms to solve the range-only attitude and positioning problem. The algorithms implemented corresponded to the Gradient descent and Newton optimization of the ML-R and ML-SR cost functions defined in (4.11)-(4.12) and summarized in Table 4.1. The estimation problem setup is shown in Fig. 4.11. A rigid body equipped with $p = 4$ beacons is located at a random position and attitude. The ranges from $m = 4$ landmarks are measured with an independent Gaussian error of 10cm standard deviation. That is, a set of noisy range measurements $\bar{\mathbf{r}} = \mathbf{r} + \mathbf{w} \in \mathbb{R}^{16}$ is

generated where $\mathbf{r}$ contains the actual range measurements and $\mathbf{w}$ is a Gaussian vector with covariance $\mathbf{R} = (0.1)^2\mathbf{I}_{16}$. The initial estimate of the rigid body attitude and position is also shown in Fig. 4.11.

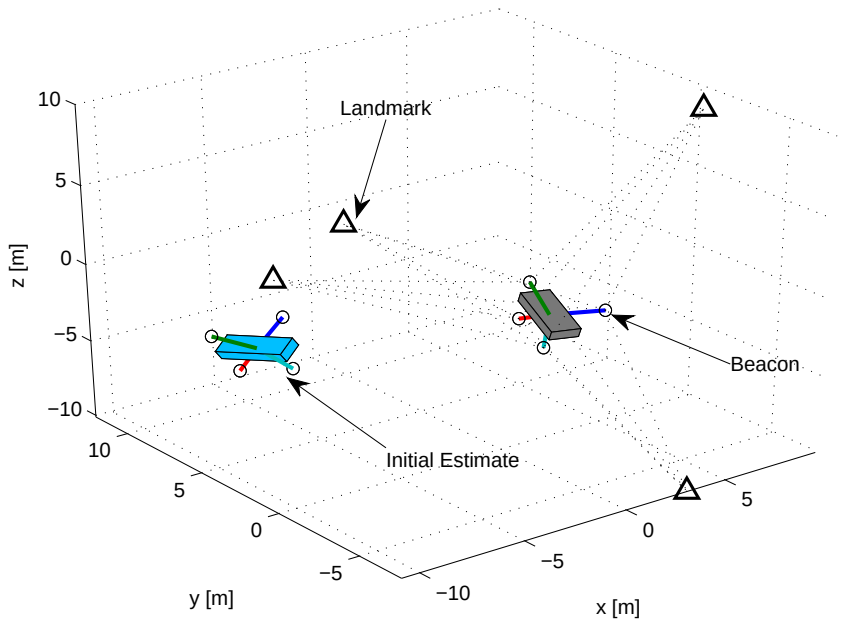


Figure 4.11: Simulation setup. Actual and initial estimated rigid body configurations together with beacon and landmark locations.

The Gradient descent and Newton algorithms to minimize the ML-R and ML-SR cost functions were implemented in MATLAB. The weighting matrix of the ML-SR cost function was chosen as $\mathbf{\Sigma} = 4\delta(\bar{\mathbf{r}})\mathbf{R}\delta(\bar{\mathbf{r}})$. The algorithms were implemented following Table 4.1 with parameters $\epsilon = 10e-6$ and $k_{max} = 20$. The Armijo's rule parameters introduced in (4.57) were taken as $\beta = 0.2$, $s = 1$, and $\sigma = 0.1$.

Fig. 4.12 shows the time evolution of the position estimation errors of the different algorithms. The initial condition and the vector of noisy range observations was exactly the same for all the algorithms in order to allow a better comparison. The figure depicts the entries of the position estimation error vector $\widetilde{\mathbf{p}}_k = \mathbf{p} - \widehat{\mathbf{p}}_k$ where $\mathbf{p}$ denotes the actual rigid body position and $\widehat{\mathbf{p}}_k$ the estimated position at iteration k. Fig. 4.13 shows the time evolution of the attitude estimation errors for the different algorithms. In order to plot the attitude estimation error, exponential coordinates of the error rotation $\mathcal{R}_{ek} = \widehat{\mathcal{R}}_k^T\mathcal{R}$ are used, i.e., the error rotation is parametrized by vector $\boldsymbol{\theta} = [\theta_1\ \theta_2\ \theta_3]^T$ where $\mathcal{R}_{ek} = \exp(\mathcal{S}(\boldsymbol{\theta}))$. Note that this is done only for visualization purposes and that no particular parametrization of the Special Orthogonal group $SO(3)$ is used elsewhere in the present work.

Fig. 4.14 and Fig. 4.15 illustrate some internal parameters of the iterative minimization algorithms. Namely, the value of the cost functions (ML-R or ML-SR)

and the value of the gradient of the cost function, which serves as a stop for the algorithms. Note that the Newton methods converge much faster than the gradient descent ones, and that close to the minimum they exhibit quadratic convergence. The gradient algorithms, however, exhibit only linear convergence near the minimum.

Fig. 4.16 shows the time evolution of the intrinsic estimation error. That is, the Riemannian distance between the actual and estimated rigid body configurations as defined in (4.24). Note that the final values of the ML-R and ML-SR optimization algorithms are slightly different, meaning that the minimums of the ML-R and ML-SR cost functions do not coincide. In theory, one is more interested in optimizing the ML-R since it coincides with the Maximum Likelihood criteria. However, the ML-SR algorithm is relatively simpler than the ML-R, in the sense that its gradient and Hessian has simpler expressions (for instance compare prop. 4.3.6 with prop. 4.3.7). This makes the ML-SR a slightly more convenient choice from the implementation point of view. The price to pay is that the solutions of the ML-R and ML-SR optimization problems do not coincide. Later on , while discussing about performance bounds, we will try to quantify this difference. Note also that none of the algorithms have intrinsic estimation errors converging to zero. This is absolutely normal, since these final residual estimation errors are due to the range measurement errors and are related to the size of the error covariance matrix $\mathbf{R}$. Only in the absence of measurement errors one would expect this intrinsic distances to converge to zero.

4.3.7 Conclusions

In this section we introduced some geometric descent optimization algorithms to solve the simultaneous range-only attitude and positioning problem. Two different cost functions inspired in the Maximum likelihood criteria were defined based on range and squared range measurements. Intrinsic gradient descent and Newton algorithms were derived by using tools from Riemannian geometry that exploit the non Euclidean nature of the Special Euclidean group $SE(3)$. The algorithms are relatively simple from a numerical point of view and do not require any normalization scheme since the estimates evolve naturally on $SE(3)$. Furthermore, no particular parametrization of the rotation matrices was needed, except for graphical representation purposes.

The ML-R and ML-SR cost function were found to have local mimimums that could eventually attract the solutions of the iterative algorithms. This problem can be minimized when the initial estimate is chosen sufficiently close to the true rigid body configuration. A simple three step closed form initialization algorithm was derived which provides initial estimates very close to the actual configuration, which allows the use of the iterative algorithms to refine the solution.

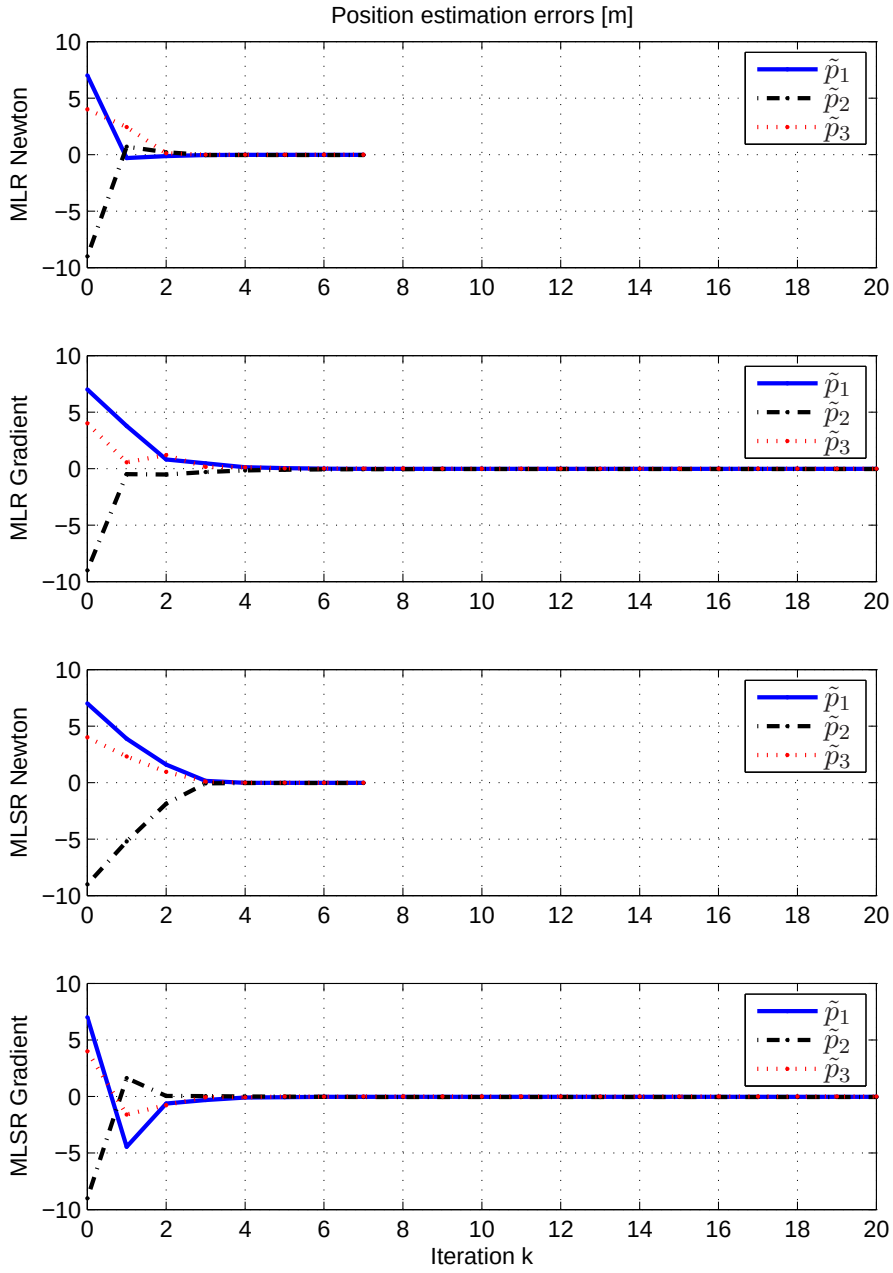


Figure 4.12: Position estimation errors of the different geometric optimization algorithms. Entries of vector $\widetilde{\mathbf{p}}_k = \mathbf{p} - \widehat{\mathbf{p}}_k = [\tilde{p}_1 \ \tilde{p}_2 \ \tilde{p}_3]^T$.

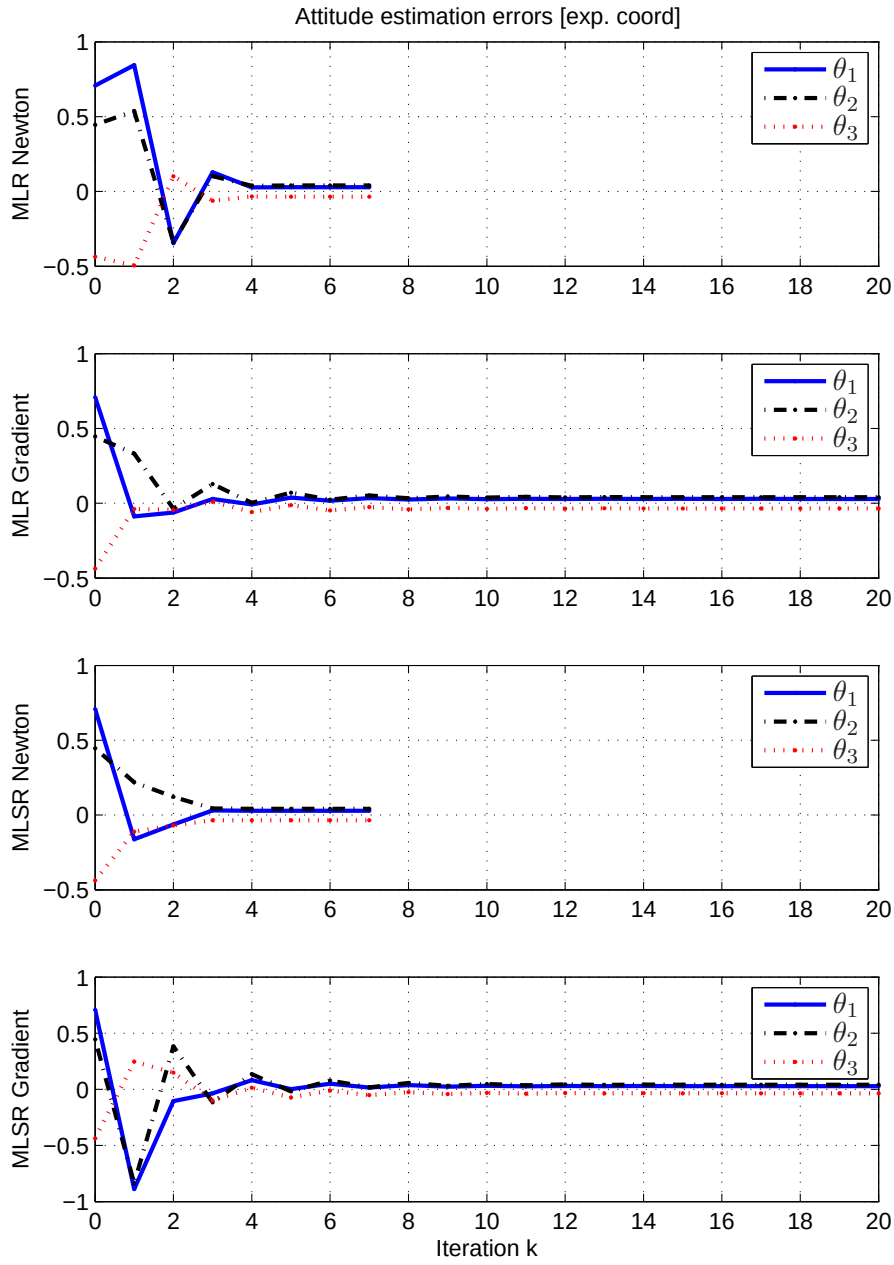


Figure 4.13: Attitude estimation errors of the different geometric optimization algorithms. Entries of vector $\boldsymbol{\theta} = [\theta_1\ \theta_2\ \theta_3]^T$, the exponential coordinates of $\mathcal{R}_{ek} = \widehat{\mathcal{R}}_k^T \mathcal{R}$, i.e., $\mathcal{R}_{ek} = \exp(\mathcal{S}(\boldsymbol{\theta}))$.

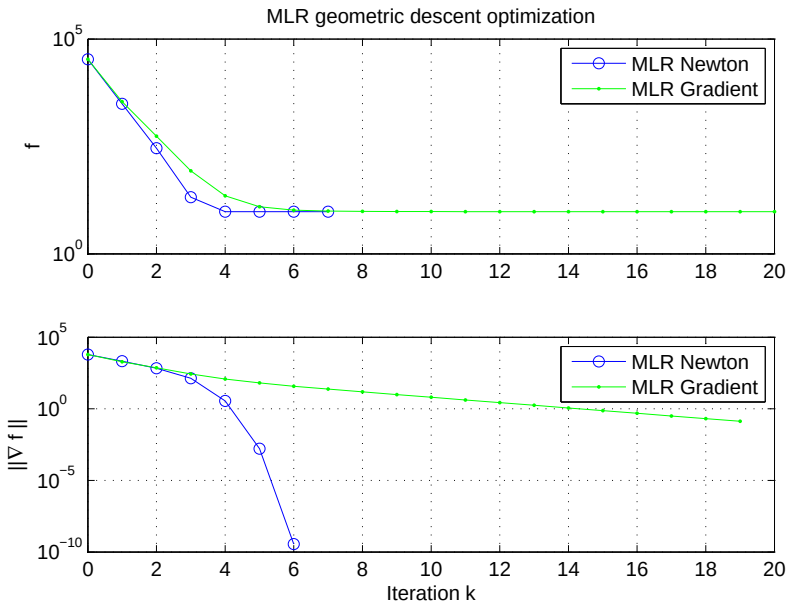


Figure 4.14: Geometric optimization of the ML-R cost function.

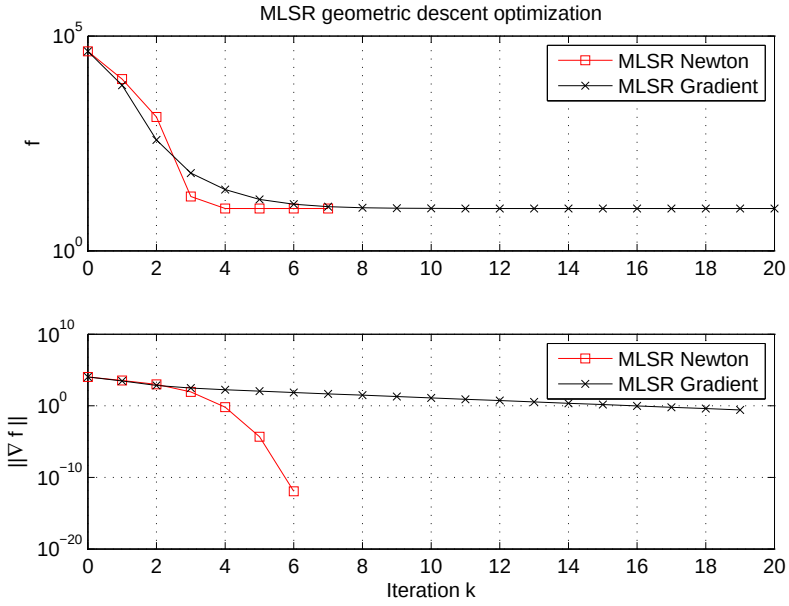


Figure 4.15: Geometric optimization of the ML-SR cost function.

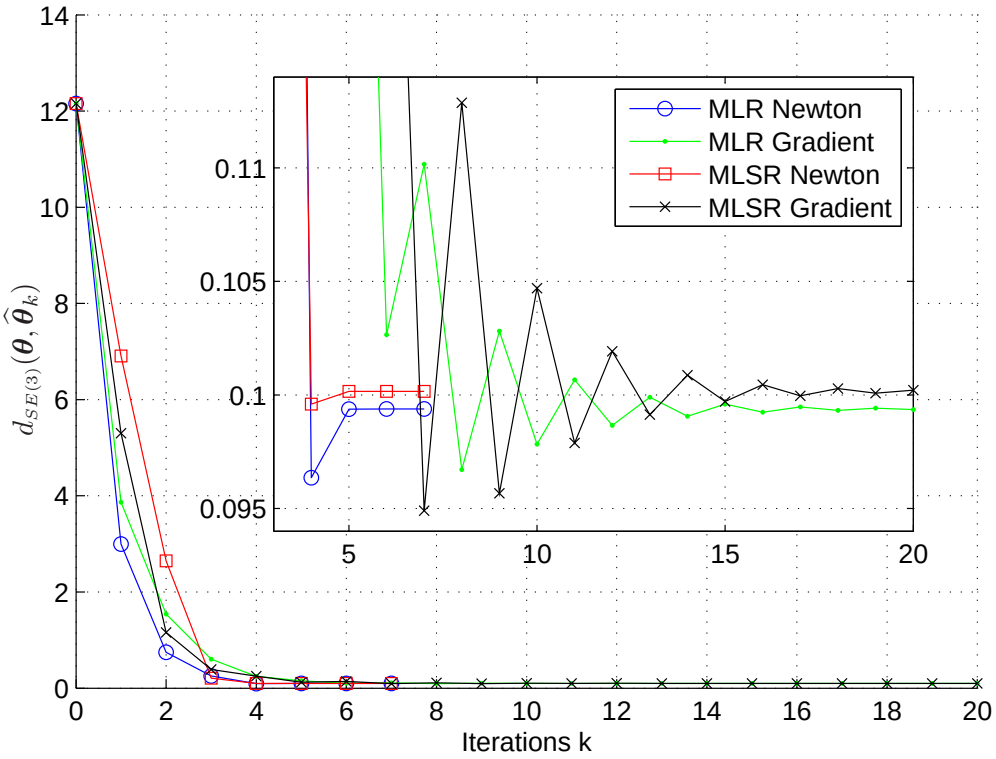


Figure 4.16: Intrinsic estimation errors. Time evolution of the Riemannian distance between the actual and estimated rigid body configuration as defined in (4.24)

.

4.4 Performance bounds

In section 2.5 we studied performance bounds of the range-only positioning problem and their important role on the design and evaluation of estimators. In this section will try to extend those ideas to the range-only attitude and positioning problem. Unlike in the range-only positioning problem, in which the parameter space was $\mathbb{R}^3$, we are now faced with a parameter space $SE(3)$ that is not Euclidean. This poses some technical difficulties and will require the use of alternative performance bounds that take this fact into account.

The Cramér-Rao Bound (CRB), introduced in section 2.5, sets a lower bound on the performance of unbiased estimators [143], [170], [95]. A part from some regularity conditions, the CRB requires the parameter space to be an open subset of an Euclidean space $\mathbb{R}^n$. That is, the parameter space $\Theta \subset \mathbb{R}^n$ must have the same n degrees of freedom as the ambient space. This condition fails if, as in our case, the parameter space is a d-dimensional (with $d < n$) Riemannian submanifold of $\mathbb{R}^n$. Moreover, the CRB bound is only valid for unbiased estimators. This condition is usually hard to proof in practice and when the parameter space is not Euclidean things get even worse.

One of the sources of difficulties is the fact that the Euclidean distance function used to define the concepts of expectation, covariance and variance, might not make sense when generalized to non Euclidean spaces. To illustrate this consider the following question: what is the expectation of a uniform distribution on the the unit circle $\Theta = \left\{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = 1\right\}$? If we use the standard definition of expectation, we would conclude that it is the origin $(0, 0)$ but this certainly does not belong to the unit circle. This does not seem to make sense as it will be natural that the expectation belongs to the parameter space.

This suggests that in order to get rid of these problems we should define expectations and measure distances in an intrinsic manner. There are recent results which extend the CRB to the case when the parameter space is not an open subset of some Euclidean space. See for instance [159], [190]. The Intrinsic Variance Lower Bound (IVLB) derived in [190] sets a lower bound on the performance of estimators in which the parameter space is a Riemannian manifold. The IVLB uses at its basis the Riemannian distance function as a way of generalizing expectations and variances. The reader is referred to [190] for further details.

Next, the IVLB for the attitude and position estimation problem with range-only measurements will be derived. The derivation is done for the case $n = 3$, but note that the results can be applied to the case $n = 2$ with some minor modifications. Given two points $\boldsymbol{\theta}_1 = (\mathcal{R}_1, \mathbf{p}_1) \in SE(3)$ and $\boldsymbol{\theta}_2 = (\mathcal{R}_2, \mathbf{p}_2) \in SE(3)$, the intrinsic (geodesic) Riemannian distance between them was defined in (4.24) and given by

[138], [126]

$$d_{SE(3)}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \sqrt{d_{SO(3)}(\mathcal{R}_1, \mathcal{R}_2)^2 + \|\mathbf{p}_1 - \mathbf{p}_2\|^2},$$

where

$$d_{SO(3)}(\mathcal{R}_1, \mathcal{R}_2) = \sqrt{2} \arccos\left(\frac{\mathrm{tr}\left(\mathcal{R}_1^T \mathcal{R}_2\right) - 1}{2}\right).$$

Let $\widehat{\boldsymbol{\theta}} : \mathbb{R}^{mp} \to SE(3)$, $\mathbf{r} \mapsto \widehat{\boldsymbol{\theta}}(\mathbf{r})$ be an unbiased attitude and position estimator taking a set of mp ranges between body fixed beacons and earth fixed landmarks and delivering a point in $SE(3)$. Let $\boldsymbol{\theta} \in SE(3)$ be the actual value of the unknown parameter. The (extrinsic) Fisher Information Matrix can be computed as

$$I(\boldsymbol{\theta}) = \mathsf{E}\left\{\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{r}) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{r})^T\right\} = \begin{bmatrix}\mathbf{F}^T \\ \mathbf{C}^T\end{bmatrix} \Big(\delta(\mathbf{r}(\boldsymbol{\theta}))\mathbf{R}\delta(\mathbf{r}(\boldsymbol{\theta}))\Big)^{-1} \begin{bmatrix}\mathbf{F}^T \\ \mathbf{C}^T\end{bmatrix}^T, \tag{4.65}$$

where we used the fact that $\log p_{\boldsymbol{\theta}}(\mathbf{r})$ is equal to the MLR cost function defined in (4.11) except for constant terms meaning that $\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{r}) = -\nabla \widetilde{f}|_{\boldsymbol{\theta}}$ as defined in (4.33).

Define a matrix $\mathbf{U}_{\boldsymbol{\theta}} \in \mathbb{R}^{12\times 6}$ whose columns form an orthonormal basis for the tangent space $T_{\boldsymbol{\theta}}SE(3)$. For instace take $\mathbf{U}_{\boldsymbol{\theta}} = [\mathrm{vec}(E_1) \ \ldots \ \mathrm{vec}(E_6)]$ where $\{E_1, \ldots, E_6\}$ are chosen as in (4.52). Define the scalar

$$\lambda_{\boldsymbol{\theta}} = \mathrm{tr}\Big((\mathbf{U}_{\boldsymbol{\theta}}^T I(\boldsymbol{\theta}) \mathbf{U}_{\boldsymbol{\theta}})^{-1}\Big). \tag{4.66}$$

Furthermore, define the constant $c = \frac{1}{8}$, which is an upper bound for the sectional curvature of $SE(3)$, and assume that the estimator satisfies the requirements of the IVLB [190]. Then, the (intrinsic) variance of the estimator, defined as

$$\mathsf{var}\left\{\widehat{\boldsymbol{\theta}}\right\} = \mathsf{E}\left\{d_{SE(3)}(\widehat{\boldsymbol{\theta}}(y), \boldsymbol{\theta})^2\right\}, \tag{4.67}$$

satisfies

$$\mathsf{var}\left\{\widehat{\boldsymbol{\theta}}\right\} \geq IVLB(\boldsymbol{\theta}) := \frac{\lambda_{\boldsymbol{\theta}} c + 1 - \sqrt{2\lambda_{\boldsymbol{\theta}} c + 1}}{c^2 \lambda_{\boldsymbol{\theta}}/2}. \tag{4.68}$$

The performance bound is given in terms of the intrinsic variance which is defined making use of the Riemannian distance function on the Special Euclidean group $d_{SE(3)}$. Some criticism might be done against this. The distance function mixes both the position and attitude error, so there is no simple way of deriving separate bounds for each of the attitude/position components. It would be more convenient to have separate bounds for the attitude and position errors but this is not yet possible with

the available tools.

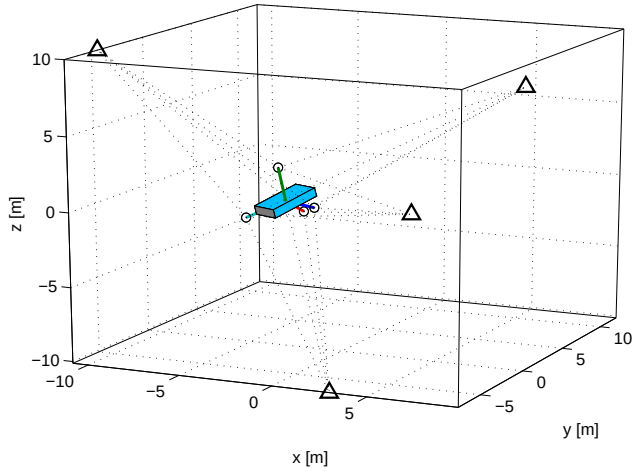
Numerical Monte-Carlo simulations were done to evaluate the performance of the geometric descent algorithms introduced in the previous section. The simulations aimed at comparing the performances of the MLR, MLSR, and 3-step closed form initialization algorithms against the IVLB for the problem at hand. Three simulations were done corresponding to different rigid body configurations with the same landmarks, as depicted in Fig. 4.17. In the first case, shown in Fig. 4.17(a), the rigid body was located near the centroid of the landmarks, yielding a relatively favorable estimation geometry (this is the same setup used in the simulations of section 4.3.6). In the second and third case, shown in Fig. 4.17(b) and Fig. 4.17(c), the riding body was moved away from the landmark centroid, yielding less favorable estimation geometries.

In each of the cases a set of Monte-Carlo simulations with different measurement error covariances were done with the Newton MLR, Newton MLSR and 3-step initialization algorithms. At each Monte Carlo run, a set of noisy range measurements was generated with error covariance $\mathbf{R} = \sigma^2 \mathbf{I}_{mp}$. The standard deviation σ was varied between $\sigma_{min} = 1$mm and $\sigma_{max} = 1$m in ten equally logarithmically spaced points. At each value of standard deviation σ and for each rigid body configuration, a set of $N = 1000$ vectors of noisy range measurements were generated. The algorithms were used to determine an estimated attitude and position with each of the range measurements and an experimental intrinsic variance was determined using (4.67). We were interested in comparing the steady state value of the MLR and MLSR geometric descent algorithms. Since the steady state value is the same for the Newton and gradient descent algorithms, we only simulated the Newton version. Of course the same results could have been found with the gradient versions of the algorithms but at the expense of a very large number of iterations for each run, and a big overall computational time. Since we were only interested in the final result of the algorithms assuming that they converged, the initial estimate was irrelevant and chosen as the actual rigid body configuration. This reduced the computational time (assuming unbiasedness, the expected value of the estimator is the actual value) and minimized the probability of converging to a local minima. Note that this was not needed for the 3-step initialization algorithm since it is a closed form algorithm that does not require an initial estimate.

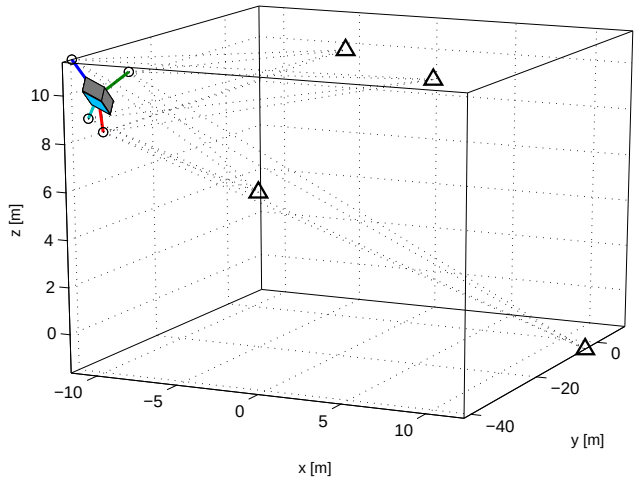
Fig. 4.18(a) shows the simulation results for the actual rigid body configuration in Fig. 4.17(a). As the value of the standard deviation σ is varied, the experimental intrinsic variances of the MLR Newton, MLSR Newton and 3-step initialization are plotted in a logarithmic scale. The figure also shows the value of the IVLB for the problem at hand, computed from (4.68). The experimental variances of all the algorithms seem to be very close to the derived IVLB. This means basically two things: That the estimators seem to be efficient under this conditions (in the sense that they achieve the performance bound) and that the IVLB is a tight bound. As

we mentioned in section 2.5, performance bounds are of little use if they are too conservative (such as the exaggerated case of the *zero* bound). In this case the experimental intrinsic variance of the algorithms seem to match the IVLB which means that the bound is not conservative. If we zoom and have a closer look at variance of the 3-step init algorithm, we see that it is consistently bigger than the IVLB and the other geometric descent algorithms. In particular, we found this difference to be approximately of 10% almost independently of the value of σ. This was a surprising result, since it revealed that the closed form initialization algorithm was able to achieve results in the same order of magnitude as the much more complicated iterative geometric descent algorithms.

