FEDERAL UNIVERSITY OF TECHNOLOGY - PARANÁ
ACADEMIC DEPARTMENT OF ELECTRICAL ENGINEERING
ELECTRICAL ENGINEERING

RENATA DA COSTA RIBEIRO

A PERFORMANCE COMPARISON BETWEEN THE EXTENDED KALMAN FILTER AND DIFFERENT APPROACHES OF SIGMA-POINT KALMAN FILTERS

CURITIBA
2021

RENATA DA COSTA RIBEIRO

A PERFORMANCE COMPARISON BETWEEN THE EXTENDED KALMAN FILTER AND DIFFERENT APPROACHES OF SIGMA-POINT KALMAN FILTERS

UMA COMPARAÇÃO DE DESEMPENHO ENTRE O FILTRO DE KALMAN ESTENDIDO E DIFERENTES ABORDAGENS DO FILTRO DE KALMAN DE PONTO-SIGMA

Bachelor's Thesis presented to the Department of Automation and Electrical Engineering of the Federal University of Technology - Paraná as a partial requirement to earn the degree of Bachelor of Electrical Engineering.

Mentor: Dr. Victor Baptista Frencl
UTFPR

CURITIBA
2021

RENATA DA COSTA RIBEIRO

A PERFORMANCE COMPARISON BETWEEN THE EXTENDED KALMAN FILTER AND DIFFERENT APPROACHES OF SIGMA-POINT KALMAN FILTERS

Bachelor's Thesis presented to the Department of Automation and Electrical Engineering of the Federal University of Technology – Paraná as a partial requirement to earn the degree of Bachelor of Electrical Engineering.

Date of endorsement: 25/october/2021

João Bosco Ribeiro do Val
Doctor of Electrical Engineering in Control Systems
University of Campinas

Luiz Ledo Mota Melo Jr
Doctor of Electrical Engineering and Industrial Informatics
Federal University of Technology - Parana

Marcelo de Oliveira Rosa
Doctor of Electrical Engineering
Federal University of Technology - Parana

Victor Baptista Frencl
Doctor of Electrical Engineering
Federal University of Technology - Paraná

CURITIBA

2021

To my mother,

the strongest and kindest woman I have ever met.

ACKNOWLEDGMENTS

I would like to pay special thankfulness, warmth, and appreciation to the persons below who made my research successful and assisted me at every point to achieve my goal.

Special thanks to Mr. Mauro Zanella for believing in my capabilities, my supervisor at ZF Friedrichshafen AG, Dr.-Ing. Tomas Szabo for sharing his expertise and wisdom, as well as emphasizing the opportunity he gave me to integrating his team. I want to thank my colleagues at ZF, Mr. Luis Ribeiro, Mr. Gerhard Freudenthaler, Mr. Felix Mesmer, and Mr. Johannes Strohm, for their input and suggestions along the way, as well as for their patience and insights.

My professors and colleagues at UTFPR for their intellectual support, and for providing a congenial and stimulating academic environment. In particular, I want to thank my supervisor, Dr. Victor Baptista Frencl, for his time, patience, insight, and knowledge imparted to me over the last year.

My mother Aparecida de Lourdes da Costa, my grandmother Joana da Costa Amorim and my brothers Paulo Sérgio da Costa and Thiago Borges Ribeiro for their love and support. Without them, this project surely would not have the same significance.

Marcus Vinícius Reis de Morais, Marcelo Vinícius de Morais, Cleria Teresa Reis, Pierina Baretta de Morais, Pedro Henrique Reis de Morais, Margô Veneide de Morais and Victor Medeiros: thank you for always believing in me and for all support you gave me not only during this project, but also in life. And to all my wonderful friends and family: thank you for your friendship and support.

"Science never solves a problem without creating ten more." — G. B. Shaw

ABSTRACT

State estimation is an essential part of several engineering scenarios. Under the hypothesis of linear systems and additive Gaussian noise, the solution of the optimal estimation problem is the Kalman Filter (KF), formulated in the 1960s by Rudolph Emil Kalman. In the case of nonlinear systems, the sub-optimal algorithm called Extended Kalman Filter (EKF) linearizes the model using the first-order truncated Taylor series expansion around the known current state estimation. Depending on the system, these approximations often introduce large errors. Jacobian-free stochastic filtering methods have been used such as the sigma-point stochastic filters, which address these approximation problems with a sampling approach instead of the Jacobian formulation. In this thesis, the performance, benefits, and handicaps of the EKF and one approach of Sigma-Point Kalman Filter (SPKF) — a variant of the Unscented Kalman Filter (UKF) — are compared when applied to a vehicle localization system with sensors, i.e. GNSS and IMU. Each sensor has its own sampling time, resulting in delayed measurements that if not properly treated, may result in considerable estimation errors. For that reason, an approach of each stochastic filter in this work should consider these delays by reformulating the stochastic filter through the delay period. For industrial purposes, having a high-performance stochastic filter can improve the quality of autonomous driving software. The stochastic filters were implemented and compared in this project considering two scenarios: with simulated data and with a real data extracted from a test vehicle. For both, the comparison method used was the Mean Squared Error (MSE) and a computational cost analysis was done. In this case, tests of consistency were also performed, like the Durbin-Watson (DW) test and the Normalized Autocorrelation test. The filters have shown a similar performance for this application, but the Unscented Kalman Filter (UKF) variant algorithm have shown a slightly better performance, mainly for the states that are not measured by the sensors. For that reason, a hybrid algorithm with both filters — EKF and UKF — were proposed. This project is being developed at ZF Friedrichshafen AG in Friedrichshafen, Germany.

Keywords: Kalman Filter, Extended Kalman Filter, Sigma-Point Kalman Filter, estimation, performance.

RESUMO

A estimativa de estados é uma parte essencial em uma série de problemas de engenharia. Sob a hipótese de que o sistema é linear e possui ruído gaussiano, a solução ótima do problema de estimativa de estados é a implementação do Filtro de Kalman (KF), formulado nos anos 1960 por Rudolph Emil Kalman. No caso de sistemas não lineares, o Filtro de Kalman Estendido (EKF) lineariza o modelo utilizando a expansão da série Taylor de primeira ordem em torno da estimativa de estado atual. Dependendo do sistema, estas aproximações causam erros consideráveis. Dessa forma foi desenvolvido um método de filtragem estocástica que não utiliza a matriz jacobiana. Os filtros estocásticos que utilizam este método são denominados *Sigma-Point Kalman Filters* (SPKFs). Estes filtros abordam estes problemas de aproximação com técnicas de amostragem de pontos no lugar da formulação da matriz jacobiana. Nesta tese, o desempenho, benefícios e desvantagens do EKF e do SPKF – mais especificamente das variantes do Filtro de Kalman *Unscented* (UKF) – são comparados quando aplicados a um problema prático de localização de um veículo autônomo acoplado com sensores, e.g. GNSS e IMU. Cada sensor tem o seu próprio tempo de amostragem, resultando em medições com atrasos que, se não forem bem tratadas, poderão levar a erros de estimativa elevados, dependendo do cenário em questão. Por esta razão, este trabalho também faz uma abordagem dos filtros estocásticos mencionados, considerando os atrasos dos sensores e recalculando as estimativas defasadas. Para fins industriais, as vantagens de ter um filtro estocástico de alto desempenho podem melhorar a qualidade dos *softwares* de localização. Os filtros foram implementados e comparados neste projeto para dois cenários distintos: com dados simulados e com dados reais extraídos de um veículo de teste. Para ambos, o impacto do atraso dos sensores foram analisados e o método de comparação utilizado foi o Erro Quadrático Médio (EQM). Além disso, foi feita uma análise de custo computacional e testes de consistência foram realizados, como o teste de Durbin-Watson (DW) e o teste de Autocorrelação Normalizada. Os filtros mostraram um desempenho semelhante para esta aplicação, mas o algoritmo da variante UKF mostrou um desempenho ligeiramente melhor, principalmente para os estados que não são medidos pelos sensores. Por essa razão, um algoritmo híbrido, com ambos os filtros, foi proposto. Este projeto foi desenvolvido na multinacional ZF Friedrichshafen AG em Friedrichshafen, na Alemanha.

Palavras-chave: Filtro de Kalman, Filtro de Kalman Extendido, Sigma-Point Kalman Filter, estimativa, desempenho.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

KF	Kalman Filter
EKF	Extended Kalman Filter
SPKF	Sigma-Point Kalman Filter
UKF	Unscented Kalman Filter
CDKF	Central Difference Kalman Filter
SRKF	Square-Root Kalman Filters
UDKF	Upper-Diagonal Kalman Filter
SSUKF	Spherical-Simplex Unscented Kalman Filter
UDEKF	Upper-Diagonal Extended Kalman Filter
SRUKF	Square-Root Unscented Kalman filter
UT	Unscented Transformation
SMC	Sequential Monte Carlo Method
SIR	Sampling Importance Resampling
GPS	Global Positioning System
DGPS	Differential Global Positioning System
GNSS	Global Navigation Satellite Systems
IMU	Inertial Measurement Unit
ECU	Engine Control Unit
ACF	Autocorrelation Function
MSE	Mean Squared Error
DW	Durbin-Watson
ODE	Ordinary Differential Equations
KST	Kinematic Single-Track
ST	Single-Track

WB	Wheelbase
DoF	Degrees-of-freedom

LIST OF SYMBOLS

μ	Mean
σ	Standard deviation
$\in$	Is a member of
$\hat{\mathbf{x}}^{+}$	*a posteriori* estimate
$\hat{\mathbf{x}}^{-}$	*a priori* estimate
$\mathbf{P}_{\mathbf{x}}^{+}$	*a posteriori* covariance of the state estimated
$\mathbf{P}_{\mathbf{x}}^{-}$	*a priori* covariance of the state estimated
$\mathbf{K}$	Kalman gain
$\tilde{\mathbf{y}}$	Innovation
$\mathbf{q}$	Residual
$\mathbf{R}$	Measurement noise covariance
$\mathbf{Q}$	Process noise covariance
S_x	Latitudinal position
S_y	Longitudinal position
ψ	Yaw angle
$\dot{\psi}$	Yaw angle derivative
v	Speed
β	Slip angle

LIST OF ALGORITHMS

CONTENTS

1 INTRODUCTION

This chapter provides a brief overview of the main topic of this thesis, an application of stochastic filters in the autonomous driving field. After that, the objectives are discussed, and finally, the structure of this work is presented.

1.1 PROJECT DESCRIPTION

Prowess in vehicle technology with hardware, software, and services to offer mobility solutions are in an increasing demand. From climate changes to road safety issues, the challenges of mobility will not be solved by themselves. However, self-driving vehicles can be an important part of the solution. Automation is one of the main drivers of a new and safe era of mobility.

The control of autonomous vehicles is an example of an application that needs real-time and accurate estimates of the system states. Despite some vehicle states (e.g., position, velocity, attitude) that can be measured through a variety of sensors, the state estimation is required because the complete state is not always observable (OGATA, 2010) or it is not always directly measured in practical situations. In these cases, the data measured by the sensors improve the reliability of the estimation of the complete state of the vehicle dynamic model through a theoretical approach.

To deal with this problem, the Kalman Filter (KF) is an optimal recursive solution for discrete-time linear stochastic systems in state-space representation, that requires low computational processing. It is a recursive process that uses a set of equations and consecutive inputs to estimate the dynamics of an object (KALMAN, 1960).

However most practical problems have nonlinearities that limit the applications of the KF. To deal with this issue, the most common approach is the Extended Kalman Filter (EKF) which linearizes the mathematical model using the Taylor series expansion (CANUTO; OGATA, 2008). Based on this linearization method, the traditional linear KF equations can be adapted and applied to nonlinear systems (Mochnac; Marchevsky; Kocan, 2009).

In practice, the EKF has some disadvantages which directly affect the accuracy of the filter depending on the model and its nonlinearities. To address this problem Julier and Uhlman proposed the Unscented Kalman Filter (UKF). In many applications, the UKF demonstrated substantial performance gains in the context of state estimation for nonlinear stochastic systems (JULIER; UHLMANN, 2004).

In this thesis, the EKF and UKF — and their different approaches, explained further in the following Chapters — are compared, in terms of performance and computational complexity for the autonomous driving field of application. This comparison is done

using both simulated and real data.

1.1.1 SCOPE AND PROJECT DELIMITATION

In general, there are no generic or standard solutions for nonlinear problems, as every nonlinear system is unique and it has its particularities. For this reason, the advantages and disadvantages of the different solutions should be analyzed for each engineering application.

The knowledge about which solution is more suitable in terms of estimation accuracy and computational complexity is an essential part of the quality and safety of localization software for autonomous driving. As a global technology company that supplies systems for passenger and commercial vehicles, ZF Friedrichshafen AG, in Friedrichshafen, Germany, is where this project was developed, specifically within the Corporate Research and Development, in the Software Engineering and Digital Technologies department.

This work focuses on analyzing the behavior of a different approach of EKF and UKF, and comparing their performances in terms of estimation accuracy and computational complexity, by applying them to the state estimation problem in an autonomous vehicle. Additionally, in real applications, there is a time delay[1] in some of the sensor devices resulting in a processing time before the measurements are available. Without proper treatment, these delays cause large errors in certain scenarios. For that reason, this thesis also presents how these delays can impact the filter's performance and how it can be properly treated to avoid inaccuracies.

1.2 PROJECT PREMISES

Software developed to be applied in a real physical situation, in general, have their quality and reliability directly dependent on how they are loyal to the reality. In this case of application, to represent all the physical properties is an impossible task. The vector of all variables and parameters is countless. For that reason, it is necessary to do a trade-off between what is enough for software quality and validation and what can be simplified without losing important requirements like safety and software reliability.

In order to avoid a mathematical complexity incompatible with the undergraduate level of the student, the process of developing the filters had some assumptions beforehand:

- Simplified 4- and 7-Degrees-of-Freedom (DoF) state models are considered for the vehicle and consequently all the assumptions arising from the application of these models.

[1]The *time delay* will henceforth be called as *delay*.

- The vehicle's mass and the friction coefficient of the road are considered constant during the entire runtime process.

- The quantity of sensors used to measure the states of the vehicle is the same during the entire runtime process;

- It is considered that the sensors and process noises follow the Gaussian distribution.

- The delays of the sensors are the same for all time steps.

These assumptions were established together with the company's software development team.

1.3 RESEARCH OBJECTIVES

Compare and determine the suitable filter for the vehicle localization problem taking into account delayed measurements.

1.3.1 SPECIFIC OBJECTIVES

In order to achieve the main goal mentioned above in 1.3, the following specific objectives were identified:

- Analyze the behavior of the estimates and covariances of the standard stochastic filters when the delay is ignored.

- Analyze the accuracy of the EKF and UKF filters with and without compensation of delayed measurements.

- Analyze the impact of measurement delays on the EKF and UKF performances.

- Perform some consistency tests for practical situations, like the Durbin-Watson (DW) test and the autocorrelation of residuals and its histograms.

- Show the differences between the filters using simulations and real world sensor data.

1.4 METHODOLOGY

In order to obtain important results and answers about the problem presented in this thesis, an exploratory research was conducted and aims to reinforce the kno-

wledge, theories and the predictions to expand the understanding about the KF family of algorithms.

This study is done based on five main steps:

- Literature research: as the KF algorithm is a very important object of study in control theory, there are several works related to this field that explains the theory behind and present topics that uses the KF's theory.

- Implementation of the filters: based on the references presented in Chapter 2, the filters, their variants and auxiliary codes are implemented (e.g. functions and classes for the vehicle model, the sensors and codes for the numerical operations).

- Application of methods for performance analysis, error, comparison and consistency tests of the filters results: Mean Squared Error (MSE), Autocorrelation tests and Durbin-Watson (DW) test, explained in more detail further in this thesis.

- Simulation of the implemented filters: performance and error analysis, and consistency tests using generated data, to ensure the correct functionality of the implementation.

- Obtaining real parameters from the test vehicle and sensor data: performance and error analysis, and comparison of the filters taking into account the real data.

In addition to these methods, a theoretical study is done about the computational complexity of EKF and UKF, which the execution time is directly proportional to the state space dimension.

1.4.1 SCRIPTING LANGUAGE AND DEVELOPMENT ENVIRONMENT

This project was developed entirely using MATLAB®. The software was chosen due to availability and also for simulation purposes and a better understanding of the numerical and graphical results. All the implementations are done using Object-Oriented Programming.

1.5 THESIS OUTLINE

The main chapters of this thesis are organized as follows:

- **Chapter 2:** This chapter provides a literature review concerning the related work in the field of stochastic filters.

- **Chapter 3:** This chapter provides concepts of stochastic filters, presents the fun-

damentals of the KF and presents the stochastic filters addressed in this thesis for nonlinear systems.

- **Chapter 4:** This chapter presents the equations of the vehicle dynamics and measurement models. In addition, it presents how the filters behave within the incorporation of delayed measurements.

- **Chapter 5:** This chapter describes the methods and tools – mentioned in Section 1.4 – used to analyse and compare the stochastic filters in this thesis.

- **Chapter 6:** This chapter presents the results of the comparisons for a simulated and real scenarios, as well as provides a discussion about these results, limitations of the study and the reasons why the analyzed stochastic filters perform similar using the vehicle model with linear measurement equations.

- **Chapter 7:** This chapter presents the conclusions of the work. It provides an overview about the study and a concise synopsis about what was done along the study, the contributions of this research for the control and academic community and future work in the related field.

2 STATE OF THE ART

2.1 APPLICATIONS OF THE KALMAN FILTER

Thomas Bayes and his interpretation of the concept of probability, in which he specifies the notion of the prior and posterior probabilities by a recursive algorithm (BOX; TIAO, 2011), and Carl Friedrich Gauss with his approach for approximating the optimal probability densities in the Bayesian recursion by a Gaussian distribution (BRYC, 2012) and estimating state variables from noisy data using the least-squares method (TAYLOR; FRANCIS, 1988) are some of the examples of mathematicians and their essential contributions in the 19^{th} century that culminated in the development of advanced state estimators years later.

Thus Rudolph Emil Kalman and his approach to linear stochastic filtering (KALMAN, 1960) served as a result of this transformative process of ideas and concepts from many other scientists over the years for the field of stochastic filtering.

Nowadays, the KF in its various forms is a highly popular tool that is applied to a wide range of dynamic systems. From its original formulation to the present day, Kalman's theory has undergone several transformations and reformulations according to each field of application and system complexity. The first known application was released at NASA Ames Research Center in the 1960s bearing some studies for the control and navigation field of the Apollo mission.

Even though the first uses of the KF were in the aerospace field, the formulation was adapted to be applied in other areas. Some issues of the IEEE Transactions on Automatic Control have focused on its usage for the state estimation problem, covering diverse subjects such as cyber-physical systems (Ni et al., 2020), sensor networks and smart grids (He et al., 2020), tracking control of mobile robots (Liang et al., 2016), reactivity estimation in nuclear power plants (Mishra; Shimjith; Tiwari, 2019), geostationary satellite orbit determination (Xu et al., 2010), and other examples.

2.2 THE EVOLUTION OF KALMAN FILTER

The increasing studies around Kalman's original theory resulted in a scientific breakthrough for nonlinear problems. In (MERWE, 2004), it was demonstrated that the optimal solution for this problem requires a large number of hypothesis, that makes the optimal solution unfeasible. In these scenarios some suboptimal approximations have been proposed. Since the end of the 1960s, the EKF has become a standard formulation used in numerous applications because it relieves the effects of the nonlinearities of the dynamic system model. Another reason for its widespread use is the relative simplicity of understanding and implementation underlying the original theory.

The dissemination of the KF to a large scientific community and consequently the intensification of studies about this topic at NASA, MIT, and other scientific institutions make many researchers realize and report problems with the stability and computational difficulties of the stochastic filter in computational simulations. Numerically more stable and better-conditioned groups of algorithms were formulated in the literature: the Square-Root Kalman Filter (SRKF) and the Upper-Diagonal Kalman Filter (UDKF), as in (BELLANTONI; DODGE, 1967) and (Bierman, 1975), respectively. They have slowly gained acceptance and their superior performance was reported in (MCGEE et al., 1973). These algorithms have faced some resistance by the scientific community since they can be more complicated to implement.

The EKF, in its standard or square root formulations, linearizes the model using the first-order truncated Taylor series expansion around the current state vector. However, this filter has proven to be difficult to tune on some systems, even with the linearization technique which often introduces large accuracy errors. To deal with these flaws, Julier and Uhlman proposed the UKF, as mentioned earlier. This stochastic filter addresses these inaccuracies by using a deterministic sampling approach achieving the third-order truncated Taylor series expansion for any nonlinearity.

In 2004, Rudolph van der Merwe attempted to extend the varieties of the UKF and the Central Difference Kalman Filter (CDKF) into a common group of stochastic filters, called Sigma-Point Kalman Filters (SPKF), and apply them to other inference problems, such as dual estimation (NELSON, 2000) and parameter estimation (BECK; ARNOLD, 1977). In his Ph.D. thesis, (MERWE, 2004), Merwe also derives stable square root versions of the SPKFs, and extend their application to the machine learning field (ALPAYDIN, 2020).

Some released papers are comparing the performances between the varieties of KFs in terms of estimation accuracy and computational complexity in highly nonlinear applications, e.g. (Allotta et al., 2015), (MERWE, 2004), (Fiorenzani et al., 2008). The large part of these works shows a better performance for the UKF algorithm, but there are still open points in the literature in the case of autonomous vehicles, in which there are both linear and nonlinear equations, as will be presented in detail in Chapter 4. In this case, as previously mentioned, each application is unique and deserves to be analyzed individually.

2.3 THE INCORPORATION OF DELAYED MEASUREMENTS

In practical applications, as mentioned in Section 1.1, there can be a delay in some of the sensor devices when they are not synchronized. Some works in the 1990s started to consider the problem of designing a discrete-time KF to incorporate these delayed cycles (Andersen; Christensen; Ravn, 1994). Since that time, several methods were proposed, but the fusion of delayed measurements in a KF algorithm is not a trivial task.

If only a few measurements are incorporated into the delayed period, and if the computational burden of the filter is not critical, an optimal filter can be formulated by recalculating the estimates through the respective delay. In practical situations, this can often consume more computational time than expected, and to deal with this problem some methods propose the calculation of a correction term to be included in the correction step when the measurement arrives. This method, however, is optimal only under certain conditions (ALEXANDER, 1991). A modification introduced in (Larsen et al., 1998) proposes a method based on extrapolating the delayed measurements, in which the current measurement could be calculated and this value can be applied to the stochastic filtering algorithm. However, a dynamic system in which the states can achieve high variation rates in time, e.g. a car moving fast, represents a weak point of this approach.

Merwe also incorporates in the SPKF's algorithms a method to deal with the Global Positioning System (GPS) latencies. The SPKF framework allows the fusion of the delayed measurements by augmenting the state vector accordingly. As the delay rises, the number of augmented states should also increase, and then, the filter requires more memory and more computational power.

As it was shown, the fusion of delayed observations is still a problem and it needs to be addressed in the vehicle localization systems. Some of these solutions are only applicable in specific situations and under certain conditions. In fact, stochastic filters that somehow deal with these delays present more consistent and accurate estimates than those that ignore them (MERWE, 2004).

2.4 THE PARTICLE FILTER

From the end of the 20^{th} century, another group of approximate solutions for the Bayesian recursive algorithm, known as Sequential Monte Carlo Method (SMC) or Particle Filter (PF), have gained significance in the research community as these methods can be applied to general nonlinear and non-Gaussian problems and work for high dimensional systems because the filter and its computational complexity does not depend on the dimension of the state vector of the system (Gustafsson et al., 2002).

Different from the Gaussian methods that approximate the optimal probability densities in the Bayesian recursion by a Gaussian distribution, PFs represent these variables by empirical point mass approximations and update them recursively, generally using the Sampling Importance Resampling (SIR) method for state estimation problems (Tulsyan; Gopaluni; Khare, 2016).

Despite the many advantages of the PF and its growth in the academic community, some disadvantages justify the fact that they are not evaluated in this thesis. The first of them is that in the PF the points are chosen randomly, whereas in the UKF the points are chosen on the basis of a specific algorithm, known as Unscented Transformation. From this, two consequences can be pointed out: even for the same dynamics and for the same

input data, the PF can perform different behaviors on different runs and another problem is that the PF requires an abundant number of particles depending on the application – generally it needs to be much greater than the number of points in a UKF – and it makes the PF computationally expensive when compared to EKF and UKF.

3 THEORETICAL BACKGROUND

3.1 STOCHASTIC PROCESS

Dynamical systems can be described as deterministic or stochastic processes. A deterministic process has a 100% chance that the same input always generates the same output. In a stochastic process the dynamic system evolves randomly in time. This means that for the same input, there are different probabilities for each possible output, respecting some probability distribution (COLEMAN, 1974).

These different possibilities lead to random behavior that produces an imprecision when trying to affirm how the system will evolve. Consequently, there are uncertainties about the dynamics of the system.

In the field of engineering, aerospace industries exploring the solar system with robotic proxies capable of autonomous reasoning, face recognition systems for homeland security or counter-terrorism systems, and neural network based systems for urban soil classification are some examples that share the fact that they all operate in the real world and consequently have to deal with uncertainties (MERWE, 2004). In many cases, these technologies can achieve better results if they have a representation of the uncertainty associated with the variables needed for carrying out tasks (Gonçalves; Zattoni, 2017). One of the tools available to model these uncertainties is a process known as probabilistic inference.

Probabilistic inference for stochastic processes is a mathematical way that tries to find the intrinsic probability density of the noisy or incomplete observed data to estimate the states or parameters of a deterministic dynamic system (MERWE, 2004). The basis of this analysis are the concepts of conditional probability and the Bayes' theorem, (CAMBANIS, 2014) ,(JOYCE, 2019).

In summary, the probability gives the filter information about the distribution of the uncertainties. With this information, the filter will try to incorporate their effects and will try to get closer to the deterministic behavior of the system.

3.2 KALMAN FILTER

As previously mentioned in Section 1.1, KF is a stochastic filter and represents a typical example of a probabilistic inference problem for discrete-time linear systems in state-space representation. This filter assumes that all the variables that compound the estimated state vector, $\mathbf{x} \in \mathbb{R}^n$, are random and have a Gaussian probability distribution (BRERETON, 2014). Each element, x, of the state vector, $\mathbf{x}$, has its own mean μ and variance σ^2.

The stochastic dynamic state-space model can be written as a set of equations, as

follows:

$$\mathbf{x}_k = \mathbf{A}_k\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k, \tag{1a}$$

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \tag{1b}$$

where the state vector evolves in time from $k-1$ to k, according to a particular case of the stochastic process, called Markov process (DYNKIN, 1971). In this recursive case, the conditional probability distribution in all steps is $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, meaning that result of the next step depends only of the current step of the dynamic system.

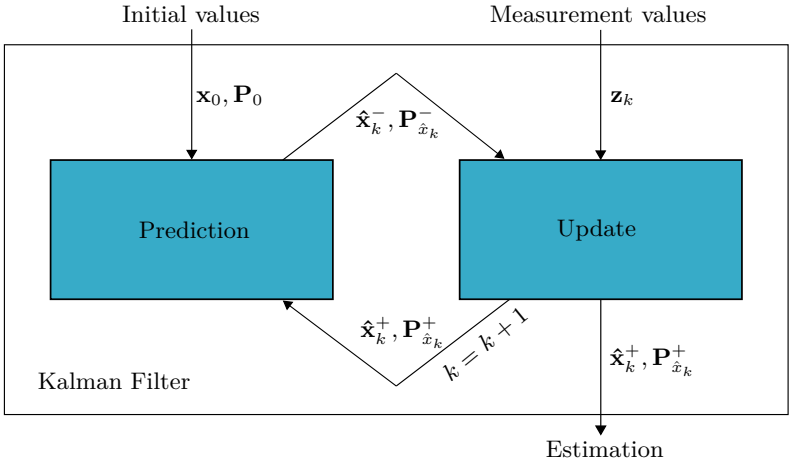
The transition matrix $\mathbf{A}_k \in \mathbb{R}^{n\times n}$ tells how the previous states interact to the current state and how the elapsed time changes the state vector. The matrix $\mathbf{B}_k \in \mathbb{R}^{n\times p}$ is the actuator matrix which is applied to the control input vector $\mathbf{u}_k \in \mathbb{R}^p$. The observations $\mathbf{y}_k \in \mathbb{R}^m$ are made according to the conditional probability density $p(\mathbf{x}_k|\mathbf{y}_k)$, where the matrix $\mathbf{H}_k \in \mathbb{R}^{m\times n}$ is the sensor matrix.

Last, the $\mathbf{w}_k \in \mathbb{R}^n$ and $\mathbf{v}_k \in \mathbb{R}^m$ represent the zero-mean Gaussian noise vectors with process noise covariance matrix $\mathbf{Q}_k \in \mathbb{R}^{n\times n}$ and measurement noise covariance matrix $\mathbf{R}_k \in \mathbb{R}^{m\times m}$, respectively, i.e. $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$ and $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)$.

Given the system equations, the filter can be used to estimate the joint probability distribution over the elements of the state vector in two main steps: prediction and update, for each time step; see Figure 1, where $\mathbf{x}_0$ and $\mathbf{P}_0$ are the initial conditions. The detailed computational and probabilistic origins of the KF can be found in (MAYBECK, 1990).

The state variables can be correlated, meaning that the likelihood of a particular state depends not only on the state itself but also on the values of other states. This correlation is defined by the covariance matrix (GREWAL, 2011). In summary, at each time step k the filter estimates the state vector $\hat{\mathbf{x}}_k$ and its uncertainty which is represented by the covariance matrix $\mathbf{P}_{\mathbf{x}_k} \in \mathbb{R}^{n\times n}$.

Figure 1 – *Simple block diagram of KF steps:* This diagram gives a basic idea of how the KF works. Basically, the filter operates in two steps with given initial values, $\mathbf{x}_0$ and $\mathbf{P}_0$: the filter predicts the state in the prediction step, incorporates the measurements that come directly from the sensors readings ($\mathbf{z}_k$), and uses it to update its prediction. At the end of each cycle, the filter outputs an estimated state and an associated covariance, representing the uncertainty. These estimated values are then used for the next step.



Source: Personal collection.

The prediction step uses the *a posteriori* estimate of the current state $\hat{\mathbf{x}}_{k-1}^+$ and current covariance $\mathbf{P}_{\mathbf{x}_{k-1}}^+$ to predict the *a priori* estimate of the next state $\hat{\mathbf{x}}_k^-$ and next covariance $\mathbf{P}_{\mathbf{x}_k}^-$, as shown in Equations (2a) and (2b), respectively.

$$\hat{\mathbf{x}}_k^- = \mathbf{A}_k \hat{\mathbf{x}}_{k-1}^+ + \mathbf{B}_k \mathbf{u}_k, \tag{2a}$$

$$\mathbf{P}_{\mathbf{x}_k}^- = \mathbf{A}_k \mathbf{P}_{\mathbf{x}_{k-1}}^+ \mathbf{A}_k^\intercal + \mathbf{Q}_k. \tag{2b}$$

If the measurement vector obtained from the sensor readings is available, the predicted results are refined with these measurements in the update step, as presented in Equations (3a) – (3e):

$$\hat{\mathbf{y}}_k^- = \mathbf{H}_k \hat{\mathbf{x}}_k^-, \tag{3a}$$

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \hat{\mathbf{y}}_k^-, \tag{3b}$$

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{x}_k}^- \mathbf{H}_k^\intercal (\mathbf{R}_k + \mathbf{H}_k \mathbf{P}_{\mathbf{x}_k}^- \mathbf{H}_k^\intercal)^{-1}, \tag{3c}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \tilde{\mathbf{y}}_k, \tag{3d}$$

$$\mathbf{P}_{\mathbf{x}_k}^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{\mathbf{x}_k}^-, \tag{3e}$$

where the vector $\tilde{\mathbf{y}}_k \in \mathbb{R}^m$ is the innovation that captures the discrepancy between the real ($\mathbf{z}_k$) and expected ($\hat{\mathbf{y}}_k^-$) measurements and the matrix $\mathbf{K}_k \in \mathbb{R}^{n \times m}$ represents the Kalman gain, which is optimal.

The Kalman gain is the relative weight given to the measurements and the prediction values. When its value is low, the filter places more weight on the model predictions and for high values the filter follows the most recent measurements more closely. More details about the Kalman gain effects can be found in (ZHANG; OLIVER, 2011).

In the update step the *a priori* estimate, $\hat{\mathbf{x}}_k^-$ and its covariance $\mathbf{P}_{\mathbf{x}_k}^-$, evolve to *a posteriori* results, $\hat{\mathbf{x}}_k^+$ and $\mathbf{P}_{\mathbf{x}_k}^+$. This new result is the best estimate of the filter for that time step and can be fed back into another cycle of prediction and update steps.

In-depth details about the whole probability knowledge behind the KF equations can be found in (PAPOULIS; PILLAI, 2001) and references therein.

3.3 KALMAN FILTER FOR NONLINEAR SYSTEMS

As mentioned in Section 1.1, there are few processes that can be represented by linear dynamic systems in the real world. This means that the KF, even though is an optimal tool for state estimation — based on the hypothesis mentioned in Section 3.2, it cannot be applied to most of the systems because the majority of dynamic systems are modeled by nonlinear equations.

After the original publication of KF, mathematical techniques were developed to adapt this stochastic filter to be used for nonlinear systems. This adaptation is necessary because one of the assumptions of the KF is that the random variables respect a Gaussian distribution. When a Gaussian is analytically fed through a nonlinear function, its output will not be a Gaussian, because nonlinear functions lead to non-Gaussian distributions (MERWE, 2004).

The best known motivation for these studies was the Apollo mission released at NASA Ames Research Center, as mentioned in Chapter 2. The variations of the filter were developed for this purpose. The EKF and UKF are two different stochastic filters with several aspects that are detailed in the next sections, including the way that both address the nonlinearities of the system: the EKF uses an analytical linearization method while the UKF uses a statistical linearization approach called the Unscented Transformation (UT).

For the next sections, it will be considered the following state-space model in Equations (4a) and (4b),

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k), \tag{4a}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{v}_k), \tag{4b}$$

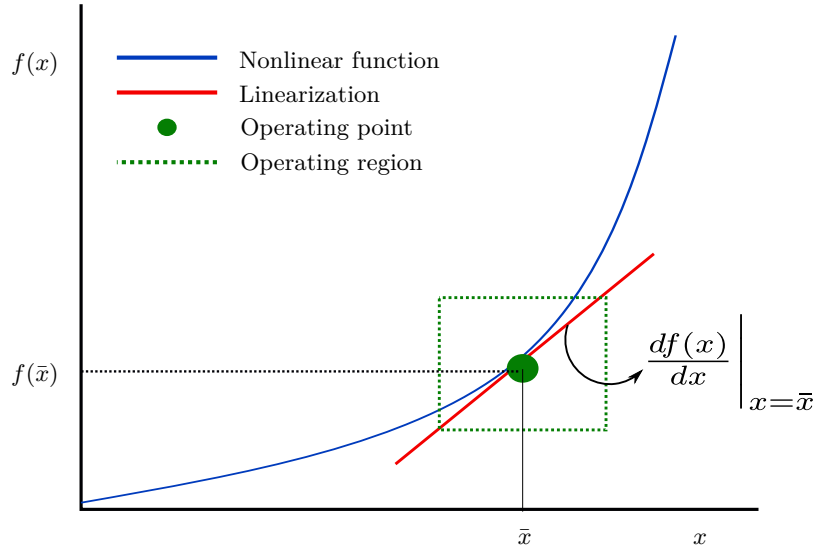
where $\mathbf{f}_k \in \mathbb{R}^n$ is the transition function and $\mathbf{h}_k \in \mathbb{R}^m$ is the observation function.

3.3.1 THE EXTENDED KALMAN FILTER

The EKF, as the name says, is an extension of the KF for nonlinear systems. The key concept in the EKF is the analytical linearization of a nonlinear system.

Linearizing a nonlinear system means determining an operating point $\bar{x}$ and finding a linear approximation to the nonlinear function $f(x)$ in the neighborhood of $\bar{x}$, called operating region. In two dimensions it means finding the tangent line to the function $f(x)$ when x equals $\bar{x}$ as shown in Figure 2.

Figure 2 – *Simple bidimensional example of linearizing a function at $\bar{x}$:* the result is a linear approximation (represented by the red line) to the nonlinear function (represented by the blue line) in the neighborhood of $\bar{x}$ delimited by the green dashed line that represents the region where the linearization is close to the original function. In 2-D, this means finding the tangent line to the function $f(x)$, when x equals $\bar{x}$.



Source: Personal collection.

Mathematically, this is done by taking the Taylor series expansion of the function $f(x)$, as follows:

$$f(x) = \sum_{a=0}^{\infty} \frac{1}{a!} f^{(a)}(\bar{x})(x - \bar{x})^a, \tag{5a}$$

$$= f(\bar{x}) + f'(\bar{x})(x - \bar{x}) + f''(\bar{x})\frac{(x - \bar{x})^2}{2!} + f'''(\bar{x})\frac{(x - \bar{x})^3}{3!} + ..., \tag{5b}$$

where $f(x)$ is the given function and the polynomial order is equal to a.

The Taylor series is the way of representing a function as an infinite sum, where its terms are calculated from the function derivatives at the operating point. For linearization, it is assumed that the operation region is very close to the operation point, meaning that the terms with higher orders ($a > 1$) are too small. For that reason, all of these higher order terms are disconsidered.

