Position estimation based on single pseudo-range measurements with unknown clock offset

Sara Alexandra Silva Barroso

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor: Prof. Pedro Tiago Martins Batista

Examination Committee

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

January 2021

Declaration

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

Acknowledgments

This work was supported by the Fundação para a Ciência e a Tecnologia (FCT) through FCT project DECENTER [LISBOA-01-0145-FEDER- 029605], funded by the Lisboa 2020 and PIDDAC programs.

I would like to start by thanking my supervisor Prof. Pedro Batista for his dedication, support, and useful insights. This work would also not have been possible without the love and support of my family and friends, to whom I would like to express my deepest gratitude.

To my three Inêses: Andrade, Nogueira, and Jesus, it has been a privilege to grow up alongside you girls. To Ashleigh Messenger for being by my side throughout a crazy Padova experience. Thank you for the bike rides, the football matches, and Corrie afternoons. To Paulo Branco for the support through some hard times. To my fellow Aalto Middle Schoolers: Jorge, Xavi, Federico, Bruno, Hugo, Jorge Jorge, Sandra, for being the most amazing Erasmus family I could have dreamed of. Rakastan teitä niin paljon! To everyone in “AA”, for years of class notes, coffee breaks, dinners, barbecues and amazing friendship. I would not have made it without you!

Special recognition needs to be given to Sebastião Quintas, Ivan Andrushka, and Madalena Moreira for an initial push and inspiration throughout this time. To Bruno Santos and Ana Rita Cóias for the support and company during this final stretch. To Sara Bernardes, without whose support this thesis would quite literally not have been possible. You rock, dudette! I would not be able to find the correct words of love and appreciation for Sofia Bebiano, so I will just say thank you. For everything. To Joana Baleiras, for being the kindest, most amazing human being. Long live Algebra! To Rita Machado. To say that I would not be who I am without you, is a gross understatement. You have held my hand through the best and the worst of times and have believed in me more than I will ever believe in myself. To make it short(ish), thank you for the balance.

To my big, fun, incredible family, for showing me the most important thing in life. For all their love and support over the past few years, and for some of the best childhood memories, Ana Paula Pereira, João Paulo, Beatriz, Vasco, and Bárbara Costeira, thank you! To Patrícia and Daniel Freire for being the best “siblings”, for all the study sessions, international calls, shopping trips, and late night talks in Justino. To Ana Luísa Paiva for trusting me with your children and putting up with our antics.

To Vanessa Barroso and Claire Tripp for always being my safe spaces, travel buddies and sounding boards. To Inês Barroso and Alan Schafer for years of fun, adventure and deep talks. To Maria José and Augusto Barroso, for quite literally showing me the world through your travels and for being my role models. I am the proudest granddaughter! Thank you for leading by example. To my parents, João Paulo Silva and Ana Cristina Barroso, for showing me love and family, and also books, science, art, music, sports, and have I mentioned books? I will never be able to fully express my gratitude for all the amazing experiences you have made available to me and the life I have been able to live, but I hope this is a good start. Last, and most definitely not least, to António Artur Alves da Silva and Clarisse Ferreira da Silva. I know you would have loved this. All six of us have now made it through, not least thanks to you. Your life story will continue to be a source of daily strength and inspiration. I love you all more than words could ever express. Thank you.

Abstract

The modern world is dependent on many applications that require the use of autonomous underwater vehicles (AUV). Thus, research into the improvement in performance and cost constraints for positioning systems related to these agents is constantly necessary. This thesis addresses the problem of position estimation based on single pseudo-range measurements. An acoustic positioning system for an AUV is considered. A one-way travel-time (OWTT) setting with an offset between the emitting and receiving clocks is assumed. Two cases are considered: one with known vehicle velocity and one where the velocity is only given in relation to the velocity of the surrounding fluid. Continuous-discrete filters are designed to obtain the desired estimates. In the first case, position and bias estimates are obtained with three different solutions: the extended Kalman filter, the unscented Kalman filter, and a linear Kalman filter with an augmented state. The performances are assessed via simulation results and complemented by a thorough analysis with Monte Carlo simulations, which provide a comparison between the root mean square error (RMSE) of each of the three filters, and these with the Bayesian Crámer-Rao bound (BCRB). In the second case, position, velocity, and bias estimates are obtained. The chosen approach is to secure preliminary results through the use of the extended Kalman filter, for which the performance is assessed via simulation results. The solution analysis is complemented with Monte Carlo simulations, which are carried out in order to compare the RMSE performance with the BCRB.

Keywords

underwater navigation; autonomous underwater vehicles; pseudo-range measurement; Kalman filter; augmented system

Resumo

O mundo moderno depende de várias aplicações que utilizam veículos autónomos. Um constante desafio é a melhoria do desempenho e dos custos dos seus sistemas de localização. Esta dissertação aborda o problema de estimação de posição através de uma medida de pseudo-distância. É considerado um sistema de localização acústica para um veículo autónomo subaquático. Assume-se *one-way travel-time* (OWTT), existindo um desvio entre os relógios do emissor e do recetor. São considerados dois casos: um em que se assume ter acesso à velocidade do veículo e um em que esta só é dada em relação à velocidade do fluído. São projetados filtros contínuo-discreto para cada caso, de modo a obter as estimativas pretendidas. No primeiro caso, estimam-se a posição do veículo e o desvio dos relógios através de três soluções: um extended Kalman filter, um unscented Kalman filter e um filtro de Kalman linear com um estado aumentado. A análise das soluções obtidas é complementada pelas simulações de Monte Carlo, que se realizam de modo a comparar os resultados da raiz do erro quadrático médio (RMSE) de cada filtro, entre si e com o Bayesian Crámer-Rao bound (BCRB). No segundo caso, obtêm-se estimativas para a posição e velocidade do veículo, bem como para o desvio dos relógios. A abordagem escolhida passa pela obtenção de resultados preliminares usando um extended Kalman filter. O seu desempenho é analisado através de ensaios simulados e complementado pelas simulações de Monte Carlo, que se realizam de modo a comparar o RMSE do filtro com o BCRB.

Palavras Chave

localização subaquática; veículos autónomos subaquáticos; medida de pseudo-range; filtro de Kalman; sistema aumentado

Contents

List of Figures

List of Tables

Acronyms

AUVs	autonomous underwater vehicles
GPS	global positioning system
LBL	long baseline
SBL	short baseline
USBL	ultra-short baseline
OWTT	one-way travel-time
TWTT	two-way travel-time
AHRS	attitude and heading reference system
DVL	Doppler velocity log
KF	Kalman filter
LKF	linear Kalman filter
EKF	extended Kalman filter
UKF	unscented Kalman filter
UT	unscented transform
GES	globally exponentially stable
BCRB	Bayesian Crámer-Rao bound
RMSE	root mean square error

1

Introduction

1.1 Motivation

The use of autonomous underwater vehicles (AUVs) has been increasing in recent years, with major applications in the industrial, military [1], and research fields. Some notable missions are the mapping of the seafloor [2], which can be used for oil and gas infrastructure installation; inspection of underwater man-made structures to prevent leakages and other potential environmental impacts; surveillance and information operations connected to military activities; fishing and fish farming; wreckage search for missing aircrafts [3] or ships; research applications, from the study of the ecosystems through the measurement of pH levels, element concentrations and presence of microscopic life in lakes and oceans, to the study of the evolution and predicted progressions of these bodies of water.

The question of localization of such vehicles is relevant whether for geo-referencing or control purposes. Traditional localization systems, such as the global positioning system (GPS), utilize electromagnetic transponders to provide range measurements from the agent to several sources, which allow for the calculation of an accurate position via trilateration (see Section 2.1.2). Such solutions are not available in an underwater scenario due to the strong attenuation that the electromagnetic waves suffer in water. To overcome this problem, it is possible to use, instead, acoustic transponders. The three major categories of underwater acoustic positioning systems are the long baseline (LBL) [4], the short baseline (SBL) [5], and the ultra-short baseline (USBL) [6], which are differentiated based on the distance separating the active sensing elements. In the case of the USBL, the cost constraints are smaller, but it presents a lower accuracy. The LBL and SBL require a higher investment in equipment, since they involve the use of more acoustic transponders, as well as clock-synchronization software. A more detailed description of these systems is provided in the following chapter.

The study of alternative localization methods, which can provide high accuracy in performance, while also achieving a lower cost of production is, hence, imperative for the continued development of these types of missions. As a result, this work addresses the problem of position estimation based on single pseudo-range measurements.

1.2 Objectives

The goal of this thesis is to provide some insight on the problem of navigation in which a single range measurement is available to the agent. These distance measurements are assumed to be corrupted by an unknown constant bias, which is explicitly taken into consideration in the design of the solutions. The problem under consideration is, then, to design a continuous-discrete filter to estimate the position of the agent, as well as the constant bias term. Two cases are considered: one with bottom-lock, i.e., with known velocity, and one without bottom-lock, i.e., with unknown velocity (see Section 2.1.3 for full definitions). The first one provides no further estimates whereas the second one gives the additional estimate of the speed of the surrounding fluid.

1.3 Contributions

The main goals of this work are:

- the development of a function to simulate an extended Kalman filter (EKF), one to simulate an unscented Kalman filter (UKF), and one to simulate a linear Kalman filter (LKF) for an augmented system, all related to the problem with bottom-lock;
- the presentation and discussion of the simulation results obtained for each function, to provide an assessment on the performance of each one, as well as a comparison between them;
- the further comparison and thorough analysis on performance of these filters via Monte Carlo simulation results;
- the performance analysis of preliminary results for the problem without bottom-lock via the simulation of an EKF, complemented by Monte Carlo simulations.

1.4 Thesis outline

Chapter 2 provides some necessary background knowledge, as well as an analysis of the state of the art technologies that address the problem of localization of AUVs. In Chapter 3, the notation used throughout the work is established. An introduction to the two problems to be addressed is provided and the two different systems are defined. In Chapter 4, a brief explanation of the functioning of the chosen filters is provided. In Chapter 5, the suggested solutions for the system with bottom-lock are formulated and initial numerical results are presented. Chapter 6 presents the suggested solution for the system without bottom-lock and shows the initial numerical results. Chapter 7 offers a brief explanation of the Bayesian Crámer-Rao bound (BCRB). In Chapter 8, the Monte Carlo simulations for the system with bottom-lock are analysed, whereas Chapter 9 provides that analysis regarding the system without bottom-lock. Chapter 10 gives a breakdown of the work accomplished and offers suggestions for future work.

2

Background and state of the art

2.1 Background

The three major categories of underwater acoustic positioning systems are the LBL, the SBL, and the USBL, for which a representation is presented in Figure 2.1. Some important definitions for understanding the workings of these methods are analysed in the following sections.

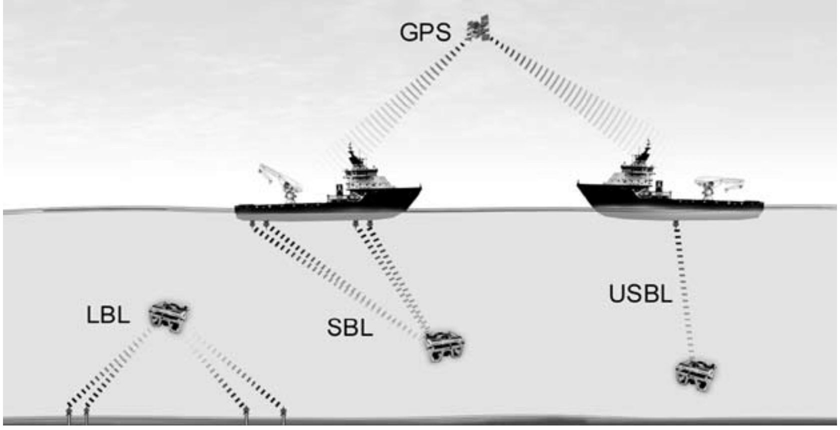

Figure 2.1: Typical configurations for AUV localization using LBL, SBL, and USBL.
Source: [7].

2.1.1 Range and bearing measurements

Range and bearing measurements are two of the foundational concepts of acoustic positioning systems. The first is defined as the distance between a source position and the agent, typically obtained via trilateration (see Section 2.1.2 for more details). The second refers to the angle difference between the object and some direction. The direction of the force of gravity is taken as a reference vector and two types of bearings are possible: absolute and relative. The former uses a fixed axis, either the Earth's magnetic North (magnetic bearing) or the true North (true bearing). The latter can also be useful and relates the direction of forward movement of the agent with that of a known object. Bearing measurements are obtained by phase-differencing, i.e., comparing the phase of the emitted acoustic beam and the received signal.

These positioning systems can work as one-way travel-time (OWTT), where the range measurement is obtained from the difference between the time at which an acoustic signal is sent from the source and the time it is received at the agent; or as two-way travel-time (TWTT), where the range measurement is calculated using half the time between the moment at which the initial signal is sent from the source and the moment at which the agent's responding signal is received back at the source. In both cases, the distances are calculated from the corresponding time-of-flight and the sound speed profile, which relates the speed of sound propagation in water to its temperature, depth, and salinity. For OWTT, it is clear to see the need for clock-synchronization between the source and the agent's transponders, since the time checks are performed at both ends.

2.1.2 Trilateration

To analyse the concept of trilateration, it is useful to start with a 2-D example, for which a representation is presented in Figure 2.2. By knowing the range between a signal source and the desired target, one has information about the distance, but not about the direction at which the object is positioned, with relation to the beacon. This way, one can only establish that the object is to be positioned at some point within the circumference around the source with radius equal to that same range. Once two circumferences are established, and assuming they are not tangent to each other and indeed intersect, they will do so at two points. Trilateration, then, implies the use of 3 beacons, giving 3 different circumferences, the intersection of which will be the true position of the target.

For the 3-D case, the circumferences are replaced by the surfaces of spheres, and so, to get only one point that gives the desired location, 4 beacons are used. Note that, in both cases, the position could have to be obtained as the most likely point from a set of possibilities, since the circumferences or surface spheres could intersect in an area or surface area, respectively, rather than a single point.

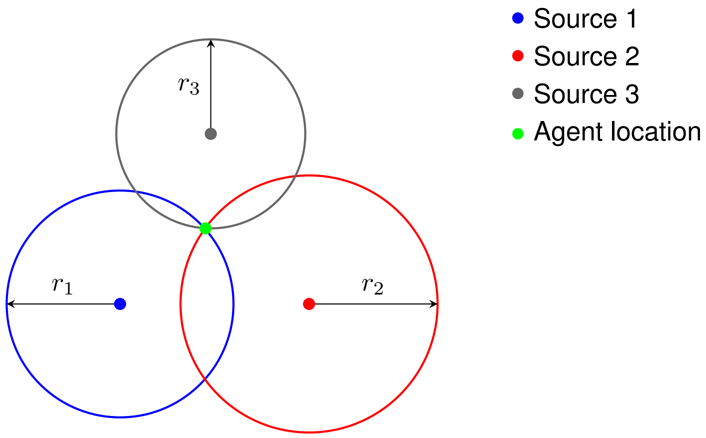

Figure 2.2: Localization by trilateration.
Adapted from: [8].

2.1.3 On-board systems

AUVs can integrate a myriad of on-board systems and sensors, depending on the mission needs. The main ones considered in this work are the attitude and heading reference system (AHRS) and the Doppler velocity log (DVL). The AHRS provides 3-D orientation by integrating data from gyroscopes, accelerometers and magnetometers. Because of this sensor fusion, the potential drift from the gyroscopes is compensated by reference vectors, namely gravity and the Earth's magnetic field, which yields a more accurate attitude information, including the roll, pitch, and yaw angles [9].

In traditional LBL navigation, the DVL provides a velocity vector for the vehicle by sending pulses along a minimum of 3 acoustic beams, each pointing in a different direction. If the DVL is sufficiently close to the ocean floor, these beams will be reflected and can then be received back by the transceiver. This necessary condition is known as bottom-lock. The difference in frequency between the emitted and the received waves is a result of the Doppler shift and can be used to calculate the desired inertial velocity values of the vehicle. If the condition is not verified, only a relative velocity can be provided, which corresponds to the velocity of the vehicle relative to the velocity of its surrounding fluid.

2.2 State of the art

The naming of the three broad classes of localization methods for AUVs is related to the distance between the necessary transponders. Indeed, in LBL, the beacons are mounted on the seafloor at significantly distanced positions, which can present some disadvantages, but produces the most accurate results, while also being water-depth independent. This type of system is most commonly used for underwater precision missions, due to the need for reliable results, but can be very costly, as a result of the need for the 4 fixed beacons, as well as clock-synchronization in OWTT.

To improve the accuracy of the LBL, a DVL system can be integrated, as mentioned in Section 2.1.3. An initial implementation of a combined Doppler/LBL based localization system is shown in [10]. An adaptation of the LBL system into a system that uses buoys with GPS is analysed, for instance, in [11].

In SBL, a similar approach is taken, with the main difference being the positioning of the beacons. In this case, the transponders are usually mounted on the bottom of a ship or dock and are separated by a smaller distance. This method can achieve accuracy levels that resemble those of the LBL when a larger baseline is provided, i.e., when the size constraints allow for larger spacing between transponders. This type of system presents similar cost constraint problems. Experimental results of a SBL system are presented in [5].

The USBL uses a different method and does not rely on trilateration to obtain the vehicle position. This system uses only one transceiver, which provides only one range measurement, but also includes an array of transducers separated by $10\,\mathrm{cm}$, which provide bearing measurements. With one distance measurement and by knowing the relative angles of the forward movement of the vehicle with relation to the source, one can obtain the location of the agent. Extensive research on the USBL has been performed, namely in [6]. One example of the design and experimental validation of this type of system can be seen in [12]. Several comparisons of the three methods have been performed, with [13] being an early approach that presented a very thorough list of advantages and disadvantages for each system.

In [14], the concept of a synthetic LBL (SLBL) was brought up. In this method, a single acoustic source is used, with range measurements being obtained with a specified sampling interval. Between sampling instants, a high performance dead-reckoning system is used to compute the motion of the vehicle. A discrete-time Kalman filter (KF) is applied to the linearized model of the system, essentially applying standard trilateration techniques. This system obtains a good performance, since it is virtually water-depth independent.

A similar idea is used in [15] with the definition of the virtual LBL (VLBL). In this case, a validation of the range measurements is performed, meaning that the time steps between the four consecutive "good ranges" recorded are not necessarily equal. The method uses dead-reckoning to virtually move the source in the same direction and distance that the vehicle travelled during the time delay between two measurements, providing virtual source locations.

The EKF is a very widespread method utilized for solution estimation in navigation problems, namely in [16, 17]. An analysis of the behaviour of the estimation error of a discrete-time EKF, applied to a stochastic framework, is presented in [18]. Conclusions are drawn about the error boundedness under specified conditions. These include the need for sufficiently small initial error and sufficiently small noise, which are heavy constraints for navigation systems. For this reason, other approaches which provide global asymptotic stability are important in order to guarantee a more robust solution.

Some notable research on single range measurement positioning systems has been performed, namely in [19], where preliminary experimental results with single beacon acoustic navigation are presented. In [20], the problems of single range navigation and source localization are addressed. A solution based on an augmented state transforms the original nonlinear system into a linear time-varying (LTV) system, which allows for the use of LTV system characteristics and enables the design of a LKF giving globally exponentially stable (GES) error dynamics. In [21], the observability of single range navigation is addressed, as well as some robustness issues. A robust predictor–corrector state estimator is proposed, to cope with possible outliers in the range measurements.

Another heavy constraint on the problem of underwater navigation is the need for clock synchronization in a OWTT setting. While this condition can be achieved prior to each mission via the appropriate calibrations, this is not only an added burden, but is also proven to be insufficient for long missions, since clock drift is inevitable unless clock synchronization is performed often. For this reason, solutions that can explicitly account for the bias term added by the clock offset, as in [22], are of great interest.

In [23], the author addresses the desire for lower operational burden and hardware requirements in

LBL navigation, as well as the clock offset concern. These were previously mentioned in [20] and [22], respectively, and their combination yields estimations based on single pseudo-ranges. The solution presented includes deriving an augmented state, for which the system dynamics become linear, so as to allow for the use of the LKF. This expands the previously obtained advantages, by solving both issues, while also obtaining GES guarantees.

One of the main contributions of this thesis is to expand upon the work in [23], by providing an extensive comparison between the performance of the proposed solution and other existing estimators, namely the EKF and the UKF. This is done through simulation results, along with Monte Carlo runs, including a comparison with the BCRB.

Finally, one can note that the use of a sound speed profile for distance computation, previously mentioned in Section 2.1.1, can pose some problems for navigation. Because the propagation speed of an acoustic beam greatly depends on water temperature, depth, and salinity, typically the speed of sound needs to be profiled for the operation area prior to deployment. This brings about an added step, which increases the necessary hardware, as well as the overall difficulty of field experiments. It can also bring errors, if the water characteristics are altered between the time of profiling and the time of deployment, namely by ocean currents. This problem was addressed in [24], where the propagation speed was explicitly estimated, along with the clock offset, in a LBL setting.

3

Problem statement and notation

3.1 Notation

Throughout this work, $\mathbf{A}$ denotes an $n \times m$ matrix, where $n > 1$ and $m > 1$, $\mathbf{a}$ denotes an $n \times 1$ vector, and a denotes a scalar. The identity matrix, of proper dimensions, is denoted by $\mathbf{I}$, and $\mathbf{I}_n$ denotes the $n \times n$ identity matrix. When matrix dimensions are omitted, the matrices are assumed to be of proper dimensions. A block diagonal matrix is represented by $\mathbf{diag}(\mathbf{A}_1, ..., \mathbf{A}_n)$. For $\mathbf{x} \in \mathbb{R}^3$, x_x, x_y, and x_z represent the x, y, and z axis components of $\mathbf{x}$, respectively. The transpose operator is denoted by $(.)^T$. A vector that follows a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{A}$ is denoted by $\mathbf{a} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{A})$. The Special Orthogonal Group is denoted by $\mathrm{SO}(3) = \{\mathbf{X} \in \mathbb{R}^{3\times3} : \mathbf{X}\mathbf{X}^T = \mathbf{X}^T\mathbf{X} = \mathbf{I}_3 \wedge \det(\mathbf{X}) = 1\}$. $\boldsymbol{E}\{\mathbf{x}\}$ represents the expected value of vector $\mathbf{x}$.

3.2 System dynamics

This thesis provides an extension of the work presented in [22], where multiple pseudo-range measurements are available. The present work considers an acoustic positioning system for an underwater vehicle, consisting of a fixed emitting source. The agent is equipped with an acoustic receiver and is assumed to have access to the inertial position of the emitter, which is constant. A OWTT setting (see Section 2.1.1) is considered, with no guarantee of clock synchronization between source and agent. This time offset results in a bias term in the range measurements, which are, then, pseudo-range measurements, obtained periodically. These values are not enough to estimate the position of the agent, as more information about the movement of the vehicle is necessary. For this purpose, it is assumed that the vehicle is equipped with an AHRS and a DVL (see Section 2.1.3 for more information about these systems).

In order to adequately describe the system dynamics, two reference frames need to be introduced: the local inertial coordinate reference frame, denoted by $\{I\}$, and a coordinate frame attached to the vehicle, usually referred to as the body-fixed reference frame, denoted by $\{B\}$. Considering $\mathbf{p}(t) \in \mathbb{R}^3$ as the inertial position of the vehicle, $\mathbf{v}(t) \in \mathbb{R}^3$ as the velocity of the vehicle relative to $\{I\}$, expressed in body-fixed coordinates, and $\mathbf{R}(t) \in \mathrm{SO}(3)$ as the rotation matrix from $\{B\}$ to $\{I\}$, given by the AHRS, the linear motion of the vehicle is, then, described by

$$\dot{\mathbf{p}}(t) = \mathbf{R}(t)\mathbf{v}(t). \tag{3.1}$$

In this work, two scenarios are considered: one in which it is assumed that the bottom-lock condition is verified, and so, the inertial velocity values are directly available to the vehicle; and one which corresponds to the absence of bottom-lock, where only the velocity of the vehicle relative to the fluid is available. The inertial position of the source, which is assumed to be fixed, is denoted by $\mathbf{s} \in \mathbb{R}^3$. Without the offset between the emitting and receiving clocks, the range measurements would be given by $r(k) = \|\mathbf{s} - \mathbf{p}(t_k)\|$. Let $b_c(t_k)$ be the bias term that accounts for the effect of the unknown clock offset. The pseudo-range measurements available to the vehicle are then given by

$$r(k) = \|\mathbf{s} - \mathbf{p}(t_k)\| + b_c(t_k), \tag{3.2}$$

with $t_k = t_0 + kT,\ k \in \mathbb{N}$, where $T > 0$ is the sampling period and t_0 is the initial time. From the conditions described, it is possible for the vehicle to receive a signal in which the time tag is greater than its own clock. For these situations, the receiver would know that its clock is behind and could adjust its value to a higher one, making it possible to consider the following assumption.

Assumption 1. All the pseudo-range measurements are positive, i.e., $r(k) > 0$ for all k.

A consideration about the factor that accounts for the offset of the clocks is also used throughout this work.

Assumption 2. The offset of the clocks is constant, i.e., $\dot{b}_c(t) = 0$.

Notice that, because the filters are designed to be noise controllable (see Section 5.2.1 for full discussion), this restriction can be loosened. Indeed, even though the bias term is assumed constant, in nominal terms, it is possible to track slow time-varying quantities.

3.2.1 System dynamics with bottom-lock

Considering the scenario in which bottom-lock can be guaranteed, the vehicle has direct access to the inertial velocity values. In this case, the system can be described by a continuous nonlinear system, because the movement of the agent is a physical phenomenon, i.e., it occurs in real-time, with discrete-time output, since the pseudo-range measurements are obtained from a sensor, which implies a sampling time. This duality of continuous-time movement and discrete-time measurements results in continuous-discrete system dynamics, which are obtained by combining the previous assumptions with (3.1) and (3.2), giving

$$\begin{cases} \dot{\mathbf{p}}(t) = \mathbf{R}(t)\mathbf{v}(t) \\ \dot{b}_c(t) = 0 \\ r(k) = \|\mathbf{s} - \mathbf{p}(t_k)\| + b_c(t_k) \end{cases}. \tag{3.3}$$

The first problem considered in this work is the design of an estimation solution for (3.3).

3.2.2 System dynamics without bottom-lock

In the absence of bottom-lock, the DVL provides only the velocity of the vehicle relative to the velocity of its surrounding fluid. In (3.1), $\mathbf{v}(t)$ is, thus, the sum of its two components. Let $\mathbf{v}_f(t) \in \mathbb{R}^3$ denote the velocity of the fluid in inertial coordinates and $\mathbf{v}_r(t) \in \mathbb{R}^3$ denote the velocity of the vehicle relative to the fluid in body-fixed coordinates. Hence,

$$\mathbf{v}(t) = \mathbf{R}^T(t)\mathbf{v}_f(t) + \mathbf{v}_r(t). \tag{3.4}$$

In order to conclude the analysis of the system dynamics, one further assumption must be considered.

Assumption 3. The inertial fluid velocity is constant, i.e., $\dot{\mathbf{v}}_f(t) = \mathbf{0}_{3\times1}$.

Once again, using the argument of noise controllability, this restriction can be loosened and it is possible to track slow time-varying quantities. With all three assumptions and combining (3.1), (3.2), and (3.4) gives the continuous nonlinear system with discrete-time output

$$\begin{cases} \dot{\mathbf{p}}(t) = \mathbf{v}_f(t) + \mathbf{R}(t)\mathbf{v}_r(t) \\ \dot{\mathbf{v}}_f(t) = 0 \\ \dot{b}_c(t) = 0 \\ r(k) = \|\mathbf{s} - \mathbf{p}(t_k)\| + b_c(t_k) \end{cases} . \tag{3.5}$$

The second problem considered in this work is the design of an estimation solution for (3.5).

3.3 Discretization

The auxiliary systems needed for vehicle positioning, including the DVL and AHRS, as well as the measurements, are discrete-time systems, with associated sampling rates. The nonlinear system dynamics are considered in a continuous-discrete framework, so that pseudo-range measurements obtained at low update rates are used to drive the estimation error to zero, while the other sensors, which operate at a higher rate, are used to drive the system dynamics. It is, therefore, necessary to rewrite them considering discrete-time measurements, with sampling period T. The discretization of the two systems tackled in this work are presented in the following sections.

3.3.1 Discretization for system with bottom-lock

By discretizing system (3.3), the discrete-time system dynamics can be described by

$$\begin{cases} \mathbf{p}(t_{k+1}) = \mathbf{p}(t_k) + \int_{t_k}^{t_{k+1}} \mathbf{R}(\tau)\mathbf{v}(\tau)\, d\tau \\ b_c(t_{k+1}) = b_c(t_k) \\ r(k) = \|\mathbf{s} - \mathbf{p}(t_k)\| + b_c(t_k) \end{cases} . \tag{3.6}$$

Using the system described in (3.6) and defining the discrete-time states

$$\begin{cases}\mathbf{x}_1(k) = \mathbf{p}(t_k) \\ x_2(k) = b_c(t_k)\end{cases},$$

one can write

$$\begin{cases}\mathbf{x}_1(k+1) = \mathbf{x}_1(k) + \mathbf{u}(k) \\ x_2(k+1) = x_2(k)\end{cases},$$

with $\mathbf{u}(k) = \int_{t_k}^{t_{k+1}} \mathbf{R}(\tau)\mathbf{v}(\tau)\, d\tau$. From here, it is possible to rewrite the system dynamics as

$$\begin{cases}\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ r(k) = \|\mathbf{s} - \mathbf{x}_1(k)\| + x_2(k)\end{cases}, \tag{3.7}$$

where $r(k)$ are the pseudo-range measurements available to the vehicle. The system and input matrices, respectively $\mathbf{A}$ and $\mathbf{B}$, are constant over time, as given by

$$\mathbf{A} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{bmatrix}. \tag{3.8}$$

3.3.2 Discretization for system without bottom-lock

By discretizing system (3.5), the discrete-time system dynamics can be described by

$$\begin{cases}\mathbf{p}(t_{k+1}) = \mathbf{p}(t_k) + T\mathbf{v}_f(t_k) + \int_{t_k}^{t_{k+1}} \boldsymbol{R}(\tau)\mathbf{v}_r(\tau)\, d\tau \\ \mathbf{v}_f(t_{k+1}) = \mathbf{v}_f(t_k) \\ b_c(t_{k+1}) = b_c(t_k) \\ r(k) = \|\mathbf{s} - \mathbf{p}(t_k)\| + b_c(t_k)\end{cases}. \tag{3.9}$$

Using the system described in (3.9) and defining the discrete-time states

$$\begin{cases}\mathbf{x}_1(k) = \mathbf{p}(t_k) \\ \mathbf{x}_2(k) = \mathbf{v}_f(t_k) \\ x_3(k) = b_c(t_k)\end{cases},$$

one can write

$$\begin{cases}\mathbf{x}_1(k+1) = \mathbf{x}_1(k) + T\mathbf{x}_2(k) + \mathbf{u}(k) \\ \mathbf{x}_2(k+1) = \mathbf{x}_2(k) \\ x_3(k+1) = x_3(k)\end{cases},$$

with $\mathbf{u}(k) = \int_{t_k}^{t_{k+1}} \mathbf{R}(\tau)\mathbf{v}_r(\tau)\, d\tau$.