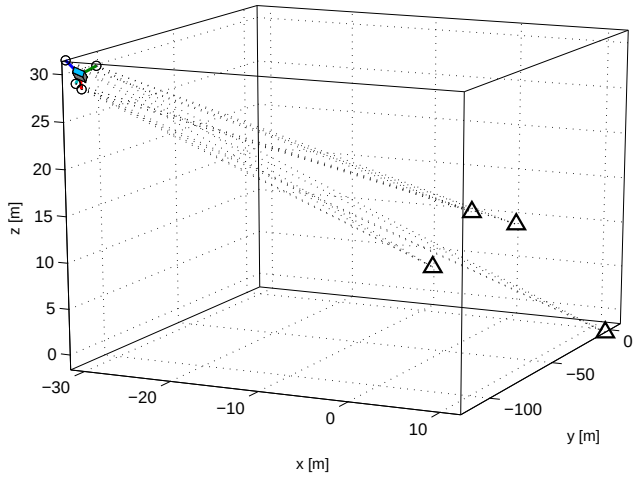
Fig.4.18(b) and Fig.4.18(c) show the results of the simulation setups depicted in Fig.4.17(b)and Fig.4.17(c) respectively. As the rigid body is moved away from the centroid of the landmarks, and a less favorable estimation geometry is found, the algorithms seem to depart from the IVLB at low signal to noise ratios (as the standard deviation σ increases). In this case the simple closed form 3-step initialization algorithm performed much worse than the iterative MLR and MLSR algorithms.



(a)

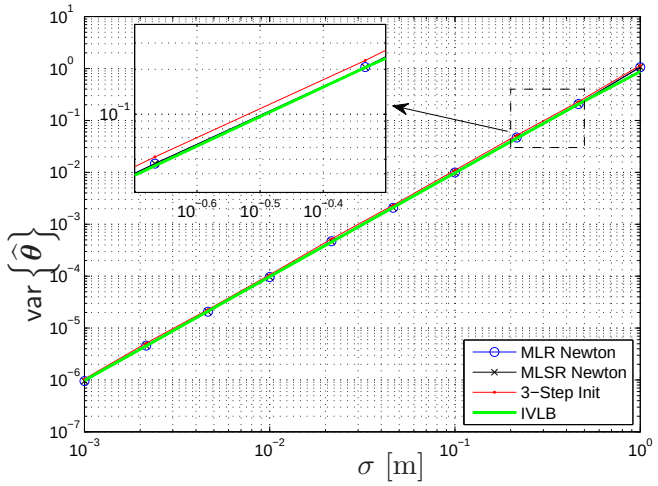


(b)

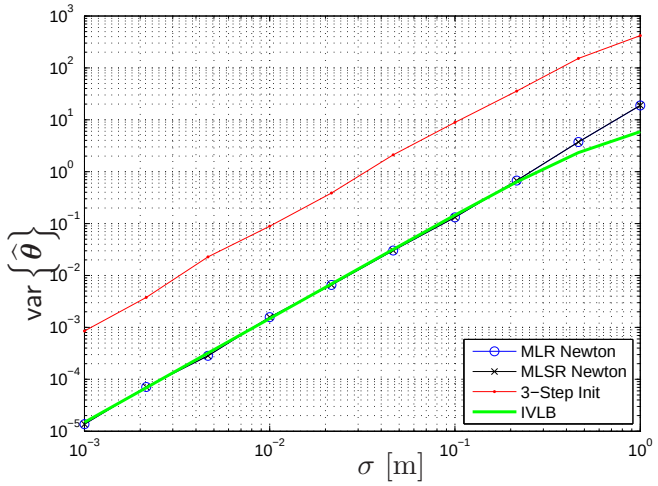


(c)

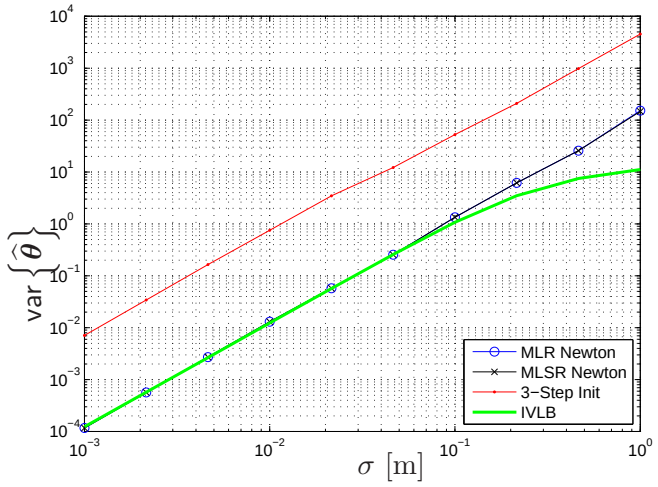
Figure 4.17: IVLB.



(a)



(b)



(c)

Figure 4.18: IVLB.

Chapter 5

Range-Only Dynamic Pose Estimation

5.1 Introduction

In the previous chapter we presented a solution to the Range-Only attitude and positioning problem based on the iterative minimization of a cost function defined on the Special Euclidean group $SE(3)$. The problem was then solved resorting to generalized intrinsic gradient descent and Newton algorithms [15], [14]. However, no convergence warranties were given and simulations revealed the existence of local minimum. Some of the difficulties encountered may originate in the complex structure of the ML-R and ML-SR cost functions which contained squared roots, cross-term products, and was not even differentiable at some points.

In the following section we present a novel approach to the range-only attitude and positioning problem that aims at overcoming some of the difficulties encountered. A modified cost function is considered based on the square of the range observations that exhibits good properties and helps to decouple the attitude and position estimation errors. The presented solution is inspired by the ideas in [33] [81] [51] [99] where, in some cases, parameter estimation and signal processing problems are solved by resorting to special classes of dynamical systems the properties of which can be analyzed from a system theoretic point of view. It is hoped that the dynamic formulation given here will give some insight to the solution of more complex problems that arise when the rigid body undergoes motion in space.

Using a suitable Lyapunov candidate function based on the range observations, and under certain conditions, local asymptotic convergence of the position and attitude estimates to the true values is proved. The conditions required to obtain this result ensure that there is at least a set of noncoplanar landmarks and beacons and that the body reference frame has its origin at the centroid of the beacons. In spite of the result obtained being local, close examination of the literature indicates that it is not trivial in that it addresses explicitly the fact that only range measurements are

available. Furthermore, and because the problem is directly formulated in SE(3), a non global result is expected if one draws an analogy with control problems. In fact, the results in [119] [19], [34] show that due to the non-Euclidean nature of SE(3) it is not possible to render a system evolving on this manifold globally asymptotically stable (GAS) by resorting to continuous feedback control laws. This is only possible when the state space is diffeomorphic to $\mathbb{R}^n$, which is not the case for $SE(3)$. At most, one may expect to obtain an Almost Global Asymptotic Stability (AGAS) type of result as in [51] and show that the region of attraction is the entire state space except for a zero measure set [119], [19]. Although not proved analytically, simulations results with the estimator here proposed did not reveal the existence of local minimums and suggest that, under the non coplanarity conditions, the presented estimator may exhibit AGAS, a conjecture that warrants further research.

5.2 Problem formulation

Inspired by the work in [33] [81] [51] [99], one possible solution to the range-only attitude and positioning problem is to use a dynamical system, or adaptive identifier, the trajectories of which converge asymptotically to the actual attitude and position. This paper will focus on this class of solutions. We are now ready to formulate the problem rigorously.

Problem 5.2.1 (Dynamic Range-Only attitude estimation)**.** *Consider a static rigid body with attitude and position represented by* $(\mathcal{R}, \mathbf{p}) \in SE(3)$*. Suppose one measures the squared distances* $d_{ij} = \|\mathcal{R}\mathbf{b}_i + \mathbf{p} - \mathbf{p}_j\|^2$ *between a set of body-fixed beacons with positions* $\mathbf{b}_i$*;* $i \in \{1, \ldots, p\}$ *expressed in* $\{\mathcal{B}\}$ *and a set of Earth-fixed landmarks with positions* $\mathbf{p}_j$*;* $j \in \{1, \ldots, m\}$ *expressed in* $\{\mathcal{I}\}$ *(see Fig.4.5). Consider the dynamical system*

$$\begin{cases} \dot{\widehat{\mathbf{p}}} = \widehat{\mathbf{v}}, \\ \dot{\widehat{\mathcal{R}}} = \widehat{\mathcal{R}}\mathcal{S}(\widehat{\boldsymbol{\omega}}), \end{cases} \tag{5.1}$$

with initial conditions $(\widehat{\mathcal{R}}(0), \widehat{\mathbf{p}}(0)) \in SE(3)$*. Compute the functions* $\widehat{\mathbf{v}} \in \mathbb{R}^3$ *and* $\widehat{\boldsymbol{\omega}} \in \mathbb{R}^3$ *as a function only of the estimated attitude and position and the squared range measurements, such that the estimated attitude and position* $(\widehat{\mathcal{R}}(t), \widehat{\mathbf{p}}(t))$ *generated by (5.1) converge to the actual rigid body attitude and position* $(\mathcal{R}, \mathbf{p}) \in SE(3)$ *as* t *tends to infinity.*

The problem to be solved is the same as the Range-Only attitude and position estimation. The only difference here is that we intend to use a set of differential equations that when integrated tend asymptotically to the desired solution $(\mathcal{R}, \mathbf{p})$. The term dynamic estimator reflects the dynamic nature of the proposed solution but we should make clear that in the above problem formulation the actual rigid

body is static, i.e., $\dot{\mathbf{p}} = 0$ and $\dot{\mathcal{R}} = 0$. However, the dynamic estimator formulation will hopefully provide insight into the case where the rigid body is not static but describing a certain trajectory in space.

Note that the pair $(\widehat{\mathcal{R}}\mathcal{S}(\widehat{\boldsymbol{\omega}}), \widehat{\mathbf{v}})$ is a valid tangent vector of $SE(3)$ at $(\widehat{\mathcal{R}}, \widehat{\mathbf{p}})$, which means that (5.1) defines a flow on the Special Euclidean group [130], [138], [81]. Hence, ideal integration of the system equations will produce an estimate that evolves naturally on $SE(3)$ without the need to chose a particular parametrization or to resort to normalization schemes. In practice however, if a standard integration method such as Runge-Kutta is used, small errors would be produced that eventually would require a normalization step. This is because standard integration methods do not take into account the non Euclidean geometry of the parameter space [79].

Define the error rotation $\mathcal{R}_e = \widehat{\mathcal{R}}^T\mathcal{R} \in SO(3)$ and the estimation errors

$$\begin{cases} \widetilde{\mathbf{p}} = \mathbf{p} - \widehat{\mathbf{p}}, \\ \widetilde{\mathcal{R}} = \mathcal{R}_e - \mathbf{I}_3 = \widehat{\mathcal{R}}^T\mathcal{R} - \mathbf{I}_3. \end{cases} \tag{5.2}$$

Note that $\widetilde{\mathbf{p}} \in \mathbb{R}^3$ but, in general, $\widetilde{\mathcal{R}} \notin SO(3)$. Assuming a static rigid body ($\dot{\mathcal{R}} = 0$ and $\dot{\mathbf{p}} = 0$), the error dynamics can be written from (5.1)-(5.2) as

$$\begin{cases} \dot{\widetilde{\mathbf{p}}} = -\widehat{\mathbf{v}}, \\ \dot{\widetilde{\mathcal{R}}} = -\mathcal{S}(\widehat{\boldsymbol{\omega}})\widehat{\mathcal{R}}^T\mathcal{R} = -\mathcal{S}(\widehat{\boldsymbol{\omega}})(\widetilde{\mathcal{R}} + \mathbf{I}_3). \end{cases} \tag{5.3}$$

An equivalent formulation of the problem statement can be made using the error variables $(\widetilde{\mathcal{R}}, \widetilde{\mathbf{p}})$ as follows. Determine $\widehat{\mathbf{v}}$ and $\widehat{\boldsymbol{\omega}}$ as functions of the current estimates $(\widehat{\mathcal{R}}, \widehat{\mathbf{p}})$ and the measurements d_{ij} such that $\lim_{t\to\infty}(\widetilde{\mathcal{R}}(t), \widetilde{\mathbf{p}}(t)) = (0, 0)$.

5.3 Statement of main result

Let $\mathbf{P} = [\mathbf{p_1} \ \dots \ \mathbf{p}_m] \in \mathbb{R}^{3\times m}$, and $\mathbf{B} = [\mathbf{b_1} \ \dots \ \mathbf{b}_p] \in \mathbb{R}^{3\times p}$ be matrices containing the landmark and beacon coordinates expressed with respect to $\{\mathcal{I}\}$ and $\{\mathcal{B}\}$ respectively. Define the centering matrices

$$\mathbf{M}_m = \mathbf{I}_m - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^T \in \mathbb{R}^{m\times m}, \tag{5.4}$$

$$\mathbf{M}_p = \mathbf{I}_p - \frac{1}{p}\mathbf{1}_p\mathbf{1}_p^T \in \mathbb{R}^{p\times p}, \tag{5.5}$$

where, in the above expressions, $\mathbf{1}_k \in \mathbb{R}^k$, $k \in \{m, p\}$ denotes a vector of ones. The centering matrices are projection operators and satisfy $\mathbf{M}_k = \mathbf{M}_k^T$, $\mathbf{M}_k\mathbf{M}_k = \mathbf{M}_k$, and $\mathbf{M}_k\mathbf{1}_k = 0$. Let $\mathbf{P}_c = \mathbf{P}\mathbf{M}_m$ be a matrix containing the landmark coordinates expressed with respect to the landmark centroid. Similarly, define $\mathbf{B}_c = \mathbf{B}\mathbf{M}_p$ as

the matrix containing the centered beacon coordinates. Further define the matrices

$$\mathbf{D} := \begin{bmatrix} d_{11} & \dots & d_{p1} \\ \vdots & \ddots & \vdots \\ d_{1m} & \dots & d_{pm} \end{bmatrix} \in \mathbb{R}^{m\times p}, \quad \widehat{\mathbf{D}} := \begin{bmatrix} \hat{d}_{11} & \dots & \hat{d}_{p1} \\ \vdots & \ddots & \vdots \\ \hat{d}_{1m} & \dots & \hat{d}_{pm} \end{bmatrix} \in \mathbb{R}^{m\times p}, \tag{5.6}$$

containing the actual and estimated square range measurements, respectively, between landmarks and beacons, i.e., with entries $d_{ij} = \|\mathcal{R}\mathbf{b}_i + \mathbf{p} - \mathbf{p}_j\|^2$ and $\widehat{d}_{ij} = \|\widehat{\mathcal{R}}\mathbf{b}_i + \widehat{\mathbf{p}} - \mathbf{p}_j\|^2$. The squared range matrices have an important property that will be used in the following developments:

Lemma 5.3.1. *The following equalities hold:*

$$\mathbf{M}_m(\mathbf{D} - \widehat{\mathbf{D}}) = -2\mathbf{P}_c^T\widetilde{\mathbf{p}}\mathbf{1}_p^T + 2\mathbf{P}_c^T\widehat{\mathcal{R}}\widetilde{\mathcal{R}}\mathbf{B}, \tag{5.7}$$

$$\mathbf{M}_m(\mathbf{D} - \widehat{\mathbf{D}})\mathbf{1}_p = -2p\mathbf{P}_c^T\widetilde{\mathbf{p}}, \tag{5.8}$$

$$\mathbf{M}_m(\mathbf{D} - \widehat{\mathbf{D}})\mathbf{M}_p = -2\mathbf{P}_c^T\widehat{\mathcal{R}}\widetilde{\mathcal{R}}\mathbf{B}. \tag{5.9}$$

Proof. The entries of $\mathbf{D}$ and $\widehat{\mathbf{D}}$ have the form

$$d_{ij} = (\mathcal{R}\mathbf{b}_i + \mathbf{p} - \mathbf{p}_j)^T(\mathcal{R}\mathbf{b}_i + \mathbf{p} - \mathbf{p}_j),$$
$$\hat{d}_{ij} = (\widehat{\mathcal{R}}\mathbf{b}_i + \widehat{\mathbf{p}} - \mathbf{p}_j)^T(\widehat{\mathcal{R}}\mathbf{b}_i + \widehat{\mathbf{p}} - \mathbf{p}_j).$$

Simple algebraic manipulations show that the ij entry of $\mathbf{D} - \widehat{\mathbf{D}}$ can be written as

$$d_{ij} - \hat{d}_{ij} = (\mathbf{p} + \widehat{\mathbf{p}} - 2\mathbf{p}_j + 2\widehat{\mathcal{R}}\mathbf{b}_i)^T\widetilde{\mathbf{p}} + 2(\mathbf{p} - \mathbf{p}_j)^T\widehat{\mathcal{R}}\widetilde{\mathcal{R}}\mathbf{b}_i.$$

Defining the vector variables $\mathbf{d}_i := [d_{i1} \ \dots \ d_{im}]^T \in \mathbb{R}^m$ and $\hat{\mathbf{d}}_i := [\hat{d}_{i1} \ \dots \ \hat{d}_{im}]^T \in \mathbb{R}^m$ for $i \in \{1, \dots, p\}$ yields

$$\mathbf{d}_i - \hat{\mathbf{d}}_i = \Big((\mathbf{p} + \widehat{\mathbf{p}})\mathbf{1}_m^T - 2\mathbf{P}\Big)^T\widetilde{\mathbf{p}} + 2\mathbf{1}_m\widetilde{\mathbf{p}}^T\widehat{\mathcal{R}}\mathbf{b}_i + 2(\mathbf{p}\mathbf{1}_m^T - \mathbf{P})^T\widehat{\mathcal{R}}\widetilde{\mathcal{R}}\mathbf{b}_i.$$

As a consequence,

$$\mathbf{D} - \widehat{\mathbf{D}} := \Big[\mathbf{d}_1 - \widehat{\mathbf{d}}_1 \ \dots \ \mathbf{d}_p - \widehat{\mathbf{d}}_p\Big]$$

can be written as

$$\mathbf{D} - \widehat{\mathbf{D}} = \Big((\mathbf{p} + \widehat{\mathbf{p}})\mathbf{1}_m^T - 2\mathbf{P}\Big)^T\widetilde{\mathbf{p}}\mathbf{1}_p^T + 2\mathbf{1}_m\widetilde{\mathbf{p}}^T\widehat{\mathcal{R}}\mathbf{B} + 2(\mathbf{p}\mathbf{1}_m^T - \mathbf{P})^T\widehat{\mathcal{R}}\widetilde{\mathcal{R}}\mathbf{B}.$$

The result follows by using the facts that $\mathbf{M}_m\mathbf{1}_m = 0$, $\mathbf{P}\mathbf{M}_m = \mathbf{P}_c$, $\mathbf{B}\mathbf{1}_p = 0$, $\mathbf{1}_p^T\mathbf{1}_p = p$, and $\mathbf{M}_p\mathbf{1}_p = 0$. □

Assume the following assumptions hold:

Assumption 5.3.2. *The body reference frame $\{\mathcal{B}\}$ has its origin at the centroid of*

the beacons.

Note that if this is true, then $\mathbf{B}_c = \mathbf{B}$ and $\mathbf{B}\mathbf{1}_p = \mathbf{B}_c\mathbf{1}_p = \mathbf{B}\mathbf{M}_p\mathbf{1}_p = 0$. This assumption is not restrictive and will help simplify greatly the developments.

Assumption 5.3.3. *There is a set of noncoplanar beacons and landmarks.*

Since three points in $\mathbb{R}^3$ always define a plane, this is equivalent to requiring that at least four beacons and landmarks be noncoplanar. Note that if this assumption is satisfied then it is easy to show that matrices $\mathbf{P}_c$ and $\mathbf{B}_c$ have full column rank.

Consider the following simple Lyapunov function candidate

$$V = \frac{1}{8}\|\mathbf{M}_m(\mathbf{D} - \widehat{\mathbf{D}})\|_F^2, \tag{5.10}$$

where given a matrix $\mathbf{A}$, $\|\mathbf{A}\|_F^2 = \operatorname{tr}\left(\mathbf{A}^T\mathbf{A}\right)$ is the matrix Frobenius norm. Note that V depends only on the available range measurements $\mathbf{D}$ and the estimated rigid body attitude and position, since $\widehat{\mathbf{D}} = \widehat{\mathbf{D}}(\widehat{\mathcal{R}}, \widehat{\mathbf{p}})$. This Lyapunov function candidate has an interesting property:

Lemma 5.3.4. *The function V in (5.10) can be decomposed in two quadratic terms, one depending only on the actual and estimated position and the other on the attitude estimation error, i.e.,*

$$V = \frac{1}{2}\widetilde{\mathbf{p}}^T\mathbf{\Theta}_1\widetilde{\mathbf{p}} + \frac{1}{2}\operatorname{vec}(\widetilde{\mathcal{R}})^T\mathbf{\Theta}_2\operatorname{vec}(\widetilde{\mathcal{R}}), \tag{5.11}$$

where vec() *is the operator that stacks the columns of a matrix from left to right,*

$$\mathbf{\Theta}_1 = m\mathbf{P}_c\mathbf{P}_c^T \in \mathbb{R}^{3\times 3}, \tag{5.12}$$

$$\mathbf{\Theta}_2 = \mathbf{B}\mathbf{B}^T \otimes \widehat{\mathcal{R}}^T\mathbf{P}_c\mathbf{P}_c^T\widehat{\mathcal{R}} \in \mathbb{R}^{9\times 9}, \tag{5.13}$$

and $\otimes$ denotes the Kronecker product of matrices.

Proof. Using Lemma 5.3.1 and the fact that $\mathbf{B}\mathbf{1}_p = 0$ yields

$$\begin{aligned}
V &= \frac{1}{8}\|\mathbf{M}_m(\mathbf{D} - \widehat{\mathbf{D}})\|_F^2 \\
&= \frac{1}{8}\operatorname{tr}\left(\mathbf{M}_m(\mathbf{D} - \widehat{\mathbf{D}})(\mathbf{D} - \widehat{\mathbf{D}})^T\mathbf{M}_m^T\right) \\
&= \frac{1}{2}\operatorname{tr}\left(m\mathbf{P}_c^T\widetilde{\mathbf{p}}\widetilde{\mathbf{p}}^T\mathbf{P}_c + \mathbf{P}_c^T\widehat{\mathcal{R}}\widetilde{\mathcal{R}}\mathbf{B}\mathbf{B}^T\widetilde{\mathcal{R}}^T\widehat{\mathcal{R}}^T\mathbf{P}_c\right) \\
&= \frac{1}{2}m\widetilde{\mathbf{p}}^T\mathbf{P}_c\mathbf{P}_c^T\widetilde{\mathbf{p}} + \frac{1}{2}\operatorname{tr}\left(\mathbf{P}_c^T\widehat{\mathcal{R}}\widetilde{\mathcal{R}}\mathbf{B}\mathbf{B}^T\widetilde{\mathcal{R}}^T\widehat{\mathcal{R}}^T\mathbf{P}_c\right).
\end{aligned}$$

The first term is clearly a quadratic function of the position estimation error $\widetilde{\mathbf{p}}$. The second term can also be shown to be a quadratic function of the attitude estimation error. Since for conformable matrices $\operatorname{vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A})\operatorname{vec}(\mathbf{B})$, we have

$$\operatorname{vec}(\mathbf{P}_c^T\widehat{\mathcal{R}}\widetilde{\mathcal{R}}\mathbf{B}) = (\mathbf{B}^T \otimes \mathbf{P}_c^T\widehat{\mathcal{R}})\operatorname{vec}(\widetilde{\mathcal{R}}). \tag{5.14}$$

Moreover, using the identity $\text{tr}\left(\mathbf{A}\mathbf{A}^T\right) = \text{vec}(\mathbf{A})^T\text{vec}(\mathbf{A})$ we obtain

$$\begin{aligned}
&\text{tr}\left(\mathbf{P}_c^T\widehat{\mathcal{R}}\widetilde{\mathcal{R}}\mathbf{B}\mathbf{B}^T\widetilde{\mathcal{R}}^T\widehat{\mathcal{R}}^T\mathbf{P}_c\right) = \text{tr}\left(\mathbf{B}^T\widetilde{\mathcal{R}}^T\widehat{\mathcal{R}}^T\mathbf{P}_c\mathbf{P}_c^T\widehat{\mathcal{R}}\widetilde{\mathcal{R}}\mathbf{B}\right)\\
&= \text{tr}\left((\mathbf{P}_c^T\widehat{\mathcal{R}}\widetilde{\mathcal{R}}\mathbf{B})^T\mathbf{P}_c^T\widehat{\mathcal{R}}\widetilde{\mathcal{R}}\mathbf{B}\right)\\
&= \text{vec}(\mathbf{P}_c^T\widehat{\mathcal{R}}\widetilde{\mathcal{R}}\mathbf{B})^T\text{vec}(\mathbf{P}_c^T\widehat{\mathcal{R}}\widetilde{\mathcal{R}}\mathbf{B})\\
&= \text{vec}(\widetilde{\mathcal{R}})^T(\mathbf{B}^T\otimes\mathbf{P}_c^T\widehat{\mathcal{R}})^T(\mathbf{B}^T\otimes\mathbf{P}_c^T\widehat{\mathcal{R}})\text{vec}(\widetilde{\mathcal{R}})\\
&= \text{vec}(\widetilde{\mathcal{R}})^T(\mathbf{B}\mathbf{B}^T\otimes\widehat{\mathcal{R}}^T\mathbf{P}_c\mathbf{P}_c^T\widehat{\mathcal{R}})\text{vec}(\widetilde{\mathcal{R}}),
\end{aligned}$$

and the result follows. □

Corollary 5.3.5. *Suppose assumptions 5.3.2-5.3.3 are satisfied. Then, V is a positive definite function of the estimation errors.*

Proof. According to Lemma 5.3.4, it is enough to show that, when assumptions 5.3.2-5.3.3 are satisfied, then $\mathbf{\Theta}_1$ and $\mathbf{\Theta}_2$ are positive definite matrices. If $\mathbf{P}_c, \mathbf{B} \in \mathbb{R}^{3\times m}$ are full column rank, then $\mathbf{P}_c\mathbf{P}_c^T \in \mathbb{R}^{3\times 3}$ and $\mathbf{B}\mathbf{B}^T \in \mathbb{R}^{3\times 3}$ are positive definite. If $\widehat{\mathcal{R}}$ is a rotation matrix it is nonsingular, and therefore $\widehat{\mathcal{R}}^T\mathbf{P}_c\mathbf{P}_c^T\widehat{\mathcal{R}}$ is also positive definite. Since the Kronecker product of two positive definite matrices is also positive definite, the result follows. □

We are now ready to state the main result of the section:

Theorem 5.3.6. *Suppose that assumptions 5.3.2-5.3.3 are fulfilled. Consider the adaptive estimator in (5.1) with*

$$\begin{cases}\widehat{\mathbf{v}} &= -\mathbf{K}_v\mathbf{P}_c(\mathbf{D}-\widehat{\mathbf{D}})\mathbf{1}_p,\\ \widehat{\boldsymbol{\omega}} &= -\mathbf{K}_\omega\mathcal{S}^{-1}\left(\mathbf{\Psi}-\mathbf{\Psi}^T\right),\end{cases} \tag{5.15}$$

where $\mathbf{\Psi} = \frac{1}{2}\mathbf{B}(\mathbf{D}-\widehat{\mathbf{D}})^T\mathbf{P}_c^T\widehat{\mathcal{R}} \in \mathbb{R}^{3\times 3}$ and $\mathbf{K}_v, \mathbf{K}_\omega \in \mathbb{R}^{3\times 3}$ are positive definite matrix gains. Then, the error system (5.3) has an asymptotically stable equilibrium point at the origin $(\widetilde{\mathcal{R}}, \widetilde{\mathbf{p}}) = (0, 0)$.

Proof. Consider the estimation error variables $(\widetilde{\mathcal{R}}, \widetilde{\mathbf{p}})$ defined in (5.2). The time derivative of the Lyapunov function along the trajectories of the system can be computed as

$$\dot{V} = m\widetilde{\mathbf{p}}^T\mathbf{P}_c\mathbf{P}_c^T\dot{\widetilde{\mathbf{p}}} + \text{tr}\left(\mathbf{P}_c^T(\dot{\widehat{\mathcal{R}}}\widetilde{\mathcal{R}} + \widehat{\mathcal{R}}\dot{\widetilde{\mathcal{R}}})\mathbf{B}\mathbf{B}^T\widetilde{\mathcal{R}}^T\widehat{\mathcal{R}}^T\mathbf{P}_c\right).$$

From the error dynamics in (5.3),

$$\dot{\widehat{\mathcal{R}}}\widetilde{\mathcal{R}} + \widehat{\mathcal{R}}\dot{\widetilde{\mathcal{R}}} = \widehat{\mathcal{R}}\mathcal{S}(\widehat{\boldsymbol{\omega}})\widetilde{\mathcal{R}} - \widehat{\mathcal{R}}\mathcal{S}(\widehat{\boldsymbol{\omega}})\widehat{\mathcal{R}}^T\mathcal{R} = -\widehat{\mathcal{R}}\mathcal{S}(\widehat{\boldsymbol{\omega}}), \tag{5.16}$$

and

$$\begin{aligned}\dot{V} &= -m\widetilde{\mathbf{p}}^T\mathbf{P}_c\mathbf{P}_c^T\widehat{\mathbf{v}} - \operatorname{tr}\left(\mathbf{P}_c^T\widehat{\mathcal{R}}\mathcal{S}(\widehat{\boldsymbol{\omega}})\mathbf{B}\mathbf{B}^T\widetilde{\mathcal{R}}^T\widehat{\mathcal{R}}^T\mathbf{P}_c\right)\\ &= -m\widetilde{\mathbf{p}}^T\mathbf{P}_c\mathbf{P}_c^T\widehat{\mathbf{v}} - \operatorname{tr}\left(\mathcal{S}(\widehat{\boldsymbol{\omega}})\mathbf{B}\mathbf{B}^T\widetilde{\mathcal{R}}^T\widehat{\mathcal{R}}^T\mathbf{P}_c\mathbf{P}_c^T\widehat{\mathcal{R}}\right)\\ &= -m\widetilde{\mathbf{p}}^T\mathbf{P}_c\mathbf{P}_c^T\widehat{\mathbf{v}} - \boldsymbol{\xi}^T\widehat{\boldsymbol{\omega}},\end{aligned} \tag{5.17}$$

where $\boldsymbol{\xi} := \mathcal{S}^{-1}(\boldsymbol{\Sigma} - \boldsymbol{\Sigma}^T) \in \mathbb{R}^3$ and $\boldsymbol{\Sigma} = \mathbf{B}\mathbf{B}^T\widetilde{\mathcal{R}}^T\widehat{\mathcal{R}}^T\mathbf{P}_c\mathbf{P}_c^T\widehat{\mathcal{R}}$. This can be shown using the property that given vector $\mathbf{a} \in \mathbb{R}^n$ and matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$, $\operatorname{tr}(\mathcal{S}(\mathbf{a})\mathbf{B}) = -\mathbf{a}^T\mathcal{S}^{-1}(\mathbf{B} - \mathbf{B}^T)$. A natural choice for $\widehat{\mathbf{v}}$ and $\widehat{\boldsymbol{\omega}}$ is

$$\begin{cases}\widehat{\mathbf{v}} &= \mathbf{K}_v\mathbf{P}_c\mathbf{P}_c^T\widetilde{\mathbf{p}},\\ \widehat{\boldsymbol{\omega}} &= \mathbf{K}_\omega\boldsymbol{\xi},\end{cases} \tag{5.18}$$

where $\mathbf{K}_v$ and $\mathbf{K}_\omega$ are positive definite matrix gains that make the derivative of V, computed as

$$\dot{V} = -m\widetilde{\mathbf{p}}^T\mathbf{P}_c\mathbf{P}_c^T\mathbf{K}_v\mathbf{P}_c\mathbf{P}_c^T\widetilde{\mathbf{p}} - \boldsymbol{\xi}^T\mathbf{K}_\omega\boldsymbol{\xi}, \tag{5.19}$$

negative semidefinite. Moreover, it will be shown that $\boldsymbol{\xi}$ is different from zero in a neighborhood of $\widetilde{\mathcal{R}} = 0$ and therefore $\dot{V}$ is a negative definite function in a neighborhood of the equilibrium $(\widetilde{\mathcal{R}}, \widetilde{\mathbf{p}}) = (0, 0)$.