For the EKF, the linearization occurs around the most recent estimate of the state. For the state model, the system is linearized around the posterior estimate of the previous state, $\hat{\mathbf{x}}_{k-1}^{+}$, and for the measurement model the linearization occurs around the prediction of the current state, $\hat{\mathbf{x}}_{k}^{-}$. The linearized dynamical system in state-space representation can be written as a set of equations,

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) \approx \mathbf{f}(\hat{\mathbf{x}}_{k-1}^{+}, \mathbf{u}_k, 0) + \mathbf{F}_{\hat{\mathbf{x}}_{k-1}}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^{+}) + \mathbf{w}_k, \tag{6a}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k) \approx \mathbf{h}(\hat{\mathbf{x}}_{k}^{-}, 0) + \mathbf{H}_{\hat{\mathbf{x}}_k}(\mathbf{x}_k - \hat{\mathbf{x}}_{k}^{-}) + \mathbf{v}_k, \tag{6b}$$

where, $\mathbf{F}_{\hat{\mathbf{x}}_{k-1}} \in \mathbb{R}^{n \times n}$ and $\mathbf{H}_{\hat{\mathbf{x}}_k} \in \mathbb{R}^{m \times n}$ are the Jacobian matrices of the system. In vector *calculus*, a Jacobian matrix is the matrix whose elements are first-order partial derivatives of a vector valued function, generically denoted as $\mathbf{J} \in \mathbb{R}^{b \times c}$, as follows:

$$\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \dfrac{\partial \mathbf{f}_1}{\partial x_1} & \cdots & \dfrac{\partial \mathbf{f}_1}{\partial x_c} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial \mathbf{f}_b}{\partial x_1} & \cdots & \dfrac{\partial \mathbf{f}_b}{\partial x_c} \end{bmatrix}. \tag{7}$$

The Jacobian matrix can be interpreted to how fast each output of the function is changing along each input dimension. Details about the Taylor series and the Jacobian matrix can be found in (TANIZAKI, 1993) and (QUARTERONI; SACCO; SALERI, 2007), respectively.

After computing the Jacobians, it is necessary to insert them into the standard KF equations. The prediction step is written as shown in Equations (8a) and (8b):

$$\hat{\mathbf{x}}_{k}^{-} = \mathbf{f}(\hat{\mathbf{x}}_{k-1}^{+}, \mathbf{u}_k, \mathbf{w}_k), \tag{8a}$$

$$\mathbf{P}_{\mathbf{x}_k}^{-} = \mathbf{F}_{\hat{\mathbf{x}}_{k-1}^{+}} \mathbf{P}_{\mathbf{x}_{k-1}}^{+} \mathbf{F}_{\hat{\mathbf{x}}_{k-1}^{+}}^{\intercal} + \mathbf{Q}_k, \tag{8b}$$

and the update step is written as shown in Equations (9a) – (9e):

$$\hat{\mathbf{y}}_{k}^{-} = \mathbf{h}(\hat{\mathbf{x}}_{k}^{-}, \mathbf{v}_k), \tag{9a}$$

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \hat{\mathbf{y}}_{k}^{-}, \tag{9b}$$

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{x}_k}^{-} \mathbf{H}_{\hat{\mathbf{x}}_k^{-}}^{\intercal} (\mathbf{R}_k + \mathbf{H}_{\hat{\mathbf{x}}_k^{-}} \mathbf{P}_{\mathbf{x}_k}^{-} \mathbf{H}_{\hat{\mathbf{x}}_k^{-}}^{\intercal})^{-1}, \tag{9c}$$

$$\hat{\mathbf{x}}_{k}^{+} = \hat{\mathbf{x}}_{k}^{-} + \mathbf{K}_k \tilde{\mathbf{y}}_k, \tag{9d}$$

$$\mathbf{P}_{\mathbf{x}_k}^{+} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_{\hat{\mathbf{x}}_k^{-}}) \mathbf{P}_{\mathbf{x}_k}^{-}. \tag{9e}$$

Note that there is an important difference between the EKF equations and the KF equations: in the prediction and update steps, the nonlinear models are still being used to propagate the state estimation and to compute the innovation. That is because the linearized model exactly coincides with the nonlinear model at the operating point (QUARTERONI; SACCO; SALERI, 2007).

3.3.2 LIMITATIONS OF THE EKF

While the EKF performs well for many practical problems and it is relatively simple to implement and to understand when compared to the UKF, see Section 3.3.3, some limitations are important to analyze when deciding if the filter is the most suitable algorithm for a specific real problem.

The previous section explained that the EKF works by linearizing the state and the observation models to update the state estimation at each time step, using the first-order truncated Taylor series expansion. The first problem is that this linearization method does not take into account the fact that the operating point is a random variable, meaning that it has an inherent probabilistic spread, captured by the uncertainty. A practical example that shows the consequences of this linearization method can be found in (MERWE, 2004).

The second problem is that a linearized model is just a local approximation of the true nonlinear function. The difference between them is the linearization error, and this value depends on two factors. The first is how nonlinear the function is: if the function varies slowly, the linear approximation is a good method for that case, but if the function varies quickly the linear approximation introduces large errors trying to capture the true shape of the function. The second is that the further away from the operating point the linear approximated function is, the more likely the local linearity assumption fails, meaning that the effects of the other terms of the Taylor series expansion become significant (TANIZAKI, 1993).

A good example to show how these linearization errors can affect the EKF accuracy is the nonlinear transformation from polar coordinates, r and θ, to Cartesian coordinates, x and y. This example represents a problem for the observation model because many real-world sensors (e.g., Sound Navigation and Ranging (SONAR), Radio Detection And Ranging (RADAR), Light Detection and Ranging (LIDAR)) return polar measurement data that are related with the Cartesian position of the target through the following nonlinear transformation

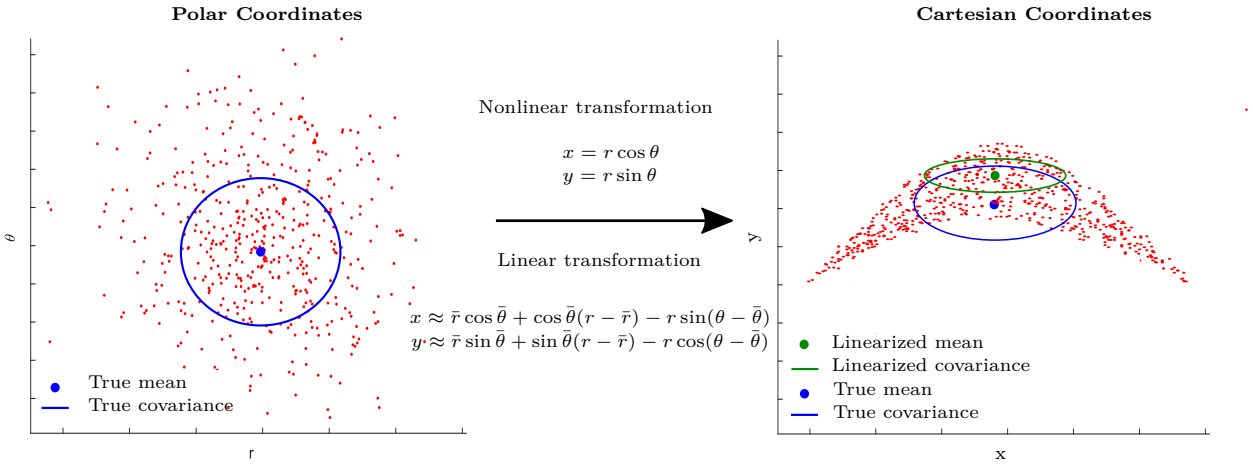
$$x = r\cos\theta, \tag{10a}$$

$$y = r\sin\theta, \tag{10b}$$

where r is the range and θ is the bearing angle of the target in the polar coordinates and x and y are the position in Cartesian coordinates of the target.

As shown in Figure 3, the linearization using a first order Taylor series truncation becomes very inaccurate for any significant bearing error. The linearized mean goes to a very different place relatively from the true mean, and the linearized covariance underestimates the spread of the true mean along the y dimension being overconfident about it.

Figure 3 – *Linear approximation from polar to cartesian coordinates:* Optimal vs. EKF transformation result on polar-to-Cartesian transformation experiment. The red dots indicate the noisy polar coordinate measurements coming from a sensor. The true covariance and mean of this distribution is indicated by the blue dot and the ellipses. The EKF calculated (linearized approximation) mean and covariance are indicated in green.



Source: Personal collection.

Since the next state depends on the current state, in some cases the linearization error is propagated and increased at the point that the estimator diverges completely. This can be a huge safety problem to deal with, not only for self-driving cars, but for all applications that requires this transformation. The UKF can solve some of those problems.

3.3.3 UNSCENTED KALMAN FILTER

The UKF is also a variation of the traditional KF, formulated by Julier and Uhlman as an alternative approach to nonlinear systems. In this variation, the stochastic processes still follow a Gaussian distribution, but instead of using an analytical linearization method, the UKF passes the probability distribution through the nonlinear functions using the UT. This method is based on the principle that it is easier to approximate a probability distribution than an arbitrary nonlinear function (JULIER; UHLMANN, 2004).

Assuming that $\mathbf{x}$ is a n dimensional random variable with a mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P_x}$ and the propagation of this variable $\mathbf{x}$ is given through an arbitrary nonlinear function,

$$\mathbf{y} = \mathbf{g}(\mathbf{x}), \tag{11}$$

the UT method can be described in three steps, as follows:

1. A set of deterministic points are chosen. Each point has its weight w and value $\mathcal{X}$.

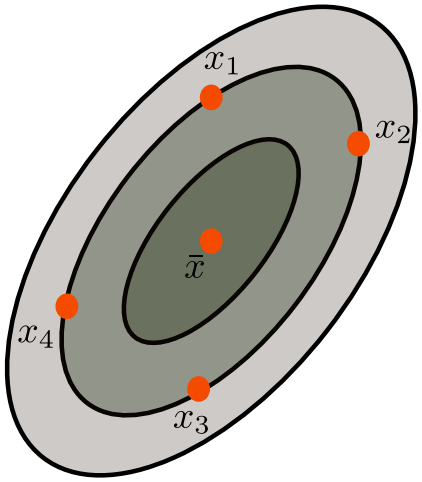
2. Each of these selected points are evaluated by the nonlinear function producing a new set of points.

3. This step is the computation of the weighted mean and covariance of transformed sigma-points that gives a good approximation of the mean and covariance of the true distribution.

Each of these steps are detailed below.

In general, for a n dimensional probability distribution function, $2n+1$ number of deterministic weighted samples, called sigma-points, $S = \{w, \mathcal{X}\}$, are required. One for the mean and the rest is distributed symmetrically around the mean (symmetrical form), as shown in Figure 4.

Figure 4 – *Sigma-points captured following the symmetrical form:* for a two dimensional probability distribution five sigma-points are obtained. One for the mean ($\bar{x}$), and the others (x_1, x_2, x_3 and x_4) symmetrically distributed around the mean.



Source: Personal collection.

The place where the weighted sigma-points should be is determinate through the following equations:

$$\mathcal{X}'_i = \bar{\mathbf{x}} \qquad w'_i = \frac{\kappa}{n+\kappa} \qquad i = 0, \tag{12a}$$

$$\mathcal{X}'_i = \bar{\mathbf{x}} + \left(\sqrt{(n+\kappa)\mathbf{P}_\mathbf{x}}\right)_i \qquad w'_i = \frac{1}{2(n+\kappa)} \qquad i = 1, ..., n, \tag{12b}$$

$$\mathcal{X}'_i = \bar{\mathbf{x}} - \left(\sqrt{(n+\kappa)\mathbf{P}_\mathbf{x}}\right)_i \qquad w'_i = \frac{1}{2(n+\kappa)} \qquad i = n+1, ..., 2n, \tag{12c}$$

where w'_i is the weight of the i-th sigma-point and the sum of the weights of all points is equals to 1 ($\sum_{i=0}^{2n} w'_i = 1$). The parameter κ can scale the sigma-points away or towards from the mean $\bar{\mathbf{x}}$ and it can assume any value. The square-root of $\mathbf{P}_\mathbf{x}$ matrix is calculated using the Cholesky factorization method that, in summary, is the decomposition in the

form $\mathbf{P} = \mathbf{S}\mathbf{S}^\intercal$. The matrix $\mathbf{S}$ is a lower triangular matrix with real and positive diagonal entries, called the Cholesky factor of the matrix $\mathbf{P}$ (LYCHE; MERRIEN, 2014).

One of the properties of the UT is that the radius of the sphere that cover all the sigma-points depends on the dimension of n. Related to the Equation (12), the distance of each sigma-point from the mean, i.e. $|\mathcal{X}'_i - \bar{\mathbf{x}}|$, is proportional to $\sqrt{(n+\kappa)}$. Without a proper value of κ this can be a problem especially when the nonlinearities of the system are high, since there is the possibility of sampling non-local points.

For Gaussian distributions, setting $\kappa = 3 - n$ is a good decision, because it cancels the effects of n (MERWE, 2004). However, this choice generates another problem: with a negative value of κ there is a possibility that the covariance calculated in the third step becomes non-positive definite and the weight w'_0 will be also negative. To prevent this issue, some scaling parameters are used to recalculate the points and weights of the Equation (12), as follows:

$$\mathcal{X}_i = \mathcal{X}'_0 + \alpha(\mathcal{X}'_i - \mathcal{X}'_0), \tag{13}$$

$$w_i^{(\mathrm{m})} = \begin{cases} w'_0/\alpha^2 + (1 - 1/\alpha^2), & i = 0, \\ w'_i/\alpha^2, & i = 1, ..., 2n, \end{cases} \tag{14}$$

$$w_i^{(\mathrm{c})} = \begin{cases} w'_0/\alpha^2 + (1 - 1/\alpha^2) + (1 - \alpha^2 + \beta), & i = 0, \\ w'_i/\alpha^2, & i = 1, ..., 2n, \end{cases} \tag{15}$$

where the superscripts m and c, represents the words 'mean' and 'covariance', respectively, α is the new sigma-point scaling parameter that controls the bounds of the distribution of the points and should be a small number to avoid non-local sampling ($0 < \alpha \leq 1$), and β is a positive weighting parameter that affects the calculation of the covariance by increasing some prior knowledge about higher order errors (MERWE, 2004). For Gaussian distributions, a good value for β is 2, according to (Julier, 2002).

The second step of the UT process is to propagate each sigma-point through the nonlinear function to get another set of points

$$\mathcal{Y}_i = g(\mathcal{X}_i), \quad i = 0, ..., 2n, \tag{16}$$

and finally, with the new set of sigma-points, it is possible to recombine them and find the approximated mean, covariance and cross-covariance — that represents the correlation between the observations and states — of the output:

$$\bar{\mathbf{y}} \approx \sum_{i=0}^{2n} w_i^{(\mathrm{m})} \mathcal{Y}_i, \tag{17a}$$

$$\mathbf{P}_\mathbf{y} \approx \sum_{i=0}^{2n} w_i^{(\mathrm{c})} (\mathcal{Y}_i - \bar{\mathbf{y}})(\mathcal{Y}_i - \bar{\mathbf{y}})^\intercal, \tag{17b}$$

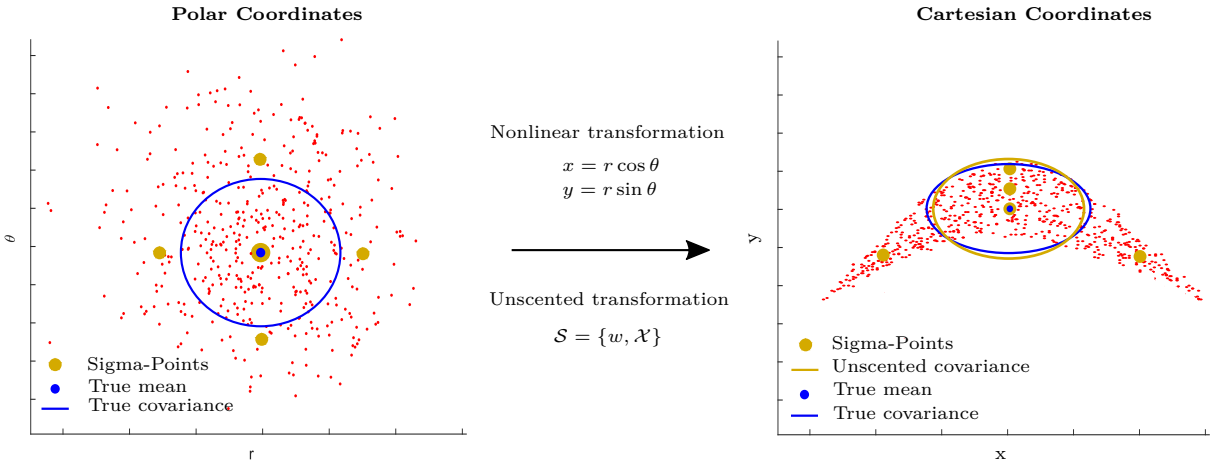
$$\mathbf{P}_\mathbf{xy} \approx \sum_{i=0}^{2n} w_i^{(\mathrm{c})} (\mathcal{X}_i - \bar{\mathbf{x}})(\mathcal{Y}_i - \bar{\mathbf{y}})^\intercal. \tag{17c}$$

Revisiting the example of Section 3.3.2, where polar coordinates were transformed into Cartesian coordinates, it is possible to see how the UT performs, as shown in Figure 5.

The dimension of the input distribution in this example equals two. Consequently, five sigma-points are required: one for the mean, and other four symmetrically distributed around the mean, as explained earlier. The new mean and covariance are computed from the transformed sigma-points and almost matches the true mean and covariance.

When compared to the result of the linear transformation method, used in the EKF, shown in Figure 3, the UT gives a better and more consistent approximation of the output probability distribution but with an increased computational cost, explained by the calculation of the sigma-points at each time step (Julier, 2002).

Figure 5 – *UT from polar to Carterian cooordinates:* Optimal vs. UKF approximation for estimating the statistics of a Gaussian random variable that undergoes a polar-to-Cartesian coordinate transformation. The red dots represent the noisy polar coordinate measurements coming from a sensor. The true covariance and mean of this distribution is indicated by the blue dot and ellipse, respectively. The UKF result is given by the sigma-points and covariance represented in orange.



Source: Personal collection.

Now that the UT was presented, its steps can be applied into the UKF with additive-noise that follows the same prediction-correction pattern as the other stochastic filters presented (KF and EKF). First, the sigma-points are calculated centered around the *a posteriori* estimate from time $k-1$ at each time step, as shown in Equations (18a) – (18d):

$$\mathcal{X}'_{0,k-1} = \hat{\mathbf{x}}^{+}_{k-1}, \qquad i = 0 \tag{18a}$$

$$\mathcal{X}'_{i,k-1} = \hat{\mathbf{x}}^{+}_{k-1} + \sqrt{(n+\kappa)\mathbf{P}^{+}_{k-1}}, \qquad i = 1, ..., n \tag{18b}$$

$$\mathcal{X}'_{i,k-1} = \hat{\mathbf{x}}^{+}_{k-1} - \sqrt{(n+\kappa)\mathbf{P}^{+}_{k-1}}, \qquad i = n+1, ..., 2n \tag{18c}$$

$$\mathcal{X}_{i,k-1} = \mathcal{X}'_{0,k-1} + \alpha(\mathcal{X}'_{i,k-1} - \mathcal{X}'_{0,k-1}), \qquad i = 0, ..., 2n. \tag{18d}$$

In the prediction step, the sigma-points are propagated through the nonlinear system and a new set of points are computed for the predicted state at time k. Then, this step is written as shown in Equation (19a) — (19c):

$$\mathcal{X}_{i,k} = \mathbf{f}(\mathcal{X}_{i,k-1}, \mathbf{u}_k, 0), \qquad i = 0, ..., 2n+1 \tag{19a}$$

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2n} w_i^{(\mathrm{m})} \mathcal{X}_{i,k}, \tag{19b}$$

$$\mathbf{P}_{\mathbf{x}_k}^- = \sum_{i=0}^{2n} (w_i^{(\mathrm{c})} (\mathcal{X}_{i,k} - \hat{\mathbf{x}}_k^-)(\mathcal{X}_{i,k} - \hat{\mathbf{x}}_k^-)^\intercal) + \mathbf{Q}_k. \tag{19c}$$

where $\mathbf{Q}_k$ is being added to the covariance of the transformed sigma-points to get the final result of the predicted covariance at each time step. The way to calculate the estimated states is different when the process noise is not additive, as can be found in (MERWE, 2004).

The update step takes the set of sigma-points calculated in the prediction step and applies them into the measurement model to get another set of points for the predicted observation, as shown in Equations (20a) – (20h):

$$\mathcal{Y}_{i,k} = \mathbf{h}(\mathcal{X}_{i,k}, 0), \qquad i = 0, ..., 2n+1 \tag{20a}$$

$$\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2n} w_i^{(\mathrm{m})} \mathcal{Y}_{i,k}, \tag{20b}$$

$$\mathbf{P}_{\tilde{\mathbf{y}}_\mathbf{k}} = \sum_{i=0}^{2n} (w_i^{(\mathrm{c})} (\mathcal{Y}_{i,k} - \hat{\mathbf{y}}_k^-)(\mathcal{Y}_{i,k} - \hat{\mathbf{y}}_k^-)^\intercal) + \mathbf{R}_k, \tag{20c}$$

$$\mathbf{P}_{\mathbf{x}_k\mathbf{y}_k} = \sum_{i=0}^{2n} w_i^{(\mathrm{c})} (\mathcal{X}_{i,k} - \hat{\mathbf{x}}_k^-)(\mathcal{Y}_{i,k} - \hat{\mathbf{y}}_k^-)^\intercal, \tag{20d}$$

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{x}_k\mathbf{y}_k} \mathbf{P}_{\tilde{\mathbf{y}}_\mathbf{k}}^{-1}, \tag{20e}$$

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \hat{\mathbf{y}}_k^-, \tag{20f}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \tilde{\mathbf{y}}_k, \tag{20g}$$

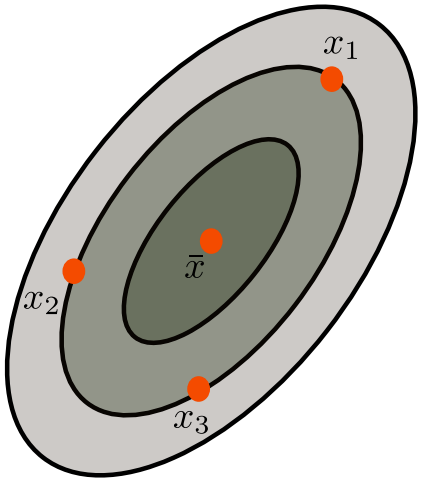
$$\mathbf{P}_{\mathbf{x}_k}^+ = \mathbf{P}_{\mathbf{x}_k}^- - \mathbf{K}_k \mathbf{P}_{\tilde{\mathbf{y}}_\mathbf{k}} \mathbf{K}_k^\intercal. \tag{20h}$$

Some approaches that are being developed for this filter are intended to improve its performance in many aspects. One example is the additive noise presented above in Equations (19c) and (20c). The additive way to incorporate the noises reduces the number of calculations in each iteration of the filter when compared to the original formulation of the UKF proposed by Julier and Uhlman (Castrejon Lozano et al., 2008), where the noises were not additive and the mean and the covariances were part of an augmented vector. This reduces significantly its computational load and it is mainly used in the UT process on the calculation and propagation of the sigma-points.

In order to also reduce the computational cost of the UT process, a variation on the way of capturing sigma-points was developed to be applied in the UKF. This selection

strategy defines a minimum set of points located in a hyper-sphere[1]. Assuming again that $\mathbf{x}$ is a n dimensional random variable with a mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{\mathbf{x}}$, instead of using $2n+1$ sigma-points for a n-dimensional space, the spherical-simplex form uses $n+2$ points, as illustrated in Figure 6 for a two-dimensional system.

Figure 6 – *Sigma-points captured following the spherical-simplex form:* for a two dimensional probability distribution four sigma-points are captured. One for the mean ($\bar{x}$), and the others (x_1, x_2 and x_3) distributed in a hyper-sphere.



Source: Personal collection.

The radius of the sphere that bounds the points is also proportional to $\sqrt{n}$, like in the symmetrical form when κ is 0. The weights of the sigma-points are calculated respecting the equation:

$$w_i' = \frac{(1 - w_0')}{(n+1)}, \qquad i = 1, ..., n+2, \tag{21}$$

where the value of w_0 has to be chosen according to the interval $[0, 1]$. To calculate where the sigma-points are, first, it is necessary to calculate $\mathcal{Z}_i$ that is i-th column of the spherical-simplex sigma-point matrix, initialized as

$$\mathcal{Z}_0^1 = [0], \ \mathcal{Z}_1^1 = \left[-\frac{1}{\sqrt{2w_1}} \right] \text{ and } \mathcal{Z}_2^1 = \left[\frac{1}{\sqrt{2w_1}} \right] \tag{22}$$

[1]Hyper-sphere is the set of points at a constant distance from a given point called its centre.

and expanded for $j = 2, ..., n$ according to

$$\mathcal{Z}_i^j = \begin{cases} \begin{bmatrix} \mathcal{Z}_0^{j-1} \\ 0 \end{bmatrix}, & i = 0 \\ \begin{bmatrix} \mathcal{Z}_i^{j-1} \\ -\frac{1}{\sqrt{j(j+1)w_1}} \end{bmatrix}, & i = 1, ..., j \\ \begin{bmatrix} 0_{j-1} \\ -\frac{j}{\sqrt{j(j+1)w_1}} \end{bmatrix}, & i = j+1, \end{cases} \tag{23}$$

and with this vector calculated, it is possible to determinate where the sigma-points should be placed using the following relation:

$$\mathcal{X}_i' = \bar{\mathbf{x}} + \mathcal{Z}_i \sqrt{\mathbf{P}_\mathbf{x}}, \qquad i = 0, ..., n+1, \tag{24}$$

where $\sqrt{\mathbf{P}_x}$ is also calculated by the Cholesky factorization method. In order to use the advantages of the scaling parameters, the previous places and weights of the points are transformed through the Equations (13) — (15).

The next steps of the UT using the spherical-simplex form remains exactly the same was previously presented for the symmetrical form, passing the points through the nonlinear function, recalculating them and using the new set to calculate the mean and the covariance.

This approach for the UT to be used in the UKF algorithm is more recommended for real-time systems, due to its low computational cost when compared with the symmetrical form. The UKF with the spherical-simplex approach is called Spherical-Simplex Unscented Kalman Filter (SSUKF).

3.3.4 SQUARE-ROOT STOCHASTIC FILTERS

Back in the 1960s, in the first applications of the KF algorithms, it was reported that very often the estimations of the KF were clearly diverging from the expected theoretical behavior. An example was that the key property that needs to be maintained is the symmetry and positive definiteness of the covariance matrix $\mathbf{P}$ in all time steps, but eventually negative variances were identified, which is theoretically impossible (BELLANTONI; DODGE, 1967).

This kind of problem was eventually caused by some roundoff errors that can affect the stochastic filters' perfomance directly. For that reason alternative implementation methods that are more robust when related to roundoff errors were developed and extensive knowledge in this field has been accumulated. More details about how these roundoff errors can degrade the filters results can be found in (GREWAL; ANDREWS, 2001).

It was shown in literature that better numerical stability, accuracy, and efficiency for the filters results can be reached when the covariance matrices are in their square-root decomposition forms. The two popular choices of the square-root form are the Cholesky decomposition

$$\mathbf{P} = \mathbf{S}\mathbf{S}^{\intercal}, \tag{25}$$

as mentioned in Section 3.3.3, and the modified Cholesky decomposition known as Upper-Diagonal (UD) decomposition

$$\mathbf{P} = \mathbf{U}\mathbf{D}\mathbf{U}^{\intercal}, \tag{26}$$

where $\mathbf{U}$ is an upper triangular matrix and $\mathbf{D}$ is a diagonal matrix.

Both methods factorize the covariance matrix into a set of matrices. These new factored matrices are propagated through the filter instead of the covariance matrix itself and for each of these choices, all covariance matrices are represented in the same form, and the equations of the KFs are obtained once again according to these factorizations.

In theory, both EKF and UKF can be written using both forms of decomposition. However, in this thesis, it will be approached in the following way: $\mathbf{U}$ and $\mathbf{D}$ factors are used in the EKF algorithm and the $\mathbf{S}$ factor is used in the UKF equations.

The EKF algorithm that uses the UD decomposition is called Upper-Diagonal Extended Kalman Filter (UDEKF). The Thornton-Bierman UD algorithm is a pair of an elegant formulation that do the evolution of the modified Cholesky factors $\mathbf{U}$ and $\mathbf{D}$ in time. Thornton is an algorithm for the prediction step and Bierman is an algorithm for the update. More details about the equations behind these two algorithms can be find in (GREWAL; ANDREWS, 2001). The UDEKF algorithm is presented in Appendix A.

The square-root stochastic filtering technique is the alternative algorithm to the UKF. The algorithm is called as Square-Root Unscented Kalman Filter (SRUKF) and it is presented in Appendix B.

4 STATE AND MEASUREMENT MODELS

Up to this point, it has been presented how stochastic filters based on the KF work and its behavior, each one of them with its advantages and disadvantages. However in practice, as already mentioned in Section 1.1, determining which approach is more appropriate depends on the nonlinear system and its characteristics. In other words, the advantages and disadvantages of the different solutions should be analyzed for each specific engineering application.

4.1 FIELD OF APPLICATION

The journey of autonomous driving technologies starts with some early advances in Europe in the 1980s. Scientists have been working on self-driving cars for over 40 years. Nowadays automotive industries are, in general, more deeply interested in this topic since driverless cars are looking more likely to become a reality. With this, it comes some benefits, including increased personal safety, time-saving for drivers, mobility for nondrivers, decreased environmental harm, and reduced transportation costs. It will also lead to drastic shifts in value chains, profit pools, and needed capabilities — for example, software expertise or cybersecurity — that could introduce new industries within the automotive sector.

As a big company in the automotive sector, ZF Friedrichshafen AG looks for this kind of technology, not only for commercial and passenger vehicles, but for solutions off-highway that reduces costs and makes materials transporting processes more efficiently, helping to optimize the company's supply chain and internal logistics operations.

This is one of the reasons why the work developed in this thesis is part of a project of the Software Engineering and Digital Technologies department to automate an eletric VOLK Tugger vehicle, model EFZ25N, which transports parts between the company's plants in Friedrichshafen, Germany. Figure 7 shows a picture of the vehicle.

Figure 7 – *Eletric ZF's VOLK Tugger:* the vehicle is coupled with a red trailer which has a 2-axle turntable steering.

Source: Personal collection.

4.1.1 VEHICLE MODEL

To deal with physical problems, with reference on a real situation, a representation through a mathematical model is required. It describes the behaviour of the vehicle when relevant parameters are given through a set of Ordinary Differential Equations (ODEs). Creating a suitable vehicle model is essential for a model-based control development.

In this thesis the vehicle is represented by the Kinematic Single-Track (KST) and Single-Track (ST) models in a state space representation. Both of them model the vehicle with two wheels, where the front and rear pair of the wheels are joined in a single front wheel and a single rear wheel. For that reason, both of these models are also known as "bicycle" models.

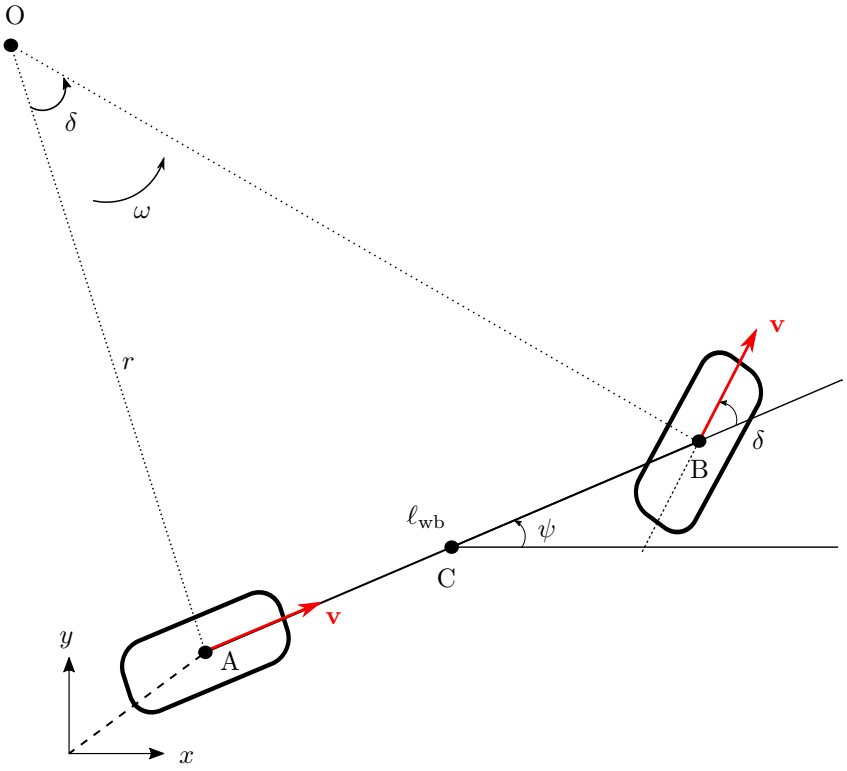
The "bicycle" models have been used as a convenient approach for representing vehicles because of their simplicity and a satisfactory representation of the nonholonomic[1] constraints of a car. The theoretical background about these models like their underlying dynamics and physical properties as well as more complex models, e.g. four-wheel model and multi-body models, can be found in detail in (ALTHOFF; KOSCHI; MANZINGER, 2017).

4.1.1.1 KINEMATIC SINGLE-TRACK MODEL

The KST model, as mentioned before, is a mathematical representation of the vehicle with two wheels, as shown in Figure 8 from a top view. Under certain assumptions described below, the KST model for the vehicle motion can be developed. Its equations are purely given by kinematic relationships, meaning that it does not consider kinetic forces.

[1]These might be constraints that are expressed in terms of coordinate velocities that are not derivable from coordinate constraints or constraints not represented as an equation at all (TALAMUCCI, 2019).

Figure 8 – *Kinematic Single-Track model:* it represents the vehicle with two wheels in a geometric plan.



Source: Personal collection.

Three DoF are required to describe the motion of the vehicle: x, y and ψ. The first two are the Cartesian coordinates of the reference point[2] B and ψ describes the vehicle's orientation.

Angle δ is the steering angle of the front wheel of the vehicle. The distance from the point A to the point B is the length of the wheelbase, denoted by l_{wb}, that is the only parameter required by the model, detailed in Table 1. The point O is the instantaneous center of rotation, that is defined by the intersection of line segment $\overline{\text{AO}}$ and $\overline{\text{BO}}$ which are drawn perpendicular to wheels' direction. $\overline{\text{AO}}$ is also the radius of the vehicle's path, represented by r.

The major assumption used in the development of the KST is that the velocity vectors, denoted by $\mathbf{v}$, at points A and B are in the direction of the orientation of the front and rear wheels, respectively. This hypothesis is referred to the fact that the model does not consider the tire slip and lateral forces. This is a reasonable assumption for the vehicle at low speeds – for example, for values smaller than 5 m/s – or for the steering angle values close to zero, since, within these two conditions, the lateral force generated by the tires is irrelevant (WASLANDER; KELLY, 2020). Because of this assumption the

[2]In this thesis it was placed in the center of rear wheel, but its location is free to be chosen. It can be placed also in the center of the front wheel or in the center of gravity of the vehicle. The selection of the reference point changes the equations.

rotation rate ω of the bicycle is equal to the velocity over the instantaneous radius r:

$$\dot{\boldsymbol{\psi}} = \boldsymbol{\omega} = \frac{\mathbf{v}}{r}. \tag{27}$$

Using trigonometric relations in triangles, it is possible to affirm that:

$$\tan(\delta) = \frac{\ell_{\text{wb}}}{r}. \tag{28}$$

Combining both Equations (27) and (28) it is possible to obtain the relation between ω and δ:

$$\dot{\boldsymbol{\psi}} = \boldsymbol{\omega} = \frac{\mathbf{v}}{r} = \left(\frac{\tan(\delta)}{l_{\text{wb}}}\right)\mathbf{v}. \tag{29}$$

Based on this model configuration, the velocity components of the reference point are:

$$\dot{x} = v\cos(\psi), \tag{30a}$$

$$\dot{y} = v\sin(\psi). \tag{30b}$$

The differential equations of the KST model are extracted from the equations above. In order to write them in a state-space representation, the states and inputs have to be specified. The state variables are introduced as $x_1 = x$, $x_2 = y$, $x_3 = \psi$, $x_4 = v$, and the inputs are $u_1 = \delta$, $u_2 = a_{\text{long}}$, where a_{long} is the longitudinal acceleration.

With the states, inputs and parameters defined, the equations in the state-space representation are:

$$\dot{x}_1 = x_4\cos(x_3), \tag{31a}$$

$$\dot{x}_2 = x_4\sin(x_3), \tag{31b}$$

$$\dot{x}_3 = \frac{x_4}{l_{\text{wb}}}\tan(u_1), \tag{31c}$$

$$\dot{x}_4 = u_2. \tag{31d}$$

Table 1 – *List of parameters used the KST model:* As the Equation (31) shows, the unique information that should be known beforehand about the vehicle representation is the length of the wheelbase.

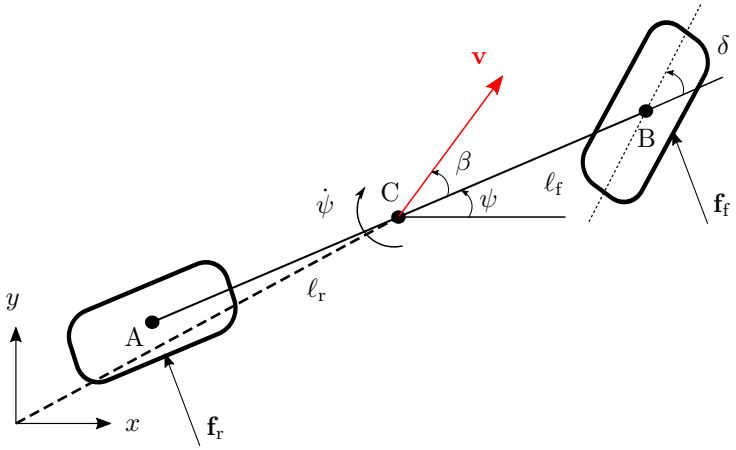
Parameters required		
Name	Symbol	Unit
Length of the wheelbase	l_{wb}	[m]

4.1.1.2 SINGLE-TRACK MODEL

As presented in the Section 4.1.1.1, the KST model does not consider any tire slip or lateral forces. At higher vehicle speeds, bigger steering angle values or small friction coefficients[3], the assumption that the velocity at each wheel is in the direction of the wheel can no longer be made. In this case, the effects caused by the lateral forces starts to be dominant. For that reason, a mathematical model that incorporates them is required.