From here, it is possible to rewrite the system dynamics as

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ r(k) = \|\mathbf{s} - \mathbf{x}_1(k)\| + x_3(k) \end{cases}, \tag{3.10}$$

where $r(k)$ are the pseudo-range measurements available to the vehicle. The system and input matrices, respectively $\mathbf{A}$ and $\mathbf{B}$, are constant over time, as given by

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{3.11}$$

4

Kalman filter

4.1 Discrete-time linear Kalman filter

The LKF [25, 26] is a recursive process which aims to estimate the state at the next step based on the current state and the measurement at the next step. This is done in two segments: a prediction step and an update step. To describe the filtering methodology, consider the general discrete-time linear time-varying system

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{w}(k) \\ \mathbf{y}(k) = \mathbf{C}(k)\mathbf{x}(k) + \mathbf{n}(k) \end{cases},$$

where $\mathbf{A}(k)$, $\mathbf{B}(k)$, $\mathbf{C}(k)$ are time-varying matrices, $\mathbf{w}(k)$ is the process noise, which follows the Gaussian distribution $\mathbf{w}(k) \sim \mathcal{N}(\mathbf{0},\, \mathbf{Q}(k))$, and $\mathbf{n}(k)$ is the observation noise, which follows the Gaussian distribution $\mathbf{n}(k) \sim \mathcal{N}(\mathbf{0},\, \mathbf{R}(k))$. The noises are considered to be mutually independent.

The following notation is used throughout the next sections: $k|k$ represents the value of a quantity at step k given the first k observations; $k+1|k$ represents the value of the variable predicted for time-step $k+1$ given the first k observations.

4.1.1 Prediction step

The filter starts by generating an *a priori* estimate for the state at the next step based on the current state estimate and input, as well as the state and input matrices, and an *a priori* estimate for the covariance matrix at the next step, as given by

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}(k)\hat{\mathbf{x}}(k|k) + \mathbf{B}(k)\mathbf{u}(k)$$

for the predicted state estimate, and

$$\mathbf{P}(k+1|k) = \mathbf{A}(k)\mathbf{P}(k|k)\mathbf{A}^T(k) + \mathbf{Q}(k)$$

for the predicted estimation error covariance, both obtained by using the results of the previous filter iteration.

4.1.2 Update step

In this step, the resulting *a priori* measurement estimate is compared to the actual system measurement, the optimal Kalman gains are calculated, and the *a posteriori* state estimate is computed. The associated covariance matrix is also compared and updated. The associated equations are

$$\hat{\mathbf{y}}(k+1|k) = \mathbf{C}(k+1)\hat{\mathbf{x}}(k+1|k)$$

for the predicted measurement,

$$\mathbf{res}(k+1) = \mathbf{y}(k+1) - \hat{\mathbf{y}}(k+1|k)$$

for the measurement residual,

$$\mathbf{S}(k+1) = \mathbf{C}(k+1)\mathbf{P}(k+1|k)\mathbf{C}^T(k+1) + \mathbf{R}(k+1)$$

for the residual covariance,

$$\mathbf{W}(k+1) = \mathbf{P}(k+1|k)\mathbf{C}^T(k+1)\mathbf{S}^{-1}(k+1) \tag{4.1}$$

for the optimal Kalman gain,

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{W}(k+1)\mathbf{res}(k+1)$$

for the updated state estimate,

$$\mathbf{P}(k+1|k+1) = (\mathbf{I} - \mathbf{W}(k+1)\mathbf{C}(k+1))\mathbf{P}(k+1|k)$$

for the updated estimation error covariance, and

$$\mathbf{res}(k+1|k+1) = \mathbf{y}(k+1) - \mathbf{C}(k+1)\hat{\mathbf{x}}(k+1|k+1)$$

for the measurement post-fit residual.

4.1.3 Limitations

Although the LKF works well in numerous situations, it has some constraints that limit the scenarios in which it can be used. One major limitation is the fact that the LKF, as described above, can only be applied to linear systems.

The problem addressed in this work is not a linear one, since the desired measurement is the distance between the source and the agent, which, in $\mathbb{R}^3$, implies the use of a norm. For this case, the LKF cannot be directly applied. To mitigate this type of problems, some other tools can be used, namely the EKF and the UKF, for which brief descriptions are presented in the following sections.

4.2 Discrete-time extended Kalman filter

The EKF [27,28] aims to generalize the results from the LKF, providing a possible solution for nonlinear systems. This variation uses mathematical techniques to linearize the model about a working point. It is applied to state-space models of the form

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{f}(k, \mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k)) \\ \mathbf{y}(k) = \mathbf{h}(k, \mathbf{x}(k), \mathbf{n}(k)) \end{cases},$$

where $\mathbf{x}(k)$ is the state vector, $\mathbf{u}(k)$ is a known input, $\mathbf{w}(k)$ is the process noise with covariance matrix $\mathbf{Q}(k)$, and $\mathbf{n}(k)$ is the observation noise with covariance matrix $\mathbf{R}(k)$. These noises are assumed to be zero-mean and mutually independent. The functions $\mathbf{f}$ and $\mathbf{h}$ can be time-varying and nonlinear, however, they must be differentiable, for a linearization about an estimate of the current mean and covariance to be possible. The problem is considered nonlinear if at least one of the functions is nonlinear.

The inner workings of the EKF are similar to the linear Kalman filter, except, instead of directly using the state, input, and output matrices, one has to use the Jacobians of functions $\mathbf{f}$ and $\mathbf{h}$, as defined by

$$\mathbf{F}(k) = \frac{\partial \mathbf{f}(k)}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}(k|k)}, \tag{4.2}$$

and

$$\mathbf{H}(k+1) = \frac{\partial \mathbf{h}(k+1)}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}(k+1|k)}, \tag{4.3}$$

respectively. These matrices can be time-varying, in which case their values must be calculated at each filter iteration. Equivalent steps to the ones presented for the LKF are then taken.

4.2.1 Prediction step

Like before, the filter starts by generating the *a priori* estimate for the state and the covariance matrix at the next step, also based on the current state estimate and input, only this time, it uses the (potentially) nonlinear function $\mathbf{f}$ and its Jacobian. The prediction step equations are given by

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{f}(k, \hat{\mathbf{x}}(k|k), \mathbf{u}(k))$$

for the predicted state estimate and

$$\mathbf{P}(k+1|k) = \mathbf{F}(k)\mathbf{P}(k|k)\mathbf{F}^T(k) + \mathbf{Q}(k)$$

for the predicted estimation error covariance.

4.2.2 Update step

With the appropriate adjustments for the nonlinear case, by using the Jacobian matrices, the equations for the update step of the EKF are given by

$$\hat{\mathbf{y}}(k+1|k) = \mathbf{h}(k+1, \hat{\mathbf{x}}(k+1|k))$$

for the predicted measurement,

$$\mathbf{res}(k+1) = \mathbf{y}(k+1) - \hat{\mathbf{y}}(k+1|k)$$

for the measurement residual,

$$\mathbf{S}(k+1) = \mathbf{H}(k+1)\mathbf{P}(k+1|k)\mathbf{H}^T(k+1) + \mathbf{R}(k+1)$$

for the residual covariance,

$$\mathbf{W}(k+1) = \mathbf{P}(k+1|k)\mathbf{H}^T(k+1)\mathbf{S}^{-1}(k+1) \tag{4.4}$$

for the filter gain,

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{W}(k+1)\mathbf{res}(k+1)$$

for the updated state estimate,

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{W}(k+1)\mathbf{S}(k+1)\mathbf{W}^T(k+1)$$

for the updated estimation error covariance, and

$$\mathbf{res}(k+1|k+1) = \mathbf{y}(k+1) - \mathbf{h}(k+1, \hat{\mathbf{x}}(k+1|k+1))$$

for the measurement post-fit residual.

Note that, if the system is linear, then $\mathbf{F}(k) = \mathbf{A}(k)$ and $\mathbf{H}(k+1) = \mathbf{C}(k+1)$, which will make these equations equivalent to the ones stated in the previous section and guarantee an optimal estimator.

4.2.3 Limitations

The EKF can be used in a much wider set of models, however, unlike its linear version, it is not, in general, an optimal estimator, i.e., it does not calculate optimal Kalman gains. In fact, if the initial state estimate is not accurate enough, the filter may diverge due to its inherent linearization.

Another problem with the EKF is that the estimated covariance matrix tends to underestimate the true covariance matrix, which may cause misleading results. For this reason, the covariance matrix obtained from the filter and the true covariance matrix are typically compared when analysing results.

4.3 Discrete-time unscented Kalman filter

The UKF [29] provides an alternative to the EKF and presents a nonlinear approach to Kalman filtering without the need for linearization. This filter aims to provide an estimation solution for systems of the form

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{f}(k, \mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k)) \\ \mathbf{y}(k) = \mathbf{h}(k, \mathbf{x}(k), \mathbf{n}(k)) \end{cases}, \tag{4.5}$$

where $\mathbf{x}(k)$ is the state vector, $\mathbf{u}(k)$ is a known input, $\mathbf{w}(k)$ is the process noise with covariance matrix $\mathbf{Q}(k)$, and $\mathbf{n}(k)$ is the observation noise with covariance matrix $\mathbf{R}(k)$. The noises are assumed to be zero-mean and mutually independent. The functions $\mathbf{f}$ and $\mathbf{h}$ can be time-varying and nonlinear and the problem is considered nonlinear if at least one of the functions is nonlinear.

The inner workings of the UKF are different from before. Indeed, instead of linearizing the system functions, one now uses the unscented transform (UT) to obtain the desired estimates.

4.3.1 Unscented Transform

The UT is a mathematical tool used to estimate the result of applying a given nonlinear transformation to a certain probability distribution, for which only a limited set of points are known. Considering a random variable $\mathbf{x} \in \mathbb{R}^n$, with expected value $\bar{\mathbf{x}}$ and covariance matrix $\mathbf{P}_x \in \mathbb{R}^{n \times n}$, by applying a nonlinear function $\mathbf{g} \colon \mathbb{R}^n \to \mathbb{R}^m$, one can obtain a new random variable $\mathbf{y} \in \mathbb{R}^m$, with expected value $\bar{\mathbf{y}}$ and covariance matrix $\mathbf{P}_y \in \mathbb{R}^{m \times m}$. The UT presents a simple way to estimate the statistical properties of this new output variable.

Firstly, the UT uses the given information about the distribution of $\mathbf{x}$ (mean and covariance), to calculate a set of sigma points which obey those statistical properties. The sigma points are then put through the exact nonlinear function and the resulting points are used to compute the estimated values for the mean and covariance of $\mathbf{y}$. To each sigma point, there are two associated weights, one for the first-order, i.e., the mean, and one for the second-order, i.e., the covariance. These weights allow for the skewing of the resulting distribution more toward any specific point that might be deemed necessary, and are chosen so that the following are true:

$$\begin{cases} \sum_{j=0}^{N} W_j^m = 1 \\ \mathrm{E}[x_i] = \sum_{j=0}^{N} W_j^m \chi_{j,i}, \text{ for all } i = 1, ..., L \end{cases}, \tag{4.6}$$

for the first-order weights, and

$$\begin{cases} \sum_{j=0}^{N} W_j^c = 1 \\ \mathrm{E}[x_i x_l] = \sum_{j=0}^{N} W_j^m \chi_{j,i} \chi_{j,l}, \text{ for all pairs } (i, l) \in \{1, ..., L\}^2 \end{cases}, \tag{4.7}$$

for the second-order weights.

The parameters α, k, β, and $\lambda = \alpha^2(L + k) - L$ are used to tune the distribution of the sigma points. The parameter α determines the spread of the sigma points around the mean of x and is usually set to a small positive value, k is a secondary scaling parameter which is usually set to 0, and λ is called the composite scaling parameter. Finally, β is used to incorporate prior knowledge of the distribution of $\mathbf{x}$ (for Gaussian distributions, $\beta = 2$ is optimal), and L is defined as the dimension of the random variable $\mathbf{x}$, which means $L = n$.

The sigma point calculation suggested in [29] is as follows

$$\begin{cases} \boldsymbol{\chi}_0 = \bar{\mathbf{x}} \\ \boldsymbol{\chi}_i = \bar{\mathbf{x}} + \left(\sqrt{(L+\lambda)\mathbf{P}_x}\right)_i, & i = 1, ..., L \\ \boldsymbol{\chi}_i = \bar{\mathbf{x}} - \left(\sqrt{(L+\lambda)\mathbf{P}_x}\right)_{i-L}, & i = L+1, ..., 2L \end{cases},$$

where $\mathcal{X}_i$ is the i-th sigma point and $\left(\sqrt{(L+\lambda)\mathbf{P}_x}\right)_i$ is the i-th column of matrix $\left(\sqrt{(L+\lambda)\mathbf{P}_x}\right)$.

The corresponding weights are also given in [29] as

$$\begin{cases} W_0^m = \dfrac{\lambda}{L+\lambda} \\ W_0^c = \dfrac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta \\ W_i^m = W_i^c = \dfrac{1}{2(L+\lambda)}, \qquad i = 1, ..., 2L \end{cases},$$

where W_i^m is the weight of the i-th sigma point in relation to the mean estimation and W_i^c is the weight of the i-th sigma point in relation to the covariance estimation.

In order to obtain the desired estimation, the sigma points are propagated through function $\mathbf{g}$ so that new points $\mathcal{Y}_i = \mathbf{g}(\mathcal{X}_i)$ are obtained. These are then used to estimate the mean and covariance of $\mathbf{y}$, as

$$\bar{\mathbf{y}} \approx \sum_{i=0}^{2L} W_i^m \mathcal{Y}_i$$

and

$$\mathbf{P}_y \approx \sum_{i=0}^{2L} W_i^c (\mathcal{Y}_i - \bar{\mathbf{y}})(\mathcal{Y}_i - \bar{\mathbf{y}})^T,$$

respectively.

4.3.2 Initialization

As with other recursive methods, in order to apply the UKF to the system presented in (4.5), one needs to set the initialization parameters. The initial state estimate is defined by $\hat{\mathbf{x}}(0) = \mathrm{E}[\mathbf{x}(0)]$, with covariance matrix $\hat{\mathbf{P}}(0) = \mathrm{E}\left[(\mathbf{x}(0) - \hat{\mathbf{x}}(0))(\mathbf{x}(0) - \hat{\mathbf{x}}(0))^T\right]$.

The UKF accounts for the noise terms $\mathbf{w}(k)$ and $\mathbf{n}(k)$, by defining an augmented state vector given by

$$\mathbf{x}^a(k) = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{w}(k) \\ \mathbf{n}(k) \end{bmatrix},$$

with covariance matrix

$$\mathbf{P}^a(k) = \begin{bmatrix} \mathbf{P}(k) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}(k) \end{bmatrix}.$$

The augmented state estimate is initialized with

$$\hat{\mathbf{x}}^a(0) = \mathrm{E}[\mathbf{x}^a(0)] = \begin{bmatrix} \hat{\mathbf{x}}(0) \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix},$$

and its covariance matrix is initialized with

$$\hat{\mathbf{P}}^a(0) = \mathrm{E}\left[(\mathbf{x}^a(0) - \hat{\mathbf{x}}^a(0))(\mathbf{x}^a(0) - \hat{\mathbf{x}}^a(0))^T\right] = \begin{bmatrix} \hat{\mathbf{P}}(0) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}(0) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}(0) \end{bmatrix}.$$

4.3.3 Prediction step

The UKF starts by calculating the sigma points, however, because of the application to systems of the kind in (4.5), with augmented state vectors, we now have augmented sigma points. The calculations are similar to the ones presented in Section 4.3.1, and one obtains

$$\begin{cases} \boldsymbol{\mathcal{X}}_0^a(k) = \hat{\mathbf{x}}^a(k) \\ \boldsymbol{\mathcal{X}}_i^a(k) = \hat{\mathbf{x}}^a(k) + \left(\sqrt{(L+\lambda)\mathbf{P}^a(k)}\right)_i, \qquad i = 1, ..., L \\ \boldsymbol{\mathcal{X}}_i^a(k) = \hat{\mathbf{x}}^a(k) - \left(\sqrt{(L+\lambda)\mathbf{P}^a(k)}\right)_{i-L}, \quad i = L+1, ..., 2L \end{cases},$$

where the weights are as defined in (4.6) and (4.7), and the general form of a sigma point is given by

$$\boldsymbol{\mathcal{X}}_i^a(k) = \begin{bmatrix} \boldsymbol{\mathcal{X}}_i^x(k) \\ \boldsymbol{\mathcal{X}}_i^w(k) \\ \boldsymbol{\mathcal{X}}_i^n(k) \end{bmatrix}.$$

The prediction step equations are given by

$$\boldsymbol{\mathcal{X}}^x(k+1|k) = \mathbf{f}(k, \boldsymbol{\mathcal{X}}^x(k), \boldsymbol{\mathcal{X}}^n(k), \mathbf{u}(k))$$

for the predicted sigma state estimate,

$$\hat{\mathbf{x}}(k+1|k) = \sum_{i=0}^{2L} W_i^m \boldsymbol{\mathcal{X}}_i^x(k+1|k)$$

for the predicted state estimate, and

$$\hat{\mathbf{P}}(k+1|k) = \sum_{i=0}^{2L} W_i^m [\boldsymbol{\mathcal{X}}^x(k+1|k) - \hat{\mathbf{x}}(k+1|k)][\boldsymbol{\mathcal{X}}^x(k+1|k) - \hat{\mathbf{x}}(k+1|k)]^T + \mathbf{Q}(k)$$

for the predicted estimation error covariance.

4.3.4 Update step

As defined in Section 4.3.1, $\boldsymbol{\mathcal{Y}}_i$ are the new points obtained from passing the sigma points through the required function. This way,

$$\boldsymbol{\mathcal{Y}}_i(k+1|k) = \mathbf{h}(k+1, \boldsymbol{\mathcal{X}}^x(k+1|k), \boldsymbol{\mathcal{X}}^n(k+1|k)).$$

The updated equations of the update step of the UKF are given by

$$\hat{\mathbf{y}}(k+1|k) = \sum_{i=0}^{2L} W_i^m \boldsymbol{\mathcal{Y}}_i(k+1|k)$$

for the predicted measurement,

$$\mathbf{res}(k+1) = \mathbf{y}(k+1) - \hat{\mathbf{y}}(k+1|k)$$

for the measurement residual,

$$\mathbf{S}(k+1) = \sum_{i=0}^{2L} W_i^c[\boldsymbol{\mathcal{Y}}_i(k+1|k) - \hat{\mathbf{y}}(k+1|k)][\boldsymbol{\mathcal{Y}}_i(k+1|k) - \hat{\mathbf{y}}(k+1|k)]^T + \mathbf{R}(k+1)$$

for the residual covariance,

$$\mathbf{P}_{xy}(k+1) = \sum_{i=0}^{2L} W_i^c[\boldsymbol{\mathcal{X}}^x(k+1|k) - \hat{\mathbf{x}}(k+1|k)][\boldsymbol{\mathcal{Y}}_i(k+1|k) - \hat{\mathbf{y}}(k+1|k)]^T$$

for the cross-covariance,

$$\mathbf{W}(k+1) = \mathbf{P}_{xy}(k+1)\mathbf{S}(k+1)^{-1} \tag{4.8}$$

for the filter gain,

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{W}(k+1)\mathbf{res}(k+1)$$

for the updated state estimate, and

$$\hat{\mathbf{P}}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{W}(k+1)\mathbf{S}(k+1)\mathbf{W}^T(k+1)$$

for the updated estimation error covariance.

4.3.5 Limitations

The UKF is overall a better alternative than the EKF, particularly for highly nonlinear functions. It also presents an opportunity for estimation when the system does not have differentiable functions, which renders the EKF incapable of performing the task, due to the need to calculate the associated Jacobians. So, if the original distribution is far from Gaussian, if the system is highly nonlinear and, particularly, if the functions are not differentiable, this filter is the best option. However, the closer to a linear problem and/or to a Gaussian prior distribution, the closer the UKF and EKF performances will be. Due to the higher computational need for the UKF, especially when the Jacobians of the EKF are fairly simple to obtain, the latter might still be a better option for problems which are close to linear.

5

Proposed solution with bottom-lock

5.1 System dynamics and models

Recall the system dynamics presented in (3.7). For simulation purposes, the desired functions were developed and tested resorting to MATLAB R2019b and Simulink. Since the system inputs and pseudo-range measurements are obtained from sensors, these involve setting sampling rates. For these simulations, the sampling times for both sensors are set to $T = 10\,\mathrm{s}$ and a final simulation time $t_{final} = 10000\,\mathrm{s}$ is considered. A Simulink model is designed to generate the system input $\mathbf{u}$, following

$$\mathbf{u}(k) = \begin{bmatrix} \cos\left(\frac{2\pi k}{30}\right) \\ 1.7\cos\left(\frac{2\pi k}{20} + \frac{\pi}{6}\right) \\ 2\cos\left(\frac{2\pi k}{45} + \frac{\pi}{9}\right) \end{bmatrix}.$$

This choice of input is related to the necessity for the system to be observable. Although the topic of the system observability is not within the scope of this work, an extensive comment is made in [23] regarding the need for rich trajectories. This input allows the trajectory of the agent to be rich enough so that it includes changes in all position components, without falling into special cases, such as circular trajectories where the range measurements would be virtually constant apart from the influence of noise.

A second Simulink model, which simulates the system dynamics, is presented in Fig. 5.1. In order to be able to perform an analysis of the simulation results, namely to able to calculate estimation errors, it is necessary to have access to all state variables. For this purpose, an auxiliary system is simulated, by using a discrete state-space Simulink block. Because the state and input matrices of the auxiliary system are the same as in (3.8), the state variables have the same behaviour as in the problem at hand. Since the output and feedforward matrices of the auxiliary system are set to $\mathbf{C} = \mathbf{I}$ and $\mathbf{D} = \mathbf{0}$, it directly outputs $\mathbf{x}$. A MATLAB function block is then used to calculate the necessary norm for the system pseudo-range measurements.

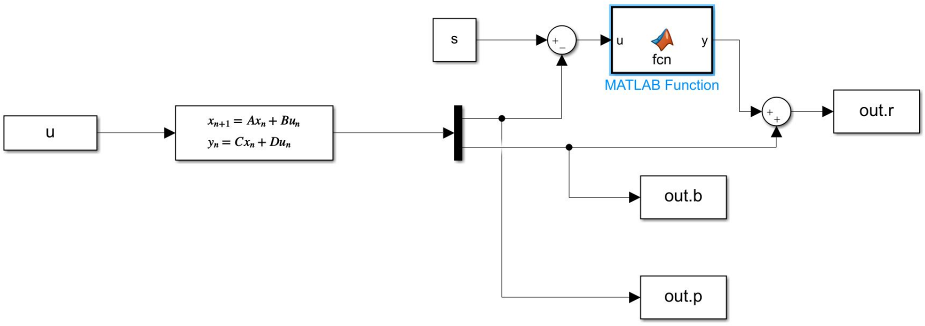

Figure 5.1: Simulink model to simulate the nominal system dynamics.

The overall model takes as input the generated $\mathbf{u}$, the state-space matrices necessary for the auxiliary system, as well as

$$\mathbf{s} = \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} [\mathrm{m}], \quad \mathbf{p}(0) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} [\mathrm{m}], \quad b_c(0) = 2\,\mathrm{m}, \quad \text{and} \quad \mathbf{x}(0) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \end{bmatrix} [\mathrm{m}],$$

where $\mathbf{s}$ is the constant inertial position of the source, $\mathbf{p}(0)$ is the initial position of the vehicle, $b_c(0)$ is the initial bias value, and so, $\mathbf{x}(0)$ is the initial state of the system. It outputs onto the workspace the values of $\mathbf{p}$, b_c, and $\mathbf{r}$ over time, which represent simulated true values for the states and pseudo-range over the run time. These allow for the visualization of the true value of the bias over the simulation time, represented in Fig. 5.2, as well as the true value of the pseudo-range measurements over time, represented in 5.3. A 3-D plot of the trajectory of the agent is also shown in Fig. 5.4.

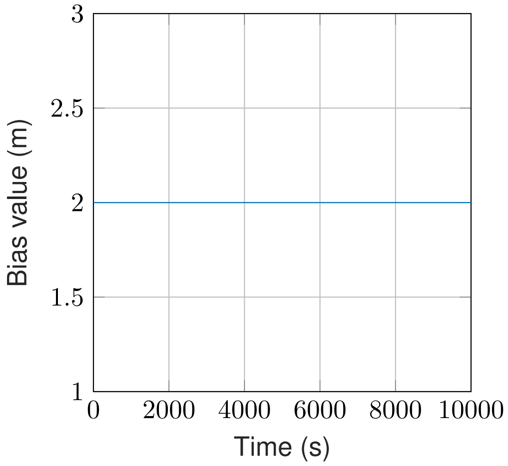

Figure 5.2: Plot of the true bias value.

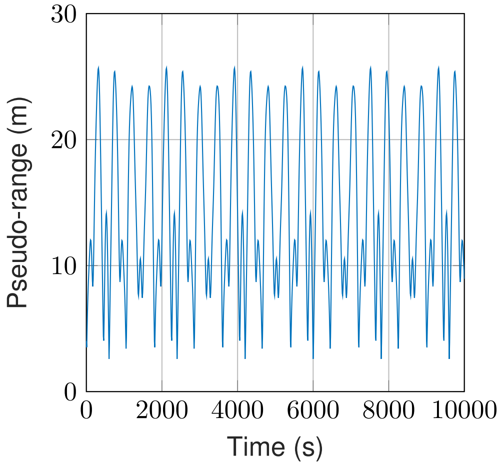

Figure 5.3: Plot of the true pseudo-range values.

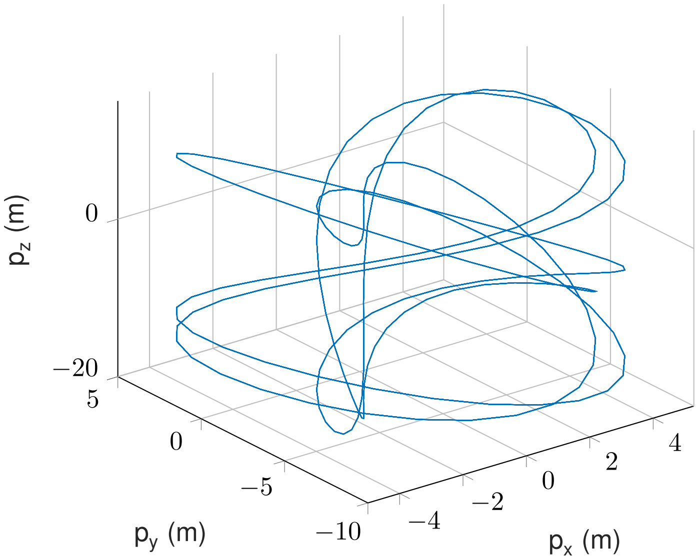

Figure 5.4: 3-D plot of the vehicle trajectory.

5.2 Noise description

Considering the discrete-time nonlinear system in (3.7), used to describe the problem at hand, it is necessary to account for potential noises related to the system. In this case, it is assumed that there are zero-mean, additive uncorrelated white Gaussian noises associated with the sensors used for the measurement of the system input and the system pseudo-range measurements. Taking these into account, a more accurate representation of the problem is

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}[\mathbf{u}(k) + \mathbf{m}(k)] \\ r(k) = \|\mathbf{s} - \mathbf{x}_1(k)\| + x_2(k) + n(k) \end{cases} \tag{5.1}$$

or, equivalently,

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{w}(k) \\ r(k) = \|\mathbf{s} - \mathbf{x}_1(k)\| + x_2(k) + n(k) \end{cases} \tag{5.2}$$

where $n(k) \in \mathbb{R}$ is the Gaussian observation noise, $n(k) \sim \mathcal{N}(0,\, R(k))$, and $\mathbf{m}(k) \in \mathbb{R}^4$ is the Gaussian process noise, $\mathbf{m}(k) \sim \mathcal{N}(\mathbf{0},\, \mathbf{M}(k))$, associated with the measurement of the vehicle position displacements, $\mathbf{u}(k)$. Because of the values of matrix $\mathbf{B}$, the values of vector $\mathbf{w}(k)$ come as $\mathbf{w}(k) = \begin{bmatrix} \mathbf{m}(k) \\ 0 \end{bmatrix}$, therefore, $\mathbf{w}(k) \sim \mathcal{N}(\mathbf{0},\, \mathbf{Q}(k))$, where $\mathbf{Q}(k) = \mathbf{diag}(\mathbf{M}_{1,1}(k), \mathbf{M}_{2,2}(k), \mathbf{M}_{3,3}(k), 0)$. In practice, zero entries in $\mathbf{diag}(\mathbf{Q}(k))$ are undesirable because of the aspects described in Section 5.2.1, and so $\mathbf{Q}_{4,4}(k)$ is always set to a small yet non-zero value.

5.2.1 Noise controllability

Considering that the system at hand is described by a state equation that is linear time-invariant, and that the controllability of the system is only dependent on its state and input matrices, respectively $\mathbf{A}$ and $\mathbf{B}$, a test on the controllability of the system can be performed by analysing the column rank of the matrix $\mathbf{C}_o$, given by

$$\mathbf{C}_o = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \mathbf{A}^3\mathbf{B} \end{bmatrix}.$$

Because the state vector is of size $n = 4$ and the column rank of $\mathbf{C}_o$ is of only 3, it shows that the system is not fully controllable by its input $\mathbf{u}(k)$. This is not a problem and has no implications in the KF, except for the fact that the control input $\mathbf{u}(k)$ does not affect the bias element of the state estimate, $\hat{\mathbf{x}}$, during the prediction step.

As argued in [30], a similar analysis can be performed regarding the process noise. By re-writing (5.2) as

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{Ax}(k) + \mathbf{Bu}(k) + \mathbf{Ww}(k) \\ r(k) = \|\mathbf{s} - \mathbf{x}_1(k)\| + x_2(k) + n(k) \end{cases},$$

one can consider $\mathbf{w}(k)$ as a second input for the system and perform another controllability analysis to obtain the ability of the noise to affect the state. In this case, $\mathbf{w}(k)$ is a vector of independent, identically distributed unit variance Gaussian noise sources, to which the gains in $\mathbf{W}$ are applied. The system noise covariance matrix $\mathbf{Q}$ is given by $\mathbf{Q} = \mathbf{W}^T\mathbf{W}$. If the system formed by $\mathbf{A}$ and $\mathbf{W}$ is uncontrollable, then the noise does not affect all the elements of the state and $\mathbf{Q}$ has zero diagonal entries for the uncontrollable ("incorruptible") states. This can be undesirable because the KF will drive the corresponding diagonal elements of $\mathbf{P}$ to zero, since it minimizes its trace. Once that happens, the estimates of these states are fixed, which means that no further observations will alter their values. To ensure a fulfilment of the KF requirement for noise stabilizability and to avoid the situation mentioned before, all the diagonal entries of matrix $\mathbf{Q}$ are set to low, yet non-zero values, to ensure that the system is noise controllable.