To prove this, let $\mathcal{R}_e = \widehat{\mathcal{R}}^T\mathcal{R} \in SO(3)$. Note that since $\widetilde{\mathcal{R}} = \mathcal{R}_e - \mathbf{I}_3$ we can write

$$\begin{aligned}\boldsymbol{\Sigma} &= \mathbf{B}\mathbf{B}^T\widetilde{\mathcal{R}}^T\widehat{\mathcal{R}}^T\mathbf{P}_c\mathbf{P}_c^T\widehat{\mathcal{R}}\\ &= \mathbf{B}\mathbf{B}^T(\mathbf{I}_3 - \mathcal{R}_e)\mathcal{R}^T\mathbf{P}_c\mathbf{P}_c^T\mathcal{R}\mathcal{R}_e^T\\ &= \mathbf{B}\mathbf{B}^T(\mathbf{I}_3 - \mathcal{R}_e)\mathbf{C}\mathbf{C}^T\mathcal{R}_e^T,\end{aligned}$$

where we defined $\mathbf{C} = \mathcal{R}^T\mathbf{P}_c$. Viewing $\mathcal{R}, \mathbf{B}$, and $\mathbf{P}_c$ as fixed parameters, matrix $\boldsymbol{\Sigma}$ can be seen as a function of the error rotation $\mathcal{R}_e$. Define a map $\phi : SO(3) \to \mathsf{K}(3, \mathbb{R})$ as $\phi(\mathcal{R}_e) = \boldsymbol{\Sigma} - \boldsymbol{\Sigma}^T$ and note that $\boldsymbol{\xi} = \mathcal{S}^{-1}(\boldsymbol{\Sigma} - \boldsymbol{\Sigma}^T)$ can be seen as the composition map $\boldsymbol{\xi} = \mathcal{S}^{-1} \circ \phi$. Moreover, note that $\boldsymbol{\xi}(\mathcal{R}_e) = 0$ if and only if $\phi(\mathcal{R}_e) = 0$. The tangent space of $SO(3)$ at the identity matrix $\mathbf{I}_3$ can be identified with the set of skew symmetric matrices $\mathsf{K}(3, \mathbb{R})$. Define the *push forward* of ϕ at $\mathcal{R}_e = \mathbf{I}_3$ as the map $\phi_* : \mathsf{K}(3, \mathbb{R}) \to \mathsf{K}(3, \mathbb{R})$ [110]. After some simplifications this map can be written as

$$\phi_*(\Delta) = -\mathbf{B}\mathbf{B}^T\Delta\mathbf{C}\mathbf{C}^T - \mathbf{C}\mathbf{C}^T\Delta\mathbf{B}\mathbf{B}^T,$$

since $\Delta^T = -\Delta$. With some abuse of notation we can write the vectorized version

of the *push forward* as $\text{vec}(\phi_*(\Delta)) = \mathbf{\Phi}\text{vec}(\Delta)$, where

$$\mathbf{\Phi} = -\mathbf{C}\mathbf{C}^T \otimes \mathbf{B}\mathbf{B}^T - \mathbf{B}\mathbf{B}^T \otimes \mathbf{C}\mathbf{C}^T.$$

It is easy to show that $\mathbf{\Phi} \in \mathbb{R}^{9\times 9}$ is negative definite and therefore invertible. Under assumptions 5.3.2-5.3.3, matrices $\mathbf{B}\mathbf{B}^T \in \mathbb{R}^{3\times 3}$ and $\mathbf{C}\mathbf{C}^T \in \mathbb{R}^{3\times 3}$ are positive definite. The Kronecker product of two positive definite matrices is also positive definite, and so is the sum of two positive definite matrices. Hence, $\mathbf{\Phi}$ is nonsingular, and we have that $\phi_*(\Delta) = 0$ if and only if $\Delta = 0$. This shows that the push forward of ϕ at $\mathcal{R}_e = \mathbf{I}_3$ is an isomorphism and therefore, using the inverse function theorem [110, p.105], ϕ is a local diffeomorphism. This implies that there is a neighborhood $\Omega \subset SO(3)$ of $\mathcal{R}_e = \mathbf{I}_3$, such that $\phi(\mathcal{R}_e) \neq 0$ for all $\mathcal{R}_e \in \Omega/\mathbf{I}_3$. This in turn implies that $\boldsymbol{\xi}(\mathcal{R}_e) \neq 0$ for all $\mathcal{R}_e \in \Omega/\mathbf{I}_3$. Because $\widetilde{\mathcal{R}} = \mathcal{R}_e - \mathbf{I}_3$, the result follows.

It is still necessary to show that the derived estimation laws can actually be computed using only the available information such that (5.15) and (5.18) are equivalent. This can be seen by using Lemma 5.3.1 in the appendix, thus concluding the proof. □

5.4 Simulation Results

Simulation results using MATLAB are presented to illustrate the behavior of the adaptive range-only attitude and position estimator. The simulation setup is shown in Fig. 5.1, where four ($m = 4$) Earth fixed landmarks and a rigid body with four ($p = 3$) beacons were considered. The initial attitude and position estimate $(\widehat{\mathcal{R}}(0), \widehat{\mathbf{p}}(0))$ were set as a random rotation and a random position, respectively as depicted in Fig. 5.1 and Fig. 5.4. The estimator differential equations (5.1) were integrated using MATLAB and a Runge-Kutta method until $t = 10$ using the gains $\mathbf{K}_v = \mathbf{K}_\omega = \gamma\mathbf{I}_3$ with $\gamma = 10^{-3}$. The actual and estimated attitude and position are plotted in Fig.5.2. The attitude and position estimation errors are depicted in Fig.5.3. In order to plot the attitude estimation error, exponential coordinates of the error rotation $\mathcal{R}_e = \widehat{\mathcal{R}}^T\mathcal{R}$ are used, i.e., the error rotation is parametrized by vector $\boldsymbol{\theta} = [\theta_1\ \theta_2\ \theta_3]^T$ where $\mathcal{R}_e = \exp(\mathcal{S}(\boldsymbol{\theta}))$. Note that this is done only for visualization purposes and that no particular parametrization of the Special Orthogonal group $SO(3)$ is used elsewhere in the paper. The residuals $\epsilon_{ij} := |d_{ij} - \hat{d}_{ij}|^{1/2}$ are shown in Fig.5.5.

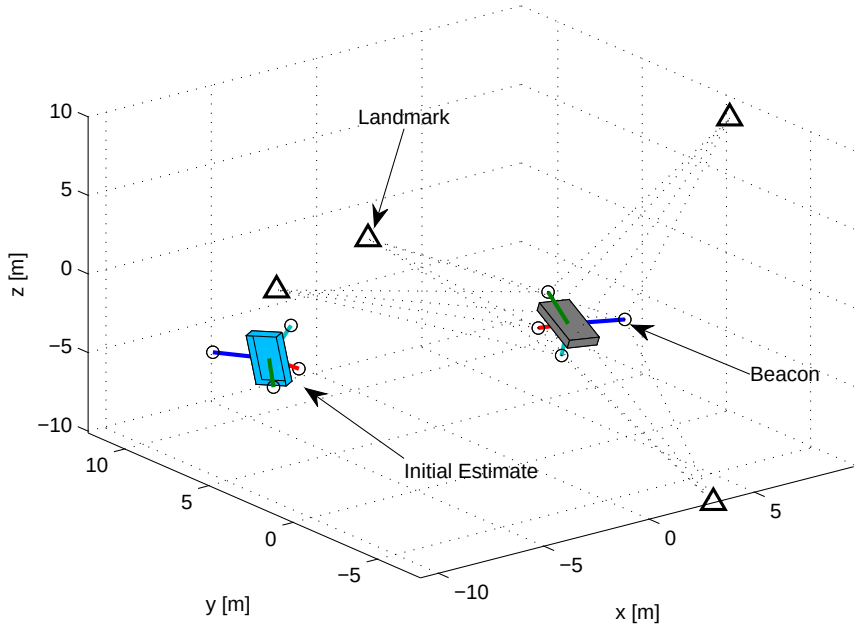


Figure 5.1: Simulation setup. Actual and initial estimated rigid body configurations together with beacon and landmark locations.

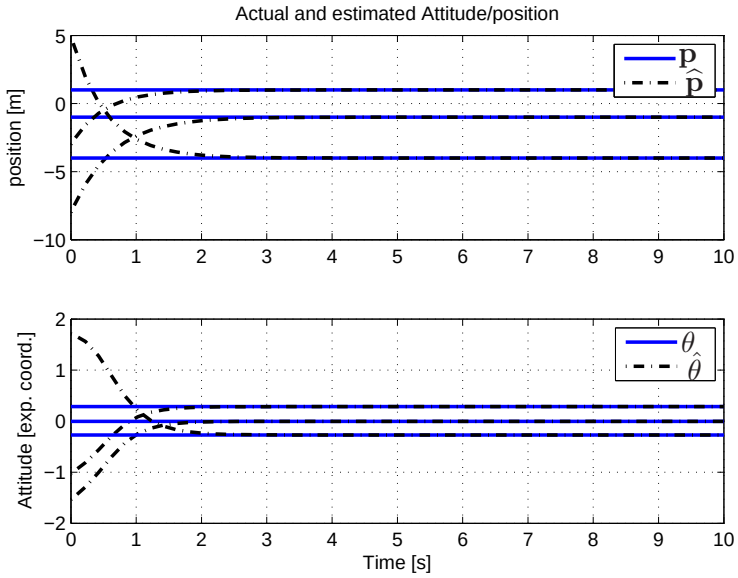


Figure 5.2: Actual and estimated position / attitude. (Top) Entries of vectors $\mathbf{p}$, and $\widehat{\mathbf{p}}$. (Bot.) Entries of vectors $\boldsymbol{\theta}$ and $\widehat{\boldsymbol{\theta}}$, the exponential coordinates of $\mathcal{R}$ and $\widehat{\mathcal{R}}$ respectively.

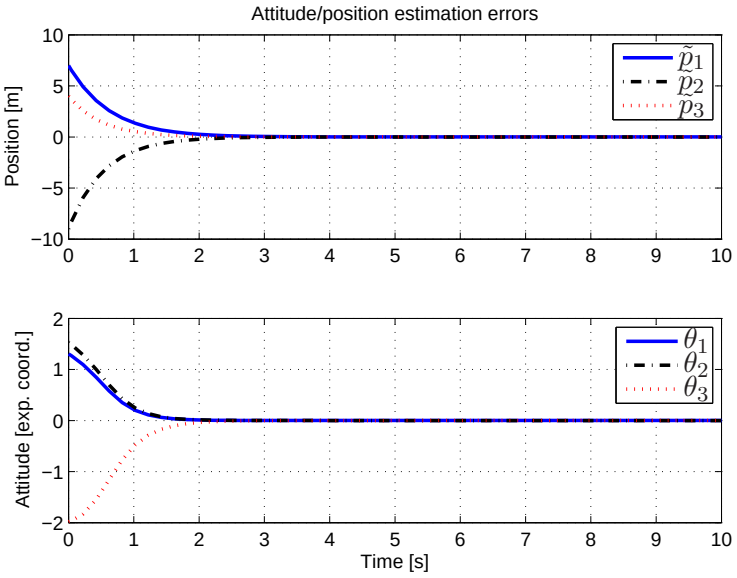


Figure 5.3: Position / attitude estimation errors. (Top) Entries of vector $\widetilde{\mathbf{p}} = \mathbf{p} - \widehat{\mathbf{p}} = [\tilde{p}_1 \ \tilde{p}_2 \ \tilde{p}_3]^T$. (Bot.) Entries of vector $\boldsymbol{\theta} = [\theta_1 \ \theta_2 \ \theta_3]^T$, the exponential coordinates of $\mathcal{R}_e = \widehat{\mathcal{R}}^T\mathcal{R}$, i.e., $\mathcal{R}_e = \exp(\mathcal{S}(\boldsymbol{\theta}))$.

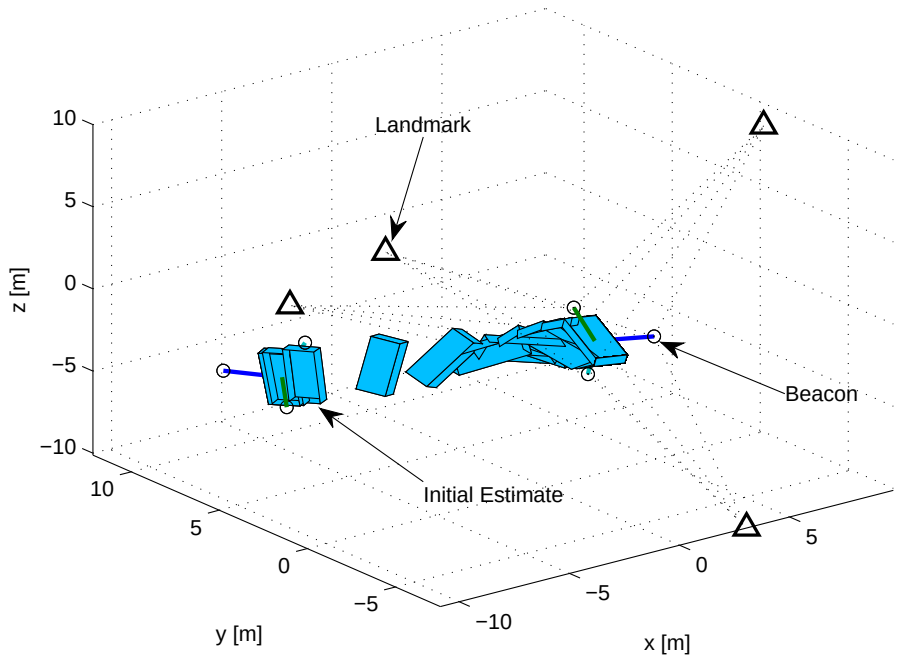


Figure 5.4: Time evolution of the estimated rigid body configuration.

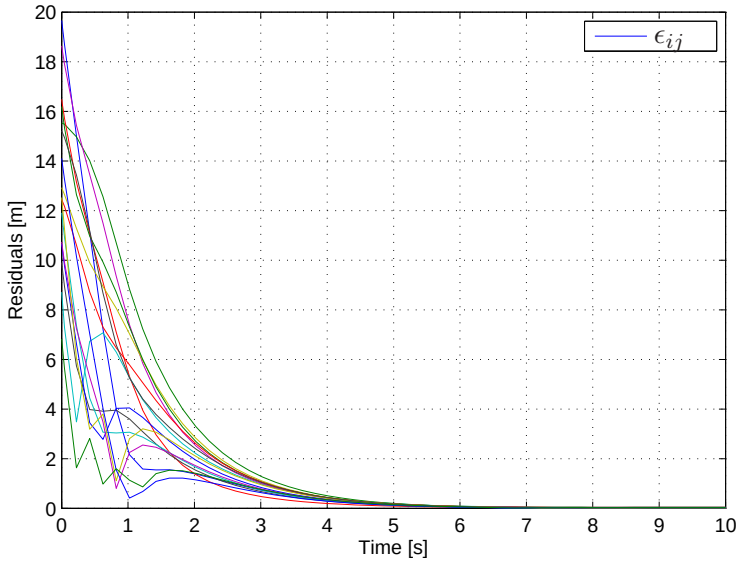


Figure 5.5: Time evolution of the square root of the absolute value of the residuals $\epsilon_{ij} := |d_{ij} - \hat{d}_{ij}|^{1/2}$ (in meters).

Chapter 6

Diffusion-based Smoothing

This chapter addresses the problem of post-processing the position data from an acoustic positioning system. An extension of diffusion-based trajectory observers is derived that incorporates information about the position measurement errors. Experimental results using data from sea trials is presented and convergence warranties are given in the appendix.

6.1 Introduction

In many applications, one is interested in determining a smooth trajectory that fits a set of sparse and noisy measurements in a fixed time interval. For instance, when post-processing underwater navigation data obtained with an acoustic positioning system and a Doppler Velocity Log (DVL) [89], [100], [61]. Although off-line navigation is of crucial practical importance for the correction of geo-referenced scientific data, it has received considerably less attention than real-time navigation [10], [98], [187].

Diffusion-based trajectory observers have been recently proposed as a simple and efficient framework to solve diverse smoothing problems in underwater navigation [88], [89]. These observers are conceptually simple, and can cope naturally with asynchronous measurements and dropouts (temporally loss of measurements). From a conceptual point of view, diffusion-based trajectory observers have strong links with several other disciplines such as de-noising and snakes in image processing [148], [46], [191], and nonparametric smoothing splines [58].

In the original formulation, the objective is to minimize an energy-like functional that penalizes the derivatives of the trajectory and a function of the distances between points along the trajectory and a set of available measurements. The trajectory is modeled as a curve $s \mapsto X(s) \in \mathbb{R}^n$, $s \in [s_b, s_e]$ where $n \in \{2, 3\}$ is the dimension of the trajectory ambient space, and $[s_b, s_e] \subset \mathbb{R}$ is the time interval of interest. Suppose there are N_m position measurements $X_m(\tau_i) \in \mathbb{R}^n$ corresponding

to times $\tau_i \in \mathcal{I} = \{\tau_1, \ldots, \tau_{N_m}\}$. Consider an energy-like functional of the form

$$E = \int_{s_b}^{s_e} \alpha_1 \|\nabla X(s)\|^2 + \alpha_2 \|\nabla^2 X(s)\|^2 + K\gamma(s)\|X(s) - X_m(s)\|^2 \mathrm{d}s \tag{6.1}$$

where $\alpha_1, \alpha_2 > 0$ are user-defined parameters, ∇^i stands for the i'th order derivative with respect to the trajectory time s, and K is a user-defined gain. The factor $\gamma(s)$ is introduced to capture the fact that position measurements are available only at some discrete instants of time and is defined as

$$\gamma(s) = \sum_{\tau_i \in \mathcal{I}} \delta(s - \tau_i), \tag{6.2}$$

where $\delta(\cdot)$ stands for the Dirac delta function. For simplicity of presentation, in the present work we will focus on energy functionals of the type (6.1). This corresponds to the *acoustic data smoothing* problem in [89] where it is assumed that no velocity measurements are available. Including continuous velocity measurements can be easily done with minor modifications in the derivation that follows by considering terms of the form $\|\nabla X(s) - V_m(s)\|^2$ in the energy functional.

The minimizing trajectory must satisfy the Euler Lagrange equation

$$-\alpha_1 \nabla^2 X(s) + \alpha_2 \nabla^4 X(s) + K\gamma(s)(X(s) - X_m(s)) = 0 \tag{6.3}$$

and can be computed using a dynamical system, called a trajectory observer, that follows the negative gradient-like flow and is defined as follows:

$$\frac{\partial}{\partial t} X(s,t) = \alpha_1 \nabla^2 X(s,t) - \alpha_2 \nabla^4 X(s,t) - K\gamma(s)(X(s,t) - X_m(s)). \tag{6.4}$$

Note that we are now considering a continuous of trajectories $X(s,t)$ where s is the *trajectory* time, and t is the *improvement* time. The observer is started at some initial trajectory $X(s,0)$, and the desired solution is the limit of $X(s,t)$ as $t \to \infty$. The question of whether the previous flow converges and has a unique stable equilibrium trajectory is of crucial importance. In [87] the stability proof for a similar flow based on Lyapunov analysis is presented. In [88] the convergence of a finite difference implementation of the observer is guaranteed given that the observer gain K is a positive constant.

In practice, the choice of the value of the observer gain is often left to the user and is done through trial and error. In many situations, however, there is some a-priori knowledge on the variance of the measurement errors that could be used in

order to chose K. In this chapter we propose a methodology to determine K based on minimizing an energy like function subject to a variance constraint that can for all purposes be viewed as an observer "tuning knob". The observer gain is viewed as a Lagrange multiplier, as inspired by [148], and determined by solving a system of equations involving that constraint.

6.2 Trajectory observers with variance constraints

Suppose the variance of the position measurement errors σ_X^2 is known. Consider the minimization of the cost function

$$E = \int_{s_b}^{s_e} \alpha_1 \|\nabla X(s)\|^2 + \alpha_2 \|\nabla^2 X(s)\|^2 \mathrm{d}s \tag{6.5}$$

subject to the constraint

$$\frac{1}{N_m} \int_{s_b}^{s_e} \gamma(s) \|X(s) - X_m(s)\|^2 \mathrm{d}s = \sigma_X^2. \tag{6.6}$$

The minimizing trajectory must now satisfy the Euler Lagrange equations (6.3) and (6.6). In order to determine the value of K that satisfies both equations, we can multiply (6.3) on the left by $\gamma(s)(X(s)-X_m(s))^T$ and integrate in the interval $[s_b, s_e]$ to obtain

$$K = \frac{1}{\sigma_X^2 N_m} \int_{s_b}^{s_e} \gamma(s)(X(s) - X_m(s))^T \\ (\alpha_2 \nabla^4 X(s) - \alpha_1 \nabla^2 X(s)) \mathrm{d}s. \tag{6.7}$$

Injecting the derived gain in the trajectory observer equations (6.4) can be interpreted as a gradient projection method for the constrained minimization problem (6.5)-(6.6).

6.2.1 Numerical Implementation

The observer derived above has the form of a nonlinear partial differential equation that is not easy to solve in practice. Next, we formulate a finite difference approximation to the observer that can be easily implemented.

The differential equation that defines the observer is well suited to estimate the trajectory at all times s in the closed interval $[s_b, s_e]$. In practice, one is often interested in estimating the trajectory only at a set of uniformly distributed instants of time $\{s_1, s_2, \ldots, s_N\}$, where $s_1 = s_b$, $s_N = s_e$, and $\Delta s = s_{i+1} - s_i$, that is, in

estimating a discretized version of the trajectory defined as

$$\mathbf{X} = \begin{bmatrix} X(s_1)^T \\ X(s_2)^T \\ \vdots \\ X(s_N)^T \end{bmatrix} \in \mathbb{R}^{N\times n}. \tag{6.8}$$

The position measurements can also be put in this form, assuming that $\mathcal{I} \subseteq \{s_1, s_2, \ldots, s_N\}$, by defining a matrix $\mathbf{X}_m \in \mathbb{R}^{N\times n}$ with row i given by $X_m(s_i)^T$ whenever there is a measurement at time s_i and a row of zeros otherwise. Let vec() denote the operator that stacks the columns of a matrix from left to right. Define the vector variables $\mathbf{x} = \text{vec}(\mathbf{X}) \in \mathbb{R}^{Nn}$, and $\mathbf{x}_m = \text{vec}(\mathbf{X}_m) \in \mathbb{R}^{Nn}$. The finite difference implementation of the trajectory observer (6.4) can now be written as

$$\begin{cases} \dot{\mathbf{x}} &= f(\mathbf{x}) = \mathbf{A}\mathbf{x} - K\mathbf{\Gamma}(\mathbf{x} - \mathbf{x}_m) + \mathbf{b} \\ K &= \frac{1}{\sigma_X^2 N_m}(\mathbf{x} - \mathbf{x}_m)^T\mathbf{\Gamma}(\mathbf{A}\mathbf{x} + \mathbf{b}) \end{cases} \tag{6.9}$$

where

$$\mathbf{A} = I_n \otimes (\alpha_1\mathbf{L}_1 - \alpha_2\mathbf{L}_2) \in \mathbb{R}^{Nn\times Nn}, \tag{6.10}$$

$$\mathbf{\Gamma} = I_n \otimes \begin{bmatrix} \gamma_1 & & 0 \\ & \ddots & \\ 0 & & \gamma_N \end{bmatrix} \in \mathbb{R}^{Nn\times Nn}, \tag{6.11}$$

$$\gamma_i = \int_{s_b}^{s_e} \gamma(s_i)\mathrm{d}s = \begin{cases} 1 & \text{if } s_i \in \mathcal{I}, \\ 0 & \text{otherwise,} \end{cases} \tag{6.12}$$

I_n is the $n \times n$ identity matrix, and $\otimes$ denotes the Kronecker product of matrices. The matrices $\mathbf{L}_1, \mathbf{L}_2$, defined as

$$\mathbf{L}_1 = \frac{1}{\Delta s^2}\begin{bmatrix} -2 & 1 & 0 & \cdots \\ 1 & -2 & 1 & \ddots \\ \ddots & \ddots & \ddots & \ddots \\ \cdots & 0 & 1 & -2 \end{bmatrix} \in \mathbb{R}^{N\times N}, \tag{6.13}$$

$$\mathbf{L}_2 = \frac{1}{\Delta s^4}\begin{bmatrix} 6 & -4 & 1 & 0 & \cdots \\ -4 & 6 & -4 & 1 & \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ & 1 & -4 & 6 & -4 \\ \cdots & 0 & 1 & -4 & 6 \end{bmatrix} \in \mathbb{R}^{N\times N}, \tag{6.14}$$

are the centered finite difference approximations with Dirichlet boundary conditions of ∇^2 and ∇^4, respectively. Vector $\mathbf{b} = \text{vec}(\mathbf{B}) \in \mathbb{R}^{Nn}$ is obtained from a constant

matrix $\mathbf{B} \in \mathbb{R}^{N \times n}$ containing the Dirichlet boundary conditions. More specifically, assuming that $X(s,t) = \mathbf{x}_b, \forall s \leq s_b$ and $X(s,t) = \mathbf{x}_e, \forall s \geq s_e$, then $\mathbf{B} = \alpha_1 \mathbf{B}_1 - \alpha_2 \mathbf{B}_2$ where

$$\mathbf{B}_1 = \frac{1}{\Delta s^2} \begin{bmatrix} \mathbf{x}_b & 0 & \dots & 0 & \mathbf{x}_e \end{bmatrix}^T, \tag{6.15}$$

$$\mathbf{B}_2 = \frac{1}{\Delta s^4} \begin{bmatrix} -3\mathbf{x}_b & \mathbf{x}_b & 0 & \dots & 0 & \mathbf{x}_e & -3\mathbf{x}_e \end{bmatrix}^T. \tag{6.16}$$

Note that this implementation differs from the original formulation, that used complex numbers [89], and allows to deal easily with both 2D and 3D trajectories.

The initial trajectory estimate $\mathbf{x}(0)$ is an important parameter that needs to be carefully chosen. As shown in the appendix, under some assumptions, the observer is only locally asymptotically stable. One must be aware that there are some initial trajectories for which the observer diverges. A natural choice is to start with a linear interpolation of the available measurements. Another possibility is to chose a constant positive gain K and use the closed form solution to $f(\mathbf{x}) = 0$ given by

$$\mathbf{x} = -(\mathbf{A} - K\mathbf{\Gamma})^{-1}(K\Gamma\mathbf{x}_m + \mathbf{b}). \tag{6.17}$$

where, provided that $\mathbf{A}$ is negative definite (see Lemma C.0.2 in the appendix) and $K\mathbf{\Gamma}$ is positive semidefinite, matrix $\mathbf{A} - K\mathbf{\Gamma}$ is always invertible.

Velocity measurements, as those provided by a DVL unit, can be easily introduced in the previous formulation with minor modifications. For instance, one way of doing so is by assuming continuous noise-free velocity measurements V_m and considering the energy functional (6.1) with the first term replaced by $\alpha_1 \|\nabla X(s) - V_m(s)\|^2$ as in [89]. The observer equations (6.9) remain unchanged, except for vector $\mathbf{b}$ that must now be defined as $\mathbf{b} = \text{vec}(\mathbf{B}) - \alpha_1 \text{vec}(\mathbf{W})$, with $\mathbf{W}$ representing the finite difference approximation of ∇V_m.

6.2.2 Tuning the trajectory observer

The free parameters in the observer are α_1, α_2 and σ_X. The first two, are also encountered in the original trajectory observers, and weigh the relative penalization of velocities and accelerations. In [46] it is recommended to have parameters of the order Δs^2 for α_1 and of the order Δs^4 for α_2. The introduction of σ_X is the main contribution of the chapter. It provides a simple and intuitive way of tuning the observer. Unlike the original formulation, where the observer gain K had no physical interpretation, the new parameter σ_X is the assumed standard deviation of the position measurement errors. It might be known a priori, from sensor specifications, or roughly estimated from the available data (for instance using a moving average filter). The important fact is that it is possible to specify a priori what will be the size of the mismatch between the available position measurements and the estimated trajectory.

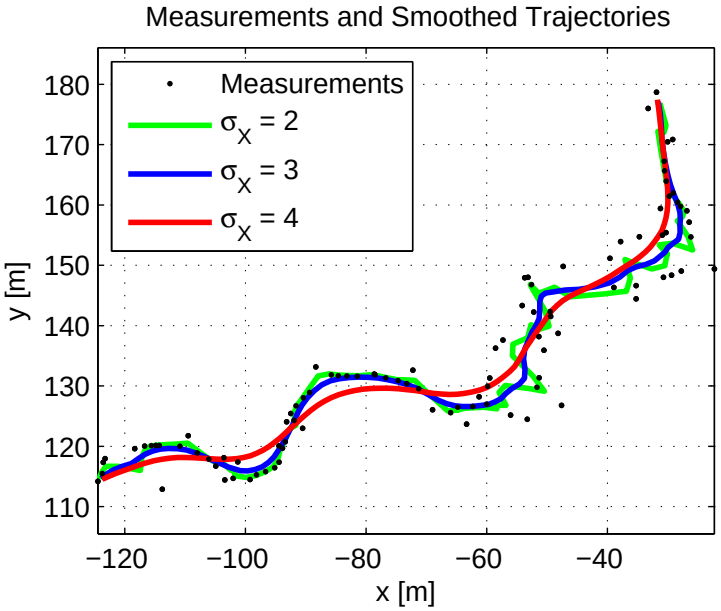


Figure 6.1: Position measurements, and final estimated trajectory using different σ_X values.