The ST model is also a "bicycle" model and, in this case, the suspension movement, road inclination and aerodynamic influences are still neglected.

Figure 9 – *Single-Track model:* it represents the vehicle with two wheels in a geometric plan, taking into account the lateral forces.



Source: Personal collection.

According to Figure 9, the pair (x, y) represents the cartesian position, now referred to the center of gravity (c.g.) of the vehicle, represented by the point C. The distances of points A and B from the c.g. of the vehicle are ℓ_r and ℓ_r, respectively. The wheelbase of the vehicle is $\ell_\mathrm{wb} = \ell_\mathrm{f} + \ell_\mathrm{f}$.

The velocity at the c.g. of the vehicle is denoted by $\mathbf{v}$ and makes an angle β with the longitudinal axis of the vehicle. This angular difference between the velocity at the c.g. and the heading, ψ, of the vehicle is the slip angle, which can be defined as:

$$\beta = \tan^{-1}\left(\frac{\ell_\mathrm{r}\tan(\delta)}{\ell_\mathrm{wb}}\right). \tag{32}$$

and the forces $\mathbf{F}_\mathrm{f}$ and $\mathbf{F}_\mathrm{r}$ on the front and rear axis become:

$$\mathbf{F}_\mathrm{f} = \frac{mg\ell_\mathrm{r} - ma_\mathrm{long}h_\mathrm{cg}}{\ell_\mathrm{r} + \ell_\mathrm{f}}, \tag{33a}$$

$$\mathbf{F}_\mathrm{r} = \frac{mg\ell_\mathrm{f} - ma_\mathrm{long}h_\mathrm{cg}}{\ell_\mathrm{r} + \ell_\mathrm{f}}, \tag{33b}$$

[3]High friction coefficient is able to provide the lateral force required by the vehicle to handle the curved road. If the coefficient of friction is small or if the vehicle speed is high, then the vehicle would be unable to follow the nominal motion required by the driver and starts do drift.

where g is the acceleration of gravity which is approximately equal to 9.81m/s^2.

Inserting those forces into the derivation of the equations for the slip angle β and the yaw rate $\dot{\psi}$ and using the previously introduced variables and the parameters in Table 2, the result is given by the Equations (34a) – (34e):

$$\dot{x} = v\cos(\psi+\beta), \tag{34a}$$

$$\dot{y} = v\sin(\psi+\beta), \tag{34b}$$

$$\dot{v} = a_{\text{long}}, \tag{34c}$$

$$\ddot{\psi} = \frac{\mu m}{I_z(l_\text{r}+l_\text{f})}\Big\{l_\text{f}C_{\text{S,f}}(gl_\text{r} - a_{\text{long}}h_{\text{cg}})\delta + [l_\text{r}C_{\text{S,r}}(gl_\text{f} + a_{\text{long}}h_{\text{cg}}) - l_\text{f}C_{\text{S,f}}(gl_\text{r} - a_{\text{long}}h_{\text{cg}})]\beta - [l_\text{f}^2C_{\text{S,f}}(gl_\text{r} - a_{\text{long}}h_{\text{cg}}) + l_\text{r}^2C_{\text{S,r}}(gl_\text{f} + a_{\text{long}}h_{\text{cg}})]\frac{\dot{\psi}}{v}\Big\}, \tag{34d}$$

$$\dot{\beta} = \frac{\mu}{v(l_\text{r}+l_\text{f})}\Big\{C_{\text{S,f}}(gl_\text{r} - a_{\text{long}}h_{\text{cg}})\delta - [C_{\text{S,r}}(gl_\text{f} + a_{\text{long}}h_{\text{cg}}) + C_{\text{S,f}}(gl_\text{r} - a_{\text{long}}h_{\text{cg}})]\beta + [C_{\text{S,r}}(gl_\text{f} + a_{\text{long}}h_{\text{cg}})l_\text{r} - C_{\text{S,f}}(gl_\text{r} - a_{\text{long}}h_{\text{cg}})l_\text{f}]\frac{\dot{\psi}}{v}\Big\} - \dot{\psi}. \tag{34e}$$

In this model, more parameters of the vehicle are required to construct the equations, the Table 2 presents all the parameters needed.

Table 2 – *List of parameters used the ST model:* As the Equation (34) shows, the parameters listed in this table is the information that should be known beforehand about the vehicle representation, including some parameters related with the mass of the vehicle and others related with the tires and about the road.

Parameters required		
Name	Symbol	Unit
Vehicle mass	m	[kg]
Moment of inertia of the mass about z axis	I_z	[kg m^2]
Distance from the center of gravity to front axle	l_f	[m]
Distance from the center of gravity to rear axle	l_r	[m]
Height of center of gravity	h_{cg}	[m]
Front cornering stiffness coefficient	$C_{\text{S,f}}$	[rad^{-1}]
Front cornering stiffness coefficient	$C_{\text{S,r}}$	[rad^{-1}]
Friction coefficient	μ	[-]

To write Equations (34a) – (34e) in state-space representation, the original four first states of the state vector remain the same of the KST model. Then the states that the ST model requires are $x_1 = x$, $x_2 = y$, $x_3 = \psi$, $x_4 = v$, $x_5 = \dot{\psi}$, $x_6 = \beta$, and the inputs also are $u_1 = \delta$, $u_2 = a_{\text{long}}$.

With the states, inputs and parameters defined, the ST model is defined by the

following equations:

$$\dot{x_1} = x_4 \cos{(x_3 + x_6)}, \tag{35a}$$

$$\dot{x_2} = x_4 \sin(x_3 + x_6), \tag{35b}$$

$$\dot{x_3} = x_5, \tag{35c}$$

$$\dot{x_4} = u_2, \tag{35d}$$

$$\dot{x_5} = \frac{\mu m}{I_z(l_\mathrm{r} + l_\mathrm{f})}\Big\{l_\mathrm{f} C_{\mathrm{S,f}}(g l_\mathrm{r} - u_2 h_\mathrm{cg})u_1 + [l_\mathrm{r} C_{\mathrm{S,r}}(g l_\mathrm{f} + u_2 h_\mathrm{cg}) - l_\mathrm{f} C_{\mathrm{S,f}}(g l_\mathrm{r} - u_2 h_\mathrm{cg})]x_6 - [l_\mathrm{f}^2 C_{\mathrm{S,f}}(g l_\mathrm{r} - u_2 h_\mathrm{cg}) + l_\mathrm{r}^2 C_{\mathrm{S,r}}(g l_\mathrm{f} + u_2 h_\mathrm{cg})]\frac{x_5}{x_4}\Big\}, \tag{35e}$$

$$\dot{x_6} = \frac{\mu}{x_4(l_\mathrm{r} + l_\mathrm{f})}\Big\{C_{\mathrm{S,f}}(g l_\mathrm{r} - u_2 h_\mathrm{cg})u_1 - [C_{\mathrm{S,r}}(g l_\mathrm{f} + u_2 h_\mathrm{cg}) + C_{\mathrm{S,f}}(g l_\mathrm{r} - u_2 h_\mathrm{cg})]x_6 + [C_{\mathrm{S,r}}(g l_\mathrm{f} + u_2 h_\mathrm{cg}) l_\mathrm{r} - C_{\mathrm{S,f}}(g l_\mathrm{r} - u_2 h_\mathrm{cg}) l_\mathrm{f}]\frac{x_5}{x_4}\Big\} + x_5. \tag{35f}$$

The limitation to use the Equations (35a) – (35f) is due to the fact that for velocities equal to zero the system is not observable and velocities close to zero make the system singular or ill-conditioned, which means that the system loses stability. To address this problem the KST should be used for the time instants when this occurs instead of the ST.

For speeds smaller than 1 m/s, the state x_5 is computed as the derivative of x_3 from the KST model, $\dot{x_6}$ is zero and the states from x_1 to x_4 are obtained from the KST model with the addition of an $u_3 = \dot{\delta}$ to calculate x_5, as shown in Equations (36a) – (36f),

$$\dot{x_1} = x_4 \cos(x_3), \tag{36a}$$

$$\dot{x_2} = x_4 \sin(x_3), \tag{36b}$$

$$\dot{x_3} = \frac{x_4}{l_\mathrm{wb}} \tan(u_1), \tag{36c}$$

$$\dot{x_4} = u_2, \tag{36d}$$

$$\dot{x_5} = \frac{u_2}{l_\mathrm{wb}} \tan(u_1) + \frac{x_4}{l_\mathrm{wb} \cos^2(u_1)} u_3, \tag{36e}$$

$$\dot{x_6} = 0. \tag{36f}$$

Although these models are a convenient approach for this project, it is important to point out that their equations clearly represent a simplification from the true vehicle model. Some other dynamics and physical forces actuate on the vehicles behavior, e.g. suspension and tire forces and its properties.

4.1.2 SENSOR DATA

The vehicles in automation design are coupled with sensors that provide information about the vehicle's state variables and standard deviations. The ZF Tugger is equipped with a high accuracy Inertial Measurement Unit (IMU) sensor and one Global

Navigation Satellite System (GNSS) receiver to measure some states precisely, e.g. position and orientation (WASLANDER; KELLY, 2020).

The IMU is responsible for measuring the movement of the object in an inertial space. It is typically composed of gyroscopes measuring the angular rotation rates about the three axes, linear accelerometers that measure specific forces also along the three axes and magnetometers or a compass to improve the tracking orientation.

The GNSS is an important navigation sensor for self-driving cars. It is a constellation of satellites providing signals from space that transmit positioning and timing data to GNSS receivers. Some examples of regional system constellations are the GPS, GLONASS and GALILEO – American, Russian and European constellations of satellites, respectively. In the application case of this project, the GNSS accuracy is improved by using a Differential GPS (DGPS) that can correct the receiver positioning estimates by using more accurately known positions through some fixed base stations. Table 3 summarizes the main differences between the basic GPS and the DGPS.

Table 3 – *Main differences between the GPS and DGPS:* comparing what both sensor devices use to get the measurements, error correction and accuracy. A static base station can be used to provide correction messages to signal delays (caused by atmospheric effects) in the DGPS. This is done by setting the base station in a set point on the ground, and then working out its exact position on the earth's surface. It is important to note that the DGPS corrections improve the accuracy of position data only. DGPS has no effect on results that are based on speed data, for example.

GPS	DGPS
Mobile receiver	Mobile receiver + fixed base stations
No error correction	Estimate the error caused by atmospheric effects
$\sim$ 10 m accuracy	$\sim$ 2 cm accuracy

In addition to these sensors (IMU and GNSS), the Embedded Control Unit (ECU) of the vehicle is able to provide information about its velocity and the instantaneous radius curvature of the vehicle given by an encoder placed on the wheels of the vehicle. From this information the acceleration and the steering angle can be computed. Those conversions and numerical differentiation can introduce some noise. In this project the measurements used in the stochastic filters come directly from the GNSS, IMU and ECU (odometry). Table 4 shows which measurements can be acquired by each device, according to the states and inputs of the previous section.

Table 4 – *Position and orientation sensors devices:* The GNSS sensor is considered to measure directly some states of the vehicle, while the IMU sensor is considered to directly measures the acceleration. Lastly, the ECU of the vehicle calculates the wheel speeds and the steering angle of the wheels with a dynamic kinematic model.

Device	Measurement given	Unit
GNSS	x position	[m]
	y position	[m]
	Yaw angle	[rad]
	Velocity	[m/s]
Odometry	Steering angle of the left and right wheels	[rad]
	Wheel speeds	[m/s]
IMU	Acceleration	[m/s^2]

The information of these sensors are combined in a way to get the best possible estimate of the vehicle state even if the sensors have different frequencies (explained further, in Chapter 6) or some of them fails. For that purpose, it is necessary to calibrate the sensors and try to combine measurements related to the same time instant.

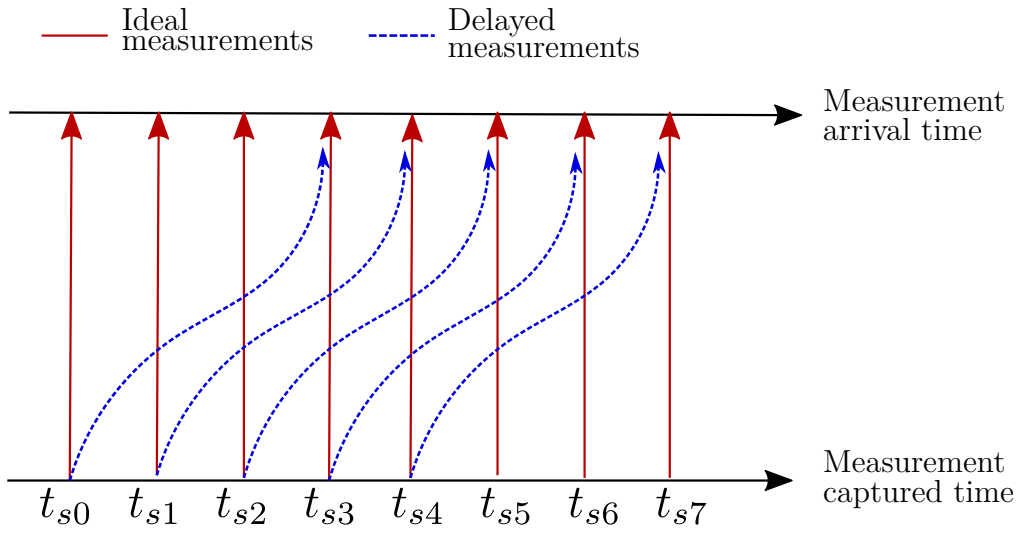
4.1.2.1 DELAYED MEASUREMENTS

Despite the intrinsic and extrinsic calibration to ensure that the measured data available comes from the same time instant, one type of sensor calibration is the temporal adjustment. A part of this involves determining the time offset between the time instant when the sensor records a measurement and the time instant when the stochastic filter receives it for the estimation processing.

For simplicity, it is often assumed that all the measurements are captured at the same time and represent exactly the current position of the vehicle. However, there is an unknown delay caused by the preprocessing steps that happen internally on the sensors. In this case, the measurements from the past are wrongly representing the current position of the vehicle, as shown in Figure 10.

As an example of how the delays can affect the measurement results, consider a car moving in a straight line at a speed of 10 m/s and equipped with a GNSS receiver that receives an updated position signal at every 10 ms. If these signals are 20 ms delayed it means that the position received does not correspond to the real time instant, but it is from the past. Using the simple kinematic relation for mean speed, $v = \Delta s/\Delta t$, it is possible to have an idea of how wrong the position value given by the GNSS is. In this case, the error is approximately 20 cm. If the velocity is low, for example 5 m/s, the position error drops to at least 5 cm but it still exists.

Figure 10 – *Delayed measurements system:* The red arrows represents an ideal scenario, where the sensors readings are not delayed. In this case, the measurement is captured and becomes available for the filter's update step in the same time instant. The dashed blue arrows is an example of a real scenario and in the case that the delay value is fixed for all time steps, where the measurement is captured in some time instant but it is only available for the filter three time steps later.



Source: Personal collection.

4.1.2.2 MEASUREMENT MODEL

Knowing the real measurements that the sensors provide and having some knowledge about the equations of the vehicle model, it is possible to determine the measurement model of the vehicle. In this study it is considered that all data is already preprocessed and measured directly, as shown above. This makes the mathematical representation of the measurements completely linear,

$$y_1 = x_1, \tag{37a}$$

$$y_2 = x_2, \tag{37b}$$

$$y_3 = x_3, \tag{37c}$$

$$y_4 = x_4. \tag{37d}$$

where $\mathbf{y}(\mathbf{x})$ is the observation function that describes the expected sensor model according to the state variables: $x_1 = x$, $x_2 = y$, $x_3 = \psi$, and $x_4 = v$.

4.2 STATE ESTIMATION IN PRACTICE

The previous models were presented in a continuous-time format and, in order to apply them on the stochastic filters, it is necessary to use some numerical procedures

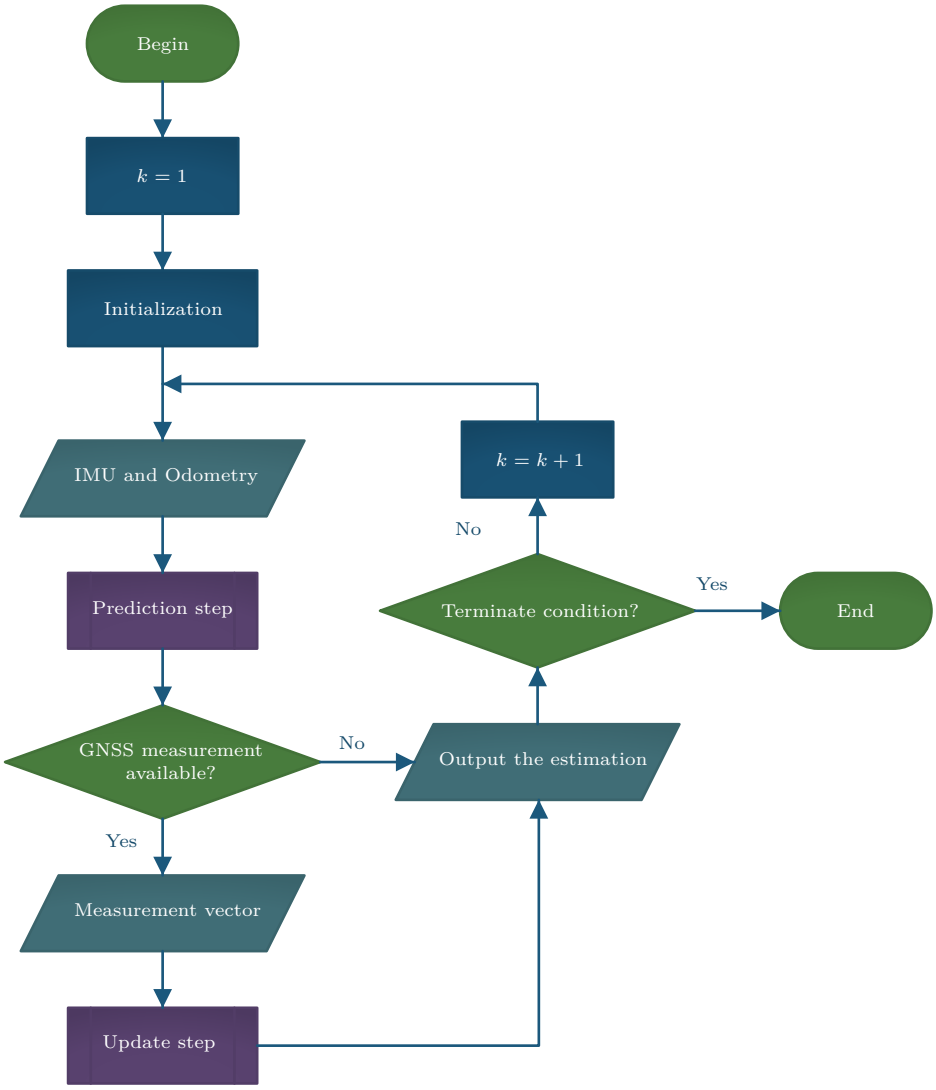
to solve the set of ODEs directly or discretize the model and use the equations in the discretized form, depending on the variant of the stochastic filter. Following, the Foward Euler method and the Runge-Kutta fourth order method are used. They are treated in more detail in Appendix C.

With the mathematical model that represents the vehicle and the expected observations and with the real measurements obtained through the coupled sensors, it is possible to show the usage of the EKF and UKF, and in their respective variant forms, in the following cases:

- Estimation of the complete state vector of the model without delays;
- Estimation of the complete state vector of the model incorporating the delayed measurements.

The odometry and the IMU data define the input vector of the vehicle model while the GNSS measures some states and refine the accuracy of the estimator in the update step of the stochastic filters. Since they are different sensors, they can have different frequencies. In the application related in this thesis, for example, the ECU of the vehicle runs at every 10 ms. It means that the measurements of the steering angle and wheel speed are updated in periods of 10 ms. Then, the IMU then should also updates at every 10 ms. In contrast, the GNSS can measures in different frequency than ECU and IMU or the signals do not reach the receiver device for some time, e.g. when a car crosses a tunnel. In this cases, the stochastic filters works to obtain the predictions every time that the inputs are available, while the update step is performed when the GNSS measurements are available. For example, if the GNSS measures every 1 s and the input data is given every 10 ms, the update occurs once at every 100 predictions. The flowchart represented by Figure 11, shows basically how the filter algorithm proceeds for this scenario.

Figure 11 – *Flowchart diagram of the KF group of algorithms:* showing the basics of the workflow process. It is a summary about how the algorithm works in practice. After it begins, the initialization is required, where the values of the initial state and covariance are specified. Then at each time step of the total time frame, the input data from the IMU and odometry are used to predict the state of the vehicle at the current time step. If the measurement from the GNSS is available, the filter uses its values to do the update step and then outputs this correction as the final result of the estimation. If the measurement from the GNSS is not available, the filter outputs the predicted state as the estimation result for the current time step.



Source: Personal collection.

In order to start the workflow using the prediction and update equations corresponding to each stochastic filter, four fundamental topics are established:

1. The states, inputs and the measurements of the sensors used are overwritten at each time step k with the updating of the estimate and reading of the sensors. The

state vector[4] is $\mathbf{x}_k^{(4)} = \begin{bmatrix} x & y & \psi & v \end{bmatrix}_k^\intercal$ or $\mathbf{x}_k^{(6)} = \begin{bmatrix} x & y & \psi & v & \dot{\psi} & \beta \end{bmatrix}_k^\intercal$ depending on the model used. The input vectors are $\mathbf{u}_k^{(4)} = \begin{bmatrix} \delta & a_{long} \end{bmatrix}_k^\intercal$ and $\mathbf{u}_k^{(6)} = \begin{bmatrix} \delta & a_{long} & \dot{\delta} \end{bmatrix}_k^\intercal$ and the sensor measurement vector, $\mathbf{z}_k$, is composed by: $\begin{bmatrix} x^{(z)} & y^{(z)} & \psi^{(z)} & v^{(z)} \end{bmatrix}_k^\intercal$.

2. There is not a systematic method to define the value of the $\mathbf{Q}$ and $\mathbf{R}$ matrices in this thesis. The matrix $\mathbf{R}$ is commonly specified by the sensor standard deviations. Usually these sensor's characteristics are known beforehand. However the choice of the $\mathbf{Q}$ matrix is critical since all the model uncertainties and the noises that affects the process should be incorporated into it. In this thesis the matrix $\mathbf{Q}$ is tuned as $\mathbf{Q} = 10^p\mathbf{R}$, where the value of p is different for the simulations, where $p = -1$, and for real application scenarios, where $p = 1$. The values of these matrices are selected for the total filtering duration, being time-invariant matrices, except for a special case when delays are incorporated, as detailed in Section 4.2.2.

3. The initial values of the states and covariance matrix are defined here as the first value of the GNSS measurement available for the states, $\mathbf{x}_0^{(4)} = \begin{bmatrix} x_0^{(z)} & y_0^{(z)} & \psi_0^{(z)} & v_0^{(z)} \end{bmatrix}^\intercal$ and $\mathbf{x}_0^{(6)} = \begin{bmatrix} x_0^{(z)} & y_0^{(z)} & \psi_0^{(z)} & v_0^{(z)} & 0 & 0 \end{bmatrix}^\intercal$ and as identity matrix for the covariances, $\mathbf{P}_0^{(4)} = \mathbf{I}_{4\times4}$ and $\mathbf{P}_0^{(6)} = \mathbf{I}_{6\times6}$. It is important to notice that $\mathbf{P}_0$ is only an initial value of the state estimation covariance $\mathbf{P}_k$. This covariance changes as the filter evaluates in time and should reach a steady value as the filter converges.

4. All the parameters of the state model must be given. In the simulated scenario all the parameters are known. In the case of the real scenario performed with the Tugger, in the KST model the only parameter used is known and for the ST model some of the parameters have the value approximated or the information is not available. To address this problem a parameter estimation is done, which is detailed later in the Section 4.2.1.

Thus, the filter starts the prediction and update equations according to each variant.

4.2.1 PARAMETER ESTIMATION

As already mentioned, mathematical models facilitate the knowledge about a real process allowing a better understanding about the system, also allowing to predict its behavior when some variable or parameter eventually changes.

Modelling real processes is traditionally based on the called white box modelling or black box model identification techniques. The white box models use physics relations

[4]Henceforth, the superscript '4' is related to the KST model while the superscript '6' is related to the ST model.

that can analytically describe the process, where all the parameters are known. Black box identification employs a model, which is adapted to measured data obtained from several possible inputs of a real process.

For many industrial processes, incomplete knowledge concerning the system is common. This implies that between the white box models and the black box models there is a "gray zone" which gives a third way of making models for engineering systems. In this thesis, *a priori* knowledge concerning the process is assumed, e.g. the known parameters and the inputs. The unknown parameters of the discrete-time nonlinear gray box model are estimated from the measured data using the System Identification toolbox in MATLAB®.

When a parameter estimation is required, an algorithm should be developed in a way that interprets it as an optimization problem where the solution is the set of parameters itself. This algorithm is composed by a cost function and the parameter constraints or bounds. This specifies some restrictions that the estimated parameters should respect. It is related to a previous prior knowledge about the target; e.g. the mass of the test vehicle must be inside the interval of $[1.5, 5.5] \times 10^3$ kg.

The algorithm estimates the parameter vector $\mathbf{p} \in \mathbb{R}^n$ and then calculates a simulated trajectory, $\mathcal{Y}_{\text{sim}}$. It should follow as closest as possible the reference signal, $\mathcal{Y}_{\text{ref}}$, that usually is the measured trajectory. In order to do that, the algorithm tries to minimize the estimation error, $e(t)$, that consists in a series of difference values between the simulated and reference trajectories, as follows

$$e(t) = \mathcal{Y}_{\text{ref}}(t) - \mathcal{Y}_{\text{sim}}(t). \tag{38}$$

The cost function, $\mathbf{J}(\mathbf{x})$ is then the Sum Squared Error (SSE) to process the error,

$$\mathbf{J}(x) = \sum_{t=0}^{N} e^2(t), \tag{39}$$

where N is the number of samples.

The bounds or constraints can be specified for the parameters based on the knowledge available about the system. These bounds are expressed as:

$$\underline{x} \leq x \leq \bar{x}, \tag{40}$$

where $\underline{x}$ and $\bar{x}$ are the minimum and maximum value that the parameter can assume, respectively.

The complete table with the set of the test vehicle's parameters is presented in Chapter 6.

4.2.2 INCORPORATION OF DELAYED MEASUREMENTS

As mentioned earlier, in this study, the incorporation of delayed measurements means that the measurement that arrives in the filter at each GNSS cycle corresponds to

an unknown number of steps back. The filter can ignore the fact that the measurements are delayed and use them to correct the current prediction — for noncompensated delays — or redo the previous estimates based on the number of delays defined previously in the algorithm – for compensated delays. In this study, both methods were evaluated for the variants of EKF and UKF.

Determining the exact number of delayed steps is not trivial. Therefore, a time range between 0 and 100 ms is considered for possible delays, which means a range between 0 and 10 time steps. Sensors with larger delays become useless for this project because it is a kind of application where the sensors should pre-process and make the updates of a time instant available as fast as possible to avoid safety problems.

Estimations resulted from filters that are not able to compensate the delays depends on the rate of the states measured by the sensor, as shown by the example given in Section 4.1.2.1.

Considering this, the $\mathbf{R}$ matrix is modeled to cover the region where the car can actually be, at each time step, according to the rate of variation over time of the i-th state variables, $\dot{x}_i$, and the considered delay. The model for each element, $\mathbf{r}^*_{i,k}$, of the main diagonal of the new $\mathbf{R}^*_k$ matrix can be considered as:

$$r^*_{i,k} = r_{i,0} + \dot{x}_i \Delta t, \tag{41}$$

$$R^*_k = \begin{bmatrix} r^*_{1,k} & 0 & 0 & 0 \\ 0 & r^*_{2,k} & 0 & 0 \\ 0 & 0 & r^*_{3,k} & 0 \\ 0 & 0 & 0 & r^*_{4,k} \end{bmatrix}, \tag{42}$$

where Δt is the assumed delay and $r_{i,0}$ is the initial value of the element. This means that as the $\mathbf{R}^*_k$ increases, the filter puts more weight on the model and the estimated states get closer to the predictions. When the delays are compensated, the filter returns to the corresponding state, where the update and the predictions of the subsequent states are recalculated. Different from standard filters, those that compensate the delays need to keep in memory the estimates and covariance matrices of the last d time steps.

5 COMPARISON TECHNIQUES AND TEST OF CONSISTENCY

5.1 CONSISTENCY OF THE FILTERS

In practice, it is necessary to check if the filter is performing correctly. Some tests can be applied to check the consistency of the filter – and if they are working as expected – from the desired characteristics of the measurement residuals. These include:

- The zero mean and the Gaussian behaviour of the residuals;
- The autocorrelation test;
- The Durbin-Watson (DW) test.

If m is the number of measurements, the measurement residuals, $\mathbf{q}_k \in \mathbb{R}^m$, consists in the subtraction between the measured data and the estimation value performed by the filter, as follows:

$$\mathbf{q}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^+. \tag{43}$$

The statistical tests used in this study are based on the following hypotheses concerning the residuals, considering i as the i-th element of the vector $\mathbf{q}_k$:

(a) The element $q_{i,k}$ is a random process with zero mean;
(b) The values of $q_{i,k}$ are uncorrelated;
(c) $q_{i,k}$ is a normally distributed random process.

If the residual term satisfies the previous hypoteses, then the model is considered valid. Since the statistical tests for significance are also based on these hypotheses, the conclusions resulting from these significance tests are called into question if the assumptions regarding the residuals fails.

The first test performed here is to verify if the residuals are normally distributed around the mean $\mu = 0$. In this case, it is possible to exploit the fact that the residuals are ergodic[1]. So then, the mean of the entire sample can be calculated as:

$$\bar{\mathbf{q}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^+ \approx 0, \tag{44}$$

where N is the data size. The respective covariance $\mathbf{P}_{r,k}$ can be obtained from the

[1] A stochastic process is ergodic when a sufficiently large data of random samples in a process can represent the average statistical properties of the entire process (Hamzah et al., 2013).

measurement noise covariance and the covariance of the prediction estimates as follows:

$$\mathbf{P}_{r,k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{R}_k(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^\intercal + \mathbf{H}_k(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}^-_{x_k}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^\intercal\mathbf{H}^\intercal_k \tag{45}$$

The second test performed is to check the autocorrelation of the residuals. The autocorrelation quantifies the degree of correlation between the values of the same variables across the measured data along the time. From the assumption that the residuals are independent, i.e. uncorrelated, the autocorrelation can indicate some problems. For example, if the filter is working properly and the autocorrelation exists, it indicates that the model may be incorrectly specified related to input data or parameters.

In order to quantify this, the Autocorrelation Function (ACF) is a function of the delay or lag τ, which determines the time shift taken into the past to estimate the similarity between data points and N total points. The ACF is defined as:

$$r(\tau) = \frac{1}{N}\sum_{k=0}^{N-\tau-1} q_k q_{k+\tau}. \tag{46}$$

Usually, the autocorrelation is normalized by the ACF when $\tau = 0$. In this test, at least 95% of the points should be inside the confidence limits, L, bounded by the values:

$$L = \pm\frac{2}{\sqrt{N}}. \tag{47}$$

The third test is about a numerical method commonly used for the autocorrelation analysis: the DW test. This test produces a statistic test that ranges from 0 to 4, in the following way:

$$d_w, i = \frac{\sum_{k=1}^{N}(q_{i,k} - q_{i,k-1})^2}{\sum_{k=0}^{N} q_{i,k}^2} \approx 2(1 - \rho). \tag{48}$$

where, $\rho \in [-1, 1]$ is the correlation coefficient. When the there is no correlation, $\rho = 0$. Table 5 summarizes the meanings of DW test results for interpretation purposes:

Table 5 – *Interpretation of the DW test results:* since the values of the correlation coefficient, ρ, are limited between -1 and 1, according to the Equation (48) the results of d_w, i are between 4 and 0. The value 2 means that the variables are not correlated at all. By the other side, values of 0 and 4 represents that the variables are perfectly correlated.

DW result	Correlation	Interpretation
2	$\rho = 0$	No autocorrelation
0	$\rho = 1$	Perfect positive autocorrelation
4	$\rho = -1$	Perfect negative autocorrelation

The presence of autocorrelation indicates that the stochastic filter is not working correctly or the model may be incorrectly specified, related to input data or parameters;

this is reflected in the DW test, as its results diverge from 2. Values for DW close to 2 indicates that the filter is working correctly and/or the model fits to the problem that is being addressed.

Assuming that the filter is working correctly, a way to optimize the results of the DW test when the model cannot be changed is augmenting the values of the process covariance matrix $\mathbf{Q}_k$ regarding to the measurement covariance matrix $\mathbf{R}_k$. This analysis is shown in the Chapter 6.

More detailed information about residual characteristics, the DW test and the ACF and its demonstrations can be found in (KRÄMER, 2011), (Hanlon; Maybeck, 2000), (Hamzah et al., 2013).

5.2 COMPARISON METHODS

In the computational simulations the performance of the filters are compared to each other with the Mean Squared Errors (MSEs) since the simulation is used as a ground-truth. In the real application, evaluating the performance of the filters is not trivial, because it is not possible to obtain this reference from real world systems, since the real behavior of the vehicle considers all the physical properties, e.g. suspension, aerodynamics and characteristics of the road. Also the true states are unknown. Therefore, the GNSS measurements are assumed as ground-truth for the calculation of the MSE.

The other form to compare the filters is analysing the computational costs in terms of the "big $\mathcal{O}$" notation. This is an important measure because high computational costs affects the filter's performance in real time applications. The Sections 5.2.1 and 5.2.2 deal with these methods in more detail.

5.2.1 MEAN SQUARED ERROR

According to (SAMMUT; WEBB, 2010), the MSE is the average squared difference between the estimated values, X, and the true/actual value, $\hat{X}$. The result measures the quality of the estimator, values close to zero are better. The MSE is computed as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)^2. \tag{49}$$

5.2.2 COMPUTATIONAL COSTS ANALYSIS

In most of the stochastic filters, the equations are formulated based on basic operations with matrices and vectors from the state-space representation. In these operations, the complexity of the algorithm also depends on the order of the respective system.

In this thesis the computational complexity indicator is defined by "the big $\mathcal{O}$" notation that indicates how the run time or space requirements of the filters grow as the matrices and vector size grows. By a definition, if $T(f(n))$ is the number of operations to

be executed to solve the problem of order n, $\mathcal{O}(f_O(n))$ can be defined as follows:

$$\mathcal{O}(f_O(n)) = \lim_{n\to\infty} T(f(n)) \tag{50}$$

According to this, the complexity for the EKF and UKF is shown in Table 6, extracted from (VALADE et al., 2017) and adapted for the spherical-simplex strategy to obtain the sigma-points. Here, n is the size of the state vector, m is defined as the measurement vector size and p as the input vector size.

Table 6 – *Computational cost of the filters equations in terms of big $\mathcal{O}$ notation:* the comparison is done for each equation of the EKF and UKF algorithms. $f(.)$ and $h(.)$ are the state model and the observation model, respectively, since their computational times are not being analysed.