5.3 Filter parameters and design

In order to design an estimator for the system described by (3.7), an EKF, an UKF, and a LKF were designed and tested, with and without sensor noises, and for different initial conditions. The construction of each filter is described in the following sections.

5.3.1 EKF parameters and design

The function developed to simulate the EKF receives as input the state and input matrices, respectively $\mathbf{A}$ and $\mathbf{B}$, as well as the constants T, representing the sampling period, t_{final} representing the final simulation time, and $\mathbf{s}$, representing the fixed location of the source. The noise covariance matrices $\mathbf{Q}$ and R are also supplied, as well as the range and input values measured by the sensors, respectively $\mathbf{r}$

and $\mathbf{u}$. Finally, the function receives the initial estimates for the state values and covariance matrix, $\mathbf{x}(0)$ and $\mathbf{P}(0)$, respectively. The function then outputs a matrix with the state estimates for each iteration and a cell-array with all the estimated covariance matrices, one $\mathbf{P}$ matrix associated with each filter iteration.

For the simulations in which no noise is considered, $\mathbf{r}$ and $\mathbf{u}$ are simply the real range and input values, whereas, when adding sensor noise, $\mathbf{r}$ and $\mathbf{u}$ are affected by additive white Gaussian noises, given by $n \sim \mathcal{N}(0,\, R)$ and $\mathbf{w} \sim \mathcal{N}(\mathbf{0},\, \mathbf{Q})$, respectively. $\mathbf{Q}$ is a diagonal matrix, $\mathbf{Q} = \mathbf{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_b^2)$, where σ_b^2 is the small value discussed in Section 5.2.1, and σ_x^2, σ_y^2, σ_z^2 are the variances of the measurements of the system inputs, which correspond to the variances of the vehicle position displacement components in the x, y, and z axes, respectively. $R = \sigma^2$, where σ^2 is the variance of the range measurement. Note that, because of the matrix inversion in equation (4.4), R cannot be null.

The simulation results of the designed EKF are shown in Section 5.4.1. The Jacobian matrices mentioned in (4.2) and (4.3), are, in the case of the system at hand

$$\mathbf{F} = \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{H}(k+1) = \begin{bmatrix} \dfrac{\hat{p}_x(k+1|k) - \mathrm{s}_x}{\|\mathbf{s} - \hat{\mathbf{p}}(k+1|k)\|} & \dfrac{\hat{p}_y(k+1|k) - \mathrm{s}_y}{\|\mathbf{s} - \hat{\mathbf{p}}(k+1|k)\|} & \dfrac{\hat{p}_z(k+1|k) - \mathrm{s}_z}{\|\mathbf{s} - \hat{\mathbf{p}}(k+1|k)\|} & 1 \end{bmatrix}. \tag{5.3}$$

Matrix $\mathbf{F}$ can be obtained from the fact that the system state equation is linear in $\mathbf{x}(k)$. As for matrix $\mathbf{H}(k{+}1)$, the derivative of $r(k{+}1)$ with respect to each of the predicted state variables must be calculated, and so

$$\mathbf{H}(k+1) = \frac{\partial r(k+1)}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\hat{\mathbf{x}}(k+1|k)} = \begin{bmatrix} \dfrac{\partial r(k+1)}{\partial \hat{p}_x(k+1|k)} & \dfrac{\partial r(k+1)}{\partial \hat{p}_y(k+1|k)} & \dfrac{\partial r(k+1)}{\partial \hat{p}_z(k+1|k)} & \dfrac{\partial r(k+1)}{\partial \hat{b}_c(k+1|k)} \end{bmatrix}.$$

From the pseudo-range expression in (3.2), it is possible to conclude that $\dfrac{\partial r(k+1)}{\partial \hat{b}_c(k+1|k)} = 1$. To compute the derivative with respect to $\hat{p}_x(k+1|k)$, it is useful to rewrite

$$r(k+1) = \sqrt{\sum_{i=x,y,z} (\mathrm{s}_i - \hat{p}_i(k+1|k))^2} + \hat{b}_c(k+1|k).$$

From here, the calculation of the partial derivative becomes straightforward and is given by

$$\frac{\partial r(k+1)}{\partial \hat{p}_x(k+1|k)} = \frac{1}{2} \times \frac{1}{\sqrt{\sum_{i=x,y,z} (\mathrm{s}_i - \hat{p}_i(k+1|k))^2}} \times 2(\mathrm{s}_x - \hat{p}_x(k+1|k)) \times (-1),$$

which can be simplified to

$$\frac{\partial r(k+1)}{\partial \hat{p}_x(k+1|k)} = \frac{\hat{p}_x(k+1|k) - \mathrm{s}_x}{\|\mathbf{s} - \hat{\mathbf{p}}(k+1|k)\|}.$$

A similar calculation is done for $\hat{p}_y(k+1|k)$ and $\hat{p}_z(k+1|k)$, which yields (5.3).

5.3.2 UKF parameters and design

The function that was developed to simulate the UKF is very similar to the one described in Section 5.3.1: it also receives as input the state and input matrices, as well as the sampling period, the final simulation time, and the fixed location of the source. The noise covariance matrices are also supplied, as well as the range and input values measured by the sensors, and the initial estimates for the state vector and covariance matrix. Finally, in order to perform the UT, the filter function needs to receive the parameters that relate to its use, namely L, k, β, α, and λ, where $\lambda = \alpha^2(L+k) - L$. The function then outputs a matrix with the state estimates and a cell-array with the estimated covariance matrices.

The considerations about the range and input values, $\mathbf{r}$ and $\mathbf{u}$, as well as the comments about $\mathbf{Q}$ and R, are the same as the ones presented in Section 5.3.1. Namely, because of the matrix inversion in equation (4.8), R cannot be null. The simulation results of the designed UKF, are shown in Section 5.4.2.

5.3.3 LKF parameters and design

The filter presented in this section is based on the solution of the single vehicle navigation problem introduced in [23], in which an augmented state is suggested. To begin the description of the filter design, one must start by analysing the problem formulation. The original formulation of the problem, in (3.6), now uses a different choice of state variables, in this case,

$$\begin{cases} \mathbf{x}_1(k) = \mathbf{p}(t_k) - \mathbf{s} \\ x_2(k) = b_c(t_k) \end{cases},$$

still resulting in the system

$$\begin{cases} \mathbf{x}_1(k+1) = \mathbf{x}_1(k) + \mathbf{u}(k) \\ x_2(k+1) = x_2(k) \end{cases},$$

with $\mathbf{u}(k) = \displaystyle\int_{t_k}^{t_{k+1}} \mathbf{R}(\tau)\mathbf{v}(\tau)\, d\tau$.

The state augmentation suggested in the paper takes advantage of the fact that the only nonlinearity in this problem comes from the system range measurements. Because these values are measured at each timestamp, and hence, are accessible to the state, it is possible to define

$$\begin{cases} \mathbf{z}_1(k) = \mathbf{x}_1(k) \\ z_2(k) = x_2(k) \\ z_3(k) = r(k) \end{cases},$$

where $r(k) = \|\mathbf{x}_1(k)\| + x_2(k)$, which yields the new state vector $\mathbf{z}(k) = \begin{bmatrix} \mathbf{z}_1(k) \\ z_2(k) \\ z_3(k) \end{bmatrix}$.

To write the system in the state-space form, one must consider that

$$\begin{cases}\mathbf{z}_1(k+1) = \mathbf{z}_1(k) + \mathbf{u}(k) \\ z_2(k+1) = z_2(k) \\ z_3(k+1) = \|\mathbf{z}_1(k+1)\| + z_2(k+1)\end{cases}.$$

Combining these equations gives

$$z_3(k+1) = \|\mathbf{z}_1(k) + \mathbf{u}(k)\| + z_2(k). \tag{5.4}$$

By squaring and expanding (5.4), one can obtain

$$z_3^2(k+1) = [\|\mathbf{z}_1(k) + \mathbf{u}(k)\| + z_2(k)]^2 = \|\mathbf{z}_1(k) + \mathbf{u}(k)\|^2 + 2\|\mathbf{z}_1(k) + \mathbf{u}(k)\| z_2(k) + z_2(k)^2,$$

which can be further expanded as

$$z_3^2(k+1) = \|\mathbf{z}_1(k)\|^2 + 2\mathbf{u}(k)^T\mathbf{z}_1(k) + \mathbf{u}(k)^T\mathbf{u}(k) + 2\|\mathbf{z}_1(k) + \mathbf{u}(k)\| z_2(k) + z_2(k)^2. \tag{5.5}$$

Noting that

$$z_3(k) = \|\mathbf{z}_1(k)\| + z_2(k) \iff z_2(k) = z_3(k) - \|\mathbf{z}_1(k)\| \iff \|\mathbf{z}_1(k)\| = z_3(k) - z_2(k),$$

it is possible to rewrite (5.5) as

$$z_3^2(k{+}1) = \|\mathbf{z}_1(k)\|[\,z_3(k){-}z_2(k)\,]{+}2\mathbf{u}(k)^T\mathbf{z}_1(k){+}\mathbf{u}(k)^T\mathbf{u}{+}2\|\mathbf{z}_1(k){+}\mathbf{u}(k)\|z_2(k){+}z_2(k)[\,z_3(k){-}\|\mathbf{z}_1(k)\|\,],$$

which leads to

$$z_3^2(k+1) = 2\mathbf{u}(k)^T\mathbf{z}_1(k) + [\,2\|\mathbf{z}_1(k) + \mathbf{u}(k)\| - 2\|\mathbf{z}_1(k)\|\,]z_2(k) + [\,z_2(k) + \|\mathbf{z}_1(k)\|\,]z_3(k) + \mathbf{u}(k)^T\mathbf{u}(k).$$

This can be simplified, as given by

$$z_3^2(k+1) = 2\mathbf{u}(k)^T\mathbf{z}_1(k) + [\,2\|\mathbf{z}_1(k) + \mathbf{u}(k)\| - 2\|\mathbf{z}_1(k)\|\,]z_2(k) + z_3^2(k) + \mathbf{u}(k)^T\mathbf{u}(k). \tag{5.6}$$

Noting that

$$z_3(k+1) - z_3(k) = \|\mathbf{z}_1(k) + \mathbf{u}(k)\| + z_2(k) - [\,\|\mathbf{z}_1(k)\| + z_2(k)\,] = \|\mathbf{z}_1(k) + \mathbf{u}(k)\| - \|\mathbf{z}_1(k)\|, \tag{5.7}$$

since $z_2(k+1) = z_2(k)$, one can replace (5.7) in (5.6) and obtain

$$z_3^2(k+1) = 2\mathbf{u}(k)^T\mathbf{z}_1(k) + 2[\,z_3(k+1) - z_3(k)\,]z_2(k) + z_3^2(k) + \mathbf{u}(k)^T\mathbf{u}(k).$$

Finally, one obtains

$$z_3(k+1) = \frac{2\mathbf{u}(k)^T}{z_3(k+1)}\mathbf{z}_1(k) + \frac{2[\,z_3(k+1) - z_3(k)\,]}{z_3(k+1)}z_2(k) + \frac{z_3(k)}{z_3(k+1)}z_3(k) + \frac{\mathbf{u}(k)^T}{z_3(k+1)}\mathbf{u}(k),$$

which is equivalent to

$$z_3(k+1) = \frac{2\mathbf{u}(k)^T}{r(k+1)}\mathbf{z}_1(k) + \frac{2[\,r(k+1) - r(k)\,]}{r(k+1)}z_2(k) + \frac{r(k)}{r(k+1)}z_3(k) + \frac{\mathbf{u}(k)^T}{r(k+1)}\mathbf{u}(k).$$

The system can, then, be expressed as

$$\begin{cases}\mathbf{z}_1(k+1) = \mathbf{z}_1(k) + \mathbf{u}(k) \\ z_2(k+1) = z_2(k) \\ z_3(k+1) = \dfrac{2\mathbf{u}(k)^T}{r(k+1)}\mathbf{z}_1(k) + \dfrac{2[\,r(k+1) - r(k)\,]}{r(k+1)} z_2(k) + \dfrac{r(k)}{r(k+1)} z_3(k) + \dfrac{\mathbf{u}(k)^T}{r(k+1)}\mathbf{u}(k) \\ y(k+1) = z_3(k+1)\end{cases},$$

which is equivalent to the state-space representation

$$\begin{cases}\mathbf{z}(k+1) = \mathbf{A}(k)\mathbf{z}(k) + \mathbf{B}(k)\mathbf{u}(k) \\ y(k+1) = \mathbf{C}\mathbf{z}(k+1)\end{cases}, \tag{5.8}$$

where

$$\mathbf{A}(k) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} & 1 & 0 \\ 2\dfrac{\mathbf{u}(k)^T}{r(k+1)} & 2\dfrac{[\,r(k+1) - r(k)\,]}{r(k+1)} & \dfrac{r(k)}{r(k+1)} \end{bmatrix} \in \mathbb{R}^{5\times 5}, \ \ \mathbf{B}(k) = \begin{bmatrix} \mathbf{I}_3 \\ \mathbf{0}_{1\times 3} \\ \dfrac{\mathbf{u}(k)^T}{r(k+1)} \end{bmatrix} \in \mathbb{R}^{5\times 3},$$

and $\mathbf{C} = \begin{bmatrix}\mathbf{0}_{1\times 3} & 0 & 1\end{bmatrix} \in \mathbb{R}^{1\times 5}$, which can be regarded as a discrete-time linear time-varying system for observer design purposes. Since the augmented system is equivalent to the system in (3.7), an observer (filter) for (5.8) is also an observer (filter) for (3.7).

Due to the time dependencies of matrices $\mathbf{A}$ and $\mathbf{B}$, they have to be calculated at every time step, which means these matrices are no longer inputs of the function designed to simulate the filter. For the LKF, the function receives, instead, matrix $\mathbf{C}$. The remaining inputs and outputs of the function remain the same as in Section 5.3.1, noting that the position estimates are now obtained by adding the position of the source, $\mathbf{s}$, to the first state variable, $\mathbf{z}_1(k)$.

Because the state and input matrices are no longer linear time-invariant, the analysis made in Section 5.2.1 no longer applies, however, a similar case can be made for the undesirable presence of zero entries in the covariance matrices for the LKF, and so, as a matter of consistency, the same non-zero entries will be considered.

It is important to note that unobservable systems prohibit the use of KFs, since no information is obtained through the observation equations for the unobservable states. This means the filter estimate for these states will not converge to a meaningful solution [30]. The thorough analysis of the observability of both the original nonlinear system and the derived augmented linear system is presented in [23]. To this end, it is noted that the new systems can still be considered linear in the state for observer design purposes, since $\mathbf{A}(k)$ depends on pseudo-range measurements which are assumed to be available and, hence, considered as functions of time for observer (filter) design purposes. The analysis performed allows for the design of these filters.

Because of the augmented state, the covariance of the process noise for this filter is now $\mathbf{Q} = \mathbf{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_b^2, \sigma_r^2)$, where σ_r^2 is also chosen as a small yet non-zero value. The value of R remains the same, i.e., $R = \sigma^2$, and, once again, because of the matrix inversion in (4.1) cannot be null. The simulation results of the designed LKF with augmented state are shown in Section 5.4.3.

5.4 Simulation results with bottom-lock

The following sections present the selected conditions that show filter convergence for each of the three designed filters, as well as the results obtained for the estimation errors. For the EKF and the UKF, a set of conditions for which the filter does not converge is also presented, along with the simulation results. Some considerations about the convergence of the LKF and a comparison with the non-convergence cases of the two alternative filters is also made. The choice of the noise covariance matrices and non-zero initial error conditions is made such that direct comparison between all filters is possible. As per the results in Section 5.4.2, the solution with the most difficult convergence is the UKF, and so, the previous values are chosen to ensure this filter converges, and used throughout all filter experiments.

5.4.1 EKF results with bottom-lock

All the simulations in this section imply a prior run of the previously mentioned Simulink models, in order to obtain the inputs, ranges, and states necessary for the estimations and error calculations. For all, the initial covariance matrix is set to $\mathbf{P}(0) = \mathbf{I}$ and the noise covariance matrices are given by $\mathbf{Q} = \mathbf{diag}(10^{-4}\,\mathbf{I}_3, 10^{-4})$ and $R = 10^{-2}$.

5.4.1.1 Filter convergence

To check for filter converge, one should initially run the filter with correct starting conditions and no noise and check for propagation errors. A run with sensor noise is performed to ensure the filter is robust to its influence. Because in real-world scenarios one does not, necessarily, have access to an accurate initial estimate, a run with non-zero initial error should also be performed, to ensure the filter can overcome this issue. Finally, the experiment should be repeated for situations with sensor noise, in order to have the best simulation of the real-world applications of the filter.

5.4.1.1.1 Correct initial conditions

For this experiment, the true starting conditions are considered, i.e., $\mathbf{x}(0) = \begin{bmatrix}1 & 1 & 1 & 2\end{bmatrix}^T$ $[\mathrm{m}]$, and the true values of the range and inputs are used. The resulting plots for the position estimation error and bias estimation error are presented in Figures 5.5 and 5.6, respectively. These show the expected outcomes, since the errors start and remain at very small values for all time, which can be taken as zero, given the scale of this problem, and are consistent with the known EKF response under these conditions. These results prove that the designed filter does not have any propagation errors.

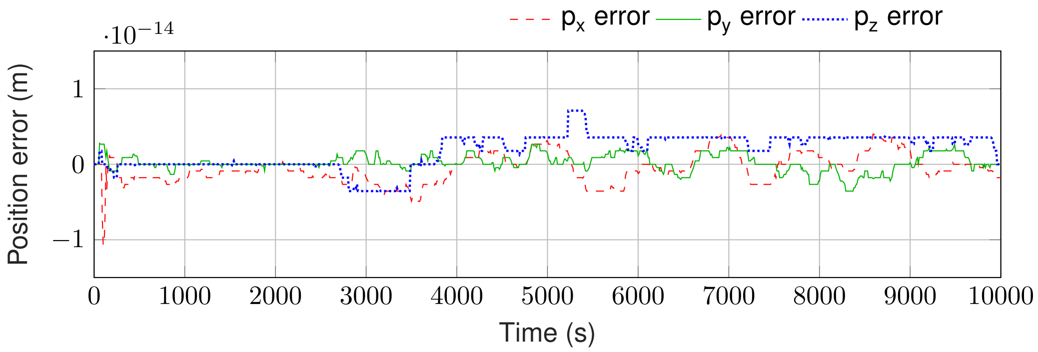

Figure 5.5: Position estimation error of the EKF in the absence of sensor noise: zero initial error.

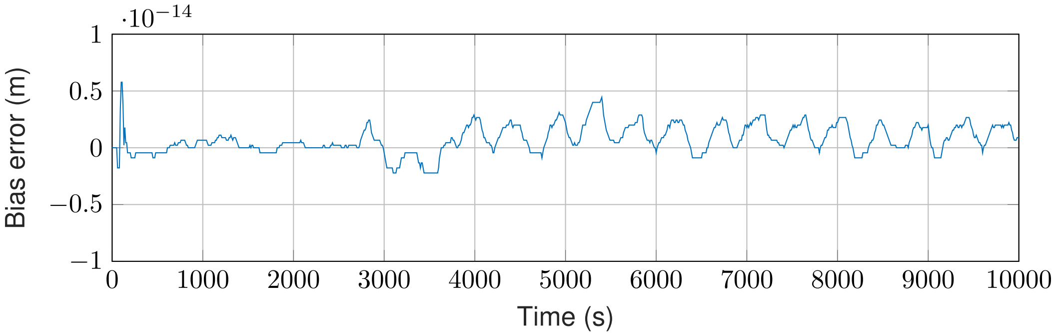

Figure 5.6: Bias estimation error of the EKF in the absence of sensor noise: zero initial error.

5.4.1.1.2 Correct initial conditions with added noise

For this experiment, the initial conditions are set to the true value, $\mathbf{x}(0) = \begin{bmatrix}1 & 1 & 1 & 2\end{bmatrix}^T$ [m], and the measured (noisy) values of the range and inputs are considered. The position estimation error and bias estimation error are presented in Figures 5.7 and 5.8, respectively. The latter exhibits some stronger initial transients and then converges, while showing a slight offset, with $0.026\,\mathrm{m}$ as the final error. As for the position, the steady-state errors are within $-0.2\,\mathrm{m}$ and $0.2\,\mathrm{m}$ of the true values, with the final error being $\mathbf{p}_{error} = \begin{bmatrix}0.090 & 0.006 & 0.038\end{bmatrix}^T$ [m]. The initial transients shown in both plots are a result of the twofold impact of the introduced noises: directly, through the measured values of $\mathbf{r}$ and $\mathbf{u}$; and indirectly, through the propagation of matrices $\mathbf{Q}$ and R throughout the estimation steps shown in Section 4.2.

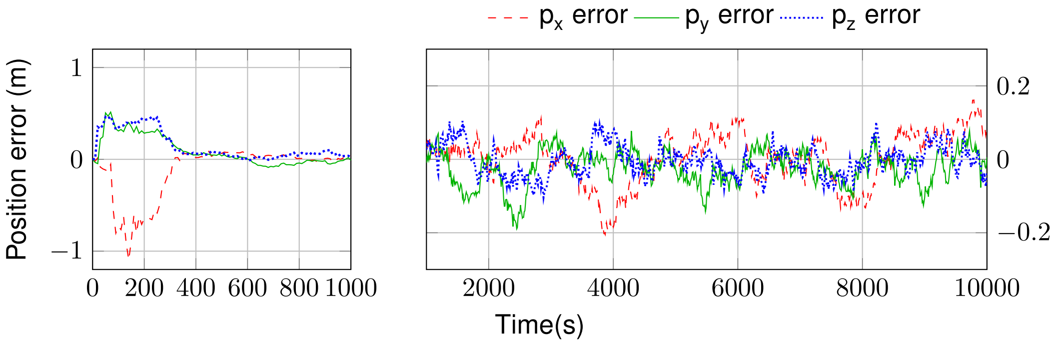

Figure 5.7: Position estimation error of the EKF in the presence of sensor noise: zero initial error.

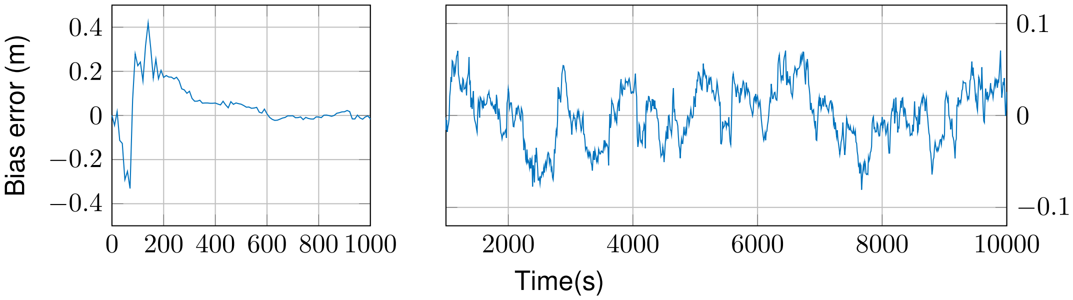

Figure 5.8: Bias estimation error of the EKF in the presence of sensor noise: zero initial error.

5.4.1.1.3 Non-zero initial error

For this experiment, the initial conditions are set to $\mathbf{x}(0) = \begin{bmatrix} -4 & -4 & -4 & 1 \end{bmatrix}^T$ $[\mathrm{m}]$ and the true values of the range and inputs are considered. The resulting plots for the position estimation error and bias estimation error are presented in Figures 5.9 and 5.10, respectively. These show that the filter is fairly robust to changes in the initial estimate, since both the position and bias errors rapidly converge to zero, after the initial transients disappear. The bias estimation error reaches approximately $3.28 \times 10^{-12}\,\mathrm{m}$ at the final simulation time and the final position vector comes to $\mathbf{p}_{error} \approx \begin{bmatrix} -1.11 & 0.37 & 0.41 \end{bmatrix}^T \times 10^{-11}\,[\mathrm{m}]$, which are both very satisfactory results and can be regarded as zero, given the scale of the experiment.

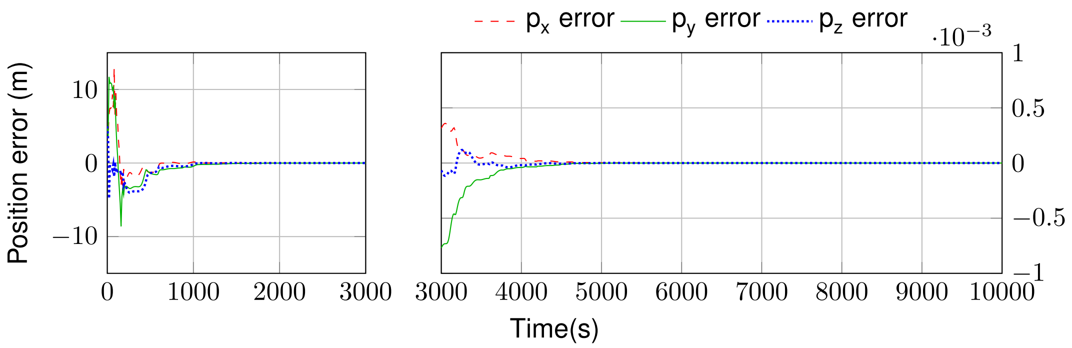

Figure 5.9: Position estimation error of the EKF in the absence of sensor noise: non-zero initial error.

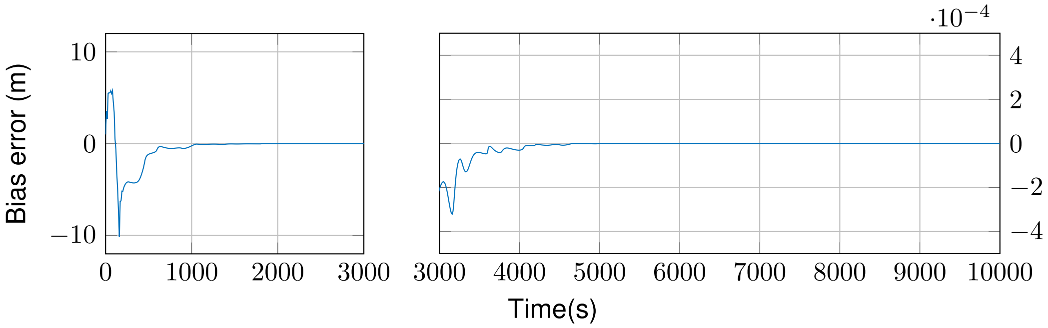

Figure 5.10: Bias estimation error of the EKF in the absence of sensor noise: non-zero initial error.

5.4.1.1.4 Non-zero initial error with added noise

For this experiment, the initial conditions are set to $\mathbf{x}(0) = \begin{bmatrix} -4 & -4 & -4 & 1 \end{bmatrix}^T$ $[\mathrm{m}]$ and the measured (noisy) values of the range and inputs are considered. The resulting plots for the position estimation error and bias estimation error are presented in Figures 5.11 and 5.12, respectively. The first starts by peaking and then converges to an interval between $-0.2\,\mathrm{m}$ and $0.2\,\mathrm{m}$. As for the bias error, it also shows a peak in the initial $200\,\mathrm{s}$ and then converges to a value close to 0, maintaining the slight estimation offset shown in 5.4.1.1.2.

From these results, one can attest to the relative robustness of the filter, since a convergence in the given simulation time can be achieved from a non-zero error starting condition with added sensor noise. When analysing these results, one should keep in mind that this is quite a tricky problem to solve, since the filter only has access to the measured pseudo-range values, which could be derived from a number of combinations of p_x, p_y, p_z, and b_c values. The better results obtained in the bias estimation can be justified by the fact that it is a simpler estimation to begin with, since it is a constant value, rather than a rapidly varying one. This information is available to the filter from the Jacobian matrix $\mathbf{H}$ in (5.3).

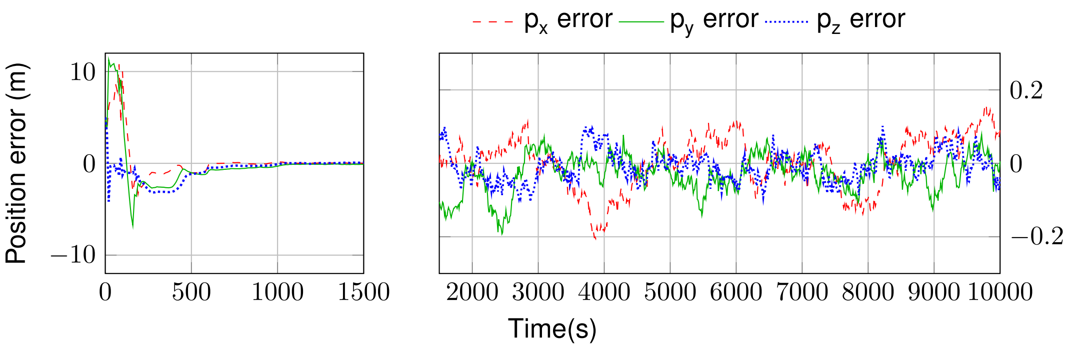

Figure 5.11: Position estimation error of the EKF in the presence of sensor noise: non-zero initial error.

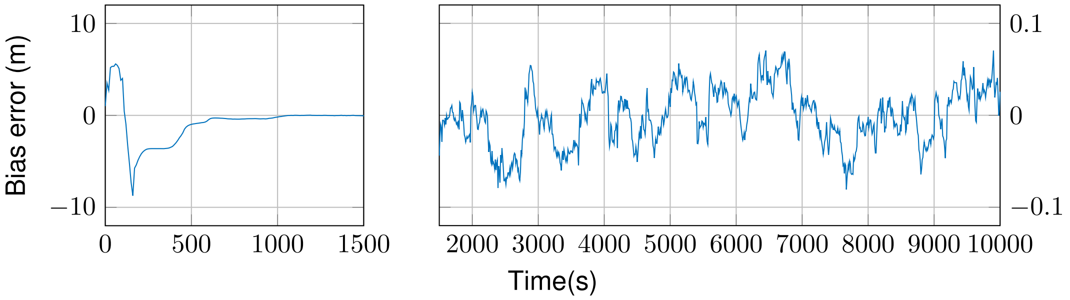

Figure 5.12: Bias estimation error of the EKF in the presence of sensor noise: non-zero initial error.