6.3 Experimental results

The trajectory observer derived was applied to the post-processing of experimental data from sea trials in Sines, Portugal, in June 2004. Position measurements were generated by trilateration of data coming from an underwater acoustic positioning system [10] while maneuvering an acoustic emitter from a surface ship. The selected trajectory corresponds to $100s$ of data, and position measurements were available every second. The trajectory was discretized in $N = 100$ elements with $\Delta s = 1$s. The observer parameters were set to $\alpha_1 = \alpha_2 = 1$, and $\sigma_X \in \{2, 3, 4\}$m. The observer was initialized at a linear interpolation of the available measurements. The results are shown in Figures 1, 2, and 3. The final trajectory estimates (the steady state solution of the observer equations) are shown in Figure 1, together with the actual position measurements. The bigger the standard deviation σ_X, the smoother the resulting trajectory is, and less weight is given to the measurements. Figure 2 shows the time evolution of the trajectory estimate, which provides a graphical intuition on how the observer behaves. Figure 3 shows the evolution of the resulting observer gain.

It is important to note that before using the derived observer, one should be careful in removing the outliers from the available measurements. Even if the smoothing process will somehow minimize their influence, the observer is not meant to deal with outliers. In [182] an integrated outlier rejection and smoothing scheme is presented.

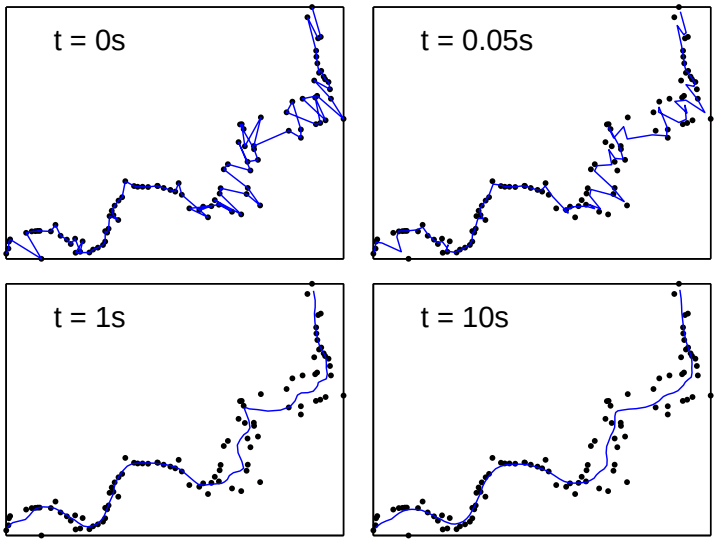


Figure 6.2: Position measurements and diffusion trajectory estimate at different improvement times t corresponding to $\sigma_X = 3m$.

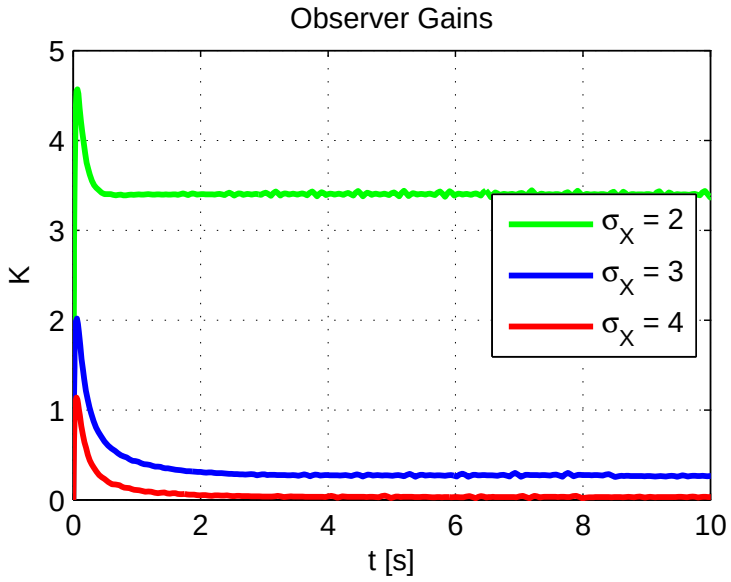


Figure 6.3: Evolution of the observer gain K with the improvement time t for different values of σ_X.

6.4 Conclusions and future work

In this chapter, we proposed an extension to diffusion-based trajectory observers that includes variance constraints on the position measurements. The derived observer can be easily implemented and tuned using physical meaningful parameters. Experimental results with data gathered from an acoustic positioning system were presented to illustrate the observer performance. Convergence analysis showed that under some conditions the observer is asymptotically stable.

There are many issues that could be further addressed. First, some improvement over the use of Dirac delta functions to weight the position measurements might be done. Instead, one could use Gaussian functions whose variance accounts for the small uncertainty present in the time tags of the position measurements. Second, the equality constrained minimization problem (6.6) may not seem natural; it was simply chosen so as to yield a simple solution in the framework of diffusion-based trajectory observers. By considering inequality constraints on the variance, the discrete version of problem could have been formulated as a Quadratically Constrained Quadratic Problem (QCQP). Considering constraints on the variance of the velocity measurements is another direction of further research.

Chapter 7

Conclusions and Future Work

This thesis has addressed several estimation problems relevant to the navigation of underwater robotic vehicles. The problems however, have also practical importance in other different fields such as indoor, urban, and space navigation, whenever GPS measurements are not available, or unreliable, and one wants to use an alternative range-based local navigation system.

Several algorithms to solve the Range-Only localization problem were presented and discussed. Simple expressions for the Hessian of the ML-R and ML-SR cost functions were derived that allow the implementation of fast Newton algorithms which were shown to achieve the Cramér Rao Bound (CRB) for the problem. Their numerical complexity is similar to the LS-GTRS algorithm, which make them an interesting solution to be implemented in embedded devices as their computational power increases. The application of Range-Only estimation methods to a real underwater acoustic positioning system was presented. Several practical issues such as sound speed estimation, multipaths and outlier mitigation were discussed. An Extended Kalman Filter that implicitly addresses the delay in the TOA measurements was derived and tested with experimental data from sea trials.

The thesis also addressed the problem of Range-Only attitude and position estimation. Several algorithms were proposed to solve the problem. The geometric descent algorithms do not require any particular parametrization of rotation matrices and evolves naturally on the Special Euclidean group $SE(3)$. Their performance was shown to be very close to the Intrinsic Variance Lower Bound for a wide range of signal to noise ratios. An alternative solution based on a dynamical system on $SE(3)$ and a suitable Lyapunov function of the range measurements was derived which exhibits asymptotic convergence.

A method to post-process position fixes from an acoustic positioning system was also discussed. n extension of diffusion-based trajectory observers was derived that incorporates knowledge on the size of estimation errors in the form of variance constraints. The method was shown to exhibit asymptotic stability.

There are several topics that were not addressed in the thesis and warrant further

research. Next we give an overview of some of them.

7.1 Single Beacon Navigation

Most of the systems previously discussed exploit the spatial diversity of a set of acoustic emitters/receivers in order to extract position information from range measurements. It is precisely the fact that range are measured with respect to points at different locations that makes it possible to determine a position fix. However, there is an alternative approach using a single beacon, that exploits both the temporal and spatial diversity in order to extract position information.

The question of how many ranges are necessary in order to compute a position fix is of utmost importance. There is no straight answer to this question as there are some important subtle details that may go unnoticed at first. From a purely algebraic point of view, one needs 3 ranges to non co-linear points in the 2-dimensional case and 4 ranges to non coplanar points in the 3-dimensional case to uniquely determine a position fix. The fact is that, in practice, it is not enough to satisfy these algebraic requirements. If one wants to obtain useful navigational data there are other important issues such as having a favorable beacon geometry that come into play.

There is a great interest in reducing the number of beacons involved in the acoustic navigation system, as they usually involve deployment, calibration and recovery time which is money and time consuming. A recurrent question arises: what is the minimum number of beacons that can be used to perform a navigation task? As previously discussed, a single range measurement does not contain enough information to uniquely determine a position, but instead it defines a whole circle (in 2 dimensions) or a sphere (in 3 dimensions) of possible positions. This does not mean, of course, that this information is not useful, but rather that this information alone is not enough to compute a position fix. If the vehicle carries an on-board navigation system capable of performing DR one can use the ranges collected over a time interval in order to correct the DR navigation errors. The locations at which the ranges are acquired act as elements of a virtual beacon array. This suggests the name of Virtual Baseline (VBL) navigation. Of course there are several limitations of this method, including the need of rich and spatially diverse vehicle trajectories, and the need of an accurate DR navigation system.

The concept of underwater navigation using ranges to a single beacon/transponder has received inceasing attention in the marine robotics community. Starting with the work of Larsen who came up with the term Synthetic Long Baseline navigation [104], [105]. Observability is the key issue, and several works have addressed this [161], [177], [69], [134], [70], [147], [90], [49], [68] . More recently, several works have addressed the problem from diverse perspectives, and pointed out its relationship with the multiple vehicle navigation problem [80], [103], [149], [59]. Note that this

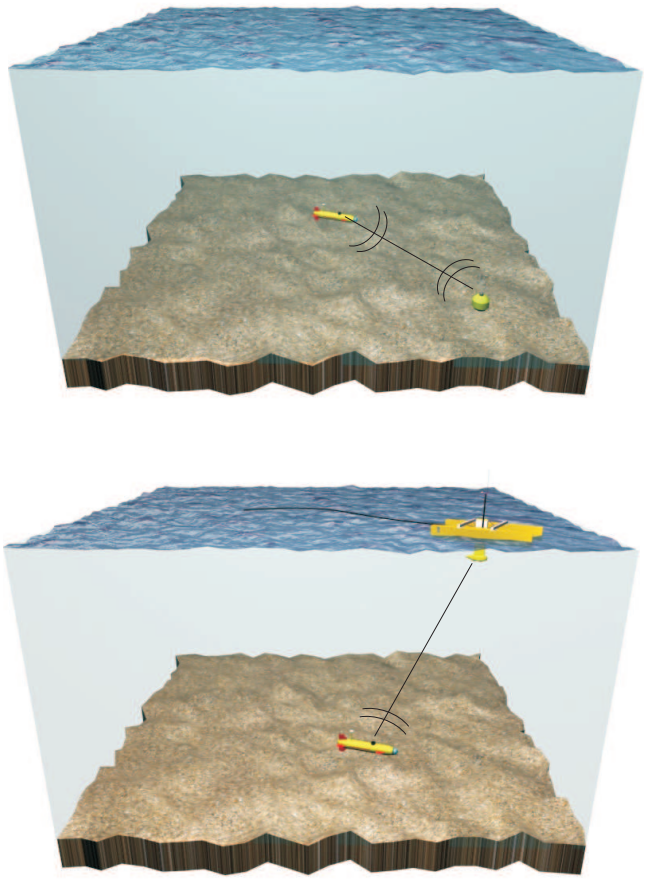


Figure 7.1: Single beacon navigation / tracking

problem is also closely related to the classic source localization problem in underwater acoustics [86].

A dual to this problem is the tracking of an underwater platform with a single range measuring device Fig. 7.1. Instead of static surface buoys as employed in a GIB-like system, one could think of a surface vehicle that, by moving in convenient trajectories, exploits its spatial diversity while measuring ranges to the underwater platform in order to determine its position.

7.2 Cooperative Navigation

One of the drawbacks of some existing underwater acoustic positioning systems is the fact that prior to operation one needs to deploy and calibrate a set of beacons on the sea bed or surface buoys as in a LBL or GIB systems. This often represents considerable operational costs. When operating multiple vehicles on the same area, it is natural to think that sea bed beacons or surface buoys could be replaced by vehicles with certain autonomy, that while navigating, also serve as navigational aid for their neighbors. The problem gets even more interesting if one considers the positive effect of communication exchange among vehicles in order to improve navigation accuracy. By communicating certain information with its neighbors, a group of underwater robots might be able to navigate with higher precision, as compared to each individual vehicle navigation system. In the near future we will probably assist to the development of fleets of underwater vehicles and or possibly some surface vehicles cooperating, exchanging range and position information, some of them popping up to the surface to acquire GPS fixes, and sharing information as to improve the overall fleet navigation performance [52] [188]. In order for this to be done there is a need to study and design distributed navigation algorithms based on range measurements and possibly other data.

The recent years have seen an increasing interest in the use of multiple underwater vehicles. Using multiple vehicles simultaneously instead of an individual one has many potential advantages. For instance, it allows to perform search and survey operations in less time and increase the area coverage. Fleets of underwater gliders have started to be used to gather oceanographic data at an unprecedented scale, and groups of AUVs have been used to perform surveying and de-mining missions successfully. Central to these kind of tasks is the navigation capabilities of each vehicle. Instead of using independent individual vehicle navigation systems, and possibly aided with some classic underwater acoustic positioning system, there has been some interest in using inter vehicle communications and ranging in order to increase the overall group navigation performance. This leads to the concept of cooperative navigation, where there is a synergy between the navigation systems of the multiple vehicles operating simultaneously, allowing them to navigate better than they would do on their own Fig. 7.2. There has been some theoretical and

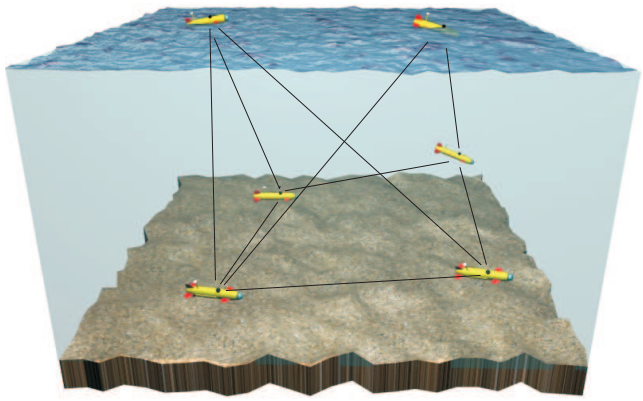


Figure 7.2: Cooperative Acoustic Navigation

experimental work on this direction but it is expected to see many more in the near future [52], [188], [144], [135], [62], [20], [21], [189], [140], [41]. Instead of having a fixed array of beacons/transponders, serving as navigation aids, the Moving Baseline (MBL) concept suggests to use some vehicles as beacons for the others. There are many fundamental theoretical and practical questions that still need to be addressed. Particularly important in the underwater environment is the characterization of the communication requirements in terms of bandwidths and communication topologies that are needed in order to achieve a certain navigation performance [72].

Appendix A

Matrix Differential Calculus and Derivations

In this chapter we will briefly introduce Matrix Differential Calculus, a technique that allows to compute gradients and Hessians of vector and matrix valued functions in a simple way. Then, this technique will be used to determine the gradients and Hessians of the different Maximum Likelihood cost functions that were introduced in the previous chapters.

For a complete introduction to the topic of Matrix Differential Calculus and its applications please refer to the book by Magnus and Neudecker [117].

A.1 Matrix Differential Calculus

In many occasions, one is faced with the problem of determining the gradient and the Hessian of scalar valued functions with vector or matrix parameters. For some of the most common functions there are available results such as those in [115]. However in some cases it is difficult to use those results since they are not intuitive and the user is often confused. Matrix differential calculus provides a simple method, starting from basic principles that both gives intuition about the concepts of differentiation and helps to determine gradients and Hessians of complex functions.

Most of the standard rules of scalar differentiation also extend to the case of vectors and matrix differentials. This together with the extensive use of the vec() operator to represent matrices as vectors, and the Kronecker product rule $\text{vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A})\text{vec}(\mathbf{B})$, lead to a set of simple basic rules from which to derive complex results.

Some basic differential rules include:

$$\mathrm{d}\mathbf{C} = 0, \tag{A.1}$$
$$\mathrm{d}(\alpha\mathbf{X}) = \alpha\mathrm{d}\mathbf{X}, \tag{A.2}$$
$$\mathrm{d}(\mathbf{X}+\mathbf{Y}) = \mathrm{d}\mathbf{X} + \mathrm{d}\mathbf{Y}, \tag{A.3}$$
$$\mathrm{d}(\mathbf{X}^T\mathbf{Y}) = \mathrm{d}\mathbf{X}^T\mathbf{Y} + \mathbf{X}^T\mathrm{d}\mathbf{Y}, \tag{A.4}$$
$$\mathrm{d}\mathrm{tr}\left(\mathbf{X}\right) = \mathrm{tr}\left(\mathrm{d}\mathbf{X}\right). \tag{A.5}$$

According to the *First Identification Theorem* in [117], suppose that $\phi : \mathbb{R}^n \to \mathbb{R}$ is a scalar valued function of vector variable and its first differential satisfies

$$\mathrm{d}\phi(\mathbf{x}) = \mathbf{a}^T\mathrm{d}\mathbf{x}, \tag{A.6}$$

where $\mathbf{a} \in \mathbb{R}^n$. Then the gradient of ϕ denoted $\nabla\phi$ is given by

$$\nabla\phi(\mathbf{x}) = \mathbf{a}. \tag{A.7}$$

The Hessian matrix can be found from the *Second Identification Theorem* in [117], which states that if the second differential of ϕ satisfies

$$\mathrm{d}^2\phi(\mathbf{x}) = \mathrm{d}\mathbf{x}^T\mathbf{A}\mathrm{d}\mathbf{x}, \tag{A.8}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$, then the Hessian of ϕ, denoted $\nabla^2\phi(\mathbf{x})$ is given by

$$\nabla^2\phi(\mathbf{x}) = \mathbf{A}.$$

A.2 Gradients and Hessians of ML cost functions for the Range-Only positioning problem

A.2.1 Gradient and Hessian of the MLR cost function

Proof of Proposition 2.4.1. We need to determine the gradient and Hessian of the function $f : \mathbb{R}^n \to \mathbb{R}$ given by

$$f(\mathbf{x}) := \frac{1}{2}(\bar{\mathbf{r}} - \mathbf{r}(\mathbf{x}))^T\mathbf{R}^{-1}(\bar{\mathbf{r}} - \mathbf{r}(\mathbf{x})), \tag{A.9}$$

where $\mathbf{r}(\mathbf{x}) = [r_1(\mathbf{x}) \ \ldots \ r_m(\mathbf{x})]^T \in \mathbb{R}^m$ is the vector of range measurements between $\mathbf{x}$ and the landmarks with coordinates $\mathbf{p}_i$. That is, with components $r_i(\mathbf{x}) = \|\mathbf{x}-\mathbf{p}_i\|$, $i \in \{1, \ldots, m\}$.

To begin we need to compute the first differential of f

$$\mathrm{d}f(\mathbf{x}) = \frac{1}{2}\mathrm{d}(\bar{\mathbf{r}} - \mathbf{r}(\mathbf{x}))^T\mathbf{R}^{-1}(\bar{\mathbf{r}} - \mathbf{r}(\mathbf{x})) + \frac{1}{2}(\bar{\mathbf{r}} - \mathbf{r}(\mathbf{x}))^T\mathbf{R}^{-1}\mathrm{d}(\bar{\mathbf{r}} - \mathbf{r}(\mathbf{x})) \tag{A.10}$$

$$= -\frac{1}{2}\mathrm{d}\mathbf{r}(\mathbf{x})^T\mathbf{R}^{-1}(\bar{\mathbf{r}} - \mathbf{r}(\mathbf{x})) - \frac{1}{2}(\bar{\mathbf{r}} - \mathbf{r}(\mathbf{x}))^T\mathbf{R}^{-1}\mathrm{d}\mathbf{r}(\mathbf{x}) \tag{A.11}$$

$$= -(\bar{\mathbf{r}} - \mathbf{r}(\mathbf{x}))^T\mathbf{R}^{-1}\mathrm{d}\mathbf{r}(\mathbf{x}). \tag{A.12}$$

The differential of $\mathbf{r}(\mathbf{x})$ is given by the differential of its components, that is,

$$\mathrm{d}\mathbf{r}(\mathbf{x}) = \begin{bmatrix} \mathrm{d}r_1(\mathbf{x}) \\ \vdots \\ \mathrm{d}r_m(\mathbf{x}) \end{bmatrix} \in \mathbb{R}^m, \tag{A.13}$$

where

$$\mathrm{d}r_i(\mathbf{x}) = \mathrm{d}\|\mathbf{x} - \mathbf{p}_i\| = \mathrm{d}\Big((\mathbf{x} - \mathbf{p}_i)^T(\mathbf{x} - \mathbf{p}_i)\Big)^{\frac{1}{2}} \tag{A.14}$$

$$= \frac{1}{2}\Big((\mathbf{x} - \mathbf{p}_i)^T(\mathbf{x} - \mathbf{p}_i)\Big)^{-\frac{1}{2}}\mathrm{d}\Big((\mathbf{x} - \mathbf{p}_i)^T(\mathbf{x} - \mathbf{p}_i)\Big) \tag{A.15}$$

$$= \frac{1}{2}\frac{1}{r_i(\mathbf{x})}\Big(\mathrm{d}\mathbf{x}^T(\mathbf{x} - \mathbf{p}_i) + (\mathbf{x} - \mathbf{p}_i)^T\mathrm{d}\mathbf{x}\Big) \tag{A.16}$$

$$= \frac{1}{2}\frac{1}{r_i(\mathbf{x})}2(\mathbf{x} - \mathbf{p}_i)^T\mathrm{d}\mathbf{x} \tag{A.17}$$

$$= \frac{1}{r_i(\mathbf{x})}(\mathbf{x} - \mathbf{p}_i)^T\mathrm{d}\mathbf{x}. \tag{A.18}$$

Then,

$$\mathrm{d}\mathbf{r}(\mathbf{x}) = \begin{bmatrix} \mathrm{d}r_1(\mathbf{x}) \\ \vdots \\ \mathrm{d}r_m(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \frac{1}{r_1(\mathbf{x})}(\mathbf{x} - \mathbf{p}_1)^T\mathrm{d}\mathbf{x} \\ \vdots \\ \frac{1}{r_m(\mathbf{x})}(\mathbf{x} - \mathbf{p}_m)^T\mathrm{d}\mathbf{x} \end{bmatrix} \tag{A.19}$$

$$= \begin{bmatrix} \frac{1}{r_1(\mathbf{x})} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{r_m(\mathbf{x})} \end{bmatrix} \begin{bmatrix} (\mathbf{x} - \mathbf{p}_1)^T \\ \vdots \\ (\mathbf{x} - \mathbf{p}_m)^T \end{bmatrix} \mathrm{d}\mathbf{x} \tag{A.20}$$

$$= \delta(\mathbf{r}(\mathbf{x}))^{-1}\mathbf{C}^T\mathrm{d}\mathbf{x}, \tag{A.21}$$

and

$$\mathrm{d}f(\mathbf{x}) = -(\bar{\mathbf{r}} - \mathbf{r}(\mathbf{x}))^T\mathbf{R}^{-1}\mathrm{d}\mathbf{r}(\mathbf{x}) = -(\bar{\mathbf{r}} - \mathbf{r}(\mathbf{x}))^T\mathbf{R}^{-1}\delta(\mathbf{r}(\mathbf{x}))^{-1}\mathbf{C}^T\mathrm{d}\mathbf{x}, \tag{A.22}$$

which has the form $\mathrm{d}f(\mathbf{x}) = \mathbf{a}^T\mathrm{d}\mathbf{x}$. According to the *First Identification Theorem* in [117], the gradient of the MLR cost function is given by

$$\nabla f(\mathbf{x}) = -\mathbf{C}\delta(\mathbf{r}(\mathbf{x}))^{-1}\mathbf{R}^{-1}(\bar{\mathbf{r}} - \mathbf{r}(\mathbf{x})). \tag{A.23}$$

In order to determine the Hessian of the MLR function we first need to determine its second differential. Define

$$\boldsymbol{\alpha} = \mathbf{R}^{-1}(\bar{\mathbf{r}} - \mathbf{r}(\mathbf{x})) = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{bmatrix} \in \mathbb{R}^m.$$

Then we can rewrite the first differential as

$$\begin{aligned} \mathrm{d}f(\mathbf{x}) &= -(\bar{\mathbf{r}} - \mathbf{r}(\mathbf{x}))^T \mathbf{R}^{-1} \mathrm{d}\mathbf{r}(\mathbf{x}) = -\boldsymbol{\alpha}^T \mathrm{d}\mathbf{r}(\mathbf{x}) \\ &= -\sum_{i=1}^{m} \alpha_i \, \mathrm{d}r_i(\mathbf{x}). \end{aligned}$$

This way of expressing $\mathrm{d}f(\mathbf{x})$ is convenient in order to compute the second differential. We have that

$$\begin{aligned} \mathrm{d}^2 f(\mathbf{x}) &= -\sum_{i=1}^{m} \mathrm{d}\alpha_i \; \mathrm{d}r_i - \sum_{i=1}^{m} \alpha_{ij} \; \mathrm{d}^2 r_i \\ &= -\mathrm{d}\boldsymbol{\alpha}^T \mathrm{d}\mathbf{r} - \sum_{i=1}^{m} \alpha_i \; \mathrm{d}^2 r_i \\ &= \mathrm{d}\mathbf{r}^T \mathbf{R}^{-1} \mathrm{d}\mathbf{r} - \sum_{i=1}^{m} \alpha_i \; \mathrm{d}^2 r_i \qquad \text{(A.24)} \\ &= \mathrm{d}\mathbf{x}^T \Big(\mathbf{C}\delta(\mathbf{r})^{-1} \mathbf{R}^{-1} \delta(\mathbf{r})^{-1} \mathbf{C}^T \Big) \mathrm{d}\mathbf{x} - \sum_{i=1}^{m} \alpha_i \; \mathrm{d}^2 r_i, \qquad \text{(A.25)} \end{aligned}$$

so it remains to compute $\mathrm{d}^2 r_i$:

$$\begin{aligned} \mathrm{d}^2 r_i = \mathrm{d}(\mathrm{d}r_i) &= \mathrm{d}\Big(\frac{1}{r_i} (\mathbf{x} - \mathbf{p}_i)^T \mathrm{d}\mathbf{x} \Big) \qquad \text{(A.26)} \\ &= \frac{1}{r_i} \mathrm{d}\mathbf{x}^T \mathrm{d}\mathbf{x} - \frac{1}{r_i^2} \mathrm{d}r_i (\mathbf{x} - \mathbf{p}_i)^T \mathrm{d}\mathbf{x} \qquad \text{(A.27)} \\ &= \frac{1}{r_i} \mathrm{d}\mathbf{x}^T \mathrm{d}\mathbf{x} - \frac{1}{r_i^3} (\mathbf{x} - \mathbf{p}_i)^T \mathrm{d}\mathbf{x} (\mathbf{x} - \mathbf{p}_i)^T \mathrm{d}\mathbf{x}. \qquad \text{(A.28)} \end{aligned}$$

Now, re-arranging the elements of the second term in (A.25) we can obtain

$$\begin{aligned} \sum_{i=1}^{m} \alpha_i \; \mathrm{d}^2 r_i &= \sum_{i=1}^{m} \alpha_i \frac{1}{r_i} \mathrm{d}\mathbf{x}^T \mathrm{d}\mathbf{x} - \alpha_i \frac{1}{r_i^3} (\mathbf{x} - \mathbf{p}_i)^T \mathrm{d}\mathbf{x} (\mathbf{x} - \mathbf{p}_i)^T \mathrm{d}\mathbf{x} \\ &= \mathrm{d}\mathbf{x}^T \left(\sum_{i=1}^{m} \frac{\alpha_i}{r_i} \mathbf{I}_n - \frac{\alpha_i}{r_i^3} (\mathbf{x} - \mathbf{p}_i)(\mathbf{x} - \mathbf{p}_i)^T \right) \mathrm{d}\mathbf{x} \qquad \text{(A.29)} \\ &= \mathrm{d}\mathbf{x}^T \left(\boldsymbol{\alpha}^T \delta(\mathbf{r})^{-1} \mathbf{1}_m \mathbf{I}_n - \mathbf{C}\delta(\mathbf{r})^{-3} \delta(\boldsymbol{\alpha}) \mathbf{C}^T \right) \mathrm{d}\mathbf{x}. \qquad \text{(A.30)} \end{aligned}$$

The second differential can then be written as $\mathrm{d}^2 f(\mathbf{x}) = \mathrm{d}\mathbf{x}^T \mathbf{A} \mathrm{d}\mathbf{x}$ where

$$\begin{aligned}\mathbf{A} &= \mathbf{C}\delta(\mathbf{r})^{-1}\mathbf{R}^{-1}\delta(\mathbf{r})^{-1}\mathbf{C}^T + \boldsymbol{\alpha}^T\delta(\mathbf{r})^{-1}\mathbf{1}_m\mathbf{I}_n - \mathbf{C}\delta(\mathbf{r})^{-3}\delta(\boldsymbol{\alpha})\mathbf{C}^T \\ &= \mathbf{C}\delta(\mathbf{r})^{-1}\Big(\mathbf{R}^{-1} - \delta(\mathbf{r})^{-1}\delta(\boldsymbol{\alpha})\Big)\delta(\mathbf{r})^{-1}\mathbf{C}^T + \boldsymbol{\alpha}^T\delta(\mathbf{r})^{-1}\mathbf{1}_m\mathbf{I}_n, \end{aligned} \tag{A.31}$$

and according to the *Second Identification Theorem* in [117] the Hessian satisfies $\nabla^2 f(\mathbf{x}) = \mathbf{A}$ as stated in (2.83). When the measurement error covariance matrix $\mathbf{R}$ is diagonal, simple computations yield the result in (2.85) which concludes the proof. □

A.2.2 Gradient and Hessian of the MLSR cost function

Proof of Proposition 2.4.2. We need to determine the gradient and Hessian of the function $f : \mathbb{R}^n \to \mathbb{R}$ given by

$$f(\mathbf{x}) := \frac{1}{2}(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x}))^T \boldsymbol{\Sigma}^{-1}(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x})), \tag{A.32}$$

where $\mathbf{d}(\mathbf{x}) = [d_1(\mathbf{x}) \ \dots \ d_m(\mathbf{x})]^T \in \mathbb{R}^m$ is the vector of squared range measurements between $\mathbf{x}$ and the landmarks with coordinates $\mathbf{p}_i$. That is, with components $d_i(\mathbf{x}) = r_i(\mathbf{x})^2 = \|\mathbf{x} - \mathbf{p}_i\|^2$, $i \in \{1, \dots, m\}$.