Algorithm	Operation	$\mathcal{O}$
EKF	$\hat{\mathbf{x}}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_k, \mathbf{w}_k)$	$2n^2$
	$\mathbf{P}_{\mathbf{x}_k}^- = \mathbf{F}_{\hat{\mathbf{x}}_{k-1}^+} \mathbf{P}_{\mathbf{x}_{k-1}}^+ \mathbf{F}_{\hat{\mathbf{x}}_{k-1}^+}^\intercal + \mathbf{Q}_k$	$4n^3$
	$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-, \mathbf{v}_k)$	$2mn + m$
	$\mathbf{K}_k = \mathbf{P}_{\mathbf{x}_k}^- \mathbf{H}_{\hat{\mathbf{x}}_k^-}^\intercal (\mathbf{R}_k + \mathbf{H}_{\hat{\mathbf{x}}_k^-} \mathbf{P}_{\mathbf{x}_k}^- \mathbf{H}_{\hat{\mathbf{x}}_k^-}^\intercal)^{-1}$	$4n^2m/4m^2n$
	$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \tilde{\mathbf{y}}_k$	$2mn$
	$\mathbf{P}_{\mathbf{x}_k}^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_{\hat{\mathbf{x}}_k^-}) \mathbf{P}_{\mathbf{x}_k}^-$	$2n^3/2m^2n$
UKF	$\mathcal{X}_i = \bar{\mathbf{x}} + \mathcal{Z}_i \sqrt{\mathbf{P}_{\mathbf{x}}}$	n^3
	$\mathcal{X}_k = \mathbf{f}(\mathcal{X}_{k-1}, \mathbf{u}_k, 0)$	$2n\mathcal{O}(f(.))$
	$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2n} w_i^{(m)} \mathcal{X}_{i,k}$	$4n^2$
	$\mathbf{P}_{\mathbf{x}_k}^- = \sum_{i=0}^{2n} w_i^{(c)} (\mathcal{X}_{i,k} - \hat{\mathbf{x}}_k^-)(\mathcal{X}_{i,k} - \hat{\mathbf{x}}_k^-)^\intercal + \mathbf{Q}_k$	$6n^3$
	$\mathcal{Y}_k = \mathbf{h}(\mathcal{X}_k, 0)$	$(n+2)\mathcal{O}(h(.))$
	$\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2n} w_i^{(m)} \mathcal{Y}_{i,k}$	$4m^2n$
	$\mathbf{P}_{\tilde{\mathbf{y}}_\mathbf{k}} = \sum_{i=0}^{2n} w_i^{(c)} (\mathcal{Y}_{i,k} - \hat{\mathbf{y}}_k^-)(\mathcal{Y}_{i,k} - \hat{\mathbf{y}}_k^-)^\intercal + \mathbf{R}_k$	$6m^2n$
	$\mathbf{P}_{\mathbf{y}_k\mathbf{y}_k} = \sum_{i=0}^{2n} w_i^{(c)} (\mathcal{X}_{i,k} - \hat{\mathbf{x}}_k^-)(\mathcal{Y}_{i,k} - \hat{\mathbf{y}}_k^-)^\intercal$	$4n^2m$
	$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{y}}_k^-)$	$2mn$
	$\mathbf{P}_{\mathbf{x}_k}^+ = \mathbf{P}_{\mathbf{x}_k}^- - \mathbf{K}_k \mathbf{P}_{\tilde{\mathbf{y}}_\mathbf{k}} \mathbf{K}_k^\intercal$	$2n^3/2m^2n$

As a result, the KFs computing complexities are summed-up and in terms of "big $\mathcal{O}$" notation the complexity of the EKF is $\mathcal{O}(4n^3)$ and the UKF is about $\mathcal{O}(7n^3)$. With this analysis, it can be inferred that the UKF algorithm demands almost twice the computing time of an equivalent EKF algorithm, which can be decisive for applications with embedded microcontrollers.

The computational complexity of the filters in their square root formats are presented in Appendices A and B, related to each filter.

6 RESULTS AND DISCUSSION

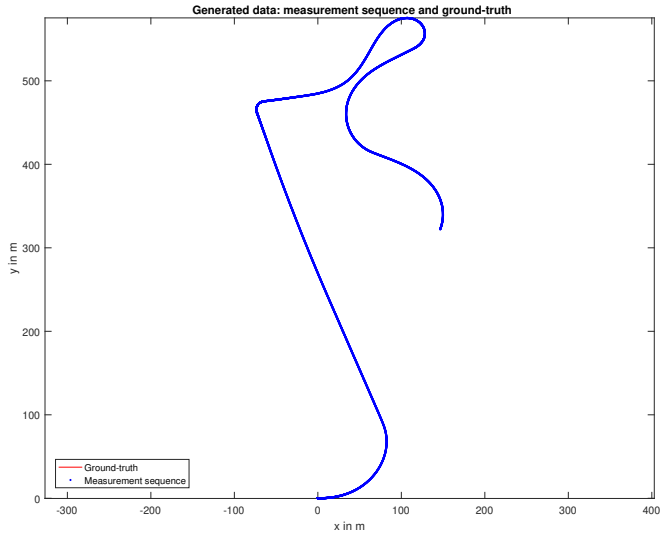
This chapter is divided into two main sections: the first section shows a simulated scenario where all the inputs and parameters are known and the model is defined as the true representation of the real model of the vehicle. The second section is about the pre-filtering and filtering results with real measurements extracted from the test vehicle. In both sections, the filters were used in their square root forms (see Appendices A and B) and the spherical-simplex form to take the sigma-points for the UKF. The discussion of the results is done simultaneously as the results are presented.

6.1 SIMULATED DATA RESULTS

In order to ensure that the filters are working properly, some computational simulations were done for considering ideal scenario, where the parameters and the inputs are known and the ST model fits all the dynamics of the vehicle perfectly.

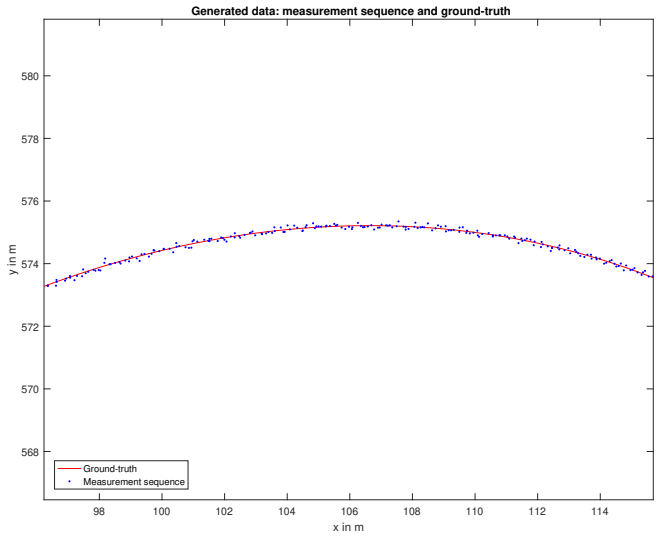
For this, a trajectory was generated as a ground-truth and some points normally distributed around this trajectory to simulate the measurement data from four — x, y, ψ, v — to six — x, y, ψ, v, $\dot{\psi}$, β — states with the simulated sensor noises, as shown in Figures 12 and 13.

Figure 12 – *Plot of the simulated trajectory:* using the ST model and (0,0) as the initial state position coordinates, the data is generated to be the ground-truth, used as a reference trajectory for the simulations. The measurement sequence was generated normally distributed around the known mean with a known measurement noise matrix $\mathbf{R}$.



Source: Personal collection.

Figure 13 – *Measurement data normally distributed around the ground-truth:* here, there is a region of the trajectory of Figure 12 to show the generated reference and the measurements distribution more closely.



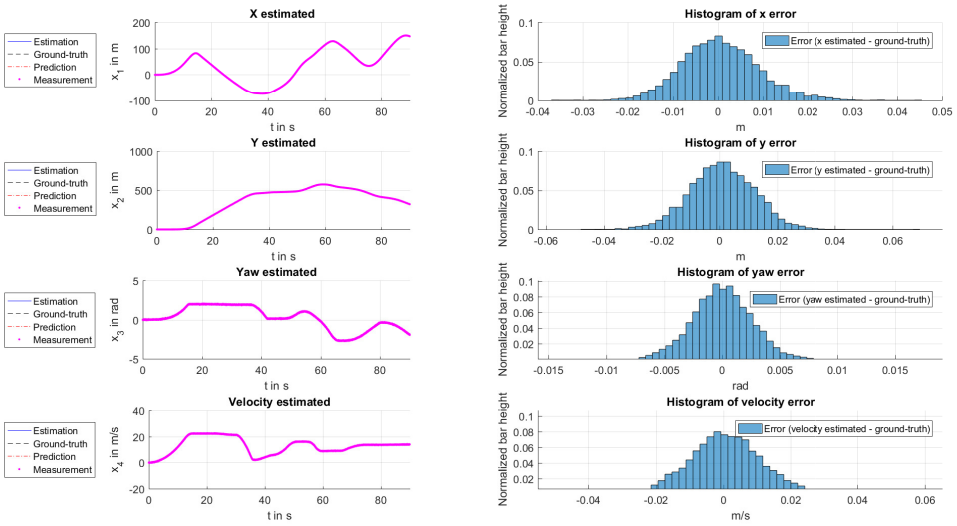
Source: Personal collection.

6.1.1 STOCHASTIC FILTERING SIMULATIONS

Since the model is known, the model covariance matrix $\mathbf{Q}$ is assumed with small values or at least smaller than the measurement covariance matrix $\mathbf{R}$ for the EKF and UKF filters, with $\mathbf{Q} = 10^{-1}\mathbf{R}$. The Figures 14 – 20 show the simulation results using the EKF and consistency tests to see how the filter performs when the model is corrected and if it is working properly.

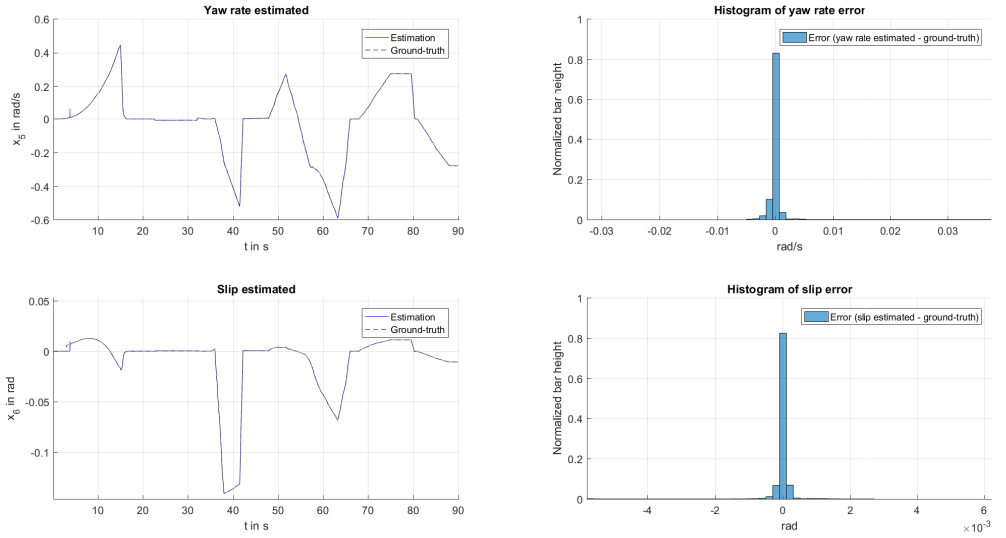
It is important to point out that in some figures the curves are very similar and therefore overlap each other. To represent the difference between these curves, the error histogram is shown on their right side.

Figure 14 – *Estimation of the measured states using EKF and simulated data:* on the left side, the four estimated states that are corrected in the update step with the measurement data generated (x, y, ψ and v). On the right side, the histograms represent the difference between these estimated values and the ground-truth. The y-axis of these histograms are bars with normalized heights, in order to represent the information as a probability density.



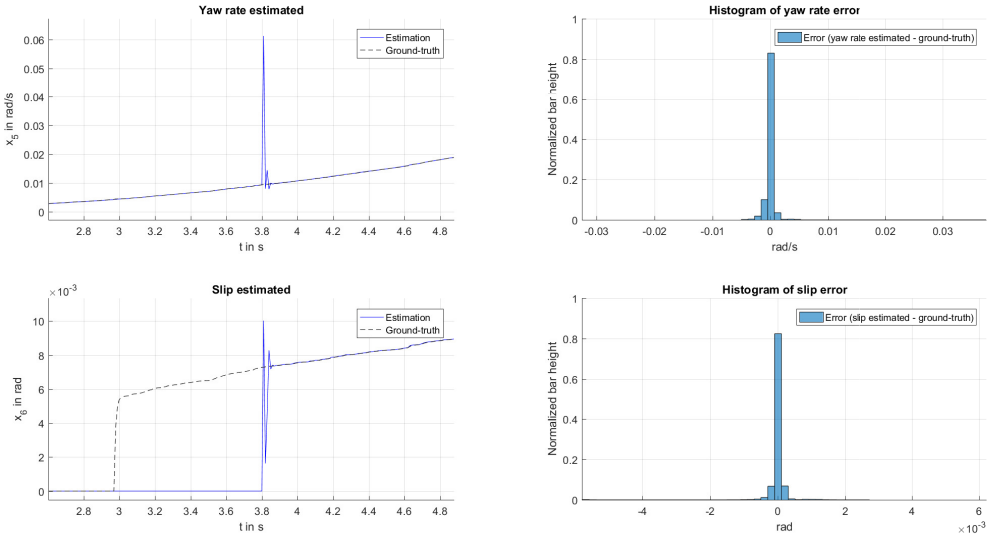
Source: Personal collection.

Figure 15 – *Estimation of the states that are not measured using EKF and simulated data:* on the left side, the estimated states that are not measured by the generated measurement data and on the right side its histograms of estimation error.



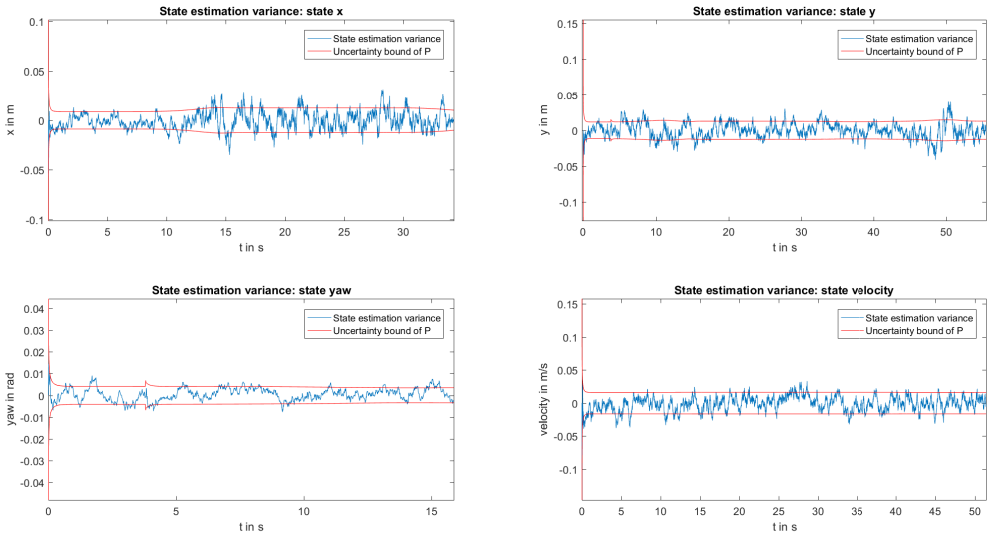
Source: Personal collection.

Figure 16 – *Disturbance in the estimation of the $\dot{\psi}$ and β using the EKF:* Here, it shows a region of Figure 15 more closely. This disturbance is caused when the model switches from the KST model to the ST model.



Source: Personal collection.

Figure 17 – *Error over time bounded by the covariance uncertainties of the states corrected by the generated measurements using the EKF:* The error is calculating by a simple difference between the ground-truth value and the estimation. This figure shows also the covariance matrix converging into a stationary value after the first estimation.

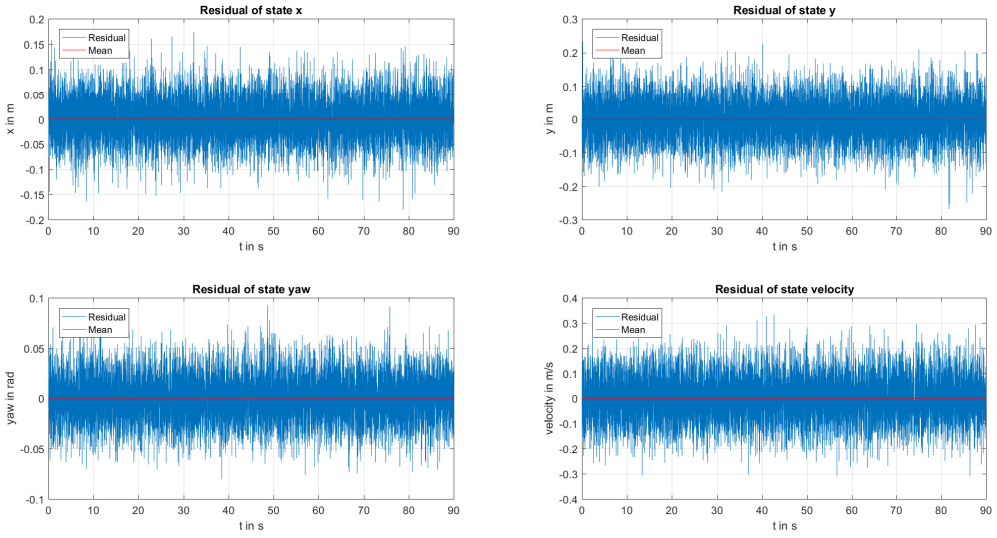


Source: Personal collection.

Next, consistency tests are performed for the EKF. The residuals and its histograms are presented in Figure 18 and 19 and the plots of the autocorrelation test are presented

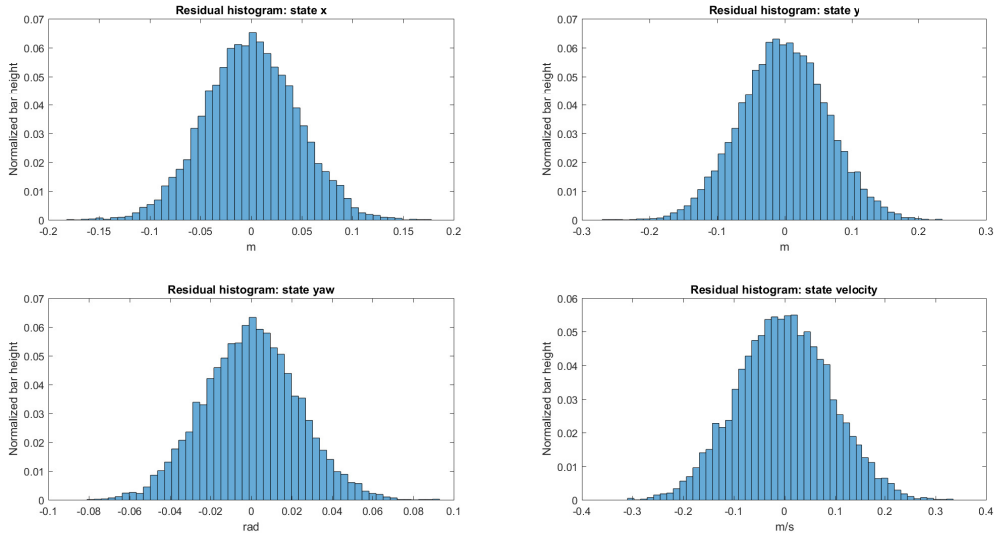
in Figure 20.

Figure 18 – *Residuals of the EKF estimations:* they are calculated as the difference between the estimated values and the generated measurement vector for every time step. The mean of the whole sample is near zero.



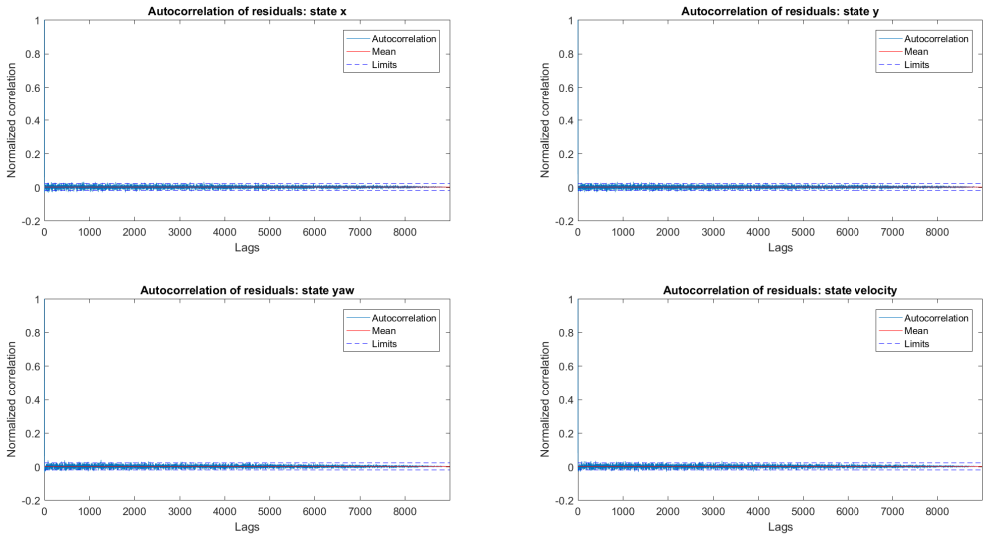
Source: Personal collection.

Figure 19 – *Histogram of the residuals of the EKF estimations using simulated data*: Here it is possible to see the Gaussian behaviour of the residuals.



Source: Personal collection.

Figure 20 – *Autocorrelation of the residuals of the EKF estimations with simulated data:* the normalized autocorrelations are plotted with their limits. More than 95% of the points are inside the limits.



Source: Personal collection

The values of the DW test and the mean of the autocorrelation are shown in Table 7.

Table 7 – *Mean of the autocorrelation and DW results of the EKF estimations for the simulated trajectory.*

State	DW result	Means of the autocorrelation
x	2.0339	7.0443×10^{-7} m
y	2.0055	2.3665×10^{-5} m
ψ	1.9975	8.7801×10^{-6} rad
v	2.035	4.0325×10^{-7} m/s

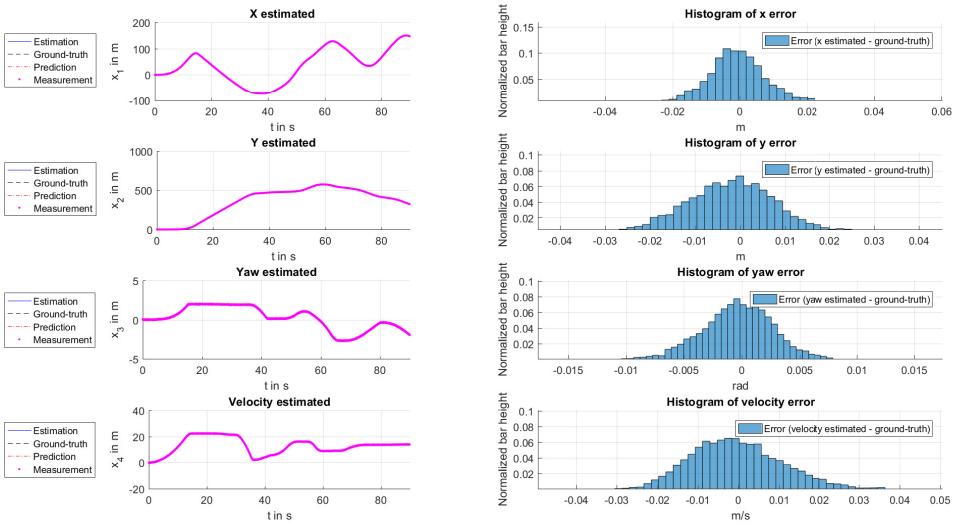
Next, the MSE errors are calculated using the generated ground-truth sequence. The values are presented in Table 8.

Table 8 – *MSE of the estimated states using the EKF and simulated data:* in this case, the ground-truth simulated is used as a reference to calculate the MSE. These values are used as a reference for the henceforth experiments with the stochastic filters.

State	MSE	Unit
x	1.5943×10^{-4}	m^2
y	1.6062×10^{-4}	m^2
ψ	1.7780×10^{-5}	rad^2
v	1.4983×10^{-4}	$(\mathrm{m/s})^2$
$\dot{\psi}$	1.6881×10^{-6}	$(\mathrm{rad/s})^2$
β	8.2819×10^{-8}	rad^2

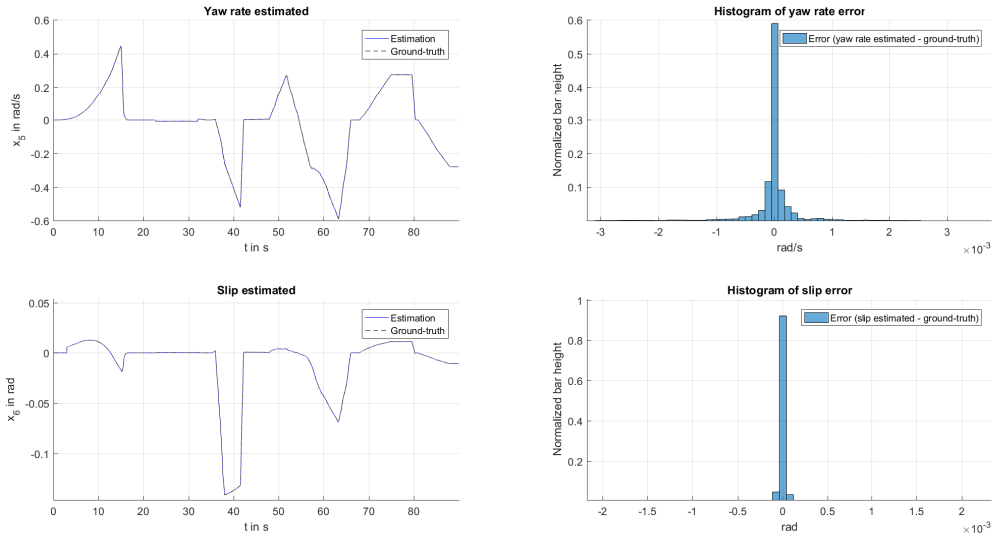
Next, the same procedure is done for the UKF and its results are presented below:

Figure 21 – *Estimation of the measured states using UKF and simulated data:* on the left side, the four estimated states that are corrected in the update step with the measurement data generated (x, y, ψ and v). On the right side, the histograms represents the difference between these estimated values and the ground-truth. The y-axis of these histograms are bars with normalized heights, in order to represent the information as a probability density.



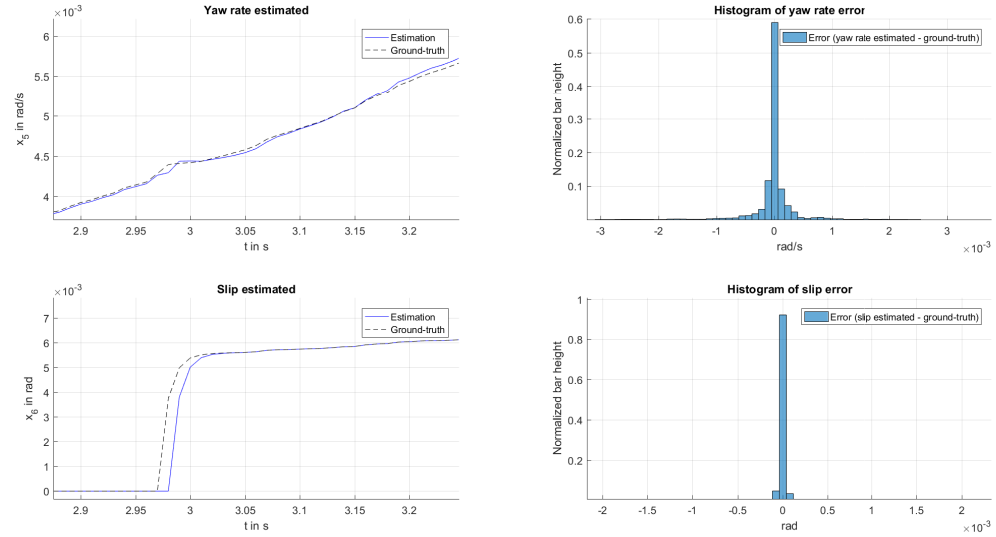
Source: Personal collection.

Figure 22 – *Estimation of the states that are not measured using UKF and simulated data:* on the left side, the estimated states that are not measured by the generated measurement data and on the right side its histograms of estimation error.



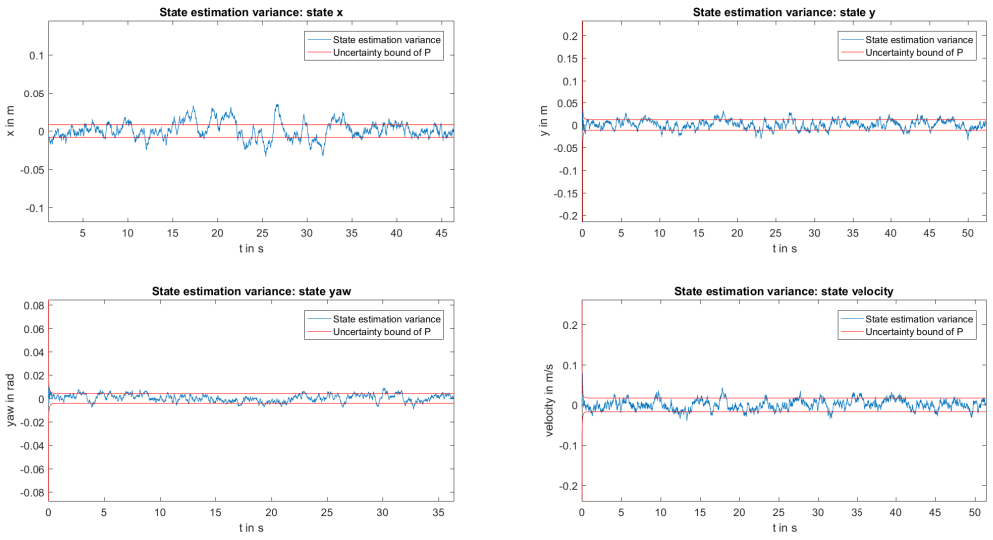
Source: Personal collection.

Figure 23 – *Lack of disturbance in the estimation of the $\dot{\psi}$ and β using the UKF:* Here, it shows a region of Figure 22 more closely. There is no disturbance caused when the model switches from KST model to the ST model.



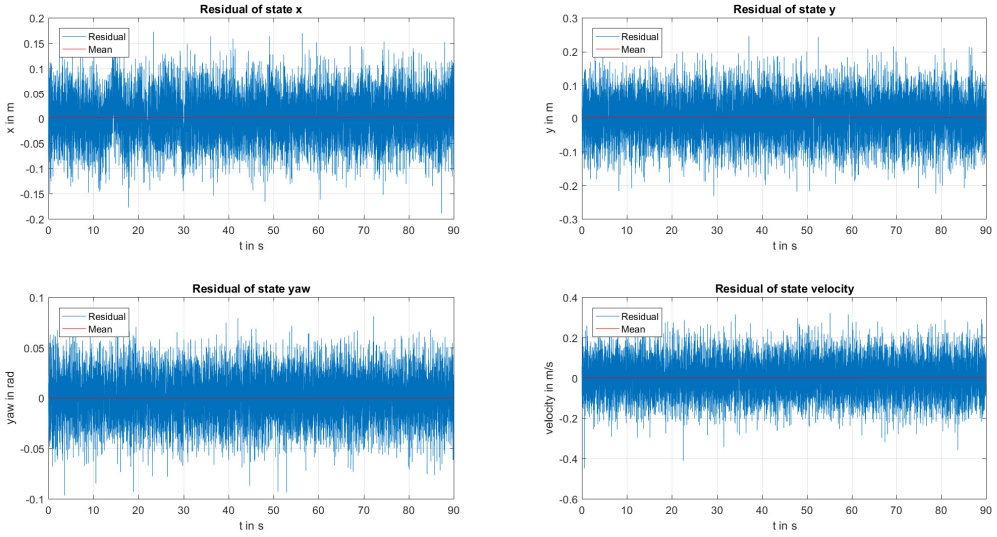
Source: Personal collection.

Figure 24 – *Error over time bounded by the covariance uncertainties of the states corrected by the generated measurements using the UKF:* The error is calculated based on a simple difference between the ground-truth value and the estimation. This figure shows also the covariance matrix converging to a stationary value right after the first estimation.



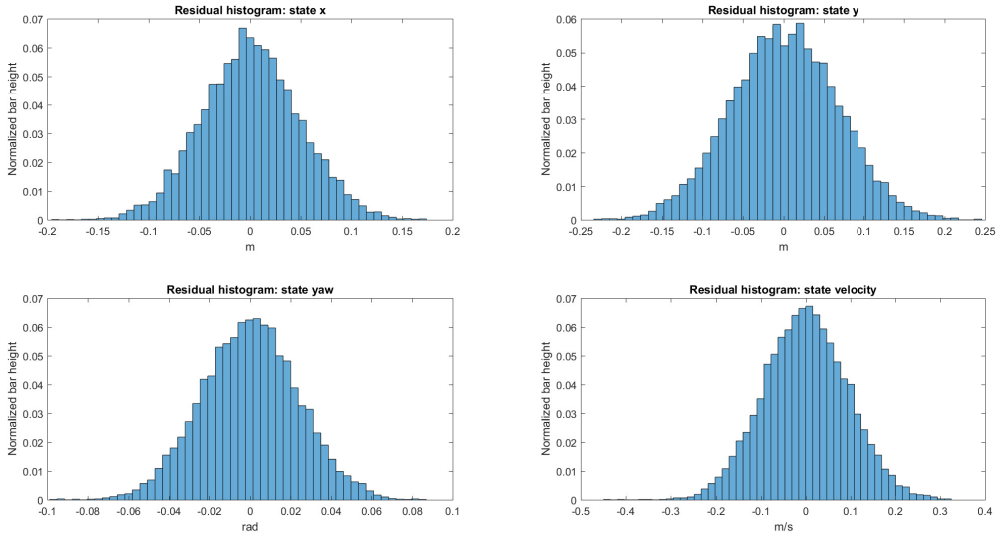
Source: Personal collection.

Figure 25 – *Residuals of the UKF estimations:* They are calculated as the difference between the estimated values and the generated measurement vector for every time step. The mean of the whole sample is near zero.



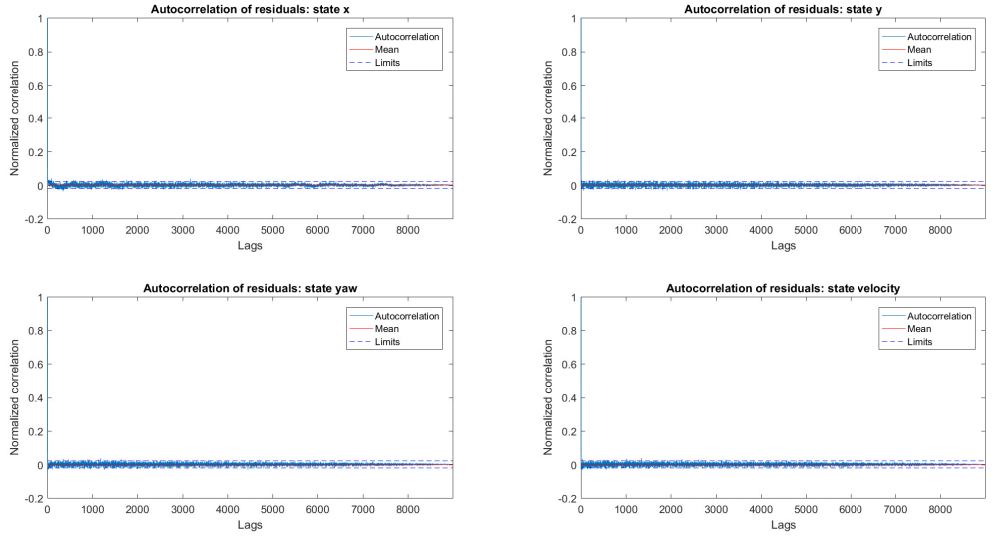
Source: Personal collection.

Figure 26 – *Histogram of the residuals of the UKF estimations with simulated data.*:Here, it is possible to see the Gaussian behaviour of the residuals.



Source: Personal collection.

Figure 27 – *Autocorrelation of the residuals of the UKF estimations with simulated data:* the normalized autocorrelations are plotted with their limits. More than 95% of the points are inside the limits.



Source: Personal collection.

The table values of the DW test and the mean of the autocorrelation are shown in Table 9.

Table 9 – *Mean of the autocorrelation and DW results of the UKF estimations for the simulated trajectory.*

State	DW result	Means of the autocorrelation
x	1.9676	3.9514×10^{-5} m
y	1.9965	3.9636×10^{-5} m
ψ	1.9984	4.2130×10^{-7} rad
v	2.036	1.2656×10^{-7} m/s

Finally, the MSE errors are calculated using the ground-truth sequence. The values are presented in Table 10.

Table 10 – *MSE of the estimated states using the UKF:* in this case, the ground-truth is simulated and used as a reference to calculate the MSE. The last column is the percentual comparison of the MSE values of the UKF against the MSE values of the EKF, presented in Table 8, and it is calculated as $\text{MSE}_{\text{UKF}}/\text{MSE}_{\text{EKF}} \times 100\%$.

State	MSE	Unit	$\frac{\text{MSE}_{\text{UKF}}}{\text{MSE}_{\text{EKF}}} \times 100\%$
x	1.6919×10^{-4}	m^2	106.12%
y	1.2006×10^{-4}	m^2	74.74%
ψ	9.3797×10^{-6}	rad^2	52.51%
v	1.5858×10^{-4}	$(\text{m/s})^2$	105.84%
$\dot{\psi}$	8.1492×10^{-8}	$(\text{rad/s})^2$	4.82%
β	$2,4591 \times 10^{-9}$	rad^2	2.97%

The graphic results exposed in Figures 14-27 show the adequate and correct functionality of both filters. Besides the difference between the measured states and the ground-truth values being Gaussian as expected, the consistency tests pointed out that the filters are working properly: the average of the residuals are visibly close to zero for all measured states, as shown in the Figures 18 and 25, and their histograms are normally distributed, as shown in Figures 19 and 26. Another important point is that the autocorrelation test was showing that the correlation points are between the established limits as presented in Figures 20 and 27. The DW test in Tables 7 and 9 confirms this as both the EKF and UKF values for this test are very close to 2, demonstrating no correlation. This indicates that the model used in the filters are correctly specified, which is true, since the ground-truth was generated using the same model used in the filter.