5.4.1.2 Failure of convergence of the EKF

The EKF does not provide any guarantees for convergence. In fact, when testing the filter, several runs were found not to converge or not converge in the established time frame. For the most part, however, one is able to obtain good results, so long as the initial state estimate is not too far from its true value. To exemplify a case in which a convergence in the given time limit is not achieved, a run with sensor noise following the same distribution as before and initial conditions as the sum of the true value plus a large offset is performed. In this case, $\mathbf{x}(0) = \begin{bmatrix} -999 & -699 & -999 & -498 \end{bmatrix}^T$ $[\mathrm{m}]$.

The resulting plots for the position estimation error and bias estimation error are presented in Figures 5.13 and 5.14, respectively. The failure to converge is noticeable in both plots. Indeed, the error of p_y starts at $700\,\mathrm{m}$, decreases to $400\,\mathrm{m}$, and then maintains an overall increase until the final simulation time. The error of p_z oscillates between $660\,\mathrm{m}$ and $640\,\mathrm{m}$ throughout the whole simulation and the error of p_x does follow a decreasing trajectory but does not come close to converging. As for the bias, its error starts at $500\,\mathrm{m}$, peaks around $1035\,\mathrm{m}$ and then maintains an oscillation between $1038\,\mathrm{m}$ and $1030\,\mathrm{m}$ throughout the simulation time, while slowly decreasing but never coming close to zero.

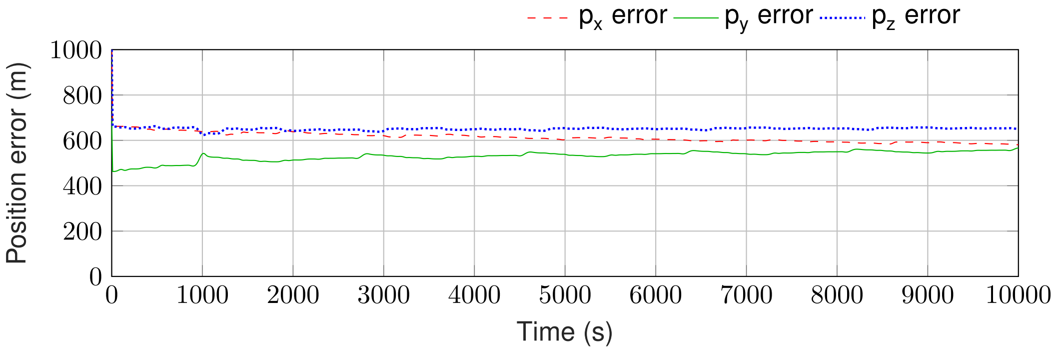

Figure 5.13: Position estimation error of the EKF: failure of convergence.

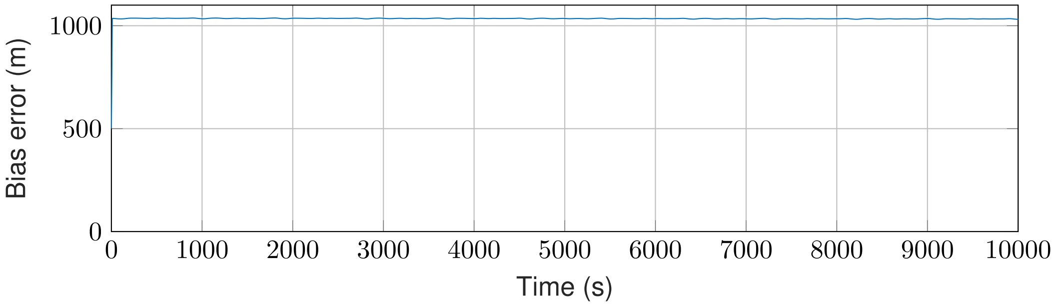

Figure 5.14: Bias estimation error of the EKF: failure of convergence.

5.4.2 UKF results with bottom-lock

Like before, all the simulations in this section imply a prior run of the Simulink models, in order to obtain the inputs, ranges, and states necessary for the estimations and error calculations. For all experiments, the initial covariance matrix is set to $\mathbf{P}(0) = \mathbf{I}$ and the noise covariance matrices are $\mathbf{Q} = \mathbf{diag}(10^{-4}\mathbf{I}_3, 10^{-4})$ and $R = 10^{-2}$. As mentioned in Section 5.3.2, the UKF requires some additional tuning parameters, which are set to the standard values of $k = 0$, $\alpha = 10^{-3}$, $\beta = 2$, and $\lambda = \alpha^2(L+k) - L$ for all experiments. L is defined as being equal to $2n + 1$, where n is the size of the state vector. In this case, it is then set to $L = 9$.

5.4.2.1 Filter convergence

The convergence of the filter is tested over several experiments, as mentioned in Section 5.4.1.1. The results for each set of conditions are shown in the following sections.

5.4.2.1.1 Correct initial conditions

For this experiment, the initial conditions are set to the true values, i.e., $\mathbf{x}(0) = \begin{bmatrix} 1 & 1 & 1 & 2 \end{bmatrix}^T$ [m], and the true values of the range and inputs are considered. The resulting plots for the position estimation error and bias estimation error are presented in Figures 5.15 and 5.16, respectively.

These plots show a few initial differences in the UKF behaviour when compared to the EKF. Firstly, whereas the EKF takes only the mean point of the Gaussian distribution and passes it through the linearization of the nonlinear function, the UKF takes several sigma points to pass through the UT, which means they do not all fall on the mean. In practical terms, this implies that, if the correct initial conditions are given with no sensor noise, i.e. the mean is zero and there are no disturbances in the measurements, the EKF has errors that start and stay at zero, as shown in Fig. 5.5, whereas the UKF does not follow this behaviour, due to the weights assigned to the sigma points that do not fall on the mean. This justifies the initial transients present in both the position and the bias error plots for this experiment. Another noticeable difference is that the final errors are much larger than the ones presented for the EKF. However, since these errors are still three orders of magnitude smaller than the nominal values and convergence is clearly shown, one can see that the designed filter does not have any propagation errors.

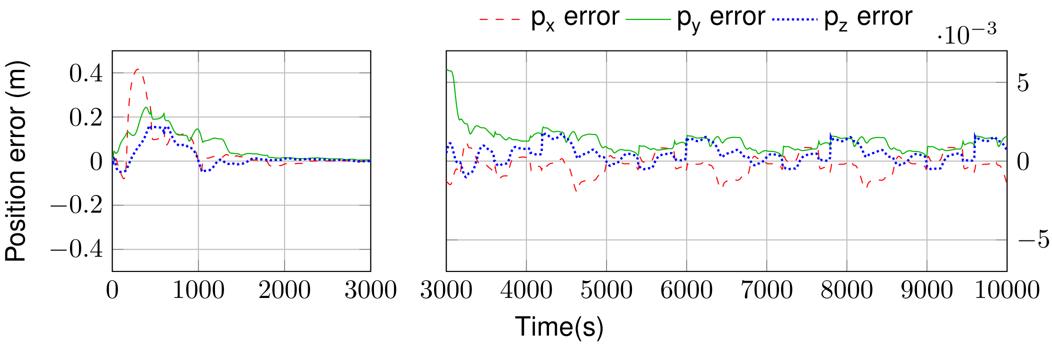

Figure 5.15: Position estimation error of the UKF in the absence of sensor noise: zero initial error.

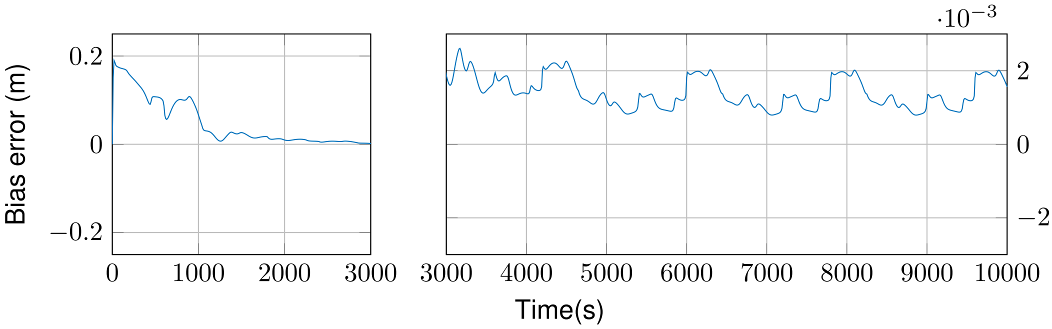

Figure 5.16: Bias estimation error of the UKF in the absence of sensor noise: zero initial error.

5.4.2.1.2 Correct initial conditions with added noise

For this experiment, the initial conditions are set to the true values, i.e., $\mathbf{x}(0) = \begin{bmatrix}1 & 1 & 1 & 2\end{bmatrix}^T$ [m] and the measured (noisy) values of the range and inputs are considered. The resulting plots for the position estimation error and bias estimation error are presented in Figures 5.17 and 5.18, respectively. Both show initial transients, with the estimation errors converging to the vicinity of zero, which shows the filter to be robust to noise, under these conditions. The position error in steady-state is within approximately $-0.2\,\mathrm{m}$ and $0.2\,\mathrm{m}$ of the true values and the bias steady-state error is between $-0.05\,\mathrm{m}$ and $0.05\,\mathrm{m}$ of the true values. The final errors achieved are $\mathbf{p}_{error} \approx \begin{bmatrix}0.049 & 0.033 & -0.022\end{bmatrix}^T$ [m] and $b_{c_{error}} \approx 0.0113\,\mathrm{m}$.

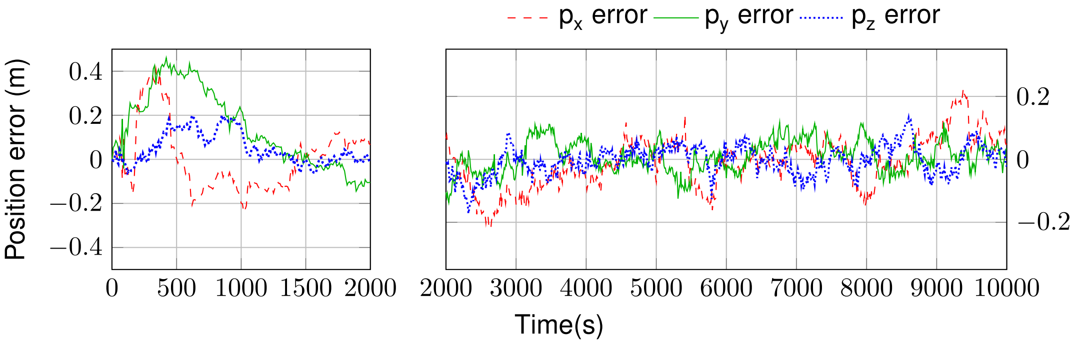

Figure 5.17: Position estimation error of the UKF in the presence of sensor noise: zero initial error.

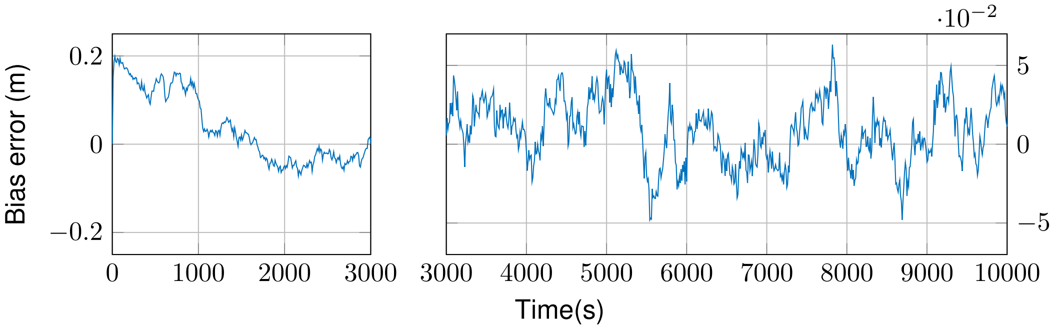

Figure 5.18: Bias estimation error of the UKF in the presence of sensor noise: zero initial error.

5.4.2.1.3 Non-zero initial error

For this experiment, the initial conditions are set to $\mathbf{x}(0) = \begin{bmatrix}-4 & -4 & -4 & 1\end{bmatrix}^T$ $[\mathrm{m}]$ and the true values of the range and inputs are considered. The resulting plots for the position estimation error and bias estimation error are presented in Figures 5.19 and 5.20, respectively.

One major difference between this experiment and the one presented in Section 5.4.1.1.3 is that this filter takes significantly longer to converge than the EKF. This is due to the fact that the system is linear in the state, which causes the UKF to lose its precision advantages, since the higher order moments in the state error distributions are not significant in this case [31]. The UKF also presents a higher computational challenge because of its need to compute and propagate several sigma points. Nonetheless, because the final errors achieved are of $1.61 \times 10^{-3}\,\mathrm{m}$ for the bias and $\begin{bmatrix}1.50 & 1.59 & 0.63\end{bmatrix}^T \times 10^{-3}\,[\mathrm{m}]$ for the position, one can consider them satisfactory in showing that the filter is fairly robust to changes in the initial estimate.

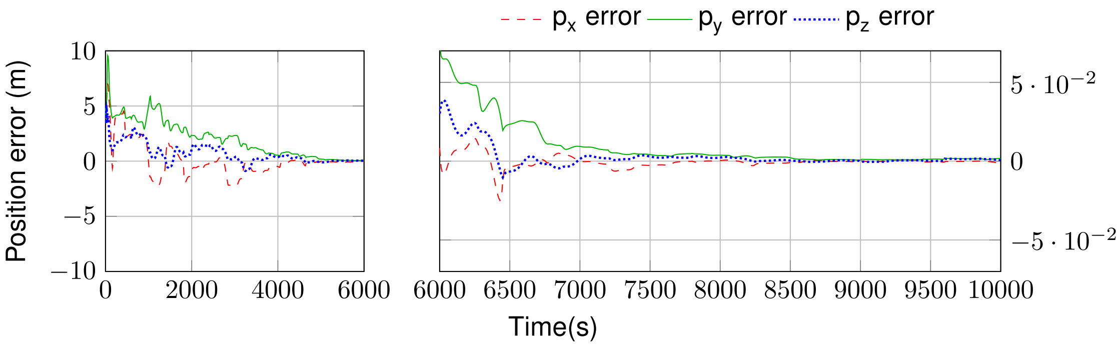

Figure 5.19: Position estimation error of the UKF in the absence of sensor noise: non-zero initial error.

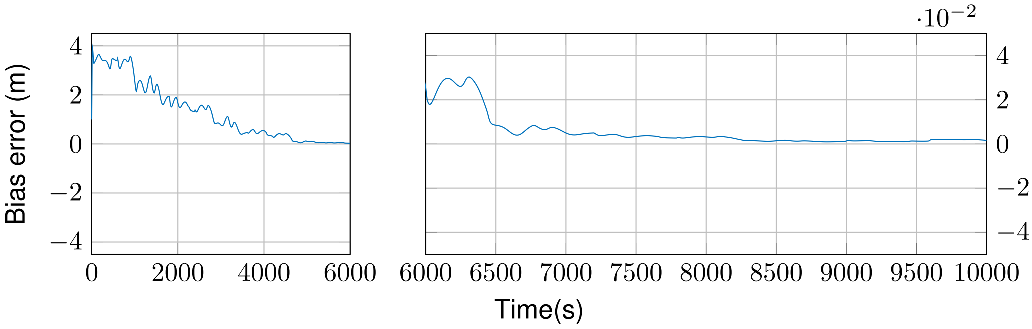

Figure 5.20: Bias estimation error of the UKF in the absence of sensor noise: non-zero initial error.

5.4.2.1.4 Non-zero initial error with added noise

For this experiment, the initial conditions are set to $\mathbf{x}(0) = \begin{bmatrix} -4 & -4 & -4 & 1 \end{bmatrix}^T$ $[\mathrm{m}]$ and the measured (noisy) values of the range and inputs are considered. The resulting plots for the position estimation error and bias estimation error are presented in Figures 5.21 and 5.22, respectively. The first starts by peaking and then converges to an interval between approximately $-0.2\,\mathrm{m}$ and $0.2\,\mathrm{m}$. The latter also shows a peak in the initial $200\,\mathrm{s}$ and then converges to a value close to zero, maintaining the slight estimation offset shown in previous experiments, with a final value of approximately $0.0113\,\mathrm{m}$.

From these results, one can attest to the relative robustness of the filter, since a convergence can be achieved from a non-zero initial error with added sensor noise. Once again, it is important to keep in mind the complexity of the problem and the heaviness of this method, when analysing the obtained errors and the convergence times.

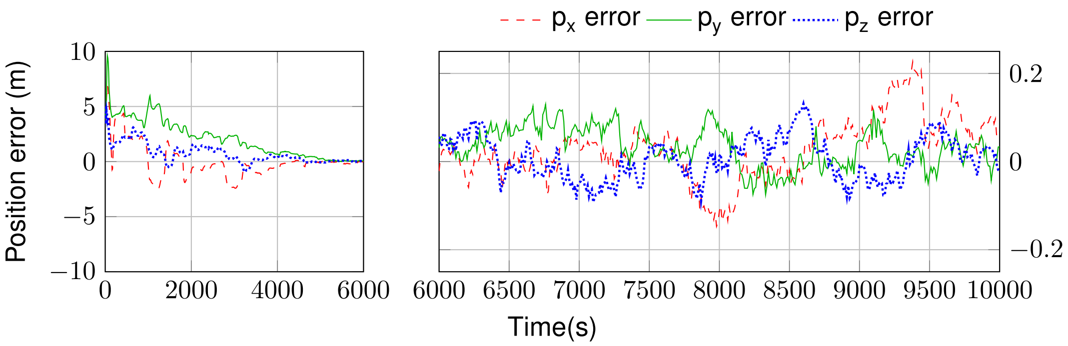

Figure 5.21: Position estimation error of the UKF in the presence of sensor noise: non-zero initial error.

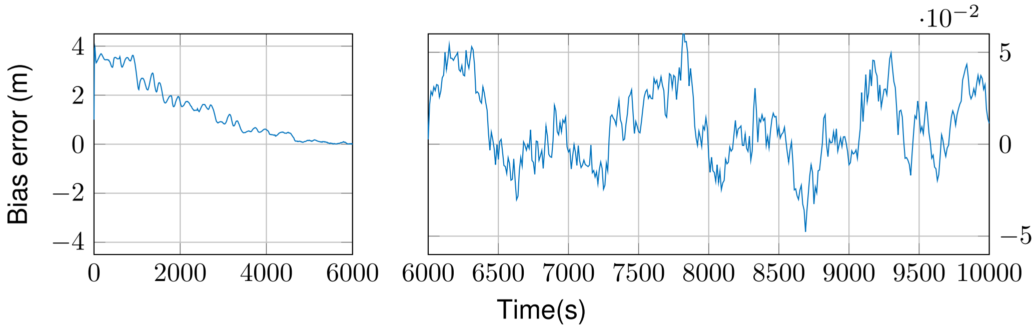

Figure 5.22: Bias estimation error of the UKF in the presence of sensor noise: non-zero initial error.

5.4.2.2 Failure of convergence of the UKF

The UKF also does not provide any convergence guarantees and so a test on its limitations is also performed. For this experiment, the initial conditions are set to $\mathbf{x}(0) = \begin{bmatrix} -24 & -14 & -24 & -2 \end{bmatrix}^T$ [m] and the measured (noisy) values of the range and inputs are considered. The resulting plots for the position estimation error and bias estimation error are presented in Figures 5.23 and 5.24, respectively. It is noticeable in both plots that the filter is not able to converge with the given conditions. Indeed, the error of p_x oscillates between $15\,\mathrm{m}$ and $24\,\mathrm{m}$ throughout the whole simulation, the error of p_y oscillates between $9\,\mathrm{m}$ and $15\,\mathrm{m}$, with a slight tendency to increase over the run time, and the error of p_z oscillates between $10\,\mathrm{m}$ and $16\,\mathrm{m}$, with a slight decline over time. As for the bias, its error maintains an oscillation between $19\,\mathrm{m}$ and $22\,\mathrm{m}$ throughout the simulation time, while slowly decreasing but never coming close to a convergence to zero.

It is important to note that the large offset applied to the correct initial conditions in order to show a case of filter non-convergence, is a few orders of magnitude smaller than the one used in the equivalent EKF experiment, presented in Section 5.4.1.2. Indeed, when the EKF is run with these initial conditions, it is able to converge, which shows that the UKF is less robust for the problem at hand. This behaviour is justified by the fact that the UKF is designed to be able to track highly nonlinear functions, whereas, the one presented in this work is mostly linear (as shown by matrices $\mathbf{A}$ and $\mathbf{B}$ being linear). The only non-linearity comes from the output expression, which makes the UKF a worse option for this problem and a more computationally heavy, and hence slower, one.

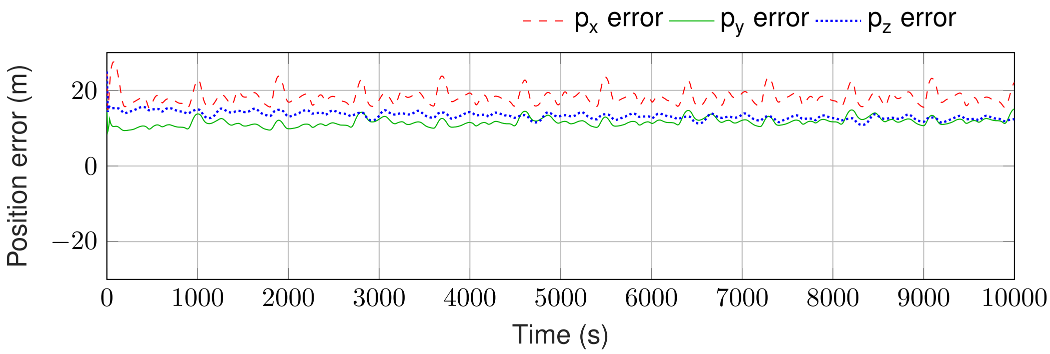

Figure 5.23: Position estimation error of the UKF: failure of convergence.

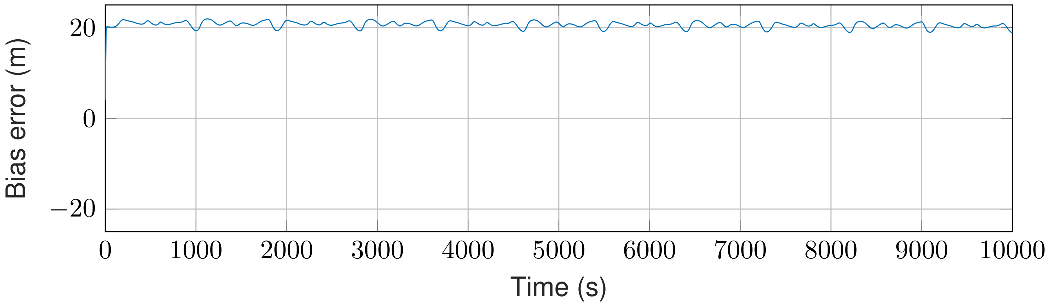

Figure 5.24: Bias estimation error of the UKF: failure of convergence.

5.4.3 LKF results with bottom-lock

Once again, the simulations in this section imply a prior run of the previously mentioned Simulink models, in order to obtain the inputs, ranges, and states necessary for the estimations and error calculations. For all, the initial covariance matrix is set to $\mathbf{P}(0) = \mathbf{I}_5$, and the noise covariance matrices are $\mathbf{Q} = \mathbf{diag}(10^{-4}\,\mathbf{I}_3, 10^{-4}\,\mathbf{I}_2)$ and $R = 10^{-2}$.

5.4.3.1 Filter convergence

The convergence of the filter is, once more, tested over the experiments mentioned in Section 5.4.1.1. The results for each set of conditions are shown in the following sections. For the sake of completeness, the estimation error of the additional state $z_3(k)$ is also presented in the final convergence experiment.

5.4.3.1.1 Correct initial conditions

For this experiment, the initial conditions are set to $\mathbf{z}(0) = \begin{bmatrix} 1 & 1 & 1 & 2 & r(0) \end{bmatrix}^T$ $[\mathrm{m}]$, where $r(0)$ is the initial true range value, i.e., the true values. The true values of the range and inputs are also considered. The resulting plots for the position estimation error and bias estimation error are presented in Figures 5.25 and 5.26, respectively. These show the expected outcomes, since the errors start and remain at very small values for all time, which can be taken as zero, given the scale of this problem, and prove that the designed filter does not have any propagation errors.

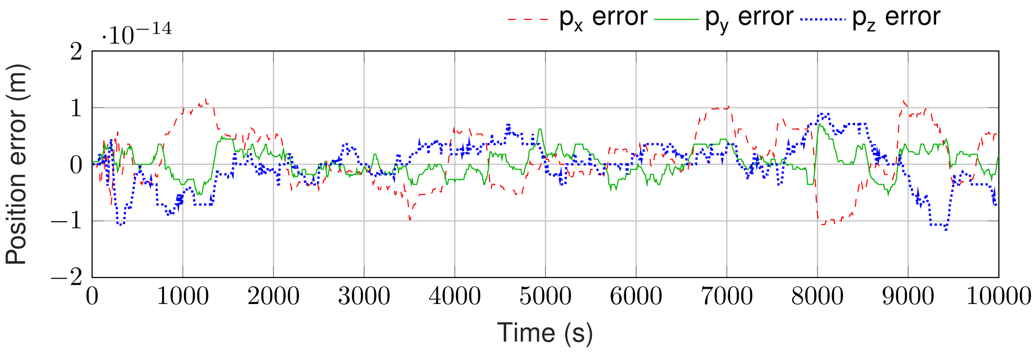

Figure 5.25: Position estimation error of the LKF in the absence of sensor noise: zero initial error.

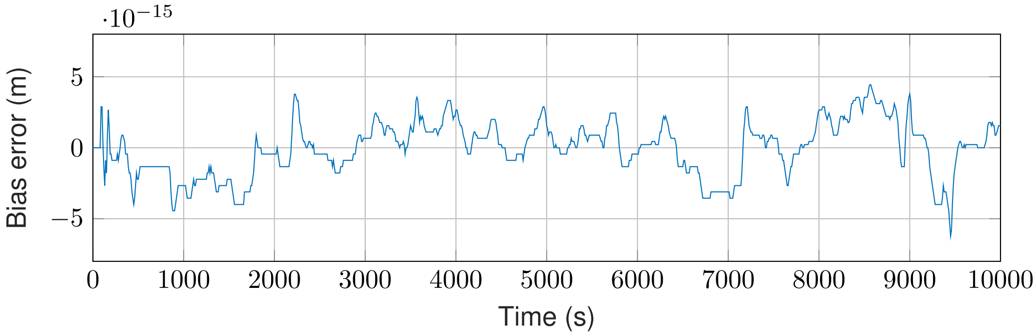

Figure 5.26: Bias estimation error of the LKF in the absence of sensor noise: zero initial error.

5.4.3.1.2 Correct initial conditions with added noise

For this experiment, the initial conditions are set to $\mathbf{z}(0) = \begin{bmatrix} 1 & 1 & 1 & 2 & r(0) \end{bmatrix}^T$ $[\mathrm{m}]$, i.e., the true values, and the measured (noisy) values of the range and inputs are considered. The position estimation error and bias estimation error are presented in Figures 5.27 and 5.28, respectively. The first shows initial transients between $-1\,\mathrm{m}$ and $1\,\mathrm{m}$, with a steady-state between $-0.2\,\mathrm{m}$ and $0.2\,\mathrm{m}$, like in Figures 5.7 and 5.17. The bias estimation error also shows an initial transient and holds a similar steady-state interval to previous experiments, however, this time the results are skewed toward negative values. The slight estimation offset still holds, with final value $b_{c_{error}} \approx -0.0264\,\mathrm{m}$. From the plots, one can see that the LKF has a much faster convergence than the previous two filters, because it is applied to a linear system and offers the guarantee of GES behaviour (see Section 5.4.3.2 for further details). This stability guarantee means that solutions not only converge, but in fact converge faster than or at least as fast as a particular exponentially decaying signal.

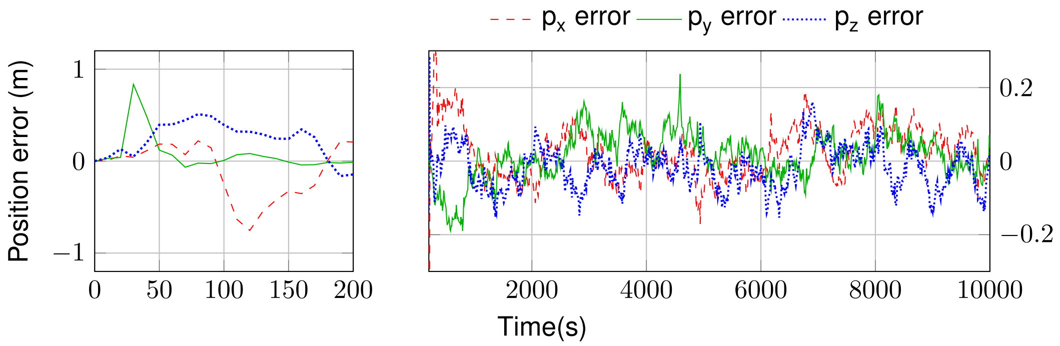

Figure 5.27: Position estimation error of the LKF in the presence of sensor noise: zero initial error.

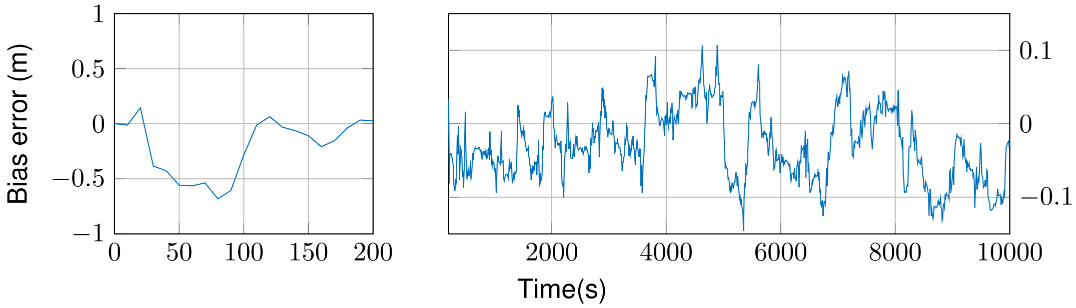

Figure 5.28: Bias estimation error of the LKF in the presence of sensor noise: zero initial error.

5.4.3.1.3 Non-zero initial error

For this experiment, the initial conditions are set to $\mathbf{z}(0) = \begin{bmatrix}-4 & -4 & -4 & 1 & r(0)\end{bmatrix}^T$ [m] and the true values of the range and inputs are considered. The resulting plots for the position estimation error and bias estimation error are presented in Figures 5.29 and 5.30, respectively. These show that the filter is very robust to changes in the initial estimate, since both the position error and the bias error converge to zero at around $200\,\mathrm{s}$. The final errors are $\mathbf{p}_{error} \approx \begin{bmatrix}3.55 & 2.55 & -7.11\end{bmatrix}^T \times 10^{-15}$ [m], and $b_{c_{error}} \approx 4.44 \times 10^{-16}\,\mathrm{m}$, which are great results, and considered zero with regard to the scale of the problem.