To begin we need to compute the first differential of f

$$\mathrm{d}f(\mathbf{x}) = -(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x}))^T \boldsymbol{\Sigma}^{-1} \mathrm{d}\mathbf{d}(\mathbf{x}). \tag{A.33}$$

The differential of $\mathbf{d}(\mathbf{x})$ is given by the differential of its components

$$\begin{aligned}\mathrm{d}d_i(\mathbf{x}) &= \mathrm{d}\|\mathbf{x} - \mathbf{p}_i\|^2 = \mathrm{d}\Big((\mathbf{x} - \mathbf{p}_i)^T(\mathbf{x} - \mathbf{p}_i)\Big) \\ &= \mathrm{d}\mathbf{x}^T(\mathbf{x} - \mathbf{p}_i) + (\mathbf{x} - \mathbf{p}_i)^T\mathrm{d}\mathbf{x} \\ &= 2(\mathbf{x} - \mathbf{p}_i)^T\mathrm{d}\mathbf{x}, \end{aligned} \tag{A.34}$$

and can be written as

$$\mathrm{d}\mathbf{d}(\mathbf{x}) = \begin{bmatrix} \mathrm{d}d_1(\mathbf{x}) \\ \vdots \\ \mathrm{d}d_m(\mathbf{x}) \end{bmatrix} = 2\begin{bmatrix} (\mathbf{x} - \mathbf{p}_1)^T \\ \vdots \\ (\mathbf{x} - \mathbf{p}_m)^T \end{bmatrix} \mathrm{d}\mathbf{x} = 2\mathbf{C}^T\mathrm{d}\mathbf{x}. \tag{A.35}$$

Then, the first differential becomes

$$\mathrm{d}f(\mathbf{x}) = -(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x}))^T \boldsymbol{\Sigma}^{-1}\mathrm{d}\mathbf{d}(\mathbf{x}) = -2(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x}))^T \boldsymbol{\Sigma}^{-1}\mathbf{C}^T\mathrm{d}\mathbf{x}, \tag{A.36}$$

and the gradient of the MLR cost function is given by

$$\nabla f(\mathbf{x}) = -2\mathbf{C}\boldsymbol{\Sigma}^{-1}(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x})), \tag{A.37}$$

as stated in (2.87).

Define vector $\boldsymbol{\alpha} = \boldsymbol{\Sigma}^{-1}(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x})) = [\alpha_1, \ \dots \ , \alpha_m]^T$. Then we can rewrite the first differential as

$$\mathrm{d}f(\mathbf{x}) = -(\bar{\mathbf{d}} - \mathbf{d}(\mathbf{x}))^T \boldsymbol{\Sigma}^{-1} \mathrm{d}\mathbf{d}(\mathbf{x}) = -\boldsymbol{\alpha}^T \mathrm{d}\mathbf{d}(\mathbf{x}) = -\sum_{i=1}^{m} \alpha_i \, \mathrm{d}d_i(\mathbf{x}).$$

This way of expressing $\mathrm{d}f(\mathbf{x})$ is convenient in order to compute the second differential. We have that

$$\begin{aligned} \mathrm{d}^2 f(\mathbf{x}) &= -\sum_{i=1}^{m} \mathrm{d}\alpha_i \ \mathrm{d}d_i - \sum_{i=1}^{m} \alpha_{ij} \ \mathrm{d}^2 d_i = -\mathrm{d}\boldsymbol{\alpha}^T \mathrm{d}\mathbf{d} - \sum_{i=1}^{m} \alpha_i \ \mathrm{d}^2 d_i \\ &= \mathrm{d}\mathbf{d}^T \boldsymbol{\Sigma}^{-1} \mathrm{d}\mathbf{d} - \sum_{i=1}^{m} \alpha_i \ \mathrm{d}^2 d_i = \mathrm{d}\mathbf{x}^T \Big(4\mathbf{C}\boldsymbol{\Sigma}^{-1}\mathbf{C}^T \Big) \mathrm{d}\mathbf{x} - \sum_{i=1}^{m} \alpha_i \ \mathrm{d}^2 d_i, \end{aligned} \tag{A.38}$$

so it remains to compute $\mathrm{d}^2 d_i$:

$$\mathrm{d}^2 d_i = \mathrm{d}(\mathrm{d}d_i) = \mathrm{d}\Big(2(\mathbf{x} - \mathbf{p}_i)^T \mathrm{d}\mathbf{x} \Big) = 2\mathrm{d}\mathbf{x}^T \mathrm{d}\mathbf{x}. \tag{A.39}$$

The second differential then becomes

$$\begin{aligned} \mathrm{d}^2 f(\mathbf{x}) &= \mathrm{d}\mathbf{x}^T \Big(4\mathbf{C}\boldsymbol{\Sigma}^{-1}\mathbf{C}^T - 2\sum_{i=1}^{m} \alpha_i \mathbf{I}_n \Big) \mathrm{d}\mathbf{x} \\ &= \mathrm{d}\mathbf{x}^T \Big(4\mathbf{C}\boldsymbol{\Sigma}^{-1}\mathbf{C}^T - 2\boldsymbol{\alpha}^T \mathbf{1}_m \mathbf{I}_n \Big) \mathrm{d}\mathbf{x}, \end{aligned} \tag{A.40}$$

and the Hessian

$$\begin{aligned} \nabla^2 f(\mathbf{x}) &= 4\mathbf{C}\boldsymbol{\Sigma}^{-1}\mathbf{C}^T - 2\boldsymbol{\alpha}^T \mathbf{1}_m \mathbf{I}_n \\ &= 4\mathbf{C}\boldsymbol{\Sigma}^{-1}\mathbf{C}^T - 2(\bar{\mathbf{d}} - \mathbf{d})^T \boldsymbol{\Sigma}^{-1} \mathbf{1}_m \mathbf{I}_n, \end{aligned} \tag{A.41}$$

as stated in (2.88) □

A.3 Gradients and Hessians of Maximum Likelihood cost functions for the Range-Only Attitude and Positioning problem

A.3.1 Gradient and Hessian of the MLR cost function

Proof of Proposition 4.3.3. We need to determine the extrinsic gradient of the function $f : SE(3) \to \mathbb{R}$ given by

$$f(\boldsymbol{\theta}) := \frac{1}{2} (\bar{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta}))^T \mathbf{R}^{-1} (\bar{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta})), \tag{A.42}$$

where $\mathbf{r}(\boldsymbol{\theta}) = [r_{11}(\boldsymbol{\theta}) \ \dots \ r_{pm}(\boldsymbol{\theta})]^T \in \mathbb{R}^{mp}$ is the vector of range measurements between the beacons solidary to the rigid body with configuration $\boldsymbol{\theta} = (\mathcal{R}, \mathbf{p}) \in SE(3)$ and the landmarks with coordinates $\mathbf{p}_i$. That is, $\mathbf{r}(\boldsymbol{\theta})$ has components $r_i(\boldsymbol{\theta}) = \|\mathcal{R}\mathbf{b}_i + \mathbf{p} - \mathbf{p}_j\|$, $i \in \{1, \dots, p\}, j \in \{1, \dots, m\}$.

We will determine the gradient and Hessian of the similar function $\widetilde{f} : \mathbb{R}^{12} \to \mathbb{R}$, where given $\mathbf{x} = [\mathrm{vec}(\mathcal{R})^T \ \mathbf{p}^T]^T \in \mathbb{R}^{12}$, $\widetilde{f}(\mathbf{x}) = f(\mathcal{R}, \mathbf{p})$. To begin we need to compute the first differential of $\widetilde{f}$

$$\mathrm{d}\widetilde{f}(\boldsymbol{\theta}) = -(\bar{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta}))^T \mathbf{R}^{-1} \mathrm{d}\mathbf{r}(\boldsymbol{\theta}). \tag{A.43}$$

The differential of $\mathbf{r}(\boldsymbol{\theta})$ is given by the differential of its components, that is,

$$\mathrm{d}\mathbf{r}(\boldsymbol{\theta}) = \begin{bmatrix} \mathrm{d}r_{11}(\boldsymbol{\theta}) \\ \mathrm{d}r_{21}(\boldsymbol{\theta}) \\ \vdots \\ \mathrm{d}r_{pm}(\boldsymbol{\theta}) \end{bmatrix} \in \mathbb{R}^{pm}, \tag{A.44}$$

where

$$\begin{aligned}
\mathrm{d}r_{ij}(\boldsymbol{\theta}) &= \mathrm{d}\|\mathcal{R}\mathbf{b}_i + \mathbf{p} - \mathbf{p}_j\| = \mathrm{d}\Big((\mathcal{R}\mathbf{b}_i + \mathbf{p} - \mathbf{p}_j)^T(\mathcal{R}\mathbf{b}_i + \mathbf{p} - \mathbf{p}_j)\Big)^{\frac{1}{2}} \\
&= \frac{1}{2r_{ij}} \mathrm{d}\Big(\mathbf{b}_i^T\mathbf{b}_i + 2\mathbf{b}_i^T\mathcal{R}^T(\mathbf{p} - \mathbf{p}_j) + (\mathbf{p} - \mathbf{p}_j)^T(\mathbf{p} - \mathbf{p}_j)\Big) \\
&= \frac{1}{2r_{ij}} \Big(2\mathbf{b}_i^T \mathrm{d}\mathcal{R}^T(\mathbf{p} - \mathbf{p}_j) + 2\mathbf{b}_i^T\mathcal{R}^T\mathrm{d}\mathbf{p} + 2(\mathbf{p} - \mathbf{p}_j)^T\mathrm{d}\mathbf{p}\Big) \\
&= \frac{1}{r_{ij}} \Big((\mathbf{p} - \mathbf{p}_j)^T \mathrm{d}\mathcal{R}\mathbf{b}_i + \mathbf{b}_i^T\mathcal{R}^T\mathrm{d}\mathbf{p} + (\mathbf{p} - \mathbf{p}_j)^T\mathrm{d}\mathbf{p}\Big) \\
&= \frac{1}{r_{ij}} \Big(\mathbf{b}_i^T \otimes (\mathbf{p} - \mathbf{p}_j)^T\Big)\mathrm{d}\mathrm{vec}(\mathcal{R}) + \frac{1}{r_{ij}} \Big(\mathbf{b}_i^T\mathcal{R}^T + (\mathbf{p} - \mathbf{p}_j)^T\Big)\mathrm{d}\mathbf{p}.
\end{aligned} \tag{A.45}$$

Defining $\mathbf{q} = \mathrm{vec}(\mathcal{R}) \in \mathbb{R}^{12}$ we can write

$$\begin{aligned}
\mathrm{d}\mathbf{r}(\boldsymbol{\theta}) &= \begin{bmatrix} \frac{1}{r_{11}}\Big(\mathbf{b}_1^T \otimes (\mathbf{p} - \mathbf{p}_1)^T\Big) \\ \frac{1}{r_{21}}\Big(\mathbf{b}_2^T \otimes (\mathbf{p} - \mathbf{p}_1)^T\Big) \\ \vdots \\ \frac{1}{r_{pm}}\Big(\mathbf{b}_p^T \otimes (\mathbf{p} - \mathbf{p}_m)^T\Big) \end{bmatrix} \mathrm{d}\mathbf{q} + \begin{bmatrix} \frac{1}{r_{11}}\Big(\mathbf{b}_1^T\mathcal{R}^T + (\mathbf{p} - \mathbf{p}_1)^T\Big) \\ \frac{1}{r_{21}}\Big(\mathbf{b}_2^T\mathcal{R}^T + (\mathbf{p} - \mathbf{p}_1)^T\Big) \\ \vdots \\ \frac{1}{r_{pm}}\Big(\mathbf{b}_p^T\mathcal{R}^T + (\mathbf{p} - \mathbf{p}_m)^T\Big) \end{bmatrix} \mathrm{d}\mathbf{p} \\
&:= \delta(\mathbf{r})^{-1}\mathbf{F}\mathrm{d}\mathbf{q} + \delta(\mathbf{r})^{-1}\mathbf{C}\mathrm{d}\mathbf{p} \\
&= \delta(\mathbf{r})^{-1} \begin{bmatrix} \mathbf{F} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathrm{d}\mathbf{q} \\ \mathrm{d}\mathbf{p} \end{bmatrix},
\end{aligned} \tag{A.46}$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{b}_1^T \otimes (\mathbf{p} - \mathbf{p}_1)^T \\ \mathbf{b}_2^T \otimes (\mathbf{p} - \mathbf{p}_1)^T \\ \vdots \\ \mathbf{b}_p^T \otimes (\mathbf{p} - \mathbf{p}_m)^T \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \mathbf{b}_1^T \mathcal{R}^T + (\mathbf{p} - \mathbf{p}_1)^T \\ \mathbf{b}_2^T \mathcal{R}^T + (\mathbf{p} - \mathbf{p}_1)^T \\ \vdots \\ \mathbf{b}_p^T \mathcal{R}^T + (\mathbf{p} - \mathbf{p}_m)^T \end{bmatrix}.$$

Simple manipulations yield to the more compact expressions

$$\mathbf{F} := \begin{bmatrix} \mathbf{B}^T \otimes (\mathbf{p} - \mathbf{p}_1)^T \\ \vdots \\ \mathbf{B}^T \otimes (\mathbf{p} - \mathbf{p}_m)^T \end{bmatrix} \in \mathbb{R}^{mp \times 9}, \tag{A.47}$$

$$\mathbf{C} := \mathbf{1}_m \otimes \mathbf{B}^T \mathcal{R} + (\mathbf{1}_m \mathbf{p}^T - \mathbf{P}^T) \otimes \mathbf{1}_p \in \mathbb{R}^{mp \times 3}. \tag{A.48}$$

The first differential can then be written as

$$\begin{aligned} \mathrm{d}\widetilde{f}(\boldsymbol{\theta}) &= -(\bar{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta}))^T \mathbf{R}^{-1} \mathrm{d}\mathbf{r}(\boldsymbol{\theta}) \\ &= -(\bar{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta}))^T \mathbf{R}^{-1} \delta(\mathbf{r})^{-1} \begin{bmatrix} \mathbf{F} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathrm{d}\mathbf{q} \\ \mathrm{d}\mathbf{p} \end{bmatrix}, \end{aligned} \tag{A.49}$$

and the extrinsic gradient as

$$\nabla \widetilde{f}(\boldsymbol{\theta}) := - \begin{bmatrix} \mathbf{F}^T \delta(\mathbf{r}(\boldsymbol{\theta}))^{-1} \mathbf{R}^{-1} (\bar{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta})) \\ \mathbf{C}^T \delta(\mathbf{r}(\boldsymbol{\theta}))^{-1} \mathbf{R}^{-1} (\bar{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta})) \end{bmatrix} \in \mathbb{R}^{12}, \tag{A.50}$$

which yields the desired result. □

Proof of Proposition 4.3.6. Define $\boldsymbol{\alpha} = \mathbf{R}^{-1}(\bar{\mathbf{r}} - \mathbf{r}(\boldsymbol{\theta}))$. The first differential can also be written as

$$\mathrm{d}\widetilde{f}(\boldsymbol{\theta}) = -\boldsymbol{\alpha}^T \mathbf{R}^{-1} \mathrm{d}\mathbf{r}(\boldsymbol{\theta}) = -\sum_{i,j=1}^{p,m} \alpha_i \mathrm{d}r_{ij}. \tag{A.51}$$

The second differential can be written as

$$\mathrm{d}^2 \widetilde{f}(\boldsymbol{\theta}) = \mathrm{d}\mathbf{r}(\boldsymbol{\theta}) \mathbf{R}^{-1} \mathrm{d}\mathbf{r}(\boldsymbol{\theta}) - \sum_{i,j=1}^{p,m} \alpha_i \mathrm{d}^2 r_{ij}. \tag{A.52}$$

From (A.45) we can write $\mathrm{d}r_{ij} = \frac{1}{r_{ij}} \beta$ where

$$\beta = (\mathbf{p} - \mathbf{p}_j)^T \mathrm{d}\mathcal{R} \mathbf{b}_i + \mathbf{b}_i^T \mathcal{R}^T \mathrm{d}\mathbf{p} + (\mathbf{p} - \mathbf{p}_j)^T \mathrm{d}\mathbf{p}$$

Then,

$$\mathrm{d}^2 r_{ij} = \frac{1}{r_{ij}}\mathrm{d}\beta - \frac{1}{r_{ij}^2}\beta \mathrm{d}r_{ij} = \frac{1}{r_{ij}}\mathrm{d}\beta - \frac{1}{r_{ij}^3}\beta^2.$$

Now we only need to determine $\mathrm{d}\beta$ and β^2. We have

$$\begin{aligned}
\mathrm{d}\beta &= \mathrm{d}\Big((\mathbf{p}-\mathbf{p}_j)^T \mathrm{d}\mathcal{R}\mathbf{b}_i + \mathbf{b}_i^T \mathcal{R}^T \mathrm{d}\mathbf{p} + (\mathbf{p}-\mathbf{p}_j)^T \mathrm{d}\mathbf{p}\Big) \\
&= \mathrm{d}\mathbf{p}^T \mathrm{d}\mathcal{R}\mathbf{b}_i + \mathbf{b}_i^T \mathrm{d}\mathcal{R}^T \mathrm{d}\mathbf{p} + \mathrm{d}\mathbf{p}^T \mathrm{d}\mathbf{p} \\
&= 2\mathrm{d}\mathbf{p}^T \mathrm{d}\mathcal{R}\mathbf{b}_i + \mathrm{d}\mathbf{p}^T \mathrm{d}\mathbf{p} \\
&= 2\mathrm{d}\mathbf{p}^T (\mathbf{b}_i^T \otimes \mathbf{I}_3)\mathrm{d}\mathbf{q} + \mathrm{d}\mathbf{p}^T \mathrm{d}\mathbf{p} \\
&= \begin{bmatrix} \mathrm{d}\mathbf{q}^T & \mathrm{d}\mathbf{p}^T \end{bmatrix} \begin{bmatrix} 0 & (\mathbf{b}_i \otimes \mathbf{I}_3) \\ (\mathbf{b}_i^T \otimes \mathbf{I}_3) & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathrm{d}\mathbf{q} \\ \mathrm{d}\mathbf{p} \end{bmatrix},
\end{aligned}$$

and

$$\begin{aligned}
\beta^2 &= \Big((\mathbf{p}-\mathbf{p}_j)^T \mathrm{d}\mathcal{R}\mathbf{b}_i + \mathbf{b}_i^T \mathcal{R}^T \mathrm{d}\mathbf{p} + (\mathbf{p}-\mathbf{p}_j)^T \mathrm{d}\mathbf{p}\Big)^2 \\
&= \mathbf{b}_i^T \mathrm{d}\mathcal{R}^T (\mathbf{p}-\mathbf{p}_j)(\mathbf{p}-\mathbf{p}_j)^T \mathrm{d}\mathcal{R}\mathbf{b}_i + \mathrm{d}\mathbf{p}^T \mathcal{R}\mathbf{b}_i \mathbf{b}_i^T \mathcal{R}^T \mathrm{d}\mathbf{p} + \mathrm{d}\mathbf{p}^T (\mathbf{p}-\mathbf{p}_j)(\mathbf{p}-\mathbf{p}_j)^T \mathrm{d}\mathbf{p} \\
&\quad + 2\mathrm{d}\mathbf{p}^T \mathcal{R}\mathbf{b}_i (\mathbf{p}-\mathbf{p}_j)^T \mathrm{d}\mathcal{R}\mathbf{b}_i \\
&\quad + 2\mathrm{d}\mathbf{p}^T (\mathbf{p}-\mathbf{p}_j)(\mathbf{p}-\mathbf{p}_j)^T \mathrm{d}\mathcal{R}\mathbf{b}_i \\
&\quad + 2\mathrm{d}\mathbf{p}^T \mathcal{R}\mathbf{b}_i (\mathbf{p}-\mathbf{p}_j)^T \mathrm{d}\mathbf{p} \\
&= \mathrm{d}\mathbf{q}^T \left\{(\mathbf{b}_i \otimes \mathbf{I}_3)(\mathbf{p}-\mathbf{p}_j)(\mathbf{p}-\mathbf{p}_j)^T (\mathbf{b}_i^T \otimes \mathbf{I}_3)\right\} \mathrm{d}\mathbf{q} \\
&\quad + 2\mathrm{d}\mathbf{p}^T \left\{\mathcal{R}\mathbf{b}_i (\mathbf{p}-\mathbf{p}_j)^T (\mathbf{b}_i^T \otimes \mathbf{I}_3) + (\mathbf{p}-\mathbf{p}_j)(\mathbf{p}-\mathbf{p}_j)^T (\mathbf{b}_i^T \otimes \mathbf{I}_3)\right\} \mathrm{d}\mathbf{q} \\
&\quad + \mathrm{d}\mathbf{p}^T \left\{\mathcal{R}\mathbf{b}_i l_I^T \mathcal{R}^T + (\mathbf{p}-\mathbf{p}_j)(\mathbf{p}-\mathbf{p}_j)^T + 2\mathcal{R}\mathbf{b}_i (\mathbf{p}-\mathbf{p}_j)^T\right\} \mathrm{d}\mathbf{p} \\
&= \begin{bmatrix} \mathrm{d}\mathbf{q}^T & \mathrm{d}\mathbf{p}^T \end{bmatrix} \left[\begin{array}{c|c} (\mathbf{b}_i \otimes \mathbf{I}_3)(\mathbf{p}-\mathbf{p}_j)(\mathbf{p}-\mathbf{p}_j)^T (\mathbf{b}_i^T \otimes \mathbf{I}_3) & * \\ \hline \mathcal{R}\mathbf{b}_i (\mathbf{p}-\mathbf{p}_j)^T (\mathbf{b}_i^T \otimes \mathbf{I}_3) & (\mathbf{p}-\mathbf{p}_j)(\mathbf{p}-\mathbf{p}_j)^T \\ \quad + (\mathbf{p}-\mathbf{p}_j)(\mathbf{p}-\mathbf{p}_j)^T (\mathbf{b}_i^T \otimes \mathbf{I}_3) & \quad + 2\mathcal{R}\mathbf{b}_i (\mathbf{p}-\mathbf{p}_j)^T \\ & \quad + \mathcal{R}\mathbf{b}_i l_I^T \mathcal{R}^T \end{array}\right] \begin{bmatrix} \mathrm{d}\mathbf{q} \\ \mathrm{d}\mathbf{p} \end{bmatrix}.
\end{aligned}$$

Then we obtain

$$\mathrm{d}^2 r_{ij} = \frac{1}{r_{ij}}\mathrm{d}\beta - \frac{1}{r_{ij}^3}\beta^2 = \begin{bmatrix} \mathrm{d}\mathbf{q}^T & \mathrm{d}\mathbf{p}^T \end{bmatrix} \mathcal{H}_{ij} \begin{bmatrix} \mathrm{d}\mathbf{q} \\ \mathrm{d}\mathbf{p} \end{bmatrix},$$

where

$$\mathcal{H}_{ij} = \left[\begin{array}{c|c} -\frac{1}{r_{ij}^3}(\mathbf{b}_i \otimes \mathbf{I}_3)(\mathbf{p}-\mathbf{p}_j)(\mathbf{p}-\mathbf{p}_j)^T(\mathbf{b}_i^T \otimes \mathbf{I}_3) & * \\ \hline -\frac{1}{r_{ij}^3}\Big(\mathcal{R}\mathbf{b}_i(\mathbf{p}-\mathbf{p}_j)^T(\mathbf{b}_i^T \otimes \mathbf{I}_3) & -\frac{1}{r_{ij}^3}\Big((\mathbf{p}-\mathbf{p}_j)(\mathbf{p}-\mathbf{p}_j)^T \\ \quad + (\mathbf{p}-\mathbf{p}_j)(\mathbf{p}-\mathbf{p}_j)^T(\mathbf{b}_i^T \otimes \mathbf{I}_3)\Big) & \quad + 2\mathcal{R}\mathbf{b}_i(\mathbf{p}-\mathbf{p}_j)^T + \mathcal{R}\mathbf{b}_i\mathbf{b}_i^T\mathcal{R}^T\Big) \\ +\frac{1}{r_{ij}}(\mathbf{b}_i^T \otimes \mathbf{I}_3) & +\frac{1}{r_{ij}}\mathbf{I}_3 \end{array}\right].$$

Now we have all the ingredients to determine the hessian. Define matrix $\mathbf{W} = \delta(\mathbf{r}(\boldsymbol{\theta}))^{-1}\mathbf{R}^{-1}\delta(\mathbf{r}(\boldsymbol{\theta}))^{-1} \in \mathbb{R}^{mp\times mp}$. Then the second differential becomes

$$\begin{aligned} \mathrm{d}^2\widetilde{f}(\boldsymbol{\theta}) &= \mathrm{d}\mathbf{r}^T\mathbf{R}^{-1}\mathrm{d}\mathbf{r} - \sum_{i,j=1}^{p,m} \alpha_{ij}\ \mathrm{d}^2 r_{ij} \\ &= \begin{bmatrix}\mathrm{d}\mathbf{q}^T & \mathrm{d}\mathbf{p}^T\end{bmatrix} \left\{ \begin{bmatrix}\mathbf{F}^T \\ \mathbf{C}^T\end{bmatrix} \mathbf{W} \begin{bmatrix}\mathbf{F} & \mathbf{C}\end{bmatrix} - \sum_{i,j=1}^{p,m} \alpha_{ij}\ \mathcal{H}_{ij} \right\} \begin{bmatrix}\mathrm{d}\mathbf{q} \\ \mathrm{d}\mathbf{p}\end{bmatrix}, \end{aligned} \tag{A.53}$$

and finally, the extrinsic Hessian becomes

$$\nabla^2\widetilde{f}(\boldsymbol{\theta}) = \begin{bmatrix}\mathbf{F}^T\mathbf{W}\mathbf{F} & \mathbf{F}^T\mathbf{W}\mathbf{C} \\ \mathbf{C}^T\mathbf{W}\mathbf{F} & \mathbf{C}^T\mathbf{W}\mathbf{C}\end{bmatrix} - \sum_{i,j=1}^{p,m} \alpha_{ij}\ \mathcal{H}_{ij}. \tag{A.54}$$

□

A.3.2 Gradient and Hessian of the MLSR cost function

Proof of Proposition 4.3.4. We need to determine the extrinsic gradient of the function $f : SE(3) \to \mathbb{R}$ given by

$$f(\boldsymbol{\theta}) := \frac{1}{2}(\bar{\mathbf{d}} - \mathbf{d}(\boldsymbol{\theta}))^T\boldsymbol{\Sigma}^{-1}(\bar{\mathbf{d}} - \mathbf{d}(\boldsymbol{\theta})), \tag{A.55}$$

where $\mathbf{d}(\boldsymbol{\theta}) = [d_{11}(\boldsymbol{\theta})\ \ldots\ d_{pm}(\boldsymbol{\theta})]^T \in \mathbb{R}^{mp}$ is the vector of squared range measurements. To begin we need to compute the first differential

$$\mathrm{d}\widetilde{f}(\boldsymbol{\theta}) = -(\bar{\mathbf{d}} - \mathbf{d}(\boldsymbol{\theta}))^T\boldsymbol{\Sigma}^{-1}\mathrm{d}\mathbf{d}(\boldsymbol{\theta}). \tag{A.56}$$

The differential of $\mathbf{d}(\boldsymbol{\theta})$ is given by the differential of its components

$$\begin{aligned}
\mathrm{d}d_{ij}(\boldsymbol{\theta}) &= \mathrm{d}\|\mathcal{R}\mathbf{b}_i + \mathbf{p} - \mathbf{p}_j\| = \mathrm{d}\Big((\mathcal{R}\mathbf{b}_i + \mathbf{p} - \mathbf{p}_j)^T(\mathcal{R}\mathbf{b}_i + \mathbf{p} - \mathbf{p}_j)\Big) \\
&= \mathrm{d}\Big(\mathbf{b}_i^T\mathbf{b}_i + 2\mathbf{b}_i^T\mathcal{R}^T(\mathbf{p} - \mathbf{p}_j) + (\mathbf{p} - \mathbf{p}_j)^T(\mathbf{p} - \mathbf{p}_j)\Big) \\
&= \Big(2\mathbf{b}_i^T\mathrm{d}\mathcal{R}^T(\mathbf{p} - \mathbf{p}_j) + 2\mathbf{b}_i^T\mathcal{R}^T\mathrm{d}\mathbf{p} + 2(\mathbf{p} - \mathbf{p}_j)^T\mathrm{d}\mathbf{p}\Big) \\
&= 2\Big((\mathbf{p} - \mathbf{p}_j)^T\mathrm{d}\mathcal{R}\mathbf{b}_i + \mathbf{b}_i^T\mathcal{R}^T\mathrm{d}\mathbf{p} + (\mathbf{p} - \mathbf{p}_j)^T\mathrm{d}\mathbf{p}\Big) \\
&= 2\Big(\mathbf{b}_i^T \otimes (\mathbf{p} - \mathbf{p}_j)^T\Big)\mathrm{d}\mathbf{q} + 2\Big(\mathbf{b}_i^T\mathcal{R}^T + (\mathbf{p} - \mathbf{p}_j)^T\Big)\mathrm{d}\mathbf{p}, \qquad \text{(A.57)}
\end{aligned}$$

which yields

$$\mathrm{d}\mathbf{d}(\boldsymbol{\theta}) = 2\mathbf{F}\mathrm{d}\mathbf{q} + 2\mathbf{C}\mathrm{d}\mathbf{p} = 2\begin{bmatrix}\mathbf{F} & \mathbf{C}\end{bmatrix}\begin{bmatrix}\mathrm{d}\mathbf{q} \\ \mathrm{d}\mathbf{p}\end{bmatrix}, \qquad \text{(A.58)}$$

$$\mathrm{d}\widetilde{f}(\boldsymbol{\theta}) = -2(\bar{\mathbf{d}} - \mathbf{d}(\boldsymbol{\theta}))^T\boldsymbol{\Sigma}^{-1}\begin{bmatrix}\mathbf{F} & \mathbf{C}\end{bmatrix}\begin{bmatrix}\mathrm{d}\mathbf{q} \\ \mathrm{d}\mathbf{p}\end{bmatrix}, \qquad \text{(A.59)}$$

and

$$\nabla\widetilde{f}(\boldsymbol{\theta}) := -2\begin{bmatrix}\mathbf{F}^T\boldsymbol{\Sigma}^{-1}\left(\bar{\mathbf{d}} - \mathbf{d}(\boldsymbol{\theta})\right) \\ \mathbf{C}^T\boldsymbol{\Sigma}^{-1}\left(\bar{\mathbf{d}} - \mathbf{d}(\boldsymbol{\theta})\right)\end{bmatrix}. \qquad \text{(A.60)}$$

□

Proof of Proposition 4.3.7. Following similar steps as in the previous derivations, the second differential can be written as

$$\mathrm{d}^2\widetilde{f}(\boldsymbol{\theta}) = \mathrm{d}\mathbf{d}(\boldsymbol{\theta})\mathbf{R}^{-1}\mathrm{d}\mathbf{d}(\boldsymbol{\theta}) - \sum_{i,j=1}^{p,m}\alpha_i\mathrm{d}^2d_{ij}, \qquad \text{(A.61)}$$

where

$$\begin{aligned}
\mathrm{d}^2d_{ij} &= 2\mathrm{d}\Big((\mathbf{p} - \mathbf{p}_j)^T\mathrm{d}\mathcal{R}\mathbf{b}_i + \mathbf{b}_i^T\mathcal{R}^T\mathrm{d}\mathbf{p} + (\mathbf{p} - \mathbf{p}_j)^T\mathrm{d}\mathbf{p}\Big) \\
&= 2\mathrm{d}\mathbf{p}^T\mathrm{d}\mathcal{R}\mathbf{b}_i + 2\mathbf{b}_i^T\mathrm{d}\mathcal{R}^T\mathrm{d}\mathbf{p} + 2\mathrm{d}\mathbf{p}^T\mathrm{d}\mathbf{p} \\
&= 4\mathrm{d}\mathbf{p}^T\mathrm{d}\mathcal{R}\mathbf{b}_i + 2\mathrm{d}\mathbf{p}^T\mathrm{d}\mathbf{p} \\
&= 4\mathrm{d}\mathbf{p}^T(\mathbf{b}_i^T \otimes \mathbf{I}_3)\mathrm{d}\mathbf{q} + 2\mathrm{d}\mathbf{p}^T\mathrm{d}\mathbf{p} \\
&= \begin{bmatrix}\mathbf{q}^T & \mathbf{p}^T\end{bmatrix}\begin{bmatrix}0 & 2\mathbf{b}_i \otimes \mathbf{I}_3 \\ 2\mathbf{b}_i^T \otimes \mathbf{I}_3 & 2\mathbf{I}_3\end{bmatrix}\begin{bmatrix}\mathbf{q} \\ \mathbf{p}\end{bmatrix} \\
&:= \begin{bmatrix}\mathbf{q}^T & \mathbf{p}^T\end{bmatrix}\mathcal{H}_{ij}\begin{bmatrix}\mathbf{q} \\ \mathbf{p}\end{bmatrix}.
\end{aligned}$$