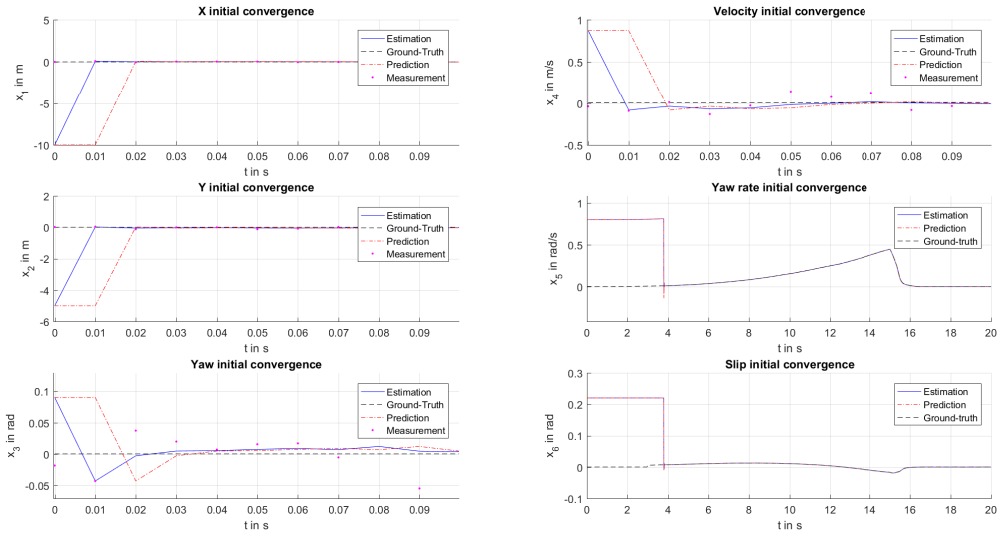
The first difference between EKF and UKF can be pointed out. Figure 16 shows a disturbance caused by switching from the KST model – where the slip angle is not considered – to the ST model – where the slip angle is considered – while the velocity is increasing. The EKF itself can not handle the abrupt change in the slip angle and some overshooting in the estimation is caused. On the other side, Figure 23 shows that the UKF can handle the model switching better, specially because the sigma-points captured by the

UT cover a region above and below the mean point. These sigma-points captured around the mean point can predict the sudden change of the slip angle and avoid undesirable overshoot in the estimations.

The second difference can also be pointed out when comparing the MSE results in Tables 8 and 10: the UKF shows a small improvement over EKF, especially for the states that are not measured by the generated measurement data, $\dot{\psi}$ and β.

The next figures show the performances of both of the filters when wrong initialization values of the state vector are set.

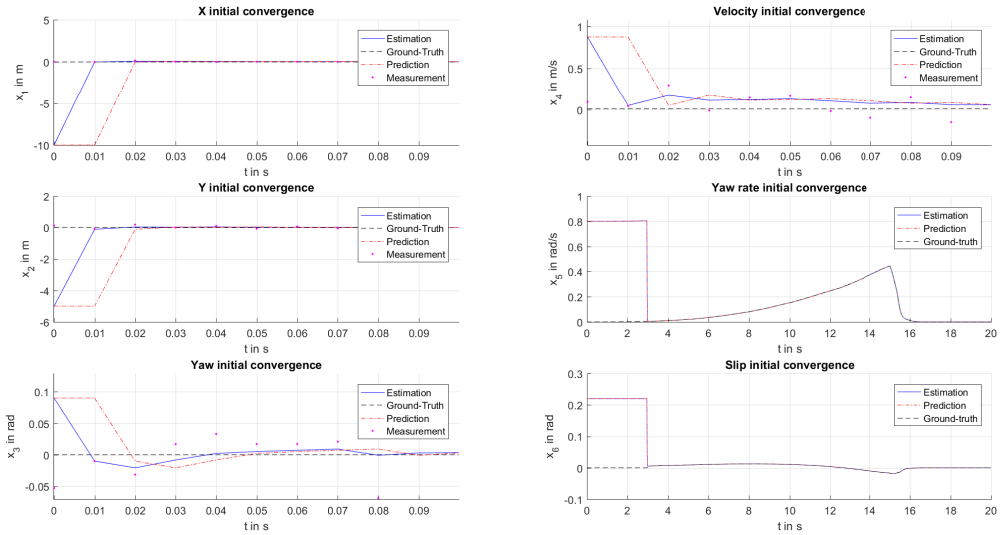
Figure 28 – *Convergence of the EKF estimations with wrong initial values:* this figure shows how the EKF converges when wrong initial values are set. For small velocities the KST model is used, where the $\dot{\psi}$ and the β are not state variables. For that reason, they are kept constant until the system switches to the ST model.



Source: Personal collection.

For the same (wrong) initialization vector, the UKF is simulated and the estimation results are given in Figure 29:

Figure 29 – *Convergence of the UKF estimations with wrong initial values:* like in Figure 28, this figure shows how the UKF converges when wrong initial values are set. The states $\dot{\psi}$ and β are also kept constant until the system switches to the ST model.

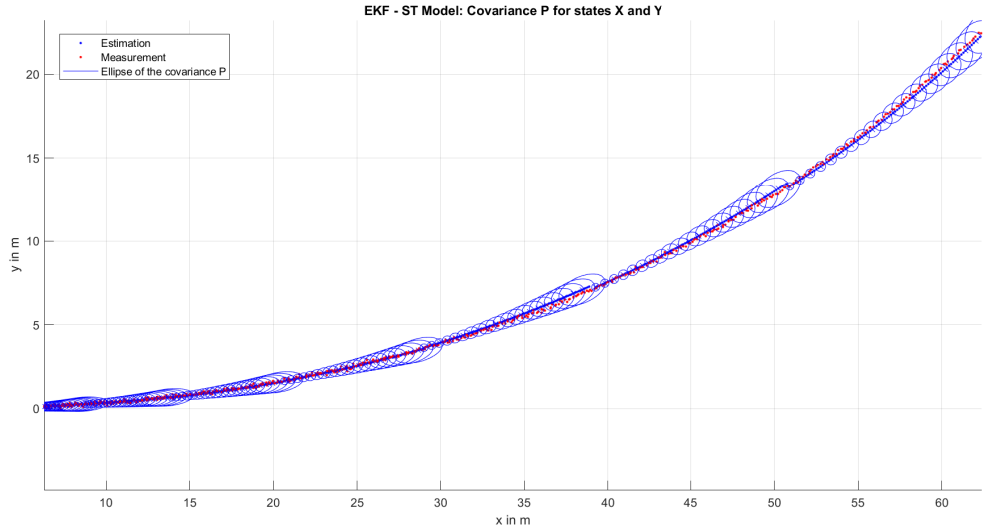


Source: Personal collection

Considering wrong initial values, the results perform as expected. The convergence occurs around the 1st and 2nd time-step, except for the states β and $\dot{\psi}$. For small velocities the model used for the filters is the KST where the $\dot{\psi}$ and the β are not variables of the state vector. For that reason, they are kept constant until the system switches to the ST model. Considering this scenario, another difference about the EKF and UKF can be pointed out: the UKF converges faster for those states that are not measured. This occurs because the UKF can switch from the KST model to the ST model for smaller velocities without oscilations or overshoots when compared to the EKF performance.

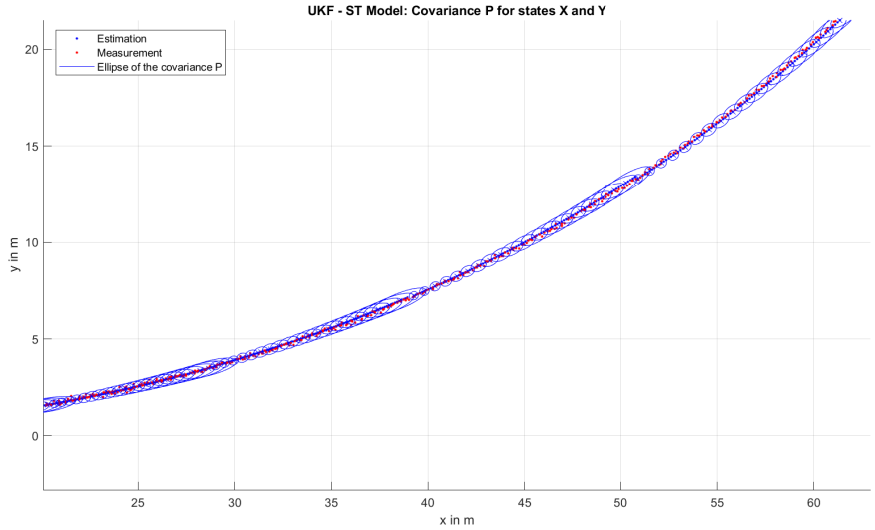
The following figures show the behavior of the ellipses in different regions of the vehicle's trajectory when the frequency of obtaining measurements is decreased. It means that as time passes and the measurements do not arrive, the reliability in the estimation result should decrease as only successive predictions steps occur. It is represented by the progressive increase of the covariance ellipse. However, when the measurement is finally obtained, prediction and update steps are performed, so the reliability in the estimation should increase again. It is represented by the covariance ellipse decreasing. The Figures 30 — 33 show it:

Figure 30 – *Region of the simulated trajectory showing the ellipses of the covariance* **P***, estimated using the EKF:* The ellipses are printed at every 0.05 seconds. In the period of time that the filter does only predictions the covariance grows. When the measurement arrives, the covariance matrix reduces its size again, represented by the decrease in the size of the ellipse.



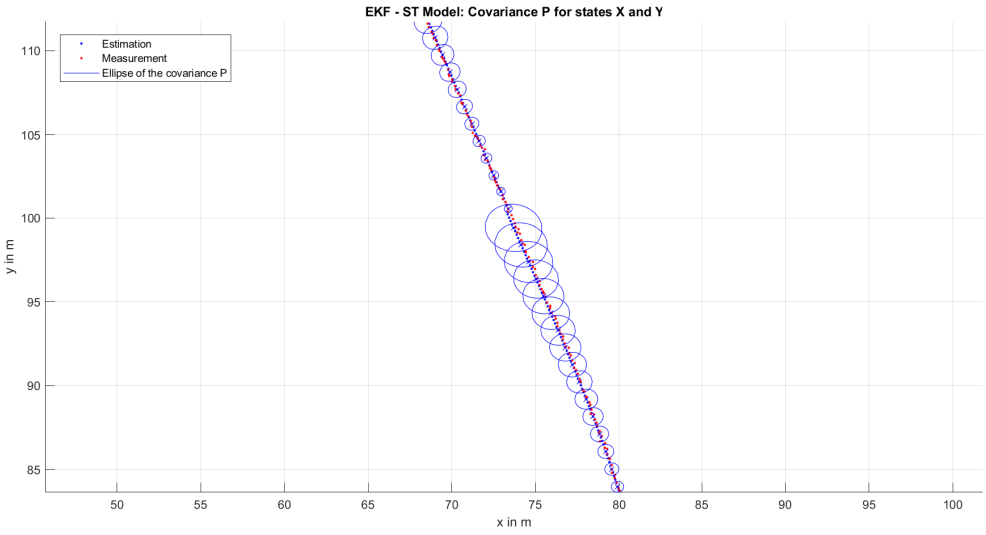
Source: Personal collection.

Figure 31 – *Region of the simulated trajectory showing the ellipses of the covariance* **P***, estimated using the UKF:* The ellipses are printed at every 0.05 seconds. In the period of time that the filter does only predictions the covariance grows. When the measurement arrives, the covariance matrix reduces its size again, represented by the decrease in the size of the ellipse.



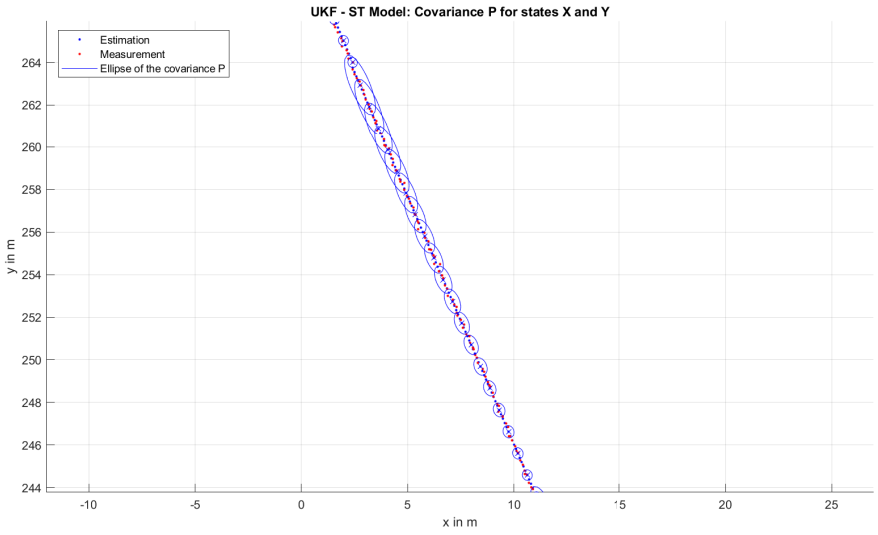
Source: Personal collection.

Figure 32 – *Another region of the simulated trajectory showing the mismatch between the vehicle's direction and the major axis of the ellipse of the covariance* **P**, *estimated using EKF:* The ellipses are printed at every 0.05 seconds. In this region, the major axis of the ellipse is not matching the vehicle's direction.



Source: Personal collection.

Figure 33 – *Another region of the simulated trajectory showing the alignment between the vehicle's direction and the major axis of the ellipse of the covariance* **P**, *estimated using UKF:* The ellipses are printed at every 0.05 seconds. In this region, the major axis of the ellipse still matches the vehicle's direction.



Source: Personal collection.

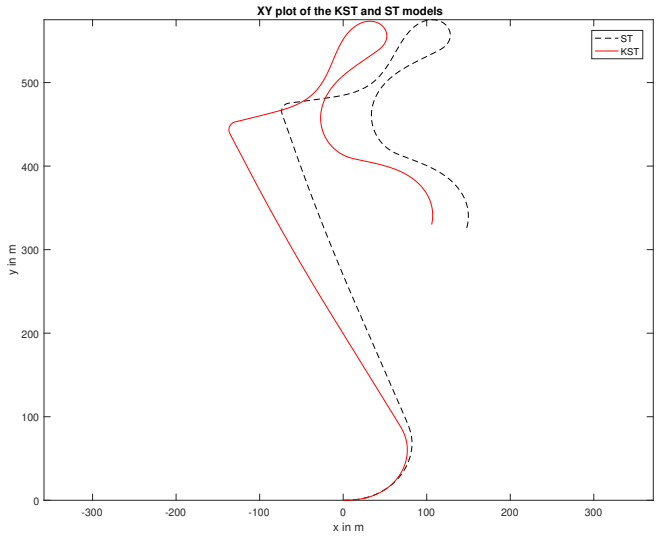
One more essential advantage of the UKF - which helps detect possible problems and increases confidence in that stochastic filter over the EKF - can be seen from the

Figures 32 and 33. The major axis of the covariance ellipse is tilted according to the vehicle's direction of motion. This can be further evidence to check if the UKF is working correctly since the equations of the x and y states show a relationship between sine and cosine of the angle ψ. On the other side, it is not a rule for the EKF due to the linearization process not reaching the covariance. For that reason, the slope of the ellipse and the vehicle's direction may not match.

6.1.2 DURBIN-WATSON TEST AND MODEL ERRORS

Next, an analysis is performed for the situation when the model used in the filter is different from the model used in the simulation. For this, the ST model remains the ground-truth and the measurement vector is generated normally distributed around it, while the filter is simulated considering a KST model.

Figure 34 – *Open-loop simulation of the KST and ST model for higher velocities:* in a cartesian plot, with (0,0) as the initial position coordinates, it is possible to see two trajectories. The red line represents the KST model simulation. The black dashed line represents the simulation of the ST model, that is considered the ground-truth for all of the experiments simulated. It is possible to see the difference between the two trajectories because the slip angle is not considered for the KST model. In a scenario with high velocities or low friction coefficients like this, the slip angle affects the trajectory considerably.



Source: Personal collection.

Since the model in this case is not representing perfectly the simulated ground-truth, the **Q** matrix should be increased to hold the effects caused by this mismodeling scenario. Table 11 shows the comparison for EKF and UKF for this scenario, while it presents how the DW test degrades to indicate that the model is incorrectly specified.

Table 11 – *MSE and DW results for a mismodeling process for the EKF and UKF:* The MSE is calculated using the simulated ground-truth as a reference. This ground-truth was simulated using the ST model and the filter results are given using the KST model. In this case, the DW test indicates the most appropriate **Q** value for the filter.

		EKF		UKF	
Q matrix	State	DW	MSE	DW	MSE
Q	x	1.4543	8.9485×10^{-4} m^2	0.96874	2.3484×10^{-3} m^2
	y	1.7070	9.4313×10^{-4} m^2	1.4913	1.6429×10^{-3} m^2
	ψ	1.7814	7.0463×10^{-5} rad^2	1.9559	2.0983×10^{-5} rad^2
	v	2.0511	1.3640×10^{-4} (m/s)^2	2.0360	1.3574×10^{-4} (m/s)^2
10**Q**	x	1.9284	3.3402×10^{-4} m^2	1.8949	3.5451×10^{-4} m^2
	y	2.0017	4.1470×10^{-4} m^2	1.9982	4.0951×10^{-4} m^2
	ψ	2.0518	2.9847×10^{-5} rad^2	2.0585	2.9919×10^{-5} rad^2
	v	2.0965	4.4995×10^{-4} (m/s)^2	2.0985	4.2932×10^{-4} (m/s)^2
100**Q**	x	2.394	4.0941×10^{-4} m^2	2.2386	4.0512×10^{-4} m^2
	y	2.2407	7.3084×10^{-4} m^2	2.2410	7.2675×10^{-4} m^2
	ψ	2.2483	8.7368×10^{-5} rad^2	2.2465	8.8362×10^{-5} rad^2
	v	2.2727	1.3846×10^{-3}(m/s)^2	2.2742	1.3655×10^{-3} (m/s)^2

For both filters, keeping the original value of the process noise covariance matrix **Q**, the DW test results have changed for this scenario. Even in this case that most of the values remain close to 2, when they are compared to the DW test values of Tables 7 and 9 (where there is no mismodeling) it is possible to identify their degradation. This indicates that there is a model error, which is true, as mentioned above. The DW test values can be improved by increasing the **Q** values. The best results for the DW is when 10**Q** is used to handle the error model effects, for this given example. With the new value of **Q** specified, the correspondent MSE values of the filters of Table 11 can be compared.

Table 12 – *Comparison of the MSE values of the estimated states using* 10**Q***:* in this case, the MSE values of Table 11 that corresponds to the optimal **Q** are compared using the values from the EKF as a reference: $\text{MSE}_{\text{UKF}}/\text{MSE}_{\text{EKF}} \times 100\%$.

Q matrix	State	$\frac{\text{MSE}_{\text{UKF}}}{\text{MSE}_{\text{EKF}}} \times 100\%$
10**Q**	x	106.13%
	y	98.74%
	ψ	100.20%
	v	95.41%

For this scenario, it is possible to infer through Table 12 that the EKF and the UKF have similar performances.

Next, an error is added in the input vector. The MSE and the DW test are also performed and the results are shown in Table 13.

Table 13 – *MSE and DW test results of a mismodeling process with input errors for the EKF and UKF:* The MSE is calculated using the simulated ground-truth as a reference. This ground-truth was simulated using the ST model and the filter results are given using the KST model. In addition to the model error, some errors were added to the input data. In this case, the DW test indicates the most appropriate **Q** value for the filter.

		EKF		UKF	
Q matrix	State	DW	MSE	DW	MSE
Q	x	0.79561	3.3391×10^{-3} m^2	0.0867	4.7812×10^{-2} m^2
	y	1.0294	4.2025×10^{-3} m^2	0.2546	2.925×10^{-2} m^2
	ψ	0.23488	4.1646×10^{-5} rad^2	0.1988	1.7426×10^{-5} rad^2
	v	0.10823	1.5140×10^{-1} (m/s)^2	0.0977	1.6031×10^{-1} (m/s)^2
10**Q**	x	1.9419	2.8824×10^{-4} m^2	1.6778	6.1357×10^{-4} m^2
	y	2.0453	3.0731×10^{-4} m^2	1.9744	4.9963×10^{-4} m^2
	ψ	1.0845	4.7830×10^{-4} rad^2	1.1063	4.7939×10^{-4} rad^2
	v	0.6995	1.5995×10^{-2} (m/s)^2	0.7011	1.5725×10^{-2} (m/s)^2
100**Q**	x	2.2421	4.0905×10^{-4} m^2	2.2404	4.0772×10^{-4} m^2
	y	2.2425	7.2756×10^{-4} m^2	2.2437	7.2022×10^{-4} m^2
	ψ	2.0273	1.2352×10^{-4} rad^2	2.0298	1.2375×10^{-4} rad^2
	v	1.8488	2.6921×10^{-3} (m/s)^2	1.8494	2.6747×10^{-3} (m/s)^2

Besides the model error, there is an error in the input vector. The DW test values for both filters have degraded even more for this scenario. The **Q** matrix needs to be even larger to incorporate these process errors. The DW test values can be improved by increasing the **Q** values. The best results for the DW test is when 100**Q** is used in order to handle the error model effects. For this scenario, the DW test values of the states ψ and v degrade even more, because these states depend directly on the inputs, according to Equation (31). With the new value of **Q** specified, the correspondent MSE values of the filters in Table 13 can be compared.

Table 14 – *Comparison of the MSE values of the estimated states using* 100**Q***:* in this case, the MSE values of Table 13 that correspond to the optimal **Q** are compared using the values from the EKF as a reference: $\text{MSE}_{\text{UKF}}/\text{MSE}_{\text{EKF}} \times 100\%$.

Q matrix	State	$\frac{\text{MSE}_{\text{UKF}}}{\text{MSE}_{\text{EKF}}} \times 100\%$
100**Q**	x	99.67%
	y	98.99%
	ψ	100.19%
	v	99.35%

Also for this scenario, it is possible to infer through Table 14 that the EKF and the UKF have similar performances.

Since there was only an error in the model, using 10**Q** was necessary to get good

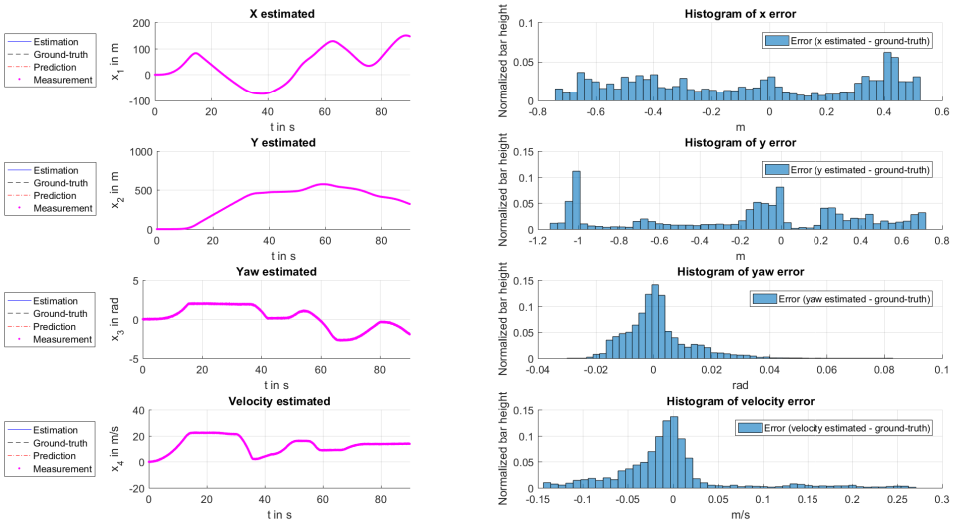
values for the DW test and guarantee that, even for mismodeling, the estimates were the best possible for that scenario. When more errors were incorporated, even using 100**Q**, the DW test values were still not optimal. These examples show that the DW results are a great tool for the detection of problems in the model and the more errors are incorporated, the more difficult it is to adjust **Q**, and consequently DW test values, for the optimal functionality of the filters.

6.1.3 SIMULATION OF SENSOR DELAYS

In this subsection the results for the simulated data when the delays are incorporated for the filter are shown for two situations: first when the delays are not compensated and second, when the delays are compensated. For the UKF and EKF and for both situations a delay of 5 time-steps, i.e. 50 ms, was considered.

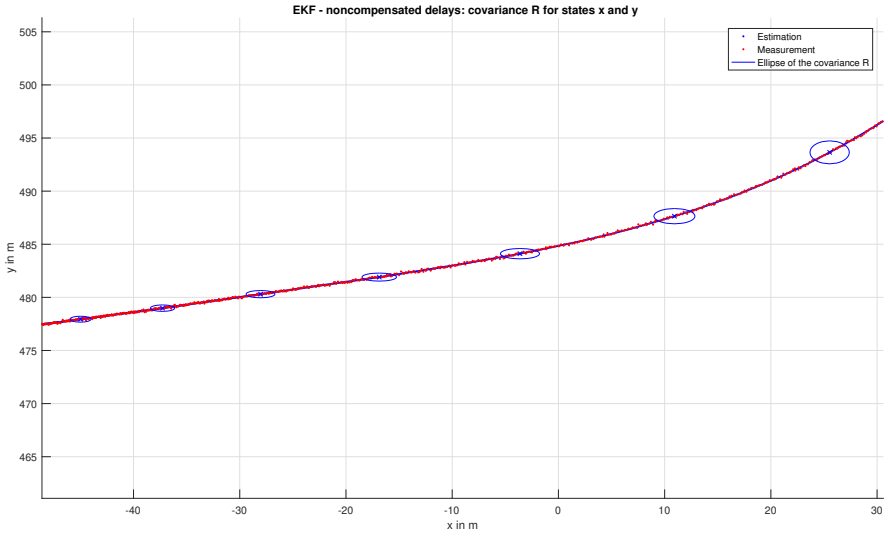
For the case that the delay is not compensated, the results are shown:

Figure 35 – *Estimation results using the EKF for noncompensated delayed measurements:* this figure represents, on the left side, the estimated states that are measured by the measurement data generated and its histograms of error on the right side.



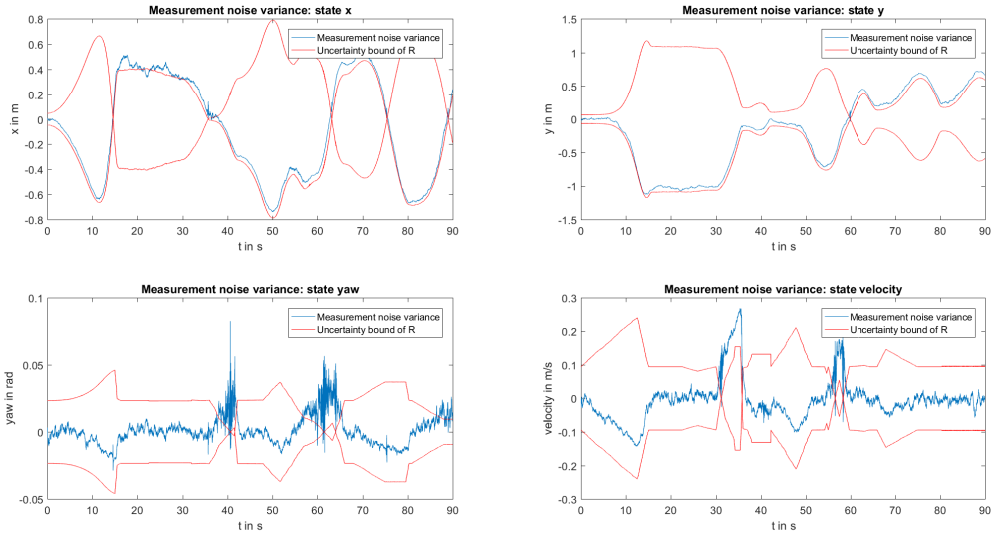
Source: Personal collection.

Figure 36 – *Estimation results using the EKF with the R matrix represented by ellipses varying along the time:* this figure shows a region of the whole trajectory more closely. Since the delays are not compensated, the ellipse of the covariance for the states x and y grows according to the vehicle's velocity.



Source: Personal collection.

Figure 37 – *Error over time bounded by the measurement uncertainties for the EKF when the delays are not compensated:* this figure shows the covariance matrix **R** varying along the time and the state error (calculated by a simple difference between the estimation and the ground-truth values) for noncompensated delay.



Source: Personal collection.

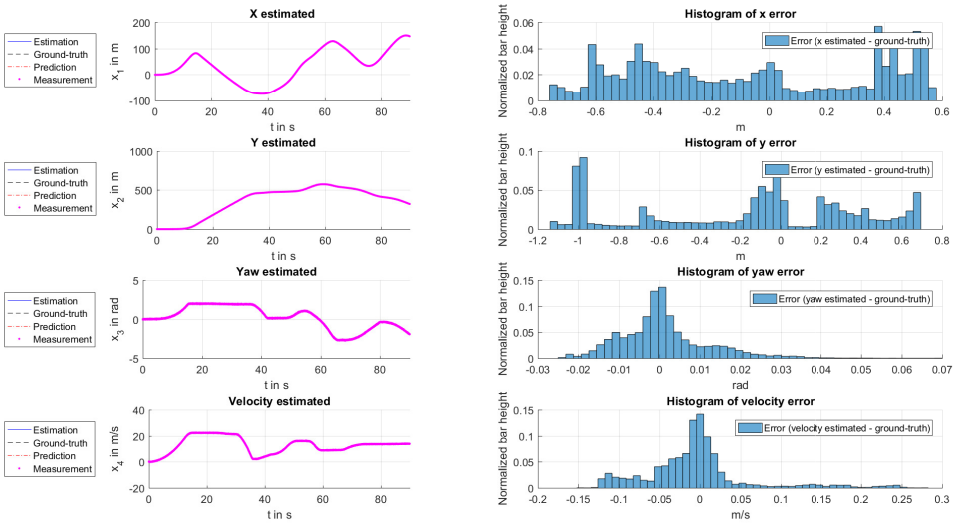
The MSE values for this noncompensation are presented in Table 15.

Table 15 – *MSE of the estimation using the EKF when the delay is not compensated:* in this case, the ground-truth simulated is used as reference to calculate the MSE values.

State	MSE	Unit
x	1.7087×10^{-1}	m^2
y	3.3711×10^{-1}	m^2
ψ	1.2282×10^{-4}	rad^2
v	4.0091×10^{-3}	$(\mathrm{m/s})^2$
$\dot{\psi}$	4.5090×10^{-6}	$(\mathrm{rad/s})^2$
β	3.2464×10^{-7}	rad^2

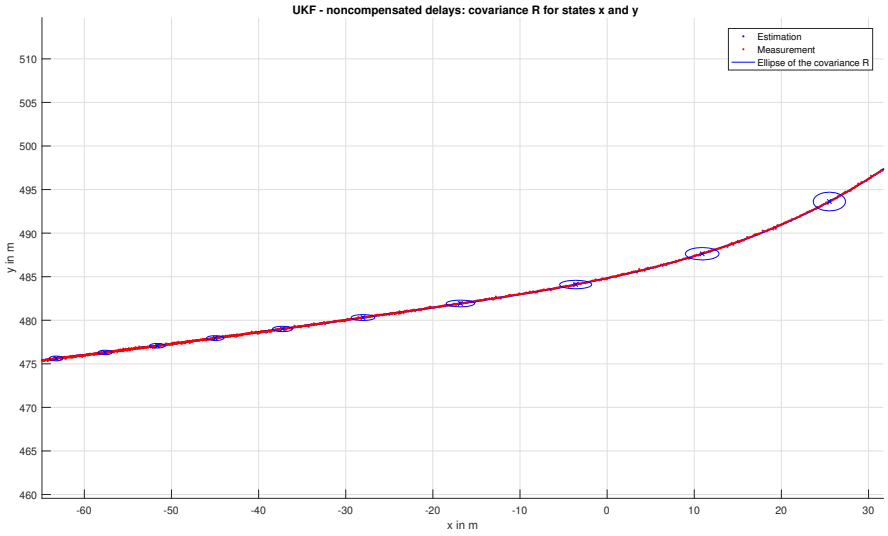
The same procedure occurs for the UKF when the delay is not compensated:

Figure 38 – *Estimation results using the UKF for noncompensated delayed measurements:* this figure represents, on the left side, the estimated states that are measured by the measurement data generated and its histograms of error on the right side.



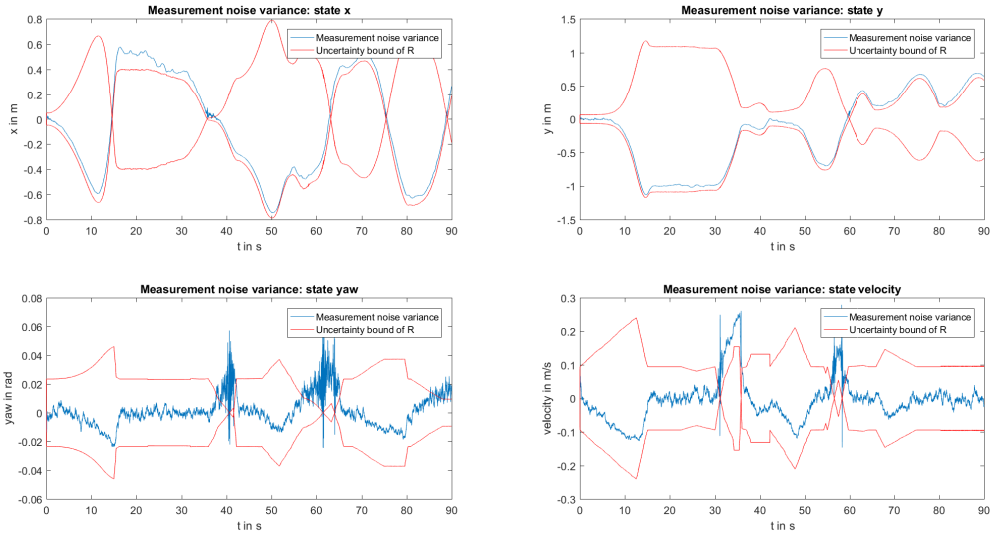
Source: Personal collection.

Figure 39 – *Estimation results using the UKF with the R matrix represented by ellipses varying along the time:* this figure shows a region of the whole trajectory more closely. Since the delays are not compensated, the ellipse of the covariance for the states x and y grows according to the vehicle's velocity.



Source: Personal collection.

Figure 40 – *Error over time bounded by the measurement uncertainties for the UKF when the delays are not compensated:* this figure shows the covariance matrix R varying along the time and the state error (calculated by a simple difference between the estimation and the ground-truth values) for noncompensated delay.



Source: Personal collection.

The MSE values are presented in Table 16 and compared with the MSE values

from the EKF considering the same scenario.

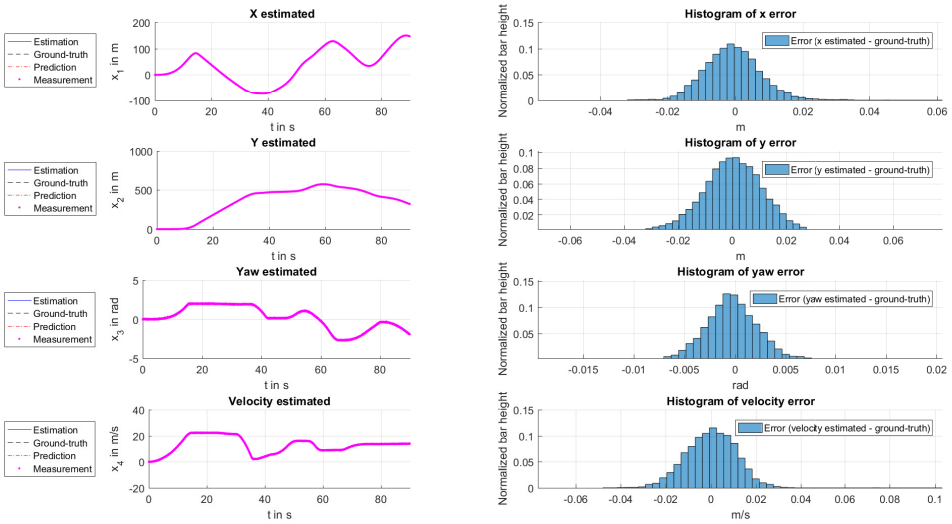
Table 16 – *MSE of the estimation using the UKF when a delay of 5 time-steps is incorporated but not compensated:* in addition to the MSE value of the estimations, the last column shows in percentual their differences for the same scenario, and is calculated as: $\text{MSE}_{\text{UKF}_{\text{nc}}}/\text{MSE}_{\text{EKF}_{\text{nc}}} \times 100\%$, where *nc* means *noncompensated*.

State	MSE	Unit	$\frac{\text{MSE}_{\text{UKF}_{\text{nc}}}}{\text{MSE}_{\text{EKF}_{\text{nc}}}} \times 100\%$
x	1.7574×10^{-1}	m^2	102.85%
y	3.1904×10^{-1}	m^2	94.64%
ψ	1.2028×10^{-4}	rad^2	97.93%
v	4.0318×10^{-3}	$(\text{m/s})^2$	100.57%
$\dot{\psi}$	1.0930×10^{-6}	$(\text{rad/s})^2$	24.24%
β	1.4905×10^{-8}	rad^2	4.59%

When the delay is noncompensated, it is possible to observe in Figures 35 and 38 that the histograms of the state estimation errors spread over the distance and lose their Gaussian distribution characteristic. This is expected since the error is calculated using the ground-truth against the estimations when the filters use the measurements from 5 backward time-steps. It is also important to point out that the comparisons in Table 16 show considerably better performance of the UKF for states that are not measured by the sensors, $\dot{\psi}$ and β.

The other possibility is to compensate those delays by recalculating the prediction and update steps over the time delay. For this situation, the EKF and UKF filters were implemented to compensate them.