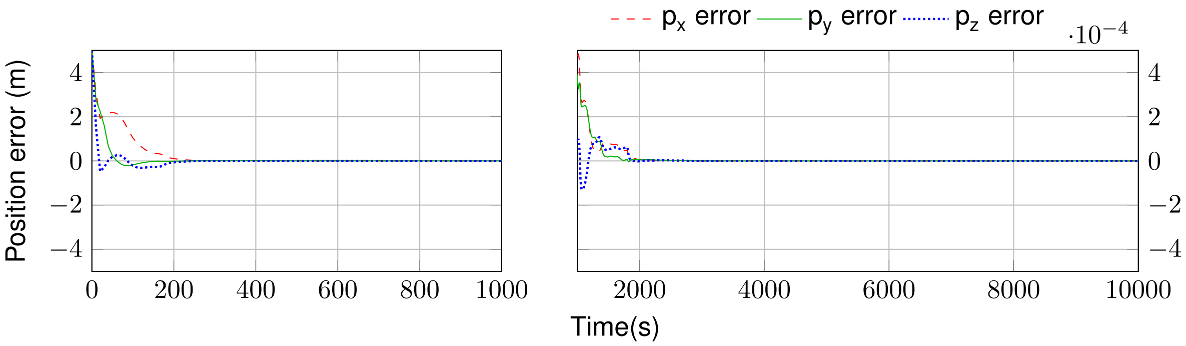

Figure 5.29: Position estimation error of the LKF in the absence of sensor noise: non-zero initial error.

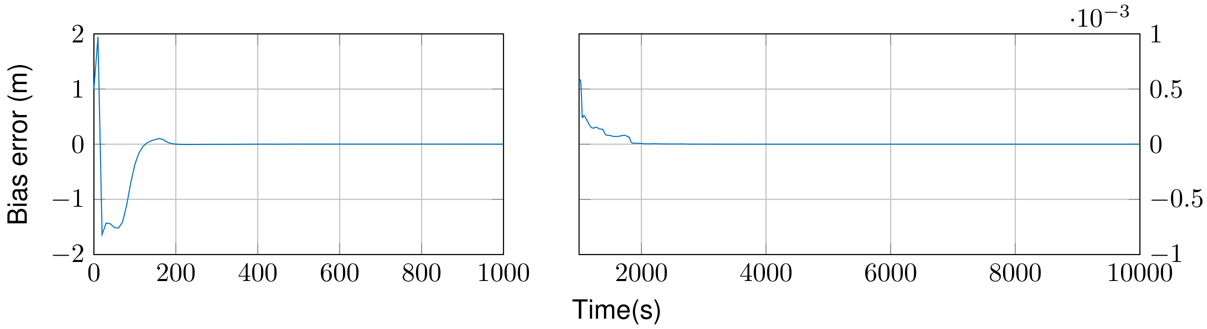

Figure 5.30: Bias estimation error of the LKF in the absence of sensor noise: non-zero initial error.

5.4.3.1.4 Non-zero initial error with added noise

For this experiment, the initial conditions are set to $\mathbf{z}(0) = \begin{bmatrix} -4 & -4 & -4 & 1 & r(0) \end{bmatrix}^T$ $[\mathrm{m}]$ and the measured (noisy) values of the range and inputs are considered. The resulting plots for the position estimation error and bias estimation error are presented in Figures 5.31 and 5.32, respectively. For the sake of completeness, the estimation error of the additional state $z_3(k)$ is also depicted in Fig. 5.33. All plots show higher transients in the initial $200\,\mathrm{s}$, converging to oscillations between $-0.2\,\mathrm{m}$ and $0.2\,\mathrm{m}$. The bias maintains the slight estimation offset, with a final value of approximately $-0.19\,\mathrm{m}$. Because there is no initial error added to the range, the plot starts at zero. From these results, one can attest to the robustness of the filter, since a very fast convergence can be achieved under these conditions.

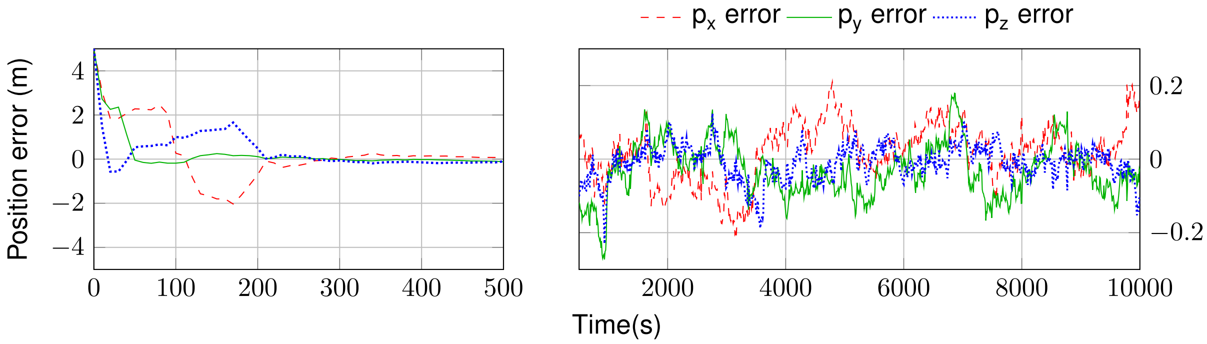

Figure 5.31: Position estimation error of the LKF in the presence of sensor noise: non-zero initial error.

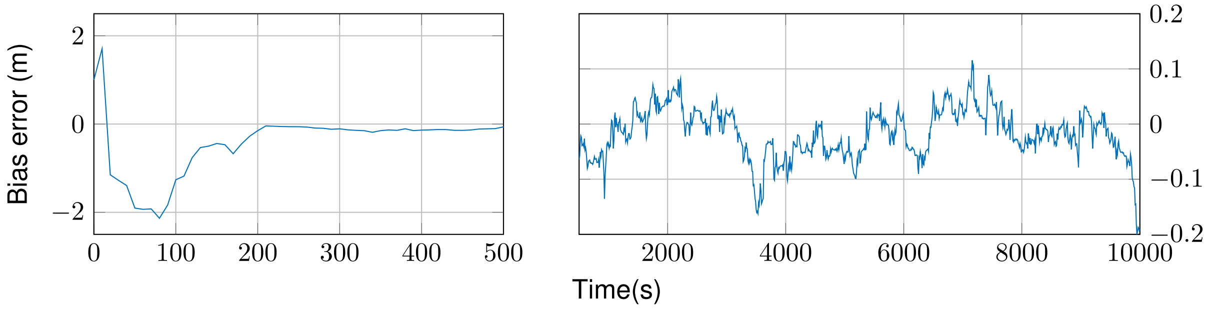

Figure 5.32: Bias estimation error of the LKF in the presence of sensor noise: non-zero initial error.

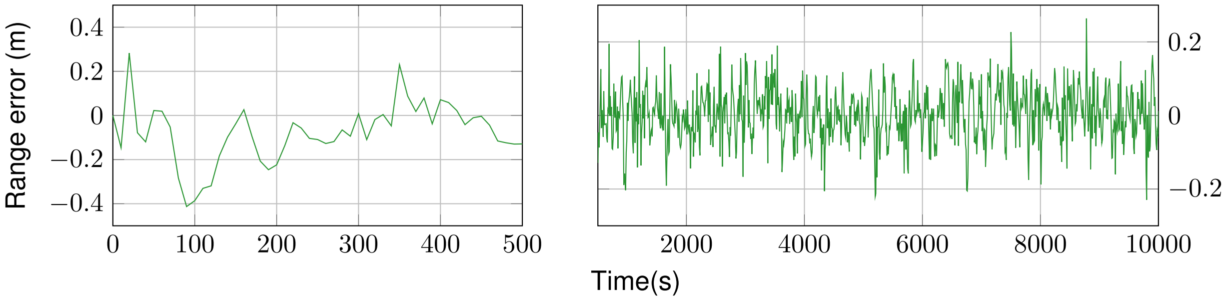

Figure 5.33: Range estimation error of the LKF in the presence of sensor noise: non-zero initial error.

5.4.3.2 Considerations on LKF convergence

The filter robustness is confirmed via the experiments in Section 5.4.3.1.4. However, it is important to note a general characteristic of this proposed solution: because this filter can be proven to be global exponentially stable (GES), as mentioned in [23], its convergence is guaranteed. For this reason, no examples of filter non-convergence are provided, however, the filter is run with the same conditions presented in Section 5.4.1.2 and in Section 5.4.2.2, to prove that the results obtained are significantly better. In both cases, the state relates to the pseudo-range measurement is initialized with $r(0)$, in order to provide comparable results.

5.4.3.2.1 Comparison with EKF non-convergence

The LKF is run with sensor noise with covariance matrices $\mathbf{Q} = \mathbf{diag}(10^{-4}\,\mathbf{I}_3, 10^{-4}\,\mathbf{I}_2)$ and $R = 10^{-2}$. The initial conditions are set to $\mathbf{P}(0) = \mathbf{I}_5$ and $\mathbf{z}(0) = \begin{bmatrix} -999 & -699 & -999 & -498 & r(0) \end{bmatrix}^T$ [m]. The resulting plots for the position estimation error and bias estimation error are presented in Figures 5.34 and 5.35, respectively. In this case, because the initial error is significantly larger than the one used in Section 5.4.3.1.4, it is noticeable that the convergence time is about four times longer. However, similar steady-state oscillations are obtained both for the position and the bias errors, proving that the LKF for the augmented state is a better solution than the EKF applied to the original state definition.

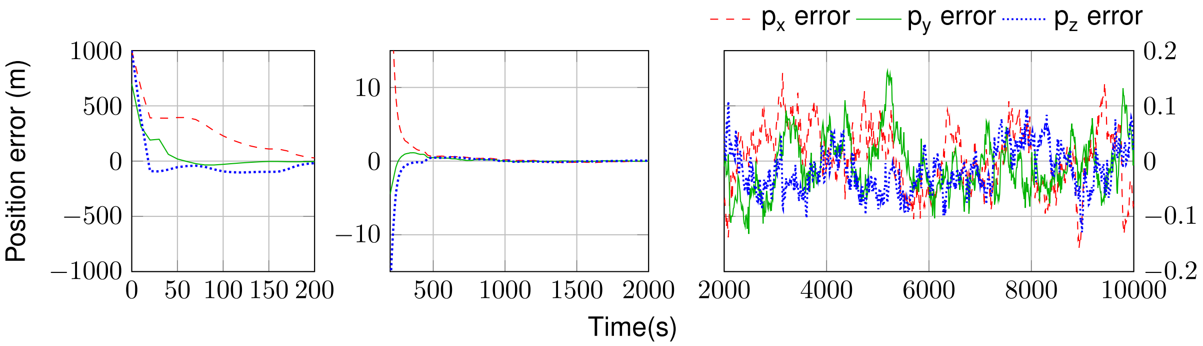

Figure 5.34: LKF convergence for an initial condition that leads to failure of the EKF: position estimation error.

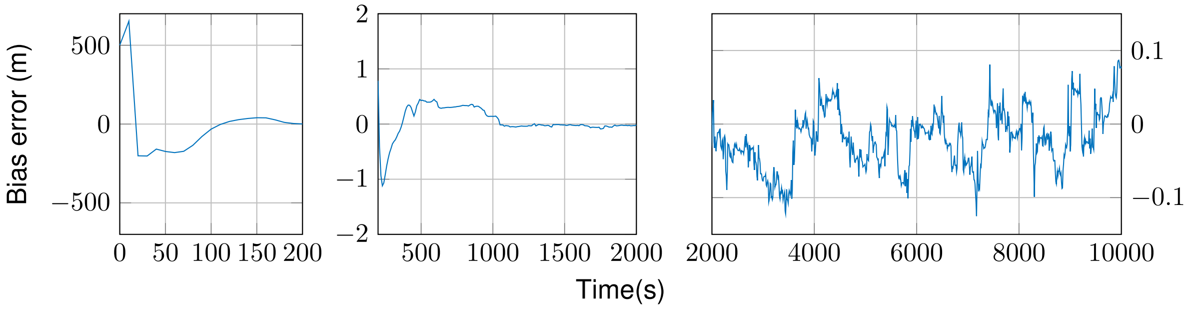

Figure 5.35: LKF convergence for an initial condition that leads to failure of the EKF: bias estimation error.

5.4.3.2.2 Comparison with UKF non-convergence

Finally, the LKF is run with sensor noise with covariance matrices $\mathbf{Q} = \mathbf{diag}(10^{-4}\,\mathbf{I}_3, 10^{-4}\,\mathbf{I}_2)$ and $R = 10^{-2}$. The initial conditions are set to $\mathbf{P}(0) = \mathbf{I}_5$ and $\mathbf{z}(0) = \begin{bmatrix}-24 & -14 & -24 & -2 & r(0)\end{bmatrix}^T$ $[\mathrm{m}]$. The resulting plots for the position estimation error and bias estimation error are presented in Figures 5.36 and 5.37, respectively. Because the UKF becomes unstable for initial errors that are only a few times larger than the ones used in Section 5.4.3.1.4, the convergence time is very similar in this experiment, in comparison with the one in that section. Once again, similar steady-state oscillations are obtained both for the position and the bias errors, proving that the LKF of the augmented state is a better solution than the UKF applied to the original state definition, which seems to indicate that the LKF is the better solution overall. Further analysis is performed through the Monte Carlo experiments in Chapter 8.

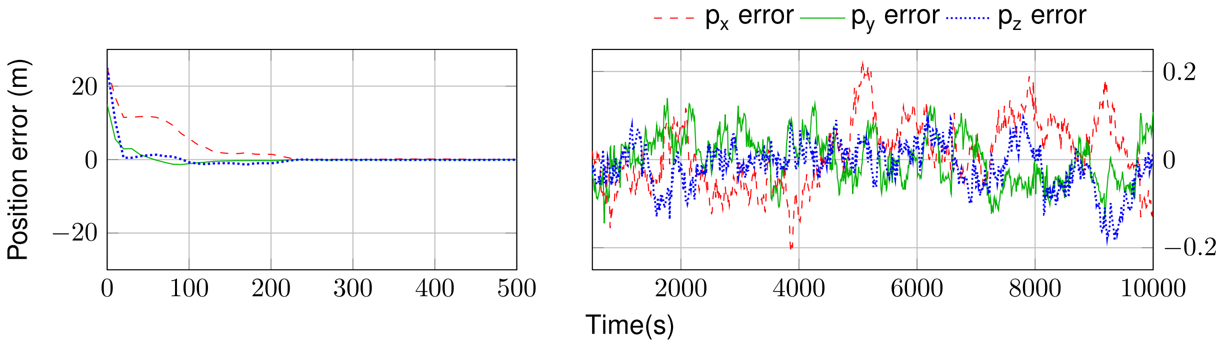

Figure 5.36: LKF convergence for an initial condition that leads to failure of the UKF: position estimation error.

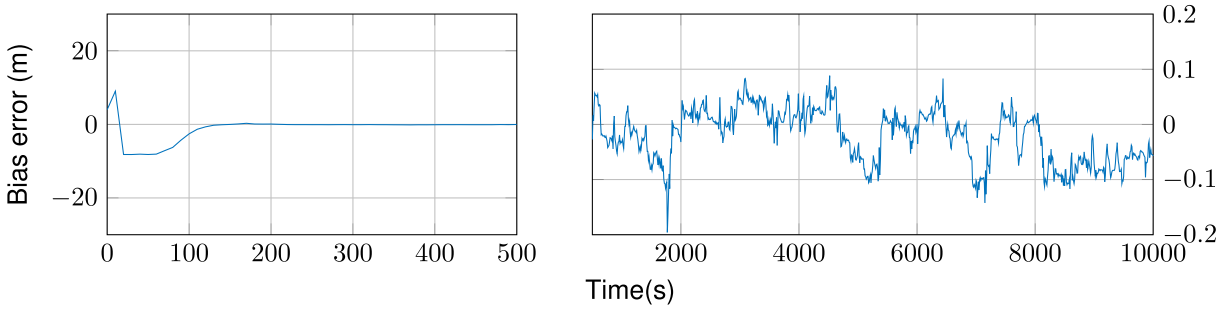

Figure 5.37: LKF convergence for an initial condition that leads to failure of the UKF: bias estimation error.

6

Proposed solution without bottom-lock

6.1 System dynamics and models

For the situation without bottom-lock, recall the discrete-time nonlinear system (3.10). For simulation purposes, the same sampling and final simulation times mentioned in Section 5.1 were considered. In order to maintain a similar trajectory to the one shown in Fig. 5.4, which gives a limited interval for the range values, one can allow the velocity vector to be set to any desired value, so long as that value is then compensated with the system entries. This gives a system in which the velocity of the vehicle relative to the fluid is simply the compensation of the velocity of the fluid. The Simulink model that provides the system input $\mathbf{u}$, follows, in this case,

$$\mathbf{u}(k) = \begin{bmatrix} \cos\left(\frac{2\pi k}{30}\right) - Tv_{fx}(k) \\ 1.7\cos\left(\frac{2\pi k}{20} + \frac{\pi}{6}\right) - Tv_{fy}(k) \\ 2\cos\left(\frac{2\pi k}{45} + \frac{\pi}{9}\right) - Tv_{fz}(k) \end{bmatrix}.$$

Once again, this choice of input provides a sufficiently rich trajectory of the agent. The second Simulink model is similar to the one presented in Fig. 5.1, this time using the state and input matrices in (3.11),

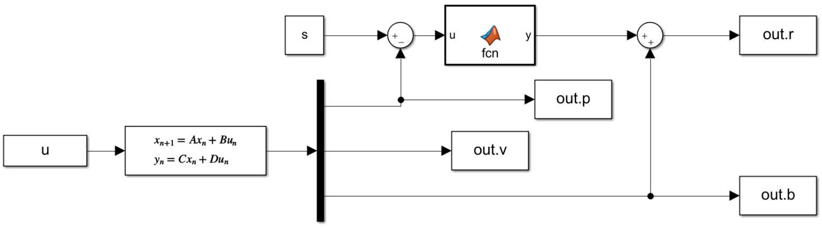

Figure 6.1: Simulink model to simulate the nominal system dynamics without bottom-lock.

and still setting the auxiliary matrices to $\mathbf{C} = \mathbf{I}$ and $\mathbf{D} = \mathbf{0}$. The remaining initial conditions are set to

$$\mathbf{s} = \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} \,[\mathrm{m}], \quad \mathbf{p}(0) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \,[\mathrm{m}], \quad \mathbf{v}_f(0) = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix} \,[\mathrm{m/s}], \quad \text{and} \quad b_c(0) = 2\,\mathrm{m}.$$

The initial state estimate for the system is, then, given by $\mathbf{x}(0) = \begin{bmatrix} 1 & 1 & 1 & 0.1 & 0.1 & 0.1 & 2 \end{bmatrix}^T$. The resulting outputs are the same as before: the true value of the bias over the simulation time, represented in Fig. 5.2, the true value of the pseudo-range measurements over time, represented in Fig. 5.3 and the trajectory of the agent, depicted in Fig. 5.4. In this case, the model also outputs the true value of the velocity over time. This new model is presented in Fig. 6.1.

6.2 Noise description

Once again, recall the system dynamics presented in (3.10). The more accurate representation of this problem, taking into account the sensor noises mentioned in Section 5.2, is then

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}[\mathbf{u}(k) + \mathbf{m}(k)] \\ r(k) = \|\mathbf{s} - \mathbf{x}_1(k)\| + x_3(k) + n(k) \end{cases} \tag{6.1}$$

or, equivalently,

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{w}(k) \\ r(k) = \|\mathbf{s} - \mathbf{x}_1(k)\| + x_3(k) + n(k) \end{cases} \tag{6.2}$$

where $n(k) \in \mathbb{R}$ is the Gaussian observation noise, $n(k) \sim \mathcal{N}(0, R(k))$, and $\mathbf{m}(k) \in \mathbb{R}^7$ is the Gaussian process noise, $\mathbf{m}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{M}(k))$, associated with the measurement of the vehicle position displacements, $\mathbf{u}(k)$. Once again, because of the values of matrix $\mathbf{B}$, the values of vector $\mathbf{w}(k)$ come as $\mathbf{w}(k) = \begin{bmatrix} \mathbf{m}^T(k) & 0 & 0 & 0 & 0 \end{bmatrix}^T$, therefore, $\mathbf{w}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(k))$, where $\mathbf{Q}(k) = \mathbf{diag}(\mathbf{M}_{1,1}(k), \mathbf{M}_{2,2}(k), \mathbf{M}_{3,3}(k), 0, 0, 0, 0)$. Recall that, in practice, $\mathbf{Q}_{4,4}(k)$, $\mathbf{Q}_{5,5}(k)$, $\mathbf{Q}_{6,6}(k)$, and $\mathbf{Q}_{7,7}(k)$ will always have to be set to small yet non-zero values.

6.2.1 Noise controllability

Once again, the system at hand is described by a state equation that is linear time-invariant, which allows for the controllability of the system to be analysed via the column rank of matrix $\mathbf{C}_o$, given by

$$\mathbf{C}_o = \begin{bmatrix} \mathbf{B} & \mathbf{A}\mathbf{B} & \mathbf{A}^2\mathbf{B} & \mathbf{A}^3\mathbf{B} & \mathbf{A}^4\mathbf{B} & \mathbf{A}^5\mathbf{B} & \mathbf{A}^6\mathbf{B} \end{bmatrix}.$$

In this case, the state vector is of size $n = 7$ and the column rank of $\mathbf{C}_o$ is of only 3, which shows that this system is also not fully controllable by its input $\mathbf{u}(k)$. The noise controllability analysis is as before, which, once again, forces all the diagonal entries of matrix $\mathbf{Q}$ to be set to low, yet non-zero values, to ensure that the system is noise controllable.

6.3 Filter parameters and design

Like before, in order to design an estimator for the system described by (3.10), an EKF was designed and tested, with and without sensor noises, and for different initial conditions. This represents an initial approach to the problem without bottom-lock. Because the UKF presented unstable responses even for the system with bottom-lock, it was not considered a good approach for this system, which represents an even more complex problem.

Once again, for the simulations in which no noise is considered, $\mathbf{r}$ and $\mathbf{u}$ are simply the real range and input values, whereas, when adding sensor noise, $\mathbf{r}$ and $\mathbf{u}$ are affected by additive white Gaussian noises, given by $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ and $n \sim \mathcal{N}(0, R)$, respectively. $\mathbf{Q}$ is a diagonal matrix, $\mathbf{Q} = \mathbf{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_{vx}^2, \sigma_{vy}^2, \sigma_{vz}^2, \sigma_b^2)$, where σ_{vx}^2, σ_{vy}^2, σ_{vz}^2, and σ_b^2 are the small values discussed in Section 6.2.1, and σ_x^2, σ_y^2, σ_z^2 are the variances of the measurements of the system inputs, which correspond to the variances of the vehicle position displacement components in the x, y, and z axes, respectively. $R = \sigma^2$, where σ^2 is the variance of the range measurement. Once again, because of the matrix inversion in (4.4), R cannot be null.

The simulation results of the designed EKF are shown in Section 6.4. The Jacobian matrices mentioned in (4.2) and (4.3), are, in the case of the system at hand,

$$\mathbf{F} = \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{H}(k+1) = \begin{bmatrix} \frac{\hat{p}_x(k+1|k) - \mathrm{s}_x}{\|\mathbf{s} - \hat{\mathbf{p}}(k+1|k)\|} & \frac{\hat{p}_y(k+1|k) - \mathrm{s}_y}{\|\mathbf{s} - \hat{\mathbf{p}}(k+1|k)\|} & \frac{\hat{p}_z(k+1|k) - \mathrm{s}_z}{\|\mathbf{s} - \hat{\mathbf{p}}(k+1|k)\|} & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{6.3}$$

Matrix $\mathbf{F}$ can be obtained from the fact that the system state equation is linear in $\mathbf{x}(k)$. As for matrix $\mathbf{H}(k{+}1)$, the derivative of $r(k{+}1)$ with respect to each of the predicted state variables must be calculated, and so

$$\mathbf{H}(k+1) = \frac{\partial r(k+1)}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}(k+1|k)},$$

which, in this case, gives

$$\mathbf{H}(k+1) = \begin{bmatrix} \frac{\partial r(k+1)}{\partial \hat{p}_x(k+1|k)} & \frac{\partial r(k+1)}{\partial \hat{p}_y(k+1|k)} & \frac{\partial r(k+1)}{\partial \hat{p}_z(k+1|k)} & \frac{\partial r(k+1)}{\partial \hat{v}_{fx}(k+1|k)} & \frac{\partial r(k+1)}{\partial \hat{v}_{fy}(k+1|k)} & \frac{\partial r(k+1)}{\partial \hat{v}_{fz}(k+1|k)} & \frac{\partial r(k+1)}{\partial \hat{b}_c(k+1|k)} \end{bmatrix}.$$

From the pseudo-range expression in (3.2), it is possible to conclude that $\dfrac{\partial r(k+1)}{\partial \hat{b}_c(k+1|k)} = 1$, and that $\dfrac{\partial r(k+1)}{\partial \hat{v}_{fx}(k+1|k)} = \dfrac{\partial r(k+1)}{\partial \hat{v}_{fy}(k+1|k)} = \dfrac{\partial r(k+1)}{\partial \hat{v}_{fz}(k+1|k)} = 0$. The calculations regarding the derivative with respect to $\hat{p}_x(k+1|k)$, $\hat{p}_y(k+1|k)$ and $\hat{p}_z(k+1|k)$ are the same as the ones presented in Section 5.3.

6.4 Simulation results without bottom-lock

All the simulations in this section imply a prior run of the previously mentioned Simulink models, in order to obtain the inputs, ranges, and states necessary for the computation of the state estimates and estimation errors. For all, the initial covariance matrix is set to $\mathbf{P}(0) = \mathbf{I}_7$ and the noise covariance matrices are given by $\mathbf{Q} = \mathbf{diag}(10^{-4}\,\mathbf{I}_3, 10^{-6}\,\mathbf{I}_4)$ and $R = 10^{-4}$.

The convergence of the filter was tested over the experiments mentioned in Section 5.4.1.1. It was proven that the filter does not have propagation errors, that it is robust to noise, and that it is robust to changes in the initial estimate, in experiments equivalent to the ones presented in Sections 5.4.1.1.1, 5.4.1.1.2, and 5.4.1.1.3, respectively. For brevity, only the last experiment, showing the filter response to non-zero initial error with added noise is shown in Section 6.4.1. A set of initial conditions that show failure of convergence are also presented in Section 6.4.2.

6.4.1 Non-zero initial error with added noise

For this experiment, the initial conditions are set to $\mathbf{x}_1(0) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ [m], $\mathbf{x}_2(0) = \begin{bmatrix} 0.03 & 0.03 & 0.03 \end{bmatrix}^T$ [m/s], and $x_3(0) = 1.7\,\mathrm{m}$. The measured (noisy) values of the range and inputs are considered, and the resulting plots for the position estimation error, velocity estimation error, and bias estimation error are presented in Figures 6.2, 6.3, and 6.4, respectively.

The plots show dampened oscillations for the position and velocity errors, and a more asymptotic convergence for the bias error. The final estimates give $\mathbf{p}_{error} \approx \begin{bmatrix} -0.308 & -0.123 & 0.248 \end{bmatrix}^T$ [m], $\mathbf{v}_{error} \approx \begin{bmatrix} 0.173 & -0.183 & 1.552 \end{bmatrix}^T \times 10^{-3}\,[\mathrm{m/s}]$, and $b_{c_{error}} \approx -0.059\,\mathrm{m}$. These errors represent only $2,95\%$ and less than $1,6\%$ of the nominal values for the bias and the velocity, respectively. Even though the plots present position errors about twice as large as the ones obtained in Section 5.4.1.1.4 and bias errors over ten times as large, they are still considered very good results, because the problem is significantly harder in the case of trying to estimate seven different variables from only one measurement, in what is an inherently nonlinear problem. Moreover, one can attest to the relative robustness of the filter, since a convergence is clearly achieved. It should be noted that the higher error present in the position estimates is to be expected because these variables can vary rapidly, whereas the velocity and bias variables are constant.

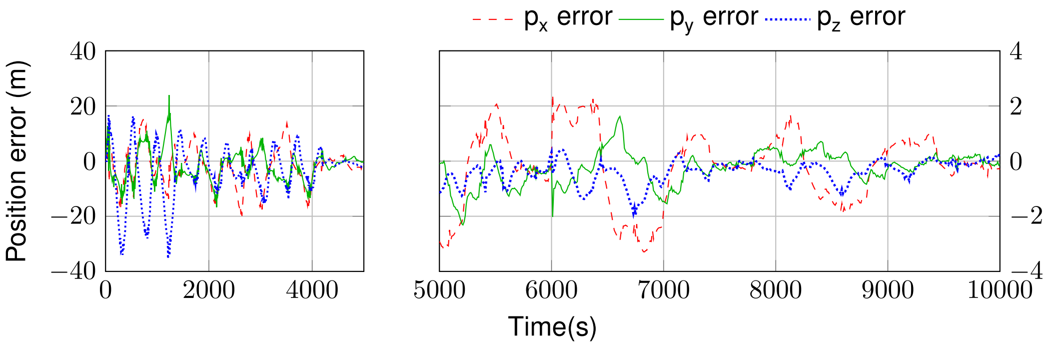

Figure 6.2: Position estimation error of the EKF without bottom-lock in the presence of sensor noise: non-zero initial error.

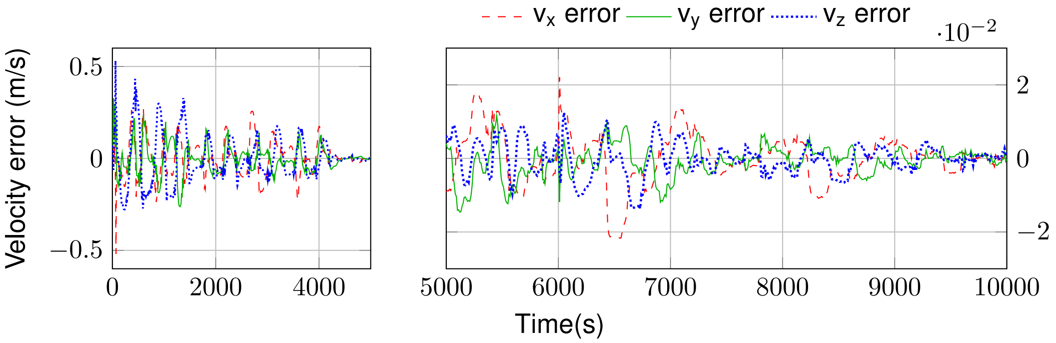

Figure 6.3: Velocity estimation error of the EKF without bottom-lock in the presence of sensor noise: non-zero initial error.