The second differential satisfies

$$\mathrm{d}^2\widetilde{f}(\boldsymbol{\theta}) = \begin{bmatrix}\mathrm{d}\mathbf{q}^T & \mathrm{d}\mathbf{p}^T\end{bmatrix}\left\{4\begin{bmatrix}\mathbf{F}^T\boldsymbol{\Sigma}^{-1}\mathbf{F} & \mathbf{F}^T\boldsymbol{\Sigma}^{-1}\mathbf{C}\\ \mathbf{C}^T\boldsymbol{\Sigma}^{-1}\mathbf{F} & \mathbf{C}^T\boldsymbol{\Sigma}^{-1}\mathbf{C}\end{bmatrix} - \sum_{i,j=1}^{p,m}\alpha_{ij}\,\mathcal{H}_{ij}\right\}\begin{bmatrix}\mathrm{d}\mathbf{q}\\ \mathrm{d}\mathbf{p}\end{bmatrix}, \tag{A.62}$$

and the extrinsic Hessian

$$\nabla^2\widetilde{f}(\boldsymbol{\theta}) = 4\begin{bmatrix}\mathbf{F}^T\boldsymbol{\Sigma}^{-1}\mathbf{F} & \mathbf{F}^T\boldsymbol{\Sigma}^{-1}\mathbf{C}\\ \mathbf{C}^T\boldsymbol{\Sigma}^{-1}\mathbf{F} & \mathbf{C}^T\boldsymbol{\Sigma}^{-1}\mathbf{C}\end{bmatrix} - \sum_{i,j=1}^{p,m}\alpha_{ij}\,\mathcal{H}_{ij}. \tag{A.63}$$

Lets try to find a simpler expression for the second term

$$\sum_{i,j=1}^{p,m}\alpha_{ij}\,\mathcal{H}_{ij} = \begin{bmatrix}0 & 2\sum_{i,j=1}^{p,m}\alpha_{ij}(\mathbf{b}_i\otimes\mathbf{I}_3)\\ 2\sum_{i,j=1}^{p,m}\alpha_{ij}(\mathbf{b}_i^T\otimes\mathbf{I}_3) & 2\sum_{i,j=1}^{p,m}\alpha_{ij}\mathbf{I}_3\end{bmatrix}. \tag{A.64}$$

clearly the lower right block can be simplified to

$$2\sum_{i,j=1}^{p,m}\alpha_{ij}\mathbf{I}_3 = 2\left(\sum_{i,j=1}^{p,m}\alpha_{ij}\right)\mathbf{I}_3 = 2\boldsymbol{\alpha}^T\mathbf{1}_{mp}\mathbf{I}_3 \tag{A.65}$$

$$= 2(\bar{\mathbf{d}}-\mathbf{d})^T\boldsymbol{\Sigma}^{-1}\mathbf{1}_{mp}\mathbf{I}_3 \in \mathbb{R}^{3\times 3}. \tag{A.66}$$

The cross term block is a little bit more messy but can also be simplified greatly. The lower left block can be written as

$$\begin{aligned}2\sum_{i,j=1}^{p,m}\alpha_{ij}(\mathbf{b}_i^T\otimes\mathbf{I}_3) &= 2\alpha_{11}(\mathbf{b}_1^T\otimes\mathbf{I}_3) + 2\alpha_{21}(\mathbf{b}_2^T\otimes\mathbf{I}_3) + \ldots\ldots + 2\alpha_{pm}(\mathbf{b}_p^T\otimes\mathbf{I}_3)\\ &= 2\begin{bmatrix}\alpha_{11}\mathbf{I}_3 & \alpha_{21}\mathbf{I}_3 & \ldots\ldots & \alpha_{pm}\mathbf{I}_3\end{bmatrix}\begin{bmatrix}\mathbf{b}_1^T\otimes\mathbf{I}_3\\ \mathbf{b}_2^T\otimes\mathbf{I}_3\\ \vdots\\ \vdots\\ \mathbf{b}_p^T\otimes\mathbf{I}_3\end{bmatrix}\\ &= 2\left(\boldsymbol{\alpha}^T\otimes\mathbf{I}_3\right)\left(\mathbf{1}_m\otimes\begin{bmatrix}\mathbf{b}_1^T\otimes\mathbf{I}_3\\ \vdots\\ \mathbf{b}_p^T\otimes\mathbf{I}_3\end{bmatrix}\right)\\ &= 2\Big((\bar{\mathbf{d}}-\mathbf{d})^T\boldsymbol{\Sigma}^{-1}\otimes\mathbf{I}_3\Big)\Big(\mathbf{1}_m\otimes\mathbf{B}^T\otimes\mathbf{I}_3\Big).\end{aligned}$$

This yields the desired result

$$\nabla^2\widetilde{f}(\boldsymbol{\theta}) = \begin{bmatrix}\mathbf{H}_{11} & \mathbf{H}_{21}^T\\ \mathbf{H}_{21} & \mathbf{H}_{22}\end{bmatrix} \in \mathbb{R}^{12\times 12}, \tag{A.67}$$

where

$$\mathbf{H}_{11} = 4\mathbf{F}^T\boldsymbol{\Sigma}^{-1}\mathbf{F}, \tag{A.68}$$

$$\mathbf{H}_{21} = 4\mathbf{C}^T\boldsymbol{\Sigma}^{-1}\mathbf{F} - 2\Big((\bar{\mathbf{d}} - \mathbf{d}(\boldsymbol{\theta}))^T\boldsymbol{\Sigma}^{-1} \otimes \mathbf{I}_3\Big)\Big(\mathbf{1}_m \otimes \mathbf{B}^T \otimes \mathbf{I}_3\Big), \tag{A.69}$$

$$\mathbf{H}_{22} = 4\mathbf{C}^T\boldsymbol{\Sigma}^{-1}\mathbf{C} - 2\Big((\bar{\mathbf{d}} - \mathbf{d}(\boldsymbol{\theta}))^T\boldsymbol{\Sigma}^{-1}\mathbf{1}_{mp}\Big)\mathbf{I}_3. \tag{A.70}$$

□

Appendix B

Some basic concepts in Riemannian geometry

In section 4.3.2 we introduced the concept of optimization on Riemannian submanifolds of $\mathbb{R}^n$. Basically we described how to generalize the classic gradient and Newton descent iterative algorithms when the cost function to minimize is not defined in an Euclidean space but in some constraint set which could be given the structure of a Riemannian manifold. We also discussed under which conditions a set of constraints defined a Riemannian submanifold of $\mathbb{R}^n$ and how the Special Euclidean group $SE(3)$ could be described in this way. The generalized gradient descent and Newton algorithms were based in defining intrinsic gradient and Newton directions and performing intrinsic line searches along geodesics. Some of these objects were then introduced in a brief manner and some important mathematical concepts were not properly defined. For the sake of completeness this appendix aims at defining these objects in a more precise manner. A good introduction to the fields of differential and Riemannian geometry can be found in the books by Lee [109], [110], and [108]. More advanced references include [31], [55], and [150].

Let us now introduce the intrinsic gradient and Hessian of an smooth function defined on a Riemannian manifold. We will focus on embedded Riemannian submanifolds of $\mathbb{R}^n$, that can be characterized as the level set of some smooth mapping. This is the case of the Special Euclidean group $SE(3)$ and of many other famous manifolds of interest in the areas of engineering and signal processing. However, we recall that most of what follows can be generalized to abstract manifolds (that is not necessarily submanifolds of $\mathbb{R}^n$).

B.1 Intrinsic Gradient

Let Θ be a connected d-dimensional, closed, embedded submanifold of $\mathbb{R}^n$. For instance as characterized by a regular level set of the smooth map $h : W \subset \mathbb{R}^n \rightarrow$

$\mathbb{R}^{n-d}$,

$$\Theta = \{\mathbf{x} \in W \subset \mathbb{R}^n \,:\, h(\mathbf{x}) = \mathbf{c}\}\,, \tag{B.1}$$

for some constant $\mathbf{c} \in \mathbb{R}^{n-d}$ as described in section 4.3.2. One of the first objects that should be introduced are tangent vectors at a point $p \in \Theta$. There are different ways of defining tangent vectors. One way is to define them as linear operators (derivations) $(\cdot) : C^\infty(\Theta) \to \mathbb{R}$, where $C^\infty(\Theta)$ denotes the set of infinitely differentiable functions on Θ. That is, a tangent vector X_p takes a smooth function $f \in C^\infty(\Theta)$, $f : \Theta \to \mathbb{R}$, and returns a real number $X_p f$, thus providing a mean of taking directional derivatives of smooth functions in an intrinsic manner.

Another point of view is to define tangent vectors as equivalence classes of curves that, roughly speaking, share their velocity vector at a given point. We refer the reader to [31], [110], and [55] for formal definitions and to [101] for a more informal but enlightening discussion on the topic. This level of abstraction is needed when dealing with abstract manifolds. Fortunately, in the case of submanifolds of $\mathbb{R}^n$ there are useful identifications that allow us to avoid those abstract constructions and work with much common objects such as vectors and linear subspaces of $\mathbb{R}^n$.

The tangent space of Θ at $p \in \Theta$ denoted by $T_p\Theta$ is the collection of all tangent vectors of Θ at p. The tangent space $T_p\Theta$ can be identified with the d-dimensional linear subspace

$$T_p\Theta = \left\{\Delta \in \mathbb{R}^n \,:\, \nabla h(p)^T \Delta = 0\right\}. \tag{B.2}$$

Recall that we assumed that $\nabla h(p)$ had rank $n-d$ for all $p \in \Theta$. Hence, $T_p\Theta$, which can be identified with the null space of $\nabla h(p)$, has the same dimension d of the base manifold Θ as we should expect. Under the above identification any tangent vector $X_p \in T_p\Theta$ corresponds uniquely to a vector in $\mathbb{R}^n$. This in in fact a great deal: we are able to identify an abstract tangent vector with a much more familiar object, a vector in $\mathbb{R}^n$. Note that in such a case we can write

$$X_p f = \frac{d}{dt} f(p + tX_p), \tag{B.3}$$

which is the familiar directional derivative of f along the direction given by X_p. The tangent bundle $T\Theta$ is the disjoint union of all the tangent spaces $T_p\Theta, p \in \Theta$. A Riemannian metric g is a positive definite symmetric smooth 2-tensor field on $T\Theta$. That is, it can be identified with a smooth assignment of an inner product at each vector space $T_p\Theta$. The ambient space $\mathbb{R}^n$ has a natural inner product $\langle\cdot,\cdot\rangle : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ given by $\langle \mathbf{x}, \mathbf{y}\rangle = \mathbf{x}^T\mathbf{y}$. We can endow Θ with a Riemannian metric g

by inheriting the one of its ambient space

$$g_p : T_p\Theta \times T_p\Theta \to \mathbb{R}, \tag{B.4}$$

$$X_p, Y_p \mapsto g_p(X_p, Y_p) = \langle X_p, Y_p \rangle = X_p^T Y_p \quad \forall p \in \Theta, \forall X_p, Y_p \in T_p\Theta, \tag{B.5}$$

where we used that any pair of tangent vectors $X_p, Y_p \in T_p\Theta$ are uniquely identified with a pair of vectors in $\mathbb{R}^n$. In differential geometry language, Θ is a Riemannian submanifold of $\mathbb{R}^n$ equipped with its canonical Riemannian structure [55], [31], [108].

We need just one more artifact in order to be able to define the intrinsic gradient of a smooth function. Let V be an n-dimensional vector space. An inner product $\langle \cdot, \cdot \rangle$ on V establishes an isomorphism $V \simeq V^*$, where V^* denotes the dual space of V, that is the space of covectors or linear functionals $(\cdot) : V \to \mathbb{R}$. To each vector $X \in V$ corresponds a covector $X^\flat \in V^*$ defined as $X^\flat : V \to \mathbb{R}$, $X^\flat(Y) = \langle X, Y \rangle, \forall Y \in V$. The map $V \to V^*$, $X \mapsto X^\flat$ is an isomorphism (a bijective linear map). Its inverse is denoted by $V^* \to V$, $\omega \mapsto \omega^\sharp$. That is given a covector $\omega \in V^*$ returns a vector $\omega^\sharp$ such that $\omega(Y) = \langle \omega^\sharp, Y \rangle, \forall Y \in V$. The operators flat $\flat$, and sharp $\sharp$ are called Raising and lowering indices [31], [108, p.28] and musical isomorphisms [150, 2.66]. One of the most important applications of the sharp operator is that it allows us to define the intrinsic gradient of a smooth function on a Riemannian manifold. Let $f : \Theta \to \mathbb{R}$ be a smooth function. Let df denote the differential of f. Note that df is a well defined object although some times is obscurely introduced in the literature. The differential df is a a 1-form or a smooth covector field (that is a "slot machine" taking tangent vectors and returning scalars) obtained as the exterior differentiation of f (when regarded as a 0-form) [110, p.71] [73]. The intrinsic gradient of f is the smooth vector field defined point-wise as

$$\text{grad} f(p) = df(p)^\sharp, \tag{B.6}$$

for all $p \in \Theta$. Thus for any tangent vector $X_p \in T_p\Theta$ we have

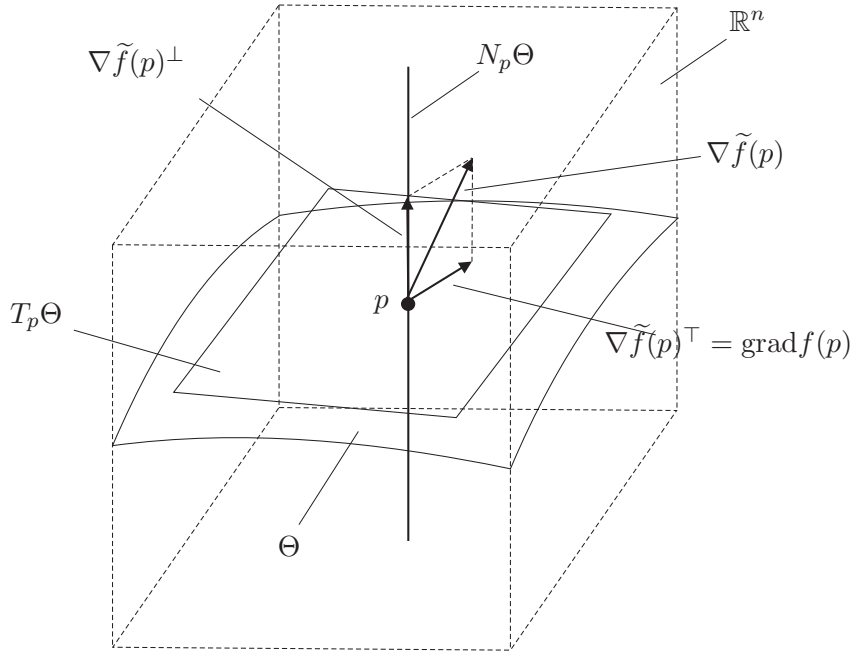
$$X_p f = df(p)(X_p) = \langle X_p, \text{grad} f(p) \rangle. \tag{B.7}$$

The tangent space of the ambient space $\mathbb{R}^n$ at any point $p \in \Theta$ can be expressed as the direct sum $T_p\mathbb{R}^n \simeq T_p\Theta \oplus N_p\Theta$ [55, p.125] where $N_p\Theta$ is the orthogonal complement of the linear subspace $T_p\Theta$ in $\mathbb{R}^n$.

Let $\widetilde{f} : \mathbb{R}^n \to \mathbb{R}$ be a smooth extension of f to $\mathbb{R}^n$. That is, in such a way that f can be regarded as the restriction of $\widetilde{f}$ to Θ. Let $\nabla\widetilde{f}(p) \in T_p\mathbb{R}^n = \mathbb{R}^n$ be the vector of partial derivatives of $\widetilde{f}$ evaluated at $p \in \Theta$, that is

$$\nabla\widetilde{f}(p) = \left[\frac{\partial}{\partial x_1}\widetilde{f}(p) \ \ldots \ \frac{\partial}{\partial x_n}\widetilde{f}(p) \right]^T \in \mathbb{R}^n. \tag{B.8}$$

We will call $\nabla\widetilde{f}(p)$ the extrinsic gradient of f at p. Since $\nabla\widetilde{f}(p) \in T_p\mathbb{R}^n = \mathbb{R}^n =$



$T_p\Theta \oplus N_p\Theta$ it can be decomposed in two orthogonal vectors $\nabla\widetilde{f}(p) = \nabla\widetilde{f}(p)^{\top} + \nabla\widetilde{f}(p)^{\perp}$ where $\nabla\widetilde{f}(p)^{\top} \in T_p\Theta$ and $\nabla\widetilde{f}(p)^{\perp} \in N_p\Theta$. The intrinsic gradient of f evaluated at $p \in \Theta$, denoted by $\mathrm{grad}f(p)$ is a tangent vector in $T_p\Theta$, and according to Proposition 4.3.1 it is precisely the projection $\nabla\widetilde{f}(p)^{\top}$ of the extrinsic gradient on the tangent space $T_p\Theta$

$$\mathrm{grad}f(p) = \nabla f(p)^{\top} \in T_p\Theta. \tag{B.9}$$

We are now ready to give a simple proof of the Proposition:

Proof of Proposition 4.3.1. The intrinsic gradient $\mathrm{grad}f(p)$ is the unique tangent vector satisfying $X_p f = \langle X_p, \mathrm{grad}f(p)\rangle$ for all tangent vectors $X_p \in T_p\Theta$ according to (B.7). Moreover, since $\widetilde{f}$ is a smooth extension of f, we have that $X_p f = X_p\widetilde{f}$, and it follows that

$$\begin{aligned} X_p f = X_p\widetilde{f} &= \langle X_p, \nabla\widetilde{f}(p)\rangle = \langle X_p, \nabla\widetilde{f}(p)^{\perp}\rangle + \langle X_p, \nabla\widetilde{f}(p)^{\top}\rangle \\ &= \langle X_p, \nabla\widetilde{f}(p)^{\top}\rangle, \end{aligned} \tag{B.10}$$

for all $X_p \in \Theta$ since $\langle X_p, \nabla\widetilde{f}(p)^{\perp}\rangle = 0$ because they are orthogonal. □

The intrinsic gradient can be then obtained by computing the extrinsic gradient and orthogonally project it onto the tangent space $T_p\Theta$. This might seem not a great deal, this is actually what intuition would suggest at a first sight. What we did is just a theoretical development that sustains intuition. Note that this is exactly the principle of the gradient projection method, compute that gradient and project it onto the tangent plane to the constraint surface.

The intrinsic gradient can be used, among other things, to compute first order local approximation of functions. Consider an Euclidean parameter space $\Theta = \mathbb{R}^n$ and a function $f : \mathbb{R}^n \to \mathbb{R}$. The first order approximation of f at a point $\mathbf{x} \in \mathbb{R}^n$ in the direction of vector $\mathbf{v} \in \mathbb{R}^n$ is given by the familiar expression

$$f(\mathbf{x} + t\mathbf{v}) = f(\mathbf{x}) + t\nabla f(\mathbf{x})^T \mathbf{v} + \text{h.o.t.}, \tag{B.11}$$

where $\nabla f(\mathbf{x})$ denotes the gradient of f evaluated at $\mathbf{x}$ and h.o.t. stands for higher order terms. In order to generalize this to functions defined in Riemannian submanifolds we need first to clarify some issues. Let $\mathbf{x} \in \Theta$ and $\Delta \in T_{\mathbf{x}}\Theta$. Let $c : I \subset \mathbb{R} \to \Theta$ denote a curve such that $c(0) = \mathbf{x}$ and $\dot{c}(0) = \Delta$. Then the first order approximation of f along the curve c is given by

$$f(c(t)) = f(\mathbf{x}) + t\langle \Delta, \text{grad} f(\mathbf{x}) \rangle + \text{h.o.t.}. \tag{B.12}$$

B.2 Intrinsic Hessian

Before we define the intrinsic Hessian we need to be able to define second order derivatives in an intrinsic manner. In differential geometry this is done be resorting to linear connections and covariant derivatives. The problem arises when we are faced with computing the rate of variation of a given vector field with respect to another. Let us illustrate this with a simple example of a surface in $\mathbb{R}^3$. Let $S \subset \mathbb{R}^3$ be a surface and let $c : I \subset \mathbb{R} \to S$ be a smooth curve on S. Let $V : I \to \mathbb{R}^3$ be a vector field along c tangent to S. The time derivative $\frac{dV(t)}{dt}$ might not be a tangent vector to the surface anymore! [55, ch.2]. Hence, we need a tool for differentiating vector fields that assures us that the result is again a tangent vector field. Let Θ be a n-dimensional smooth manifold. Let $\mathcal{T}(\Theta)$ denote the space of vector fields on Θ (that is smooth assignments of tangent vectors at each tangent space, or more technically the collection of sections of $T\Theta$). A linear connection is a map

$$\nabla : \mathcal{T}(\Theta) \times \mathcal{T}(\Theta) \to \mathcal{T}(\Theta), \quad (X, Y) \mapsto \nabla_X Y, \tag{B.13}$$

satisfying the following properties:

$$(a)\ \nabla_{f_1 X 1 + f_2 X_2} Y = f_1 \nabla_{X_1} Y + f_2 \nabla_{X_2} Y, \tag{B.14}$$

$$(b)\ \nabla_X (a_1 Y_1 + a_2 Y_2) = a_1 \nabla_X Y_1 + a_2 \nabla_X Y_2, \tag{B.15}$$

$$(c)\ \nabla_X (fY) = (Xf)Y + f\nabla_X Y, \tag{B.16}$$

for all $f, f_1, f_2 \in C^\infty(\Theta)$, $a_1, a_2 \in \mathbb{R}$, and $X, X_1, X_2, Y, Y_1, Y_2 \in \mathcal{T}(\Theta)$. We read $\nabla_X Y$ as the covariant derivative of Y with respect to X [108], [31], [55]. A connection is uniquely specified by the so-called Christoffel symbols. Given a local frame of vector fields $\{E_1, \ldots, E_n\}$, that is$\{E_{1p}, \ldots, E_{np}\}$ form a basis for $T_p\Theta$ for all $p \in U \subset \Theta$,

for any $1 \leq i, j \leq n$ we have the expansion

$$\nabla_{E_i} E_j = \sum_{k=1}^{n} \Gamma_{ij}^k E_k, \tag{B.17}$$

where the n^3 functions $\Gamma_{ij}^k : U \subset \Theta \to \mathbb{R}$ are the Christoffel symbols of ∇ with respect to $\{E_1, \ldots, E_n\}$. When defining the connection, we did not need a Riemannian metric on Θ, in fact connections can be defined for arbitrary smooth manifolds. However in a Riemannian manifold there is a special kind of connection called the Levi-Civita connection that is deeply linked to the Riemannian metric. The Levi-Civita connection is the unique connection that is symmetric and compatible with the metric. The definitions of compatibility with the metric and symmetry are beyond the scope of the present work. We refer the reader to [108], [55]. Let us simply mention that the connection is compatible with the metric if it preserves the angles of parallel vector fields along a curve (where again parallelness is left undefined), and is symmetric if the Christoffel symbols satisfy $\Gamma_{ij}^k = \Gamma_{ji}^k, \forall 1 \leq i, j \leq n$. It can be in fact proved that such a connection always exists and is unique.

In order to introduce the intrinsic hessian we need furthermore to define the covariant derivative of smooth covector fields. Let $\omega : T^*\Theta \to \mathbb{R}$ be a smooth covector field of Θ and let ∇ be a linear connection. Then the covariant derivative of ω with respect to the vector field X is the smooth covector field

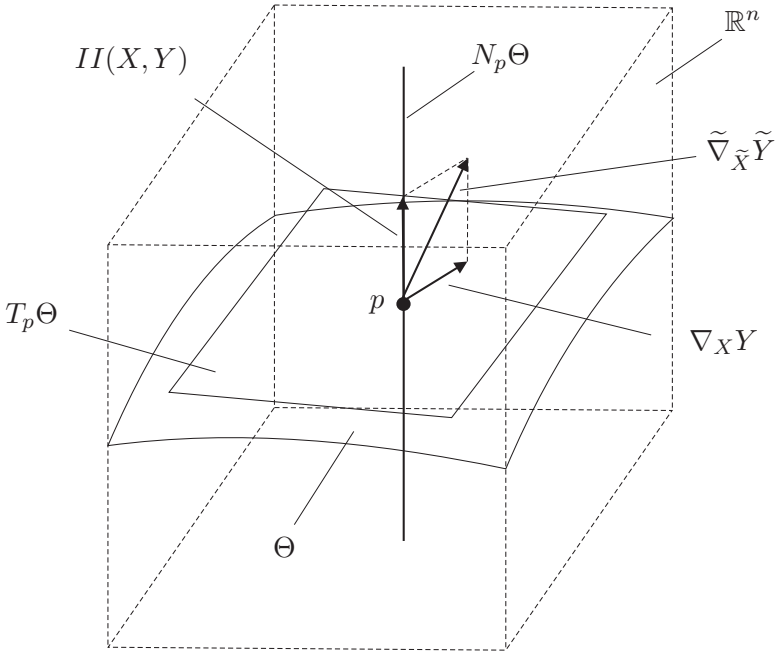
$$(\nabla_X \omega)(Y) = X\omega(Y) - \omega(\nabla_X Y), \quad \forall Y \in \mathcal{T}(\Theta). \tag{B.18}$$

We are now ready to introduce the main definition of the subsection. The intrinsic Hessian of a smooth function f with respect to a connection ∇ denoted by $\mathrm{Hess} f$ is the smooth 2-tensor field on Θ defined as

$$\begin{aligned} \mathrm{Hess} f(X, Y) &= (\nabla_Y df)(X) = Y df(X) - df(\nabla_X Y) \\ &= Y(Xf) - (\nabla_X Y) f, \quad \forall X, Y \in \mathcal{T}(\Theta). \end{aligned} \tag{B.19}$$

This is still a quite abstract definition hard to use in practice. Let us now focus on the case where Θ is a Riemannian submanifold of $\mathbb{R}^n$ and both are endowed with their Levi-Civita connections ∇ and $\widetilde{\nabla}$ respectively. We saw that for $p \in \Theta$ we had the decomposition $T_p\mathbb{R}^n \simeq T_p\Theta \oplus N_p\Theta$ where $T_p\Theta$ and $N_p\Theta$ could be identified with subspaces of $\mathbb{R}^n$. The geometries of Θ and $\mathbb{R}^n$ are in fact related by the so-called Second Fundamental form $\mathrm{II} : T_p\Theta \times T_p\Theta \to N_p\Theta$. Let X, Y be vector fields on Θ and $\widetilde{X}, \widetilde{Y}$ be local extensions on $\mathbb{R}^n$. The Second fundamental form is defined by [55, p.128],[31],[150, 5.1]

$$II(X, Y) = \widetilde{\nabla}_{\widetilde{X}} \widetilde{Y} - \nabla_X Y = (\widetilde{\nabla}_{\widetilde{X}} \widetilde{Y})^{\perp}. \tag{B.20}$$



If we define $\nabla^2 h_i(p)$ as the $n \times n$ hessian matrix of second order derivatives of the component function h_i whose level set characterized Θ, the Second fundamental form can be actually computed according to [190] by

$$II(\Delta_p, \Omega_p) = -\nabla h(p)(\nabla h(p)^T \nabla h(p))^{-1} \begin{bmatrix} \Delta_p^T \nabla^2 h_1(p) \Omega_p \\ \vdots \\ \Delta_p^T \nabla^2 h_{n-d}(p) \Omega_p \end{bmatrix}, \tag{B.21}$$

for all $\Delta_p, \Omega_p \in T_p\Theta$.

We are now ready to show how to actually compute the intrinsic Hessian of a smooth function defined in a Riemannian submanifold of $\mathbb{R}^n$. Let $\widetilde{f} : \mathbb{R}^n \to \mathbb{R}$ be a smooth extension of function $f : \Theta \to \mathbb{R}$, and let $\text{Hess}\widetilde{f}$ denote its Hessian. That is, for each $\mathbf{x} \in \Theta \subset \mathbb{R}^n$, $\text{Hess}\widetilde{f}(\mathbf{v}_1, \mathbf{v}_2) = \mathbf{v}_1^T \nabla^2 \widetilde{f}(\mathbf{x}) \mathbf{v}_2$ where $\nabla^2 \widetilde{f}(\mathbf{x})$ is the standard matrix of second order partial derivatives of $\widetilde{f}$. According to Proposition 4.3.2, the intrinsic Hessian of the smooth function f at $\mathbf{x} \in \Theta$ satisfies

$$Hess f(\Delta_1, \Delta_2) = Hess\widetilde{f}(\Delta_1, \Delta_2) + \langle II(\Delta_1, \Delta_2), \nabla \widetilde{f}(\mathbf{x}) \rangle \tag{B.22}$$

$$= \Delta_1^T \left(\nabla^2 \widetilde{f}(\mathbf{x}) \right) \Delta_2 + II(\Delta_1, \Delta_2)^T \nabla \widetilde{f}(\mathbf{x}), \tag{B.23}$$

for any tangent vectors $\Delta_1, \Delta_2 \in T_{\mathbf{x}}\Theta$.

Proof of Proposition 4.3.2. Let X, Y be vector fields on Θ and $\widetilde{X}, \widetilde{Y}$ be local smooth extensions on $\mathbb{R}^n$ in such a way that $Y(Xf) = \widetilde{Y}(\widetilde{X}\widetilde{f})$. From (B.20) and (B.19) it

follows that

$$\begin{aligned}\mathrm{Hess} f(X,Y) &= (\nabla_X df)Y = Y(Xf) - (\nabla_X Y)f \\ &= Y(Xf) - (\widetilde{\nabla}_{\widetilde{X}}\widetilde{Y} - II(X,Y))\widetilde{f} \\ &= \widetilde{Y}(\widetilde{X}\widetilde{f}) - (\widetilde{\nabla}_{\widetilde{X}}\widetilde{Y})\widetilde{f} + II(X,Y)\widetilde{f} \\ &= \mathrm{Hess}\widetilde{f}(\widetilde{X},\widetilde{Y}) + II(X,Y)\widetilde{f}, \end{aligned} \tag{B.24}$$

which yields the desired result. □

The intrinsic Hessian is also helpful as it allows to compute second order local approximation of functions defined on Riemannian manifolds. Let $c : I \subset \mathbb{R} \to \Theta$ denote a curve such that $c(0) = \mathbf{x}$ and $\dot{c}(0) = \Delta$. Then the second order approximation of f along the curve c around the point $\mathbf{x} \in \Theta$ is given by

$$f(c(t)) = f(\mathbf{x}) + t\langle \Delta, \mathrm{grad} f(\mathbf{x})\rangle + \frac{1}{2}t^2 \mathrm{Hess} f(\Delta,\Delta) + \text{h.o.t.}. \tag{B.25}$$

B.3 Geodesics

One of the key steps of geometric descent optimization were the geodesic line searches. This allowed us to generate iterates that evolved naturally on the constraint and thus avoiding normalization schemes. In section 4.3.2 we briefly said that geodesics were generalizations of straight lines on a Riemannian manifold.