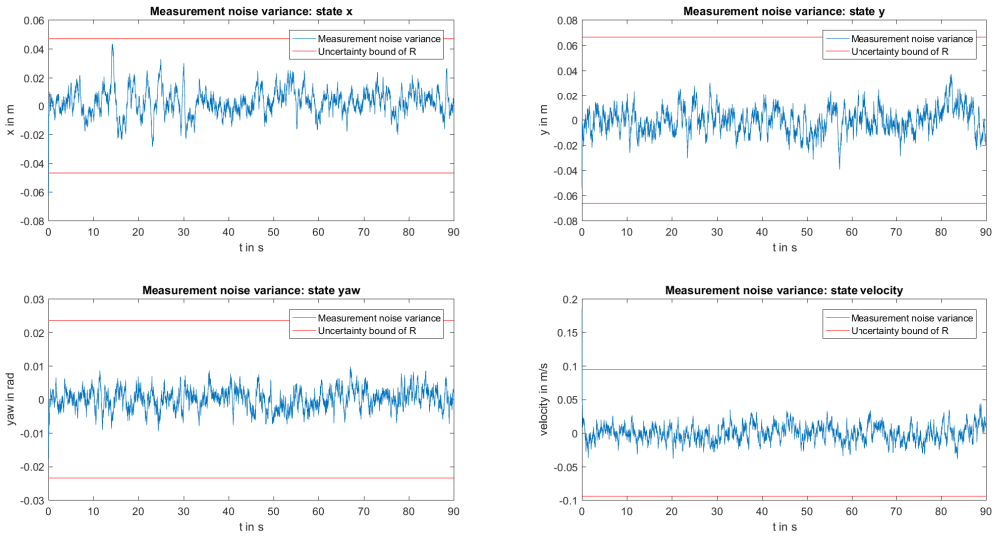
Figure 41 – *Estimation results using the EKF for compensated delayed measurements:* this figure represents, on the left side, the estimated states that are measured by the measurement data generated and its histograms of error on the right side.



Source: Personal collection.

For this scenario the **R** matrix is constant for all filtering times, as shown in Figure 42:

Figure 42 – *Error over time bounded by the measurement uncertainties for the EKF when the delays are compensated:* this figure shows the covariance matrix **R** constant along the time and the state error (calculated by a simple difference between the estimation and the ground-truth values) for compensated delays.



Source: Personal collection

The MSE values of the EKF estimations for compensated delays are shown in Table 17.

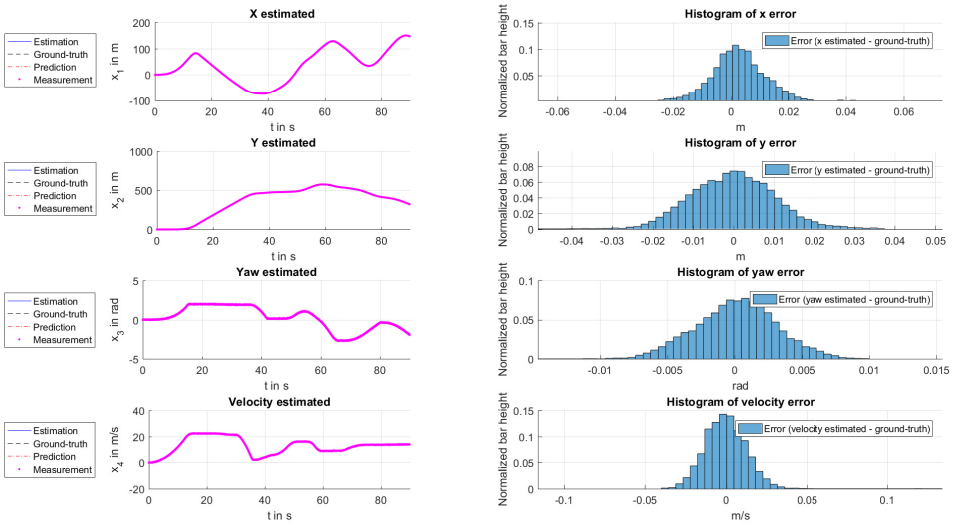
Table 17 – *MSE of the estimation using the EKF when a delay of 5 time-steps is incorporated and compensated:* in addition to the MSE value of the estimations, the last column shows in percentual the comparison of the MSE values of the EKF delayed and compensated, against the optimal EKF (without delays), presented in Table 8, and is calculated as: $\text{MSE}_{\text{EKF}_c}/\text{MSE}_{\text{EKF}_{\text{opt}}} \times 100\%$, where c means *compensated* and *opt* means *optimal*.

State	MSE	Unit	$\frac{\text{MSE}_{\text{EKF}_c}}{\text{MSE}_{\text{EKF}_{\text{opt}}}} \times 100\%$
x	1.5944×10^{-4}	m^2	100.00%
y	1.6326×10^{-4}	m^2	101.64%
ψ	1.7797×10^{-6}	rad^2	100.10%
v	1.4907×10^{-4}	$(\text{m/s})^2$	99.49%
$\dot{\psi}$	1.6877×10^{-7}	$(\text{rad/s})^2$	99.93%
β	8.287×10^{-7}	rad^2	99.98%

The values of Table 17 are expected because all of the delayed steps are compensated, meaning that the filter should performs close to the the optimal scenario, when there is no delay within the measurement vectors.

Next, the results for the UKF is also presented when the delay is compensated.

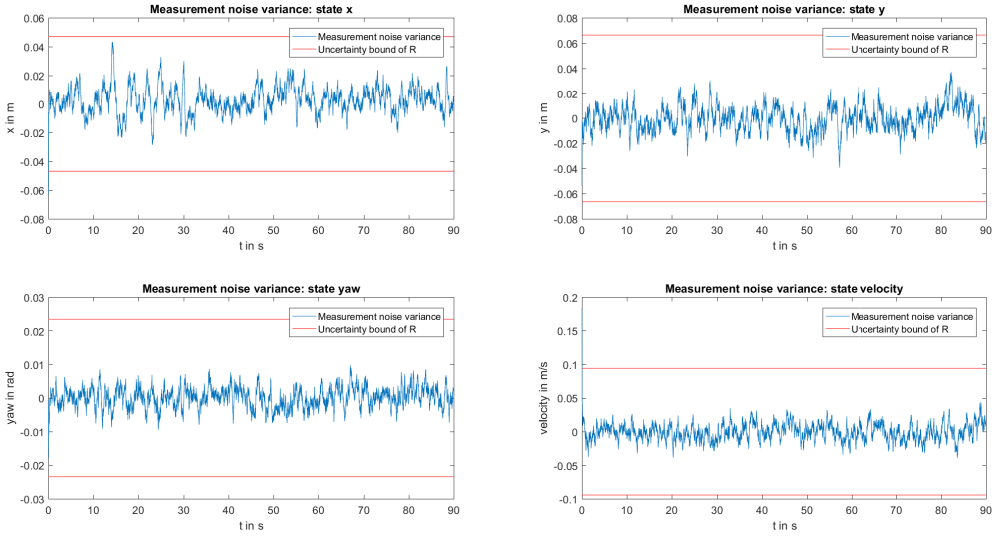
Figure 43 – *Estimation results using the UKF for compensated delayed measurements:* this figure represents, on the left side, the estimated states that are measured by the measurement data generated and its histograms of error on the right side.



Source: Personal collection.

For this scenario, the $\mathbf{R}$ matrix is also constant for all filtering times:

Figure 44 – *Error over time bounded by the measurement uncertainties for the UKF when the delays are compensated:* this figure shows the covariance matrix **R** varying along the time and the state error (calculated by a simple difference between the estimation and the ground-truth values) for compensated delay.



Source: Personal collection.

The MSE of the UKF estimations for compensated delays are shown in Table 18 and compared against the MSE values of the EKF for the same scenario.

Table 18 – *MSE of the estimation using the UKF when a delay of 5 time-steps is incorporated and compensated:* in addition to the MSE value of the estimations, the last column shows in percentual the comparison of the MSE values of the UKF for compensated delays against the EKF values for the same scenario, presented in Table 8, and is calculated as $\text{MSE}_{\text{UKF}_c}/\text{MSE}_{\text{EKF}_c} \times 100\%$, where c means *compensated*.

State	MSE	Unit	$\frac{\text{MSE}_{\text{UKF}_c}}{\text{MSE}_{\text{EKF}_c}} \times 100\%$
x	1.6925×10^{-4}	m^2	106.15%
y	1.2269×10^{-4}	m^2	75.15%
ψ	9.3791×10^{-6}	rad^2	52.75%
v	1.4842×10^{-4}	$(\text{m/s})^2$	105.56%
$\dot{\psi}$	8.1520×10^{-8}	$(\text{rad/s})^2$	4.83%
β	2.4394×10^{-9}	rad^2	2.94%

As the MSE values of the estimations for compensated delays should be very similar to their correspondent optimal situation, the values of Table 18 are expected because they are consistent with the optimal values and percentage values shown in Table 10.

For this situation it is possible to observe in Figures 41 and 43 that the histograms of the state errors were restored to their Gaussian distribution characteristics. Besides the better performance for the EKF in some states, up to this point it is shown that UKF performs better for the states that are not measured by the sensors, $\dot{\psi}$ and β.

The problem is that, in practice, the exact delay of the sequence is not perfectly known and some times the filter can consider more or less than the true value. In order to show the effects of this wrong compensation, Table 19 shows the results when a delay smaller than 50ms is considered, and in addition, compares these values with the perfectly compensated values using the EKF.

Table 19 – *MSE results for the EKF and UKF estimations when when a delay smaller than 50ms is considered:* in addition, these results are compared against the EKF results when the delays are perfectly compensated. The percentages are calculated as $\text{MSE}_{\text{UKF}_{30\text{ms}}}/\text{MSE}_{\text{EKF}_{50\text{ms}}} \times 100\%$.

	State	MSE 30ms	Unit	$\frac{\text{MSE}_{\text{UKF}_{30\text{ms}}}}{\text{MSE}_{\text{EKF}_{50\text{ms}}}} \times 100\%$
EKF	x	2.6397×10^{-2}	m^2	165.56 ‰
	y	5.3704×10^{-2}	m^2	328.94‰
	ψ	2.9168×10^{-5}	rad^2	163.18%
	v	7.2519×10^{-4}	$(\text{m/s})^2$	517.34%
	$\dot{\psi}$	2.488×10^{-6}	$(\text{rad/s})^2$	147.41%
	β	1.3765×10^{-7}	rad^2	166.10%
UKF	x	2.6720×10^{-2}	m^2	167.58‰
	y	5.4509×10^{-2}	m^2	33,387%
	ψ	2.6058×10^{-5}	rad^2	146.41%
	v	7.6086×10^{-4}	$(\text{m/s})^2$	543.19%
	$\dot{\psi}$	2.7039×10^{-7}	$(\text{rad/s})^2$	16.21%
	β	3.4139×10^{-9}	rad^2	4,12%

Table 20 shows the results when a delay larger than 50ms is considered and in addition, compares these values with the perfectly compensated values using the EKF.

Table 20 – *MSE results for the EKF and UKF estimations when a delay larger than 50ms is considered:* in addition, these results are compared against the EKF results when the delays are perfectly compensated. The percentages are calculated as $\text{MSE}_{\text{UKF}_{70\text{ms}}}/\text{MSE}_{\text{EKF}_{50\text{ms}}} \times 100\%$.

	State	MSE 70ms	Unit	$\frac{\text{MSE}_{\text{UKF}_{70\text{ms}}}}{\text{MSE}_{\text{EKF}_{50\text{ms}}}} \times 100\%$
EKF	x	2.7173×10^{-2}	m^2	170.42‰
	y	5.4621×10^{-2}	m^2	334.56‰
	ψ	3.9721×10^{-5}	rad^2	223.18%
	v	8.7196×10^{-4}	$(\text{m/s})^2$	622.51%
	$\dot{\psi}$	1.8088×10^{-6}	$(\text{rad/s})^2$	107.20%
	β	9.5361×10^{-8}	rad^2	115.71%
UKF	x	2.7067×10^{-2}	m^2	169.76‰
	y	5.3653×10^{-2}	m^2	328.63‰
	ψ	2.5706×10^{-5}	rad^2	144.44%
	v	8.4908×10^{-4}	$(\text{m/s})^2$	606.18%
	$\dot{\psi}$	2.1637×10^{-7}	$(\text{rad/s})^2$	12.82%
	β	3.7963×10^{-9}	rad^2	4.58%

Analysing these results, it is possible to confirm that even for wrong compensation of the delay, the UKF performs better based on the states that are not measured by the sensors when compared to the EKF with perfect compensation. At this point it is possible to see that, even wrong compensations with larger or smaller assumptions for the delay, is better than simply ignore the delay and do not compensate at all. It is important to point out that the position states, x and y, are more affected by the wrong compensation of the delays.

It is also important to point out that in this experiments it is not possible to determine if underestimating delays is better than overestimating or vice-versa. To determinate this, a larger set of data needs to be analysed.

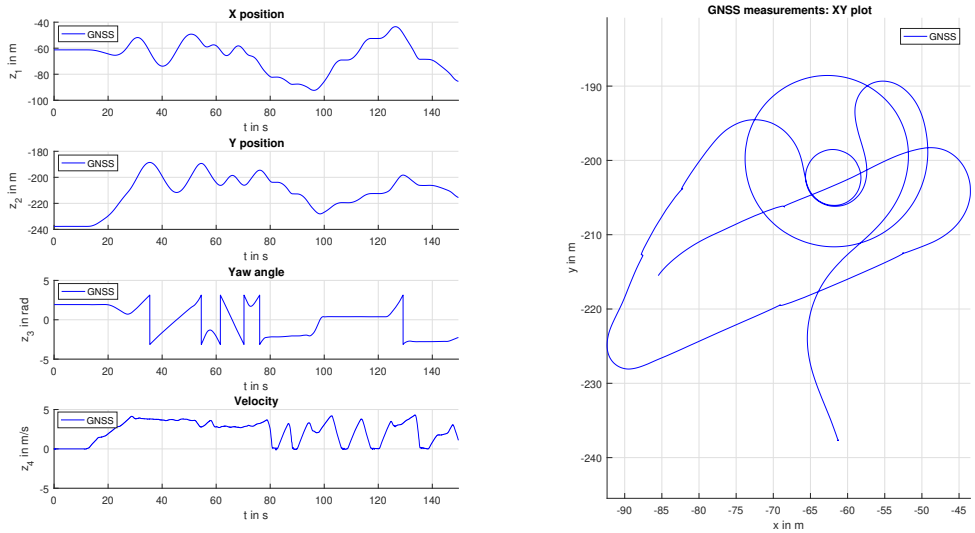
6.2 REAL DATA RESULTS

This section presents the results of the study with real data. First the results that should be identified before the state estimation itself. The plots of the data from the sensors and the trajectory obtained from an open-loop simulation of the KST and ST models are presented. Afterwards, the stochastic filtering results including the incorporation of delays.

6.2.1 SENSOR DATA

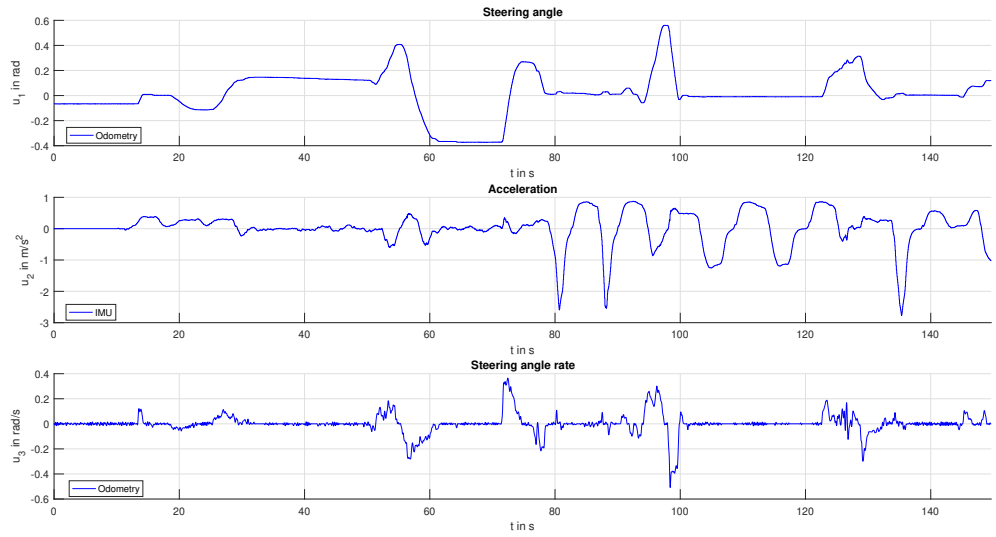
The GNSS, IMU and odometry data obtained from the test vehicle are shown in Figures 45 and 46, respectively.

Figure 45 – *GNSS data obtained from the test vehicle:* the left side of this figure shows the GNSS data for the states x, y, ψ and v. The measurement vector $\mathbf{z}$ is composed by these GNSS measurements. The right side of the figure shows the trajectory in cartesian coordinates done by the test vehicle.



Source: Personal collection.

Figure 46 – *IMU and odometry data obtained from the test vehicle:* the input vector for the KST model, $\begin{bmatrix}\delta & a_{\text{long}}\end{bmatrix}^\intercal$, and for the ST model, $\begin{bmatrix}\delta & a_{\text{long}} & \dot{\delta}\end{bmatrix}^\intercal$, are composed by these data.



Source: Personal collection.

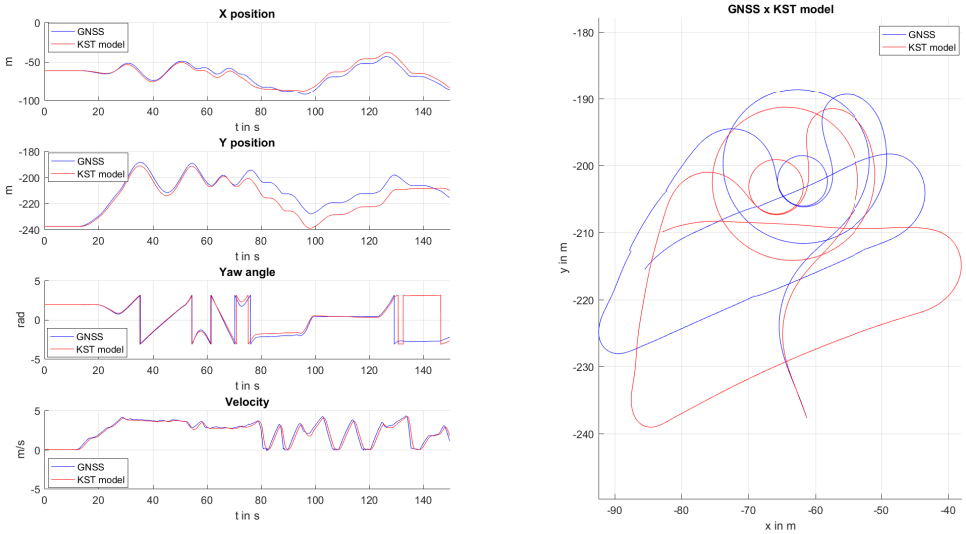
6.2.2 MODEL VALIDATION/TESTING

Since the parameter used in the KST model is known, as seen in Table 21, the simulation can be done with the given inputs. The result of the open-loop simulation compared with the GNSS data is shown in Figure 47.

Table 21 – *List of parameters used in the KST model:* the only parameter required for the KST model is the length of the wheelbase of the test vehicle.

Parameters required			
Name	Symbol	Unit	Test vehicle value
Length of the wheelbase	l_{wb}	[m]	1.588

Figure 47 – *KST model compared with the GNSS data:* with the given parameter, the open-loop simulation was performed with the input data using the KST model, the result is compared with the GNSS data.



Source: Personal collection.

This result is not good, since the simulated data diverges when compared to the GNSS measurements, mainly in the curves sections. This is caused by some errors in the parameters or in the input data. As the only parameter used in the simulation is the length of the test vehicle's wheelbase, it is possible to infer that there are some problems within the input data. Analysing the plot of Figure 47 it is possible to see that the input data is given with smaller acceleration and steering values than they should actually have, based on the trajectory read by the GNSS measurements.

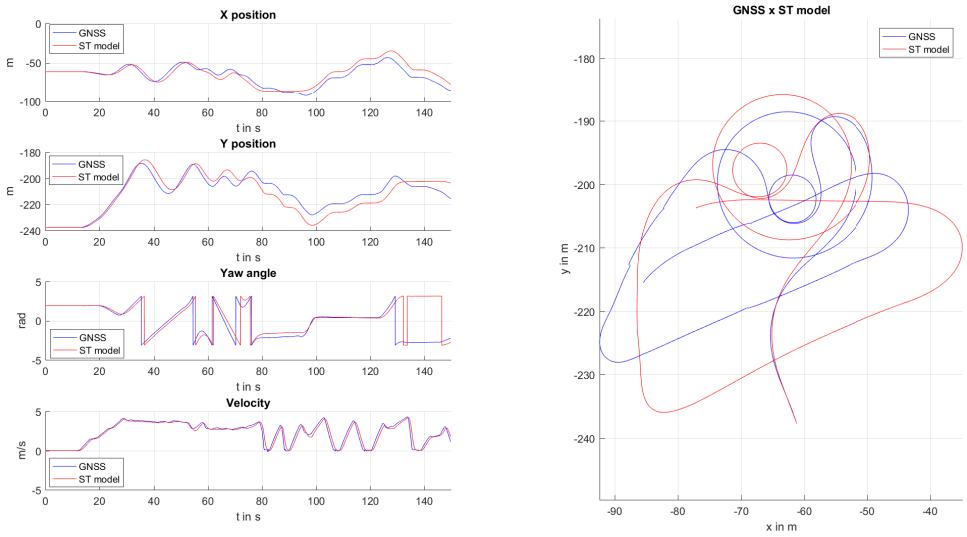
For the open-loop simulation of the ST model, the missing parameters were estimated and the results are presented in Table 22.

Table 22 – *List of parameters used in the ST model:* some of the parameters required for the ST model were not available (n.a.) beforehand. They are represented in the column *Known parameters*. A parameter estimation was performed to estimate the missing parameters and the result is presented in the column *Estimated values*.

Parameters required		Parameters values	
Symbol	Unit	Known parameters	Estimated values
m	10^3[kg]	≈ 4.5	–
I_z	10^3[kg m^2]	n.a.	7,199.5252
l_f	[m]	≈ 0.794	–
l_r	[m]	≈ 0.794	–
h_cg	[m]	n.a.	1.3213
$C_\mathrm{S,f}$	[rad^{-1}]	n.a.	0.0270
$C_\mathrm{S,r}$	[rad^{-1}]	n.a.	0.0270
μ	[–]	n.a.	25.2035

With all the parameters known, the open-loop simulation is done with considering the given inputs. Figure 48 also shows a comparison between the GNSS data and the correspondent result of the states.

Figure 48 – *ST model compared considering the GNSS data:* with the known and estimated parameters, the open-loop simulation was performed with the input data using the ST model, the result is compared with the GNSS data.



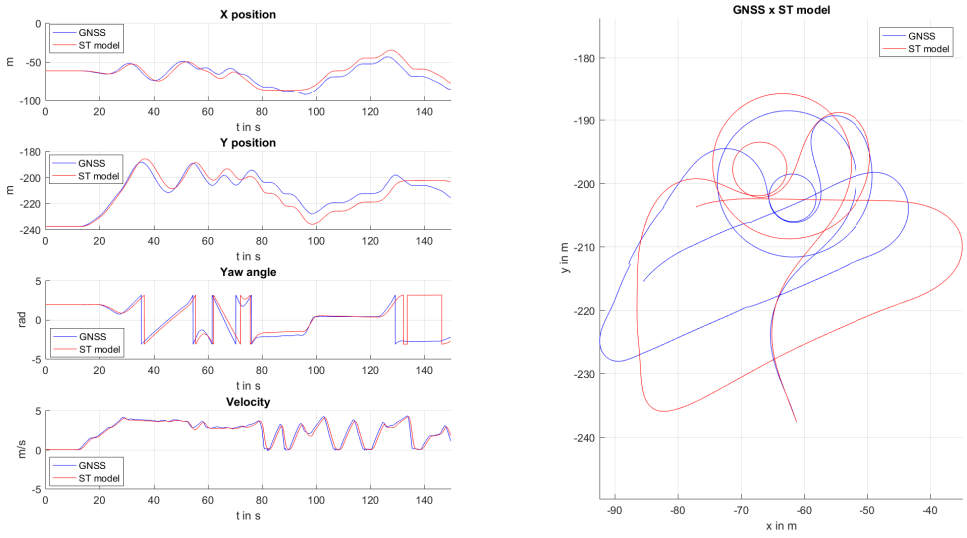
Source: Personal collection.

This result is also not satisfactory since the simulated data diverges from the GNSS measurements in the curves. This situation is worsened because besides the problems within the inputs, most of the parameters required in the model are not known beforehand

and the others that were assumed to be known were actually approximated values. The parameter estimation is used as an optimization tool for the whole time steps to try to give the optimal results with the known data about the model and the vehicle; if the known data is poorly specified, the quality of the parameter estimation degrades. Analysing the plot of Figure 48 it is possible to confirm that the input data is given with smaller acceleration and steering values than they should actually have, according to the GNSS measurements.

The ST model has 6 DoF and the remaining states that have no correspondent GNSS measurements are presented in Figure 49.

Figure 49 – *Simulated results of $\dot{\psi}$ and β from the open-loop simulation using the ST model.*



Source: Personal collection.

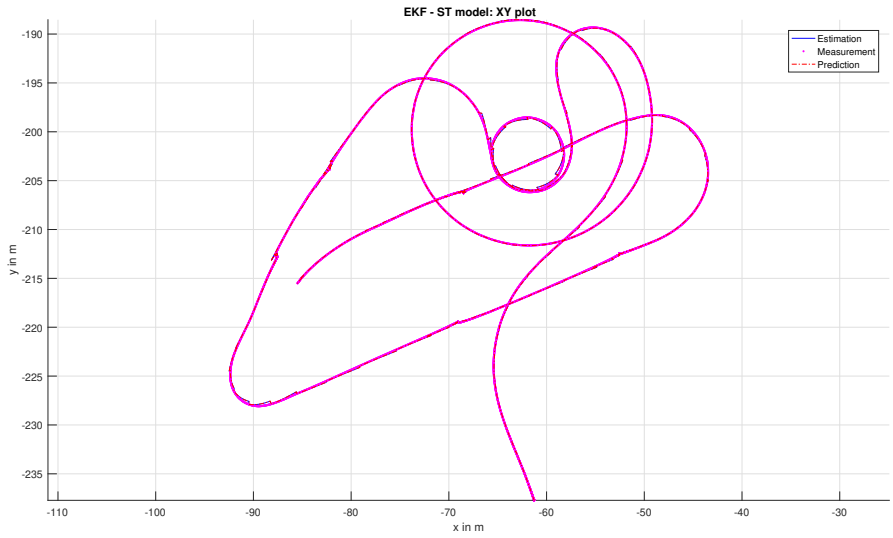
6.2.3 FILTERING PROCESS

In practice, rarely all the states can be measured. Here, the results for the ST model will be presented, where only four of six states were measured by GNSS, exactly like the simulation results from Section 6.1.1.

The Figures 50 – 63 show the results of the filtering process of EKF and UKF when the GNSS gives new readings about the state of the test vehicle in a different rates of IMU and ECU signals. The GNSS updates at every 1 second while the IMU and ECU at each 10 ms.

Starting from the EKF, the results are presented:

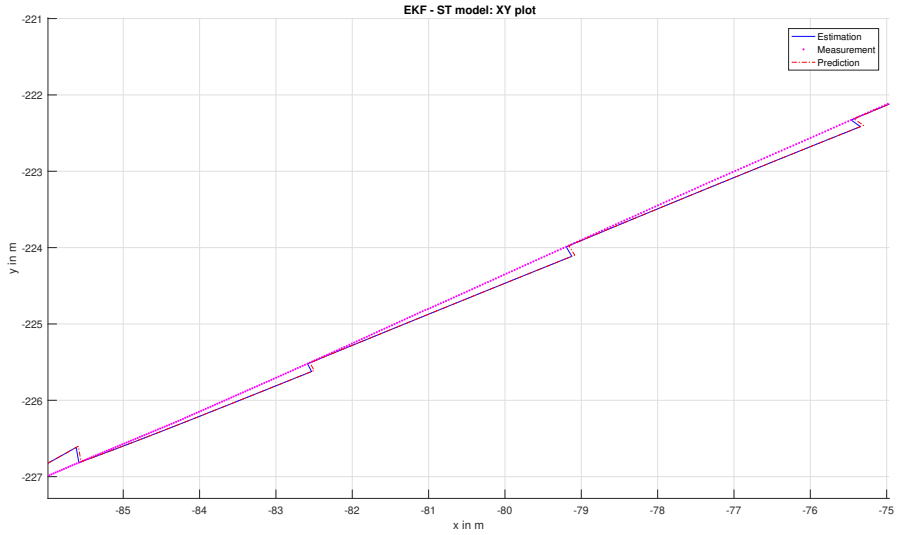
Figure 50 – *Estimation of the trajectory of the vehicle test using the EKF:* this figure shows the XY plot for the estimates, the measurements, and the predictions of the filter. Note that the estimations are equal to the predictions when the GNSS values are not available.



Source: Personal collection.

As the number of samples is very large, it is not possible to notice the mismatch of the curves. For that reason, Figure 51 represents a region of the trajectory more closely:

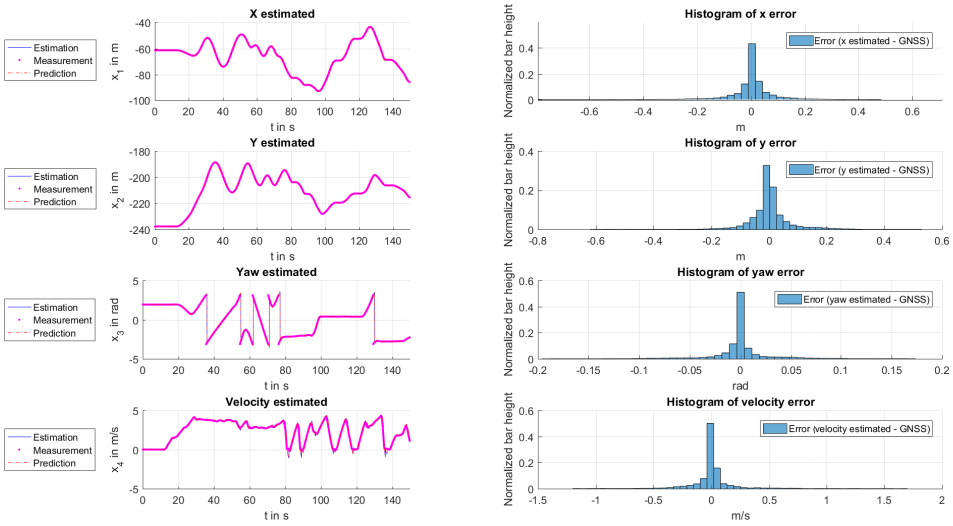
Figure 51 – *Region of the estimated sequence using the EKF:* this figure has the purpose of better visualization of the EKF results. When the GNSS is not available, the filter consider only predictions and, for that reason, the estimations start to exclusively follow the model. When the GNSS measurement arrives, it is possible to observe the correction of the estimation: the estimation gets closer to the GNSS measurements.



Source: Personal collection.

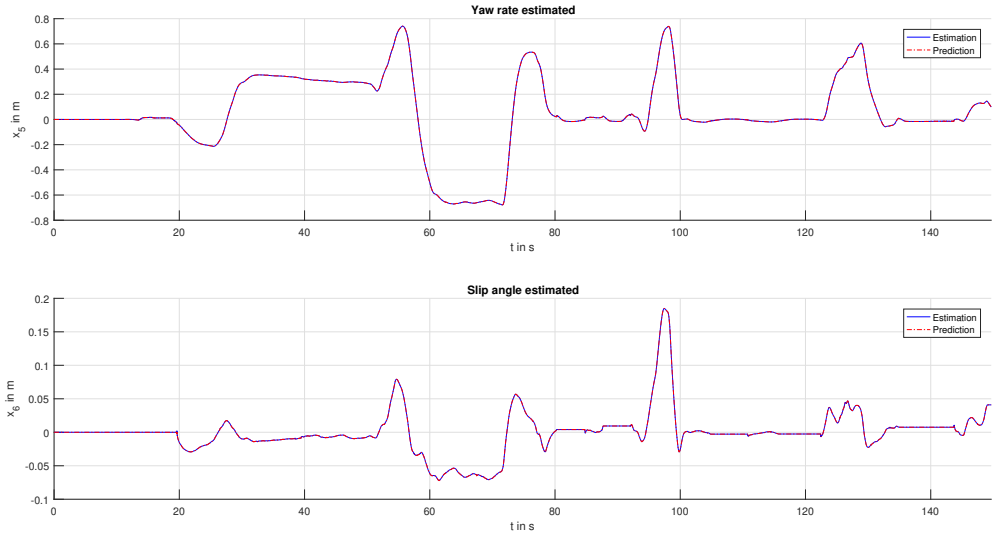
The six states can be analysed over time, as shown in Figures 52 and 53.

Figure 52 – *Estimation results using the EKF and real data:* this figure represents, on the left side, the estimated states measured by the GNSS sensor and its histograms of error on the right side using the measurement data as reference.



Source: Personal collection.

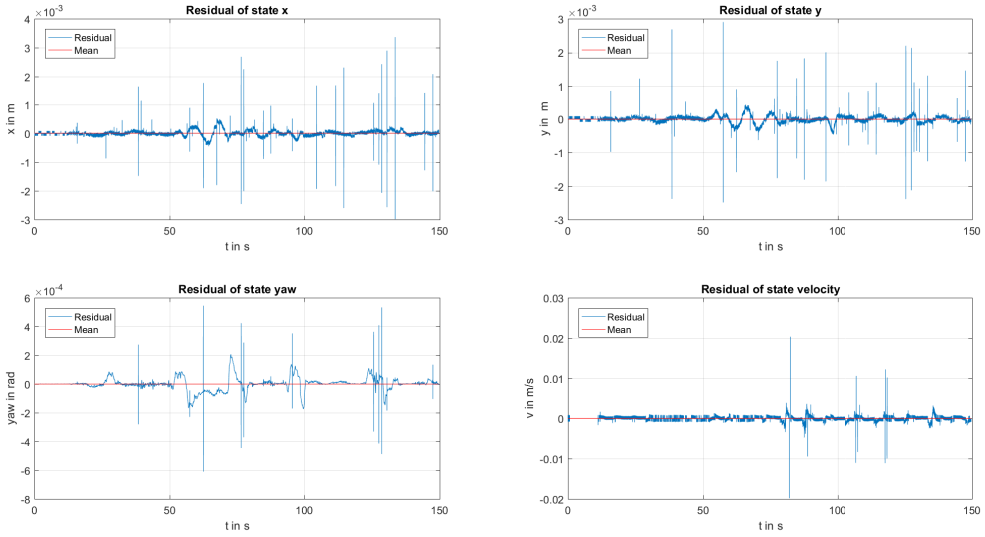
Figure 53 – *Estimation of the not measured states using EKF:* this figure represents the estimated states that are not measured by the GNSS.



Source: Personal collection.

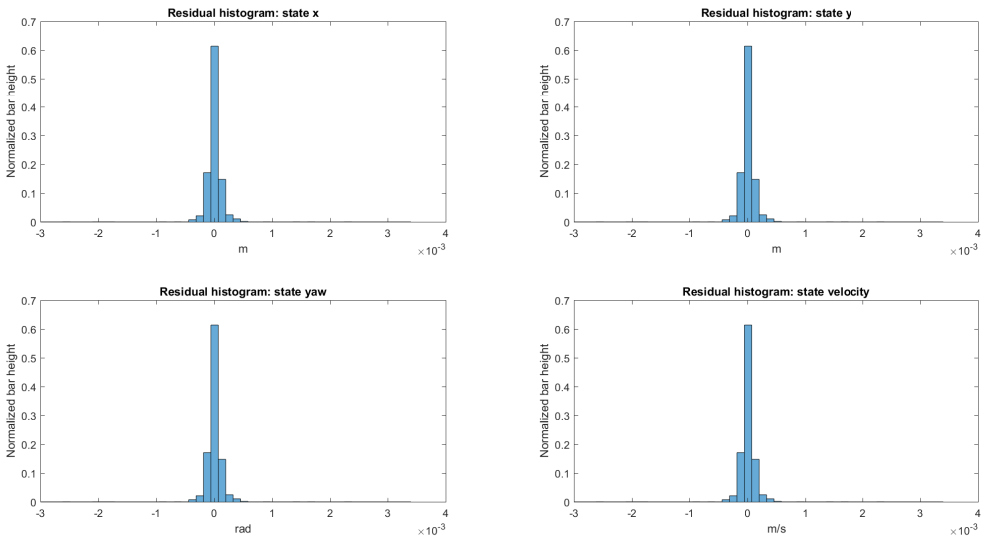
The tests of consistency related to the EKF residuals are presented in the Figures 54 and 55 and the autocorrelation test is observed in Figure 56.

Figure 54 – *Residuals of the EKF estimations for real measurements:* they are calculated as the difference between the estimated values and the GNSS data for every time step. The mean of the whole sample should be close to zero.



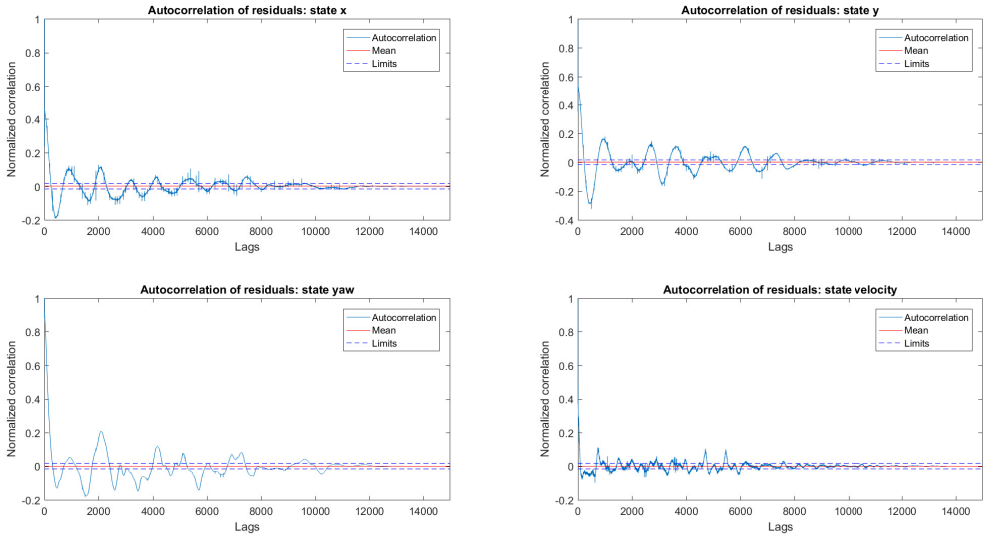
Source: Personal collection.