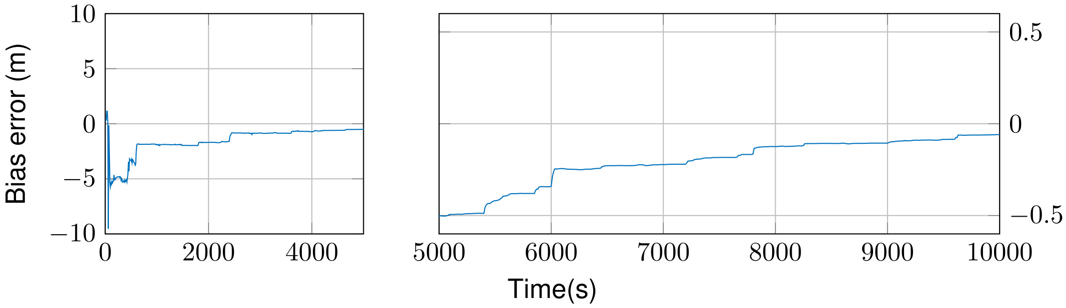

Figure 6.4: Bias estimation error of the EKF without bottom-lock in the presence of sensor noise: non-zero initial error.

6.4.2 Failure of convergence of the EKF

As mentioned in Section 5.4.1.2, the EKF does not provide any guarantees for convergence. To exemplify a case in which a convergence in the given time limit is not achievable, a run with sensor noise following the same distribution as before and initial conditions as the sum of the true value plus a large offset is performed. In this case, the conditions are set to $\mathbf{x}_1(0) = \begin{bmatrix} -4999 & -3999 & -4999 \end{bmatrix}^T$ [m], $\mathbf{x}_2(0) = \begin{bmatrix} -1.9 & -0.9 & -1.9 \end{bmatrix}^T$ [m/s], and $x_3(0) = -4498\,\mathrm{m}$. The resulting plots for the position estimation error, velocity estimation error and bias estimation error are presented in Figures 6.5, 6.6, and 6.7, respectively. The first panel of each plot seems to indicate a convergence, however, one can see from the second panels that both the position and the velocity errors oscillate about zero, but with values that are very large when compared with the nominal values of these variables. Further proof of the failure of convergence comes from the bias error plot, which slowly decreases but never gets close to converging.

The filter was tested with the same $\mathbf{Q}$ and R matrices as in Section 5.4.1 throughout all experiments, however that choice of parameters showed poorer results overall. Instead, the chosen values presented in Section 6.4 prove to make the filter fairly resistant to initial estimation errors, which justifies the need for a much larger offset to clearly show an example of EKF failure of convergence, in comparison with the equivalent experiment, shown in Section 5.4.1.2. In testing the failure of convergence, changes in the initial bias estimation proved to affect the outcome more than changes in any of the other variables.

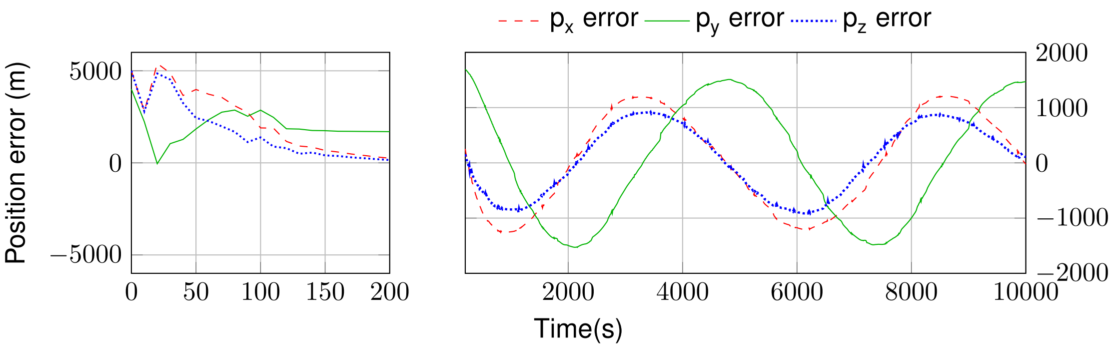

Figure 6.5: Position estimation error of the EKF without bottom-lock: failure of convergence.

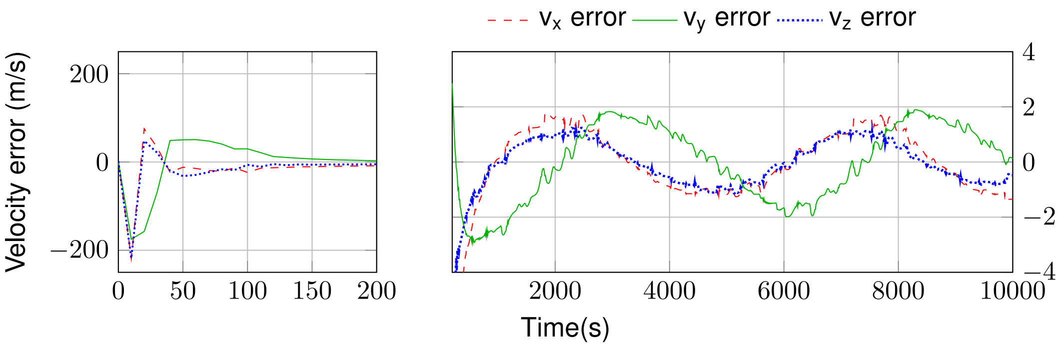

Figure 6.6: Velocity estimation error of the EKF without bottom-lock: failure of convergence.

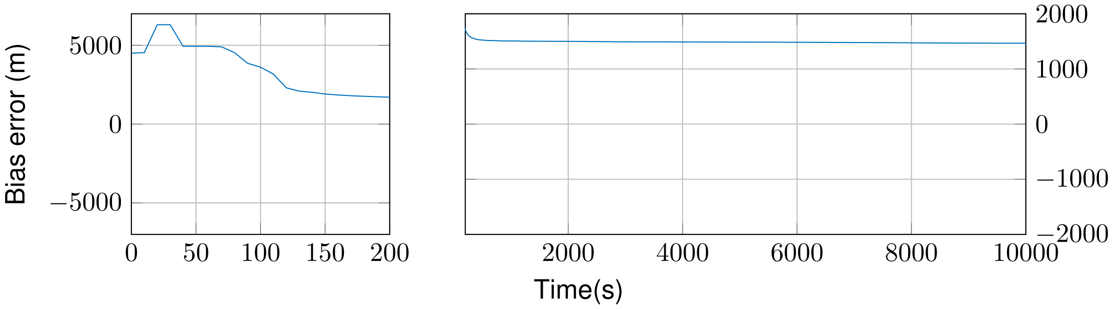

Figure 6.7: Bias estimation error of the EKF without bottom-lock: failure of convergence.

7

Bayesian Crámer-Rao bound

The BCRB is a lower bound on the variance of unbiased estimators and it states that the variance of any unbiased estimator is at least as high as the inverse of the Fisher information. An unbiased estimator that reaches this lower bound is said to be fully efficient and, generally, the closer to the bound, the more efficient the estimator.

The use of the BCRB in the analysis of the results of the designed filters, in spite of the algorithms potentially providing biased estimations, is justified. Indeed, the BCRB can also provide a bound for the variance of biased estimators, with the particularity of potentially finding points where the resulting variance falls below the theoretical BCRB. This means the bound can serve as a term of comparison, even if it is not actually a strictly lower bound for biased estimators.

Because of the inherent process and measurement noises, the systems considered in this work are stochastic, since they do not always produce the same output for a given input. There is still no general BCRB computation formula for this type of system, however, there is a solution for systems of the form

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{w}(k) \\ \mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k)) + \mathbf{n}(k) \end{cases},$$

where $\mathbf{w}$ is the process noise, which follows a Gaussian distribution, $\mathbf{w}(k) \sim \mathcal{N}(\mathbf{0},\, \mathbf{Q}(k))$, and $\mathbf{n}$ is the observation noise which follows the Gaussian distribution, $\mathbf{n}(k) \sim \mathcal{N}(\mathbf{0},\, \mathbf{R}(k))$. Because the analysed systems fall in this category, since they are of the form

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{w}(k) \\ \mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k)) + \mathbf{n}(k) \end{cases},$$

the computations for the BCRB are possible. They are similar to the ones made for the EKF, only this time the Jacobian of the output matrix is calculated at the true state. The information matrix $\mathbf{J}(k)$ is given by the recursion

$$\mathbf{J}(k+1) = (\mathbf{Q}(k) + \mathbf{F}(k)\mathbf{J}^{-1}(k)\mathbf{F}^T(k))^{-1} + \mathbf{P}_m(k+1),$$

where $\mathbf{F}$ is the Jacobian of $\mathbf{f}$, calculated as in (4.2). $\mathbf{P}_m(k+1)$ is given by

$$\mathbf{P}_m(k+1) = \boldsymbol{E}_{\mathbf{x}(k+1)}\{\mathbf{H}^T(\mathbf{x}(k+1))\mathbf{Q}^{-1}(k+1)\mathbf{H}^T(\mathbf{x}(k+1)))\},$$

and $\mathbf{H}(k+1)$ is the Jacobian of $\mathbf{h}$ calculated as

$$\mathbf{H}(k+1) = \frac{\partial \mathbf{h}(k+1)}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}(k+1)}.$$

To allow for the evaluation of the performance of the estimator given the specific structure of a particular problem, one can perform the evaluation along nominal state trajectories, $\bar{\mathbf{x}}(k)$, for which the term $\mathbf{P}_m(k+1)$ can be simplified as

$$\mathbf{P}_m(k+1) = \mathbf{H}^T(\bar{\mathbf{x}}(k+1))\mathbf{Q}^{-1}(k+1)\mathbf{H}^T(\bar{\mathbf{x}}(k+1))).$$

Finally, the BCRB is given by $\mathbf{P}_L(k) = \mathbf{J}(k)^{-1}$. Since this is a measure regularly used for comparison with root mean square error (RMSE) values, one can take the square root of the diagonal entries of $\mathbf{P}_L$ to obtain comparable results. For the systems at hand, $\sqrt{\mathbf{P}_{L_{1,1}}}$ would give the desired BCRB value related to position p_x and used for comparison with RMSE or other quantities related to the standard deviation rather than the covariance.

7.1 BCRB for the system with bottom-lock

The computations of the Jacobians are similar to the ones presented in 5.3 and yield

$$\mathbf{F} = \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } \mathbf{H}(k+1) = \begin{bmatrix} \frac{p_x(k+1)-s_x}{\|\mathbf{s}-\mathbf{p}(k+1)\|} & \frac{p_y(k+1)-\ s_y}{\|\mathbf{s}-\mathbf{p}(k+1)\|} & \frac{p_z(k+1)-s_z}{\|\mathbf{s}-\mathbf{p}(k+1)\|} & 1 \end{bmatrix}.$$

Note that this is also the result used for the LKF, since this analysis is related to the original system and not the derived augmented one.

7.2 BCRB for system without bottom-lock

The computations of the Jacobians are similar to the ones presented in 6.3 and yield

$$\mathbf{F} = \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{H}(k+1) = \begin{bmatrix} \frac{\hat{p}_x(k+1|k)-\mathrm{s}_x}{\|\mathbf{s}-\hat{\mathbf{p}}(k+1|k)\|} & \frac{\hat{p}_y(k+1|k)-\ \mathrm{s}_y}{\|\mathbf{s}-\hat{\mathbf{p}}(k+1|k)\|} & \frac{\hat{p}_z(k+1|k)-\mathrm{s}_z}{\|\mathbf{s}-\hat{\mathbf{p}}(k+1|k)\|} & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{7.1}$$

8

Monte Carlo simulations with bottom-lock

Monte Carlo simulations are commonly used in engineering, mathematics, and physics, with different purposes and outcomes. As it relates to the problem at hand, this method is useful to obtain statistical properties by using repeated sampling. As the KF response depends greatly on a number of parameters, to have a better sense of its behaviour, one can sample a set universe of possible values for said parameters and run the algorithm. The resulting outputs can be used for statistical analysis.

8.1 EKF with bottom-lock Monte Carlo simulations

For each Monte Carlo run, the values of $\mathbf{u}_{\text{measured}}(k)$ and $r_{\text{measured}}(k)$ are generated by adding white Gaussian noise to the true values of the system inputs and range measurements. The process noise is, then, centered around the true values, with a chosen covariance matrix $\mathbf{Q} = \mathbf{diag}(10^{-4}\,\mathbf{I}_3, 10^{-4})$. The observation noise is also centered around the true values, with covariance matrix $R = 10^{-2}$. At each run, an initial condition is also sampled from the Gaussian distribution with zero mean, i.e., centered around the true initial conditions, with covariance matrix $\mathbf{P} = \mathbf{diag}(100\,\mathbf{I}_3, 25)$, which is also the given initial covariance matrix of the filter.

For the purposes of this experiment, 10000 runs were performed and a threshold for convergence was set to $1\,\mathrm{m}$ for the positions and $0.3\,\mathrm{m}$ for the bias. This means that if the absolute value of the final state estimate errors are above this value, the filter is said to not converge. These values were chosen empirically, but can be considered to provide good outcomes, since all 10000 runs converged.

The average error over time for each of the state variables is presented in Figures 8.1 and 8.2. From these, one can verify that during the initial transients a much higher error is present, with the values growing smaller for larger timestamps. The errors in steady-state stay between $-3 \times 10^{-3}\,\mathrm{m}$ and $3 \times 10^{-3}\,\mathrm{m}$ for the position coordinates and also reach a very low value for the bias variable (around $8.2065 \times 10^{-4}\,\mathrm{m}$ at the final time). The final bias error can be disregarded when compared to the true value of the bias term (indeed, since the nominal value is $2\,\mathrm{m}$, the error presented in the plot represents only around 0.04% at the final simulation time and around 0.11% at the last peak, shown at $t = 9600\,\mathrm{s}$).

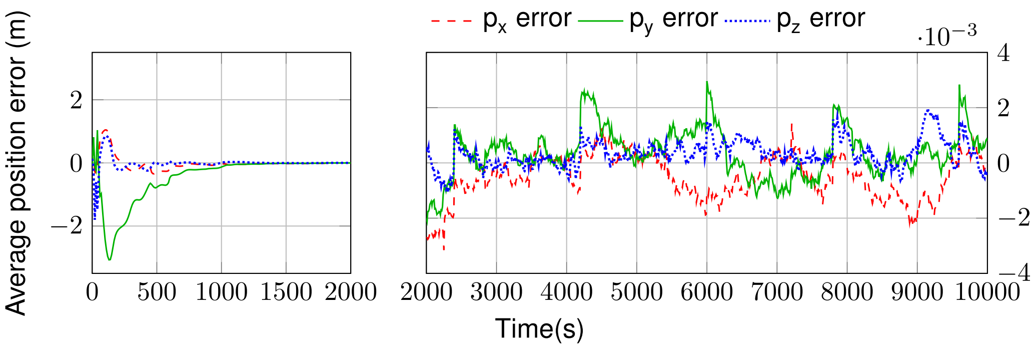

Figure 8.1: Position estimation average error of the EKF.

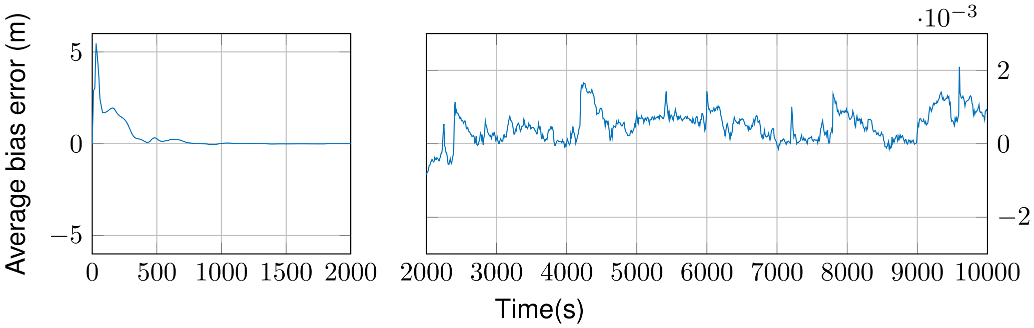

Figure 8.2: Bias estimation average error of the EKF.

The standard deviation for each state variable is given by the square root of its corresponding diagonal element of the average covariance matrix, i.e., for p_x, the standard deviation is given by $\sqrt{\mathbf{P_{1,1}}}$, where $\mathbf{P}$ is the average covariance matrix of the filter over all runs that converge. Because this value tends to overestimate the performance of the filter, it is interesting to compare the internal perception of convergence with the actual outcomes. Therefore, a comparison between the EKF standard deviation, the BCRB, and the RMSE obtained for each of the state variables is shown in Figures 8.3 - 8.6. For these, one can see three notable sections. The first allows for the confirmation of the initial value, where the BCRB begins at $10\,\mathrm{m}$ for the positions and at $5\,\mathrm{m}$ for the bias, which correspond to the standard deviations associated with the covariance matrix of initial conditions mentioned before. The RMSE starts at values close to the BCRB, although not exactly matching. The second panel shows the initial convergence and the third panel shows the oscillatory results for the positions and bias.

The third panel of Figures 8.3 - 8.6 prove that the RMSE does a good job in following the behaviour of the BCRB, which might suggest that this is a good algorithm for the desired estimation. One should note that there are occasional points at which the RMSE falls below the BCRB line. This behaviour can be explained by the BCRB characteristics mentioned in Chapter 7, where it is a definite lower bound for any unbiased estimators for a certain system, whereas the EKF gives no guarantees of an unbiased estimation. In fact, we know from the previous analysis that this estimation is indeed a biased one, which is again visible in the third panel of Fig. 8.6, where the bias RMSE is mostly below the BCRB. Table 8.1 provides a summary of the results obtained for this filter.

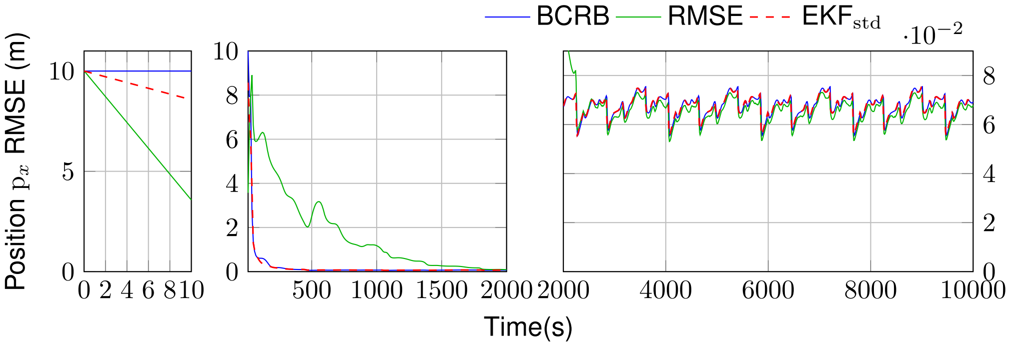

Figure 8.3: Position p_x estimation RMSE of the EKF.

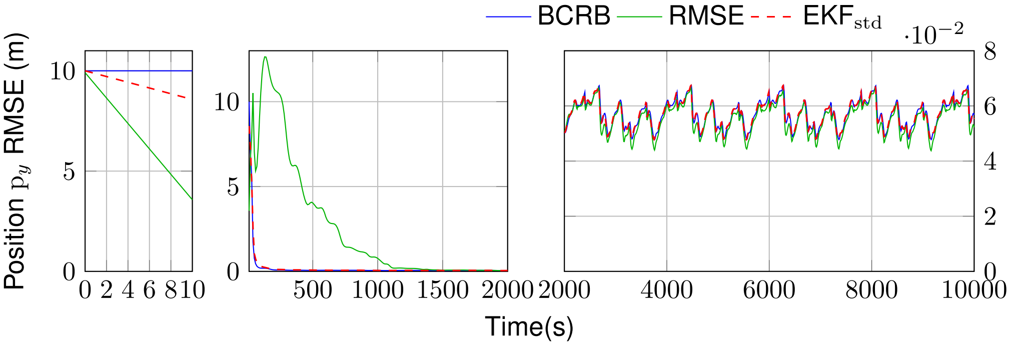

Figure 8.4: Position p_y estimation RMSE of the EKF.

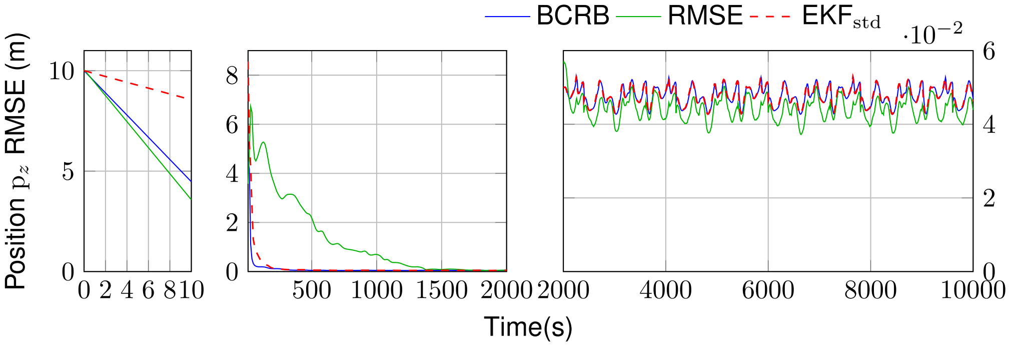

Figure 8.5: Position p_z estimation RMSE of the EKF.

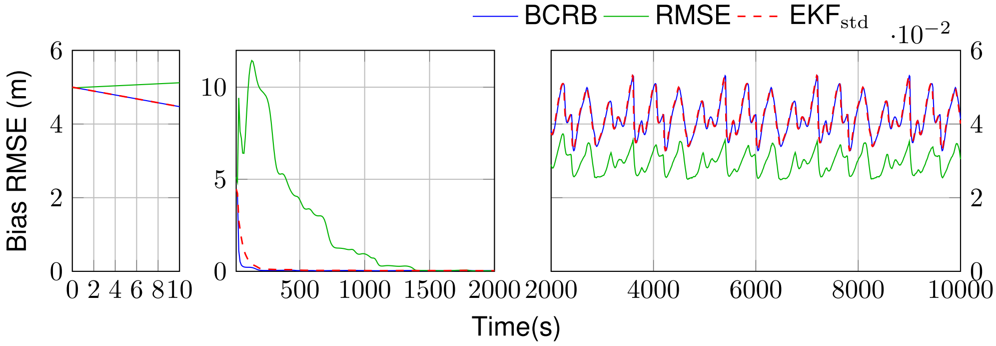

Figure 8.6: Bias estimation RMSE of the EKF.

Table 8.1: Monte Carlo outcomes of the EKF for system with bottom-lock.

Number of convergences	Run time (min)	Final average position error (m)	Final average bias error (m)
10000	4.55	$[-2.18\ \ 7.94\ \ -4.60]^T \times 10^{-4}$	8.21×10^{-4}

A brief analysis on the initial and final Gaussian distributions of the state variables is performed. Because the different components of the position evidence similar results, the analysis can be performed only on the x component, without loss of information. The Gaussian distribution fitted to the initial error of p_x is presented in Fig. 8.7, whereas the Gaussian distribution fitted to the initial error of b_c is presented in Fig. 8.8. The difference between the internal perception of convergence and the actual results is further exemplified for the variables p_x and b_c, in Figures 8.9 and 8.10, respectively. These show the Gaussian distribution fitted to the histogram of the final error of each variable, as well as the internal distribution assumed by the filter for each one. The fact that the plot of the Gaussian distribution of the EKF falls below the plot of the fitted Gaussian distribution in Fig. 8.10 is related to the previously mentioned behaviour in the third panel of Fig. 8.6, where the bias RMSE is mostly below the BCRB.

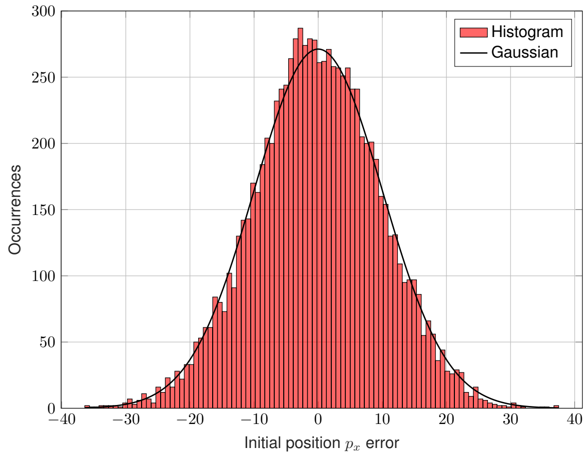

Figure 8.7: Initial p_x distribution of the EKF.

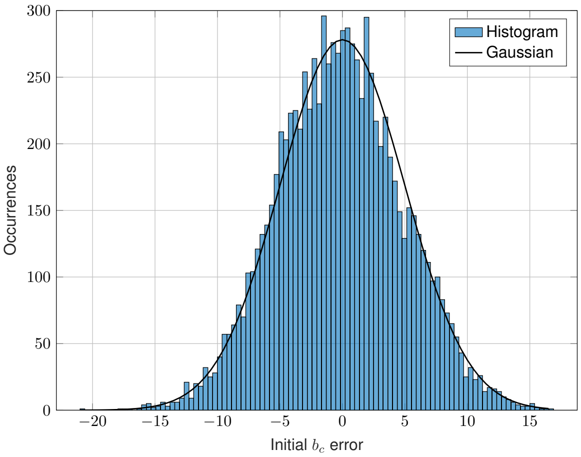

Figure 8.8: Initial b_c distribution of the EKF.

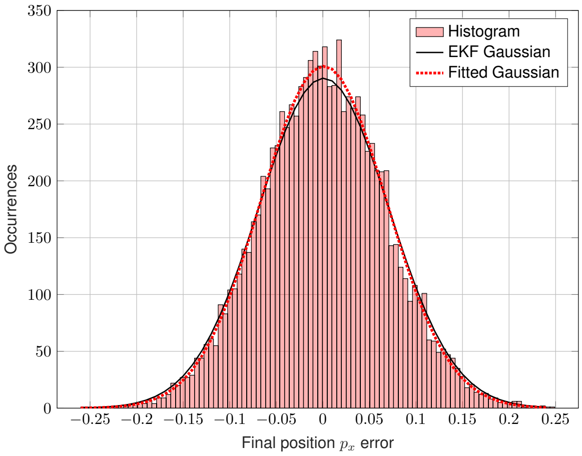

Figure 8.9: Final p_x distribution of the EKF.

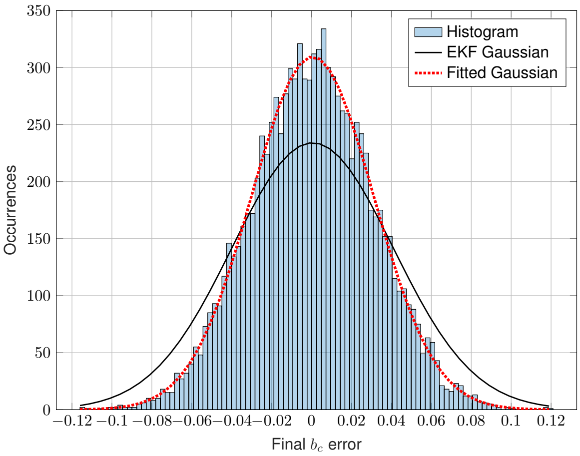

Figure 8.10: Final b_c distribution of the EKF.

8.2 UKF with bottom-lock Monte Carlo simulations

The same method described in Section 8.1 is used, with chosen covariance matrices $\mathbf{Q} = \mathbf{diag}(10^{-4}\,\mathbf{I}_3, 10^{-4})$, $R = 10^{-2}$, and $\mathbf{P} = \mathbf{diag}(100\,\mathbf{I}_3, 25)$. For the UKF the same number of runs is performed, maintaining the thresholds for convergence as $1\,\mathrm{m}$ for the positions and $0.3\,\mathrm{m}$ for the bias. This means that if the absolute value of the final state estimate errors are above this value, the filter is said to not converge. In this case, one can start by attesting the worse performance of this filter, since only 8278 convergences were achieved.

The average error over time for each of the state variables is presented in Figures 8.11 - 8.12. From these, one can verify that much higher values are present during the initial transients, with the results diminishing for larger timestamps. The steady-state errors are now only achieved at around $9000\,\mathrm{s}$, with its values being between around $-0.1\,\mathrm{m}$ and $0.1\,\mathrm{m}$ for the position coordinates and reaching $0.023\,\mathrm{m}$ for the bias. Comparing the results with the ones in Section 8.1, it is clear that the UKF presents much higher average errors for this problem than the EKF, with an increase of around 30 times for both the position and bias errors. The final bias error can still be disregarded in comparison with the nominal bias value, since it represents only a 1.15% error.

A comparison between the BCRB, the RMSE, and the UKF standard deviation obtained for each of the state variables is also shown in Figures 8.13 - 8.16. Regarding these plots, one can see three notable sections. The first allows for the confirmation of the initial value, where the BCRB begins at $10\,\mathrm{m}$ for the positions and at $5\,\mathrm{m}$ for the bias, which correspond to the standard deviations associated with the covariance matrix of initial conditions mentioned before. The RMSE starts at values close to the BCRB. The second panel shows the initial convergence and the third shows the oscillatory results for the positions and bias.

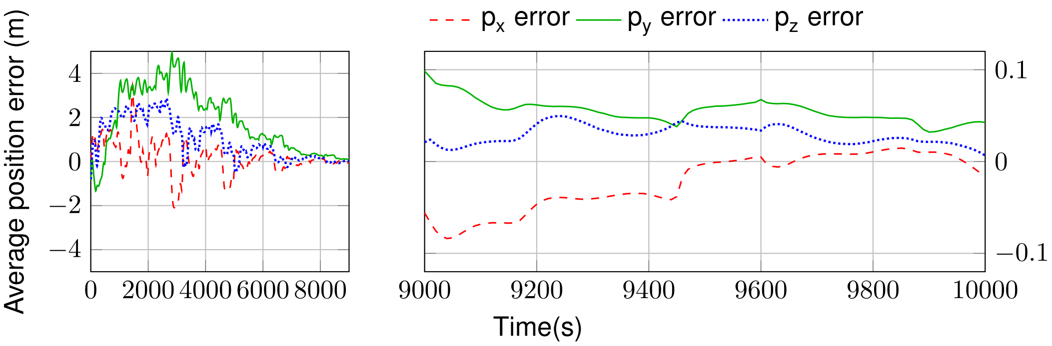

Figure 8.11: Position estimation average error of the UKF.