For the sake of completeness let us give now a more formal definition of geodesics. Let Θ be a Riemannian submanifold of $\mathbb{R}^n$ endowed with its canonical Riemannian metric inherited from $\mathbb{R}^n$ and the Levi-Civita connection ∇. Given a smooth curve $c : I \subset \mathbb{R}^n \to \Theta$ we say that c is a geodesic if roughly speaking it has zero acceleration. In intrinsic terms this translates as the covariant derivative of its velocity vector $\dot{c}(t)$ with respect to itself vanishing identically on its domain of definition. That is if

$$\nabla_{\dot{c}(t)}\dot{c}(t) = 0, \quad \forall t \in I. \tag{B.26}$$

Let $\gamma : I \subset \mathbb{R} \to \Theta$ be a geodesic. Let (φ, U) be a chart, that is U is an open set of $\mathbb{R}^d$ and $\varphi : U \subset \mathbb{R}^d \to \Theta$ is a diffeomorphism. The coordinate representation of γ can be written as

$$\widehat{\gamma}(t) = (\varphi^{-1} \circ \gamma)(t) = (x_1(t), \ldots, x_d(t)). \tag{B.27}$$

In these coordinates, the geodesic condition translates to [55, p.62]

$$\frac{d^2 x_k(t)}{dt^2} + \sum_{i,j=1}^{d} \Gamma_{ij}^k \frac{dx_i(t)}{dt}\frac{dx_j(t)}{dt} = 0, \quad k = 1, \ldots, n, \tag{B.28}$$

which is a set of second order nonlinear ODE's usually hard to solve. However, we

already showed that in the case of $SE(3)$ there are easy closed form formulas to compute its geodesics.

Appendix C

Diffusion observer convergence analysis

The important question of whether the derived observer converges warrants careful analysis. Note that the observer equations are nonlinear due the gain K which is a quadratic function of the state. We start by characterizing the equilibrium points of the observer and then show that, under some conditions, the desired points are asymptotically stable.

The observer equations can be written in compact form as

$$\begin{cases} \dot{\mathbf{x}} = f(\mathbf{x}) = \boldsymbol{\xi} - K\mathbf{e} \\ K = c\mathbf{e}^T\boldsymbol{\xi}, \end{cases} \tag{C.1}$$

where

$$\mathbf{e} = \boldsymbol{\Gamma}(\mathbf{x} - \mathbf{x}_m) \in \mathbb{R}^{Nn\times 1} \tag{C.2}$$

$$\boldsymbol{\xi} = \mathbf{A}\mathbf{x} + \mathbf{b} \in \mathbb{R}^{Nn\times 1} \tag{C.3}$$

$$c = \frac{1}{n_X\sigma_X^2} \in \mathbb{R}. \tag{C.4}$$

The equilibrium trajectories are those satisfying $f(\mathbf{x}) = 0$. From (C.1), and since K is a scalar, it can be seen that for this to happen vectors $\boldsymbol{\xi}$ and $\mathbf{e}$ must be aligned. That is, their inner product must be $\boldsymbol{\xi}^T\mathbf{e} = \pm\|\boldsymbol{\xi}\|\|\mathbf{e}\|$. The set of equilibrium trajectories can then be characterized as $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$, where

$$\begin{aligned} \Omega_1 &= \{\boldsymbol{\xi} = 0\}, \\ \Omega_2 &= \{\boldsymbol{\xi}^T\mathbf{e} = +\|\boldsymbol{\xi}\|\|\mathbf{e}\| > 0,\ c\|\mathbf{e}\|^2 = 1\}, \\ \Omega_3 &= \{\boldsymbol{\xi}^T\mathbf{e} = -\|\boldsymbol{\xi}\|\|\mathbf{e}\| < 0,\ c\|\mathbf{e}\|^2 = 1\}. \end{aligned}$$

The first set Ω_1 consists only of the solution $\mathbf{x} = -\mathbf{A}^{-1}\mathbf{b}$, which in general does not satisfy the variance constraint and can be therefore considered as degenerate.

The following proposition gives sufficient conditions that ensure that the sets Ω_2 and Ω_3 are composed of isolated asymptotically stable and unstable equilibrium points, respectively.

Theorem C.0.1. *Define the constants*

$$\beta_2 = \sqrt{c}(\underline{\sigma}(\mathbf{A})\|\mathbf{x}_m\| - \|\mathbf{b}\|) - \overline{\sigma}(\mathbf{A})$$
$$\beta_3 = \sqrt{c}(\underline{\sigma}(\mathbf{A})\|\mathbf{x}_m\| - \|\mathbf{b}\|) - 2\overline{\sigma}(\mathbf{A}) - \underline{\sigma}(\mathbf{A})$$

where $\underline{\sigma}(\mathbf{A}), \overline{\sigma}(\mathbf{A})$ *denote the minimum and maximum singular values of matrix* $\mathbf{A}$*, respectively. Then,*

1. *If* $\beta_2 > 0$*, the equilibrium points in* Ω_2 *are isolated and (locally) asymptotically stable.*

2. *If* $\beta_3 > 0$*, the equilibrium points in* Ω_3 *are isolated and unstable.*

Matrix $\mathbf{A}$ has an important role in determining the observer properties. The following result will be useful in proving the previous proposition:

Lemma C.0.2. *Matrix* $\mathbf{A}$ *defined in (6.10) is negative definite, and in particular invertible.*

Proof If the matrix $\mathbf{C} = (\alpha_1\mathbf{L}_1 - \alpha_2\mathbf{L}_2)$ is negative definite the result follows since $\mathbf{A} = I_n \otimes \mathbf{C}$ is symmetric and with the same eigenvalues as $\mathbf{C}$ (with n times its multiplicities). Matrix $\mathbf{L}_1$ defined in (6.13) is negative definite [16, p.57]. It follows that its square $\mathbf{L}_1^2$ is positive definite. Moreover, it is easy to see that matrix $\mathbf{L}_2$ defined in (6.14) is also positive definite since it can be expressed as

$$\mathbf{L}_2 = \mathbf{L}_1^2 + \mathbf{e}_1\mathbf{e}_1^T + \mathbf{e}_N\mathbf{e}_N^T \succeq \mathbf{L}_1^2$$

where $\mathbf{e}_i \in \mathbb{R}^N$ has a 1 on its i'th entry and zeros elsewhere, and where given two symmetric matrices $\mathbf{A}, \mathbf{B}$, the expression $\mathbf{A} \succeq \mathbf{B}$ denotes that the difference $\mathbf{A} - \mathbf{B}$ is positive semidefinite. Because $\alpha_1, \alpha_2 > 0$, $\mathbf{C}$ is the sum of two negative definite matrices and is itself negative definite. The result follows immediately. □

Note that if instead of Dirichlet (fixed extreme positions) one had considered Neumann boundary conditions (fixed extreme velocities), the corresponding finite difference approximation matrix of ∇^2, $\mathbf{L}_1$ would be only negative semidefinite, and the result would not be valid.

Proof (Proposition C.0.1) After some computations, the Jacobian of the system can be found to be:

$$\frac{\partial f}{\partial \mathbf{x}} = (I - c\mathbf{e}\mathbf{e}^T)\mathbf{A} - c\,\mathbf{e}\boldsymbol{\xi}^T\boldsymbol{\Gamma} - c\,\mathbf{e}^T\boldsymbol{\xi}I \tag{C.5}$$

When evaluated in Ω_2,

$$\frac{\partial f}{\partial \mathbf{x}}|_{\Omega_2} = (I - c\mathbf{e}\mathbf{e}^T)\mathbf{A} - c\sqrt{c}\|\boldsymbol{\xi}\|\mathbf{e}\mathbf{e}^T - \sqrt{c}\|\boldsymbol{\xi}\|I$$

since $\boldsymbol{\Gamma}\mathbf{e} = \mathbf{e}$, $\boldsymbol{\Gamma}\boldsymbol{\xi} = K\boldsymbol{\Gamma}\mathbf{e} = \boldsymbol{\xi}$, and $\sqrt{c}\|\mathbf{e}\| = 1$. Moreover, when restricted to Ω_2 the following bounds apply

$$\begin{aligned}\sqrt{c}/c = \|\mathbf{e}\| &= \|\boldsymbol{\Gamma}(\mathbf{x} - \mathbf{x}_m)\| \geq \|\boldsymbol{\Gamma}\mathbf{x}_m\| - \|\boldsymbol{\Gamma}\mathbf{x}\| \\ &= \|\mathbf{x}_m\| - \|\boldsymbol{\Gamma}\mathbf{x}\| \Longrightarrow \\ \|\mathbf{x}\| \geq \|\boldsymbol{\Gamma}\mathbf{x}\| &\geq \|\mathbf{x}_m\| - \sqrt{c}/c \end{aligned} \tag{C.6}$$

$$\begin{aligned}\|\boldsymbol{\xi}\| = \|\mathbf{A}\mathbf{x} + \mathbf{b}\| &\geq \|\mathbf{A}\mathbf{x}\| - \|\mathbf{b}\| \\ &\geq \underline{\sigma}(\mathbf{A})\|\mathbf{x}\| - \|\mathbf{b}\| \\ &\geq \underline{\sigma}(\mathbf{A})(\|\mathbf{x}_m\| - \sqrt{c}/c) - \|\mathbf{b}\| \end{aligned} \tag{C.7}$$

$$\begin{aligned}c\mathbf{e}\mathbf{e}^T\mathbf{A} + c\mathbf{A}\mathbf{e}\mathbf{e}^T &\preceq \|c\mathbf{e}\mathbf{e}^T\mathbf{A} + c\mathbf{A}\mathbf{e}\mathbf{e}^T\|I \\ &\preceq \|c\mathbf{e}\mathbf{e}^T\mathbf{A}\|I + \|c\mathbf{A}\mathbf{e}\mathbf{e}^T\|I \\ &= 2\|c\mathbf{e}\mathbf{e}^T\mathbf{A}\|I \\ &\preceq 2\|c\mathbf{e}\mathbf{e}^T\|\|\mathbf{A}\|I \\ &= 2\overline{\sigma}(\mathbf{A})I \end{aligned} \tag{C.8}$$

where we used the fact that, since matrix $\mathbf{A}$ is negative definite, $-\overline{\sigma}(\mathbf{A})I \preceq \mathbf{A} \preceq -\underline{\sigma}(\mathbf{A})I$. The symmetric part of the Jacobian

$$\mathbf{J}_s = \frac{1}{2}\left(\frac{\partial f}{\partial \mathbf{x}} + \frac{\partial f}{\partial \mathbf{x}}^T\right), \tag{C.9}$$

when evaluated in Ω_2, satisfies

$$\begin{aligned}\mathbf{J}_s|_{\Omega_2} &= \mathbf{A} - \frac{c}{2}(\mathbf{e}\mathbf{e}^T\mathbf{A} + \mathbf{A}\mathbf{e}\mathbf{e}^T) - \sqrt{c}\|\boldsymbol{\xi}\|I \\ &\quad - c\sqrt{c}\|\boldsymbol{\xi}\|\mathbf{e}\mathbf{e}^T \\ &\preceq -\underline{\sigma}(\mathbf{A})I + \frac{1}{2}\|c\mathbf{e}\mathbf{e}^T\mathbf{A} + c\mathbf{A}\mathbf{e}\mathbf{e}^T\|I - \sqrt{c}\|\boldsymbol{\xi}\|I \\ &\preceq -\underline{\sigma}(\mathbf{A})I + \overline{\sigma}(\mathbf{A})I - \sqrt{c}\|\boldsymbol{\xi}\|I \\ &\preceq -\overline{\sigma}(\mathbf{A})I + \sqrt{c}(\underline{\sigma}(\mathbf{A})\|\mathbf{x}_m\| - \|\mathbf{b}\|)I \\ &= -\beta_2 I \end{aligned} \tag{C.10}$$

As a result, if $\beta_2 > 0$, $\mathbf{J}_s|_{\Omega_2}$ is negative definite. This implies that the Jacobian has all of its eigenvalues negative and, in particular, it is invertible. By the inverse function theorem, there is a neighborhood D of every point in Ω_2 in which f is bijective, and

thus the equilibrium points are isolated. Moreover, the stability of such equilibriums can be analyzed by using a Lyapunov function candidate $V : D \to \mathbb{R}$ given by

$$V(\mathbf{x}) = \|f(\mathbf{x})\|^2 \tag{C.11}$$

that is positive definite in D. Its time derivative, given by

$$\dot{V} = f(\mathbf{x})^T \mathbf{J}_s f(\mathbf{x}) \leq -\beta_2 \|f(\mathbf{x})\|^2, \tag{C.12}$$

is negative definite in D. It follows that the equilibrium points in Ω_2 are isolated and asymptotically stable [96]. In order to prove part (2) of the proposition we follow a similar reasoning. Evaluating the symmetric part of the Jacobian on Ω_3, and using (C.6)-(C.8), we find that

$$\begin{aligned} \mathbf{J}_s|_{\Omega_3} &= \mathbf{A} - \frac{c}{2}(\mathbf{e}\mathbf{e}^T\mathbf{A} + \mathbf{A}\mathbf{e}\mathbf{e}^T) + \sqrt{c}\|\boldsymbol{\xi}\| I \\ &\quad + c\sqrt{c}\|\boldsymbol{\xi}\|\mathbf{e}\mathbf{e}^T \\ &\succeq -2\overline{\sigma}(\mathbf{A})I + \sqrt{c}\|\boldsymbol{\xi}\| I \\ &\succeq \beta_3 I \end{aligned} \tag{C.13}$$

Now if $\beta_3 > 0$, then $\mathbf{J}_s|_{\Omega_3}$ is positive definite, and all of its eigenvalues are positive. The equilibrium points in Ω_3 can then be shown to be isolated and unstable. □

Appendix D

Geodetic Coordinate Systems

Most of the estimation problems addressed in the present report, were formulated as determining the attitude and position of a frame $\{\mathcal{B}\}$ attached to the vehicle with respect to some reference frame $\{\mathcal{I}\}$. However, we did not further comment on how to chose that reference frame in practice. In this section we will briefly introduce some common reference frames, their associated coordinate systems and transformations between them. The reader is referred to [63], [75], [92], [163], [91], [186], [194] and references therein for more details.

First, we should distinguish between relative navigation, in which the position and orientation of $\{\mathcal{I}\}$ is not known or irrelevant, and absolute navigation where the position and orientation of $\{\mathcal{I}\}$ are known in some geodetic coordinate system. Absolute navigation is vital if one wants to georeference scientific data, when sharing data with other users, or simply be able to reach a certain previously visited point with high accuracy.

We will consider the following Earth fixed reference frames as depicted in Fig. D.1: the Earth Centered Earth Fixed (ECEF) and North East Down (NED). The ECEF frame has its origin at the center of the Earth and has its first axis pointing towards the intersection between the equator and the Greenwich meridian, and its third axis pointing towards the north pole. The right-handed Cartesian coordinates associated to the ECEF frame will be denoted by $(x_{\text{ECEF}}, y_{\text{ECEF}}, z_{\text{ECEF}})$. GPS receivers often output position data in WGS84 ellipsoidal coordinates (φ, λ, h) where φ is the latitude, λ is the longitude and h is the ellipsoidal height, see Fig. D.2. The North East Down (NED) frame is often used in ocean and avionics applications. Its origin is usually fixed at a point on the surface of the earth, and axis points towards North, East and the perpendicular to the reference ellipsoid at that point. The right-handed Cartesian coordinates associated to the NED frame will be denoted by $(x_{\text{NED}}, y_{\text{NED}}, z_{\text{NED}})$.

Usually, the reference frame $\{\mathcal{I}\}$ on which the navigation problem is solved is taken as some local frame, which origin is near the area of study. By doing this, all the numbers involved in computations are kept small and numerical conditioning

problems are minimized. Moreover, it becomes easier to interpret, visualize and display the quantities involved. One of the most common choices is to chose $\{\mathcal{I}\}$ as the NED reference frame at a point near the area of study. Most of the times one is faced with the following practical problem: converting some position data (for instace coming from GPS receivers) from WGS84 ellipsoidal coordinates (φ, λ, h) to NED Cartesian coordinates $(x_{\text{NED}}, y_{\text{NED}}, z_{\text{NED}})$, perform some computations, and eventually transform the results back into WGS84 ellipsoidal coordinates. This is usually done using an intermediate step using ECEF Cartesian coordinates as illustrated in Fig.D.3. We will next give a summary of the necessary coordinate transformations in order to solve the problem.

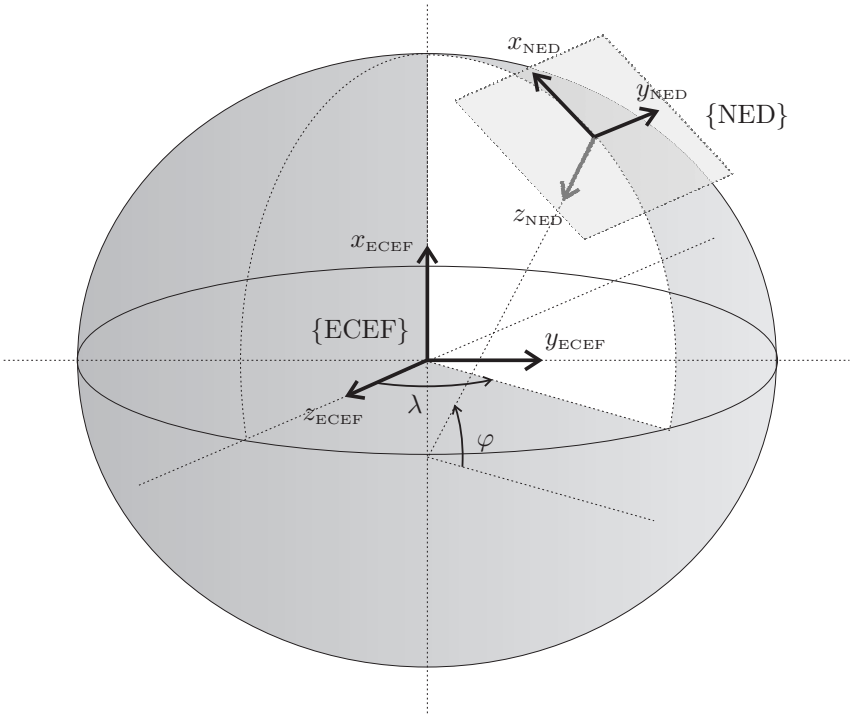


Figure D.1: ECEF and NED inertial reference frames together with a graphical representation of the latitude (φ) and longitude (λ) of the WGS84 geodetic coordinate system.

D.1 WGS84 to ECEF coordinates

The transformation from a WGS84 latitude, longitude and ellipsoidal height triple (φ, λ, h) to an ECEF Cartesian coordinate vector $\mathbf{p}^{\text{E}} = [x_{\text{E}}, y_{\text{E}}, z_{\text{E}}]^T$ is given by

$$\begin{bmatrix} x_{\text{E}} \\ y_{\text{E}} \\ z_{\text{E}} \end{bmatrix} = \begin{bmatrix} (N+h)\cos\varphi\cos\lambda \\ (N+h)\cos\varphi\sin\lambda \\ (\frac{b^2}{a^2}N+h)\sin\varphi \end{bmatrix} \tag{D.1}$$

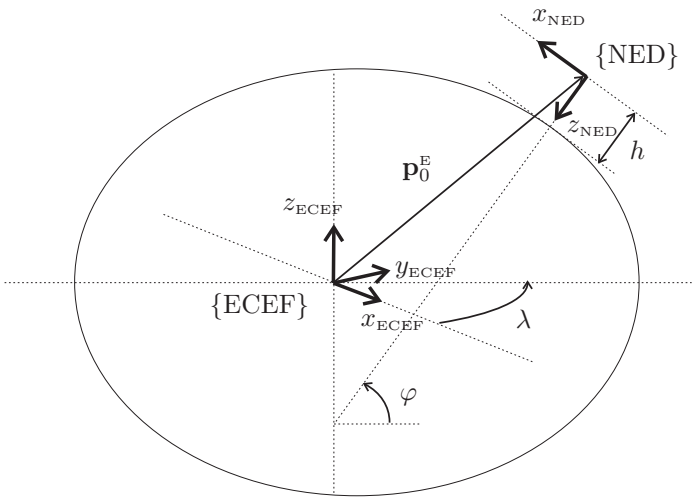


Figure D.2: Latitude, longitude, and ellipsoidal height (φ, λ, h) of the WGS84 geodetic coordinate system.

$$\begin{array}{ccccc} \text{WGS84} & & \text{ECEF} & & \text{ENU} \\ (\varphi, \lambda, h) & \longleftrightarrow & (x_{\text{E}}, y_{\text{E}}, z_{\text{E}}) & \longleftrightarrow & (x_{\text{N}}, y_{\text{N}}, z_{\text{N}}) \end{array}$$

Figure D.3: Geodetic coordinate system transformations.

where

$$N = \frac{a}{(1 - \epsilon^2 \sin^2 \varphi)^{\frac{1}{2}}} \tag{D.2}$$

and a, b, and ϵ are WGS84 ellipsoid parameters given in Table D.1.

D.2 ECEF to NED coordinates

This is a transformation between two Cartesian coordinate systems, which basically involves a rotation and a translation. Suppose we want to transform the ECEF Cartesian coordinates $\mathbf{p}^{\text{E}} = [x_{\text{E}}, y_{\text{E}}, z_{\text{E}}]^T$ into a vector of NED coordinates

Parameters		Comments
a	$= 6378137.0$ m	Semi-major axis
f	$= 1/298.257223563$	Reciprocal of flattening
b	$= 6356752.3142$ m	Semi-minor axis
ϵ^2	$= 6.69437999014 \cdot 10^{-3}$	First Eccentricity Squared
ϵ'^2	$= 6.73949674228 \cdot 10^{-3}$	Second Eccentricity Squared

Table D.1: WGS84 ellipsoid parameters.

$\mathbf{p}^{\mathrm{N}} = [x_{\mathrm{N}}, y_{\mathrm{N}}, z_{\mathrm{N}}]^T$ with origin in $(\varphi_0, \lambda_0, h_0)$. Let $\mathbf{p}_0^{\mathrm{E}} = [x_0, y_0, z_0]$ denote the ECEF coordinates of the WGS84 geodetic coordinates $(\varphi_0, \lambda_0, h_0)$, obtained for instance using (D.1). Then,

$$\mathbf{p}^{\mathrm{N}} = {}^{\mathrm{N}}_{\mathrm{E}}\mathcal{R}(\mathbf{p}^{\mathrm{E}} - \mathbf{p}_0^{\mathrm{E}}), \tag{D.3}$$

where ${}^{\mathrm{N}}_{\mathrm{E}}\mathcal{R} \in SO(3)$ is the rotation matrix between the ECEF and NED frames given by

$${}^{\mathrm{N}}_{\mathrm{E}}\mathcal{R} = \begin{bmatrix} -\sin\varphi_0 \cos\lambda_0 & -\sin\varphi_0 \sin\lambda_0 & \cos\varphi_0 \\ -\sin\lambda_0 & \cos\lambda_0 & 0 \\ -\cos\varphi_0 \cos\lambda_0 & -\cos\varphi_0 \sin\lambda_0 & -\sin\varphi_0 \end{bmatrix}. \tag{D.4}$$

The inverse transformation from NED to ECEF coordinates can then be easily computed as

$$\mathbf{p}^{\mathrm{N}} = \mathbf{p}_0^{\mathrm{E}} - {}^{\mathrm{E}}_{\mathrm{N}}\mathcal{R}\mathbf{p}^{\mathrm{N}}, \tag{D.5}$$

where ${}^{\mathrm{E}}_{\mathrm{N}}\mathcal{R} = {}^{\mathrm{N}}_{\mathrm{E}}\mathcal{R}^T \in SO(3)$ is the rotation matrix between the NED and ECEF frames.

D.3 ECEF to WGS84 coordinates

This is possibly the most involved among the coordinate transformations that we will describe. There are several iterative and closed form methods and approximations to solve the problem. See for instance [63] or [186] for an iterative solution. However, the most accurate method, which also turns out to be numerically simple, is the exact closed form solution presented in [91], [92]. Given an ECEF position $\mathbf{p}^{\mathrm{E}} = [x_{\mathrm{E}}, y_{\mathrm{E}}, z_{\mathrm{E}}]^T$ the corresponding WGS84 geodetic latitude, longitude and height (φ, λ, h) is given by

$$\varphi = \operatorname{atan}\left(\frac{z_{\mathrm{E}} + \epsilon'^2 z_0}{r}\right), \tag{D.6}$$

$$\lambda = \operatorname{atan2}(y_{\mathrm{E}}, x_{\mathrm{E}}), \tag{D.7}$$

$$h = U\left(1 - \frac{b^2}{aV}\right), \tag{D.8}$$

where

$$
\begin{aligned}
r &= \sqrt{x_{\mathrm{E}}^2 + y_{\mathrm{E}}^2}, \\
E &= \sqrt{a^2 - b^2}, \\
F &= 54 b^2 z_{\mathrm{E}}^2, \\
G &= r^2 + (1-\epsilon^2) z_{\mathrm{E}}^2 - \epsilon^2 E^2, \\
c &= \frac{\epsilon^4 F r^2}{G^3}, \\
s &= (1 + c + \sqrt{c^2 + 2c})^{\frac{1}{3}}, \\
P &= \frac{F}{3(s + \frac{1}{s} + 1)^2 G^2}, \\
Q &= \sqrt{(}1 + 2\epsilon^4 P), \\
r_0 &= -\frac{\epsilon^2 P r}{1+Q} + \sqrt{\frac{1}{2} a^2 (1 + \frac{1}{Q}) - \frac{(1-\epsilon^2) P z_{\mathrm{E}}^2}{Q(1+Q)} - \frac{P r^2}{2}}, \\
U &= \sqrt{(r - \epsilon^2 r_0)^2 + z_{\mathrm{E}}^2}, \\
V &= \sqrt{(r - \epsilon^2 r_0)^2 + (1-\epsilon^2) z_{\mathrm{E}}^2}, \\
z_0 &= \frac{b^2 z_{\mathrm{E}}}{aV}.
\end{aligned}
$$

Bibliography

[1] J. Abel. A divide and conquer approach to least-squares estimation with application to range-difference-based localization. In *Proc. International Conference on Acoustics, Speech, and Signal Processing ICASSP'89*, pages 2144–2147, May 1989.

[2] J. Abel. Optimal sensor placement for passive source localization. In *Proc. International Conference on Acoustics, Speech, and Signal Processing ICASSP'90*, pages 2927–2930, Apr 1990.

[3] J. Abel. Existence and Uniqueness of GPS Solutions. *IEEE Transactions on Aerospace and Electronic Systems*, 27(6):952–956, Novemeber 1991.

[4] J. Abel and J. Smith. The spherical interpolation method for closed-form passive source localization using range difference measurements. In *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 471–474, April 1987.

[5] J. Abel and J. Smith. Source range and depth estimation from multipath range difference measurements. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37:1157–1165, August 1989.

[6] J.S. Abel. *Localization using Range Differences*. PhD thesis, Stanford University, 1989.

[7] A.C. Aitken. On Least Squares and Linear Combinations of Observations. *Proceedings of the Royal Society of Edinburgh*, 55:42–48, 1935.

[8] A. Alcocer, J. Jouffroy, P. Oliveira, and A. Pascoal. Diffusion-based trajectory observers with variance constraints. In *Proc. of the IFAC Conference on Control Applications in Marine Systems (CAMS'07), Bol, Croatia.*, 2007.

[9] A. Alcocer, P. Oliveira, and A. Pascoal. A nonlinear filter for Range-Only attitude and position estimation. In *In Proc. of MCMC'2006 - 7th Conference on Manoeuvring and Control of Marine Craft, Lisbon, Portugal*, 2006.

[10] A. Alcocer, P. Oliveira, and A. Pascoal. Study and Implementation of an EKF GIB-Based Underwater Positioning System. *Journal of Control Engineering Practice*, 15(6):689–701, 2007.

[11] A. Alcocer, P. Oliveira, and A. Pascoal. Study and Implementation of an EKF GIB-Based Underwater Positioning System. In *Proc. of the IFAC Conference on Control Applications in Marine Systems (CAMS'04), Ancona, Italy.*, July 2004.

[12] A. Alcocer, P. Oliveira, and A. Pascoal. Underwater Acoustic Positioning Systems Based on Buoys with GPS. In *Proceedings of Eigth European Conference on Underwater Acoustics ECUA'06, Carvoeiro, Portugal*, June 2006.

[13] A. Alcocer, P. Oliveira, A. Pascoal, R. Cunha, and C. Silvestre. A dynamic estimator on $SE(3)$ using Range-Only measurements. In *Proceedings of the 47th IEEE Conference on Decision and Control*, December 2008.

[14] A. Alcocer, P. Oliveira, A. Pascoal, and J. Xavier. Estimation of Attitude and Position from Range only Measurements using Geometric Descent Optimization on the Special Euclidean Group. In *International Conference on Information Fusion, FUSION'06, Florence, Italy.*, July 2006.

[15] A. Alcocer, P. Oliveira, A. Pascoal, and J. Xavier. Maximum Likelihood Attitude and Position Estimation from Pseudo-Range Measurements using Geometric Descent Optimization. In *Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, USA*, pages 3754–3759, December 2006.

[16] W.F. Ames. *Numerical Methods for Partial Differential Equations (2nd Ed.).* Academic Press, 1977.

[17] B. D. O. Anderson and J. B. Moore. *Optimal Filtering.* Prentice Hall, 1979.

[18] T.W. Anderson. *The Statistical Analysis of Time Series.* Wiley, 1971.

[19] D. Angeli. Almost global stabilization of the inverted pendulum via continuous state feedback. *Automatica, 12(2):161–184*, 37:1103–1108, July 2001.

[20] A. Bahr and J. Leonard. Cooperative Localization for Autonomous Underwater Vehicles. In *In Proceedings of the 10th International Symposium on Experimental Robotics (ISER), Rio de Janeiro, Brasil, July 2006*, 2006.

[21] A. Bahr and J. Leonard. Minimizing trilateration errors in the presence of uncertain landmark positions. In *Proc. 3rd European Conference on Mobile Robots (ECMR)*, pages 48–53, Freiburg, Germany, September 2007.

[22] S. Bancroft. An Algebraic Solution of the GPS Equations. *IEEE Transactions on Aerospace and Electronic Systems*, 21(1):56–59, January 1985.

[23] I.Y. Bar-Itzhack and Y. Oshman. Attitude Determination from Vector Observations: Quaternion Estimation. *IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-21, No. 1*, pages 128–136, January 1985.

[24] Y. Bar-Shalom and T. Fortmann. *Tracking and data association.* New York: Academic Press, 1988.

[25] A. Beck and P. Stoica. Exact and approximate solutions source localization problems. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 56:1770–1778, May 2008.