Figure 55 – *Histogram of the residuals of the EKF estimations.*



Source: Personal collection.

Figure 56 – *Autocorrelation of the residuals of the EKF estimations:* the normalized autocorrelations are plotted with their limits. More than 5% of the points are outside the limits



Source: Personal collection.

The correspondent values of the autocorrelation mean and the DW test values are presented in Table 23.

Table 23 – *Means of the autocorrelation and DW test results for the EKF using real data.*

State	DW test result	Mean of the autocorrelation
x	1.532	9.213×10^{-4}
y	1.298	5.505×10^{-4}
ψ	0.236	5.376×10^{-5}
v	1.258	2.270×10^{-5}

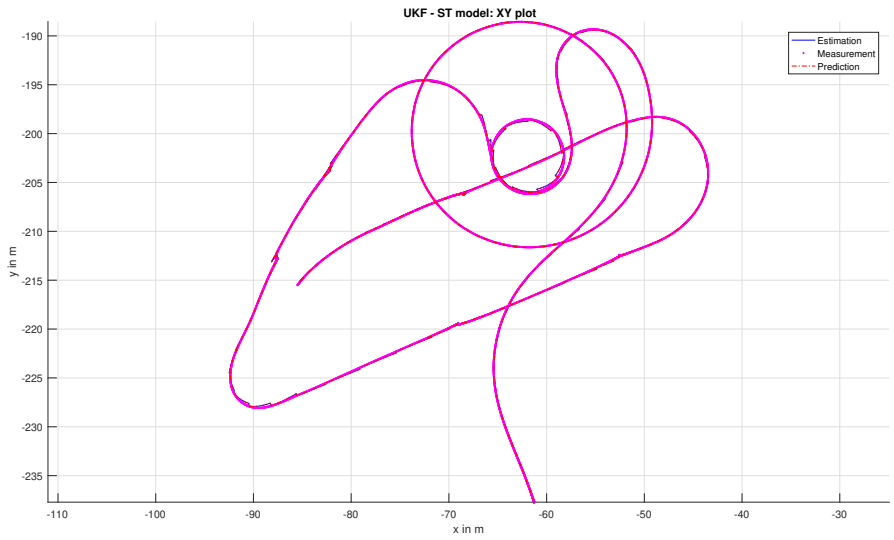
Next, the MSE values calculated using the GNSS data as ground-truth are presented in Table 24.

Table 24 – *MSE of the estimated states using the EKF and GNSS data:* in this case, the GNSS is used as a reference to calculate the MSE.

State	MSE	Unit
x	5.8814×10^{-3}	m^2
y	5.2173×10^{-3}	m^2
ψ	7.6722×10^{-4}	rad^2
v	4.1729×10^{-2}	$(m/s)^2$

Next, with a similar procedure of the last results, the results of the UKF using real data from the sensors are presented.

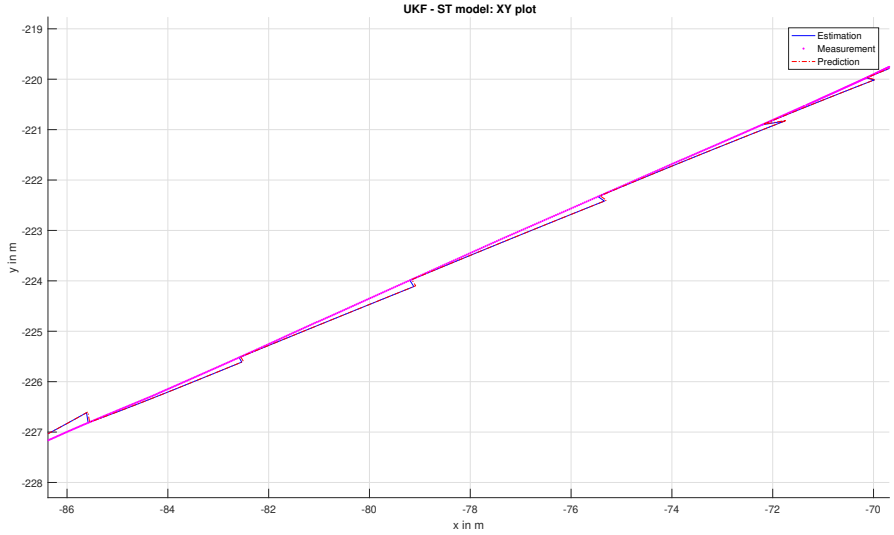
Figure 57 – *Estimation of the trajectory of the vehicle test using the UKF:* this figure shows the plot for the estimates, the measurements and the predictions of the filter. The estimations are equal to the predicted results when the GNSS values are not available.



Source: Personal collection.

Here it is also not possible to notice the mismatch of the curves. For that reason, Figure 58 represents a region of the trajectory more closely.

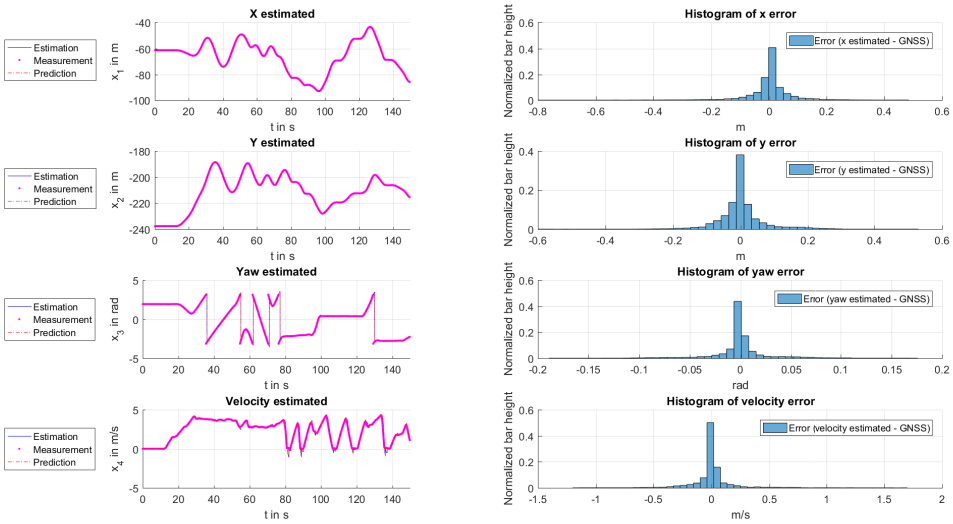
Figure 58 – *Region of the trajectory of the test vehicle:* this figure is for a better visualization of the UKF results.



Source: Personal collection.

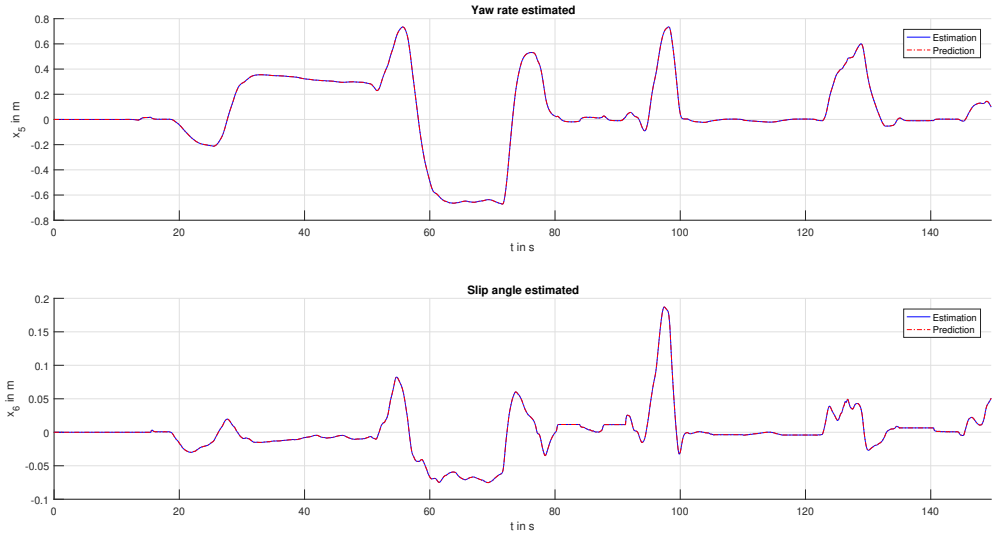
The six estimated states are shown in Figures 59 and 60, where four of them are in Figure 59 and the other two are in Figure 60.

Figure 59 – *Estimation results using the UKF and real data:* this figure represents, on the left side, the estimated states measured by the GNSS sensor and its histograms of error on the right side using the measurement as data as reference.



Source: Personal collection.

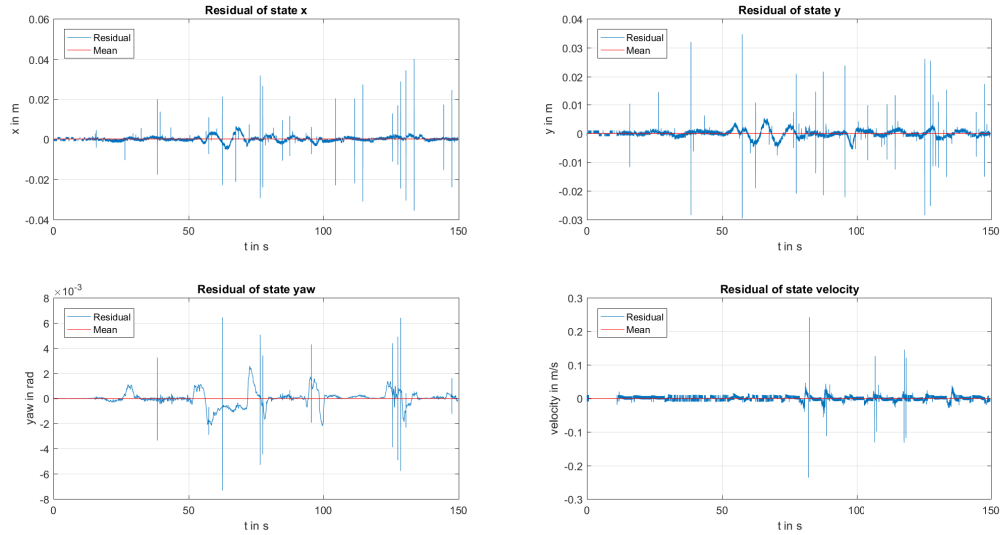
Figure 60 – *Estimation of the not measured states using UKF:* this figure represents the estimated states that are not measured by the GNSS.



Source: Personal collection.

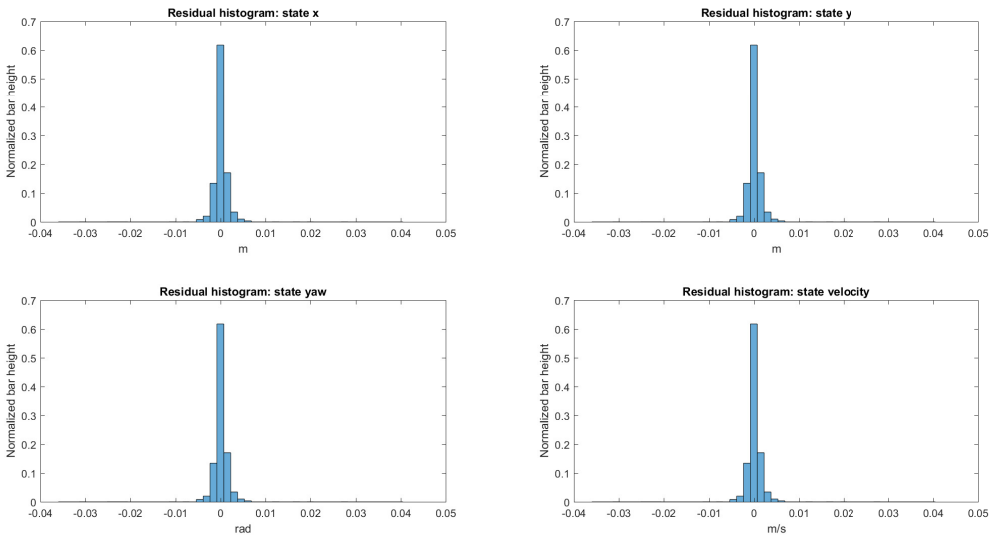
The tests of consistency also related to the UKF residuals are presented below, in the Figures 61 and 62 and the autocorrelation test is in Figure 63.

Figure 61 – *Residuals of the UKF estimations for real measurements:* they are calculated as the difference between the estimated values and the GNSS data for every time step. The mean of the whole sample is close to zero.



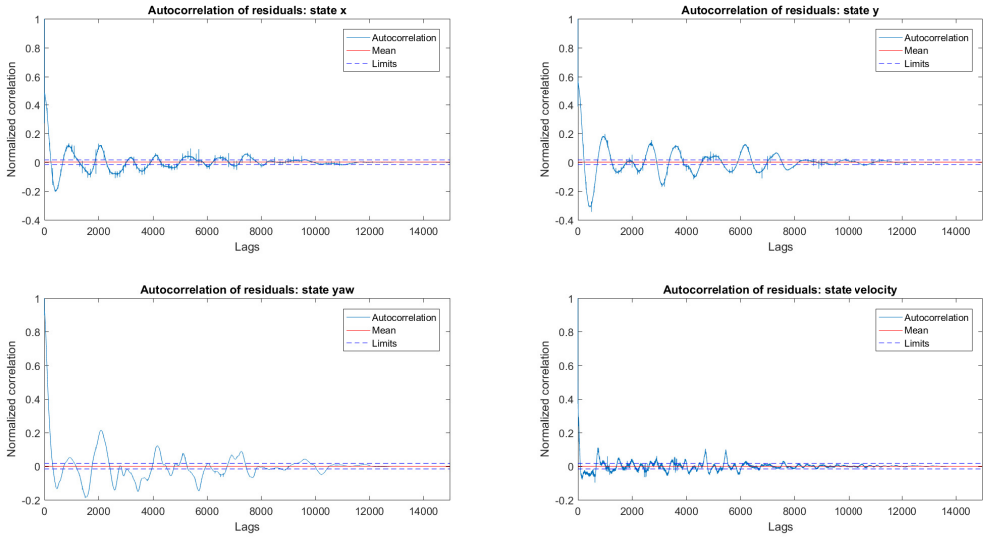
Source: Personal collection.

Figure 62 – *Histogram of the residuals of the UKF estimations.*



Source: Personal collection.

Figure 63 – *Autocorrelation of the residuals of the UKF estimations:*the normalized autocorrelations are plotted with their limits. More than 5% of the points are outside the limits.



Source: Personal collection.

The results of the autocorrelation mean and the DW test values with the real data are also presented for the UKF estimations and are shown in Table 25.

Table 25 – *Means of the autocorrelation and DW test results for the UKF using real data.*

State	DW result	Means of the autocorrelation
x	1.457	1.156×10^{-3}
y	1.225	6.254×10^{-4}
ψ	0.204	1.321×10^{-4}
v	1.258	2.321×10^{-5}

Now, the MSE values calculated for the UKF estimations are presented in Table 26. Those values are calculated using the GNSS data and a percentual comparison is done against the EKF results for the same scenario.

Table 26 – *MSE of the estimated states using the UKF and GNSS data:* in this case, the GNSS is used as a reference to calculate the MSE. The last column is the percentual comparison of the MSE values of the UKF against the MSE values of the EKF, presented in Table 8 and is calculated as $\text{MSE}_{\text{UKF}}/\text{MSE}_{\text{EKF}} \times 100\%$.

State	MSE	Unit	$\frac{\text{MSE}_{\text{UKF}}}{\text{MSE}_{\text{EKF}}} \times 100\%$
x	5.9585×10^{-3}	m^2	101.12%
y	5.2197×10^{-3}	m^2	100.05%
ψ	9.7866×10^{-5}	rad^2	124.94%
v	4.1590×10^{-2}	$(\text{m/s})^2$	99.67%

The filters results exposed in the Section 6.2 have shown non satisfactory results about the data estimated compared to the GNSS measurements. Despite the correct filter's functionality proven by the simulated data in the previous section, the consistency tests performed for the real data indicate problems with the model. This was expected as a consequence of the poor ST model specification caused by the errors in the input data and approximations of the known parameters. For both filters, the autocorrelation and the DW tests indicated correlation. The normalized points in autocorrelation test are outside the limits, as shown in Figures 56 and 63, and the results of the DW test confirms it as the EKF and UKF values for this test are totally degraded.

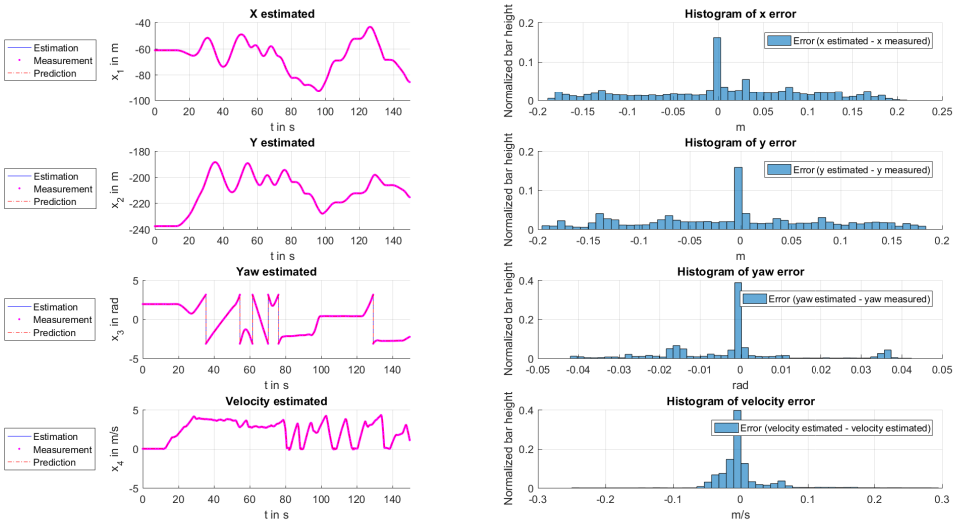
As a consequence of it, both of the filters worked correctly, but not in an optimal way. This can be seen in Figures 51 and 57. An optimal filter should result in better estimates for this kind of application at the intervals when GNSS measurements are not available. Moreover, they computed very similar results, according to the MSE results presented in Tables 24 and 26.

The next results show the filters performances when delays in the measurement vector are incorporated. For simplification, in this scenario it is considered that the GNSS gives new readings about the state of the test vehicle with the same frequency of the IMU and ECU signals. It means that they update at every 10 ms.

With a similar procedure to the simulated results, the results are presented for two situations: first when the delays are not compensated and second, when the delays are compensated. For the UKF and EKF and for both situations listed above, a delay of 5 time-steps was considered again.

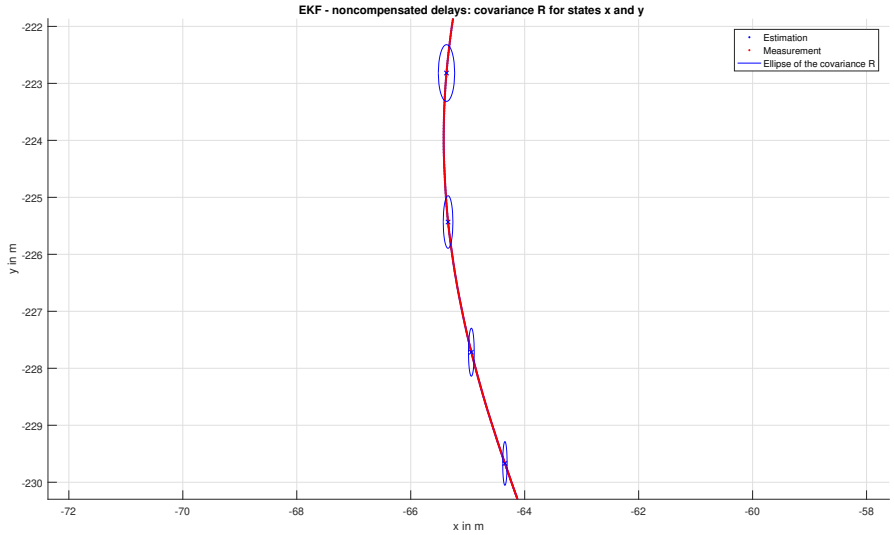
First, the results of the EKF are shown when the delay is not compensated using real data:

Figure 64 – *Estimation results using the EKF for noncompensated real delayed measurements:* this figure represents, on the left side, the estimated states that are corrected by the GNSS and its histograms of error on the right side.



Source: Personal collection.

Figure 65 – *Region of the trajectory of the test vehicle showing the ellipses of the covariance estimated using the EKF for noncompensated delays.*



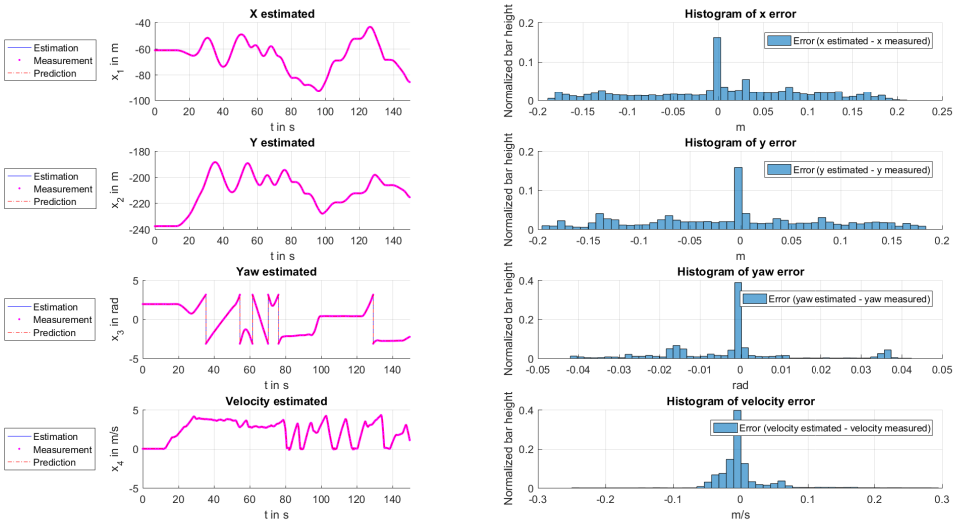
Source: Personal collection.

Table 27 – *MSE of the estimated states using the EKF and GNSS data for noncompensated delay:* in this case, the GNSS is used as a reference to calculate the MSE.

State	MSE	Unit
x	8.6545×10^{-3}	m^2
y	8.6286×10^{-3}	m^2
ψ	2.5807×10^{-4}	rad^2
v	1.1468×10^{-3}	$(\mathrm{m/s})^2$

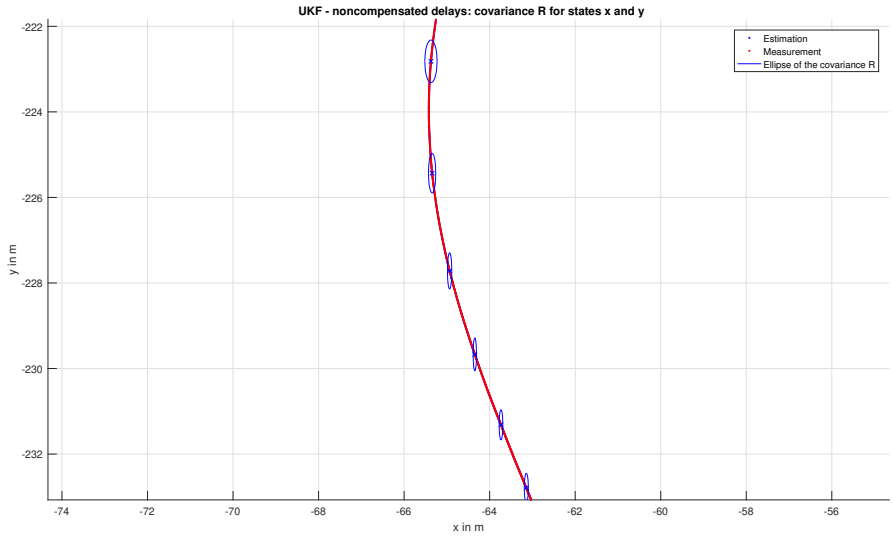
Next, the results of the UKF when the delay is noncompensated are presented:

Figure 66 – *Estimation results using the UKF for noncompensated real delayed measurements:* this figure represents, on the left side, the estimated states that are corrected by the GNSS and its histograms of error on the right side.



Source: Personal collection.

Figure 67 – *Region of the trajectory of the test vehicle showing the ellipses of the covariance estimated using the UKF for noncompensated delays.*



Source: Personal collection.

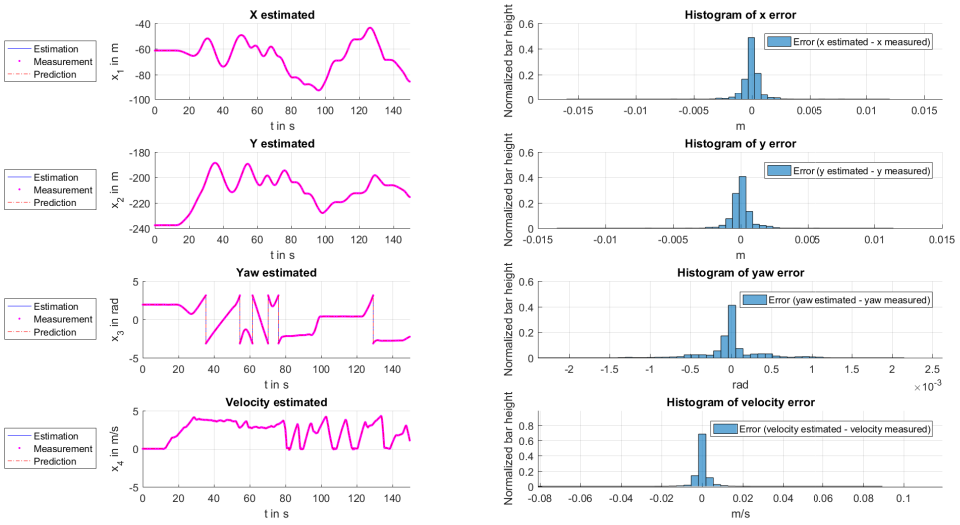
Table 28 – *MSE of the estimated states using the UKF and GNSS data and noncompensated delay:* in this case, the GNSS is used as a reference to calculate the MSE. The last column is the percentual comparison of the MSE values of the UKF against the MSE values of the EKF for the same scenario, presented in Table 27, $\text{MSE}_{\text{UKF}_{\text{nc}}}/\text{MSE}_{\text{EKF}_{\text{nc}}} \times 100\%$, where *nc* means *noncompensated*.

State	MSE	Unit	$\frac{\text{MSE}_{\text{UKF}_{\text{nc}}}}{\text{MSE}_{\text{EKF}_{\text{nc}}}} \times 100\%$
x	8.6629×10^{-3}	m^2	100.10%
y	8.6314×10^{-3}	m^2	100.03%
ψ	2.5920×10^{-4}	rad^2	100.43%
v	1.1621×10^{-3}	$(\text{m/s})^2$	101.33%

When the delay is noncompensated, it is possible to notice that the histograms of the state errors – the Figures 64 and 66 – spread over the distance or lose their Gaussian distribution characteristic and the measurement noise variances varying to handle the errors caused by the delay, as shown in Figures 65 and 67. The results are expected since the error is calculated using the measurement data without delays against the estimations when the filters use the measurements from 5 backward time-steps. The performance of both of the filters are very similar for this scenario.

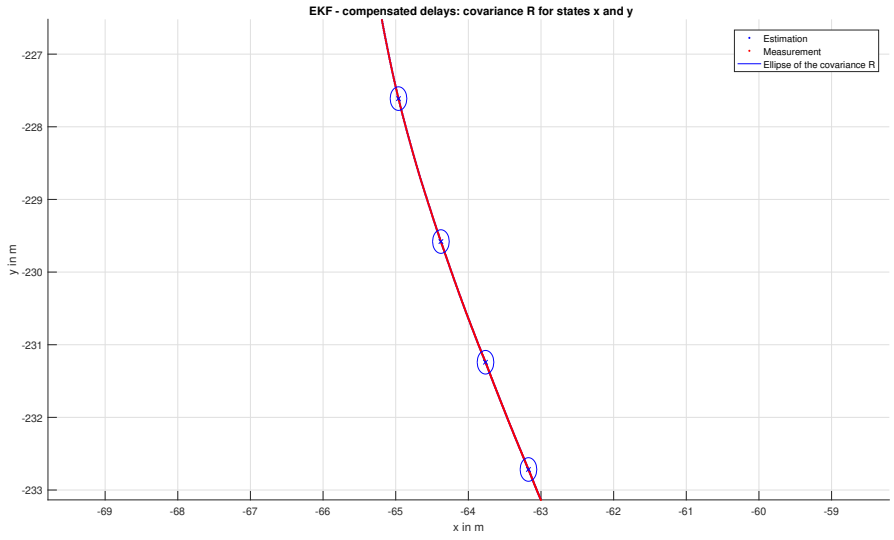
The other possibility is to compensate the delay by recalculating the prediction and update steps over the delay, as already mentioned. The next results are presented for this scenario using the EKF:

Figure 68 – *Estimation results using the EKF for compensated real delayed measurements:* this figure represents, on the left side, the estimated states that are corrected by the GNSS and its histograms of error on the right side.



Source: Personal collection.

Figure 69 – *Region of the trajectory of the test vehicle showing the ellipses of the covariance estimated using the EKF for compensated delays.*



Source: Personal collection.

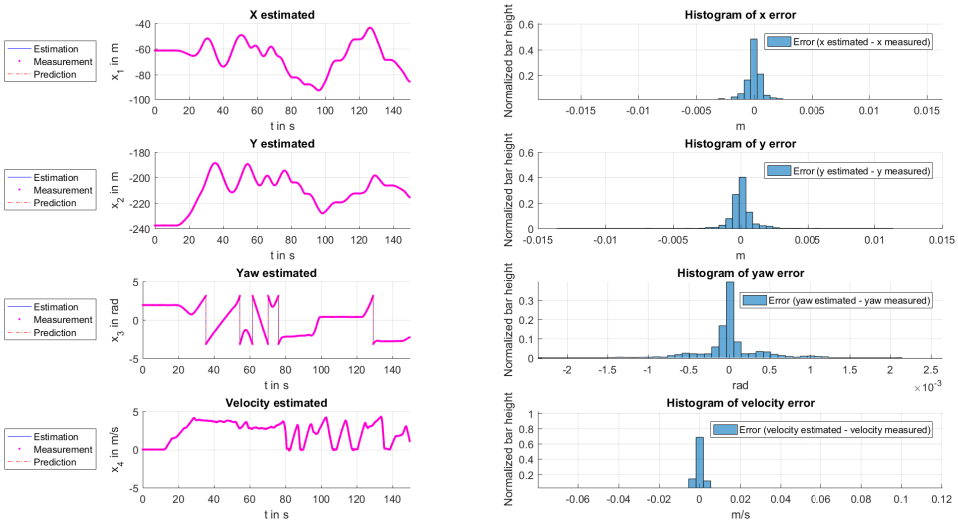
The MSE results for this scenario using the EKF are presented:

Table 29 – *MSE of the estimated states using the EKF and GNSS data for compensated delay:* in this case, the GNSS is used as a reference to calculate the MSE.

State	MSE	Unit
x	5.4617×10^{-7}	m^2
y	5.2573×10^{-7}	m^2
ψ	8.4157×10^{-8}	rad^2
v	1.1700×10^{-5}	$(\mathrm{m/s})^2$

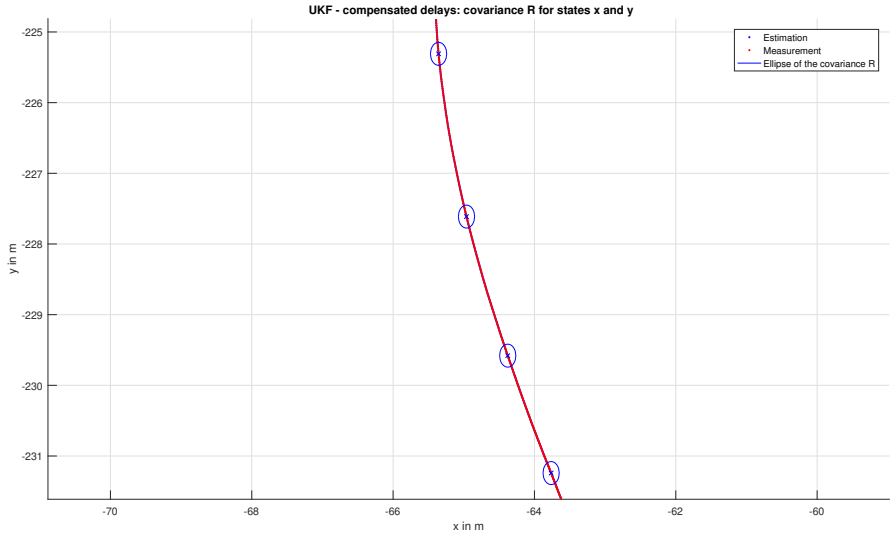
The same procedure is done to get the UKF results for the compensated delay.

Figure 70 – *Estimation results using the UKF for compensated real delayed measurements:* this figure represents, on the left side, the estimated states that are corrected by the GNSS and its histograms of error on the right side.



Source: Personal collection.

Figure 71 – *Region of the trajectory of the test vehicle showing the ellipses of the covariance estimated using the UKF for compensated delays.*



Source: Personal collection

Next, the MSE values of the UKF are presented and compared with the MSE of the EKF for the same scenario.

Table 30 – *MSE of the estimated states using the UKF and GNSS data and compensated delay:* in this case, the GNSS is used as a reference to calculate the MSE. The last column is the percentual comparison of the MSE values of the UKF against the MSE values of the EKF for the same scenario, presented in Table 29, $\text{MSE}_{\text{UKF}_c}/\text{MSE}_{\text{EKF}_c} \times 100\%$, where c means *compensated*.

State	MSE	Unit	$\frac{\text{MSE}_{\text{UKF}_c}}{\text{MSE}_{\text{EKF}_c}} \times 100\%$
x	6.0736×10^{-7}	m^2	111.21%
y	5.8641×10^{-7}	m^2	111.57%
ψ	1.3993×10^{-7}	rad^2	166.27%
v	1.1698×10^{-5}	$(\text{m/s})^2$	99.98%

The values of Tables 29 and 30 are expected, because as the GNSS measurements are updated at every 10 ms, the MSE values for the compensation should be small since the estimations basically follows the GNSS measurements.

It is possible to see through the Figures 68 and 70 that the histograms of the states errors were restored to their approximate Gaussian characteristic. In this situation, the EKF performed better for some states but the results are still quite similar for both filters.

Finally, similar to Table 31, the MSE of the measured states are compared for the cases that the delays are wrongly compensated.

Table 31 – *MSE results for the EKF and UKF estimations when a smaller or larger delay is considered:* for both filters the exact delay time was 5 time-steps and their MSE values are presented, represented in the table with the overscript "*". To simulate practical situations, a smaller and a larger delay is considered to see how the MSE values degrades when the compensation is done with a value different from 5 time-steps.

Filter compensation	State	3	5*	7	Unit
EKF	x	1.3945×10^{-3}	5.654×10^{-7}	1.4116×10^{-3}	m^2
	y	1.3857×10^{-3}	5.4282×10^{-7}	1.3802×10^{-3}	m^2
	ψ	4.2019×10^{-5}	9.8159×10^{-8}	3.8738×10^{-5}	rad^2
	v	2.5708×10^{-4}	1.1700×10^{-5}	1.6196×10^{-4}	$(m/s)^2$
UKF	x	1.3941×10^{-3}	6.0578×10^{-7}	1.4120×10^{-3}	m^2
	y	1.3859×10^{-3}	5.8505×10^{-7}	1.3802×10^{-3}	m^2
	ψ	4.2437×10^{-5}	1.1413×10^{-7}	3.8567×10^{-5}	rad^2
	v	2.5711×10^{-4}	1.1698×10^{-5}	1.6193×10^{-4}	$(m/s)^2$

This results are also expected. It shows that even for wrong compensation the MSE values are better than when the delay is ignored for both filters.

6.3 FILTER COMPARISONS × COMPUTATIONAL COSTS

Although the UKF has a slightly better performance than the EKF, for some theoretical scenarios seen in this thesis, the difference between these filters are minimal since the sensors equations are completely linear. On the other hand, the computational cost of the UKF is higher, mainly for the case where the delays are compensated, as the filter needs to recalculate all sigma-points along the delayed time-steps, which can be decisive on embedded microcontrollers.

The biggest computational cost of the UKF, shown in Table 6, is during the prediction step. If the computational time is not a critical problem, the SSSRUKF can be considered the most suitable filter to be performed in all the time-steps. In the case that the computational time is a critical problem, an alternative way can be formulated to use both filters: a hybrid algorithm to use the SRSSUKF only when the GNSS measurements are not available, and the UDEKF when the measurements are available, because it is less accurate than the UKF for certain cases.