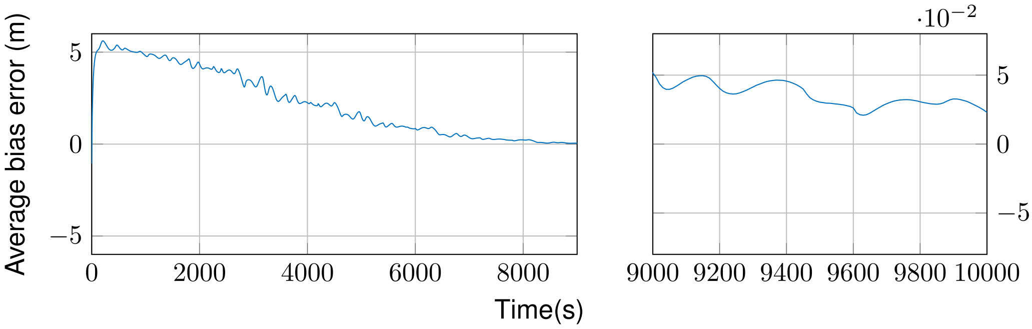

Figure 8.12: Bias estimation average error of the UKF.

It can be noted, from the third panel of Figures 8.13 - 8.15, that the RMSE now shows a higher value when compared to the BCRB, and so, the offsets shown in the estimations now come from the filter's inability to produce better results in the given time, rather than a necessarily intrinsic bias in its estimations. For this filter, the BCRB does seem to provide a definite lower bound, although the UKF also gives no guarantees of an unbiased estimation. Table 8.2 provides a summary of the results obtained for this filter.

Figures 8.17 and 8.18 present the Gaussian distribution fitted to the histogram of the final error of p_x and b_c, respectively, as well as the internal distribution assumed by the filter for each one. The convergence difficulty previously mentioned is further shown in these plots, since the disparity between the two Gaussians is higher than the ones presented in Section 8.1. The Gaussian distribution fitted to the initial errors are similar to the ones presented in Figures 8.7 and Fig. 8.8.

Table 8.2: Monte Carlo outcomes of the UKF for system with bottom-lock.

Number of convergences	Run time (min)	Final average position error (m)	Final average bias error (m)
8278	17.45	$[-1.69\ \ 4.26\ \ 0.67]^T \times 10^{-2}$	0.023

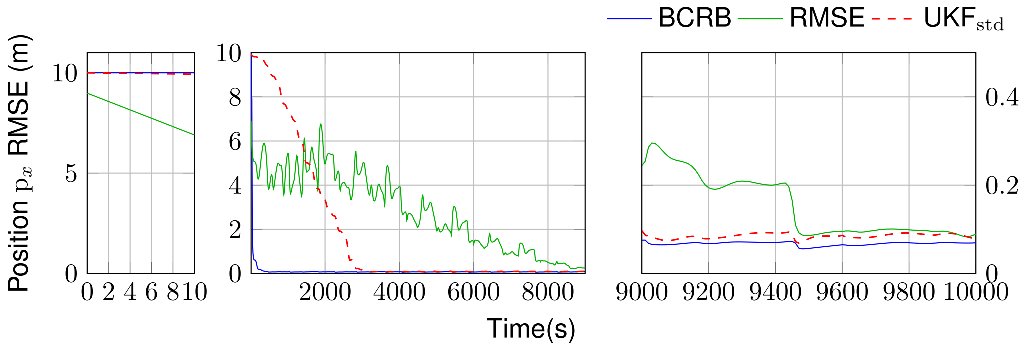

Figure 8.13: Position p_x estimation RMSE of the UKF.

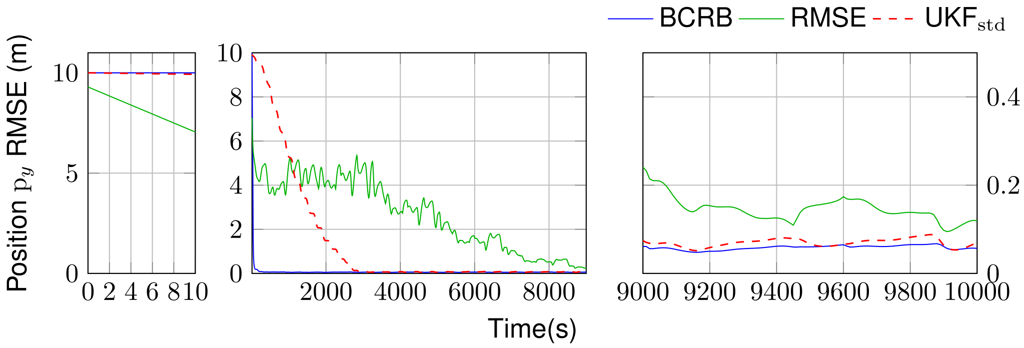

Figure 8.14: Position p_y estimation RMSE of the UKF.

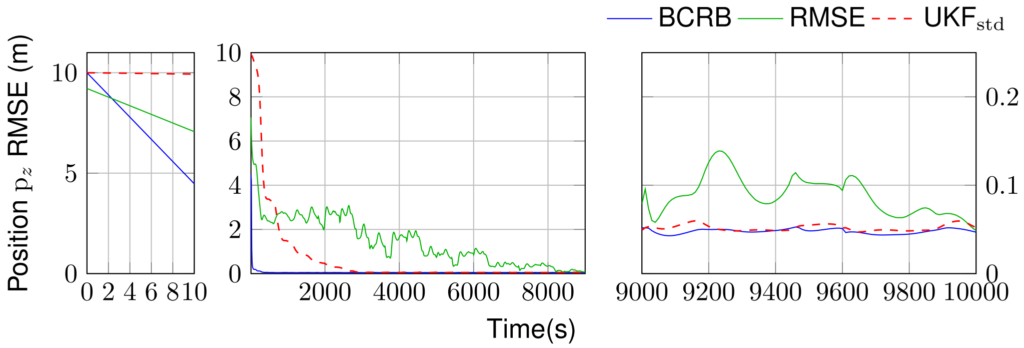

Figure 8.15: Position p_z estimation RMSE of the UKF.

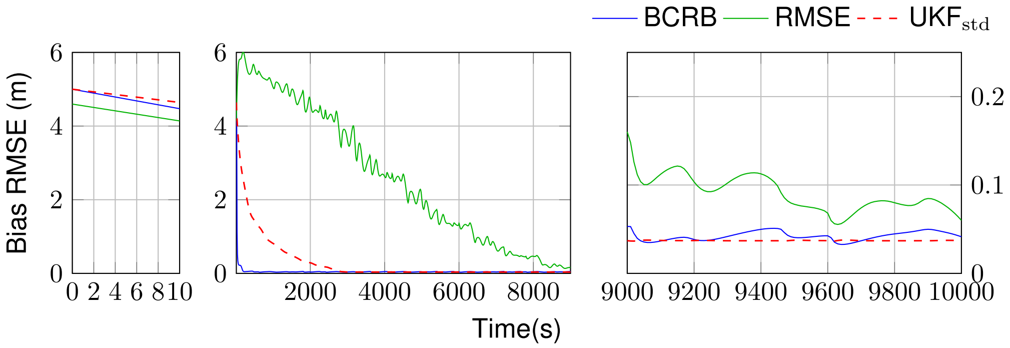

Figure 8.16: Bias estimation RMSE of the UKF.

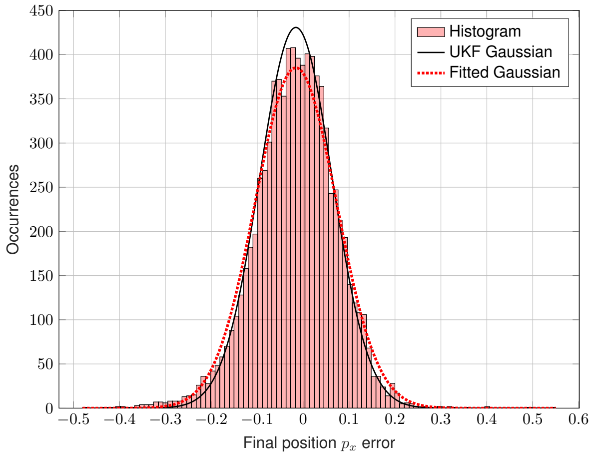

Figure 8.17: Final p_x distribution of the UKF.

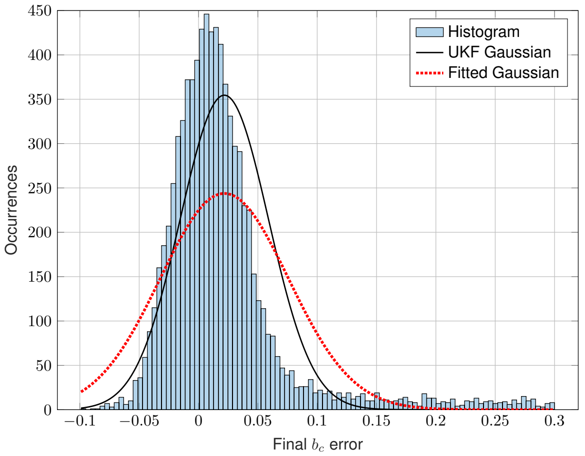

Figure 8.18: Final b_c distribution of the UKF.

8.3 LKF with bottom-lock Monte Carlo simulations

The same method described in Section 8.1 is used, with chosen covariance matrices $\mathbf{Q} = \mathbf{diag}(10^{-4}\,\mathbf{I}_3, 10^{-4}\,\mathbf{I}_2)$, $R = 10^{-2}$, and $\mathbf{P} = \mathbf{diag}(100\,\mathbf{I}_3, 25, 10^{-4})$. For the LKF the same number of runs is performed, maintaining the thresholds for convergence at $1\,\mathrm{m}$ for the positions and $0.3\,\mathrm{m}$ for the bias, and adding a threshold of $0.3\,\mathrm{m}$ for the range. This means that if the absolute value of the final state estimate errors are above this value, the filter is said to not converge. In this case, one can attest to the global convergence guarantee mentioned in Section 5.4.3.2, since 9999 of the runs converge under the above circumstances. The only run that does not converge under this description, is not a case of failure of convergence, but a case in which the final errors are above the considered thresholds.

The average error over time for each of the state variables is presented in Figures 8.19 - 8.21. From these, one can verify that the LKF presents the lowest peaks in the initial transients, out of all the tested filters. The steady-state errors are now achieved very fast, at around $500\,\mathrm{s}$, with the plots presenting clear periodic behaviour. There are noticeable skews towards negative values for both the bias and p_z. All errors are of the order of $10^{-2}\,\mathrm{m}$, which is in fact higher than the steady state errors achieved with the EKF. However, because these results are still very good when compared to the true values, and taking into consideration the initial transients and the fast convergence, this seems to be the best filter option for the problem at hand.

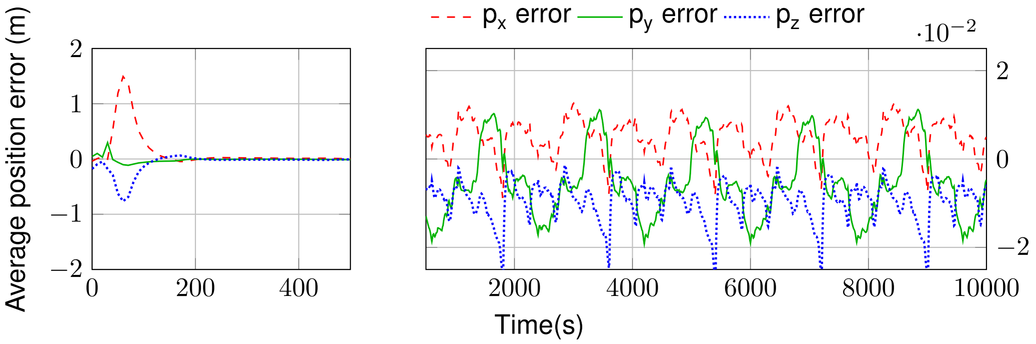

Figure 8.19: Position estimation average error of the LKF.

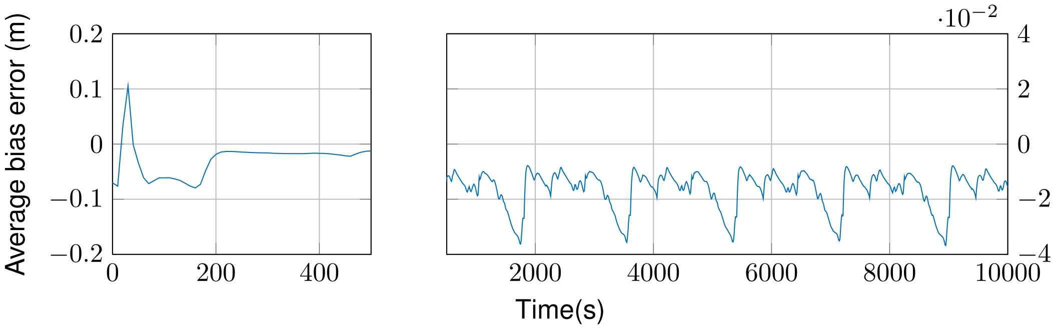

Figure 8.20: Bias estimation average error of the LKF.

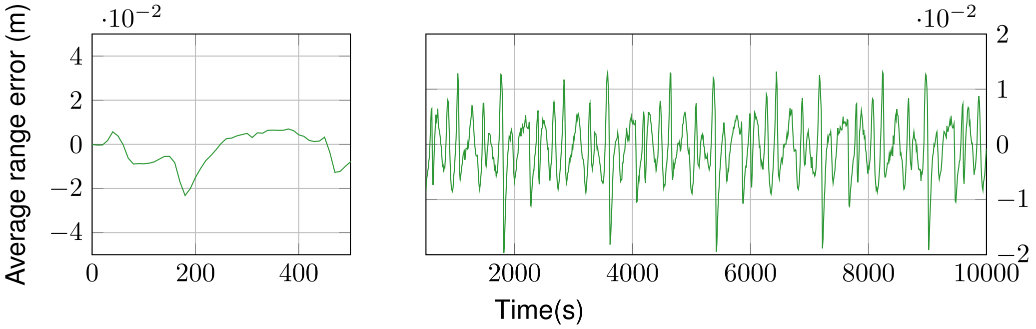

Figure 8.21: Range estimation average error of the LKF.

A comparison between the BCRB, the RMSE, and the LKF standard deviation obtained for each variable is shown in Figures 8.22 - 8.25. Regarding these plots, the three notable sections remain. The first confirms the initial value, the second shows the initial convergence, and the third shows the oscillatory results obtained for the positions and bias. The first panels show the expected results, where the BCRB begins at $10\,\mathrm{m}$ for the positions and at $5\,\mathrm{m}$ for the bias, which correspond to the standard deviations associated with the covariance matrix of initial conditions mentioned before. The RMSE starts at values close to the BCRB, although not exactly matching.

When comparing the third panel of the plots with their equivalents from Section 8.1, one can see similar RMSE values, which suggests the filter obtains similarly accurate results. We can again verify the faster convergence of this filter and a fairly decent tracking of the BCRB by the RMSE. One should note, once more, that the occasional points at which it falls below the bound, show the biased nature of the filter (also seen in the negative skews mentioned in the analysis of the plots of the average errors).

The asymptotic convergence shown in the second panels, heavily contrasts with the high peaks shown in the second panels of Figures 8.3 - 8.6, which provides further proof that this filter is a better option for the desired estimation. Table 8.3 provides a summary of the results obtained for this filter.

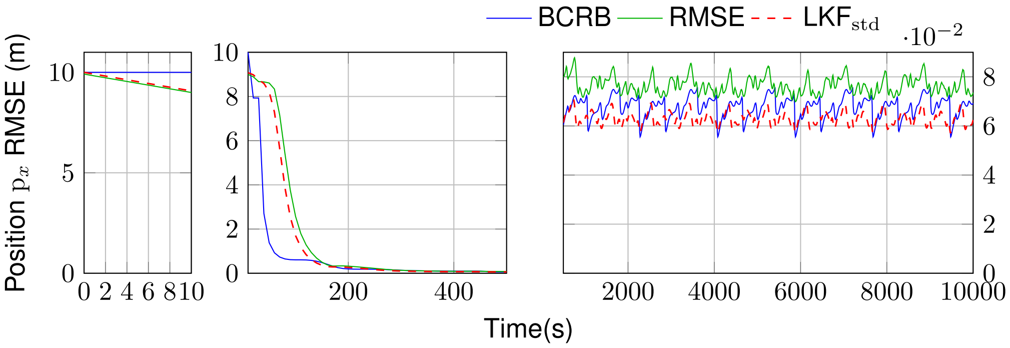

Figure 8.22: Position p_x estimation RMSE of the LKF.

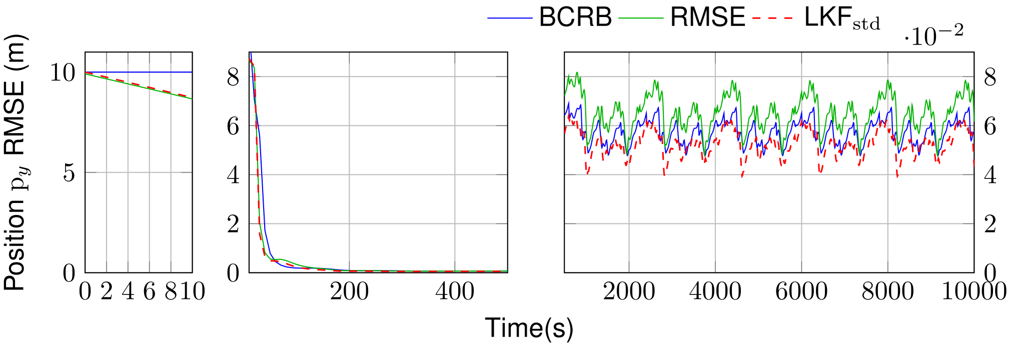

Figure 8.23: Position p_y estimation RMSE of the LKF.

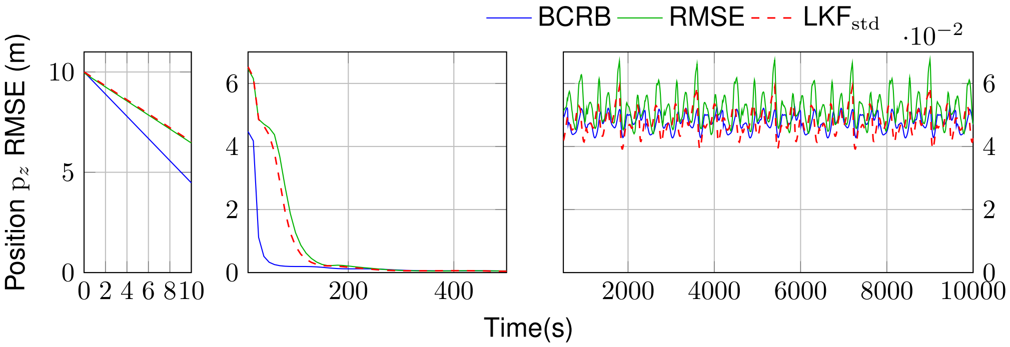

Figure 8.24: Position p_z estimation RMSE of the LKF.

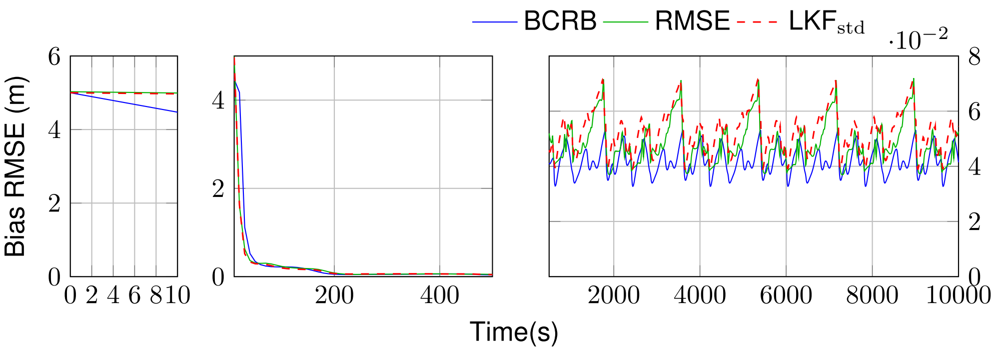

Figure 8.25: Bias estimation RMSE of the LKF.

Figures 8.27 - 8.29 present the Gaussian distribution fitted to the histogram of the final error of p_x, b_c, and r, respectively, as well as the internal distribution assumed by the filter for each one. The Gaussian distribution fitted to the initial range errors is displayed in Fig. 8.26, whereas the position and bias distributions are still similar to the ones presented in Figures 8.7 and 8.8. Once again, the persistence of RMSE values below the BCRB for the bias variable is associated with Fig. 8.28, where the LKF Gaussian distribution falls below the fitted Gaussian distribution.

Table 8.3: Monte Carlo outcomes of the LKF for system with bottom-lock.

Number of convergences	Run time (min)	Final average position error (m)	Final average bias error (m)	Final average range error (m)
9999	6.03	$[4.87 \quad -4.70 \quad -6.32]^T \times 10^{-3}$	-0.016	-7.74×10^{-4}

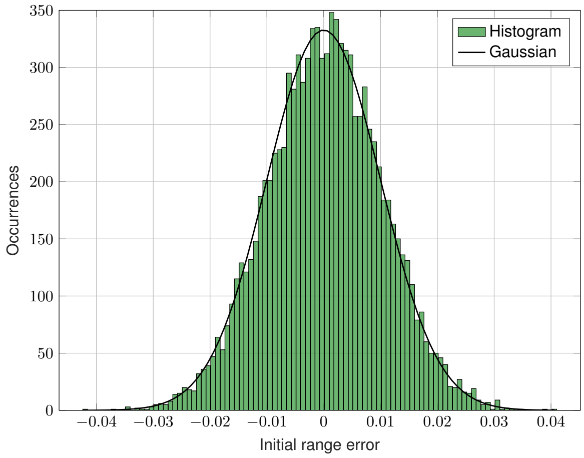

Figure 8.26: Initial r distribution of the LKF.

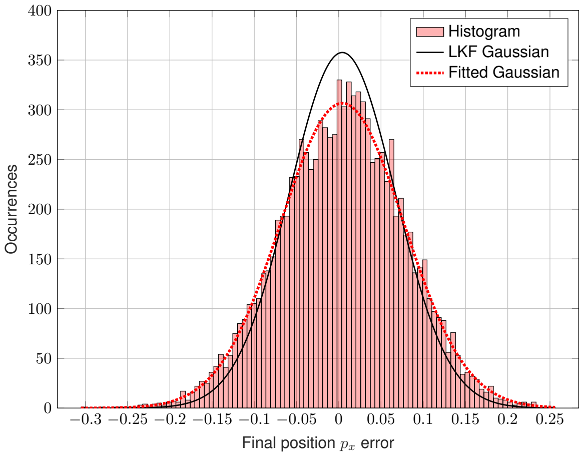

Figure 8.27: Final p_x distribution of the LKF.

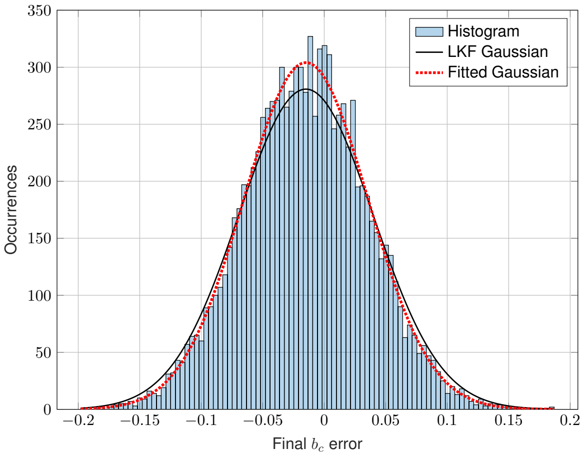

Figure 8.28: Final b_c distribution of the LKF.

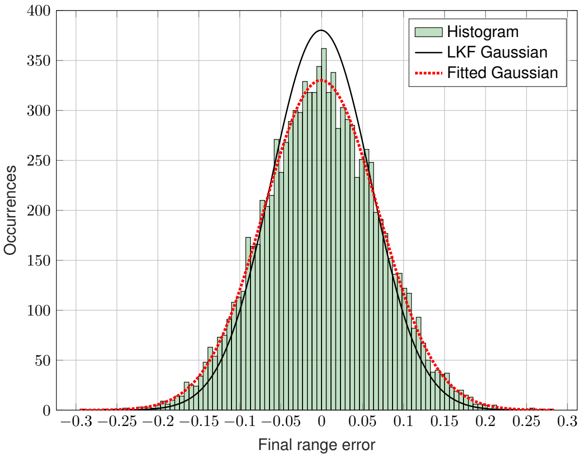

Figure 8.29: Final r distribution of the LKF.

8.4 Filter comparison

In order to weigh up the three simulated filters, some further analysis is performed. Once again, because the different components of the position evidence similar results, the analysis can be performed only on the x component, without loss of information. The RMSE results for p_x and b_c for each filter, as well as the corresponding BCRB values are shown in Figures 8.30 and 8.31, respectively. These allow for a better comparison between the obtained tracking for each filter, showing clear poorer results for the UKF and reinforcing the biased nature of the EKF, since its RMSE consistently falls below the BCRB. In order to perform a numerical comparison, the average value is calculated for the final half of the simulation time, i.e., between $5000\,\mathrm{s}$ and $10000\,\mathrm{s}$. The results are shown in Table 8.4 and show that the EKF RMSE has the closest average values, however, it is consistently below the BCRB. The LKF presents the lowest difference to the BCRB without going under those values, while the UKF is clearly the worst performing filter.

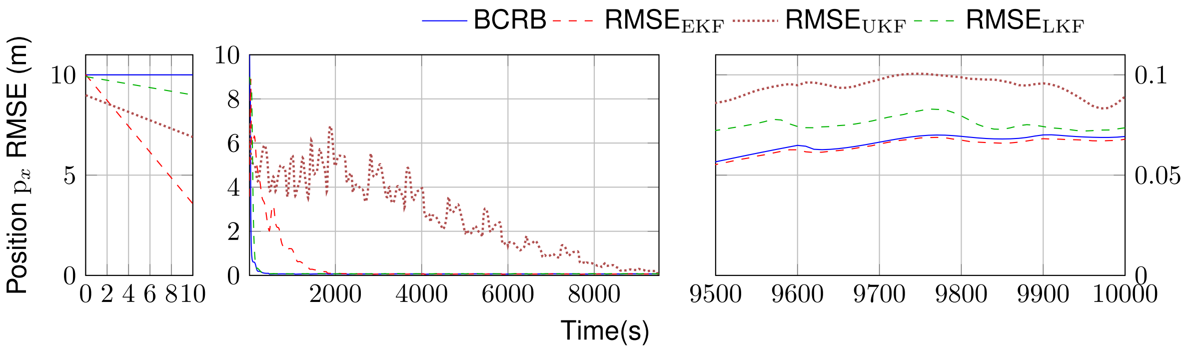

Figure 8.30: Position p_x estimation RMSE comparison.

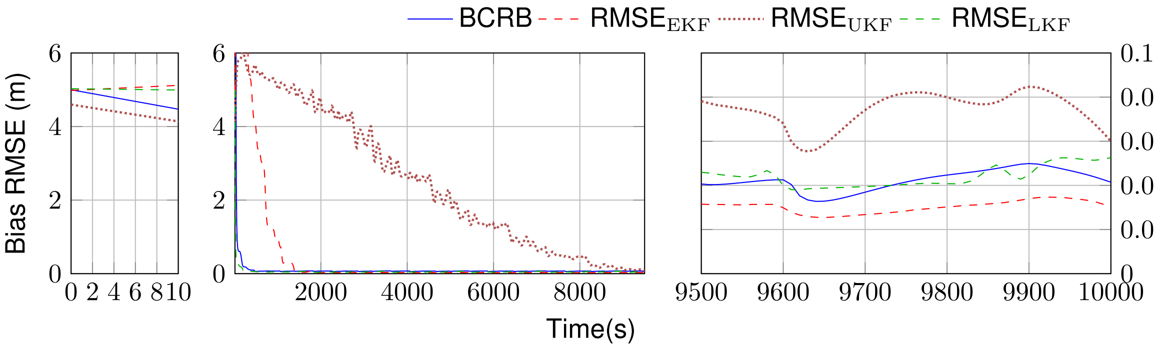

Figure 8.31: Bias estimation RMSE comparison.

Table 8.4: Average RMSE and BCRB values between 5000 s and 10000 s.

	$p_x(\mathrm{m})$	$p_y(\mathrm{m})$	$p_z(\mathrm{m})$	$b_c(\mathrm{m})$
BCRB	0.0678	0.0570	0.0478	0.0421
RMSE EKF	0.0656	0.0551	0.0441	0.0296
RMSE UKF	0.9525	0.9005	0.4022	0.6831
RMSE LKF	0.0742	0.0647	0.0520	0.0483

Referring back to Tables 8.1, 8.2, and 8.3, one can see that the UKF has significantly less runs that achieve a convergence under the specified thresholds, whereas the EKF and LKF have comparable performances. Evaluating the final average errors, the EKF considerably outperforms the other two filters with regard to the position. The UKF shows much worse results related to the final average bias error, whereas the EKF and the LKF have comparable results. A laptop with the following specifications was used: Intel (R) Core (TM) i7-1065G7 CPU @ 1.30GHz 1.50GHz processor, NVidia GeForce MX250 graphics card, 16 GB LPDDR4X-3733 RAM, SDD 1TB disk and Windows 10 Home 64 bits operating system. It is clear from the previously mentioned tables that the UKF has a much longer processing time. As far as convergence times, the LKF shows clearly faster results.

9

Monte Carlo simulations without bottom-lock

For each Monte Carlo run, the values of $\mathbf{u}_{\mathrm{measured}}(k)$ and $r_{\mathrm{measured}}(k)$ are generated by adding white Gaussian noise to the true values of the system inputs and range measurements. The process noise is, then, centered around the true values, with a chosen covariance matrix $\mathbf{Q} = \mathbf{diag}(10^{-4}\,\mathbf{I}_3, 10^{-6}\,\mathbf{I}_4)$. The observation noise is also centered around the true values, with covariance matrix $R = 10^{-4}$. At each run, an initial condition is also sampled from the Gaussian distribution with zero mean, i.e., centered around the true initial conditions, with covariance matrix $\mathbf{P} = \mathbf{diag}(100\,\mathbf{I}_3, 0.01\,\mathbf{I}_3, 25)$, which is also the given initial covariance matrix of the filter.

For the purposes of this experiment, 10000 runs were performed and a threshold for convergence was set to $5\,\mathrm{m}$ for the positions, $0.05\,\mathrm{m/s}$ for the velocities and $0.8\,\mathrm{m}$ for the bias. This means that if the absolute value of the final estimation errors are above these values, the filter is said to not converge. These values were chosen empirically, but can be considered realistic, since 9199 convergences were achieved. Because this is a harder problem, one allows for larger thresholds than the ones used for the equivalent simulations with bottom-lock.