[26] A. Beck, M. Teboulle, and Z. Chikishev. Iterative Minimization Schemes for Solving the Single Source Localization Problem. *SIAM Journal on Optimization*, 19(3):1397–1416, 2008.

[27] C. Belta and V. Kumar. A SVD-Based Projection Method for Interpolation on SE(3). *IEEE Trans. Robotics and Automation, 18(3):334–345.*, June 2002.

[28] N. Bergman. *Recursive Bayesian Estimation: Navigation and Trucking Applications.* PhD thesis, Linkoping University, 1999.

[29] D.P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, MA, 1995. 2nd edition 1999.

[30] V.P. Bingman and K. Cheng. Mechanisms of animal global navigation: comparative perspectives and enduring challenges. *Ethology Ecology & Evolution*, 17(4):295–318, 2005.

[31] W.M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry.* Academic Press, New York, 1975.

[32] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, March 2004.

[33] R.W. Brockett. Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems. In *Proceedings of the 27th IEEE Conference on Decision and Control, Austin, TX, USA*, pages 799–803, December 1988.

[34] F. Bullo and A. Lewis. *Geometric Control of Mechanical Systems.* Springer Verlag, 2004.

[35] A. Caiti, A. Garulli, F. Livide, and D. Prattichizzo. Localization of Autonomous Underwater Vehicles by Floating Acoustic Buoys: A Set-Membership Approach. *IEEE Journal of Oceanic Engineering*, 30(1):140–152, January 2005.

[36] M. Cao, B.D.O. Anderson, and A.S. Morse. Sensor network localization with imprecise distances. *Systems & Control Letters*, 55:887–893, 2006.

[37] D. Carta. Optimal estimation of undersea acoustic transponder locations. In *Proc. OCEANS*, volume 10, pages 466–471, 1978.

[38] J. Chaffee and J. Abel. GDOP and the Cramer-Rao Bound. In *In Proc. IEEE Position, Location, And Navigation Symposium, Las Vegas, NV, USA*, pages 663–668, 1994.

[39] J. Chaffee and J. Abel. On the exact solutions of pseudorange equations. *IEEE Transactions on Aerospace and Electronic Systems*, 30(4):1021–1030, Oct 1994.

[40] Y.T. Chan and K.C. Ho. A Simple and Efficient Estimator for Hyperbolic Location. *IEEE Transactions on Signal Processing, vol. 42, Issue: 8*, pages 1905–1915, August 1994.

[41] S.G. Chappell, J.C. Jalbert, P. Pietryka, and J. Duchesney. Acoustic communication between two autonomous underwater vehicles. In *Proc. of the 1994 Symposium on Autonomous Underwater Vehicle Technology AUV'94*, pages 462–469, Jul 1994.

[42] K.W. Cheung, W.K Ma, and Y.T. Chan. Least squares algorithms for time-of-arrival based mobile location. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 52:1121–1128, April 2004.

[43] K.W. Cheung, W.K Ma, and H.C. So. Accurate approximation algorithm for toa-based maximum likelihood mobile location using semidefinite programming. In *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 145–148, 2004.

[44] J. Christensen. LBL NAV - An Acoustic Transponder Navigation System. In *Proc. OCEANS'79*, pages 507–512, 1979.

[45] C.E. Cohen. *Attitude Determination Using GPS*. PhD thesis, Dept. of Aeronautics and Astronautics, Stanford Univ., Stanford, CA., December 1992.

[46] L.D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Trans. Pattern Analysis and Machine Inteligence*, 15(11):1131–1147, November 1993.

[47] John J. Craig. *Introduction to robotics: mechanics and control*. Addison Wesley, 2nd edition, 1986.

[48] J.L. Crassidis and F.L. Markley. New Algorithm for Attitude Determination Using Global Positioning System Signals. *AIAA Journal of Guidance, Control, and Dynamics, Vol. 20, No. 5*, pages 891–896, Sept.-Oct. 1997.

[49] F. Cretollier and P. Morvan. Use of a Rangemeter in Advanced and Modular Subsea Positioning Solutions. In *In Proc. of MCMC'2006 - 7th Conference on Manoeuvring and Control of Marine Craft, Lisbon, Portugal*, 2006.

[50] Y. Cui and S.S. Ge. Autonomous vehicle positioning with gps in urban canyon environments. *IEEE Transactions on Robotics and Automation*, 19:15–25, February 2003.

[51] R. Cunha, C. Silvestre, and J. Hespanha. Output feedback control for stabilization on SE(3). In *Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, USA*, pages 3825–3830, December 2006.

[52] J. Curcio, J. Leonard, J. Vaganay, A. Patrikalakis, A. Bahr, D. Battle, H. Schmidt, and M. Grund. Experiments in Moving Baseline navigation using autonomous surface craft. In *Proc. IEEE OCEANS'05 , Washington, D.C.*, 2005.

[53] John Dattorro. *Convex Optimization & Euclidean Distance Geometry*. Meboo Publishing, USA, 2006.

[54] M. Deffenbaugh, J.G. Bellingham, and H. Schmidt. The relationship between spherical and hyperbolic positioning. In *Proceedings of MTS/IEEE Conference OCEANS '96, Fort Lauderdale, FL, USA*, volume 2, pages 590–595, September 1996.

[55] M. do Carmo. *Riemannian Geometry*. Birkhäuser, Boston, 1992.

[56] L. Doherty, L. E. Ghaoui, and S. J. Pister. Convex position estimation in wireless sensor networks. In *Proc. 20th INFOCOM*, volume 3, pages 1655–1663, 2001.

[57] A. Edelman, T.A. Arias, and S.T. Smith. The Geometry of Algorithms with Orthogonality Constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[58] R.L. Eubank. *Nonparametric Regression and Spline Smoothing (2nd Ed.)*. Marcel Dekker, 1999.

[59] R. Eustice, L. Whitcomb, H. Singh, and M. Grund. Experimental Results in Synchronous-Clock One-Way-Travel-Time AcousticNavigation for Autonomous Underwater Vehicles. In *In Proc. of the 2007 IEEE International Conference on Robotics and Automation Roma, Italy*, April 2007.

[60] N. Fairfield, G. Kantor, and D. Wettergreen. Real-time slam with octree evidence grids for exploration in underwater tunnels. *Journal of Field Robotics, Special Issue on SLAM in the Field*, 24(1-2):03–21, February 2007.

[61] V.L. Ferrini, D.J. Fornari, T.M. Shank, J.C. Kinsey, S.A. Soule, S.M. Carbotte, M.A. Tivey, L.L. Whitcomb, D.R. Yoerger, and J. Howland. Submeter bathymetric mapping of volcanic and hydrothermal features on the East Pacific Rise crest at 9'50^{o}N. *Geochem. Geophys. Geosyst.*, 8(1):1–33, 2007.

[62] E. Fiorelli, N.E. Leonard, P. Bhatta, D. Paley, R. Bachmayer, and D.M. Fratantoni. Multi-AUV Control and Adaptive Sampling in Monterey Bay. *IEEE Journal of Oceanic Engineering*, 31(4):935–948, October 2006.

[63] Thor I. Fossen. *Marine Control Systems: Guidance, Navigation, and Contol of Ships, Rigs and underwater Vehicles*. Tapir Trykkeri, 2002.

[64] R. Franklin. LBL NAV - An Acoustic Transponder Navigation System. In *Proc. OCEANS'72, Newport, RI, USA*, pages 451–455, 1972.

[65] L. Freitag, M. Johnson, M. Grund, S. Singh, and J. Preisig. Integrated Acoustic Communication and Navigation for Multiple UUVs. In *Proceedings of MTS/IEEE Conference OCEANS '96, Fort Lauderdale, FL, USA.*, pages 59–64, 1996.

[66] B. Friedlander. On the Cramer-Rao bound for time delay and Doppler estimation. *IEEE Transactions on Information Theory*, IT-30(3):575–580, May 1984.

[67] B. Friedlander. A Passive Localization Algorithm and Its Accuracy Analysis. *IEEE Journal of Oceanic Engineering*, 12(1):234–245, January 1987.

[68] A. Gadre. *Observability Analysis in Navigation Systems with an Underwater Vehicle Application*. PhD thesis, Virginia Polytechnic Institute and State University, January 2007.

[69] A. Gadre and D. Stilwell. Toward Underwater Navigation Based on Range Measurements From a Single Location. In *Proc. IEEE International Conference on Robotics and Automation, New Orleans, LA*, pages 4472–4477, 2004.

[70] A. Gadre and D. Stilwell. A Complete Solution to Underwater Navigation In the Presence of Unknown Currents Based On Range Measurements From a Single Location. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems, Edmonton AB, Canada*, 2005.

[71] E. Geyer, P. Creamer, J. D'Appolito, and R. Gains. Characteristics and capabilities of navigation systems for unmanned untethered submersibles. In *In Proc. 5th International Symposium on Unmanned Untethered Submersible Technology, Durhan, NH*, pages 320–347, 1987.

[72] R. Ghabcheloo, A. Aguiar, A. Pascoal, and C. Silvestre. Coordinated path-following control of multiple auvs in the presence of communication failures and time delays. In *In Proc. of MCMC'2006 - 7th Conference on Manoeuvring and Control of Marine Craft, Lisbon, Portugal*, 2006.

[73] L. Godinho and J. Natário. An Introduction to Riemannian Geometry with Applications. Lecture Notes. Lisbon, August 2004.

[74] G. Golub and C. Van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, MD, second edition., Baltimore, MD, 1989.

[75] Mohinder S. Grewal, Lawrence R. Weill, and Angus P. Andrews. *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley and Sons, New York, 2001.

[76] H.L. Groginsky. Position estimation using only multiple simultaneous range measurements. *IRE Transactions on Aeronautical and Navigational Electronics*, pages 178–187, September 1959.

[77] F. Gustafsson and F. Gunnarsson. Positioning using time-difference of arrival measurements. In *InProc. IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP'03*, volume 6, pages 553–556, April 2003.

[78] W.R. Hahn and S.A. Tretter. Optimum processing for delay-vector estimation in passive signal arrays. *IEEE Transactions on Information Theory*, IT-10:608–614, May 1973.

[79] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration: Structure-preserving Algorithms for Ordinary Differential Equations*. Springer, 2nd edition, 2006.

[80] J. Hartsfield. Single transponder range only navigation geometry (strong) applied to remus autonomous under water vehicles. Master's thesis, Massachusetts Institute of Technology and the Woods Hole Oceanographic Institution, August 2005.

[81] U. Helmke and J. Moore. *Optimization and Dynamical Systems*. Springer-Verlag, London, 1994, ISBN: 4-540-19857-1.

[82] T.D. Henry. Acoustic Transponder Navigation. In *IEEE Position Location and Navigation Symp.*, pages 237–244, 1978.

[83] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, Boca Raton, FL, 1985.

[84] Y. Huang, J. Benesty, G. Elko, and R. Mersereau. Real-Time Passive Source Localization: A Practical Linear-Correction Least-Squares Approach. *IEEE Transactions on Speech and Audio Processing*, 9(8):943–956, November 2001.

[85] M.M. Hunt, W.M. Marquet, D.A. Moller, K.R. Peal, W.K. Smith, and R.C. Spindell. An Acoustic Navigation System. Technical Report WHOI-74-6, Woods Hole Oceanographic Institution, Woods Hole, Massachusetts 02543 USA, December 1974.

[86] S. Jesus, M. Porter, Y. Stephane, X. Demoulin, O. Rodriguez, and E. Coelho. Single Hydrophone Source Localization. *IEEE Journal of Oceanic Engineering*, 25(3):337–346, 2000.

[87] J. Jouffroy and T.D. Nguyen. Towards on-line underwater vehicle trajectory estimation using trajectory observers. In *Proc. IEEE OCEANS'04 , Kobe, Japan, 2004*, 2004.

[88] J. Jouffroy and J. Opderbecke. Underwater vehicle trajectory estimation using contracting PDE-based observers. In *Proceedings of the American Control Conference,Boston, Massachusetts USA*, 2004.

[89] J. Jouffroy and J. Opderbecke. Underwater vehicle navigation using diffusion-based trajectory observers. *IEEE Journal of Oceanic Engineering, to appear*, 32(2), 2007.

[90] J. Jouffroy and J. Reger. An algebraic perspective to single-transponder underwater navigation. In *In Proc. of the 2006 IEEE International Conference on Control Applications Munich, Germany*, October 2006.

[91] J.Zhu. Conversion of Earth-centered Earth-fixed coordinates to geodetic coordinates. *IEEE Transactions on Aerospace and Electronic Systems*, 30(3):957–961, 1994.

[92] E.D. Kaplan. *Understanding GPS: principles and applications.* Artech House, 1996.

[93] R. Karlsson, F. Gusfafsson, and T. Karlsson. Particle filtering and cramer-rao lower bound for underwater navigation. In *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003*, April 2003.

[94] R. Karlsson, F. Gustafsson, and T. Karlsson. Particle filtering and Cramer-Rao lower bound for underwater navigation. Technical Report LiTH-ISY-R-247 4 2002, Linkoping University, 2002.

[95] S.M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory.* Prentice-Hall, Upper Saddle River, New Jersey, USA, 1993.

[96] H.K. Khalil. *Nonlinear Systems.* Prentice-Hall, Englewood Cliffs, NJ, 3rd edition, 2000.

[97] J.C. Kinsey, R. Eustice, and L.L. Whitcomb. A Survey of Underwater Vehicle Navigation: Recent Advances and New Challenges. In *In Proc. of MCMC'2006 - 7th Conference on Manoeuvring and Control of Marine Craft, Lisbon, Portugal*, 2006.

[98] J.C. Kinsey and L.L. Whitcomb. Preliminary Field Experience with the DVL-NAV Integrated Navigation System for Oceanographic Submersibles. *Journal of Control Engineering Practice*, 12(12):1541–1548, 2004.

[99] J.C. Kinsey and L.L. Whitcomb. Adaptive Identification on the Group of Rigid-Body Rotations and its Application to Underwater Vehicle Navigation. *IEEE Trans. on Robotics , 23(1):124–136*, Feb. 2007.

[100] J.C. Kinsey, L.L. Whitcomb, D.R. Yoerger, J.C. Howland, V.L. Ferrini, and O. Hegrenes. New navigation post-processing tools for oceanographic submersibles. In *In Eos Trans. AGU, 87(52), Fall Meet. Suppl., 2006. Abstract OS33A-1678*, 2006.

[101] Jan J. Koenderink. *Solid shape*. MIT Press, Cambridge, MA, USA, 1990.

[102] A.M. Ladd, K.E. Bekris, A.P. Rudys, D.S. Wallach, and L.E. Kavraki. On the feasibility of using wireless ethernet for indoor localization. *IEEE Transactions on Robotics and Automation*, 20:555–559, June 2004.

[103] C. LaPointe. Virtual long baseline (vlbl) autonomous underwater vehicle navigation using a single transponder. Master's thesis, Massachusetts Institute of Technology and the Woods Hole Oceanographic Institution, June 2006.

[104] M.B. Larsen. *Autonomous Navigation of Underwater Vehicles*. PhD thesis, Department of Automation, Technical University of Denmark, February 2001.

[105] M.B. Larsen. Synthetic long baseline navigation of underwater vehicles. In *Proc. IEEE OCEANS'00 , Providence, RI, USA*, pages 2043–2050, September 2000.

[106] T.D. Larsen, N.K. Poulsen, N.A. Andersen, and O. Ravn. Incorporation of time delayed measurements in a discrete-time kalman filter. In *Proceedings of the 37th Conference on Decision and Control, Tampa, Florida, USA*, pages 3972–3977, 1998.

[107] H.H. Lee. Optimal estimation of vehicle position in an acoustic transponder navigation system. In *Proc. OCEANS'73*, pages 596–601, 1973.

[108] John M. Lee. *Riemannian Manifolds: An Introduction to Curvature*. Springer Verlag, New York, 1997.

[109] John M. Lee. *Introduction to Topological Manifolds*. Springer Verlag, New York, 2000.

[110] John M. Lee. *Introduction to Smooth Manifolds*. Springer Verlag, New York, 2003.

[111] J. Leonard, A. Bennett, C. Smith, and H. Feder. Autonomous underwater vehicle navigation. MIT Marine Robotics Laboratory Technical Memorandum 98-1. 1998.

[112] J. Leonard, A. Bennett, C. Smith, and H. Feder. Autonomous underwater vehicle navigation. MIT Marine Robotics Laboratory Technical Memorandum 98-1. 1998.

[113] X.R. Li and V.J. Jilkov. Survey of maneuvering target tracking. part i: dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, Oct 2003.

[114] David G. Luenberger. *Introduction to Linear and Nonlinear programming*. Addison Wesley, 1984.

[115] H. Lütkepohl. *Handbook of Matrices*. John Wiley and Sons, 1996.

[116] K.V. Mackenzie. Nine-term equation for the sound speed in the oceans. *J. Acoust. Soc. Am. 70(3)*, pages 807–812, 1981.

[117] J.R. Magnus and H. Neudecker. *Matrix Differential Calculus With Applications in Statistics and Econometrics*. Wiley Series in Probability and Statistics, Chichester, 1988.

[118] R. Mahony and J.H. Manton. The Geometry of the Newton Method on Non-Compact Lie Groups. *Journal of Global Optimization, Vol. 23*, pages 309–327, 2002.

[119] M. Malisoff, M. Krichman, and E. Sontag. Global stabilization for systems evolving on manifolds. *Journal of Dynamical and Control Systems , 12(2):161–184*, April. 2006.

[120] D.E. Manolakis. Efficient solution and performance analysis of 3-d position estimation by trilateration. *IEEE Transactions on Aerospace and Electronic Systems*, 32(4):1239–1248, Oct 1996.

[121] F.L. Markley. Attitude Determination using Vector Observations and the Singular Value Decomposition. *Journal of the Astronautical Sciences, Vol. 36, No. 3*, pages 245–258, 1988.

[122] F.L. Markley. Attitude Determination using Vector Observations: a Fast Optimal Matrix Algorithm. *Journal of the Astronautical Sciences, Vol. 41, No. 2*, pages 261–280, June 1993.

[123] A. Matos and N. Cruz. AUV navigation and guidance in a moving acoustic network. In *Proc. IEEE OCEANS'2005 Europe, Brest, France*, June 2005.

[124] L. Mili, M. Cheniae, N. Vichare, and P. Rousseeuw. Robust state estimation based on projection statistics. *IEEE Transactions on Power Systems*, 11(2):1118–1127, May 1996.

[125] P. H. Milne. *Underwater Acoustic Positioning Systems*. Gulf Publishing, Houston, 1983.

[126] M. Moakher. Means and Averaging in the Group of Rotations. *SIAM Journal on Matrix Analysis and Applications 24/1*, pages 1–16, 2002.

[127] M.C. Moreau, E.P. Davis, J.R. Carpenter, D. Kelbel, G.W. Davis, and P. Axelrad. Results from the gps flight experiment on the high earth orbit amsat oscar-40 spacecraft. In *Proceedings of the ION GPS*, 2002.

[128] M. Morgado, P. Oliveira, C. Silvestre, and J. Vasconcelos. USBL/INS Integration Technique for Underwater Vehicles. In *In Proc. of MCMC'2006 - 7th Conference on Manoeuvring and Control of Marine Craft, Lisbon, Portugal*, 2006.

[129] D. Mortari. The Attitude Error Estimator. In *International Conference on Dynamics and Control of Systems and Structures in Space 2002*, King College, Cambridge, England, July 14–18 2002.

[130] R.M. Murray, Z. Li, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, FL, USA, 1994.

[131] A. Nadler, Y. Bar-Itzhack, and H. Weiss. On Algorithms for Attitude Estimation using GPS. In *Proceedings of the 2000 IEEE Conference on Decision and Control, Sidney, Australia*, pages 3321–3326, December 2000.

[132] P. Oliveira. MMAE Terrain Navigation for Underwater Vehicles using PCA. *International Journal of Control*, 80(7):1008–1017, 2007.

[133] P. Oliveira. MMAE Terrain Reference Navigation for Underwater Vehicles using Eigen Analysis. In *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference, Sevilla, Spain*, December 2005.

[134] E. Olson, J. Leonard, and S. Teller. Robust range-only beacon localization. In *Proc. of IEEE/OES Autonomous Underwater Vehicles AUV'04*, 2004.

[135] E. Olson, J. Leonard, and S. Teller. Robust Range-Only Beacon Localization. *IEEE Journal of Oceanic Engineering*, 31(4):949–958, October 2006.

[136] E. Olson, J. Leonard, and S. Teller. Robust Range-Only Beacon Localization. *IEEE Journal of Oceanic Engineering*, 31(4):949–958, October 2006.

[137] J. Opderbecke. At-sea calibration of a usbl underwater vehicle positioning system. In *Proc. MTS/IEEE OCEANS'97 , Halifax, NS, Canada*, 1997.

[138] F.C. Park. Distance Metrics on the Rigid Body Motions with Applications to Mechanism Design. *Transactions of the ASME Journal of Mechanical Design, Vol.117(1), pp:48–54.*, March 1995.

[139] F.C. Park, J. Kim, and C. Kee. Geometric Descent Algorithms for Attitude Determination Using the Global Positioning System. *AIAA Journal of Guidance, Control, and Dynamics. 23(1)*, January-February 2000.

[140] D. Popaf, A. Sandersont, R. Komerskat, S. Mupparapd, D. Blidberg, and S. Chappelf. Adaptive sampling algorithms for multiple autonomous underwater vehicles. In *Proc. IEEE/OES Autonomous Underwater Vehicles AUV'04*, 2004.

[141] J.W. Powers. Range trilateration error analysis. *IEEE Transactions on Aerospace and Electronic Systems*, AES-2:572–585, July 1966.

[142] N. Priyantha. *The Cricket Indoor Location System*. PhD thesis, Massachusetts Institute of Technology, June 2005.

[143] C.R. Rao. *Linear Statistical Inference and Its Applications*. 2nd ed., John Wiley and Sons, New York, 1973.

[144] C. Reeder, D. Odell, A. Okamoto, M. Anderson, and D. Edwards. Two-hydrophone heading and range sensor applied to formation-flying for auvs. In *Proc. IEEE OCEANS'04 , Kobe, Japan, 2004*, pages 517–523, 2004.

[145] M.J. Rendas and I.M.J Lourtie. Hybrid Navigation System for Long Range Operation. In *Proceedings of the 1994 Symposium on Autonomous Underwater Vehicle Technology, AUV '94*, pages 353–359, Cambridge, MA, USA, July 1994.

[146] H. Rice, S. Kelmenson, and L. Mendelsohn. Geophysical navigation technologies and applications. In *In Proc. IEEE/ION Position, Location, And Navigation Symposium, Monterey, California, USA 2004*, pages 618–624, 2004.

[147] A. Ross and J. Jouffroy. Remarks on the observability of single beacon underwater navigation. In *Int. Symp. on Unmanned Untethered Submersible Technology (UUST'05). Durham, NH.*, 2005.

[148] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268, November 1992.

[149] S. Dasgupta S. Dandach, B. Fidan and B. Anderson. Adaptive Source Localization by Mobile Agents. In *Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, USA*, pages 2045–2050, December 2006.

[150] S. Gallot and D. Hulin and J. Lafontaine. *Riemannian Geometry*. 2nd ed., Springer Verlag, Berlin, 1990.

[151] A.H. Sayed, A. Tarighat, and N. Khajehnouri. Network based wireless location. *IEEE Signal Process. Mag.*, 22(4):24–40, July 2005.

[152] H. Schau and A. Robinson. Passive source localization employing intersecting spherical surfaces from time-of-arrival differences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35:1223–1225, August 1987.

[153] R.O. Schmidt. A new approach to geometry of range difference location. *IEEE Transactions on Aerospace and Electronic Systems*, AES-8:821–835, November 1972.

[154] L. Silva, J. Santos, M. Neves, C. Silvestre, P. Oliveira, and A. Pascoal. Tools for the diagnosis and automated inspection of semi-submerged structures. In *Proceedings of the 13th International Harbour Congress, Antwerpen, Belgium, March-April 2003*, March-April 2003.

[155] J. Smith and J. Abel. Closed-form least-squares source location estimation from range-difference measurements. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-35:1661–1669, December 1987.

[156] J. Smith and J. Abel. The spherical interpolation method of source localization. *IEEE Journal of Oceanic Engineering, vol. 12, Issue: 1*, pages 246–252, January 1987.

[157] S. Smith and D. Kronen. Experimental results of an inexpensive short baseline acoustic positioning system for AUV navigation. In *Proceedings of MTS/IEEE Conference OCEANS '97, Halifax, NS*, pages 714–720, 1997.

[158] S.T. Smith. *Geometric Optimization Methods for Adaptive Filtering.* PhD thesis, Harvard University, Cambridge, Massachussets, 1993.

[159] S.T. Smith. Covariance, Subspace, and Intrinsic Cramér-Rao Bounds. *IEEE Trans. Signal Processing, 53(5): 1610–1630*, May 2005.

[160] Anthony Man-Cho So and Yinyu Ye. Theory of semidefinite programming for sensor network localization. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 405–414, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[161] T. Song. Observability of Target Tracking with Range-Only Measurements. *IEEE Journal of Oceanic Engineering*, 24(3):383–387, 1999.

[162] M.A. Stephens. Vector Correlation. *Biometrika, 66(1):41-48*, 1979.

[163] Gilbert Strang and Kai Borre. *Linear Algebra, Geodesy, and GPS.* Wellesley-Cambridge Press, 1997.

[164] D. Sun and J.L. Crassidis. Observability Analysis of Six Degree of Freedom State Determination Using Vector Observations. *AIAA Journal of Guidance, Control, and Dynamics, Vol. 25, No. 6*, pages 1149–1157, Nov.-Dec. 2002.

[165] D. Sun and J.L. Crassidis. Observability Analysis of Six Degree of Freedom State Determination Using Vector Observations. *AIAA Journal of Guidance, Control, and Dynamics, Vol. 25, No. 6*, pages 1149–1157, Nov.-Dec. 2002.

[166] F. Teixeira. *Terrain-Aided Navigation and Geophysical Navigation of Autonomous Underwater Vehicles*. PhD thesis, ISR/IST, Portugal, 2007.

[167] F. Teixeira and A. Pascoal. Geophysical navigation of autonomous underwater vehicles. In *IFAC Conference on Control Applications in Marine Systems - CAMS'07*, Bol, Croatia, 2007.

[168] F. Thomas and L. Ros. Revisiting Trialateration for Robot Localization. *IEEE Trans. on Robotics*, 21(1):93–98, February 2005.

[169] H.G. Thomas. GIB buoys: An interface between space and depths of the oceans. In *Proceedings of IEEE Autonomous Underwater Vehicles, Cambridge, MA, USA*, pages 181–184, August 1998.

[170] H.L. Van Trees. *Detection, Estimation, and Modulation Theory (Part I)*. John Wiley and Sons, 1968.

[171] S. Tuohy. *Geophysical Map Representation, Abstraction and Interrogation for Autonomous Underwater Vehicle Navigation*. PhD thesis, MIT, 1993.

[172] S. Tuohy, J. Leonard, J. Bellingham, N. Patrikalakis, and C. Chryssostomidis. Map based navigation for autonomous underwater vehicles. *International Journal of Offshoreand PolarEngineering*, 6(1):9–18, March 1996.

[173] C. Tyren. Magnetic anomalies as a reference for ground-speed and map-matching navigation. *The Journal of Navigation*, 35(2):242–254, May 1982.

[174] C. Tyren. Magnetic terrain navigation. In *In Proc. 5th International Symposium on Unmanned Untethered Submersible Technology, Durhan, NH*, pages 245–256, 1987.

[175] D. Ucinski. *Optimal Measurement Methods for Distributed Parameter System Identification*. CRC Press, Boca Raton, FL, 2004.

[176] R.J. Urick. *Principles of Underwater Sound*. Peninsula Pub, 3 edition, 1996.

[177] J. Vaganay, P. Baccou, and B. Jouvencel. Homing by acoustic ranging to a single beacon. In *Proc. IEEE OCEANS'00 , Providence, RI, USA*, pages 1457–1461, September 2000.

[178] J. Vaganay, J. Bellingham, and J. Leonard. Comparison of Fix Computation and Filtering for Autonomous Acoustic Navigation. In *Proceedings of the 6th IARP Workshop on Underwater Robotics, Toulon, La Seyne, France, 1996*, March 1996.

[179] J. Vaganay, J. Leonard, and J. Bellingham. Outlier Rejection for Autonomous Acoustic Navigation. In *Proceedings of IEEE International Conference on Robotics and Automation, Minneapolis, MN, USA*, pages 2174–2181, 1996.

[180] K. Vickery. Acoustic positioning systems. A practical overview of current systems. In *Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles., Fort Lauderdale, FL, USA.*, pages 5–17, August 1998.

[181] K. Vickery. Acoustic positioning systems. New concepts-the future. In *Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles., Fort Lauderdale, FL, USA.*, pages 103–110, August 1998.

[182] S. Vike and J. Jouffroy. Diffusion-based outlier rejection for underwater navigation. In *Proc. IEEE OCEANS'05 , Washington, D.C., 2005*, 2005.

[183] M. Žefran, V. Kumar, and C.B. Croke. On the Generation of Smooth Three-Dimensional Rigid Body Motions. *IEEE Trans. Robotics and Automation, 14(4):576–589*, August 1998.

[184] G. Wahba. A Least-Squares Estimate of Spacecraft Attitude. *SIAM Review, Vol. 7, No.3*, page 409, July 1965.

[185] M.M. Walker, T.E. Dennis, and J.L. Kirschvink. The magnetic sense and its use in long-distance navigation by animals. *Current Opinion in Neurobiology 2002, 12:735–744*, 12:735–744, 2002.

[186] B.H. Wellenhof, H. Lichtenegger, and J. Collins. *GPS: Theory and Practice.* Springer-Verlag, Wien New York, 3rd edition, 1994.

[187] L.L. Whitcomb, D.R. Yoerger, and H. Singh. Combined Doppler/LBL Based Navigation of Underwater Vehicles. In *Proc. 11th UUST*, Durham, New Hampshire, USA, August 1999.

[188] S. Willcox, D. Goldberg, J. Vaganay, and J. Curcio. Multi-Vehicle cooperative navigation and autonomy with the Bluefin Cadre system. In *In Proc. of MCMC'2006 - 7th Conference on Manoeuvring and Control of Marine Craft, Lisbon, Portugal*, 2006.

[189] R. Williams. Design and experimental evaluation of an autonomous surface craft to support auv operations. Master's thesis, Massachusetts Institute of Technology, February 2007.

[190] J. Xavier and V. Barroso. Intrinsic Variance Lower Bound (IVLB): an extension of the Cramér-Rao bound to Riemannian manifolds. In *IEEE Int. Conf. on Acoust., Sp. and Sig. Proc. (ICASSP)*, March 2005.

[191] C. Xu and J.L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Trans. Image Processing*, 7(3):359–369, March 1998.

[192] D. Yoerger and D. Mindell. Precise navigation and control of an rov at 2200 meters depth. In *Proceedings of Intervention/ROV 92, San Diego, USA.*, June 1992.

[193] James W. Youngberg. Method for extending GPS to Underwater Applications. US Patent 5,119,341, June 2 1992, 1992.

[194] P.H. Zipfel. *Modeling and Simulation of Aerospace Vehicle Dynamics*. AIAA Education Series, Reston, VA, 2nd edition, 2007.