Theoretically, the benefits of the UKF increase compared to the EKF according to the nonlinearities of the system. As nonlinear the model is, the greater the benefits of using the UKF instead of the EKF. For linear systems, the performances of the two filters are the same. For this field of application, the nonlinearities are found in the vehicle's model used in the prediction step, while the sensor model, used in the update step, is

linear. This means that the UKF can have a benefit when compared to the EKF only in the prediction step. In the update step, they perform identically.

That is the idea behind the hybrid algorithm proposed here. When GNSS measurements are not available, only prediction is done, so the UKF is used to output the best possible prediction. When the GNSS measurements are available, in addition to the prediction step, the correction step is done, so an EKF can be used as the correction will improve the filter estimates and this will not only be a role of the predictions. With this, there will be a trade-off between computational cost and estimation accuracy.

6.4 1994aTATIONS OF THIS WORK

The only two limitations identified were in terms of the data collected from the IMU sensor and odometry to be used for the filters, and the characteristics of the ZF Tugger, in order to know its parameters to improve the parameter estimation. An improvement in these two problems cited above can assuredly provide an improvement in the performance of the filters.

7 CONCLUSION

This chapter makes a retrospective synopsis about the core idea of this study and concisely emphasizes the results.

7.1 GENERAL OVERVIEW

This work proposed a study to compare two stochastic filters and determinate the most suitable for the autonomous driving field of application to account with delayed measurements. The first chapter have presented the main idea of the project, a brief concept about the stochastic filters, the state-of-the-art, related works in the field and the thesis outline.

The second chapter have shown the theory behind an estimator, given an theoretical explanation about how the KF behaves and how their variants were formulated. It was shown in the literature that better numerical stability, accuracy and efficiency for the filters results can be reached when the covariance matrices are in their square root decomposition forms, using the Cholesky decomposition or the modified Cholesky decomposition. Theoretically, the best variants of the stochastic filters were compared: the EKF was performed using the Upper-Diagonal Factorization technique, and the UKF was performed using its square root formulation and the spherical-simplex format to obtain the sigma-points in the UT, in order to decrease the computational costs of the filter.

To perform this comparison, some methods and materials were used which were described in this thesis. The third chapter shown: the vehicle models and the sensors used to perform the comparisons, as well as the comparison methods and consistency tests as a tool to certificate with the filters implemented were performing correctly.

The result of the comparisons, in Chapter 6, have shown that for the application the two filters are very similar in terms of accuracy and a slight better performance (shown in the simulated values) of the UKF variant compared to the EKF variant for the states that are not measured. It was suggested through these results that if the computational burden is not critical, the SRSSUKF should be used. If it is critical, a hybrid algorithm should be implemented to keep the quality of the estimation mainly for the states that are not directly measured by the sensors. This hybrid algorithm should use the SRSSUKF for the time instants when the GNSS data is not available. For the time instants that the filter accounts with the measurement corrections, the UDEKF can be used.

The autocorrelation and the DW test shown a great role for the detection of problems in the model. Although it is not a tool that affects on some filter's result directly, it was used as an indicator tool in this study.

The incorporation of delays is another important point: it is worth emphasizing

the improvement of results when the delays are compensated, even with erroneously compensations, than when they are noncompensated at all.

The ST model with errors in the inputs and with a few known parameters are the two limitations of the real data results showing that in real problems the number of problems to be dealt with is much higher. This limitations degrade the consistency test results, meaning that the filters did not operate in an optimal way. Despite of that, the simulated values presented satisfactory results, where were possible to point out some differences of the filters.

During the development of the work it was possible to detect points of improvement for future work in the area, as seen in the following section.

7.2 FUTURE WORK IN THE FIELD

At least three possible work extensions can be pointed out from this study:

- Online estimation of the covariance matrices of the process and measurement noises. Recent literature presented the Adaptive Kalman Filters (AKF) which estimate the $\mathbf{Q}$ and $\mathbf{R}$ matrices at each time-step. An extension of this work using the AKFs would be interesting to try to deal with the model errors.

- Optimal determination of the parameters α, β and κ for the UKF filters and its variants. Despite of certain guidelines present in literature about how this parameters could be chosen, the values of these scaling parameters could be made to be adapted online depending on factors such as the severity of nonlinearities, variances of noise sources, etc.

- Estimation of the delay of the measurement data is another difficult point but can generates significant advantages since the estimations are more accurate with the correct compensation of the delays.

7.3 FINAL CONSIDERATIONS

The research solved the proposed problem, met the initial objectives and broadened the understanding of the problem in order to demonstrate real world problems that filters eventually need to deal with.

The bibliographical references were extremely useful for all the development of the project and it is expected that the proposed topic and the way it was presented contribute not only to industrial objectives, but also to be a didactic complement reference for university students who are interested in the field.

References

ALEXANDER, H. L. State estimation for distributed systems with sensing delay. In: LIBBY, V. (Ed.). **Data Structures and Target Classification**. [S.l.]: SPIE, 1991. v. 1470, p. 103 – 111. https://doi.org/10.1117/12.44843. Cited in page 26.

Allotta, B. et al. A comparison between ekf-based and ukf-based navigation algorithms for auvs localization. In: **OCEANS 2015 - Genova**. [S.l.: s.n.], 2015. p. 1–5. Cited in page 25.

ALPAYDIN, E. **Introduction to Machine Learning**. [S.l.]: MIT Press, 2020. (Adaptive Computation and Machine Learning series). https://books.google.de/books?id=tZnSDwAAQBAJ. ISBN 9780262043793. Cited in page 25.

ALTHOFF, M.; KOSCHI, M.; MANZINGER, S. Commonroad: Composable benchmarks for motion planning on roads. In: **Proc. of the IEEE Intelligent Vehicles Symposium**. [S.l.: s.n.], 2017. ISBN 9781509048045. Cited in page 44.

Andersen; Christensen; Ravn. Augmented models for improving vision control of a mobile robot. In: **1994 Proceedings of IEEE International Conference on Control and Applications**. [S.l.: s.n.], 1994. p. 53–58 vol.1. Cited in page 25.

BECK, J.; ARNOLD, K. **Parameter Estimation in Engineering and Science**. [S.l.]: Wiley, 1977. (Probability and Statistics Series). https://books.google.de/books?id=_qAYgYN87UQC. ISBN 9780471061182. Cited in page 25.

BELLANTONI, J. F.; DODGE, K. W. A square root formulation of the Kalman- Schmidt filter. In: NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. [S.l.]: AIAA JOURNAL, 1967. Cited 2 times in pages 25 and 41.

Bierman, G. J. Measurement updating using the u-d factorization. In: **1975 IEEE Conference on Decision and Control including the 14th Symposium on Adaptive Processes**. [S.l.: s.n.], 1975. p. 337–346. Cited in page 25.

BOX, G.; TIAO, G. **Bayesian Inference in Statistical Analysis**. [S.l.]: Wiley, 2011. (Wiley Classics Library). https://books.google.de/books?id=T8Askeyk1k4C. ISBN 9781118031445. Cited in page 24.

BRERETON, R. G. The normal distribution. **Journal of Chemometrics**, v. 28, n. 11, p. 789–792, 2014. https://onlinelibrary.wiley.com/doi/abs/10.1002/cem.2655. Cited in page 28.

BRYC, W. **The Normal Distribution: Characterizations with Applications**. [S.l.]: Springer New York, 2012. (Lecture Notes in Statistics). https://books.google.de/books?id=tyXjBwAAQBAJ. ISBN 9781461225607. Cited in page 24.

CAMBANIS, S. Conditional probability and expectation. In: _____. **Wiley StatsRef: Statistics Reference Online**. [S.l.]: American Cancer Society, 2014. ISBN 9781118445112. https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118445112.stat02846. Cited in page 28.

CANUTO, C.; OGATA, A. **Mathematical Analysis I**. [S.l.]: Springer, Milano, 2008. ISBN 978-88-470-0876-2. Cited in page 19.

Castrejon Lozano, J. G. et al. Spherical simplex sigma-point kalman filters: A comparison in the inertial navigation of a terrestrial vehicle. In: **2008 American Control Conference**. [S.l.: s.n.], 2008. p. 3536–3541. Cited in page 39.

COLEMAN, R. What is a stochastic process? In: _____. **Stochastic Processes**. Dordrecht: Springer Netherlands, 1974. p. 1–5. ISBN 978-94-010-9796-3. https://doi.org/10.1007/978-94-010-9796-3_1. Cited in page 28.

DYNKIN, E. B. Wanderings of a markov process. **Theory of Probability & Its Applications**, v. 16, n. 3, p. 401–428, 1971. https://doi.org/10.1137/1116046. Cited in page 29.

Fiorenzani, T. et al. **Comparative Study of Unscented Kalman Filter and Extended Kalman Filter for Position/Attitude Estimation in Unmanned Aerial Vehicles**. Tese (Doutorado) — Istituto di Analisi Dei Sistemi ed Informatica "Antonio Ruberti", 2008. Cited in page 25.

Gonçalves, J. P. S.; Zattoni, P. S. **Estudo e Aplicação de Filtragem Estocástica Utilizando o Filtro de Kalman.** Tese (Doutorado) — Federal University of Technology - Paraná, 2017. Cited in page 28.

GREWAL, M.; ANDREWS, A. Kalman filtering: theory and practice using matlab. **New York: John Wiley and Sons**, v. 14, 01 2001. Cited 2 times in pages 41 and 42.

GREWAL, M. S. Kalman filtering. In: _____. **International Encyclopedia of Statistical Science**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 705–708. ISBN 978-3-642-04898-2. https://doi.org/10.1007/978-3-642-04898-2_321. Cited 2 times in pages 29 and 119.

Gustafsson, F. et al. Particle filters for positioning, navigation, and tracking. **IEEE Transactions on Signal Processing**, v. 50, n. 2, p. 425–437, 2002. Cited in page 26.

Hamzah, N. H. et al. Analysis of the residual between the model and the data using autocorrelation function for satellite attitude estimation. In: **2013 IEEE 9th International Colloquium on Signal Processing and its Applications**. [S.l.: s.n.], 2013. p. 78–82. Cited 2 times in pages 58 and 60.

Hanlon, P. D.; Maybeck, P. S. Characterization of kalman filter residuals in the presence of mismodeling. **IEEE Transactions on Aerospace and Electronic Systems**, v. 36, n. 1, p. 114–131, 2000. Cited in page 60.

He, X. et al. Distributed kalman filters with state equality constraints: Time-based and event-triggered communications. **IEEE Transactions on Automatic Control**, v. 65, n. 1, p. 28–43, 2020. Cited in page 24.

JOYCE, J. Bayes' Theorem. In: ZALTA, E. N. (Ed.). **The Stanford Encyclopedia of Philosophy**. Spring 2019. [S.l.]: Metaphysics Research Lab, Stanford University, 2019. https://plato.stanford.edu/archives/spr2019/entries/bayes-theorem/. Cited in page 28.

Julier, S. J. The scaled unscented transformation. In: **Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)**. [S.l.: s.n.], 2002. v. 6, p. 4555–4559 vol.6. Cited 2 times in pages 37 and 38.

JULIER, S. J.; UHLMANN, J. K. Unscented Filtering and Nonlinear Estimation. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. [S.l.]: IEEE, 2004. v. 92. Cited 2 times in pages 19 and 35.

KALMAN, R. E. A new approach to linear filtering and prediction problems. **Transactions of the ASME–Journal of Basic Engineering**, v. 82, n. Series D, p. 35–45, 1960. Cited 2 times in pages 19 and 24.

KRÄMER, W. Durbin–watson test. In: _____. **International Encyclopedia of Statistical Science**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 408–409. ISBN 978-3-642-04898-2. https://doi.org/10.1007/978-3-642-04898-2_219. Cited in page 60.

Larsen, T. D. et al. Incorporation of time delayed measurements in a discrete-time kalman filter. In: **Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)**. [S.l.: s.n.], 1998. v. 4, p. 3972–3977 vol.4. Cited in page 26.

Liang, X. et al. Leader-following formation tracking control of mobile robots without direct position measurements. **IEEE Transactions on Automatic Control**, v. 61, n. 12, p. 4131–4137, 2016. Cited in page 24.

LYCHE, T.; MERRIEN, J.-L. Matrices and linear systems. In: _____. **Exercises in Computational Mathematics with MATLAB**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 9–24. ISBN 978-3-662-43511-3. https://doi.org/10.1007/978-3-662-43511-3_2. Cited in page 37.

MAYBECK, P. S. The kalman filter: An introduction to concepts. In: _____. **Autonomous Robot Vehicles**. New York, NY: Springer New York, 1990. p. 194–204. ISBN 978-1-4613-8997-2. https://doi.org/10.1007/978-1-4613-8997-2_15. Cited in page 29.

MCGEE, L. et al. **Flight Results from a Study of Aided Inertial Navigation Applied to Landing Operations**. [S.l.]: National Aeronautics and Space Administration, 1973. (NASA TN D-7302). https://books.google.de/books?id=a8A_tXjKdlgC. Cited in page 25.

MERWE, R. van der. **Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models**. Tese (Doutorado) — Oregon Health and Science University, 2004. Cited 9 times in pages 24, 25, 26, 28, 31, 34, 37, 39, and 124.

Mishra, A. K.; Shimjith, S. R.; Tiwari, A. P. Adaptive unscented kalman filtering for reactivity estimation in nuclear power plants. **IEEE Transactions on Nuclear Science**, v. 66, n. 12, p. 2388–2397, 2019. Cited in page 24.

Mochnac, J.; Marchevsky, S.; Kocan, P. Bayesian filtering techniques: Kalman and extended kalman filter basics. In: . [S.l.: s.n.], 2009. p. 119–122. Cited in page 19.

NELSON, A. T. **Nonlinear estimation and modeling of noisy timeseries by dual Kalman filtering methods**. Tese (Doutorado) — Oregon Health and Science University, 2000. Cited in page 25.

Ni, Y. et al. On the performance analysis of reset attack in cyber-physical systems. **IEEE Transactions on Automatic Control**, v. 65, n. 1, p. 419–425, 2020. Cited in page 24.

OGATA, K. **Modern Control Engineering**. [S.l.]: Prentice Hall, 2010. ISBN 9780136156734. Cited in page 19.

PAPOULIS, A.; PILLAI, U. **Probability, random variables and stochastic processes**. 4th. ed. [S.l.]: McGraw-Hill, 2001. Cited in page 31.

QUARTERONI, A.; SACCO, R.; SALERI, F. Nonlinear systems and numerical optimization. In: _____. **Numerical Mathematics**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 285–331. ISBN 978-3-540-49809-4. https://doi.org/10.1007/978-3-540-49809-4_7. Cited in page 33.

Mean squared error. In: SAMMUT, C.; WEBB, G. I. (Ed.). **Encyclopedia of Machine Learning**. Boston, MA: Springer US, 2010. p. 653–653. ISBN 978-0-387-30164-8. https://doi.org/10.1007/978-0-387-30164-8_528. Cited in page 60.

TALAMUCCI, F. **Nonlinear nonholonomic constraints**. 2019. Cited in page 44.

TANIZAKI, H. Nonlinear filters based on taylor series expansion. In: _____. **Nonlinear Filters: Estimation and Applications**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993. p. 35–67. ISBN 978-3-662-22237-9. https://doi.org/10.1007/978-3-662-22237-9_3. Cited 2 times in pages 33 and 34.

TAYLOR; FRANCIS. **Linear Least Squares Computations**. [S.l.: s.n.], 1988. (Statistics: A Series of Textbooks and Monographs). https://books.google.de/books?id=aCS0zw7SztEC. ISBN 9780824776619. Cited in page 24.

Tulsyan, A.; Gopaluni, R. B.; Khare, S. R. Particle filtering without tears: A primer for beginners. In: LIBBY, V. (Ed.). **Computers and Chemical Engineering**. [S.l.]: Elsevier, 2016. v. 95, p. 130–145. Cited in page 26.

VALADE, A. et al. A study about kalman filters applied to embedded sensors. **Sensors**, v. 17, p. 2810, 12 2017. Cited 2 times in pages 61 and 119.

WASLANDER, S.; KELLY, J. **Self-Driving Cars Specialization**. 2020. https://www.coursera.org/learn/state-estimation-localization-self-driving-cars. Cited 2 times in pages 45 and 50.

Xu, T. et al. A maneuvered geo satellite orbit determination using robustly adaptive kalman filter. In: **2010 International Conference on Intelligent System Design and Engineering Application**. [S.l.: s.n.], 2010. v. 1, p. 55–59. Cited in page 24.

ZHANG, Y.; OLIVER, D. Evaluation and error analysis: Kalman gain regularization versus covariance regularization. **Computational Geosciences**, v. 15, p. 489–508, 06 2011. Cited in page 31.

Appendix

APPENDIX A – UPPER-DIAGONAL EXTENDED KALMAN FILTER

As mentioned in Chapter 3, Thornton-Bierman UD method is a pair of algorithms for prediction and measurement updates more robust against roundoff errors. In this Appendix, the Thornton and the Bierman algorithms are presented as well as the UD decomposition algorithm to calculate the $\mathbf{U}$ and $\mathbf{D}$ matrices from the initial covariances. The Bierman's algorithm, developed by Gerald J. Bierman is one of the more stable implementations of the measurement update step of the KF in general. Its algorithm is presented below: The computational complexity of Bierman's algorithm in terms of big $\mathcal{O}$ notation is $\mathcal{O}(2n^2)$.

Catherine Thornton's modified weighted Gram-Schmidt orthogonalization method for the predictor update of the UD factors of the covariance matrix, known as Thornton temporal update algorithm, is also presented below. More details about the modified weighted Gram-Schmidt method are found in (GREWAL, 2011), (VALADE et al., 2017), and in references therein.

The computational complexity of Catherine Thornton's algorithm in terms of big $\mathcal{O}$ notation is $\mathcal{O}(2n^3)$.

With this two algorithms presented it is possible to finally describe the EKF applying the Thornton-Bierman algorithm. The Thornton-Bierman EKF algorithm differs a little from the standard implementation but the equations of the prediction and measurement steps keep the same main idea, since it is only a variation of the standard algorithm. The algorithm is described below, considering the noise covariances $\mathbf{Q}$ and $\mathbf{R}$ constants:

An algorithm (called *udu* in the initialization of the EKF) is necessary to decompose the $\mathbf{Q}$ matrix in $\mathbf{U_Q}$ and $\mathbf{D_Q}$ as well as decompose the initial value of the covariance matrix $\mathbf{P_{\hat{x}_0}}$ in $\mathbf{U_{P_0}}$ and $\mathbf{D_{P_0}}$. The algorithm is also described in Algorithm 1:

Algorithm 1: Bierman observational update

Input:

z	scalar measurement
R	variance of the measurement error
H	measurement sensitivity ($1 \times n$ matrix)
x	the prior estimate of the state vector
U	upper triangular factor of the prior covariance matrix
D	diagonal factor of the prior covariance matrix

Output:

x	the posterior estimate of the state vector
U	upper triangular factor of the posterior covariance matrix
D	diagonal factor of the posterior covariance matrix

$delta = z$
for j = *1:n* **do**
  $delta = delta - H(j)x(j)$ $v(j) = H(j)$
  for *i=1:j-1* **do**
    $v(j) = v(j) + U(i,j)H(i)$
  end
end
$sigma = R$
for j = *1:n* **do**
  $aux = v(j)$
  $v(j) = v(j)D(j,j)$
  $w(j) = aux$
  for *i=1:j-1* **do**
    $tau = U(i,j)aux$
    $U(i,j) = U(i,j) - w(i)aux/sigma$
    $w(i) = w(i) + tau$
  end
  $D(j,j) = D(j,j)sigma$
end
$epsilon = delta/sigma$
for *i=1:n* **do**
  $x(i) = x(i) + v(i)epsilon$
end

Algorithm 2: Thornton temporal update

Input:

Phi state transition matrix (Jacobian result, in this case)
$\mathbf{U_{in}}$ upper triangular factor of the corrected covariance matrix
$\mathbf{D_{in}}$ diagonal factor of the corrected covariance matrix
$\mathbf{U_Q}$ upper triangular factor of the covariance matrix of process noise
$\mathbf{D_Q}$ diagonal diagonal factor of the covariance matrix of process noise

Output:

U upper triangular factor of the predicted covariance matrix
D diagonal factor of the predicted covariance matrix

$PhU = PhiU_{in}$
for *i=n-1:1* **do**
  $sigma = 0$
  for *j=1:n* **do**
    $sigma = sigma + PhU(i,j)^2 D_{in}(j,j)$
    if $j <= n$ **then**
      $sigma = sigma + U_Q(i,j)^2 D_Q(j,j)$
    end
  end
  $D(i,i) = sigma$
  for *j=1:i-1* **do**
    $sigma = 0$
    for *k=1:n* **do**
      $sigma = sigma + PhU(i,k) D_{in}(k,k) PhU(j,k)$
    end
    for *k=1:r* **do**
      $sigma = sigma + U_Q(i,k) D_Q(k,k) U_Q(j,k);$
    end
    $U(j,i) = sigma / D(i,i)$
    for *k=1:n* **do**
      $PhU(j,k) = PhU(j,k) - U(j,i) PhU(i,k)$
    end
    for *k=1:r* **do**
      $U_Q(j,k) = U_Q(j,k) - U(j,i) U_Q(i,k)$
    end
  end
end

Algorithm 3: Upper-Diagonal Extended Kalman Filter

Initialization:

$$\begin{aligned}
\hat{\mathbf{x}}_0 &= \mathbb{E}[\mathbf{x_0}] \\
\mathbf{P}_{\hat{\mathbf{x}}_\mathbf{0}} &= \mathbb{E}[(\mathbf{x_0} - \hat{\mathbf{x}}_\mathbf{0})(\mathbf{x_0} - \hat{\mathbf{x}}_\mathbf{0})^\intercal] \\
\mathbf{Q} &= \mathbb{E}[(\mathbf{w} - \bar{\mathbf{w}})(\mathbf{w} - \bar{\mathbf{w}})^\intercal] \\
\mathbf{R} &= \mathbb{E}[(\mathbf{v} - \bar{\mathbf{v}})(\mathbf{v} - \bar{\mathbf{v}})^\intercal] \\
[\mathbf{U}_{\mathbf{P_0}}, \mathbf{D}_{\mathbf{P_0}}] &= udu(\mathbf{P}_{\hat{\mathbf{x}}_\mathbf{0}}) \\
[\mathbf{U}_\mathbf{Q}, \mathbf{D}_\mathbf{Q}] &= udu(\mathbf{Q})
\end{aligned}$$

for *k = 1,2,..., N* **do**

Prediction:

Compute the Jacobian Matrix of the process model:

$$\mathbf{F}_{\mathbf{x_k}} = \nabla_\mathbf{x}\mathbf{f}(x, u_k, 0)|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}}$$

Compute the prediction mean and covariance of the state:

$$\begin{aligned}
\hat{\mathbf{x}}_k^- &= \mathbf{f}(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_k, \mathbf{w}_k) \\
[\mathbf{U}_{\mathbf{x}_k}^-, \mathbf{D}_{\mathbf{x}_k}^-] &= thornton(\hat{\mathbf{x}}_k^-, \mathbf{F}_{\mathbf{x_k}}, \mathbf{U}_{\mathbf{x}_{k-1}}^+, \mathbf{D}_{\mathbf{x}_{k-1}}^+, \mathbf{U}_\mathbf{Q}, \mathbf{U}_\mathbf{Q})
\end{aligned}$$

Measurement update:

Compute the Jacobian Matrix of the measurement model:

$$\mathbf{H}_{\mathbf{x_k}} = \nabla_\mathbf{x}\mathbf{h}(\mathbf{x}, 0)|_{\mathbf{x}=\hat{\mathbf{x}}_k^-}$$

Correct the prediction state and covariance with the latest measurement $\mathbf{z}_k \in \mathbb{R}^p$:

for *j=1:p* **do**

$$\begin{aligned}
[\hat{\mathbf{x}}_k^+, \mathbf{U}_{\mathbf{x}_k}^+, \mathbf{D}_{\mathbf{x}_k}^+] &= bierman(\mathbf{z}(j), \mathbf{R}(j,j), \mathbf{H}_k(j,:), \hat{x}_k^-, \mathbf{U}_{\mathbf{x}_k}^-, \mathbf{D}_{\mathbf{x}_k}^-) \\
\mathbf{U}_{\mathbf{x}_k}^- &= \mathbf{U}_{\mathbf{x}_k}^+ \\
\mathbf{D}_{\mathbf{x}_k}^- &= \mathbf{D}_{\mathbf{x}_k}^+
\end{aligned}$$

end

end

Algorithm 4: UDU decomposition

Input:
P square matrix of order m
Output:
U square matrix of the same dimensions as P, with zeros below its main diagonal and ones on its main diagonal.
D a square matrix of the same dimensions as P, with zeros above and below its main diagonal

$P = (P + P^\intercal)/2$
for j = *m:-1:1* **do**
  for i = *j:-1:1* **do**
    sigma = P(i,j)
    for k = *j+1:m* **do**
      $sigma = sigma - U(i,k)D(k,k)U(j,k)$
    end
    if $i == j$ **then**
      $D(j,j) = sigma$
      $U(j,j) = 1$
    else
      $U(i,j) = sigma/D(j,j)$
    end
  end
end

The computational costs in terms of big $\mathcal{O}$ notation of this algorithm is $\mathcal{O}(\frac{m^3}{6})$, where m is the order of the input positive-definite matrix **P**.

APPENDIX B – SQUARE-ROOT UNSCENTED KALMAN FILTER

As mentioned in Chapter 3, the Square-Root approach of the UKF propagates and updates the Cholesky factor $\mathbf{S}$ directly instead of the covariance matrix $\mathbf{P}$. The prediction and measurement is presented in Algorithm 5.

When the filter starts to run, it is necessary to calculate the matrix Square-Root $\mathbf{S}$ of the state covariance using the Cholesky decomposition. After this, the Cholesky factor is used instead of the covariance matrix itself, as already mentioned in Chapter 3.

Two linear algebra techniques are used in this algorithm: the QR decomposition and the Cholesky factor updating (or downdating). In the prediction step, the Cholesky factor, $\mathbf{S_x}$, is calculated using the QR decomposition of the compound matrix containing the weighted propagated sigma-points and the matrix square-root of the additive process noise covariance. The subsequent Cholesky update (or downdate) is necessary because in this application the zeroth weight is equal to κ and since $\kappa = 3 - n$ can generate negative weights, $w_0^{(c)}$.

The same procedure occurs again in the measurement update step for the calculation of $\mathbf{S}_{\tilde{\mathbf{y}}}$, the residual covariance. In contrast to the way the Kalman gain is calculated in the standard UKF, now is used least squares solutions to the following expansion:

$$\mathbf{K}_k(\mathbf{S}_{\tilde{\mathbf{y}}_k}\mathbf{S}_{\tilde{\mathbf{y}}_k}^{\intercal}) = \mathbf{P}_{\mathbf{x}_k\mathbf{y}_k}. \tag{51}$$

Since $\mathbf{S}_{\tilde{\mathbf{y}}}$ form is square and triangular, efficient "back-substitutions" can be used to solve for $\mathbf{K}_k$ directly without the need for a matrix inversion. Finally, the posterior measurement update of the Cholesky factor of the state covariance, $\mathbf{S}_{\mathbf{x_k}}$ is calculated. Sequential Cholesky downdates are done to $\mathbf{S}_{\mathbf{x}_k}$. The downdate vectors are the columns of $\mathbf{U} = \mathbf{K}_k\mathbf{S}_{\tilde{\mathbf{y}}_k}$. This replaces the posterior update of $\mathbf{P}_{\mathbf{x}_k}$. To make it clear, the Algorithm 5 shows the complete set of equations of the SRUKF.

The QR decomposition and the Cholesky update have both the computational complexity of $\mathcal{O}(n^3/6)$, where is the size of the state vector. The whole algorithm is also around $\mathcal{O}(n^3)$ like the other implementations, but with an improved numerical propertied: it guarantees positive semi-definiteness of the state-covariances (MERWE, 2004).

Algorithm 5: Square-Root Unscented Kalman Filter

Initialization:

$$\begin{aligned}
\hat{\mathbf{x}}_0 &= \mathbb{E}[\mathbf{x_0}] \\
\mathbf{S}_{\mathbf{x_0}} &= chol\{\mathbb{E}[(\mathbf{x_0} - \hat{\mathbf{x}}_\mathbf{0})(\mathbf{x_0} - \hat{\mathbf{x}}_\mathbf{0})^T]\} \\
\mathbf{Q} &= chol\{\mathbb{E}[(\mathbf{w} - \bar{\mathbf{w}})(\mathbf{w} - \bar{\mathbf{w}})^T]\} \\
\mathbf{R} &= chol\{\mathbb{E}[(\mathbf{v} - \bar{\mathbf{v}})(\mathbf{v} - \bar{\mathbf{v}})^T]\} \\
[w^{(m)}, w^{(c)}] &= computeWeights(\kappa, \alpha, \beta)
\end{aligned}$$

for *k = 1,2,..., N* **do**

Calculate sigma points according to the equations presented in Chapter 3:

$$\mathcal{X}_{k-1} = computeSigmaPoints(\hat{\mathbf{x}}^+_{k-1}, \mathbf{S}^+_{k-1}, \kappa, \alpha, \beta)$$

Prediction:

$$\begin{aligned}
\mathcal{X}_k &= \mathbf{f}(\mathcal{X}_{k-1}, \mathbf{u}_k, 0) \\
\hat{\mathbf{x}}^-_k &= \sum_{i=0}^{2n} w_i^{(m)} \mathcal{X}_{i,k} \\
\mathbf{S}^-_{\mathbf{x}_k} &= qr\left\{\left[\sqrt{w_i^{(c)}}(\mathcal{X}_{1:2n,k} - \mathbf{x}^-_k) \quad Q\right]\right\} \\
\mathbf{S}^-_{\mathbf{x}_k} &= cholUpdate\{\mathbf{S}^-_{\mathbf{x}_k}, \mathcal{X}_{0,k} - \mathbf{x}^-_k, w_0^{(c)}\}
\end{aligned}$$

Measurement update:

$$\begin{aligned}
\mathcal{Y}_k &= \mathbf{h}(\mathcal{X}_k, 0) \\
\hat{\mathbf{y}}^-_k &= \sum_{i=0}^{2n} w_i^{(m)} \mathcal{Y}_{i,k} \\
\mathbf{S}_{\tilde{\mathbf{y}}_\mathbf{k}} &= qr\left\{\left[\sqrt{w_i^{(c)}}((\mathcal{Y}_{1:2n,k} - \mathbf{y}^-_k) \quad R\right]\right\} \\
\mathbf{S}_{\tilde{\mathbf{y}}_\mathbf{k}} &= cholUpdate\{\mathbf{S}_{\tilde{\mathbf{y}}_\mathbf{k}}, \mathcal{Y}_{0,k} - \mathbf{y}^-_k, w_0^{(c)}\} \\
\mathbf{P}_{\mathbf{x}_k\mathbf{y}_k} &= \sum_{i=0}^{2n} w_i^{(c)}(\mathcal{X}_{i,k} - \hat{\mathbf{x}}^-_k)(\mathcal{Y}_{i,k} - \hat{\mathbf{y}}^-_k)^T \\
\mathbf{K}_k &= (P_{\mathbf{x}_k\mathbf{y}_k})/\mathbf{S}^T_{\tilde{\mathbf{y}}_\mathbf{k}})/\mathbf{S}_{\tilde{\mathbf{y}}_\mathbf{k}} \\
\hat{\mathbf{x}}^+_k &= \hat{\mathbf{x}}^-_k + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}^-_k) \\
U &= \mathbf{K}_k\mathbf{S}_{\tilde{\mathbf{y}}_\mathbf{k}} \\
\mathbf{S}_{\mathbf{x}_k} &= cholUpdate\{\mathbf{S}_{\mathbf{x}^-_k}, U, w_0^{(c)}\}
\end{aligned}$$

end

APPENDIX C – NUMERICAL METHODS FOR SOLVING THE ODEs

The EKF requires this equations in a discrete-time format to linearize the model though the Jacobian matrix. For the other side, the UKF does not require the discrete-time format of the equations explicitly, but need to solve the ODEs anyway using numerical methods. For that purpose, the Forward Euler and the 4^{th}-order Runge-Kutta methods are used, respectively.

C.1 FORWARD EULER METHOD AND DISCRETE-TIME VEHICLE MODEL

The Forward Euler method used for the EKF is a first-order numerical procedure for temporal discretization. Considering a first-order differential equation with the form:

$$x'(t) = f(t, x(t)). \tag{53}$$

This derivative function can be replaced by a finite difference approximation:

$$x'(t) \approx \frac{x(t+h) - x(t)}{h}, \tag{54}$$

where h is the discrete interval. Since this Euler method is a first-order approximation, the value h should be as small as possible to approximate accurately. The Equation (54) is rearranged in the following way:

$$x(t+h) \approx x(t) + hx'(t), \tag{55}$$

and using Equation (53) in (55):

$$x(t+h) \approx x(t) + hf(t, x(t)). \tag{56}$$

Motivated by this last equation, the discrete-time form denotes x_k a numerical estimate of the exact solution $x(k)$ in a recursive way to:

$$x_{k+1} = x_k + hf(k, x_k). \tag{57}$$

Making the same process with the set of ODEs that describes the vehicle model, the discrete-time model format is obtained. The discrete interval h is represented by t in the equations below.

Starting from the KST model, it is possible to rewrite the Equation (31) as follows:

$$x_{1,k+1} = x_{1,k} + \Delta t x_{4,k} \cos(x_{5,k}), \tag{58a}$$

$$x_{2,k+1} = x_{2,k} + \Delta t x_{4,k} \sin(x_{5,k}), \tag{58b}$$

$$x_{3,k+1} = x_{3,k} + \Delta t \frac{x_{4,k}}{l_{\text{wb}}} \tan(u_{1,k}), \tag{58c}$$

$$x_{4,k+1} = x_{4,k} + \Delta t u_{2,k}, \tag{58d}$$

$$\tag{58e}$$

Finally, for the ST model the Equation (59) is rewritten as follows;

$$x_{1,k+1} = x_{1,k} + \Delta t x_{4,k} \cos(x_{5,k} + x_{7,k}), \tag{59a}$$

$$x_{2,k+1} = x_{2,k} + \Delta t x_{4,k} \sin(x_{5,k} + x_{7,k}), \tag{59b}$$

$$x_{3,k+1} = x_{3,k} + \Delta t x_{5,k}, \tag{59c}$$

$$x_{4,k+1} = x_{4,k} + \Delta t u_{2,k}, \tag{59d}$$

$$\begin{aligned} x_{5,k+1} = x_{5,k} + \Delta t \Big\{ & \frac{\mu m}{I_z(l_\mathrm{r} + l_\mathrm{f})} [l_\mathrm{f} C_{\mathrm{S,f}}(g l_\mathrm{r} - u_{2,k} h_\mathrm{cg}) u_{1,k} + (l_\mathrm{r} C_{\mathrm{S,r}}(g l_\mathrm{f} + u_{2,k} h_\mathrm{cg}) \\ & - l_\mathrm{f} C_{\mathrm{S,f}}(g l_\mathrm{r} - u_{2,k} h_\mathrm{cg})] x_{6,k} - [l_\mathrm{f}^2 C_{\mathrm{S,f}}(g l_\mathrm{r} - u_{2,k} h_\mathrm{cg}) + l_\mathrm{r}^2 C_{\mathrm{S,r}}(g l_\mathrm{f} \\ & + u_{2,k} h_\mathrm{cg})] \frac{x_{5,k}}{x_{4,k}} \Big\}, \end{aligned} \tag{59e}$$

$$\begin{aligned} x_{6,k+1} = x_{6,k} + \Delta t \Big\{ & \frac{\mu}{x_4(l_\mathrm{r} + l_\mathrm{f})} [C_{\mathrm{S,f}}(g l_\mathrm{r} - u_{2,k} h_\mathrm{cg}) u_{1,k} - (C_{\mathrm{S,r}}(g l_\mathrm{f} + u_{2,k} h_\mathrm{cg}) \\ & + C_{\mathrm{S,f}}(g l_r - u_{2,k} h_\mathrm{cg})] x_{6,k} + [C_{\mathrm{S,r}}(g l_\mathrm{f} + u_2 h_\mathrm{cg}) l_\mathrm{r} - C_{\mathrm{S,f}}(g l_\mathrm{r} \\ & - u_{2,k} h_\mathrm{cg}) l_\mathrm{f}] \frac{x_{5,k}}{x_{4,k}} + x_{5,k} \Big\}. \end{aligned} \tag{59f}$$

C.2 4^{th} - ORDER RUNGE-KUTTA

In the UKF a more accurate method is used to solve the ODEs of the continuous-time model. The Runge-Kutta 4^{th} is a family of algorithms to numerically solve differential equations by estimating the slope of the function.

Starting from the same definition of the previous subsection, presented through the Equation (53), the system in this method evaluates as follow:

$$x_{k+1} = x_k + \frac{1}{6} h (k_1 + 2k_2 + 2k_3 + k_4), \tag{60a}$$

$$t_{k+1} = t_k + h, \tag{60b}$$

where $(k_1 + 2k_2 + 2k_3 + k_4)/6$ is the avarege slope over the interval h. Different Runge-Kutta method have different ways to calculate this average. For the UKF the way described in the Equation (60) was used. It is the weighted mean of four slopes evaluated in different places over the interval:

$$k_1 = f(h, x_n), \tag{61a}$$

$$k_2 = f\Big(t_n + \frac{h}{2}, x_n + h\frac{k_1}{2}\Big), \tag{61b}$$

$$k_3 = f\Big(t_n + \frac{h}{2}, x_n + h\frac{k_2}{2}\Big), \tag{61c}$$

$$k_4 = f(t_n + h, x_n + h k_3), \tag{61d}$$

where k_1 is the slope of the beginning of the interval and k_4 is the slope at the end of the interval. Also greater weights are given for the slopes at the midpoints k_2 and k_3. In the UKF algorithm used in this thesis, the transition of the points through the non-linear function is done through this method.