The average error over time for each of the state variables is presented in Figures 9.1 - 9.3. From these, one can verify that during the initial transients a much higher error is present, with the values growing smaller for larger time-stamps. The first panel of the plots shows a much more oscillatory response, when compared with the situation with bottom-lock, which shows the higher sensitivity of the experiment without bottom-lock. The initial transients are about 10 times larger than the ones obtained in Section 8.1 for position p_x, whereas positions p_y and p_z are increased 5 times. The second panel shows that the oscillatory responses for the position and velocity variables continue to be dampened for larger time-stamps, which suggests a longer convergence time, however good results can be achieved, considering the final average errors are $\mathbf{p}_{error} \approx [0.89\ \ -0.012\ \ -0.178]^T\,[\mathrm{m}]$, $\mathbf{v}_{error} \approx [1.50\ 1.78\ 4.84]^T \times 10^{-3}\,[\mathrm{m/s}]$, and $b_{c_{error}} \approx -0.052\,\mathrm{m}$.

The first panel of Fig. 9.3 clearly shows the previously mentioned offset in this estimation, which is further verified in the considerable difference between the two Gaussian distributions in Fig. 9.14. The second panel of Fig. 9.3 shows that the estimation follows an asymptotic convergence to the true value, but the results are not as good as before. The final time average bias error achieves approximately a $2,6\%$ error, clearly much higher than the previous 0.04%, or even the 0.11% (considering the last peak from Figure 8.2, rather than the final value). The velocity presents a larger error along the z axis than the x or y axes, with the final error being 4.84% of the nominal value.

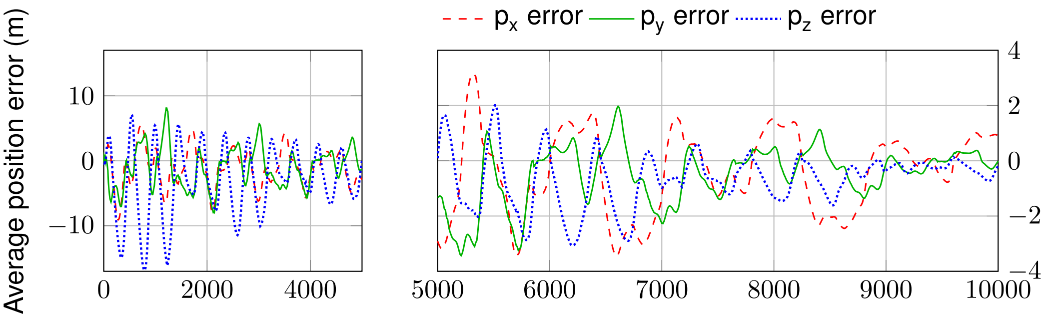

Figure 9.1: Position estimation average error of the EKF without bottom-lock.

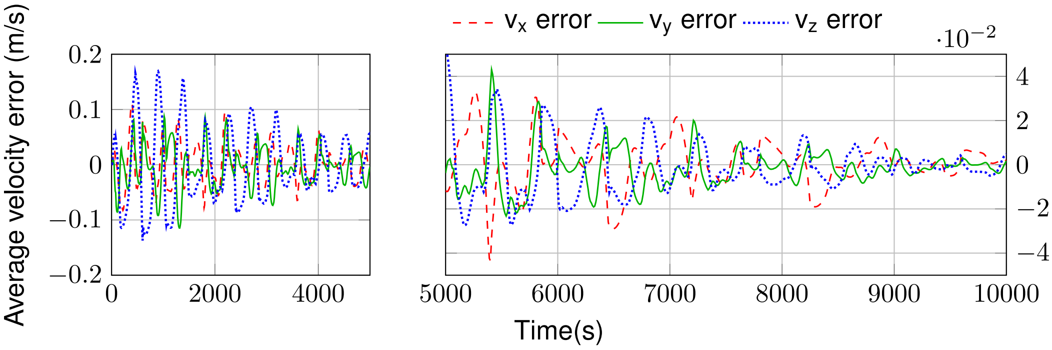

Figure 9.2: Velocity estimation average error of the EKF without bottom-lock.

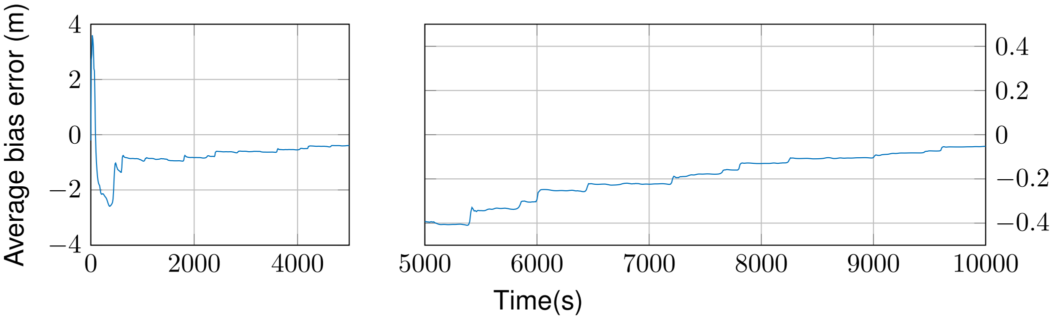

Figure 9.3: Bias estimation average error of the EKF without bottom-lock.

The standard deviation for each state variable is given by the square root of its corresponding diagonal element of the average covariance matrix, i.e., for p_x the standard deviation is given by $\sqrt{\mathbf{P_{1,1}}}$, where $\mathbf{P}$ is the average covariance matrix of the filter over all runs that converge. Because this value tends to overestimate the performance of the filter, it is interesting to compare the internal perception of convergence with the actual outcomes. Therefore, a comparison between the EKF standard deviation, the BCRB, and the RMSE obtained for each of the state variables is shown in Figures 9.4 - 9.10.

Regarding these plots, there are, once again, three notable sections. Like before, the first panel of the figures confirm that the values of the BCRB begin at $10\,\mathrm{m}$ for the positions, $0.1\,\mathrm{m/s}$ for the velocities and at $5\,\mathrm{m}$ for the bias, which correspond to the standard deviations associated with the covariance matrix of initial conditions. The RMSE starts at values close to the BCRB. The second panel shows the initial convergence and the third shows the oscillatory results for the positions and velocities and the more asymptotic behaviour for the bias. All images exhibit a close tracking of the BCRB by the EKF standard deviation, with the RMSE results being the most different. This indicates that the filter is overestimating its performance and can explain some of the issues with convergence. The third panel of Figures 9.7 - 9.9 show that the filter has a much harder time following the correct behaviour than in Section 8.1, presenting a lot more oscillations than in the previous case. However, because the correct order of magnitude is achieved, one can still assume this to be a good solution for the desired estimation. The velocity errors are the ones where the filter has the most accurate perception of its estimations, as shown by the nearly overlapping Gaussian distributions in Fig. 9.13.

In this case, the fact that the resulting EKF estimations are biased is hinted at in Figures 9.7 and 9.8, where the plot of the RMSE falls below the BCRB. The plot in Figure 9.10 explains the conclusion from Section 6.4.2, where the bias initial condition seemed to have a bigger effect on the overall convergence. This is because the RMSE does not achieve a closer tracking of the BCRB, which justifies the potential problems with the divergence of the filter. A summary of the outcomes of the Monte Carlo runs for this filter can be seen in Table 9.1, which allows for a direct comparison with the results in Table 8.1.

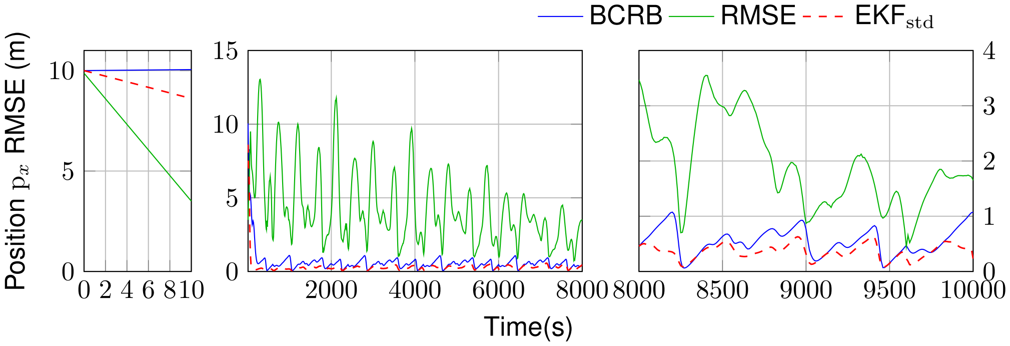

Figure 9.4: Position p_x estimation RMSE of the EKF without bottom-lock.

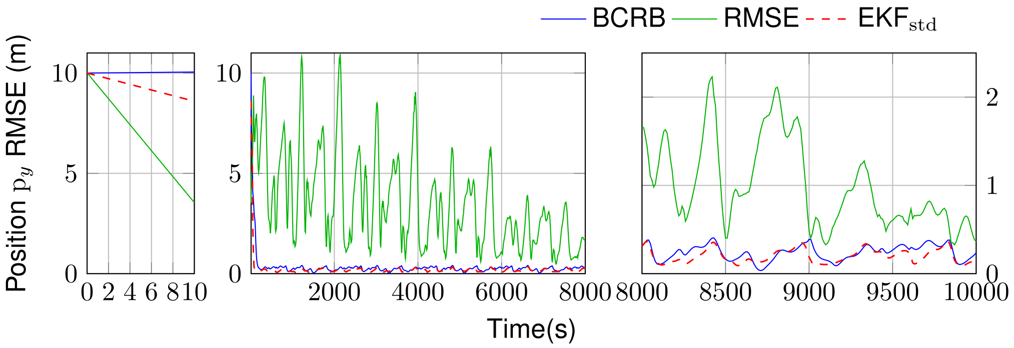

Figure 9.5: Position p_y estimation RMSE of the EKF without bottom-lock.

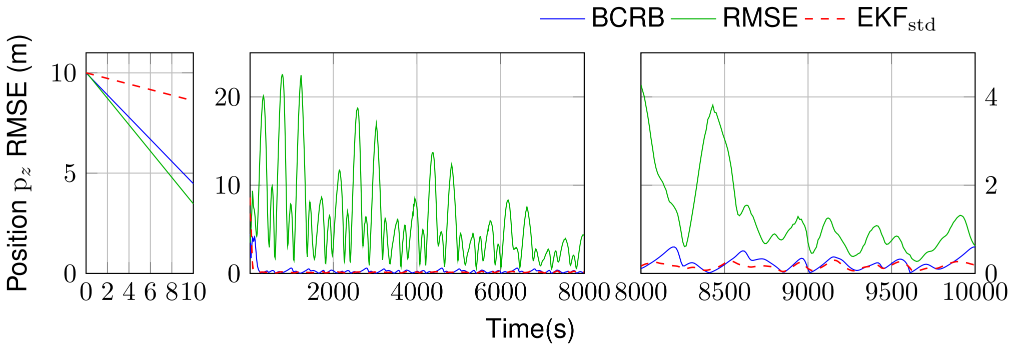

Figure 9.6: Position p_z estimation RMSE of the EKF without bottom-lock.

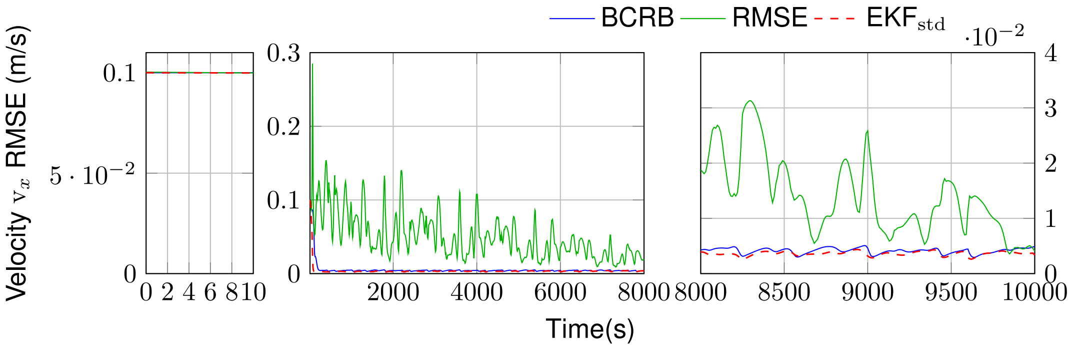

Figure 9.7: Velocity v_x estimation RMSE of the EKF without bottom-lock.

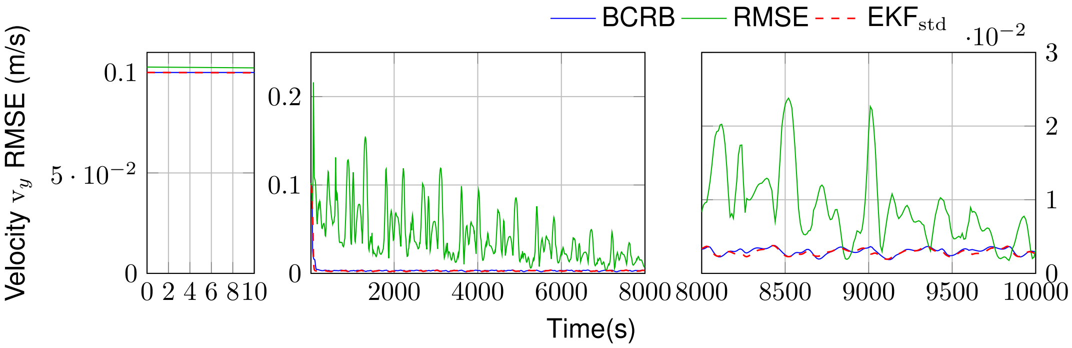

Figure 9.8: Velocity v_y estimation RMSE of the EKF without bottom-lock.

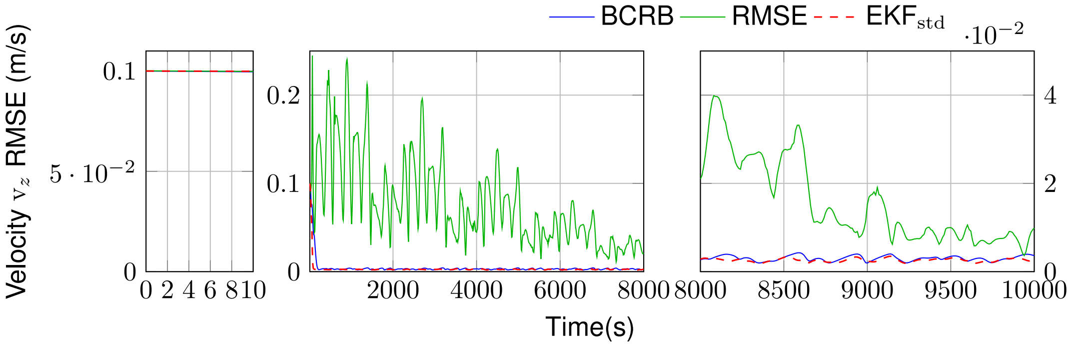

Figure 9.9: Velocity v_z estimation RMSE of the EKF without bottom-lock.

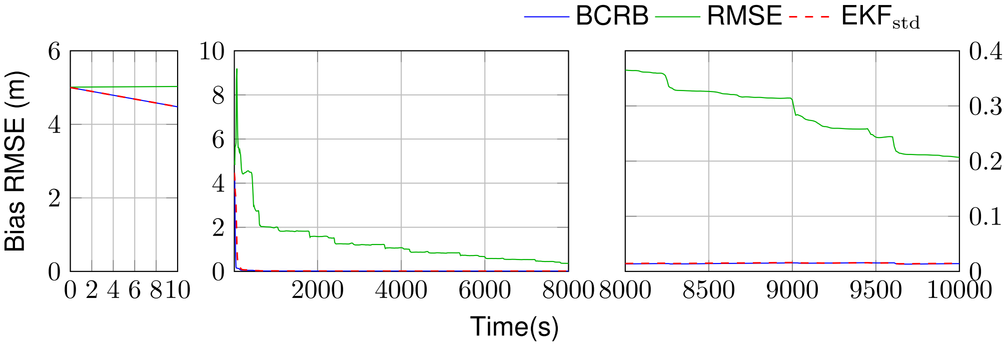

Figure 9.10: Bias estimation RMSE of the EKF without bottom-lock.

The difficulty in BCRB tracking by the RMSE is further exemplified for the variables p_x, v_x, and b_c, in Figures 9.12 - 9.14. These show the Gaussian distribution fitted to the histogram of the final error of each variable, as well as the internal distribution assumed by the filter for each one. The Gaussian distribution fitted to the initial error of v_x is presented in Fig. 9.11, whereas p_x and b_c follow distributions similar to the ones shown in Fig. 8.7 and Fig. 8.8, respectively. Once again, because the different components of the position and of the velocity evidence similar results, the analysis is performed only on the p_x and v_x components, without loss of information.

Having performed an impartial analysis of these results, it is important to, once again, note the difficulty of this problem. Indeed, achieving the correct order of magnitude of the BCRB by the position RMSE is, in and of itself, a very satisfactory result. The fact that the average position error in steady-state converges to below $2\,\mathrm{m}$ indicates that the filter will provide the desired tracking of the vehicle with decent accuracy, however it does imply a much longer convergence time. Overall, the obtained results prove to be very good, as it is excellent to achieve such small errors for a problem of this complexity.

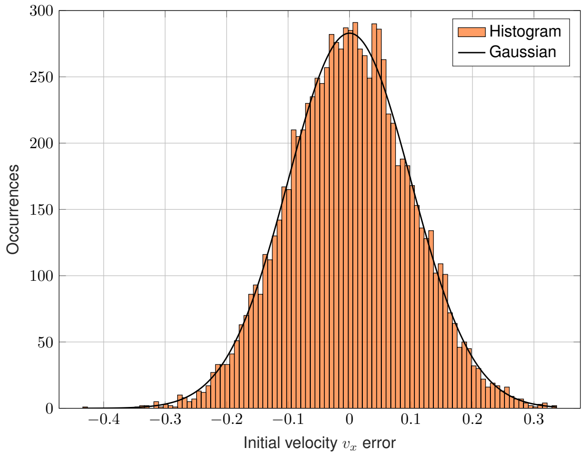

Figure 9.11: Initial v_x distribution of the EKF without bottom-lock.

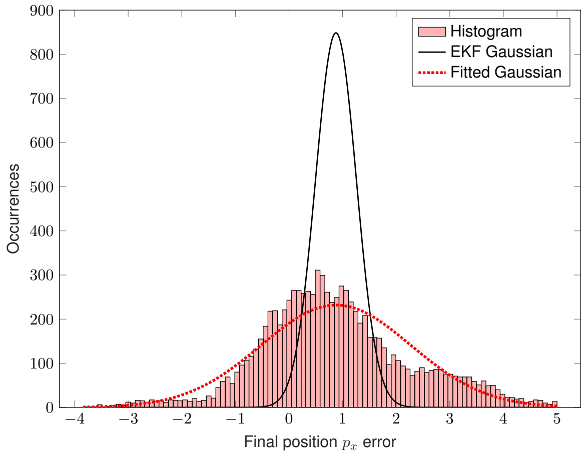

Figure 9.12: Final p_x distribution of the EKF without bottom-lock.

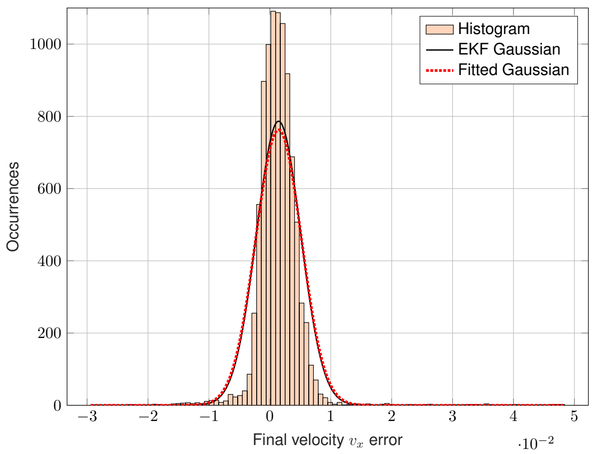

Figure 9.13: Final v_x distribution of the EKF without bottom-lock.

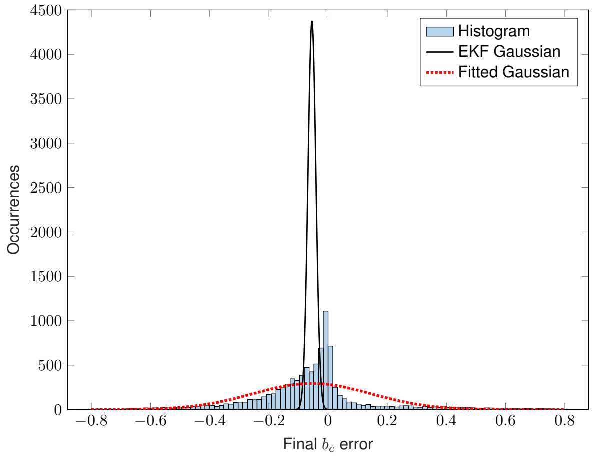

Figure 9.14: Final b_c distribution of the EKF without bottom-lock.

Table 9.1: Monte Carlo outcomes of the EKF for system without bottom-lock.

Number of convergences	Run time (min)	Final average position error (m)	Final average velocity error (m/s)	Final average bias error (m)
9199	7.70	$[0.89\ \ -0.012\ \ -0.178]^T$	$[1.50\ \ 1.78\ \ 4.84]^T \times 10^{-3}$	-0.052

10

Conclusion

The work developed in this thesis addressed two problems of localization based on single pseudo-range measurements. The first is related to a system with bottom-lock, for which the three position components, as well as the bias associated with the clock offset, were successfully estimated. The three proposed solutions for this problem were compared, via simulation results, under the same conditions. A thorough Monte Carlo analysis was performed, for which the RMSE and average error of the solutions were computed and compared. The UKF clearly underperformed, when compared with the remaining solutions, presenting the largest average errors and RMSE, as well as the longest convergence time. The LKF presented slightly higher steady-state errors than the EKF but a much faster convergence and less biased estimations.

Initial conditions with which the filters would not converge were shown for the EKF and the UKF, noting that the latter needed a much smaller increment to present this failure of convergence. Equivalent simulations were performed with the LKF, which proved to converge in both cases due to its GES guarantee, further illustrating the advantages of this filter. Taking all factors into account, because the EKF and LKF present comparable results, as well as computational costs, and since the LKF shows a much faster convergence time and offers global convergence guarantees, one concludes that this is the best solution among all three. It is noted that all simulations were run with the same tuning parameters, chosen to allow for a convergence of the UKF, which means that further tuning of the EKF and LKF parameters could potentiate better results for these filters.

The second problem is related to a system without bottom-lock, for which only an introductory approach was taken, with the design and testing of an EKF. The three position components, the three velocity components, and the bias associated with the clock offset were successfully estimated. This filter presented a much longer convergence time when compared with the version with bottom-lock, which is to be expected considering the increased difficulty of the problem. The resulting steady-state errors for this system, as well as the difference between the obtained RMSE and the BCRB, were also higher than for the system with bottom-lock. However, because the results are still deemed satisfactory, and convergence within the set thresholds is obtained for most Monte Carlo runs, one can consider the estimations to be successful.

Future work can be done for the system with bottom-lock, mainly relating to the tuning of the parameters, although the best solution is proven to be the LKF, for which more thorough results were presented in [23]. Extrapolating from the results obtained for the system with bottom-lock, one could take a similar approach and design a LKF for the system without bottom-lock by further augmenting the state. The same analysis performed in this work for the system with bottom-lock should be performed for the system without bottom-lock. This way, an extensive comparison between the EKF and LKF should be performed. A comparison with the UKF for the system without bottom-lock could also be attempted, although, given the results obtained for the system with bottom-lock, it is unlikely that this filter will converge, whatever the parameter tuning. If a better performance of the LKF is also obtained for the case without bottom-lock, the final step would be to test the application of both of the developed LKF solutions in field experiments.

Bibliography

[1] S. A. Gafurov and E. V. Klochkov, “Autonomous Unmanned Underwater Vehicles Development Tendencies,” *Procedia Engineering*, vol. 106, pp. 141–148, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.proeng.2015.06.017

[2] K. Zwolak, B. Simpson, B. Anderson, E. Bazhenova, R. Falconer, T. Kearns, H. Minami, J. Roperez, A. Rosedee, H. Sade, N. Tinmouth, R. Wigley, and Y. Zarayskaya, “An unmanned seafloor mapping system: The concept of an auv integrated with the newly designed usv sea-kit,” in *Proceedings of the 2017 MTS/IEEE Oceans Conference - Aberdeen*, Aberdeen, Scotland, 2017, pp. 1–6.

[3] H. Bray. (2014, Apr. 14) Bluefin robot joins search for missing malaysian plane. Accessed Aug. 14, 2020. [Online]. Available: https://www3.bostonglobe.com/business/2014/04/14/bluefin/L9to17XsyJsxqRtoNigKkO/story.html?arc404=true

[4] A. Turetta, G. Casalino, E. Simetti, A. Sperindè, and S. Torelli, “Analysis of the accuracy of a LBL-based underwater localization procedure,” in *Proceedings of the 2014 MTS/IEEE Oceans Conference - St. John’s*, St. John’s, NL, Canada, 2014, pp. 1–7.

[5] S. M. Smith and D. Kronen, “Experimental results of an inexpensive short baseline acoustic positioning system for auv navigation,” in *Proceedings of the 1997 MTS/IEEE Oceans Conference*, vol. 1, Halifax, NS, Canada, 1997, pp. 714–720.

[6] J. . Peyronnet, R. Person, and F. Rybicki, “Posidonia 6000: a new long range highly accurate ultra short base line positioning system,” in *Proceedings of the 1998 IEEE Oceans Conference*, vol. 3, Nice, France, 1998, pp. 1721–1727.

[7] A. Mallios, D. Ribas, and P. Ridao, “Localization Advances in the Unstructured Underwater Environment,” *Proceedings of the 9th Hellenic*, vol. 1, pp. 111–116, 2009. [Online]. Available: http://symposia.ath.hcmr.gr/oldver/symposia9/Book1/0111.pdf

[8] gisgeography. Accessed Aug. 16, 2020. [Online]. Available: https://gisgeography.com/trilateration-triangulation-gps/

[9] Inertial sensor modules. Accessed Sep. 3, 2020. [Online]. Available: https://www.xsens.com/inertial-sensor-modules

[10] L. L. Whitcomb, D. R. Yoerger, and H. Singh, “Combined Doppler/LBL based navigation of underwater vehicles,” *11th International Symposium on Unmanned Untethered Submersible Technology*, no. 10015, pp. 1–7, 1999.

[11] A. Alcocer, P. Oliveira, and A. M. Pascoal, “Underwater Acoustic Positioning Systems Based on Buoys With GPS,” *European Conference on Underwater Acoustics (ECUA)*, 2006.

[12] J. Reis, M. Morgado, P. Batista, P. Oliveira, and C. Silvestre, “Design and experimental validation of a USBL underwater acoustic positioning system,” *Sensors (Switzerland)*, vol. 16, no. 9, pp. 1–23, 2016.

[13] K. Vickery, “Acoustic positioning systems - a practical overview of current systems,” *Proceedings of the IEEE Symposium on Autonomous Underwater Vehicle Technology*, pp. 5–17, 1998.

[14] M. B. Larsen, “Synthetic long baseline navigation of underwater vehicles,” in *Proceedings of the 2000 MTS/IEEE Oceans Conference*, vol. 3, Providence, RI, USA, 2000, pp. 2043–2050.

[15] C. E. LaPointe, “Virtual long baseline (VLBL) autonomous underwater vehicle navigation using a single transponder,” *Master’s thesis, Massachusetts Institute of Technology and Woods Hole Oceanographic Institution*, 2006.

[16] T. Casey, B. Guimond, and J. Hu, “Underwater vehicle positioning based on time of arrival measurements from a single beacon,” in *Proceedings of the 2007 MTS/IEEE Oceans Conference - Vancouver*, Vancouver, BC, Canada, 2007, pp. 1–8.

[17] P. Lee, B. Jun, K. Kim, J. Lee, T. Aoki, and T. Hyakudome, “Simulation of an inertial acoustic navigation system with range aiding for an autonomous underwater vehicle,” *IEEE Journal of Oceanic Engineering*, vol. 32, no. 2, pp. 327–345, 2007.

[18] K. Reif, S. Günther, E. Yaz, and R. Unbehauen, “Stochastic stability of the discrete-time extended Kalman filter,” *IEEE Transactions on Automatic Control*, vol. 44, no. 4, pp. 714–728, 1999.

[19] S. E. Webster, R. M. Eustice, H. Singh, and L. L. Whitcomb, “Preliminary deep water results in single-beacon one-way-travel-time acoustic navigation for underwater vehicles,” *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, no. December, pp. 2053–2060, 2009.

[20] P. Batista, C. Silvestre, and P. Oliveira, “Single range aided navigation and source localization: Observability and filter design,” *Systems and Control Letters*, vol. 60, no. 8, pp. 665–673, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.sysconle.2011.05.004

[21] G. Indiveri, D. De Palma, and G. Parlangeli, “Single Range Localization in 3-D: Observability and Robustness Issues,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1853–1860, 2016.

[22] P. Batista, “Long baseline navigation with clock offset estimation and discrete-time measurements,” *Control Engineering Practice*, vol. 35, pp. 43–53, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.conengprac.2014.10.009

[23] ——, “Navigation and source localization based on single pseudo-ranges,” vol. 2020-July, 2020, pp. 5237–5242.

[24] ——, “Long baseline navigation with explicit pseudo-range clock offset and propagation speed estimation,” *European Journal of Control*, vol. 49, pp. 116–130, 2019.

[25] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2001, vol. 9, pp. 202–206.

[26] Y. Bar-Shalom, *Tracking and Data Association*. Academic Press Professional, Inc., 1987.

[27] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2001, vol. 9, pp. 382–386.

[28] A. Gelb and T. A. S. Corporation, *Applied Optimal Estimation*. The MIT Press, 1974.

[29] Z. Cai and D. Zhao, “Unscented Kalman filter for non-linear estimation,” *Geomatics and Information Science of Wuhan University*, vol. 31, no. 2, pp. 180–183, 2006.

[30] B. Southall, B. F. Buxton, and J. A. Marchant, “Controllability and observability: Tools for kalman filter design,” in *Proceedings of the British Machine Vision Conference*. BMVA Press, 1998, pp. 17.1–17.10.

[31] J. J. LaViola, “A Comparison of Unscented and Extended Kalman Filtering for Estimating Quaternion Motion,” in *Proceedings of the American Control Conference*, vol. 3, 2003, pp. 2435–2440.